МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания

по выполнению лабораторных работ
по дисциплине
«ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ РАСПРЕДЕЛЕННЫХ ЦЕНТРОВ ОБРАБОТКИ ДАННЫХ»
для студентов специальности 10.05.03 Информационная безопасность автоматизированных систем, специализация Анализ безопасности информационных систем

Ставрополь
2026

Общие положения

Цель лабораторных работ – закрепление теоретических знаний и приобретение навыков защиты информации в распределенных ЦОД.

Лабораторная работа 1. Тема: Анализ архитектуры распределенных ЦОД и угроз информационной безопасности.

Задачи: Изучить типовые схемы РЦОД (гибридные, облачные, географически распределенные). Составить модель угроз для ЦОД (DDoS, инсайдеры, утечки данных).

1. Теоретическая часть

1.1. Типовые архитектуры РЦОД

1. **Гибридные ЦОД**
   - Сочетание локальных серверов и облачных ресурсов (AWS, Azure, OpenStack).
   - Пример: критичные данные хранятся локально, а вычислительные мощности масштабируются в облаке.
2. **Облачные ЦОД**
   - Полностью виртуализированная инфраструктура (IaaS, PaaS).
   - Пример: AWS EC2, Google Cloud, Yandex Cloud.
3. **Географически распределенные ЦОД**
   - Несколько дата-центров в разных регионах (репликация данных, отказоустойчивость).
   - Пример: CDN (Cloudflare, Akamai), банковские системы.

1.2. Основные угрозы безопасности РЦОД

Категория угроз	Примеры
DDoS-атаки	UDP-флуд, SYN-флуд, DNS-усиление (например, через уязвимые DNS-серверы).
Инсайдерские угрозы	Утечки данных сотрудниками, саботаж, непреднамеренные ошибки.
Утечки данных	Неправильная настройка S3-хранилищ, SQL-инъекции, перехват трафика.

Категория угроз	Примеры
Атаки на виртуализацию	VM Escape (CVE-2021-22061), взлом гипервизоров (Xen, VMware ESXi).
Физические угрозы	Кража оборудования, отключение электропитания, стихийные бедствия.

---

2. Лабораторная часть

2.1. Анализ архитектуры РЦОД

Задание:

1. Выбрать один из типов РЦОД (гибридный, облачный, распределенный).
2. Схематично изобразить его архитектуру (использовать **Draw.io**, **Lucidchart**).
3. Описать ключевые компоненты:
   - Серверы и хранилища.
   - Сетевые элементы (маршрутизаторы, межсетевые экраны).
   - Средства виртуализации (VMware, Hyper-V, KVM).

Пример схемы гибридного ЦОД:

Copy

Download

```
[Локальный сервер] ←→ [Файрволл] ←→ [Облако (AWS S3)]
      ↑
[Резервный ЦОД в другом регионе]
```

2.2. Построение модели угроз

Задание:

Используя методологию **STRIDE** (Microsoft), заполнить таблицу:

Угроза	Сценарий	Вероятность	Последствия
Spoofing	Подмена IP-адреса для обхода NAC.	Высокая	Доступ к данным.
Tampering	Изменение данных в хранилище S3.	Средняя	Потеря целостности.
Repudiation	Сотрудник отрицает удаление файлов.	Низкая	Юридические риски.

Дополнительно:

- Провести анализ уязвимостей для выбранной архитектуры с помощью **OpenVAS** или **Nessus**.
- Сгенерировать отчет в **Dradis Framework**.

---

3. Выводы и рекомендации

1. Какая архитектура РЦОД наиболее уязвима и почему?
2. Какие меры защиты следует применить (например, WAF для веб-приложений, сегментация сети)?
3. Как минимизировать риски инсайдерских угроз (DLP, аудит доступа)?

---

4. Контрольные вопросы

1. Чем отличаются гибридные и облачные ЦОД?
2. Назовите три самых опасных типа атак на РЦОД.
3. Как географическая распределенность повышает отказоустойчивость?

Лабораторная работа 2.

Тема: Нормативно-правовое регулирование защиты данных в РЦОД.

Задачи:

Анализ требований ФСТЭК, ФСБ, GDPR для ЦОД.

Разработка политики безопасности на основе ГОСТ Р 56939-2022.

1. Теоретическая часть

1.1. Обзор регулирующих документов

1.1.1. Требования ФСТЭК России

- **Приказ ФСТЭК №17** (2014):
  - Обязательная категоризация информационных систем (ИС).
  - Требования к СЗИ (антивирусы, МЭ, DLP).
  - Аудит безопасности раз в 3 года.
- **Приказ ФСТЭК №239** (2021):
  - Защита персональных данных в облачных сервисах.
  - Шифрование данных при передаче и хранении (ГОСТ 28147-89, ГОСТ Р 34.10-2012).

1.1.2. Требования ФСБ России

- **Приказ ФСБ №378** (2018):
  - Использование сертифицированных СКЗИ (КриптоПро, ViPNet).
  - Регламент обработки гостайны в ЦОД.
- **СРДК (Система распределения ключей доверия)** – для защиты каналов связи.

1.1.3. GDPR (General Data Protection Regulation, EC)

- **Статья 32**: Обязательное шифрование и псевдонимизация персональных данных.
- **Право на забвение**: Удаление данных по запросу субъекта.
- **Штрафы**: До 4% глобального оборота компании.

1.1.4. ГОСТ Р 56939-2022

- **Защита информации в облачных средах**:
  - Разделение ответственности между провайдером и клиентом.
  - Контроль физического доступа к серверам.
  - Мониторинг инцидентов 24/7.

2. Лабораторная часть

2.1. Сравнительный анализ требований

Задание:

Заполнить таблицу, выделив общие и уникальные требования:

Критерий	ФСТЭК №17	ФСБ №378	GDPR	ГОСТ Р 56939-2022
Шифрование данных	ГОСТ 28147-89	СКЗИ (КриптоПро)	AES-256	Рекомендовано TLS 1.3
Аудит доступа	Раз в 3 года	Ежедневные журналы	По запросу	Реальное время
Хранение данных в РФ	Обязательно	Обязательно	Нет	Обязательно

Вывод: В чем основные противоречия между GDPR и российскими нормами?

2.2. Разработка политики безопасности для ЦОД

Шаблон структуры (на основе ГОСТ Р 56939-2022):

markdown

Copy

Download

```
1. **Область применения**
   - ЦОД провайдера «X» с гибридной инфраструктурой (AWS + локальные серверы).

2. **Ответственные лица**
   - CISO (Chief Information Security Officer) – контроль выполнения политики.

3. **Меры защиты данных**
   - **Физическая безопасность**: Биометрия, видеонаблюдение в серверных.
   - **Криптография**: TLS 1.3 для передачи, ГОСТ 34.10-2021 для ЭП.
   - **Инцидент-менеджмент**: Регламент реагирования на утечки (срок – 1 час).
```

4. ****Соответствие стандартам****
   - ФСТЭК №17: Категория ИС – К1 (критическая).
   - GDPR: Назначен DPO (Data Protection Officer).

Задание:

1. Дополнить раздел «Меры защиты» для своего сценария (например, защита от DDoS).
2. Указать, какие СКЗИ и DLP-системы будут использоваться (например, **СёрчИнформ** или **RSA**).

---

3. Выводы и рекомендации

1. Какие стандарты сложнее всего реализовать в российских ЦОД? (*Пример: GDPR требует удаления данных по запросу, но ФЗ-152 обязывает хранить логи 6 месяцев*).
2. Как минимизировать юридические риски при работе с иностранными клиентами? (*Заключение отдельного соглашения о соответствии GDPR*).

---

4. Контрольные вопросы

1. Назовите 3 обязательных требования ФСТЭК для ЦОД.
2. В чем разница между ГОСТ Р 56939 и ISO 27001?
3. Какие штрафы предусмотрены за нарушение GDPR?

Лабораторная работа 3.

Тема: Настройка межсетевых экранов (NGFW) в распределенной среде.

Задачи: Конфигурация правил фильтрации в pfSense/iptables. Анализ трафика с помощью Wireshark.

1. Теоретическая часть

1.1. Основы NGFW (Next-Generation Firewall)

Отличие от традиционных МЭ:

- Глубокая инспекция пакетов (**DPI**).
- Блокировка угроз на уровне приложений (например, SQL-инъекций).
- Интеграция с системами **IDS/IPS** (Suricata, Snort).

Примеры NGFW:

- **pfSense** (на базе FreeBSD с GUI).
- **iptables** (Linux, низкоуровневый инструмент).
- Промышленные решения: **Palo Alto, FortiGate**.

1.2. Ключевые правила фильтрации

Тип правила	Пример	Применение в ЦОД
Блокировка по IP	`deny 192.168.1.100 → 10.0.0.0/24`	Защита внутренних серверов.
Фильтрация портов	`allow TCP 443 (HTTPS)`	Открытие доступа к веб-интерфейсам.
GeoIP-фильтры	`block RU, CN`	Защита от сканирования извне.
Анти-DDoS	`limit 100 connections/s per IP`	Защита от флуда.

2. Лабораторная часть

2.1. Настройка pfSense

Лабораторная среда:

- Виртуальная машина с **pfSense 2.7**.
- Две сети: LAN (`192.168.1.0/24`) и WAN (эмулирует интернет).

Задание 1: Создание правил фильтрации

1. Запретить весь входящий трафик из Интернета (WAN), кроме:
   - HTTP/HTTPS (порты 80, 443).
   - SSH (порт 22) только для IP администратора (`192.168.1.50`).
2. Включить **NAT** для доступа LAN в Интернет.

Решение (скриншоты + команды):

sh

Copy

Download

```
# В веб-интерфейсе pfSense (https://192.168.1.1):
1. Firewall → Rules → WAN → "Add".
2. Action: Pass, Protocol: TCP, Source: 192.168.1.50, Destination Port: 22.
3. Аналогично для портов 80, 443 (без ограничения IP).
4. Firewall → NAT → Outbound → "Hybrid mode".
```

2.2. Настройка iptables (Linux)

Задание 2: Заблокировать ICMP-запросы (ping) и разрешить только внутренний трафик.

sh

Copy

Download

```
# Сброс текущих правил
iptables -F

# Разрешить локальный трафик
iptables -A INPUT -i lo -j ACCEPT
```

```
# Блокировка ICMP
iptables -A INPUT -p icmp --icmp-type echo-request -j DROP

# Разрешить ESTABLISHED-соединения
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Сохранить правила
iptables-save > /etc/iptables.rules
```

2.3. Анализ трафика в Wireshark

Задание 3:

1. Запустить Wireshark на интерфейсе `eth0`.
2. Сгенерировать трафик (например, зайти на сайт через браузер).
3. Найти HTTP-запрос и проверить, не передаются ли пароли в открытом виде.

Фильтры для анализа:

- `http.request.method == "POST"` – поиск отправляемых форм.
- `tcp.port == 22` – мониторинг SSH-сессий.
- `ip.src == 192.168.1.100` – трафик от конкретного IP.

---

3. Выводы и рекомендации

1. **Какие правила критичны для ЦОД?**
   - Блокировка неиспользуемых портов (например, Telnet 23).
   - Логирование всех попыток доступа к МЭ.
2. **Как улучшить защиту?**
   - Включить **Suricata** в pfSense для детектирования атак.
   - Настроить **Fail2Ban** для автоматической блокировки IP после 5 попыток входа.

---

4. Контрольные вопросы

1. Чем отличается `iptables` от `nftables`?

2. Как заблокировать трафик из Китая в pfSense?
3. Какие поля в TCP-пакете анализирует Wireshark для детектирования DDoS?

Лабораторная работа 4. Тема: Защита от DDoS-атак в РЦОД. Задачи: Тестирование методов защиты (Anycast, Cloudflare, Arbor Peakflow). Симуляция атаки с помощью hping3.

1. Теоретическая часть

1.1. Типы DDoS-атак и их воздействие на ЦОД

Тип атаки	Цель	Пример
Объёмные (Volumetric)	Перегрузка канала связи	UDP-флуд, DNS-усиление
Протокольные (Protocol)	Эксплуатация уязвимостей сетевых протоколов	SYN-флуд, ICMP-флуд (Smurf)
Прикладные (Application)	Атака на уровень приложений	HTTP-флуд, Slowloris

1.2. Методы защиты от DDoS

1.2.1. Anycast

- **Принцип работы:** Разделение трафика между географически распределёнными серверами.
- **Плюсы:** Снижение нагрузки на один ЦОД, высокая отказоустойчивость.
- **Примеры:** CDN (Cloudflare, Akamai).

1.2.2. Cloudflare

- **Основные функции:**
  - Фильтрация трафика на edge-узлах.
  - Автоматическое блокирование ботнетов.
  - Защита от HTTP-флуда (WAF + Rate Limiting).

1.2.3. Arbor Peakflow

- **Использование в корпоративных ЦОД:**
  - Анализ трафика в реальном времени (NetFlow/sFlow).
  - Автоматическое перенаправление атакуемого IP через BGP.

---

2. Лабораторная часть

2.1. Настройка защиты на основе Anycast (теория + демонстрация)

Задание 1:

1. Развернуть тестовую сеть с двумя узлами (например, в GNS3).
2. Настроить BGP-анонсирование одного IP-адреса с двух серверов.
3. Проверить распределение трафика с помощью `traceroute`.

Пример команды для BGP (Cisco):

sh

Copy

Download

```
router bgp 65001
 neighbor 192.168.1.1 remote-as 65002
 network 203.0.113.1 mask 255.255.255.255
```

2.2. Тестирование Cloudflare

Задание 2:

1. Зарегистрировать домен и подключить его к Cloudflare.
2. Включить **Under Attack Mode** и проанализировать:
   - Как Cloudflare фильтрует трафик.
   - Какие запросы блокируются (проверить через логи).

Пример настройки Rate Limiting:

sh

Copy

Download

```
Правило: Блокировать IP при >100 запросах/сек к /api.
```

2.3. Симуляция DDoS-атаки с помощью hping3

Задание 3:

1. Запустить TCP-SYN флуд на тестовый сервер:

   sh

   Copy

   Download

   ```
   hping3 -S -p 80 --flood 192.168.1.100
   ```
2. Проверить нагрузку на сервере:
   sh

   Copy

   Download

   ```
   netstat -anp | grep SYN_RECV | wc -l
   ```
3. Включить защиту в pfSense (SYN Proxy) и сравнить результаты.

Варианты атак для теста:

- **UDP-флуд:** `hping3 --udp -p 53 --flood 192.168.1.100`
- **HTTP-флуд:** `siege -c 1000 -t 60S http://example.com`

---

3. Анализ результатов

1. **Эффективность Anycast:**
   - Уменьшилась ли нагрузка на основной ЦОД?
2. **Cloudflare vs Arbor Peakflow:**
   - Какой инструмент быстрее детектирует атаку?

3. **hping3:**

- Какие методы защиты сработали, а какие нет?

---

4. Выводы и рекомендации

1. **Для малых ЦОД:** Cloudflare + базовый Rate Limiting.
2. **Для крупных сетей:** Arbor Peakflow + BGP Blackholing.
3. **Обязательные меры:**

- Настройка **SYN Cookies** на Linux-серверах.
- Использование **IP reputation** (например, Spamhaus DROP List).

---

5. Контрольные вопросы

1. Чем отличается UDP-флуд от SYN-флуда?
2. Как Cloudflare определяет бот-трафик?
3. Зачем нужен BGP при использовании Anycast?

Лабораторная работа 5. Тема: Виртуальные частные сети (VPN) для безопасного доступа.

Задачи: Настройка IPsec/OpenVPN между узлами ЦОД. Анализ уязвимостей VPN (CVE-2023-36672).

1. Теоретическая часть

1.1. Типы VPN для ЦОД

Тип VPN	Протоколы	Применение в ЦОД
IPsec	IKEv2, AES-256-GCM, SHA-384	Защищённое соединение между филиалами.

Тип VPN	Протоколы	Применение в ЦОД
OpenVPN	TLS, UDP/TCP	Доступ удалённых сотрудников.
WireGuard	ChaCha20, Poly1305	Высокопроизводительные туннели.

1.2. Ключевые уязвимости VPN

1. **CVE-2023-36672** (уязвимость в OpenVPN):
   - **Описание:** Ошибка в обработке буфера, приводящая к RCE (Remote Code Execution).
   - **Условия эксплуатации:** Версии OpenVPN < 2.6.6.
   - **Метод защиты:** Обновление до актуальной версии.
2. **Другие риски:**
   - Слабые алгоритмы шифрования (например, SHA-1).
   - Утечки DNS при разрыве соединения.

---

2. Лабораторная часть

2.1. Настройка IPsec-туннеля (Linux, pfSense)

Задание 1: Создать туннель между двумя ЦОД (например, Москва и Санкт-Петербург).

Вариант А: Настройка на Linux (strongSwan)

sh

Copy

Download

```
# Установка strongSwan
sudo apt install strongswan

# Конфигурация /etc/ipsec.conf
conn msk-spb
   left=95.165.1.1      # Москва
   right=95.165.2.1     # СПб
   leftsubnet=10.0.1.0/24
   rightsubnet=10.0.2.0/24
```

```
ike=aes256-sha384-modp2048!
esp=aes256gcm16!
keyexchange=ikev2
auto=start
```

Вариант B: Настройка в pfSense

1. **VPN → IPsec → Add P1**:
   - **Phase 1:** IKEv2, AES-256-GCM, DH Group 20.
2. **Phase 2:**
   - **Encryption:** AES-256-GCM.
   - **PFS:** Enabled (Group 20).

Проверка:

sh

Copy

Download

```
ping 10.0.2.1  # С московского узла
```

2.2. Развёртывание OpenVPN

Задание 2: Настроить OpenVPN для удалённого доступа к ЦОД.

1. **Генерация сертификатов:**

   sh

   Copy

   Download

   ```
   ./easyrsa build-ca
   ./easyrsa gen-req server nopass
   ./easyrsa sign-req server server
   ```

2. **Конфигурация сервера (**`server.conf`**):**
   ini

   Copy

   Download

```
proto udp
dev tun
ca /etc/openvpn/ca.crt
cert /etc/openvpn/server.crt
cipher AES-256-GCM
tls-version-min 1.3
```

3. **Подключение клиента:**
   sh

   Copy

   Download

   ```
   openvpn --config client.ovpn
   ```

   2.3. Анализ уязвимостей (CVE-2023-36672)

   Задание 3:

1. Проверить версию OpenVPN:

   sh

   Copy

   Download

   ```
   openvpn --version | grep "OpenVPN"
   ```

2. Если версия < 2.6.6, смоделировать атаку:

- Использовать **Metasploit-модуль** `exploit/unix/openvpn_auth_bypass`.

3. Применить заплатки:
   sh

   Copy

   Download

   ```
   sudo apt update && sudo apt upgrade openvpn
   ```

   Дополнительно:

- Проверить VPN на утечки с помощью **DNSLeakTest** (https://www.dnsleaktest.com/).

---

3. Выводы и рекомендации

1. **IPsec vs OpenVPN:**
   - IPsec лучше для site-to-site, OpenVPN — для удалённых пользователей.
2. **Защита от CVE-2023-36672:**
   - Обновлять ПО, отключить устаревшие алгоритмы.
3. **Дополнительные меры:**
   - Использовать **MFA** (например, Google Authenticator для OpenVPN).
   - Включить **логирование** всех подключений.

---

4. Контрольные вопросы

1. Какой протокол VPN обеспечивает лучшую производительность?
2. В чём разница между IKEv1 и IKEv2?
3. Как предотвратить утечку DNS в OpenVPN?

Лабораторная работа 6. Тема: Развертывание SIEM-системы (ELK Stack, Splunk). Задачи: Настройка сбора логов с серверов. Создание правил детектирования атак.

1. Теоретическая часть

1.1. Обзор SIEM-систем

SIEM (Security Information and Event Management) — система для сбора, корреляции и анализа событий безопасности.

Решение	Плюсы	Минусы
ELK Stack	Бесплатно (OpenSource), гибкость	Требует ручной настройки
Splunk	Готовые дашборды, ML-аналитика	Дорогая лицензия

1.2. Ключевые компоненты

- **Logstash/Fluentd** — сбор и обработка логов.
- **Elasticsearch** — хранение и поиск данных.
- **Kibana/Splunk Web** — визуализация.
- **Правила корреляции** — детектирование аномалий (например, множественные failed login attempts).

---

2. Лабораторная часть

2.1. Развертывание ELK Stack

Задание 1: Настроить сбор логов с веб-сервера (Nginx/Apache) и системных событий.

Шаг 1: Установка Elasticsearch + Kibana

bash

Copy

Download

```
# Для Ubuntu/Debian
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-7.x.list
sudo apt update && sudo apt install elasticsearch kibana
sudo systemctl start elasticsearch kibana
```

Шаг 2: Настройка Filebeat для сбора логов

1. Конфигурация `/etc/filebeat/filebeat.yml`:

yaml

Copy

Download

```
output.elasticsearch:
  hosts: ["localhost:9200"]
filebeat.inputs:
 - type: filestream
   paths:
```

```
    - /var/log/nginx/access.log
    - /var/log/auth.log
```

2. Запуск:

bash

Copy

Download

```
sudo filebeat setup --pipelines --modules system,nginx
sudo systemctl start filebeat
```

Шаг 3: Визуализация в Kibana (http://localhost:5601)

- Создать индекс `filebeat-*` в **Management → Stack Management**.
- Импортировать готовые дашборды для Nginx (например, "Nginx overview").

2.2. Настройка Splunk

Задание 2: Настроить мониторинг атак на SSH-сервер.

Шаг 1: Установка Splunk Free

bash

Copy

Download

```
wget -O splunk.tar.gz "https://download.splunk.com/products/splunk/releases/9.x/linux/splunk-9.x.x-linux-x86_64.tar.gz"
tar -xzvf splunk.tar.gz
./splunk/bin/splunk start --accept-license
```

Шаг 2: Добавление данных

1. В веб-интерфейсе (`http://localhost:8000`):
   - **Settings → Data inputs → Files & directories** → Добавить `/var/log/auth.log`.
2. Создать правило детектирования (Search & Reporting):

sql

Copy

Download

```
index=main sourcetype=auth.log "Failed password"
| stats count by src_ip
| where count > 5
```

2.3. Создание правил детектирования атак

Для ELK (KQL-запрос в Kibana):

sql

Copy

Download

```
event.dataset: "auth.log" and message: "Failed password"
| aggregate count() by source.ip
| where count > 3
```

Для Splunk (SPL):

sql

Copy

Download

```
index=main sourcetype=nginx access_denied
| top 10 clientip
```

Примеры детектируемых атак:

- **Brute Force SSH:** Множественные failed login attempts.
- **SQL-инъекции:** Наличие `' OR 1=1 --` в логах Nginx.
- **Сканирование портов:** Логи firewall с множеством SYN-пакетов.

---

3. Анализ результатов

1. **ELK vs Splunk:**
   - Какой инструмент быстрее обрабатывает большие объемы логов?

- Где удобнее настраивать правила?

2. **Ложные срабатывания:**

- Какие события ошибочно попали в детекцию (например, легитимные сканы SEO-ботов)?

---

4. Выводы и рекомендации

1. **Для небольших ЦОД:** ELK Stack (бесплатно, но требует времени).
2. **Для корпоративных сетей:** Splunk Enterprise (готовые интеграции с Cisco/Windows).
3. **Обязательные правила:**

- Мониторинг аномальных login-попыток.
- Алерт при массовом удалении файлов.

---

5. Контрольные вопросы

1. Какой компонент ELK отвечает за хранение данных?
2. Как фильтровать логи веб-атак в Splunk?
3. Зачем нужны **pipelines** в Filebeat?

Лабораторная работа 7. Тема: Анализ вторжений с помощью Suricata. Задачи: Захват и анализ подозрительного трафика. Написание своих правил для Suricata.

1. Теоретическая часть

1.1. Suricata — ключевые возможности

- **Многофункциональность:** IDS/IPS, анализ трафика в реальном времени.
- **Поддержка протоколов:** HTTP, DNS, TLS, SSH и др.
- **Форматы правил:** Совместимость с **Snort**, поддержка **Emerging Threats**.

1.2. Структура правил Suricata

yaml

Copy

Download

```
alert http any any -> any any (
  msg:"SQL Injection Attempt";
  flow:to_server;
  content:"' OR 1=1";
  nocase;
  sid:1000001;
  rev:1;
)
```

- **Поля:**
  - action (alert, drop, reject).
  - protocol (http, tls, dns).
  - source/destination (IP, порты).
  - content (шаблон для поиска в пакетах).

---

2. Лабораторная часть

2.1. Установка и настройка Suricata

Задание 1: Развернуть Suricata в режиме IDS.

Шаг 1: Установка (Ubuntu)

bash

Copy

Download

```
sudo add-apt-repository ppa:oisf/suricata-stable
sudo apt update && sudo apt install suricata -y
```

*Шаг 2: Настройка (**/etc/suricata/suricata.yaml**)*

yaml

Copy

Download

```
default-rule-path: /var/lib/suricata/rules
rule-files:
  - suricata.rules
  - custom.rules  # Для своих правил
```

Шаг 3: Запуск

bash

Copy

Download

```
sudo suricata -c /etc/suricata/suricata.yaml -i eth0
```

2.2. Захват и анализ трафика

Задание 2: Выявить подозрительную активность в pcap-файле.

Вариант A: Анализ готового pcap

1. Скачать образец трафика с атакой (например, из Malware-Traffic-Analysis.net).
2. Запустить анализ:
   bash

   Copy

   Download

   ```
   sudo suricata -r attack.pcap -l /var/log/suricata/
   ```
3. Проверить логи в `/var/log/suricata/fast.log`:
   bash

   Copy

   Download

   ```
   grep "ET EXPLOIT" /var/log/suricata/fast.log
   ```

Вариант B: Захват в реальном времени

1. Запустить генерацию атак (например, `nmap -sV 192.168.1.1`).

2. Проверить алерты:
   bash

   Copy

   Download

   ```
   tail -f /var/log/suricata/eve.json | jq 'select(.alert)'
   ```

2.3. Создание своих правил

Задание 3: Написать правило для детектирования:

- **Сканирования портов** (Nmap).
- **Попытки входа по SSH с подбором пароля.**

Правило 1: Детектирование Nmap

yaml

Copy

Download

```
alert tcp any any -> any any (
  msg:"Nmap TCP Scan Detected";
  flow:to_server;
  content:"|00 00 00 00|";  # Сигнатура Nmap
  sid:1000002;
  rev:1;
)
```

Правило 2: Bruteforce SSH

yaml

Copy

Download

```
alert tcp any any -> any 22 (
  msg:"SSH Bruteforce Attempt";
  flow:established;
  content:"Failed password";
```

```
  threshold: type threshold, track by_src, count 5, seconds 60;
  sid:1000003;
  rev:1;
)
```

Проверка:

1. Разместить правила в /var/lib/suricata/rules/custom.rules.
2. Перезапустить Suricata:
   bash

   Copy

   Download

   ```
   sudo systemctl restart suricata
   ```

---

3. Анализ результатов

1. **Эффективность правил:**
   - Сколько ложных срабатываний вызвало правило для Nmap?
2. **Сравнение с стандартными правилами:**
   - Какие атаки детектирует emerging-exploit.rules, но не детектируют ваши?

---

4. Выводы и рекомендации

1. **Оптимизация правил:**
   - Использовать threshold для уменьшения ложных срабатываний.
2. **Интеграция с SIEM:**
   - Настроить отправку логов в **ELK** или **Splunk**.
3. **Готовые наборы правил:**
   - Подключить **ET Open Rules** (sudo suricata-update).

---

5. Контрольные вопросы

1. Чем отличается `flow:to_server` от `flow:from_client`?
2. Как детектировать HTTPS-атаки, если трафик зашифрован?
3. Зачем нужен параметр `sid`?

Лабораторная работа 8. Тема: Разведка в распределенных ЦОД. Задачи: Использование Nmap, Shodan для сканирования. Построение карты сети.

1. Теоретическая часть

1.1. Инструменты для разведки

- **Nmap** – мощный сканер сетей, позволяющий обнаруживать хосты, открытые порты, сервисы и ОС.
- **Shodan** – поисковая система для интернет-устройств, помогает находить открытые сервисы, уязвимые системы и данные о ЦОД.

1.2. Основные методы разведки

- **Сканирование портов** (TCP/UDP)
- **Определение сервисов** (версии ПО)
- **ОС-фингерпринтинг** (определение операционной системы)
- **Построение карты сети** (топология, активные хосты)

---

2. Лабораторная часть

Задача 1: Сканирование сети с помощью Nmap

Цель: Обнаружить активные хосты, открытые порты и сервисы в заданной подсети.

Шаги:

1. **Обнаружение хостов** (Ping-сканирование):

bash

Copy

Download

```
nmap -sn 192.168.1.0/24
```

- -sn – проверка активности хостов без сканирования портов.

2. **Сканирование портов** (TCP-сканирование):
   bash

   Copy

   Download

   ```
   nmap -sV -p 1-1000 192.168.1.1
   ```

   - -sV – определение версий сервисов.
   - -p 1-1000 – проверка первых 1000 портов.
3. **Агрессивное сканирование** (с определением ОС):
   bash

   Copy

   Download

   ```
   nmap -A 192.168.1.1
   ```

   - -A – включает ОС-детекцию, скрипты и трассировку.
4. **Сохранение результатов в файл:**
   bash

   Copy

   Download

   ```
   nmap -oN scan_result.txt 192.168.1.1
   ```

Вывод:

Составить таблицу обнаруженных хостов, открытых портов и сервисов.

Задача 2: Использование Shodan для разведки ЦОД

Цель: Найти информацию о публичных серверах, маршрутизаторах и уязвимых системах в ЦОД.

Шаги:

1. **Поиск по ключевым словам** (например, "datacenter", "Cisco"):
   - Перейти на Shodan.io
   - Ввести запрос:

     Copy

     Download

     ```
     org:"Amazon" product:"Apache"
     ```

     (поиск серверов Apache в Amazon)

2. **Фильтрация по портам и странам:**

   Copy

   Download

   ```
   port:22 country:US
   ```

   (поиск SSH-серверов в США)

3. **Анализ результатов:**
   - IP-адреса, сервисы, геолокация.
   - Проверка на наличие уязвимостей (CVE).

Вывод:

Составить список найденных IP с открытыми сервисами и потенциальными уязвимостями.

Задача 3: Построение карты сети

Цель: Визуализировать топологию сети на основе данных сканирования.

Шаги:

1. **Использование Zenmap** (GUI для Nmap):
   - Запустить сканирование с опцией -O (определение ОС).
   - Перейти во вкладку **Topology** для визуализации.
2. **Ручное построение схемы** (на основе данных Nmap/Shodan):
   - Использовать **draw.io** или **Lucidchart**.
   - Указать:
     - Основные маршрутизаторы
     - Серверы (веб, БД)
     - Межсетевые экраны

Вывод:

Представить схему сети с пояснениями.

---

3. Выводы и рекомендации

1. Nmap эффективен для локальной разведки, Shodan – для анализа публичных ЦОД.
2. Обнаруженные открытые порты могут быть уязвимыми (SSH, RDP).
3. Рекомендуется закрывать ненужные сервисы и использовать фаерволы.

Форма отчета:

- Результаты сканирования (таблицы, скриншоты).
- Карта сети.
- Выводы по безопасности.

1. Directive 95/46/EC ofthe European Parliamentandof the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data
2. Directive 2002/58/EC of the European Parliament and of the Councilof 12 July 2002concerning the processing of personal data and the protection of privacy in the electroniccommunications sector (Directive on privacy and

electronic communications)
3. Article 29 Working Party Working document on determining the international application of EU data protection law to personal data processing on the Internet by non-EU based web sites (WP56 of 30 May 2002)
4. Доктрина информационной безопасности РФ 9 сентября 2000 г.

5. Федеральный закон от 27.07.2006 г. №149-ФЗ «Об информации, информационных технологиях и о защите информации».
6. Федеральный закон Российской Федерации от 27.07.2006 г. N 152-ФЗ «О персональных данных»
7. Ярочкин В.И. Информационная безопасность. Учебник для вузов.– М.: Академический Проект, Мир, 2004. – 544 с.
8. Левин В.К. Защита информации в информационно-вычислительных системах и сетях. Программирование. - 1994. - N5. - с. 5- 16.
9. Баранов А.П «Можно ли защитить в «облаке» конфиденциальную информацию?» \\ Системы высокой доступности, 2012. Т. 8. № 2. С. 12—15
10. Шаньгин В.Ф. Защита компьютерной информации. Эффективные методы и средства. − М.: ДМК Пресс, 2008. – 544 с.
11. Белогрудов В.М. Облачные вычисления - достоинства и недостатки \\ http://www.smart-cloud.org/sorted-articles/44-for-all/96-cloud-computing-plus-minus. 03.03.2012
12. Кондрашин М.С. Безопасность облачных вычислений \\ http://www.pcmag.ru/solutions/detail.php?ID=38248. 15.02.2010
13. Мельников В.И. Защита информации в компьютерных системах. – М.: Финансы и статистика, Электронинформ, 1997. – с. 123 – 128.
14. Угрозы информационной безопасности \\ http://www.internet- technologies.ru/articles/article_1147.html. 3.10.2007
15. Интернет-Университет Информационных Технологий \\ http://www.intuit.ru/department/security/secbasics/1/secbasics_1.html
16. Хореев П.В. Методы и средства защиты информации в компьютерных системах. – М.: издательский центр "Академия", 2005. – с. 205.
17. Фатьянов А.А. Концептуальные основы обеспечения безопасности на современном этапе \\ Безопасность информационных технологий – 1999 - №1. С.26-40.
18. Приходько А.Я. Словарь-справочник по информационной безопасности. М.: СИНТЕГ, 2001. 124 с.
19. Гольев Ю.И., Ларин Д.А., Тришин А.Е., Шанкин Г.П. Криптографическая деятельность в США XVIII-XIX веков. \\ Защита информации. Конфидент. №6, 2004, с.68-74.
20. Тейер Т., Липов М., Нельсон Э. Надежность программного

обеспечения. - Пер. с англ. - М.: Мир, 1981.

21. Липаев В.В. Надежность программных средств. - М.: Синтег, 1998.

22. Бабаш А.В., Гольев Ю.И., Ларин Д.А., Шанкин Г.П. О развитии криптографии в XIX веке // Защита информации. Конфидент. №5, 2003, с.90-96.

23. www.ec.europa.eu/justice

24. www.enisa.europa.eu