

ВВЕДЕНИЕ В КИБЕРБЕЗ

05. Средства Защиты Информации

СЗИ



Это **совокупность технических, программных, организационных и смешанных** средств, направленных на обеспечение безопасности информации от различных угроз. Они призваны обеспечить конфиденциальность, целостность и доступность данных

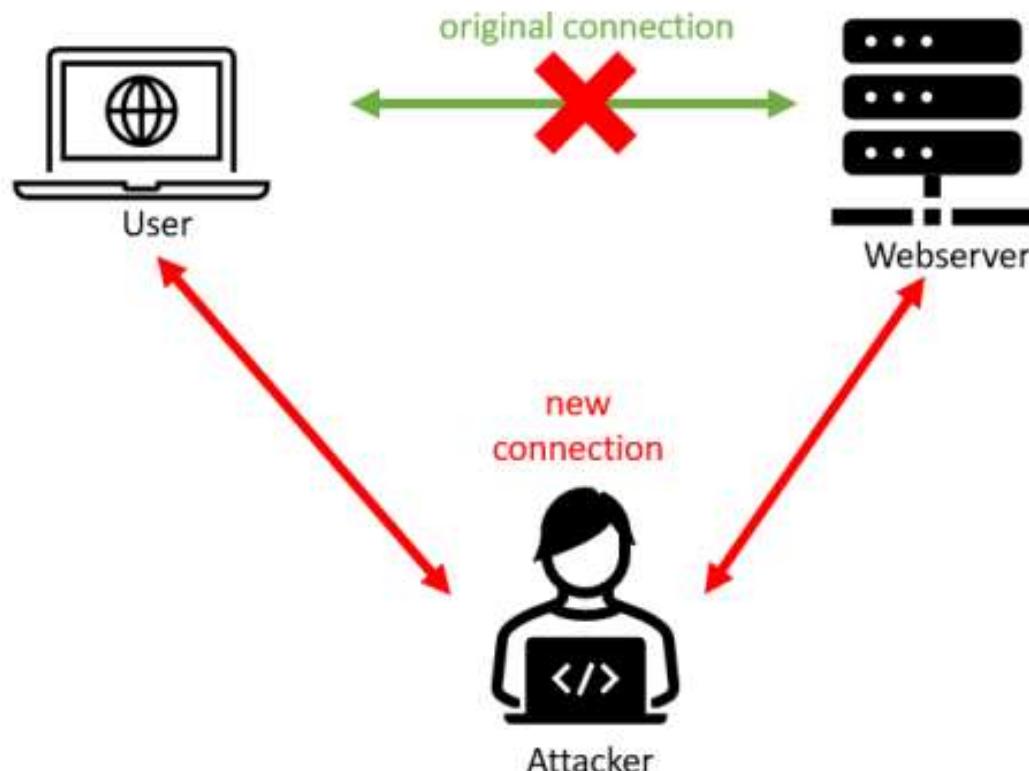
Нас интересуют **программные средства защиты информации**.

Триада СIA

- » **Доступность** (свойство информации быть доступной и готовой к использованию по запросу авторизованного субъекта, имеющего на это право)
- » **Целостность** (свойство сохранения правильности и полноты активов)
- » **Конфиденциальность** (свойство информации быть недоступной или закрытой для неавторизованных лиц, сущностей или процессов)

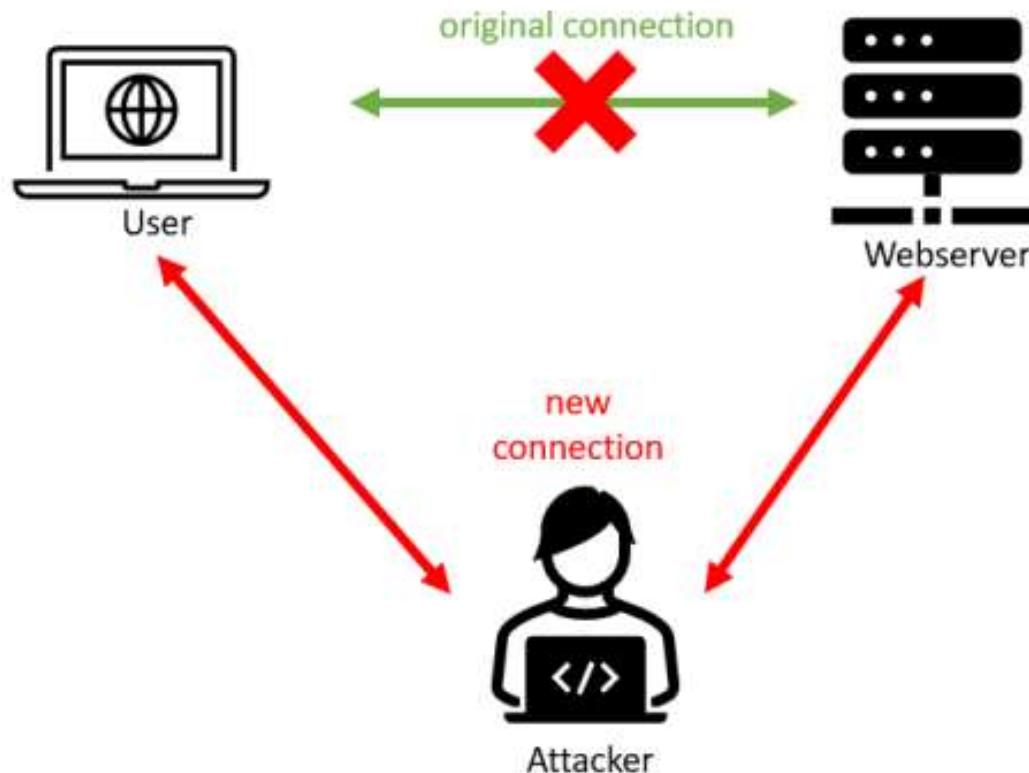
Конфиденциальность

MiM не может
прочитать
информацию
переданную User



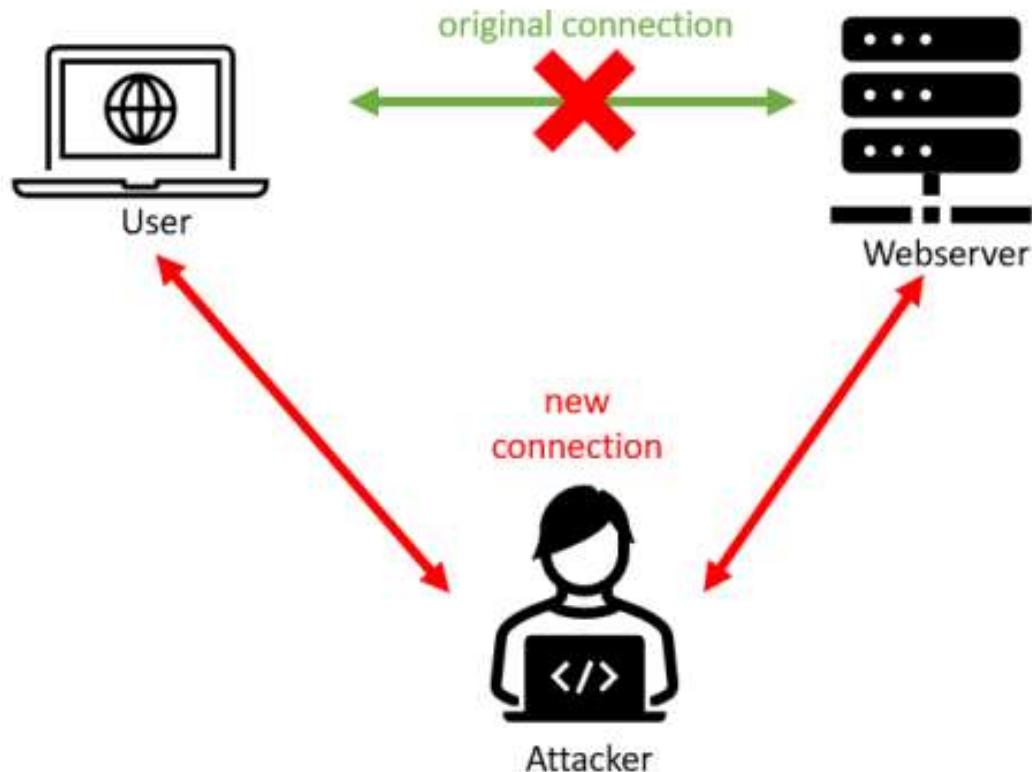
Целостность

Информация, переданная User, не была изменена несанкционированным образом



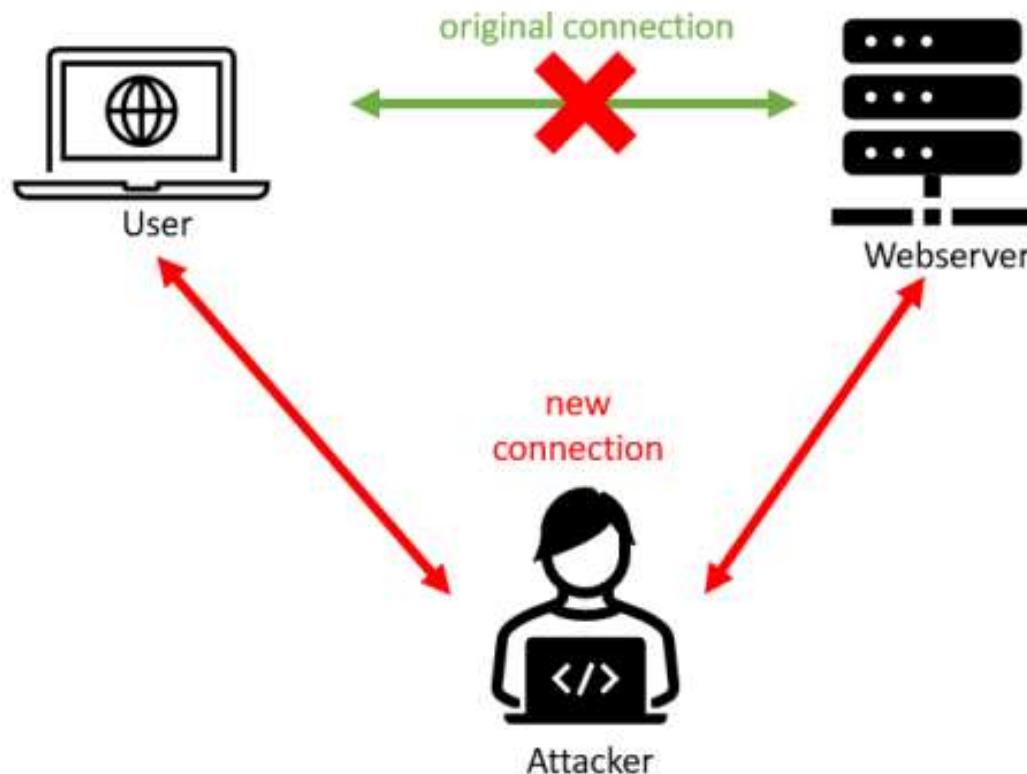
Неотказуемость

После того как User купил что-то и потратил деньги, он не может заявить банку, что деньги он не тратил



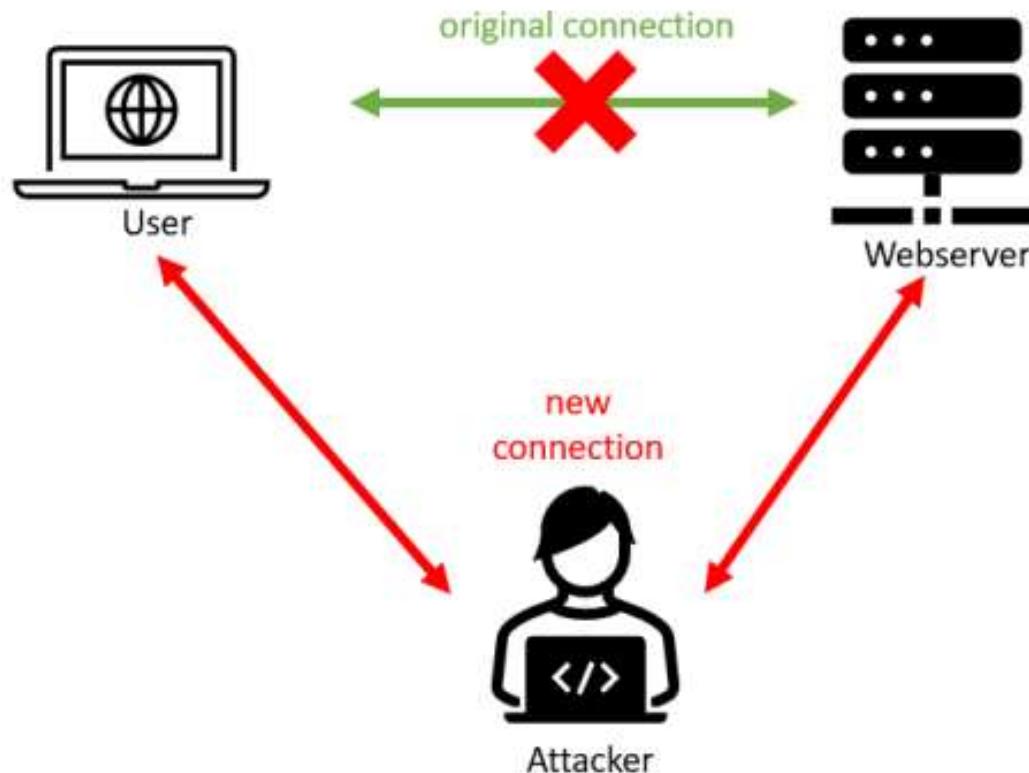
Авторизация

User не может зайти в админ панель, если он не админ. Не путать с аутентификацией



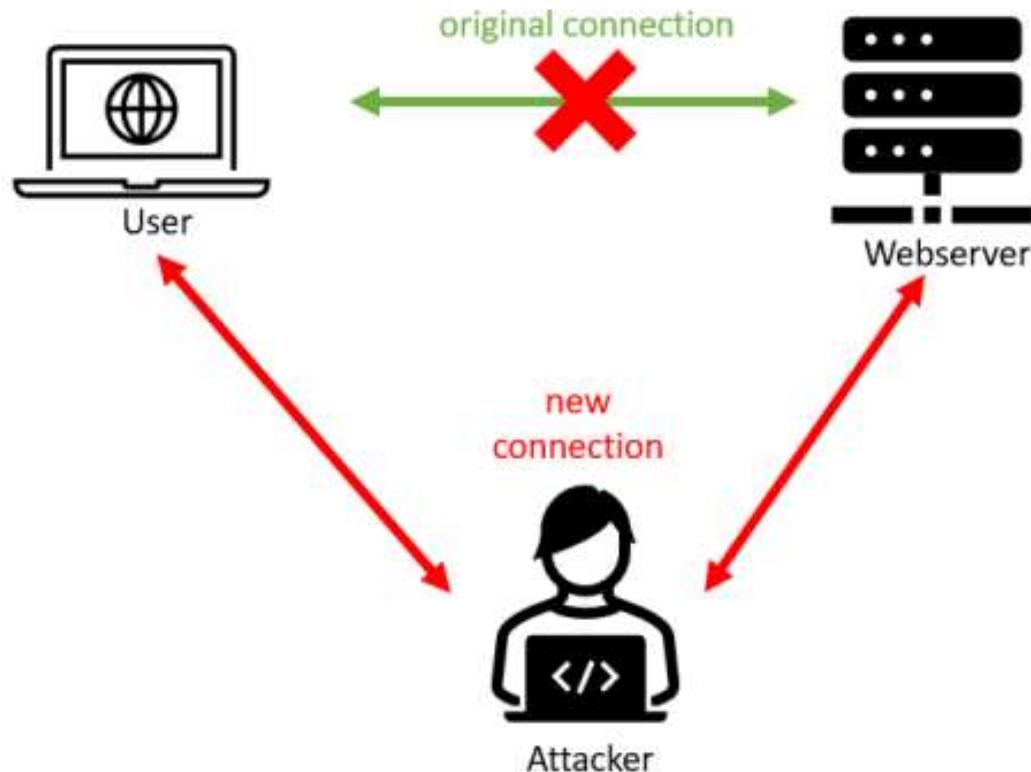
Надежность

User не может зайти в админ панель, если он не админ. Не путать с аутентификацией



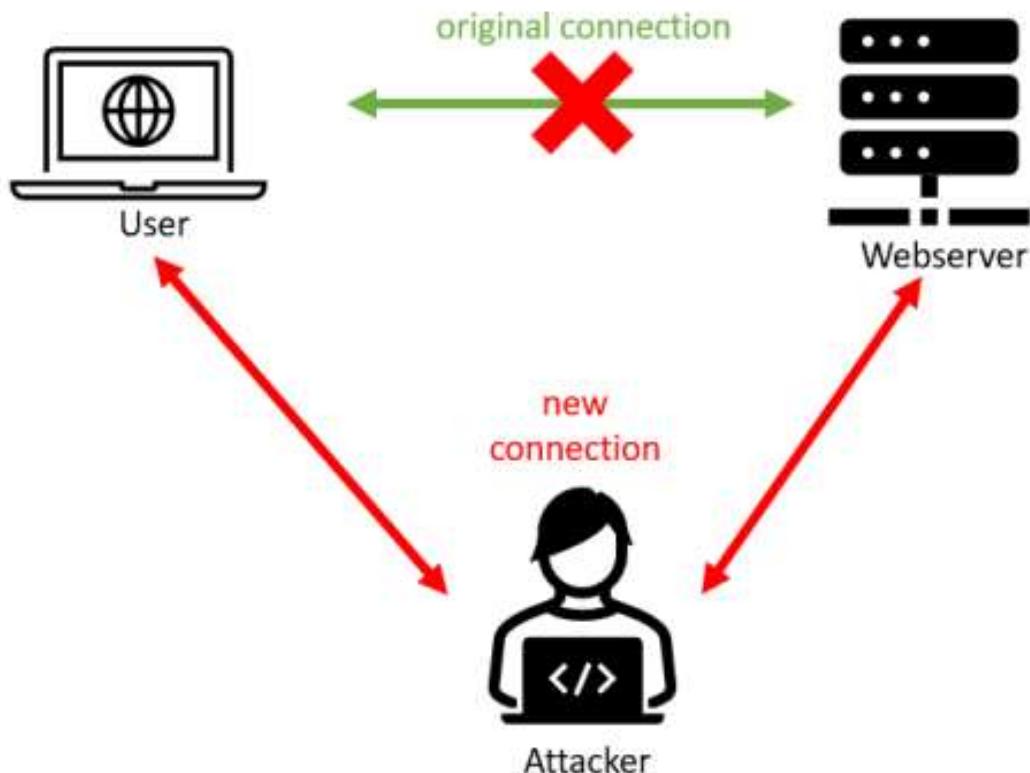
Доступность

Информация доступна и может быть изменена своевременно и с доступными в данный момент правами. MiM не может прервать соединение User и Webserver



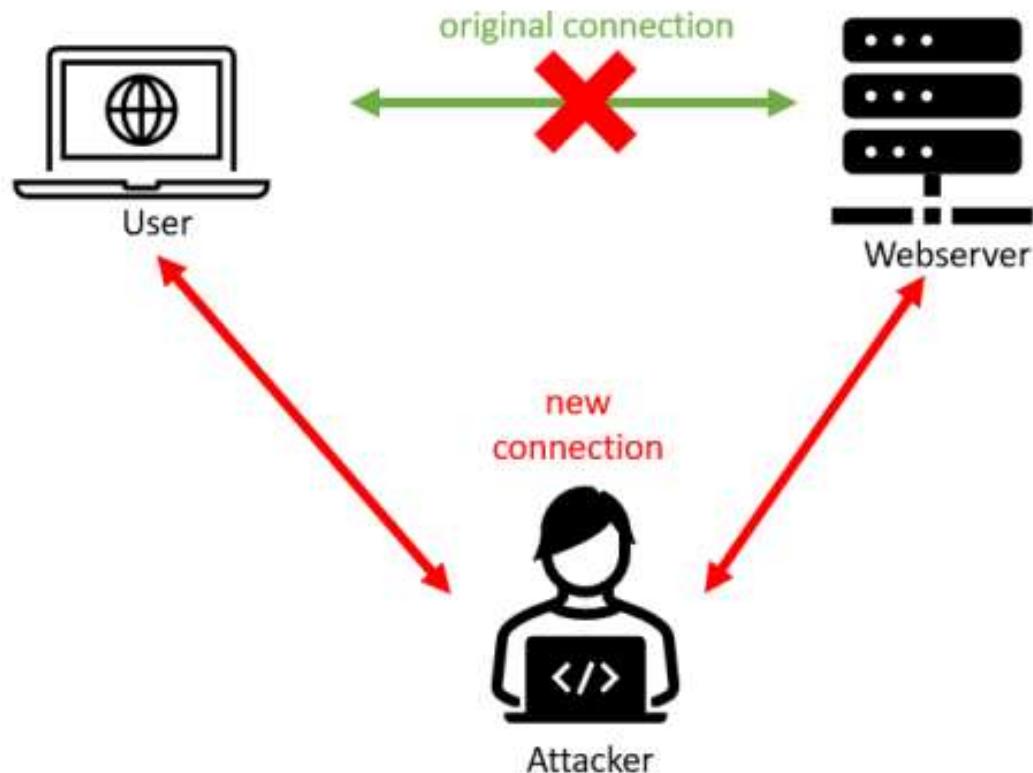
Аутентификация

Соотношение личности/роли. Это определение может быть выполнено несколькими различными способами, но обычно оно основано на комбинации. Например:



Аутентификация

Соотношение личности/роли. Это определение может быть выполнено несколькими различными способами, но обычно оно основано на комбинации
Например:
- Пароль + 2FA



Как нам обеспечить защиту данных?

Мониторинг событий



SIEM (Security Information & Event Management) – Средства сбора и анализа событий информационной безопасности

> **Основная цель SIEM:** собирать логи и события со всех элементов инфраструктуры и превращать данные в понятные инциденты, пригодные для расследования

SIM

SEM

SIEM

SIM (*Security Information Management*) —
управление и хранение логов

SEM (*Security Event Management*) — корреляция
событий и реагирование в реальном времени

Функции SIEM

- » Сбор и агрегация логов
файерволы, антивирусы, DLP, IDS/IPS, сети...
- » Нормализация и фильтрация
Приведение всех логов к единому формату, удаление шумовых событий
- » Корреляция событий и аналитика
Связывание событий из разных источников по шаблонам: неудачные входы, сетевые сканирования
- » Поведенческий анализ (UEBA User and Entity Behavior Analytics)
Выявление аномалий на основе изучения нормального поведения пользователей и объектов



Функции SIEM

» Оповещения и инцидент-менеджмент

Автогенерация предупреждений и создание карты инцидента для рассмотрения

» Отчеты и визуализация

Дашборды и отчеты для быстрого обзора событий

» Корреляция событий и аналитика

Длительное хранение логов и возможность ретроспективного анализа инцидентов

Зачем использовать SIEM?

- » **Централизованное мониторинг**
Единый обзор всех событий безопасности
- » **Обнаружение сложных атак**
Связывание событий из разных систем позволяет быстро реагировать
- » **Снижение ложных срабатываний**
Фильтрация, UEBA и аналитика помогают устраниить шум
- » **Соблюдение нормативов**
Автоматические отчеты для Audit/GDPR/PCI-DSS и др
- » **Ускоренная реакция на инциденты**
SIEM помогает сокращать время реакции благодаря детализации инцидента

СЗИ

Современные средства анализа защищенности обеспечивают:

- » Автоматическое сканирование в сетевом режиме
- » Автоматическое сканирование в «агентском» режиме
- » Выявление изменений конфигурации
- » Проверку простых паролей, типичных недостатков конфигурирования

Популярные SIEM

Dashboard Events

Search

Jan-Feb

2024

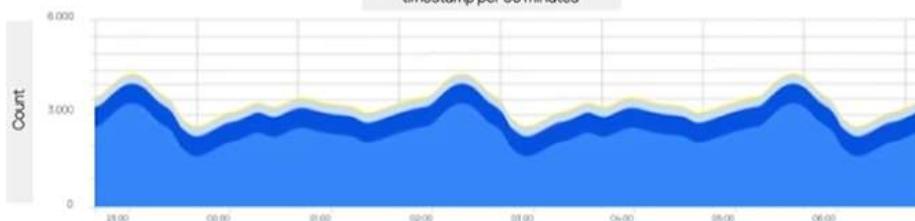
Explore agent

Generate report

Total
176957Level 12 or above alerts
9Authentication failure
33882Authentication success
45

Alerts level evolution

timestamp per 60 minutes



MITRE ATT&CK



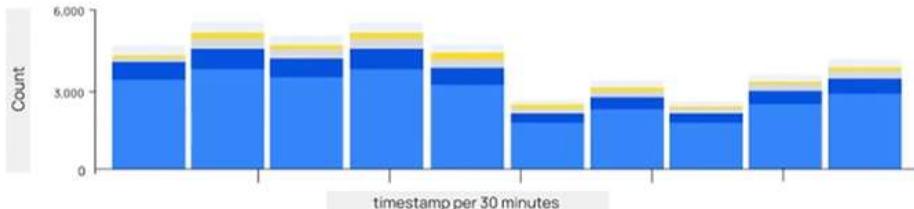
- Password Guessing
- SSH
- Brute Force
- Valid Accounts
- System Binary Proxy...
- Account Access Re...

Top 5 agents



- macOS
- Centos
- RHEL7
- Windows
- Debian

Alerts evolution - Top 5 agents



- macOS
- Centos
- RHEL7
- Windows
- Debian

Security alerts

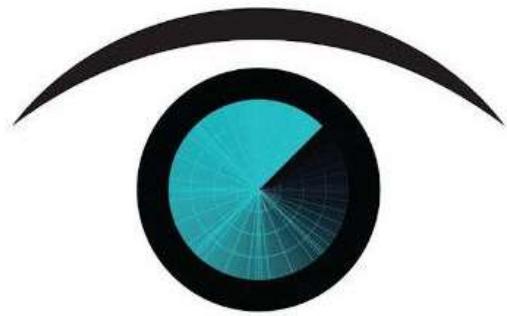
Time	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
> Jan 22, 2024 @ 09:55:20.518	004	Windows	T1218	Defense Evasion	Signed Script Proxy Execution: C:\Windows\...	10	255563

Security  Onion



MaxPatrol

SIEM



RUSIEM

Всё под контролем

Примеры правил корреляции



- > 3 или более неудачных логинов за 1 минуту с одного IP → возможная атака типа brute-force.
- > 15 событий DROP/DENY с файрвола от одного IP за минуту → попытка сканирования сети.
- > 7 событий IDS-alarms с одного IP → подозрение на сетевую атаку.

Также правило: заражение без успешного удаления сигнатуры в течение часа — возможный пропущенный вирус

Сканеры защищенности исходников

Сканеры защищенности исходников

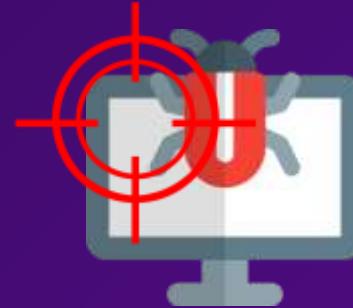
Современные сканеры исходников предоставляют:

- » Выявление уязвимостей и закладок в приложениях на этапе их разработки и приемки
 - » Автоматическое сканирование
 - » Важное преимущество перед сетевыми сканерами защищенности – возможность быстрого анализа исходного кода с целью выявления ошибок
- !!** Проблема сканнеров исходного кода – огромное количество ложных срабатываний

Противодействие сетевым атакам

Средства выявления/предотвращения сетевых атак:

- » Средства обнаружения вторжений – СОВ
- » Средства обнаружения атак – СОА
- » Intrusion detection system – IDS
- » Intrusion prevention system – IPS



Задача IPS – «очистка» траффика

Задача IDS – выявление и уведомление об атаках
сигнатурный анализ; поведенческий анализ; выявление
аномалий; антивирусная защита; защита от DDoS

Основа СЗИ в Linux

Контроль доступа



Правила и политики, которые ограничивают доступ к конфиденциальной информации этим людям и / или системам, которые не должны иметь туда доступ. Необходимость знать может определяться по пользователю, например, именем человека или серийным номером компьютера, а также роль, которую имеет человек, например, является менеджером или специалистом по компьютерной безопасности.

Контроль доступа

Стандартная модель UNIX/Linux, основанная на правах владельца файла. Так же мы дополнительно можем использовать MAC

Утилиты:

```
$ chmod XXX file
```

```
$ chown XXX:XXX file
```

Особенности: доступ к файлу зависит от прав владельца и группы

Контроль доступа

Mandatory Access Control (MAC) — это система контроля доступа, при которой **права доступа определяются политиками безопасности, а не владельцем объекта** (в отличие от классической модели).

!!! Пользователи не могут изменить права доступа к файлам

MAC может запретить даже root'у доступ, если политики безопасности это запрещают

Linux поддерживает несколько систем MAC. Например **SELinux** - самая мощная и гибкая MAC-система

Контроль доступа

Каждый объект (файл, процесс, сокет, да что угодно) имеет **контекст безопасности** (выделен красным)

```
$ ls -l домашка
$ -rw-r--r--. root root system_u:object_r:httpd_sys_content_t:s0
$ # [пользователь]:[роль]:[тип]:[уровень безопасности]
$ # пользователь system_u - системный пользователь SELinux, не обязательно рут
```

Совет: Использование МАС значительно **повышает безопасность**.
Только вот настройка...

Контроль доступа

Основные команды SELinux

```
$ getenforce # Показать режим SELinux (Enforcing/Permissive/Disabled)  
$ setenforce 0/1 # Включить/выключить проверку политик  
$ sestatus # Статус SELinux  
$ semanage # Управление политиками и контекстами  
.....
```

Enforcing: Политики применяются, нарушения блокируются

Permissive: Политики проверяются, но не применяются (только лог)

Disabled: SELinux отключён

Контроль доступа

Допустим, мы хотим, чтобы веб-сервер не мог читать ничего, кроме /var/www/html. (Вспомним любимого юзера **www-data**)

SELinux следит за этим через контексты:

> /var/www/html/index.html → httpd_sys_content_t

> В конфигурации **SELinux** указано, что httpd_t (контекст процесса веб-сервера) может читать только httpd_sys_content_t

Если положить файл в */home/user/file.txt* и указать путь в nginx.conf, сервер не получит к нему доступ, пока ты не задашь корректный **SELinux**-контекст

```
$ chcon -t httpd_sys_content_t /home/user/file.txt
```

Контроль доступа

Допустим, мы хотим, чтобы веб-сервер не мог читать ничего, кроме /var/www/html. (Вспомним любимого юзера **www-data**)

SELinux следит за этим через контексты:

> /var/www/html/index.html → httpd_sys_content_t

> В конфигурации **SELinux** указано, что httpd_t (контекст процесса веб-сервера) может читать только httpd_sys_content_t

Если положить файл в */home/user/file.txt* и указать путь в nginx.conf, сервер не получит к нему доступ, пока ты не задашь корректный **SELinux**-контекст

```
$ chcon -t httpd_sys_content_t /home/user/file.txt
```

Журналирование и аудит

Система аудита событий:

```
$ /var/log/syslog, /var/log/messages, /var/log/auth.log
```

Так же:

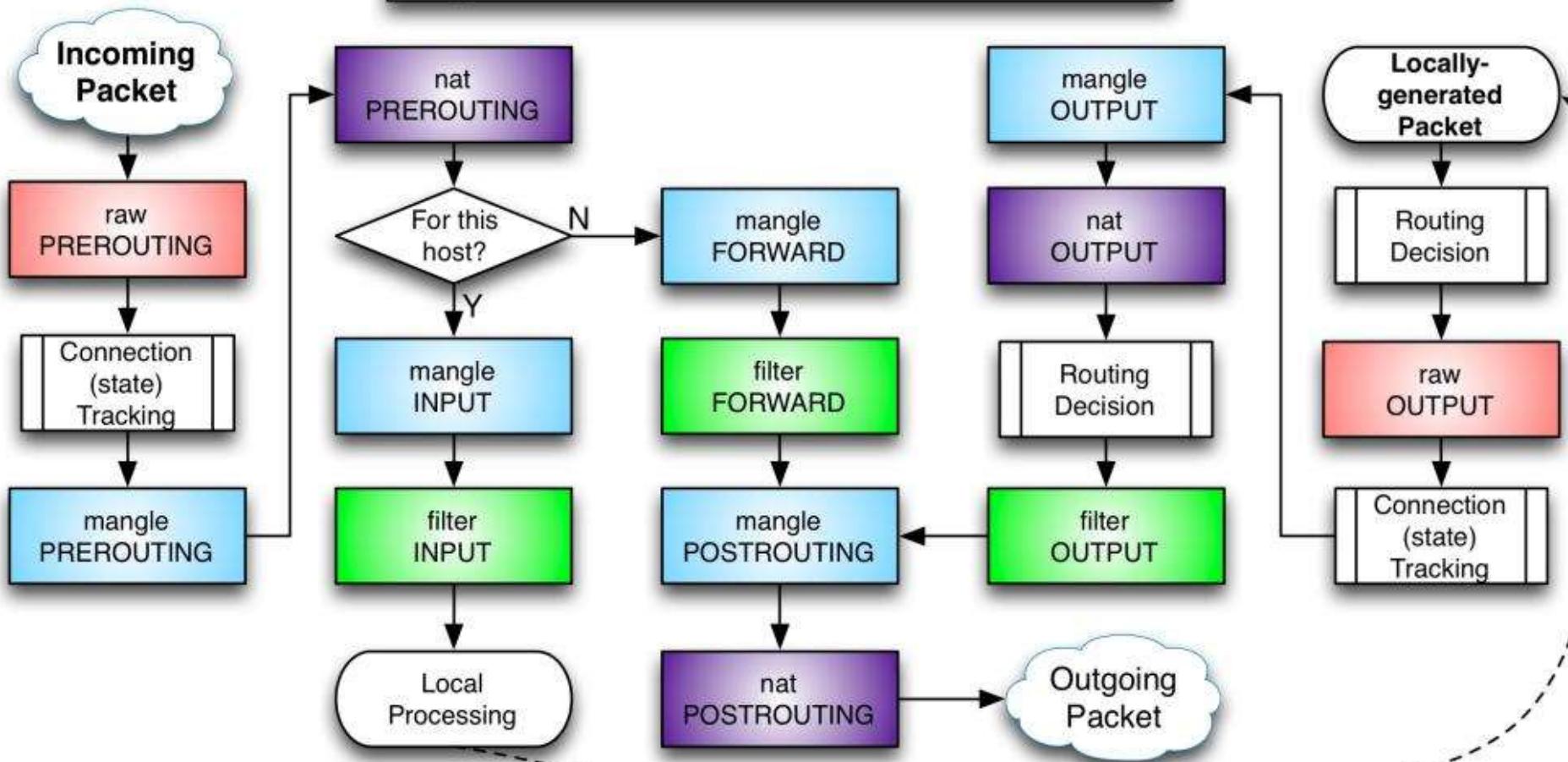
```
$ journalctl
```

Важно!

Здесь записываются ВСЕ события, не только безопасности

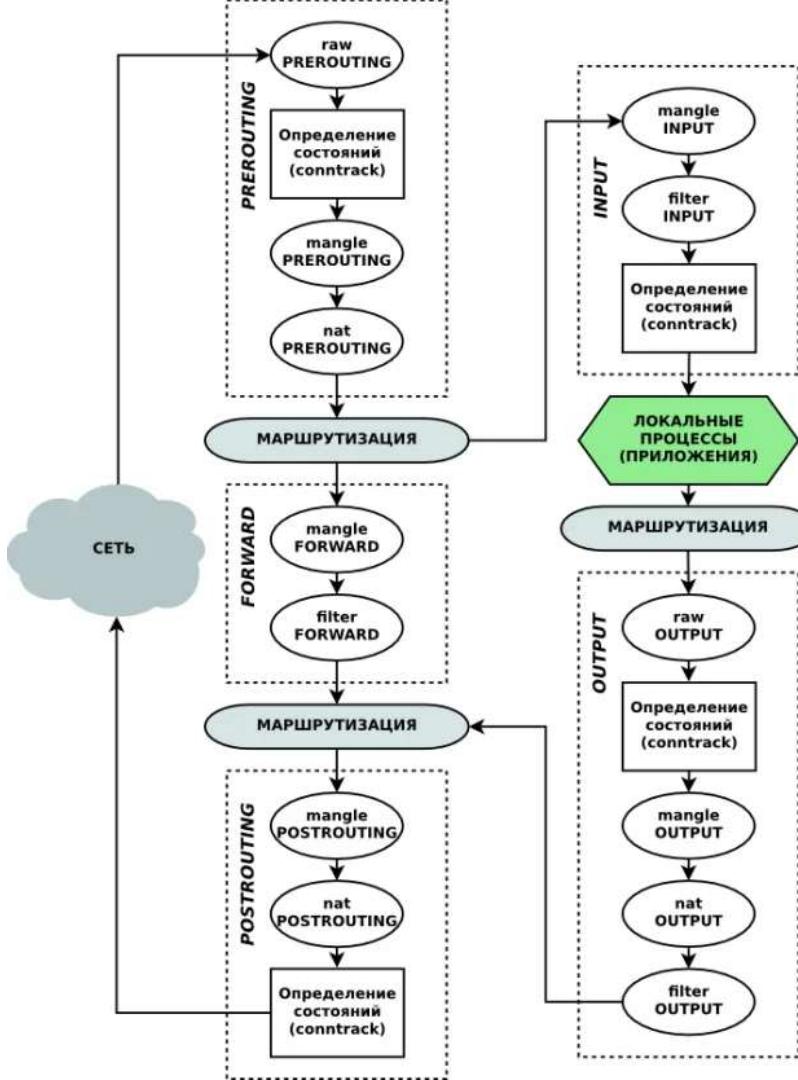
Сетевая защита

iptables Process Flow



Существует **5 типов стандартных цепочек**, встроенных в систему:

- > **PREROUTING** — для изначальной обработки входящих пакетов.
- > **INPUT** — для входящих пакетов адресованных непосредственно локальному процессу (клиенту или серверу)
- > **FORWARD** — для входящих пакетов перенаправленных на выход (заметьте, что перенаправляемые пакеты проходят сначала цепь PREROUTING, затем FORWARD и POSTROUTING).
- > **OUTPUT** — для пакетов генерируемых локальными процессами
- > **POSTROUTING** — для окончательной обработки исходящих пакетов.
Также можно создавать и уничтожать собственные цепочки при помощи утилиты iptables



Во ВСОШе есть таска на iptables!

СЗИ - Тайны Сети

- Проведен и расписан анализ кода вредоносного файла - 2 балла
- Создано правило iptables для блокировки аналогичных обращений - 1 балл

Примеры

> remote.ssh

```
$ iptables -A INPUT -p tcp -m conntrack --ctstate NEW -m tcp --dport 22  
-j ACCEPT
```

> web.http, web.https

```
$ iptables -A INPUT -p tcp -m conntrack --ctstate NEW -m multiport --  
dports 80,443 -j ACCEPT
```

> deny all traffic

```
$ iptables -P INPUT DROP  
$ iptables -P FORWARD DROP  
$ iptables -P OUTPUT ACCEPT
```

Примеры

Возвращаясь к слайду с таской из ВСОШ, давайте представим, что вредоносный траффик приходил с тачки 228.228.228.228 на порт 5432 БД Postgres. Как запретить вообще доступ к бд всем внешним адресам?

Примеры

Возвращаясь к слайду с таской из ВСОШ, давайте представим, что вредоносный траффик приходил с тачки **228.228.228.228** на порт **5432** БД Postgres. Как запретить вообще доступ к бд всем внешним адресам?

Можем кинуть в бан этот ip, но это не совсем эффективно:

```
$ iptables -A INPUT -p tcp -s 228.228.228.228 --dport 5432 -j DROP
```

Примеры

Возвращаясь к слайду с таской из ВСОШ, давайте представим, что вредоносный траффик приходил с тачки **228.228.228.228** на порт **5432** БД Postgres. Как запретить вообще доступ к бд всем внешним адресам?

Поэтому запретим доступ извне:

```
$ iptables -A INPUT -p tcp --dport 5432 -j DROP
```

Примеры

Возвращаясь к слайду с таской из ВСОШ, давайте представим, что вредоносный траффик приходил с тачки **228.228.228.228** на порт **5432** БД Postgres. Как запретить вообще доступ к бд всем внешним адресам?

Поэтому запретим доступ извне:

```
$ iptables -A INPUT -p tcp --dport 5432 -j DROP
```

Совсем не иметь доступ к бд – грустно. Давайте хотя бы на локалхосте дадим доступ:

Примеры

Возвращаясь к слайду с таской из ВСОШ, давайте представим, что вредоносный траффик приходил с тачки **228.228.228.228** на порт **5432** БД Postgres. Как запретить вообще доступ к бд всем внешним адресам?

Поэтому запретим доступ извне:

```
$ iptables -A INPUT -p tcp --dport 5432 -j DROP
```

Совсем не иметь доступ к бд – грустно. Давайте хотя бы на локалхосте дадим доступ:

```
$ iptables -I INPUT -p tcp -s 127.0.0.1 --dport 5432 -j ACCEPT
```

Практика

1. Узнайте IP адрес соседа. Пинганите его тачку, а потом настройте iptables на блокировку пинга (эхо запросы обозначаются как `--icmp-type 8`). Проверьте, что все работает
2. Поднимите nginx и убедитесь, что он поднялся. Как угодно измените `/var/www/html/index.html`. Подключитесь к сайтам соседей. А теперь настройте iptables так, чтобы доступ был только с 127.0.0.1
3. Запретите доступ по SSH к своей тачке. Перед тем как дропнуть соединение — запиши его в лог
Чек логи - `journalctl -k | grep "IPTables-Dropped"`
4. Откройте wireshark. Попробуйте поставить RST на попытку подключения к nginx, а затем DROP. Посмотрите, что изменилось в потоке пакетов и сделайте выводы безопасности

Шифрование

В Linux дистро применяются различные современные алгоритмы шифрования, что позволяет надежно хранить/передавать инфу

> GPG (GNU Privacy Guard)

Ассиметричное и симметричное шифрование файлов.

Подпись и проверка данных

> LUKS (Linux Unified Key Setup)

Шифрование целых разделов

Используется с cryptsetup

> dm-crypt

Механизм блочного шифрования в Linux

GPG

GPG — это свободная реализация OpenPGP (стандарта PGP). Используется для:

- > Шифрования файлов (Конфиденциальность)
- > Цифровой подписи (Аутентификация)
- > Проверки подписей (Авторизация)

```
$ gpg -c secret.txt # "c" = symmetric encr
```

```
$ gpg --encrypt --recipient alice@example.com report.pdf # asymmetric
```

```
$ gpg --sign file.txt # sign smth
```

LUKS

LUKS — стандарт шифрования дисков в Linux. Используется для шифрования **жёстких дисков, SSD, флешек, разделов**

Интеграции с паролем (или ключом), который **нужно вводить при загрузке** или монтировании

Связь с dm-crypt:

dm-crypt делает реальное шифрование и дешифрование "на лету". Таким образом, данные зашифрованы ВСЕГДА!!!

Стоит отметить, что данные, попадая в RAM, расшифровываются и висят в таком виде

Практика

1. Создайте файл `secret.txt` и запишите туда что угодно. Зашифруйте файл с помощью `gpg`. Дайте соседу, пусть попробует вскрыть ;P
2. Подпишите любой файл, а затем проверьте его подпись (тоже `gpg`)