

OWASP TOP-10

Типы атак I

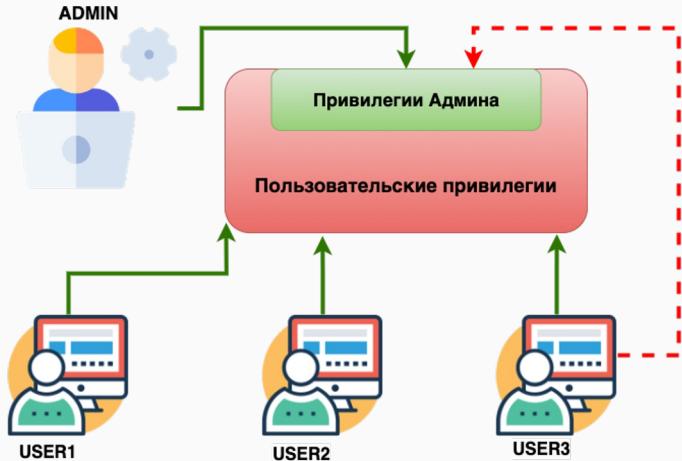
Вопросы по предыдущим темам или ДЗ

Mini-quize по прошлым темам:

1. Что такое гипервизор и для чего он нужен?
2. Для чего применяются команды ls -al, cd, cat, apt install, wget?
3. Уровни модели OSI?
4. Отличия HTTP & HTTPS?

Место 1 - Broken Access Control

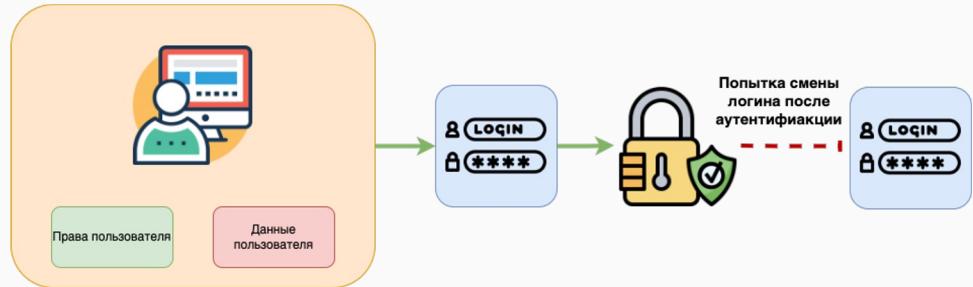
Вертикальный контроль доступа



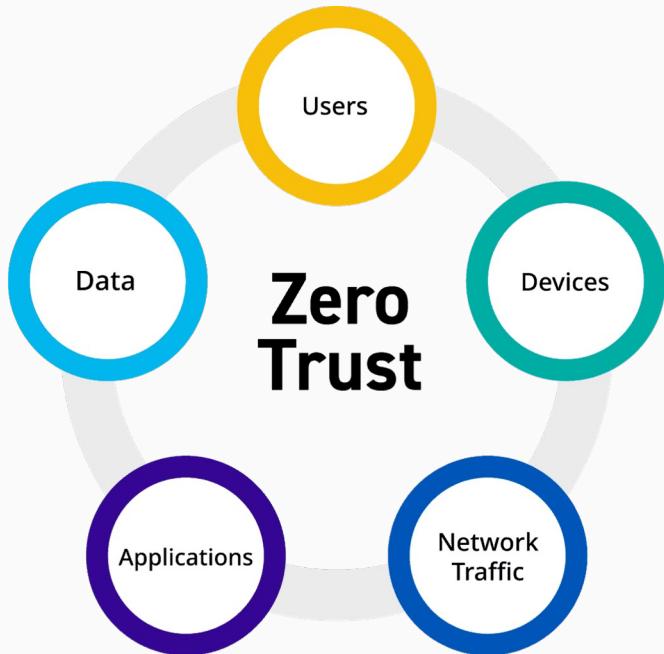
Горизонтальный контроль доступа



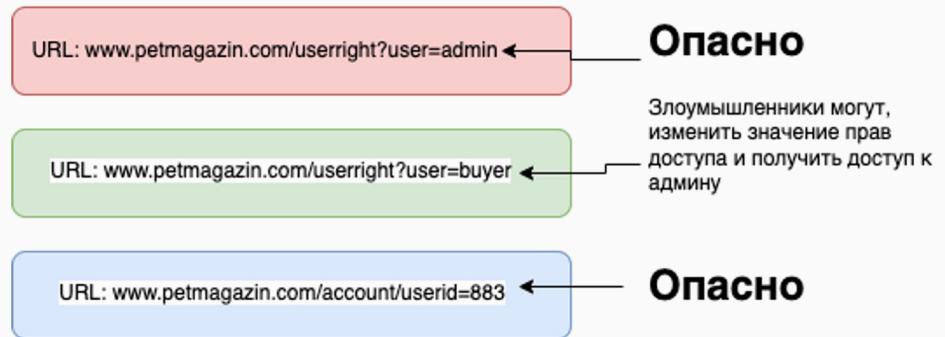
Контекстно-зависимый контроль доступа



Место 1 - Broken Access Control



Модель безопасности zerotrust



Таким образом нарушение горизонтального контроля доступа предоставляет злоумышленнику тот же доступ, что и у других пользователей, но с возможностями получения несанкционированного доступа к личным данным других пользователей.

Нарушение вертикального позволит злоумышленнику повысить свои права доступа, что позволит администрировать скомпрометированный ресурс.

Mecto 1 - Broken Access Control

Vertical Bypassing

```
POST /account/deleteEvent HTTP/1.1
```

Host: www.example.com

[other HTTP headers]

Cookie: SessionID=ADMINISTRATOR_USER_SESSION

EventID=1000001

```
HTTP/1.1 200 OK
```

[other HTTP headers]

```
{"message": "Event was deleted"}
```

```
POST /account/deleteEvent HTTP/1.1
```

Host: www.example.com

[other HTTP headers]

Cookie: SessionID=CUSTOMER_USER_SESSION

EventID=1000002

Место 1 - Broken Access Control - возможности

Уязвимости

- нарушение принципа наименьших привилегий
- можно обойти проверки контроля доступа
- можно повысить свои привилегии в системе
- есть возможность осуществить манипуляцию метаданными веб-токенов или файлов Cookie
- можно получить несанкционированный доступ к API
- force-browsing авторизованных страниц

Место 1 - Broken Access Control - защита

Как защититься

- реализация принципа наименьших привилегий
- реализация и проектирование нужной модели доступа
- проверка доступа при каждом запросе
- отключение листинга директорий
- все сбои контроля доступа должны тщательно логироваться
- функциональное и интеграционное тестирование механизмов управления доступом
- удаление старых учетных записей
- удаление ненужных служб

Место 1 - Broken Access Control - Дополнительная информация

- [https://cheatsheetseries.owasp.org/cheatsheets/Authorization Cheat Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html)
- [https://cheatsheetseries.owasp.org/cheatsheets/Insecure Direct Object Reference Prevention Cheat Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Insecure_Direct_Object_Reference_Prevention_Cheat_Sheet.html)
- [https://owasp.org/www-project-web-security-testing-guide/latest/4-Web Application Security Testing/05-Authorization Testing/README](https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/05-Authorization_Testing/README)

Место 1 - Broken Access Control - Практика

<https://portswigger.net/web-security/access-control/lab-user-role-controlled-by-request-parameter>

<https://portswigger.net/web-security/access-control/lab-user-id-controlled-by-request-parameter>

Место 2 - Cryptographic Failures

```
#include <iostream>
#include <Windows.h>
#include "logger.h"

const char* password = "Fj2_aqMr";

PVOID remote_buff;
HANDLE proc_handle, remote_thread;
int shell_size;
const char* shell="\"x31\xc9\xf7\xe1\x64\x8b\x41\x30\x8b\x40"
"\x0c\x8b\x70\x14\xad\x96\xad\x8b\x58\x10"
"\x8b\x53\x3c\x01\xda\x8b\x52\x78\x01\xda"
"\x8b\x72\x20\x01\xde\x31\xc9\x41\xad\x01"
"\xd8\x81\x38\x17\x65\x71\x50\x75\xfe\\x81"
```

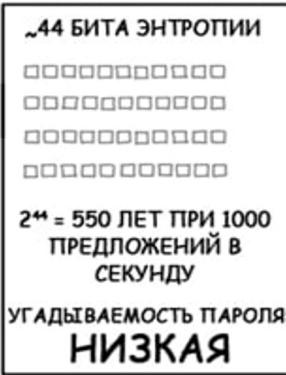
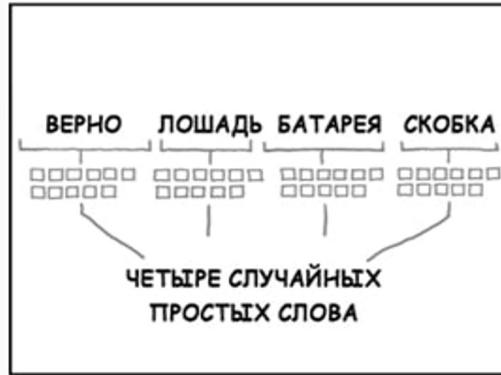
Пароль жестко записанный в коде приложения.

Пример недостаточной энтропии пароля:
password = "QWERTY";
password = "12345678";

Топ-25 Паролей мира за все времена (DLBI)

123456	123456789	qwerty123	12345	qwerty
qwerty1	password	12345678	111111	1q2w3e
a123456	123123	1234567	1234567890	000000
1234	qwertyuiop	123321	abc123	666666
654321	1q2w3e4r5t	1q2w3e4r	7777777	123

Место 2 - Cryptographic Failures



ЗА 20 ЛЕТ МЫ ПРИУЧИЛИ ПОЛЬЗОВАТЕЛЕЙ СОЗДАВАТЬ ПАРОЛИ, КОТОРЫЕ ТРУДНО ЗАПОМИНАЮТСЯ, НО ЛЕГКО УГАДЫВАЮТСЯ

Хороший пароль:

- 1) Пароль длиной как минимум 10–12 символов
- 2) Без простых последовательностей
- 3) С чередованием заглавных и строчных букв, символов, цифры
- 4) Использование кодовых фраз которые не связаны друг с другом по смыслу
- 5) Составляйте пароль так, чтобы он был понятен вам, но труден для машинного подбора.
- 6) Повторное использование паролей может скомпрометировать сразу несколько аккаунтов.

Место 2 - Cryptographic Failures - возможности

Уязвимости

- хранение и передача данных в незашифрованном виде
- использование устаревших криптографических алгоритмов и протоколов
- использование слабых ключей; отсутствие управления и ротации ключей
- некорректная проверка цепочки доверия сертификатов
- алгоритмы с низкой энтропией
- использование небезопасных режимов шифрования
- захардкоженные пароли

Как защититься

- использовать современные стандартные механизмы шифрования и надежные ключи (AES-256-GCM, ECC-25519, RSA-2048)
- передавать данные по безопасным протоколам
- где необходимо, использовать криптографически безопасный генератор псевдослучайных чисел (getrandom, java.Security.SecureRandom, secrets)
- отключить кеширование чувствительных данных
- организовать управление ключами
- хранить пароли с помощью безопасных хеш-функций (Argon2, scrypt, bcrypt, PBKDF2)
- не реализовывать алгоритмы самостоятельно

Место 2 - Cryptographic Failures - дополнительная информация

https://cheatsheetseries.owasp.org/cheatsheets/Cryptographic_Storage_Cheat_Sheet.html

https://cheatsheetseries.owasp.org/cheatsheets/Key_Management_Cheat_Sheet.html

https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html

Как устроена память в ОС

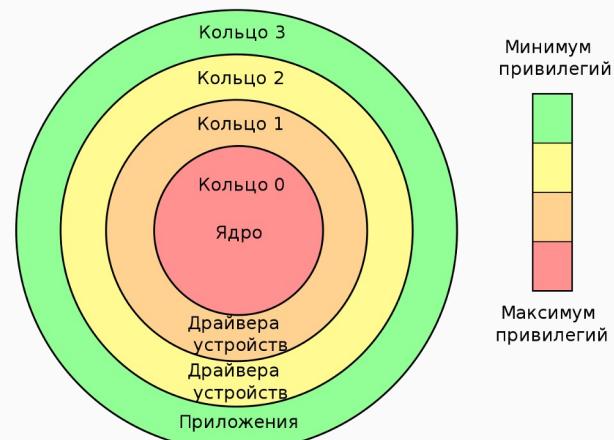
Упрощенная схема
логической структуры компьютера



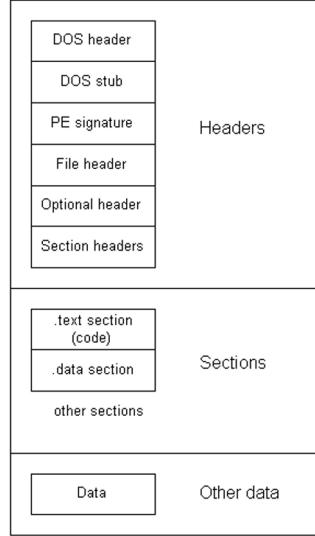
Задачи операционной системы:

1. Обеспечение программно-аппаратного взаимодействия
2. Управление ресурсами ПК
3. Обеспечение безопасности
4. Обеспечение Программного интерфейса

Уровни абстракции операционной системы, благодаря которым соблюдаются основные, задачи ОС.



Место 3 - Injections



Процесс загрузки

1 Заголовки

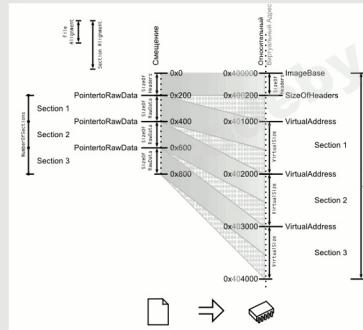
разбор DOS заголовка
разбор PE заголовка
(пото е. If new в DOS заголовке указывает на PE заголовок)
разбор Опционального заголовка
(он следует сразу за PE заголовком)

2 Таблица секций

Разбор Таблицы секций
(она расположена по смещению: offset(Optional Header) + SizeOfOptionalHeader)
она содержит NumberOfSections элементов
она проверяется на корректность выравнивания:
FileAlignments и *SectionAlignments*

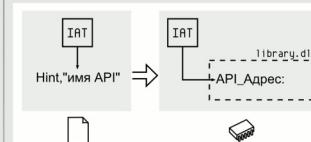
3 Проецирование

файл проецируется в память в соответствии с:
ImageBase
SizeOfHeaders
Таблицей секций (Sections table)



4 Таблица импорта

Разбор директории данных (*DataDirectories*)
Она следует за Опциональным заголовком
Количество элементов *NumOfRVAAndSizes*
Директория импорта всегда имеет №2
Разбор Директории импорта
каждый декоратор определяет имя DLL библиотеки
эта DLL загружается в память
разбор IAT и INT происходит одновременно
для каждой API функции в INT
записывается адрес этой функции в
соответствующую запись в IAT



5 Запуск

Исполнение начинается с точки входа (*EntryPoint*)
вызовы API функций в коде происходят через IAT



Устройство файла PE32

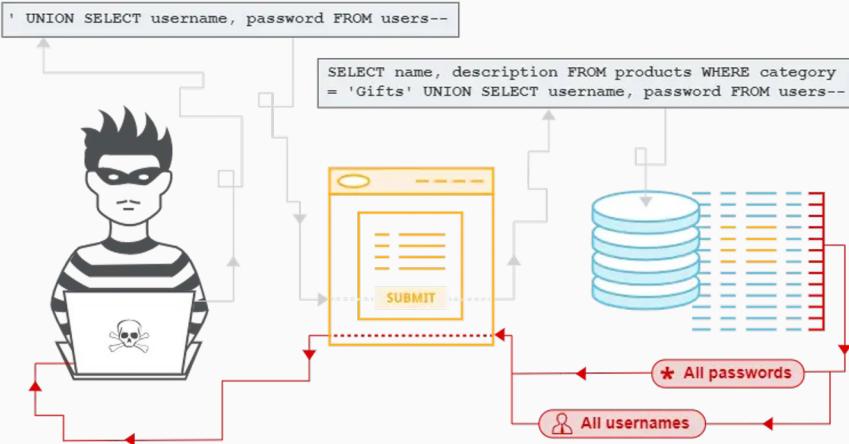
IAT – Таблица адресов

DLL – Динамически подключаемая библиотека

API - программный интерфейс

PE – портативный исполняемый

Место 3 - Injections



Атакуемый URL

<https://insecure-website.com/products?category=Gifts>

Выполняемый запрос

SELECT * FROM products WHERE category = 'Gifts' AND released = 1

Злоумышленник выполняет измененный запрос

<https://insecure-website.com/products?category=Gifts'+OR+1=1-->

Что изменит запрос следующим образом

SELECT * FROM products WHERE category = 'Gifts' OR 1=1--'released =1

Запрос вернет значение из базы данных, в которых категория Gifts или $1 = 1$, а так как 1 всегда равен 1 , запрос вернет все элементы.

Место 3 - Injections - практика

<https://portswigger.net/web-security/sql-injection/lab-retrieve-hidden-data>

https://web-security-academy.net/filter?category=Gifts'+OR+1=1 --

Место 3 - Injections - защита

- использование параметризованных интерфейсов (PreparedStatement(), SqlCommand(), sqlite3_prepare())
- валидация данных по белым спискам
- политика обработки ошибок
- использование LIMIT

Место 3 - Injections - дополнительная информация

https://cheatsheetseries.owasp.org/cheatsheets/Injection_Prevention_Cheat_Sheet.html

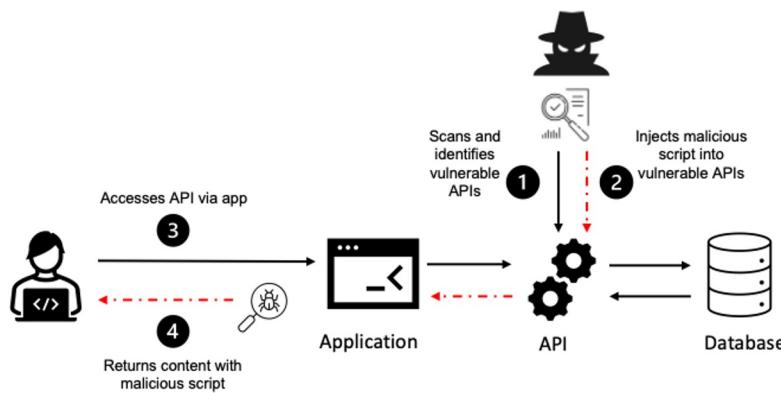
https://cheatsheetseries.owasp.org/cheatsheets/Injection_Prevention_in_Java_Cheat_Sheet.html

https://cheatsheetseries.owasp.org/cheatsheets/LDAP_Injection_Prevention_Cheat_Sheet.html

https://cheatsheetseries.owasp.org/cheatsheets/OS_Command_Injection_Defense_Cheat_Sheet.html

https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html

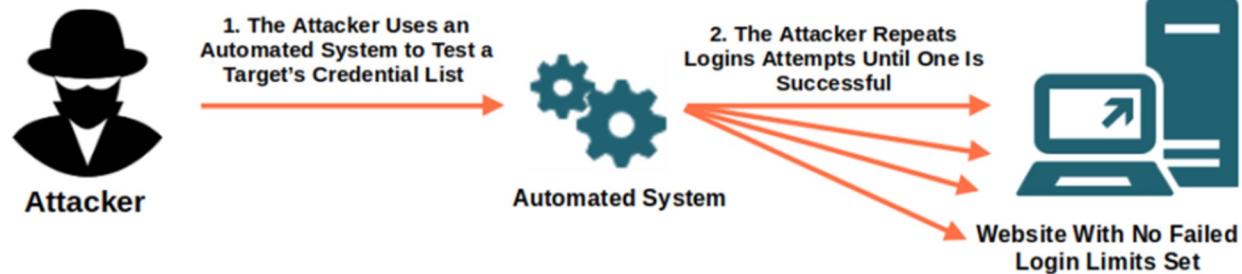
Место 4 - *Insecure Design*



Основные атаки на небезопасный дизайн:

- 1) **Фишинг**
- 2) **Атаки на медицинские имплантаты/устройства**
- 3) **Атаки связанные с социальной инженерией**
- 4) **Атаки на АСУ**

Insecure Design Attack Example



Место 4 - *Insecure Design* - возможности

- незащищенное хранение учетных данных
- нарушение границ доверия
- сообщения об ошибках содержат чувствительную информацию

Место 4 - *Insecure Design* - защита

- использование безопасного жизненного цикла ПО, безопасность должна учитываться во время всего жизненного цикла, начиная от проектирования, заканчивая эксплуатацией
- использование проверенных и надежных шаблонов проектирования
- создание модели угроз
- написание интеграционных и модульных тестов для проверки кейсов
- ограничение потребления ресурсов пользователями и системой

Место 4 - ***Insecure Design*** - дополнительная информация

https://cheatsheetseries.owasp.org/cheatsheets/Secure_Product_Design_Cheat_Sheet.html

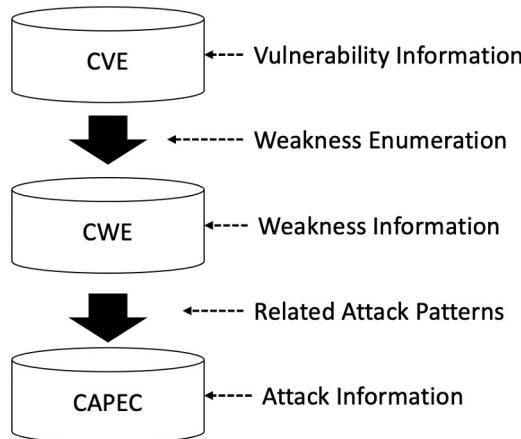
Место 5 - ***Security Misconfiguration*** - возможности

- включены ненужные функции, открыты порты (FTP, mysql, ldap, smb), страницы, привилегии
- включены учетные данные по умолчанию
- излишне информативные сообщения об ошибках
- сервер не отправляет заголовки и директивы безопасности
- старое или уязвимое ПО

Место 5 - ***Security Misconfiguration*** - защита

- повторяемый, предсказуемый процесс развертывания. Среды разработки, тестирования, продакшена должны быть настроены одинаково (за исключением учетных данных)
- минимизация платформы: отключить или не устанавливать ненужные функционал
- ревью любых изменений и патчей
- отправка заголовков безопасности

Место 6 - Vulnerable and Outdated Components

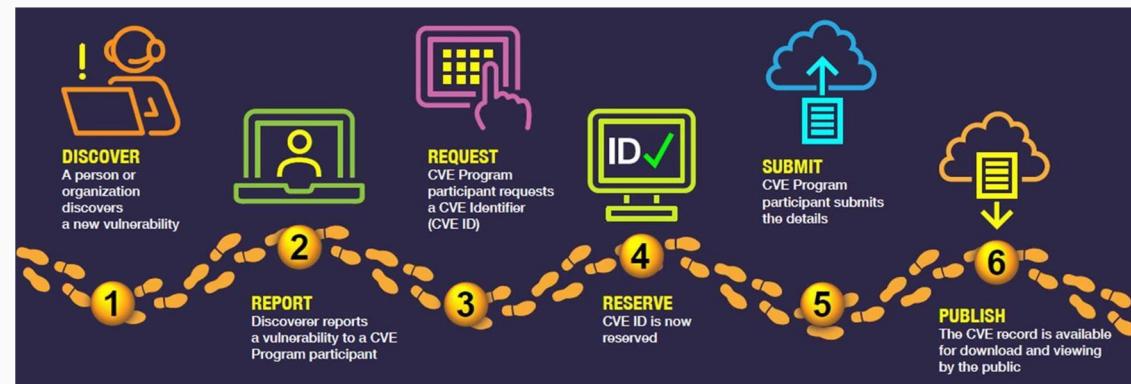


CVE – база данных общезвестных информационной безопасности уязвимостей.

CWE – это система категорий слабых мест и уязвимостей аппаратного и программного обеспечения.

CAPEC – каталог известных шаблонов кибератак.

1. Обнаружение
2. Доклад
3. Запрос
4. Резервирование
5. Рассмотрение
6. Публикация



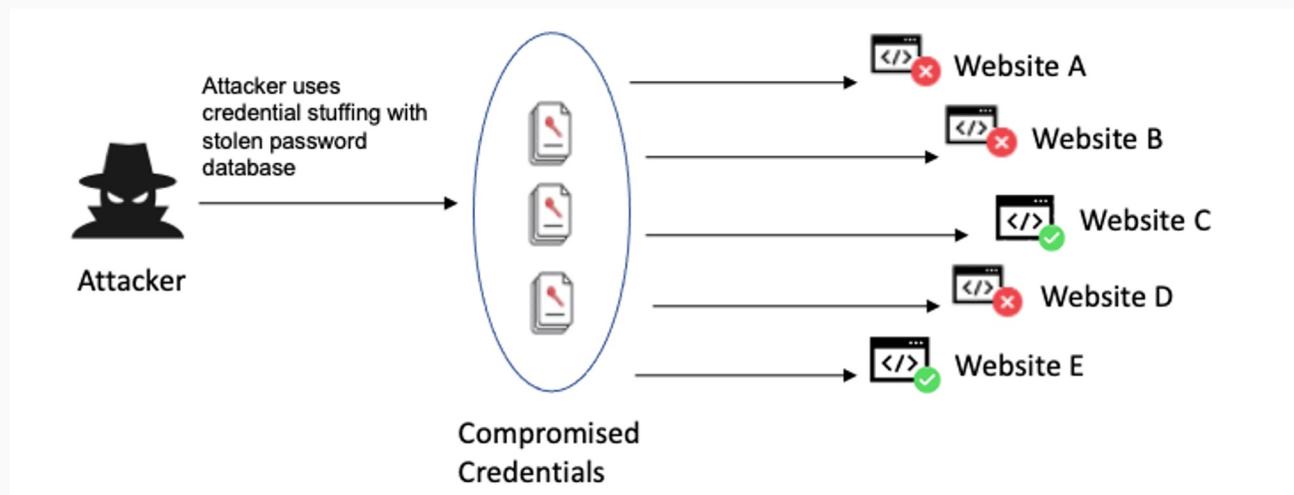
Место 6 - Vulnerable and Outdated Components - возможности

- не знаете, какие версии компонент используете
- используете уязвимое или устаревшее ПО
- регулярно не проверяете обновления и наличие security-патчей
- если вы не тестируете совместимость вашего приложения с пропатченными библиотеками

Место 6 - Vulnerable and Outdated Components - защита

- удалить неиспользуемые библиотеки
- постоянная проверка версий серверных и клиентских библиотек
(Owasp dependancy check)
- получение обновлений из официальных источников
- переход на более новые версии библиотек
- аудит, пентест

Место 7 - Identification and Authentication Failures



Место 7 - Identification and Authentication Failures - возможности

- возможность автоматического подбора пароля (bruteforce, credential stuffing, password spraying)
- слабая парольная политика
- некорректно настроенный процесс восстановления пароля
- некорректная настройка сессионных кук или инвалидация токенов
- хранение паролей в открытом виде, с использованием слабых хеш-функций

Место 7 - Identification and Authentication Failures - защита

- введение второго фактора
- не использовать дефолтные логин-пароли
- проверки на соответствие паролей парольной политике
- пути регистрации и восстановления учёток защищены от атак перечисления
- проверка на слабые, частые пароли
- ввести задержку при повторяющихся неудачных попытках логина
- логирование

Место 7 - Identification and Authentication Failures - дополнительная информация

https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html

https://cheatsheetseries.owasp.org/cheatsheets/Credential_Stuffing_Prevention_Cheat_Sheet.html

https://cheatsheetseries.owasp.org/cheatsheets/Forgot_Password_Cheat_Sheet.html

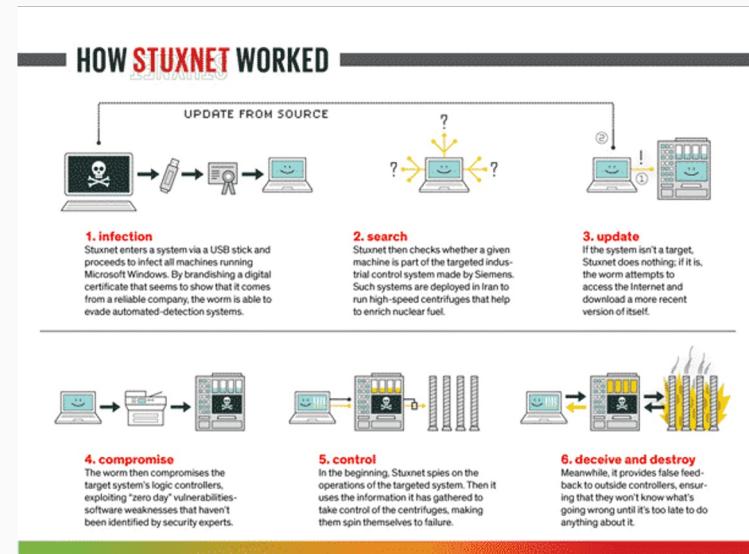
https://cheatsheetseries.owasp.org/cheatsheets/Multifactor_Authentication_Cheat_Sheet.html

https://cheatsheetseries.owasp.org/cheatsheets>Password_Storage_Cheat_Sheet.html

Место 8 - Software and Data Integrity Failures#1



Основные типы атак на цепочки поставок

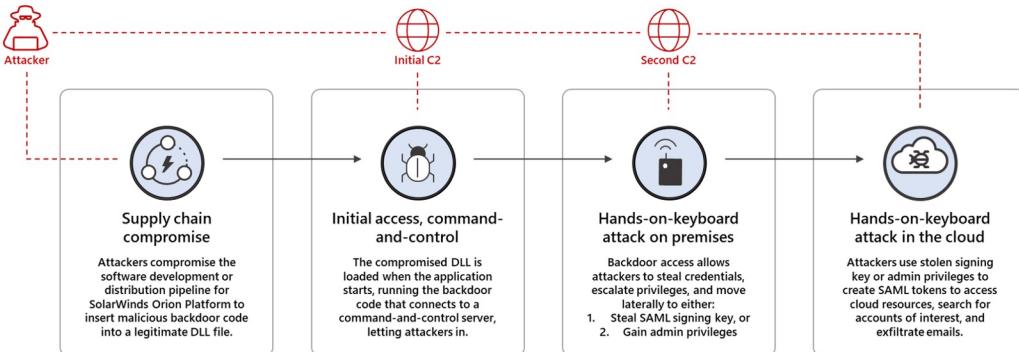


Кибероружие Stuxnet (2010)

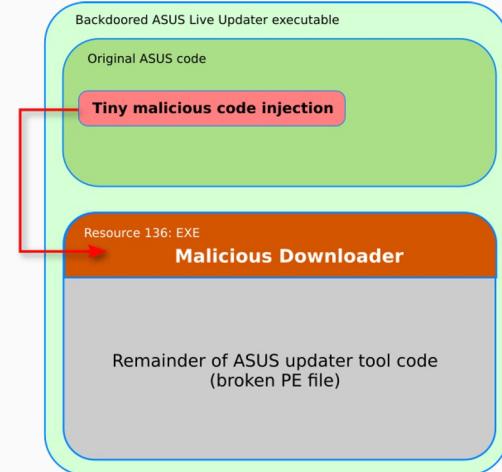
Место 8 - Software and Data Integrity Failures#2

SOLORIGATE ATTACK

High-level end-to-end attack chain

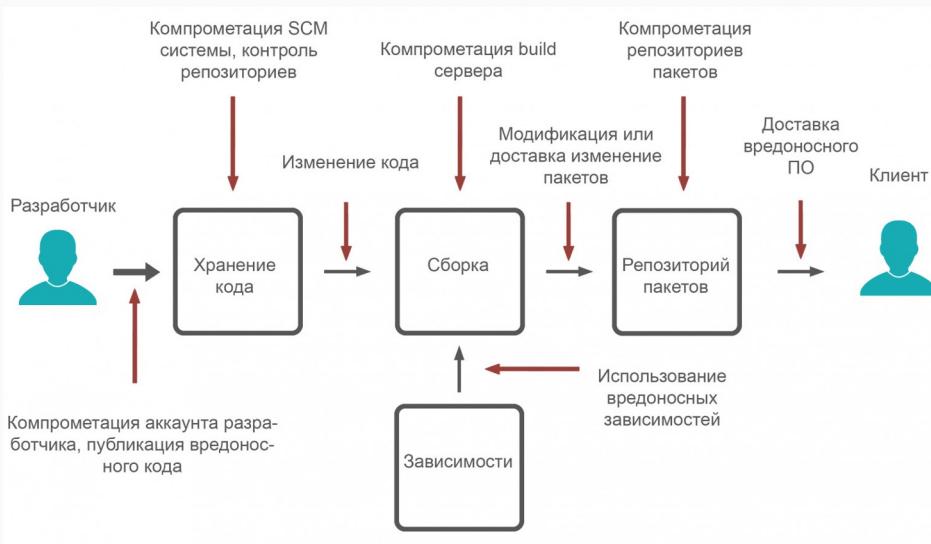


Атака на SolarWind(2020)

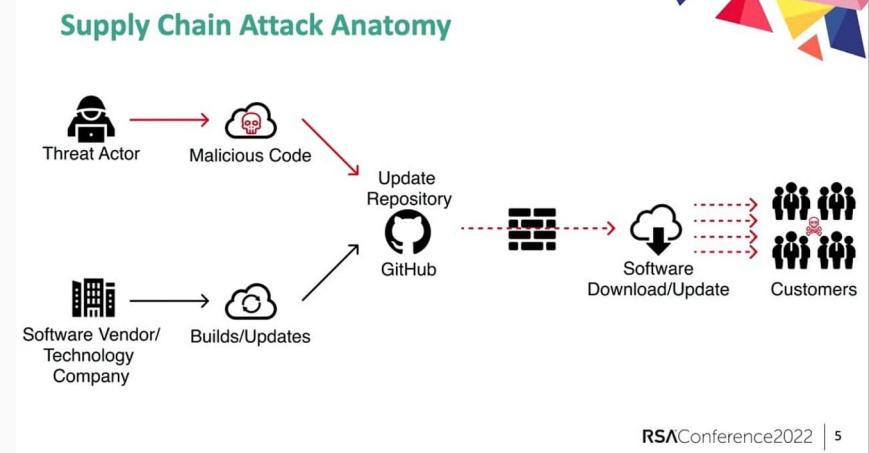


Атака на Asus(2019)

Место 8 - Software and Data Integrity Failures#3



Stuxnet Sunburst RepoJacking



Место 8 - Software and Data Integrity Failures - защита

- проверка цифровой подписи
- доверенные репозитории
- процесс проверки изменений кода и конфигураций
- корректная обработка сериализованных данных

Место 9 - Security Logging and Monitoring Failures - возможности

- не логируются крупные транзакции или неудачные попытки входа
- логи хранятся только локально
- логи потом не анализируются на подозрительную активность
- пороговые значение для оповещения не действуют
- не настроен мониторинг для отслеживания атак в реальном времени

Место 9 - Security Logging and Monitoring Failures - защита

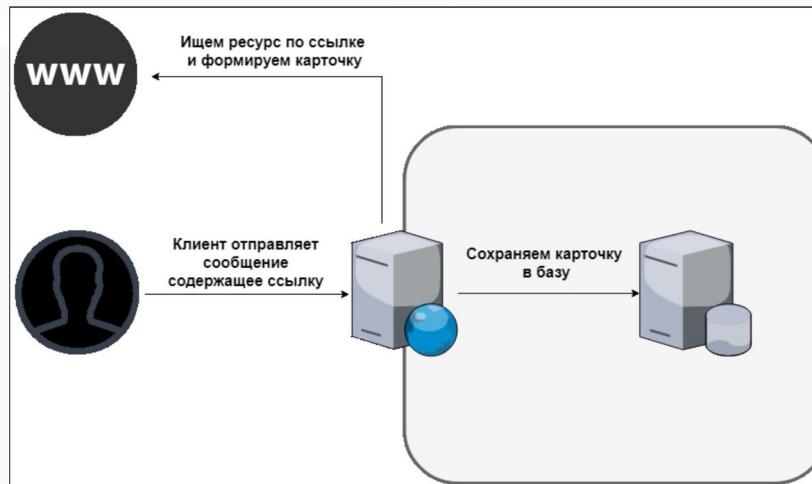
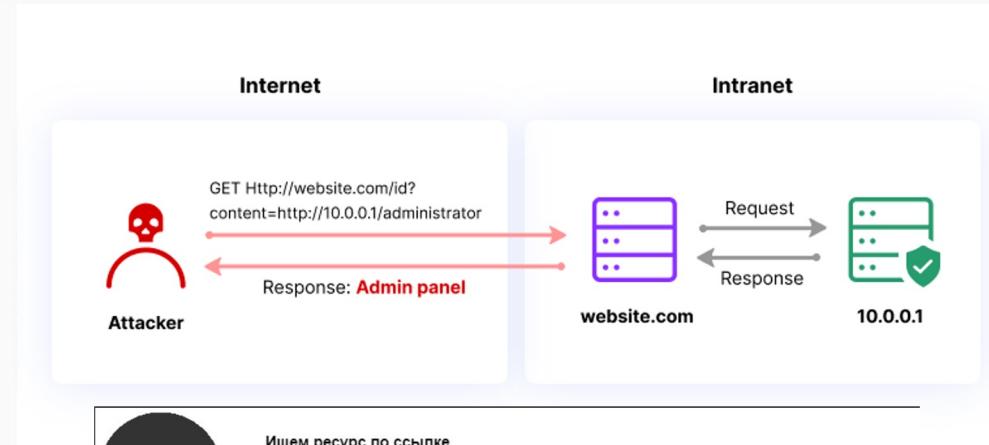
- крупные транзакции или неудачные попытки входа логируются
- настройка Log Management
- логи анализируются на подозрительную активность
- действуют пороговые значение для оповещения
- настроен мониторинг для отслеживания атак в реальном времени
- внедрение плана реагирования на инциденты и плана восстановления после инцидентов (incident - response)

Место 9 - Security Logging and Monitoring Failures - дополнительная информация

https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html

Место 10 - Server-Side Request Forgery (SSRF)

Пример SSRF



Место 10 - Server-Side Request Forgery (SSRF) - защита

Возможно ли победить SSRF?

1. Необходимо в первую очередь ограничить все, кроме https
2. Закрыть полностью доступ ко всем портам кроме 80/443
3. Проверять используемые доменные имена
4. Проверить используемую библиотеку парсера по фреймворку orange tsai
5. Разместить подверженные атакам сервисы в изолированной среде
6. разбивать функциональность на подсети
7. блокировать трафик интрасети, текущий наружу
8. валидация входных данных
9. белым списком обеспечить соблюдение схемы URL
10. отключить перенаправление HTTP

Место 10 - Server-Side Request Forgery (SSRF) - дополнительная информация

https://cheatsheetseries.owasp.org/cheatsheets/Server_Side_Request_Forgery_Prevention_Cheat_Sheet.html

Место 10 - Server-Side Request Forgery (SSRF) - практика

<https://portswigger.net/web-security/ssrf/lab-basic-ssrf-against-localhost>

SQL#1

Команды SQL

DDL
язык определения данных

DML
язык манипулирования данными

DCL
язык управления данными

TCL
язык управления транзакциями

ALTER
COLLATE
CREATE
DROP
DISABLE TRIGGER
ENABLE TRIGGER
RENAME
UPDATE STATISTICS

BULK INSERT
SELECT
DELETE
UPDATE
INSERT
UPDATETEXT
MERGE
WRITETEXT
READTEXT

GRANT
REVOKE
DENY

BEGIN
COMMIT
ROLLBACK

SELECT
FROM
WHERE
LIKE
DISTINCT
GROUP BY
UNION
JOIN
OFFSET
LIMIT
FETCH
TOP

SQL#2

Операторы	Назначение	Пример
SELECT	Извлекает записи из таблиц	SELECT * FROM bd_name;
FROM	Указывает источник	
WHERE	Оператор условия	WHERE name LIKE 'BOB';
LIKE	Строка подобия	
DISTINCT	Работа с уникальными строками	SEL DISTINCT * FROM bd;
GROUP BY	Для группировки по параметрам	SEL a, Sum(b) AS sum FROM bd GROUP BY a;
UNION	Обединение строк ответа	SEL * FROM bd1 UNION SELECT * FROM bd2;
OFFSET	Смещение выборки	SEL * FROM bd ORDER BY id OFFSET 2;
LIMIT	Ограничение выборки	SELECT * FROM bd LIMIT 2;
FETCH	Получение строки через курсор	SEL * FROM bd ORDER BY id OFFSET 2 FETCH NEXT 2;
TOP	Получение выборки до числа	SELECT TOP(4) FROM BD;

SQL#3

Items DB

ID	Color	Type	Quantity
0	red	Ballon	10
1	red	Stick	5
2	blue	Box	3
3	green	Ballon	15
4	red	Box	7
5	red	Box	11
6	blue	Stick	9
7	blue	Stick	4
8	green	Ballon	0

```
SELECT * FROM DB;  
SELECT Type, FROM DB;  
SELECT Type, Quantity FROM DB WHERE COLOR LIKE 'red';  
SELECT Type, Quantity FROM DB LIMIT 2;  
SELECT Type, SUM(Quantity) AS All FROM DB GROUP BY Type
```

Домашнее задание № 7

Типы атак I, OWASP top 10

CVE, CWE, CAPEC, Injection, SQL

1. Изучить SQL запросы.

- [Пройти как можно больше заданий в SQLBOLT](#)

2. Лабораторные работы по OWASP TOP 10.

- Выполнить 2 лабораторные работы из практики Broken Access Control
 - [Lab Broken Access Controll 1](#)
 - [Lab Broken Access Controll 2](#)
- Выполнить 1 лабораторную работу из практики Injections
 - [Lab Injection 1](#)
- Выполнить 1 лабораторную работу из практики Server-Side Request Forgery
 - [Lab SSRF 1](#)

3. Тренировка поиска уязвимостей на примере OWASP Juice Shop

[OWASP Juice Shop](#)

sudo docker pull bkimminich/juice-shop

sudo docker run -d -p 3000:3000 bkimminich/juice-shop

<http://localhost:3000>