

**Основные вектора атак на приложения. Способы их достижения и возможные последствия**






# Вопросы

- 1. Вектора угроз, для информационных объектов.
- 2. Обзор основных этапов атаки на приложение.
- 3. Общее представление, об уязвимостях приложений.



# Вопрос №1

- Вектора угроз, для информационных объектов



# Вектора угроз, для информационных объектов


- Для того, что бы понять все вектора угроз, необходимо ответить на следующие вопросы:
  - - Что это?
  - - Какие цели?
  - - Кто это?
  - - Как возможно достигнуть цели?

# Вектора угроз, для информационных объектов

- **Под угрозой** понимаю вероятность успешной атаки.
- **Под атакой** на приложение, понимаю такое воздействие на приложение, в результате которого будет получен несанкционированный доступ к информации, получение новых возможностей, различными способами, а так же вывод приложения в нерабочее состояние. Аналогично рассматривается атака на информационную систему или инфраструктуру.


# Вектора угроз, для информационных объектов

- К основным целям атаки, на информационные объекты считаются:
  - - социально-политическая, цель - получение закрытой информации, в социальном или политическом сегменте, а так же политическая дискредитация.
  - - военный шпионаж.
  - - деструктивная деятельность, направленная на полный захват, или «уничтожение» инфраструктуры других заинтересованных лиц.



# Вектора угроз, для информационных объектов

- - получение экономической выгоды, через промышленный шпионаж, или деструктивную деятельность, в инфраструктуре конкурента.
- - личные мотивы.



# Вектора угроз, для информационных объектов

- Категории атакующих:
  - - Сканер
  - - Организованная кибергруппа
  - - Коммерческая кибергруппа
  - - АРТ(Advanced persistent threat)





## Вопрос №2

- Рассмотрение основных этапов атаки на приложение

# Рассмотрение основных этапов атаки на приложение

- Атака на информационные объекты делится на следующие этапы:
  - 0. Разведка;
  - 1. Проведение атаки;
  - 2. Повышение привилегии;
  - 3. Боковое перемещение;
  - 4. Закрепление в системе;
  - 5. Поиск критически важных данных;
  - 6. Кража данных;
  - 7. Влияние на инфраструктуру

# Рассмотрение основных этапов атаки на приложение

- Для приложения характерны только два этапа атаки:
- - **Сканирование и изучение приложения;**
- - **Проведение атаки;**
- Не зависимо, является это приложение часть информационной системы, инфраструктуры или это отдельное приложение.
- В зависимости от типа, класса и используемых технологии, отличаются **техники и тактики** проведения атаки.
- **Под техникой** понимается набор и последовательность действий, для проведения этапа атаки.
- **Под тактикой** понимается совокупность **техник**, возможных на данном этапе.

# Рассмотрение основных этапов атаки на приложение

```
Nmap scan report for stacked.htb (10.10.11.112)
Host is up (0.17s latency).

PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_   3072 12:8f:2b:60:bc:21:bd:db:cb:13:02:03:ef:59:36:a5 (RSA)
|_   256  af:f3:1a:6a:e7:13:a9:c0:25:32:d0:2c:be:59:33:e4 (ECDSA)
|_   256  39:50:d5:79:cd:0e:f0:24:d3:2c:f4:23:ce:d2:a6:f2 (ED25519)
80/tcp    open  http         Apache httpd 2.4.41
|_ http-title: STACKED.HTB
|_ http-server-header: Apache/2.4.41 (Ubuntu)
2376/tcp  open  ssl/docker?
|_ ssl-cert: Subject: commonName=0.0.0.0
| Subject Alternative Name: DNS:localhost, DNS:stacked, IP Address:0.0.0.0, IP Address:127.0.0.1, IP Address:172.17.0.1
| Not valid before: 2021-07-17T15:37:02
|_ Not valid after: 2022-07-17T15:37:02
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

```
. [Status: 200, Size: 5055, Words: 367, Lines: 159]
# [Status: 200, Size: 5055, Words: 367, Lines: 159]
index.html [Status: 200, Size: 5055, Words: 367, Lines: 159]
```

```
images [Status: 301, Size: 311, Words: 20, Lines: 10]
css [Status: 301, Size: 308, Words: 20, Lines: 10]
js [Status: 301, Size: 307, Words: 20, Lines: 10]
fonts [Status: 301, Size: 310, Words: 20, Lines: 10]
sass [Status: 301, Size: 309, Words: 20, Lines: 10]
```

- Пример сканирования

# Рассмотрение основных этапов атаки на приложение

- Для проведения атаки необходимо изучить приложение, выяснить, что это за приложение, на каких технологиях, модулях, библиотеках оно построено. Какая версия самого приложения или компонентов. Необходимо это для поиска готовых техник атаки на приложения. Если нет готовых техник, для перехода на следующий этап, приложение активно изучается и тестируется.
- Существует три типа тестирования приложения:
  - - Тестирование белого ящика;
  - - Тестирование серого ящика;
  - - Тестирование чёрного ящика;

# Рассмотрение основных этапов атаки на приложение


Request	Payload	Status	Error	Timeout	Length	Comment
284	../../../../index.jsp	200	<input type="checkbox"/>	<input type="checkbox"/>	11612	
285	../../../../index.jsp	200	<input type="checkbox"/>	<input type="checkbox"/>	11612	
286	../../../../index.jsp	200	<input type="checkbox"/>	<input type="checkbox"/>	11612	
287	../../../../index.jsp	200	<input type="checkbox"/>	<input type="checkbox"/>	11612	
289	../../../../index.jsp	200	<input type="checkbox"/>	<input type="checkbox"/>	11612	

Request    Response


Pretty   Raw   Hex   Render

Home   Documentation   Configuration   Examples   Wiki   Mailing Lists   Find Help

## Apache Tomcat/8.5.23

 APACHE SOFTWARE FOUNDATION  
http://www.apache.org/

If you're seeing this, you've successfully installed Tomcat. Congratulations!

 **Recommended Reading:**  
[Security Considerations HOW-TO](#)  
[Manager Application HOW-TO](#)  
[Clustering/Session Replication HOW-TO](#)

[Server Status](#)  
[Manager App](#)  
[Host Manager](#)

### Developer Quick Start

[Tomcat Setup](#)   [Realms & AAA](#)   [Examples](#)   [Servlet Specifications](#)  
[First Web Application](#)   [JDBC DataSources](#)   [Tomcat Versions](#)

- Пример, тестирования чёрного ящика

# Рассмотрение основных этапов атаки на приложение

- Техники проведение атак:
- - Эксплуатация уязвимости, подтехники:
  - - Эксплуатация бинарной уязвимости в компилируемых низкоуровневных ЯП;
  - - Эксплуатация ошибок реализации в интерпретируемых ЯП;
  - - SQL инъекции
- - Эксплуатация некорректной настройки приложения.
- - Патч приложения
- Чаще всего применяется комбинация из первой и второй техник. Такая техника, как патч, может применяться в других тактиках(этапах атаки).



## Вопрос №3

- Общее представление, об уязвимостях приложений



# Общее представление, об уязвимостях приложений

- **Уязвимость** — это ошибка в реализации приложения, в результате которой, возможно получить непредусмотренный функционал.

# Общее представление, об уязвимостях приложений

- По получаемому результату, уязвимости делятся на несколько категории:
- - Утечка данных(Data leak), как правило имею слабый ущерб, но чаще всего применяются в связке, с другими классами уязвимостей.
- - Отказ в обслуживании(Denial of Service,DoS), имеет более серьёзный ущерб, вызывает крах приложения.
- - Удалённое выполнение кода(Remote Code Execution, RCE)
- - Повышение привилегий(Privilege escalation)

# Общее представление, об уязвимостях приложений

- `router.post('/', (req, res) => { if (req.app.get('env') === 'development') {`
- `console.log(req.body);}`
- `var { name, age, city, affiliation } = req.body;`
- `var handler = require(`../lib/offices/${city}`);`
- `var { office, error } = handler.save(name, age);`
- `msg = error && {type: 'error', code: 'HE-DOESNT-LIKE-YOU', description: error}`  
 `||`
  - `{type: 'success', office: office};`
- `res.send(msg);`
- `});`

Пример уязвимого кода

# Общее представление, об уязвимостях приложений

- Наиболее частой уязвимостью, всех веб-приложений, является SQL-инъекция, так как редкое веб-приложение обходится без базы данных. Возникает уязвимость в результате некорректной обработки параметров, вводимых пользователем и формирование из них запроса в БД.

# Общее представление, об уязвимостях приложений

- `$user = $_GET['user'];`
- `$query = "SELECT * FROM news WHERE user='$user'";`

Пример SQL-инъекции

# Общее представление, об уязвимостях приложений

- Определить факт наличия SQL-инъекции можно через анализ логов веб-приложений, поиск необходимо производить по ключевым словам, характерных для языка SQL, таких как:
  - - SELECT
  - - FROM
  - - OR
  - - UNION
  - - AND
  - - PG\_SLEEP
  - - и др.

# Общее представление, об уязвимостях приложений

- GET  
`/faces/wcnav_defaultSelection?query=query+AND+%28SELECT+*+FROM+%28SELECT%28SL  
EEP%285%29%29%29ZcZQ%29`
- GET  
`/faces/wcnav_defaultSelection?query=query+AND+%28SELECT+*+FROM+%28SELECT%28SL  
EEP%285%29%29%29uaMI%29--+WfeI`
- GET  
`/faces/wcnav_defaultSelection?query=query+AND+6764%3D%28SELECT+6764+FROM+PG_SL  
EEP%285%29%29`
- GET  
`/faces/wcnav_defaultSelection?query=query+AND+8157%3D%28SELECT+8157+FROM+PG_SL  
EEP%285%29%29--+FQrH`
- GET  
`/?query=%28SELECT++UTL_INADDR.get_host_name%28%2710.0.0.1%27%29+from+dual+uni  
on+SELECT++UTL_INADDR.get_host_name%28%2710.0.0.2%27%29+from+dual+union+SELEC  
T++UTL_INADDR.get_host_name%28%2710.0.0.3%27%29+from+dual+union+SELECT++UTL_I  
NADDR.get_host_name%28%2710.0.0.4%27%29+from+dual+union+SELECT++UTL_INADDR.g  
et_host_name%28%2710.0.0.5%27%29+from+dual%29`

Пример, определение SQL-инъекции

# Общее представление, об уязвимостях приложений

GET ?query=%3C%21--%23EXEC+cmd%3D%22dir+%5C%22--%3E

GET /faces/wcnav\_defaultSelection?query=%22%3E%3C%21--%23EXEC+cmd%3D%22dir+%5C%22--%3E%3C

GET /faces?query=%22%3E%3C%21--%23EXEC+cmd%3D%22dir+%5C%22--%3E%3C

GET /?query=%3C%21--%23EXEC+cmd%3D%22dir+%5C%22--%3E

GET ?query=%22%3E%3C%21--%23EXEC+cmd%3D%22dir+%5C%22--%3E%3C

GET /?query=%22%3E%3C%21--%23EXEC+cmd%3D%22dir+%5C%22--%3E%3C

## Декодированные запросы

GET ?query=<!--#EXEC+cmd="dir+\ "-->

GET /faces/wcnav\_defaultSelection?query="><!--#EXEC+cmd="dir+\ "--><

GET /faces?query="><!--#EXEC+cmd="dir+\ "--><

GET /?query=<!--#EXEC+cmd="dir+\ "-->

GET ?query="><!--#EXEC+cmd="dir+\ "--><

GET /?query="><!--#EXEC+cmd="dir+\ "--><



- 
- Спасибо за внимание!