HIGH-TECH BRIDGE®

INFORMATION SECURITY SOLUTIONS

Web Application Security and ImmuniWeb® Self-Fuzzer Firefox Extension

September 10, 2013

Xavier ROUSSEL
Security Auditor at High-Tech Bridge

HIGH-TECH BRIDGE®
INFORMATION SECURITY SOLUTIONS

- Web applications are the most widespread applications on the internet. Almost all companies in the world have a web site.
- Web applications quite often host sensitive data like customer database containing personal information, passwords, credit cards numbers, etc.
- Even web applications that do not host sensitive data are targeted by hackers. Once compromised, they are used as proxies by hackers to attack bigger resources, to spread malware among website visitors turning their PCs into zombies, or simply to perform phishing campaigns.
- Web applications vulnerabilities are usually much easier to exploit than “low level” vulnerabilities, such as buffer overflows in binaries for example.

- The amount of sensitive information that web applications contain today and simplicity of web attack vectors drastically expose web applications to both massive and targeted attacks.

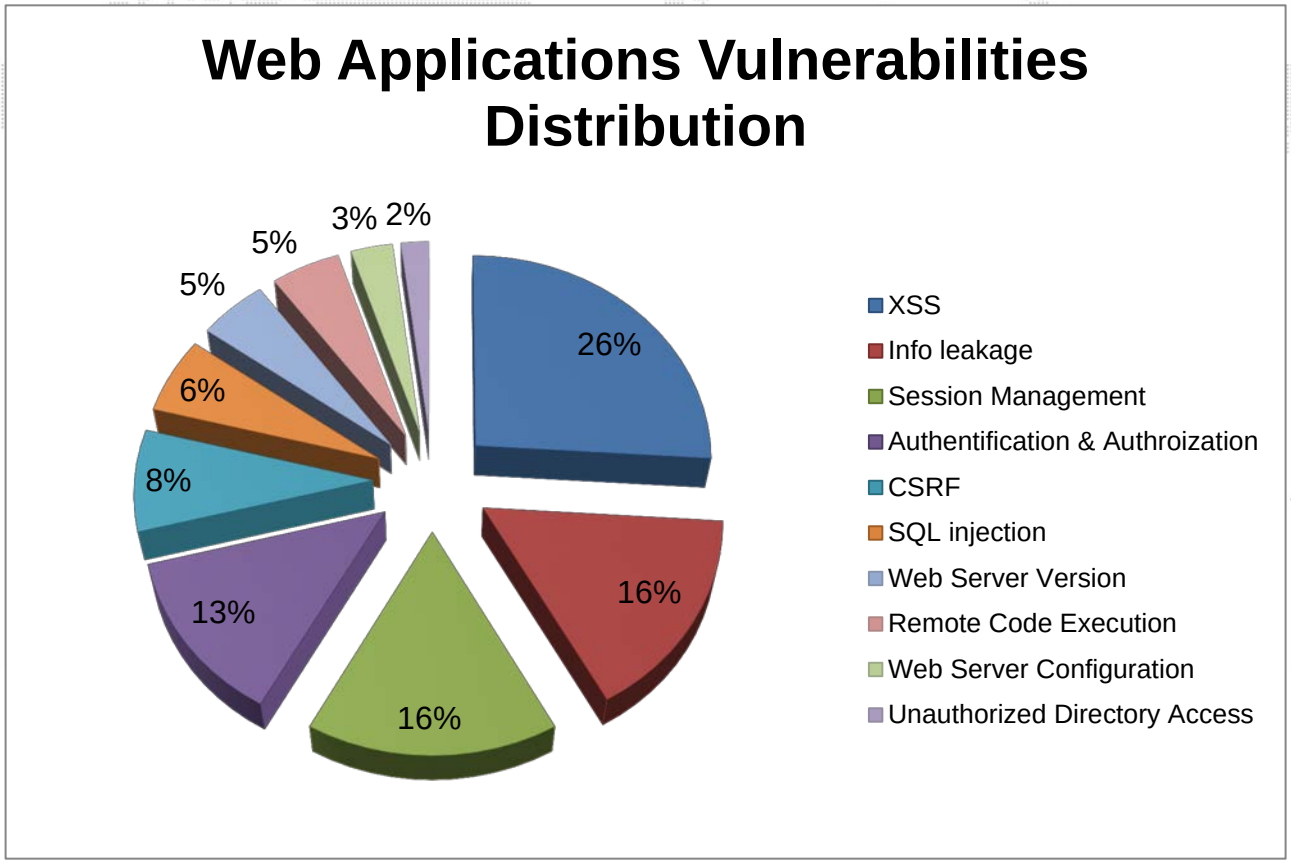
- Depending on the complexity and the size of a website, security assessment services, such as penetration test or security audit, can be quite expensive and long.

- As a consequence, many SMBs simply do not perform any kind of web security assessments at all, leaving their website an easy entry point to their local networks and other corporate IT assets for hackers.

- More information about web applications security can be found in Frost & Sullivan White Paper written in collaboration with MITRE, OTA and High-Tech Bridge experts: "The Growing Hacking Threat to Websites: An Ongoing Commitment to Web Application Security" and in its video version.

HIGH-TECH BRIDGE
INFORMATION SECURITY SOLUTIONS

According to Net-Security, the following graph represents distribution of the most common web vulnerabilities :



- Cross-Site Scripting [CWE-79], also known as XSS, is the most widespread vulnerability affecting web applications.

- Basically, if a user-controlled variable is outputted in the source code of HTML page without being properly filtered, it is possible to execute arbitrary JavaScript code on the client side, and steal sensitive information, such as cookies, authentication sessions or even browsing history.

- The vulnerability is present on the server side (in web application) but affects the client (website visitor) as malicious code is executed in client's browser.

Example :

Normal request :

Your name is Peter

Input

Not filtered, vulnerable:

Your name is <script>alert('xss')</script>

Filtered, not vulnerable:

Your name is <script>alert('xss')</script>

HIGH-TECH BRIDGE
INFORMATION SECURITY SOLUTIONS

- SQL Injection [CWE-89] is another well-known web application vulnerability, permitting attacker to alter SQL queries sent by vulnerable web application to a database and get almost any information, such as user's logins and passwords, from the database.

- Basically, SQL injection occurs when user-controlled variable is not filtered correctly (e.g. to escape characters such as double quote or single quote) and passed directly to SQL query.

Example :

Normal request :

*SELECT * FROM users WHERE username="peter"*

Input

Result : "peterpasswd"

Not filtered, injection :

*SELECT * FROM users WHERE username="peter" or username="admin"*

Result : "adminpasswd"

Filtered :

*SELECT * FROM users WHERE username="peter\" or username=\"admin"*

Result : null

HIGH-TECH BRIDGE®
INFORMATION SECURITY SOLUTIONS

I. Web Application Security in Brief

II. ImmuniWeb® SaaS in Brief

III. ImmuniWeb® Self-Fuzzer Firefox Extension

- ImmuniWeb® is a next-generation web application security assessment solution with Software-as-a-Service (SaaS) delivery model, entirely developed and supported by High-Tech Bridge.
- ImmuniWeb® is a **hybrid** of advanced **automated web vulnerability scanning** and accurate **manual web application penetration testing** performed in parallel.
- ImmuniWeb® SaaS consists of three main parts: ImmuniWeb® Portal, ImmuniWeb® Security Scanner and ImmuniWeb® Auditors.
- ImmuniWeb® enables people to assess their websites security in a simple, fast, accurate and cost-affordable manner.

- ImmuniWeb® Security Assessment detects the following types of web application vulnerabilities:

Path Traversal CWE-22	OS Command Injection CWE-78	Stored and Reflected XSS CWE-79	DOM-Based XSS CWE-79
JSON/AJAX Injection CWE-79	SQL Injection CWE-89	Blind SQL Injection CWE-89	LDAP Injection CWE-90
XML/XPath/XXE Injection CWE-91	Code Injection CWE-94	PHP File Inclusion CWE-98	HTTP Response Splitting CWE-113
Information Disclosure CWE-200	Authentication Bypass CWE-287	Arbitrary File Upload CWE-434	Open Redirect CWE-601

- To get more information about ImmuniWeb® SaaS please visit our website: http://www.htbridge.com/immuniweb/ or read Frost & Sullivan Movers & Shakers publication: «High-Tech Bridge Moves Ethical Hacking to the Cloud with ImmuniWeb® SaaS»

HIGH-TECH BRIDGE®
INFORMATION SECURITY SOLUTIONS

I. Web Application Security in Brief

II. ImmuniWeb® SaaS in Brief

III. ImmuniWeb® Self-Fuzzer Firefox Extension

- ImmuniWeb® Self-Fuzzer is a simple **Firefox browser extension** designed to detect **Cross-Site Scripting** and **SQL Injection** vulnerabilities in web applications.

- ImmuniWeb® Self-Fuzzer **is not a web application security scanner or crawler,** but a **real-time web fuzzer**. Once being activated by user in his browser, it carefully follows user's HTTP requests and fuzzes them in real time, carefully checking all HTTP parameters passed within the requests. Results of fuzzing are also displayed in real-time, notifying user immediately upon vulnerability detection.

- It demonstrates how rapidly and easily these two most common types of web vulnerabilities can be found even by a person who is not familiar with web security. It is a sort of **decision-making tool or Proof-of-Concept** for SMBs and private persons who hesitate whether to order ImmuniWeb® Security Assessment or not.

- The main advantage of ImmuniWeb® Self-Fuzzer is that it allows user to control the process of fuzzing in real-time, providing user with flexible configuration of the process.

- Quite often security scanners cannot follow human logic implemented in web application, or complex web 2.0 application structure, and simply do not arrive to certain “far away” zones of web applications. ImmuniWeb® Self-Fuzzer carefully follows the user and fuzzes all the HTTP requests user performs, efficiently replacing a crawler.

- Just carefully following the user in real-time, ImmuniWeb® Self-Fuzzer will not fuzz any potentially dangerous zones where arbitrary or non-standard HTTP requests may destroy the entire database behind the web application (of course if user won’t do it himself).

- ImmuniWeb® Self-Fuzzer can be safely used even in production environment, because it does not send any potentially dangerous requests that may damage database (it uses only “AND” operator checking for blind SQL injections, to prevent altering data in UPDATE, DELETE or INSERT SQL statements).

- The extension is user-friendly and does not require any technical skills from the user. Fuzzing process may be started and stopped by just one click.
- Being a Firefox extension it does not require any additional configuration of Firefox browser, which user have to do for proxy-based web security testing tools for example.
- The extension quickly searches Cross-Site Scripting and SQL Injection vulnerabilities in pages that user visit.
- It locks the fuzzing on one target domain to avoid accidental fuzzing of others websites, which user may not intend to scan.

- To detect vulnerabilities ImmuniWeb® Self-Fuzzer fuzzes various input vectors: **POST** and **GET** HTTP requests, **COOKIE** field and even the **URL** itself if HTTP parameters are encapsulated into the URL by mod-rewrite for example.
- User can manually select which input vectors to use. The vectors can be selected in the configuration panel of the extension.
- Once configured, the extension will automatically identify and fuzz all the variables passed via selected input vectors.
- As, already mentioned ImmuniWeb® Self-Fuzzer does not use “dangerous” injection patterns for blind SQL injection detection.

- Fuzzing technique varies depending on vulnerability type that we are trying to detect.

- For SQL injection detection ImmuniWeb® Self-Fuzzer will fuzz the variables using special SQL characters (to trigger errors), like :

Char	NUL	BS	TAB	LF	CR	SUB	"	%	'	\	_
Hex	0x00	0x08	0x09	0x0a	0x0d	0x1a	0x22	0x25	0x27	0x5c	0x5f

- XSS detection is performed by injecting HTML special characters (to check if they are filtered before being displayed), like :

Char	&	<	>	"	'	/
Hex	0x26	0x3C	0x3E	0x22	0x27	0x2F

- Example :

  http://www.site.com/page.php?var=1

  Vector : GET

  Variable : *var* ; Value : *1*

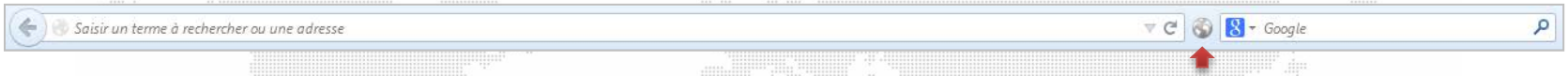
  Fuzzed Variable Value : 1’”\’\”\

- Vulnerabilities are detected by parsing the content returned by web server, depending on the HTTP request previously sent.
- For example to identify SQL injection vulnerability, ImmuniWeb® Self-Fuzzer searches for SQL database error messages.
- XSS vulnerabilities are identified by searching for non-filtered HTML special characters in page source code previously sent to the script.

- Download and install the extension from Firefox add-ons page:

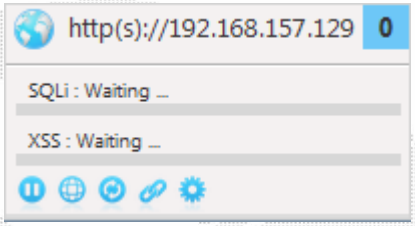
https://addons.mozilla.org/en/firefox/addon/immuniweb-self-fuzzer/

- Restart Firefox, and a little icon will appear next to the navigation bar:



- Go to the website you want to fuzz and click on the icon, the toolbar will appear with the name of the website. The number at the right displays the number of detected vulnerabilities:

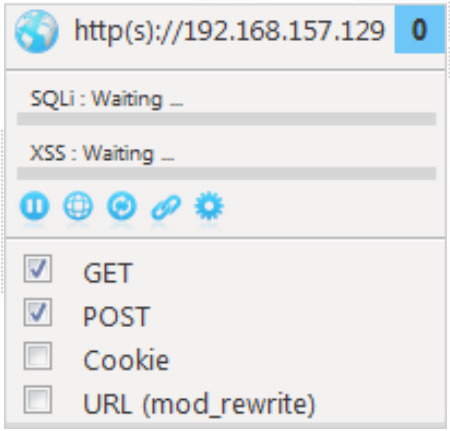
- Move the mouse cursor over the bar to display the control panel:



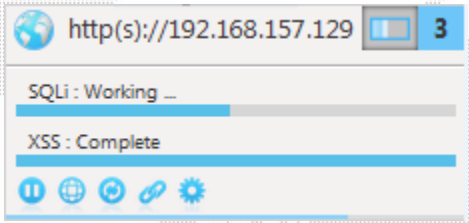
- There are 5 buttons on this panel :
  - Start/stop the fuzzing process (it queues captured packets)
  - Display results
  - Reset the counter of detected vulnerabilities
  - Enable/disable packets capture
  - Display configuration panel

- Two progress bars show the fuzzing status for each fuzzer (SQL injection and XSS), and the third one indicates the global progress of fuzzing (at the very bottom of the panel).

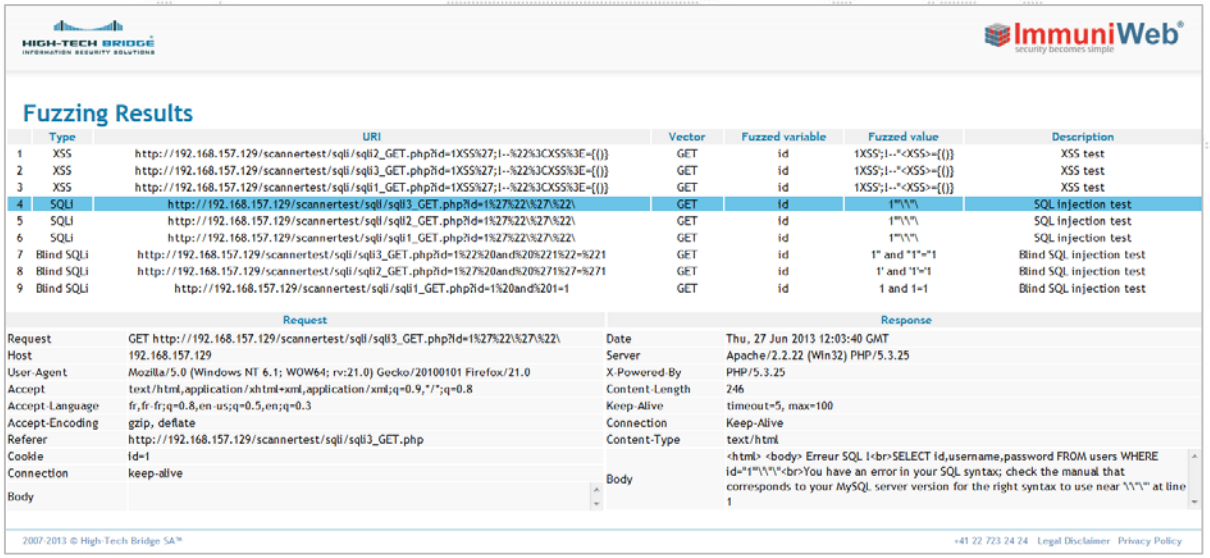
- Click the configuration button to choose which fuzzing vector(s) to use:



- After you have configured the extension, just browse the website you want, every variable used in the selected input vector will be automatically fuzzed while you are surfing:



- Results can be displayed by clicking the second button:



HIGH-TECH BRIDGE INFORMATION SECURITY SOLUTIONS — ImmuniWeb® security becomes simple

Fuzzing Results

	Type	URI	Vector	Fuzzed variable	Fuzzed value	Description
1	XSS	http://192.168.157.129/scannertest/sqli/sqli2_GET.php?id=1XSS%27;!--%22%3CXSS%3E={()}	GET	id	1XSS';!--"<XSS>={()}	XSS test
2	XSS	http://192.168.157.129/scannertest/sqli/sqli3_GET.php?id=1XSS%27;!--%22%3CXSS%3E={()}	GET	id	1XSS';!--"<XSS>={()}	XSS test
3	XSS	http://192.168.157.129/scannertest/sqli/sqli1_GET.php?id=1XSS%27;!--%22%3CXSS%3E={()}	GET	id	1XSS';!--"<XSS>={()}	XSS test
4	SQLi	http://192.168.157.129/scannertest/sqli/sqli3_GET.php?id=1%27%22\%27\%22\	GET	id	1"\'\"	SQL injection test
5	SQLi	http://192.168.157.129/scannertest/sqli/sqli2_GET.php?id=1%27%22\%27\%22\	GET	id	1"\'\"	SQL injection test
6	SQLi	http://192.168.157.129/scannertest/sqli/sqli1_GET.php?id=1%27%22\%27\%22\	GET	id	1"\'\"	SQL injection test
7	Blind SQLi	http://192.168.157.129/scannertest/sqli/sqli3_GET.php?id=1%22%20and%20%221%22=%221	GET	id	1" and "1"="1	Blind SQL injection test
8	Blind SQLi	http://192.168.157.129/scannertest/sqli/sqli2_GET.php?id=1%27%20and%20%271%27=%271	GET	id	1' and '1'='1	Blind SQL injection test
9	Blind SQLi	http://192.168.157.129/scannertest/sqli/sqli1_GET.php?id=1%20and%201=1	GET	id	1 and 1=1	Blind SQL injection test

Request		Response	
Request	GET http://192.168.157.129/scannertest/sqli/sqli3_GET.php?id=1%27%22\%27\%22\	Date	Thu, 27 Jun 2013 12:03:40 GMT
Host	192.168.157.129	Server	Apache/2.2.22 (Win32) PHP/5.3.25
User-Agent	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:21.0) Gecko/20100101 Firefox/21.0	X-Powered-By	PHP/5.3.25
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8	Content-Length	246
Accept-Language	fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3	Keep-Alive	timeout=5, max=100
Accept-Encoding	gzip, deflate	Connection	Keep-Alive
Referer	http://192.168.157.129/scannertest/sqli/sqli3_GET.php	Content-Type	text/html
Cookie	id=1		
Connection	keep-alive	Body	<html> <body> Erreur SQL ! SELECT id,username,password FROM users WHERE id="1"\'\" You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '\'\"' at line 1
Body			

2007-2013 © High-Tech Bridge SA™ — +41 22 723 24 24 — Legal Disclaimer — Privacy Policy

References

ImmuniWeb® Self-Fuzzer home page:
https://addons.mozilla.org/en/firefox/addon/immuniweb-self-fuzzer/

ImmuniWeb® Self-Fuzzer video:
http://www.youtube.com/watch?v=2USAnmzuRB8

CWE vulnerabilities glossary:
https://www.htbridge.com/vulnerability/

ImmuniWeb® SaaS home page:
https://www.htbridge.com/immuniweb/

Help Net Security web vulnerabilities statistics by Cenzic:
http://www.net-security.org/secworld.php?id=14556

OWASP project:
https://www.owasp.org/

Thank you for reading!

HIGH-TECH BRIDGE
INFORMATION SECURITY SOLUTIONS

Your questions are always welcome:

xavier.roussel [at] htbridge.com