Rapid Assessment of Web Resources (RAWR!)

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

In memory and honor of Leonard Nimoy (1931-2015):
"Of all the souls I have encountered in my travels, his was the most... human."

Prerequisites

Typically *nix, tested here on Ubuntu & Kali

Kali and Ubuntu recommended, virtual machine or physical

Overview

Confession, and it shouldn't be a shocker: I'm a huge military science fiction fan. As such, John Ringo is one of my absolute favorites. I'm in the midst of his Black Tide Rising campaign, specifically Book 2, *To Sail a Darkling Sea*. As we contemplate this month's *ISSA Journal* topic, Security Architecture/Security Management, let's build a bit on this theme of dark seas as we look out at our probable futures. Keren Elazari (she is speaking at RSA 2015, where you may well be reading this in print), in her April 2015 *Scientific American* article, "How To Survive Cyberwar," states that "in the coming years, cyberattacks will almost certainly intensify, and that is a problem for all of us. Now that everyone is connected in some way to cyberspace—through our phones, our laptops, our corporate networks—we are all vulnerable." If you haven't caught up with this perspective yet, wake up. I'm a Premera customer, you with me? Ringo starts Chapter 1 of *To Sail a Darkling Sea* with Sir Edmund Burke: "When bad men combine, the good must associate; else they will fall, one by one, an unpitied sacrifice in a contemptible struggle."

You'll therefore forgive me if I stay on a bit of a pentesting and assessment run, having just covered Faraday last month, but it's for good reason. While unable to be specific, I can tell you that at multiple intervals in the last few of months I have seen penetration testing and the brilliant testers executing them provide extraordinary value for their "customers." One way to try and avoid sailing the darkling, compromised seas of the Intarwebs is with a robust penetration testing program, as integral to security management as any governance, risk, and compliance program. Bad men combine, we know that; the good must pentest. ☺

Let's put philosophy into action this month with Adam Byers' RAWR[1] (NJ Ouchn, our friend @toolswatch, is on the RAWR team too). I asked Adam for the typical tool author's contribution to the column and was treated to such robust content that I'm going to take a slightly different approach this month where I'll weave in Adam's feedback throughout as we take RAWR on a walkabout. For a proper introduction per Adam: "RAWR was designed to ease the process of the mapping, discovery, and reporting phases of an assessment with a focus primarily on web resources. It was built to be quick, scalable, and productive for the assessor. From the ground up, it accepts input from multiple different known scanning solutions, as well as leveraging Nmap if no pre-existing scan data is available. The goal of RAWR is to consolidate and capture the pieces of information that are most useful while performing a web assessment, and produce output that is normalized and functional. There are many common checks performed in the process of a web assessment, each yielding information that is useful. The problem is that there are many different tools capable of gathering these singular bits of data, and each one produces output unlike the last. This further complicates the job that the assessor is tasked to perform, because producing a report that effectively compiles all of this information is the end goal. With RAWR we want to take any type of relevant input, perform enumeration, and effectively pass the data on to the next phase—whether that phase be a tool, a person, or the end report itself."

If you've conducted an assessment at any time, unless you're working for one of those cookie cutter, CEH-certified, checklist-compliance mills, you've run into the issue of many different tools gathering data but generating output unlike the last. Outstanding efforts such as RAWR contribute greatly to the collaborate, combine, and complete cause, resulting in improved results and happier customers.

Adam also indicated that in addition to assessments, RAWR is useful during audits. Its ability to capture artifacts such as screenshots of disclaimer statements and login banners drastically reduces the amount of time it takes to complete audit tasks. The RAWR road map includes full header specification (including cookies), email/SMS notifications (already in the dev branch), better DNS functionality, and the acceptance of new input formats. The RAWR development pipeline also includes database integration for comparison and differential analysis of historical scan data via a PostgreSQL database, allowing trending as an example.

With that in mind, let's begin our exploration.

1 https://bitbucket.org/al14s/rawr/wiki/Home.

url	ipv4	port	returncode	hostnames	title	robots	script	file_includes
http://holisticinfosec.org	70.40.197.78	80	200	['holisticinfosec.org' '70-40-197-78.unifiedlayer.com']	welcome | HolisticInfoSec	y	0	['/themes/forging/assets/js/libs/modernizr-2.6.2.mi
https://holisticinfosec.org	70.40.197.78	443	200	['holisticinfosec.org' '70-40-197-78.unifiedlayer.com']	welcome | HolisticInfoSec	y	0	['/themes/forging/assets/js/libs/modernizr-2.6.2.mi

Figure 2 – RAWR server information

IP	HOSTNAME	80	443		TOTAL
70.40.197.78	holisticinfosec.org	x	x		2
TOTAL		1	1		

Figure 1 – RAWR Threat Matrix results

Ready RAWR

Some quick installation and setup notes. I initially installed RAWR on Kali but received rather buggy and incomplete results. Suspecting more of a Kali libs issue versus a RAWR shortcoming, I moved to an Ubuntu instance and received far better results.

At an Ubuntu terminal prompt, I executed:

```
git clone https://bitbucket.org/al14s/rawr.git
cd rawr
sudo ./install.sh
```

The RAWR installer will help acquire dependencies including Ghost or phantomJS, python-lxml for parsing XML and HTML, and python-pygraphviz to create PNG diagrams from a site crawl. You'll be asked to confirm for installation of missing dependencies; on Kali Nmap is native, but phantomJS, DPE (from @toolswatch), and others will need to be installed. If you end up with pygraphviz errors on Ubuntu you may need to force installation with `sudo apt-get install python-pygraphviz` then run `./install.sh -u`.

Run RAWR!

Adam provided us details for a typical assessment with RAWR; we'll follow his steps and provide results as we go along.

Adam: A typical assessment begins with performing enumeration with RAWR. Executing something like `rawr <scope> -a -o -r -x -S3 --dns` gives us quite a bit of information to work with.

toolsmith: I ran `./rawr.py holisticinfosec.org -p fuzzdb -a -o -r -x -S3 –ssl --dns`. To use FuzzDB Common Ports, I set `-p fuzzdb`. The `-a` switch is used to include all open ports in the CSV output and the Threat Matrix. The `–o` switch grabs HTTP OPTIONS, `-r` acquires robots.txt, and `–x` pulls down crossdomain.xml.[2] The `-S3` flag controls crawl intensity, 3 is default, `–ssl` tells Nmap to call enum-ciphers.nse for more in-depth SSL data, and `--dns` queries Bing for other hostnames and adds them to the queue. Results are written to a log directory, specifically `/rawr/log_20150321-143627_rawr` on my Ubuntu instance. Therein, a cornucopia of useful results abound.

Adam: The Attack Surface (Threat) Matrix provides a quick view of ports on hosts, as well as patterns that reveal clustered services, similar host configurations, HA setups, and potential firewalls.

toolsmith: As seen in figure 1, my Threat Matrix does not offer much to be concerned about, just 80 and 443 listening.

Adam: We can take serverinfo.csv and use it as a checklist, notating any interesting hosts and possible vulnerabilities. Because we specified `-a` in the command line, all port data is included in the output regardless of whether or not it is web based.

toolsmith: The server information feature returns results for url, ipv4, port, returncode, hostnames, title, robots, script, file_includes, ssl_cert-daysleft, ssl_cert-validityperiod, ssl_cert-md5, ssl_cert-sha-1, ssl_cert-notbefore, ssl_cert-notafter, cpe, cve, service_version, server, endurl, date, content-type, description, author, revised, docs, passwordfields, email_addresses, html5, comments, defpass, and diagram. As seen in figure 2, my serverinfo.csv provides a sample of such details.

Adam: The "index" HTML report provides a dynamic (jQuery-driven) way to sift through screenshots and information captured from each host. While performing a site spider, RAWR pulls metadata from any docs found, which usually hands us a list of usernames, email addresses, domains, server names, phone numbers, and more in an HTML format, linked to in the index report. Also available via the index is a report that addresses the security headers of the target scope, alerting the assessor of improperly configured websites and services.

toolsmith: I'll share the results of each of the three succinct reports generated. Figure 3 represents the initial index report.

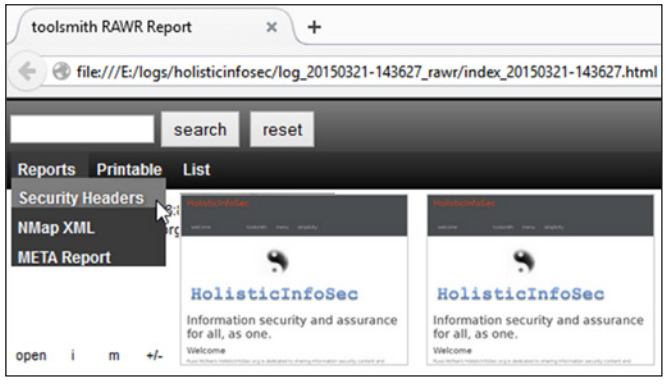

Figure 3 – RAWR Index Report

Figure 4 indicates the Nmap results.

Figure 5 displays the security headers report. This report includes definitions from https://securityheaders.com to provide clarity.

Figure 6 provides the results of the metadata extracted during the RAWR scan.

Adam: For web-focused testing, we also use the `--proxy` switch to push all of the traffic through Portswigger's BurpSuite or OWASP Zed Attack Proxy. RAWR isn't a vulnerability scanner, so it's helpful to leverage an interception proxy for further scans and session data when we go hunting. Most IDS/IPS configurations will not alert on any of RAWR's ac-

2 http://jeremiahgrossman.blogspot.com/2008/05/crossdomainxml-invites-cross-site.html.

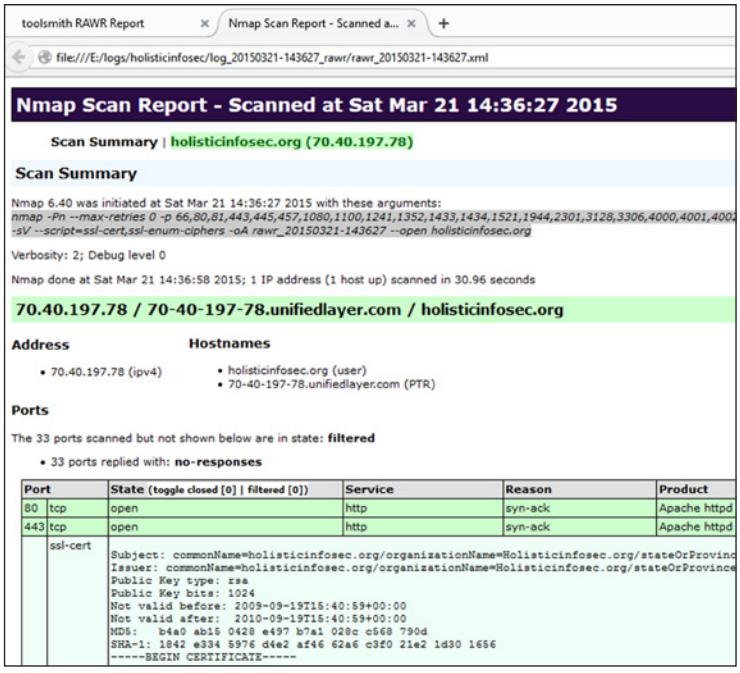

Figure 4 – RAWR Nmap Scan Report

tivity aside from the initial Nmap scan, as every request is a valid HTTP call to a verified host.

toolsmith: I ran RAWR separately through Burp with the `--proxy` switch enabled; this is a fabulous way to build an initial footprint, then proceed with more aggressive web application security testing. You can customize your RAWR scans via /rawr/conf/settings.py, including the user-agent (useful when you want to be uniquely identified during assessments), Nmap speed, CSV settings, spider aggression, and default ports.

Keep in mind, you should only be using RAWR against resources you are authorized to assess.

In conclusion

Adam reminds us that no one tool does it all and that it would be great to see more integration and data synchronicity between different tool sets. RAWR developers seek to overcome this by facilitating its acceptance of multiple input formats, as well as outputs like JSON, CSV, ShelvDB, and the aforementioned planned PostgreSQL integration. There are also word lists for hydra and input lists for Nmap, as well as other lists with Dirbuster, Nikto, and MetaSploit in mind. As information security tools developers work to build tools that are modular and scalable, so too should they consider making them compatible. Great work here by Adam and team, I really look forward to continued RAWR development and the principles it aligns with. A bright beacon in the darkling sea, if you will.

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

Cheers…until next month.

Acknowledgements

—Adam Byers (@al14s), RAWR (@RapidWebEnum) project lead

—Tom Moore (@c0ncealed), RAWR (@RapidWebEnum) project lead

About the Author

Russ McRee manages the Threat Intelligence & Engineering team 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 or @holisticinfosec.

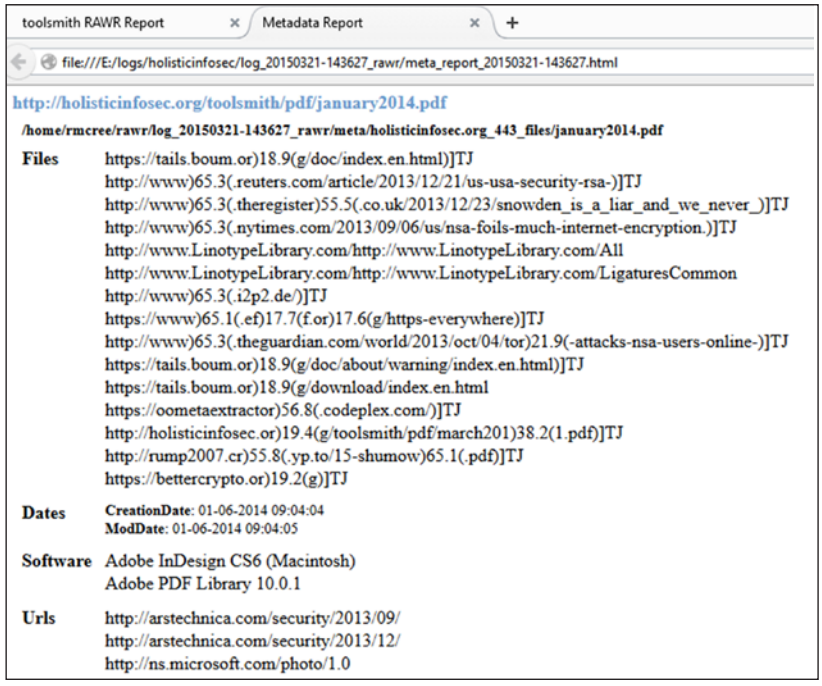

Figure 6 – RAWR Metadata Report

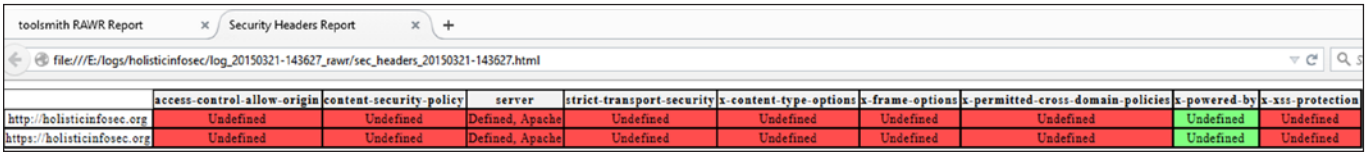

Figure 5 – RAWR Security Headers Report