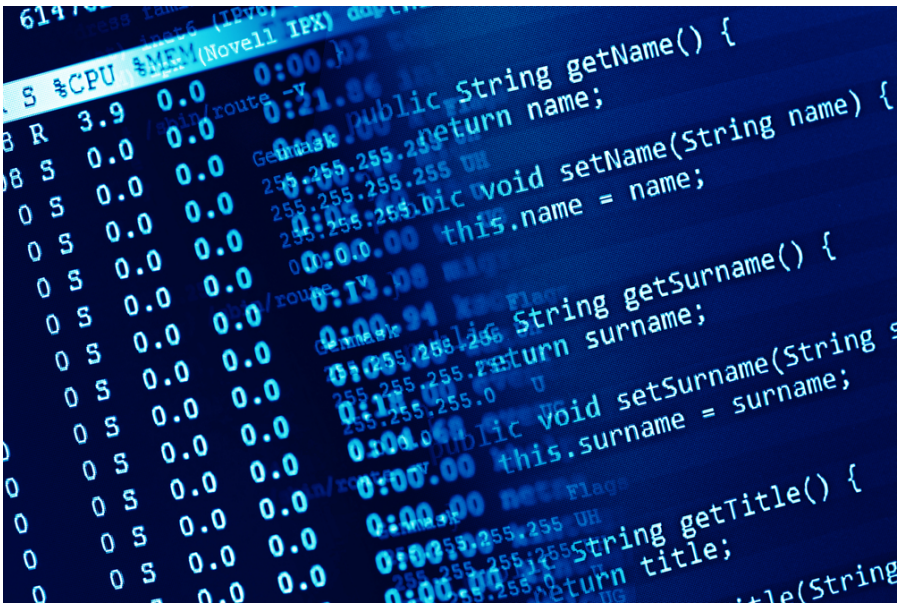
Mo’ Shells Mo’ Problems – Web Server Log Analysis

March 19, 2014 Chad Tilbury Research & Threat Intel



Disclaimer: CrowdStrike derived this information from investigations in unclassified environments. Since we value our clients’ privacy and interests, some data has been redacted or sanitized.

Web shells epitomize the hacking tenant of hiding in plain sight. In a previous post, we showed how a web shell could hide as a single file among thousands present on a web server and as a single line of code in an otherwise legitimate page on a site. The best web shells are not detected by anti-virus and can defeat vulnerability scanning applications using novel techniques like cookie and HTTP header authentication. Identifying the presence of a web shell can be difficult, but there are effective and repeatable ways to find them in your network. Today we will cover log review, concentrating on the following techniques:

- SQL injection identification
- Directory enumeration

- Statistical web log analysis

Get to Know Your Web Logs

Good logging is a requirement for successfully detecting and mitigating network intrusions. Luckily, web servers have far better logging available by default than other servers in the enterprise. It is common to find web server logs dating back a year or more, but web logs are typically stored in text format and are particularly at risk for deletion and modification by attackers. Given their intrinsic value, organizations should aim for daily aggregation and centralized archiving of web logs. Their location can be highly dependent on the operating system and server version, but the following locations are a good place to start:

- C:inetpublogsLogFiles (IIS)
- C:WindowsSystem32LogFiles (IIS)
- /var/log/httpd/ (Apache)
- /var/log/apache/ (Apache)

There are three popular standards for web logging. The techniques discussed here will work for any format, but some formats provide significantly less data to analyze. Apache servers often employ the NCSA, or Common Log Format. This format tends to record less information than others do. The W3C extended log file format, commonly used by Microsoft IIS, provides the most data and is the best-case scenario from a log analysis perspective. W3C extended logs can provide additional helpful information such as the client query, time duration of the request, and user agent. You may also see the proprietary IIS log format, which provides more information than NCSA, but less than W3C extended logs.

SQL Injection Identification

As discussed in earlier posts in this series, Deep Panda is a sophisticated China based threat group CrowdStrike has observed targeting companies in the defense, legal, telecommunication and financial industries. Deep Panda often

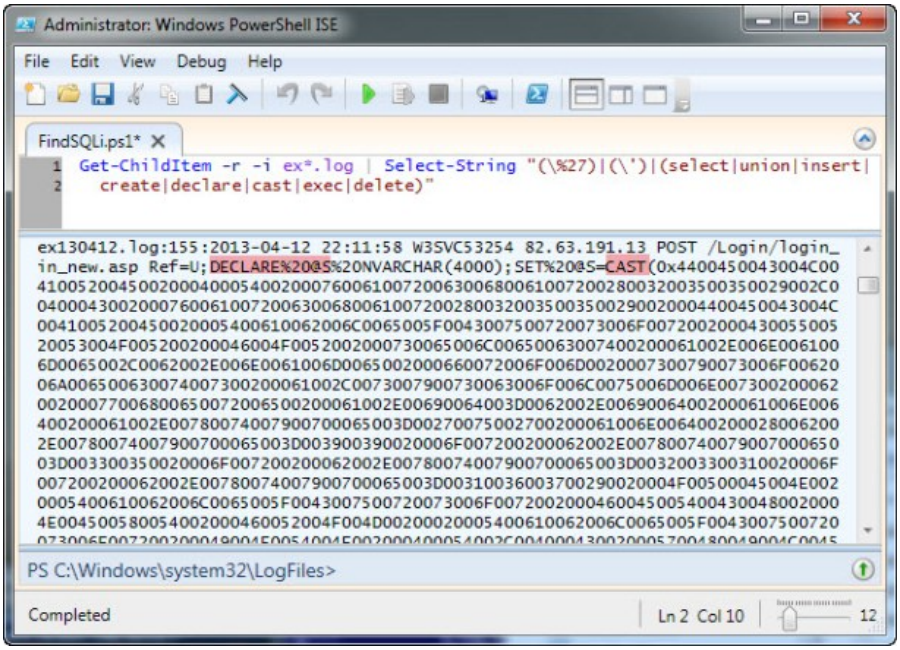
gains entry into an environment by exploiting vulnerabilities in poorly patched web servers or web servers running legacy or custom applications. Detecting successful SQL injection is an effective way to identify attacks early in the kill chain. Finding SQL injection attempts also provides a good starting point for discovering additional malicious activity. Here are a few things to look for to help narrow the search:

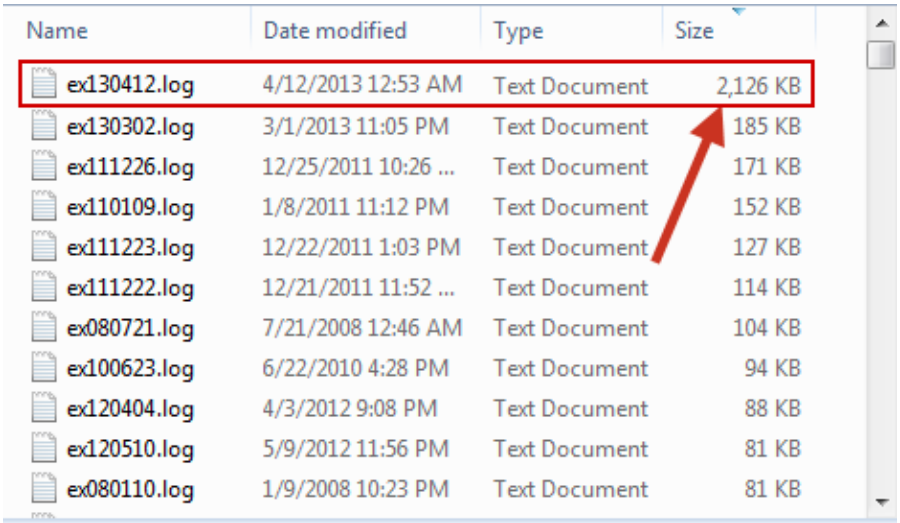
· Investigate server logs that are significantly larger compared to others. SQL injection is very noisy and requires numerous connection attempts to a web server to be successful. Automated SQL injection tools are particularly noisy. This activity can create many large entries, causing the daily logs to exceed the average size.

· SQL commands should be a very rare occurrence in standard web logs. Search for commands often passed during SQL injection such as ‘, %27, –, SELECT, INSERT, UNION, CREATE, DECLARE, CAST, EXEC, and DELETE (this is only a subset and should be tailored to your environment). Regular expression searching with grep or PowerShell can lead to quick wins.

· Identify HTTP 500, 404, 403 and 400 status codes that occur in long successions within your logs. This will help identify enumeration and patterns typical of SQL injection attacks.

· On IIS servers, look for references to “cmd.exe” and “xp_cmdshell” to identify possible privilege escalation due to successful SQL injection exploitation. Execution of these commands is often the ultimate goal of an attacker. If you find successful entries (HTTP status code 200) containing these commands, you likely have a confirmed intrusion.





Directory Enumeration

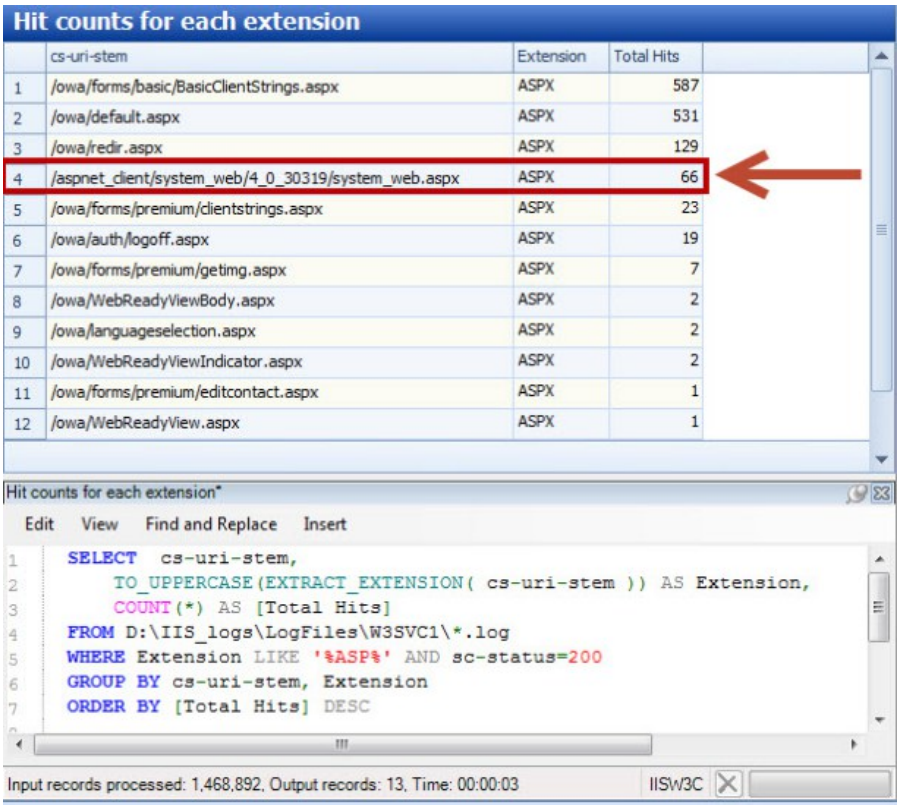
Enumerating the web server file system is a common way for an adversary to identify the type of scripting language a web server is running and what scripts can be used to further escalate privileges. Directory enumeration is a noisy technique, and one of the easiest to spot. In the example below, the IP address 60.166.3.22 enumerated an IIS web server. Shortly after these requests, we discovered a successful SQL injection attack recorded in the logs.

```
2013-03-20 19:03:11 60.166.3.22 5081 192.168.100.11 80 HTTP/1.1 GET
/images"OTA2NjAw%40 400 - URL -

2013-03-20 19:03:36 60.166.3.22 5083 192.168.100.11 80 HTTP/1.1 GET
/Login//..%5c..%5c..%5c..%5c..%5c..%5c..%5cetc/passwd 403 -
Forbidden -

2013-03-20 19:03:36 60.166.3.22 5109 192.168.100.11 80 HTTP/1.1 GET
/Login//..%5c..%5c..%5c..%5c..%5c..%5c..%5cwindows/win.ini 403 -
Forbidden -

2013-03-20 19:03:37 60.166.3.22 5093 192.168.100.11 80 HTTP/1.1 GET
/Login//../../../../../../../etc/passwd 403 - Forbidden -
```

Statistical Web Log Analysis

There is more to web log analysis than just looking for SQL injection patterns and enumeration attacks. Web logs can stretch back for years and contain hundreds of millions of entries. To make review feasible, we often use statistical analysis to analyze web log entries in different ways. Similar to our previous post on file system stacking, the goal of statistical analysis is to identify outliers for further analysis. If you do not have an enterprise tool that can do this, Microsoft Log Parser and Log Parser Lizard (shown below) are fantastic free options. Log Parser takes nearly any log format as input and allows analysis via SQL queries.

In Figure 4 below, we executed a query to collect all successful (status code 200) .asp and .aspx page requests within the log, count them, and then sort by the total number of entries. Web shells in IIS servers often take the form of .asp pages and we were looking for outliers, either in the path or in the number of hits. As you can see, system_web.aspx is immediately recognizable as an outlier based upon its path. In fact, out of ~1.5 million log entries on this server, it was the only .aspx page successfully requested outside of the /owa/ folder. Similar queries are useful for other requests such as those referencing .php, .exe, or .dll files.



Hit counts for each extension

	cs-uri-stem	Extension	Total Hits
1	/owa/forms/basic/BasicClientStrings.aspx	ASPX	587
2	/owa/default.aspx	ASPX	531
3	/owa/redir.aspx	ASPX	129
4	/aspnet_client/system_web/4_0_30319/system_web.aspx	ASPX	66
5	/owa/forms/premium/clientstrings.aspx	ASPX	23
6	/owa/auth/logoff.aspx	ASPX	19
7	/owa/forms/premium/getimg.aspx	ASPX	7
8	/owa/WebReadyViewBody.aspx	ASPX	2
9	/owa/languageselection.aspx	ASPX	2
10	/owa/WebReadyViewIndicator.aspx	ASPX	2
11	/owa/forms/premium/editcontact.aspx	ASPX	1
12	/owa/WebReadyView.aspx	ASPX	1

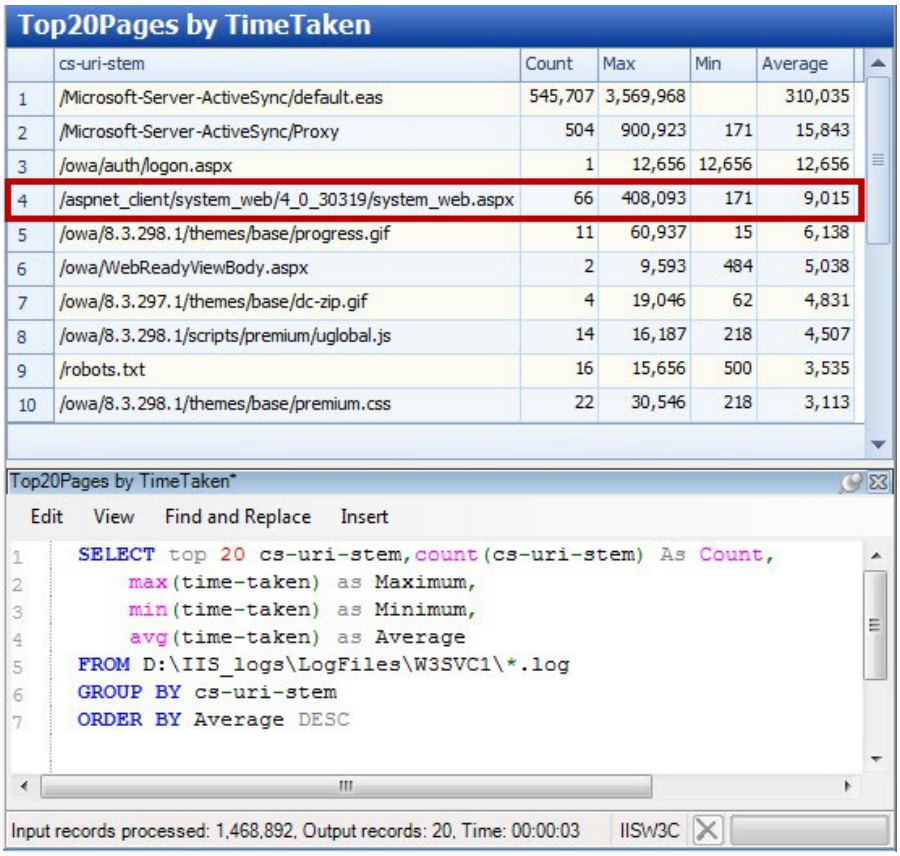
Hit counts for each extension*

Edit View Find and Replace Insert

```
1  SELECT  cs-uri-stem,
2      TO_UPPERCASE(EXTRACT_EXTENSION( cs-uri-stem )) AS Extension,
3      COUNT(*) AS [Total Hits]
4  FROM D:\IIS_logs\LogFiles\W3SVC1\*.log
5  WHERE Extension LIKE '%ASP%' AND sc-status=200
6  GROUP BY cs-uri-stem, Extension
7  ORDER BY [Total Hits] DESC
```

Input records processed: 1,468,892. Output records: 13. Time: 00:00:03 IISW3C

A second analysis example takes advantage of the W3C extended log option to track the amount of time taken by the web server to process a request. Administrators often use this field to optimize the server or find problematic and broken pages. We use it to find problematic pages of a different type. Because web shells and other malicious code often contain complicated queries and requests, they can take longer to process than an average page and tend to hang on some requests. Both scenarios cause those pages to bubble up to the top of a list like this, and here we see our Deep Panda web shell showing up in fourth place on this compromised system.



Similar to other analysis techniques, there is no easy button

possible, find an administrator or subject matter expert to assist you in determining what is normal on your servers. Web logs are a rich data source and possible queries are only limited by your imagination. As you discover novel queries that help identify evil in your environment, automate them to create daily reports for review.

Additional Logs

Web server logs are undoubtedly valuable, but do not underestimate the other log sources available on your server. System logs such as Windows event logs or Apache error logs can provide deeper insight and help pinpoint when malicious activity has occurred. As an example, the Application event log on a compromised IIS server held the event seen in Figure 6. The event was logged when a SQL injection attack successfully executed the xp_cmdshell stored procedure. Correlating the time and date of this event with the web server logs for that day allowed reconstruction of the entire attack in a short amount of time.

```
Description:

The description for Event ID ( 17055 ) in Source ( MSSQLSERVER ) could
not be found.

Either the component that raises this event is not installed on the
computer or the installation is corrupted. You can install or repair the
component or try to change Description Server.

The following information was included with the event (insertion
strings):

8128

Using 'xplog70.dll' version '2000.80.2039' to execute extended stored
procedure 'xp_cmdshell'.
```

Server logs are an invaluable resource for detecting intrusions. While the examples in this post have focused on web shell detection, it is important to note that these techniques are useful for finding just about any malicious activity on your web servers. Building a log collection and review process is one of the most important things you can do to ensure the health of your servers.

Coming next: In our final post of the four-part series, we will transition to network-based detection methods. Registration is now open for our April 1, 2014 CrowdCast, Going Beyond the Indicator.

Chad Tilbury

Chad Tilbury has over 15 years experience investigating computer crimes, specializing in intrusion incident response, digital forensic examinations, and corporate espionage. His extensive law enforcement and international experience stems from working with a broad cross-section of Fortune 500 corporations and government agencies around the world. As faculty with the SANS Institute in digital forensics, Chad is responsible for educating thousands of students each year in advanced forensics and incident response techniques. As Technical Director for CrowdStrike, Chad provides leadership for the services team, driving innovation to support customers in a variety of

offerings. Chad is a graduate of the U.S. Air Force Academy and holds a M.S. and B.S. in Computer Science and GCFA, GREM, GCIH, ENCE, and CISSP certifications.

Related Content

Registry Analysis with CrowdResponse

The third release of the free CrowdResponse in response collection tool is now available! This time around we…

Peering Around the Corner

After the better part of a decade chasing adversaries around the Internet, there are a few…

Cyber Kung-Fu: The Great Firewall Art of DNS Poisoning

Wing Chun (咏春拳), the first Chinese martial art learned by the legendary Bruce Lee, is often…