Mo' Shells Mo' Problems – File List Stacking

March 7, 2014 RyanJ From The Front Lines, Research & Threat Intel

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

In our first blog post, "Mo' Shells Mo' Problems: Deep Panda Web Shells – Part 1", we discussed two web shells leveraged by a Chinese threat group we call Deep Panda. In case you forgot, a web shell is a file containing backdoor functionality written in a web scripting language such ASP, ASPX, PHP, JSP or CFM. When a web shell is hosted on an internet facing victim system, an adversary can remotely access the system to perform malicious actions.

Today we'll cover one of three ways to help hunt for web shells in your environment: file stacking. We often use this method during investigations to help identify web shells in environments where they would normally go unnoticed. An adversary typically creates web shells on only a small number of servers in a victim environment, providing us with a "needle in many haystacks" problem. Luckily, most servers in an organization have very little end user interaction and contain far less user artifacts than a typical workstation. As a result, it is typically easier to identify anomalies on servers.

The main goal of file stacking is to identify outliers. By identifying uncommon files across a set of servers, we hope to find the most uncommon file type of all, malware.

In the case of web shells, there are certain search criteria we may want to adopt to speed up our efforts:

- If the environment is a mix of Windows Server and other web hosting solutions (such as Apache Tomcat and Adobe Cold Fusion), search for extensions of ASP, ASPX, JSP, PHP and CFM. Focus searches on %systemdrive%Inetpubwwwroot, %programfiles%Apache Software Foundation, %systemdrive%ColdFusion<version> and other file paths to 3rd party web hosting solutions in your environment.

- If the environment is Windows based with no other 3rd party web hosting solutions, limit search to %systemdrive%Inetpubwwwroot and search for extensions ASP and ASPX. This will restrict to searching only for scripts that are accessible externally from the internet.

File Stacking with Timestamps

To illustrate file stacking, consider the following real world network:

- Predominately Windows environment

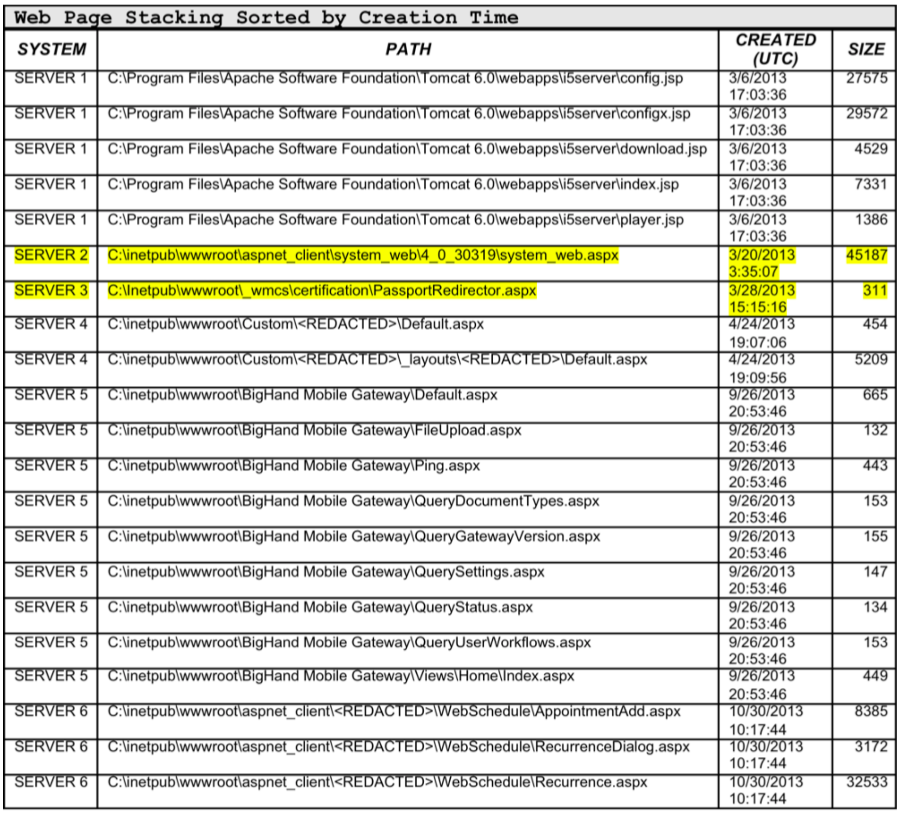
- About 200 Servers

- Usage of third party web hosting solutions

In this example, the wide variety of custom web applications causes many legitimate and unique web scripts across the enterprise. It would be extremely difficult to identify web shells solely based upon file occurrences in this environment. However, by restricting the search to the aforementioned criteria and sorting by file creation time, we can identify uncommon web scripts not created at similar times to one another. We would normally expect groupings of web scripts to be created in succession if they are part of a normal deployment by an administrator or part of an installation. Files that do not adhere to this model are worth looking at further. Our method of filtering by file path and extension, coupled with sorting by file creation time, helps identify suspicious outliers. Pretty straightforward, right? Here's how it looks in this network:

Ignoring my previous search criteria advice and searching for all the aforementioned extensions, results in 179,231 files found on the 200 servers in our example environment. I don't know about you, but I don't often have the time to sift through 179,231 web scripts. Limiting our search criteria to specific directories results in 1,671 files, a significant improvement. Timestamp analysis can be used to further reduce this set, and we were most interested in files created between April and November 2013.

The below table shows an abbreviated listing of the file stacking results for the 200 windows servers in the sample victim organization, sorted by file creation time:



Web Page Stacking Sorted by Creation Time

SYSTEM	PATH	CREATED (UTC)	SIZE
SERVER 1	C:\Program Files\Apache Software Foundation\Tomcat 6.0\webapps\i5server\config.jsp	3/6/2013 17:03:36	27575
SERVER 1	C:\Program Files\Apache Software Foundation\Tomcat 6.0\webapps\i5server\configx.jsp	3/6/2013 17:03:36	29572
SERVER 1	C:\Program Files\Apache Software Foundation\Tomcat 6.0\webapps\i5server\download.jsp	3/6/2013 17:03:36	4529
SERVER 1	C:\Program Files\Apache Software Foundation\Tomcat 6.0\webapps\i5server\index.jsp	3/6/2013 17:03:36	7331
SERVER 1	C:\Program Files\Apache Software Foundation\Tomcat 6.0\webapps\i5server\player.jsp	3/6/2013 17:03:36	1386
SERVER 2	C:\inetpub\wwwroot\aspnet_client\system_web\4_0_30319\system_web.aspx	3/20/2013 3:35:07	45187
SERVER 3	C:\Inetpub\wwwroot_wmcs\certification\PassportRedirector.aspx	3/28/2013 15:15:16	311
SERVER 4	C:\inetpub\wwwroot\Custom\<REDACTED>\Default.aspx	4/24/2013 19:07:06	454
SERVER 4	C:\inetpub\wwwroot\Custom\<REDACTED>_layouts\<REDACTED>\Default.aspx	4/24/2013 19:09:56	5209
SERVER 5	C:\inetpub\wwwroot\BigHand Mobile Gateway\Default.aspx	9/26/2013 20:53:46	665
SERVER 5	C:\inetpub\wwwroot\BigHand Mobile Gateway\FileUpload.aspx	9/26/2013 20:53:46	132
SERVER 5	C:\inetpub\wwwroot\BigHand Mobile Gateway\Ping.aspx	9/26/2013 20:53:46	443
SERVER 5	C:\inetpub\wwwroot\BigHand Mobile Gateway\QueryDocumentTypes.aspx	9/26/2013 20:53:46	153
SERVER 5	C:\inetpub\wwwroot\BigHand Mobile Gateway\QueryGatewayVersion.aspx	9/26/2013 20:53:46	155
SERVER 5	C:\inetpub\wwwroot\BigHand Mobile Gateway\QuerySettings.aspx	9/26/2013 20:53:46	147
SERVER 5	C:\inetpub\wwwroot\BigHand Mobile Gateway\QueryStatus.aspx	9/26/2013 20:53:46	134
SERVER 5	C:\inetpub\wwwroot\BigHand Mobile Gateway\QueryUserWorkflows.aspx	9/26/2013 20:53:46	153
SERVER 5	C:\inetpub\wwwroot\BigHand Mobile Gateway\Views\Home\Index.aspx	9/26/2013 20:53:46	449
SERVER 6	C:\inetpub\wwwroot\aspnet_client\<REDACTED>\WebSchedule\AppointmentAdd.aspx	10/30/2013 10:17:44	8385
SERVER 6	C:\inetpub\wwwroot\aspnet_client\<REDACTED>\WebSchedule\RecurrenceDialog.aspx	10/30/2013 10:17:44	3172
SERVER 6	C:\inetpub\wwwroot\aspnet_client\<REDACTED>\WebSchedule\Recurrence.aspx	10/30/2013 10:17:44	32533

Referencing Table 1, the web scripts on Servers 1 and 4-6 appear to be created contemporaneous with one another and all reside in the same directories. This type of activity is likely a result of legitimate installation of a 3rd party application or other custom applications used for legitimate purposes within the organization. However, you can also see two particular web scripts that stand out:

- C:inetpubwwwrootaspnet_clientsystem_web<VERSION>system_web.aspx

- C:Inetpubwwwroot_wmcscertificationPassportRedirector.aspx

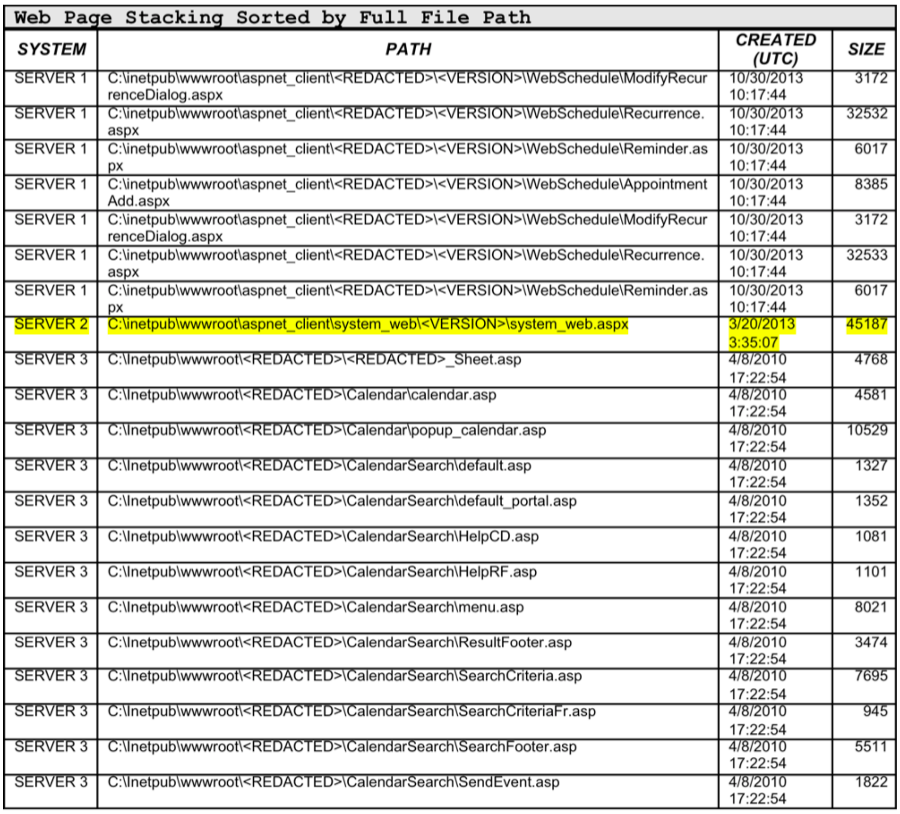
Both of these scripts were created at isolated times from all other web scripts created between our target dates of April 2013 and November 2013. Further investigation of PassportRedirector.aspx, showed it to be a false positive related to the Windows Rights Management – Account Certification Service.

Taking a closer look at System_web.aspx shows us it was created at an atypical time of 3:35:17 UTC (Roughly 23:35:17 EDT). This suggests it may not have been created by an administrator during normal business hours. Upon further inspection, this file indeed was a web shell covered in our first post “Mo Shells Mo' Problems: Deep Panda Web Shells – Part 1”.

File Stacking with File Paths

We can also search on full file path, using the same data above, to identify web scripts that are not grouped with others. Additionally, we'll be on the lookout for files in unusual or unique locations.

Table 2 shows an abbreviated listing of the file stacking results for the 200 windows servers in our sample victim organization, sorted by full file path:



Web Page Stacking Sorted by Full File Path

SYSTEM	PATH	CREATED (UTC)	SIZE
SERVER 1	C:\inetpub\wwwroot\aspnet_client\<REDACTED>\<VERSION>\WebSchedule\ModifyRecurrenceDialog.aspx	10/30/2013 10:17:44	3172
SERVER 1	C:\inetpub\wwwroot\aspnet_client\<REDACTED>\<VERSION>\WebSchedule\Recurrence.aspx	10/30/2013 10:17:44	32532
SERVER 1	C:\inetpub\wwwroot\aspnet_client\<REDACTED>\<VERSION>\WebSchedule\Reminder.aspx	10/30/2013 10:17:44	6017
SERVER 1	C:\inetpub\wwwroot\aspnet_client\<REDACTED>\<VERSION>\WebSchedule\AppointmentAdd.aspx	10/30/2013 10:17:44	8385
SERVER 1	C:\inetpub\wwwroot\aspnet_client\<REDACTED>\<VERSION>\WebSchedule\ModifyRecurrenceDialog.aspx	10/30/2013 10:17:44	3172
SERVER 1	C:\inetpub\wwwroot\aspnet_client\<REDACTED>\<VERSION>\WebSchedule\Recurrence.aspx	10/30/2013 10:17:44	32533
SERVER 1	C:\inetpub\wwwroot\aspnet_client\<REDACTED>\<VERSION>\WebSchedule\Reminder.aspx	10/30/2013 10:17:44	6017
SERVER 2	C:\inetpub\wwwroot\aspnet_client\system_web\<VERSION>\system_web.aspx	3/20/2013 3:35:07	45187
SERVER 3	C:\Inetpub\wwwroot\<REDACTED>\<REDACTED>_Sheet.asp	4/8/2010 17:22:54	4768
SERVER 3	C:\Inetpub\wwwroot\<REDACTED>\Calendar\calendar.asp	4/8/2010 17:22:54	4581
SERVER 3	C:\Inetpub\wwwroot\<REDACTED>\Calendar\popup_calendar.asp	4/8/2010 17:22:54	10529
SERVER 3	C:\Inetpub\wwwroot\<REDACTED>\CalendarSearch\default.asp	4/8/2010 17:22:54	1327
SERVER 3	C:\Inetpub\wwwroot\<REDACTED>\CalendarSearch\default_portal.asp	4/8/2010 17:22:54	1352
SERVER 3	C:\Inetpub\wwwroot\<REDACTED>\CalendarSearch\HelpCD.asp	4/8/2010 17:22:54	1081
SERVER 3	C:\Inetpub\wwwroot\<REDACTED>\CalendarSearch\HelpRF.asp	4/8/2010 17:22:54	1101
SERVER 3	C:\Inetpub\wwwroot\<REDACTED>\CalendarSearch\menu.asp	4/8/2010 17:22:54	8021
SERVER 3	C:\Inetpub\wwwroot\<REDACTED>\CalendarSearch\ResultFooter.asp	4/8/2010 17:22:54	3474
SERVER 3	C:\Inetpub\wwwroot\<REDACTED>\CalendarSearch\SearchCriteria.asp	4/8/2010 17:22:54	7695
SERVER 3	C:\Inetpub\wwwroot\<REDACTED>\CalendarSearch\SearchCriteriaFr.asp	4/8/2010 17:22:54	945
SERVER 3	C:\Inetpub\wwwroot\<REDACTED>\CalendarSearch\SearchFooter.asp	4/8/2010 17:22:54	5511
SERVER 3	C:\Inetpub\wwwroot\<REDACTED>\CalendarSearch\SendEvent.asp	4/8/2010 17:22:54	1822

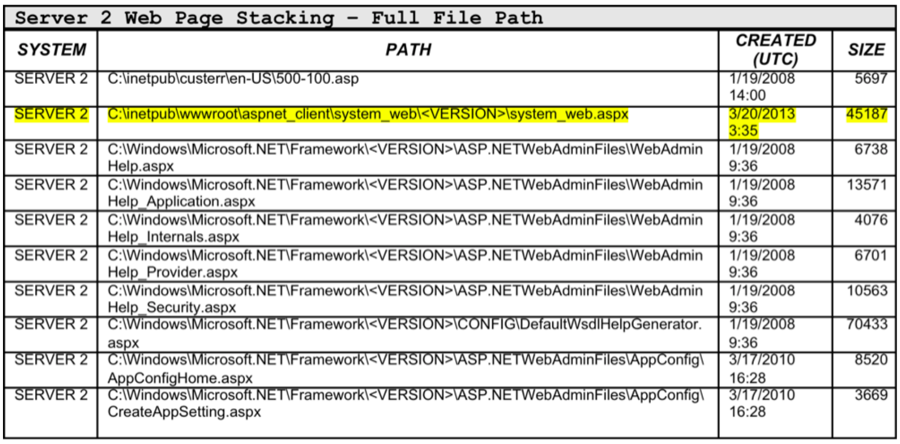
In Table 2, the web scripts on Servers 1 and 3 appear to be consistently residing in the same subdirectories. This type of activity suggests that these web scripts may have been part of a 3rd party installation or custom application. We can also reference the file creation times and see that these scripts were all consistently created around the same time as one another.

Looking again at Server 2, you can see our malicious Deep Panda web shell stand out:

- C:inetpubwwwrootaspnet_clientsystem_web<VERSION>system_web.aspx

System_web.aspx is the only file across all 200 systems that resides in that specific subdirectory.

We can also perform similar sanity checks on individual servers. If we look at all files with web script extensions on Server 2, there are a total of 616 files returned. Table 3 shows an abbreviated listing of these results.



Server 2 Web Page Stacking – Full File Path

SYSTEM	PATH	CREATED (UTC)	SIZE
SERVER 2	C:\inetpub\custerr\en-US\500-100.asp	1/19/2008 14:00	5697
SERVER 2	C:\inetpub\wwwroot\aspnet_client\system_web\<VERSION>\system_web.aspx	3/20/2013 3:35	45187
SERVER 2	C:\Windows\Microsoft.NET\Framework\<VERSION>\ASP.NETWebAdminFiles\WebAdminHelp.aspx	1/19/2008 9:36	6738
SERVER 2	C:\Windows\Microsoft.NET\Framework\<VERSION>\ASP.NETWebAdminFiles\WebAdminHelp_Application.aspx	1/19/2008 9:36	13571
SERVER 2	C:\Windows\Microsoft.NET\Framework\<VERSION>\ASP.NETWebAdminFiles\WebAdminHelp_Internals.aspx	1/19/2008 9:36	4076
SERVER 2	C:\Windows\Microsoft.NET\Framework\<VERSION>\ASP.NETWebAdminFiles\WebAdminHelp_Provider.aspx	1/19/2008 9:36	6701
SERVER 2	C:\Windows\Microsoft.NET\Framework\<VERSION>\ASP.NETWebAdminFiles\WebAdminHelp_Security.aspx	1/19/2008 9:36	10563
SERVER 2	C:\Windows\Microsoft.NET\Framework\<VERSION>\CONFIG\DefaultWsdlHelpGenerator.aspx	1/19/2008 9:36	70433
SERVER 2	C:\Windows\Microsoft.NET\Framework\<VERSION>\ASP.NETWebAdminFiles\AppConfig\AppConfigHome.aspx	3/17/2010 16:28	8520
SERVER 2	C:\Windows\Microsoft.NET\Framework\<VERSION>\ASP.NETWebAdminFiles\AppConfig\CreateAppSetting.aspx	3/17/2010 16:28	3669

Interestingly, System_web.aspx was the only web script located in C:inetpubwwwroot. This directory is publically accessible through the internet and it is extremely unlikely that an application would only have one web script to provide its functionality to an end user. If this location is legitimate, where are the other scripts? The fact that System_web.aspx was the only web script hosted publically by this web server is highly suspicious.

Summary

- Focus searching on only web servers in an environment

- Filter for extensions ASP, ASPX, JSP, PHP and CFM

- Limit searches to:

– %systemdrive%\Inetpub\wwwroot

– %programfiles%\Apache Software Foundation

– %systemdrive%\ColdFusion<version>

– Other 3rd party web hosting solutions

- Sort by file creation time

- Sort by full file path

In our next post, we will dissect web logs to help identify web shells in your environment. In the meantime, **register now** for the April 1st CrowdCast.

Tweet Share

Related Content

Compromise Attack Targets: Corporate Printers?

The term, “Advanced persistent threat” (APT), has become almost as mainstream as security breaches in everyday…

CrowdStrike Intelligence – Adversary-based Approach

Treating the problem, not the symptomsHaving spent the better part of the last 10 years dealing…

Is Your Cybersecurity Approach Based on Myth? Get a Reality Check.

As human beings, we make thousands of assumptions every day. It is our way of coping…

CROWDSTRIKE

| Privacy | Request Info | Blog | Contact Us | 1.888.512.8906