Mining for Malware

There's Gold in Them Thar Proxy Logs!

Author: Joe Griffin, jgriff9900@gmail.com

Adviser: Pedro Bueno

Accepted: 2008-04-15

Outline

1. Introduction

Some time ago, our management was given a tip by a three letter organization that shall remain nameless. The tip went something like this: Check your web proxy logs for sources of malware being introduced into your environment. We checked into it and as it turns out our source was not only correct about that tip, they were damn correct. This paper is about this journey and describes methods for identifying sources of malware and lowering the threat by taking action.

2. Background

Web Proxies Explained

Before getting into the details, a brief explanation of web proxies is in order. A Web proxy server is situated on the network between the clients and the Internet. When used in conjunction with caching and content filtering, proxies can provide a useful service to any sized organization.

Caching

When Bob from Accounting wants the latest news from CNN.com, his request goes to the proxy server. The proxy will check its cache and if there's an object there matching Bob's request and it has not expired, it will be served to Bob. If the object has expired or does not exist, the proxy fetches it from the Internet and serves it to Bob. This caching feature has a positive impact on network performance at the Internet gateway. However if there is an enormous volume of traffic through a given web proxy, the likelihood of a cached object being available becomes lower.

Content Filtering

Content Filtering allows controlling what sites clients are allowed to visit, often based on site categorization. Example categories include Auctions, Business, Computing/Internet, Job Searches, Pornography, etc. The focus and attention for filtering has been and probably always will be on preventing access to adult web sites. There are obvious reasons for this. However another category or area of concern to the security professional and business manager is sites that are known to serve malware, also known as malicious content. Adult web sites are notorious for serving malicious content as well, another reason aside from the legal issues, to prevent access to them.

One proxy vendor uses an example of how far you can go with filtering technology. If Bob from Accounting isn't satisfied with his job and uses a job search engine to look for employment elsewhere, he would get a pop-up window in his browser. It would read something to the effect of, “Bob, we noticed you are looking for jobs outside our fine Company X. If you wish to look for other jobs, we recommend you use our Company X job posting site located here: http://companyx.inc/jobs. Thanks and have a nice day!” A rather draconian approach, however that is possible if content filtering is enabled and proxy authentication is employed. This “Big Brother” approach would likely lower morale though, and employees might leave in droves.

Proxy servers can be configured to log information about requests and responses provided by the Internet web servers. Each log entry reveals information about the client making the request, date and time of the request, and the name of the object requested. Other parameters are also revealed including User-Agent, which will be addressed in more detail.

Forward vs. Reverse Proxies

A forward web proxy can be used to cache and filter content for internal users. A reverse web proxy can cache and filter content for external users. Say for example you have a web server and you only want to allow access to particular files on the server and you want to accelerate the speed at which pages are displayed to the external user, a reverse proxy could be used. The proxy hardware and operating system can be exactly the same in forward and reverse proxies provided by some vendors, only the proxy software is configured differently.

Surfing as a Malware Vector

If a web server is compromised, the web pages it serves can be altered to serve malicious code. For example, in February 2007 Super Bowl XLI was hosted at the Miami Dolphins Stadium. Days before Super Bowl XLI, the server hosting the Miami Dolphins Stadium web site became "owned". The compromise allowed a nefarious individual or group of individuals to add a single line of JavaScript code to the site home page. If a client browsed to the exploited site using an unpatched version of the Windows operating system, the client would connect to a remote server registered to a nameserver in China and download a Trojan keylogger/backdoor giving the attacker "full access to the compromised computer."[1] The OS vulnerabilities exploited were MS06-014 released April 2006, and MS07-004 released January 2007.

Methodology

Where We Were

A proxy log query script was already available used by another group for investigating cases involving acceptable use policy violations regarding Internet usage. If this script could be modified to look for say, .exe files, that seemed like a good place to start. Content filtering was already enabled with categories such as those mentioned earlier. The filtering vendor also provided a category of Malicious Sites which was already being blocked – perfect. It became obvious that the majority of the suspect malicious software being downloaded was from the category of “None”, or no category. Identifying .exe files categorized as “None” was chosen to target the “low-hanging fruit”.

After identifying the suspect malware via the proxy log query results, samples were manually obtained using the wget method. Samples were submitted to a site designed to help identify malware, VirusTotal.com. VirusTotal allows individuals to submit suspect files that are then scanned by 32 Anti-Virus (AV) engines produced by different vendors. If several vendors on VirusTotal detected a known virus and our AV vendor did not detect it, the sample would be submitted to our vendor requesting a virus signature (DAT) file to protect the enterprise. To further protect the enterprise, the site found to be hosting the malicious software would be submitted to the content filtering vendor to categorize the site as Malicious Content.

To demonstrate the VirusTotal submission process, sample file passxp.exe was obtained and submitted. See Table 1 for VirusTotal results. To help validate the results from

VirusTotal, the free browser add-on McAfee SiteAdvisor, can be useful. For example, the site hosting the sample for this demo was reported as follows in McAfee SiteAdvisor: “In our tests, we found downloads on this site that some people consider adware, spyware or other potentially unwanted programs.” There were 7 “red downloads” identified by SiteAdvisor for this site. These downloads were identified as Adware-BDSearch, BackDoor-BAC Trojan, Adware-CDNHelper, Generic Dropper Trojan, Generic Downloader.z Trojan, and Downloader-BAY Trojan. Each file had achieved a 7 or 8 out of 10 on the SiteAdvisor “Nuisance Meter” scale.

Table 1: Example of VirusTotal Sample Submission

Note: AV products with no detections are not shown here.

File passxp.exe

Result: 7/32 (21.88%)

Antivirus	Version	Last Update	Result
Avast	4.7.1098.0	2007.12.09	Win32:Adware-gen
ClamAV	0.91.2	2007.12.09	Adware.Cdn-5
Ikarus	T3.1.1.12	2007.12.09	Backdoor.Win32.Rbot.fo
Panda	9.0.0.4	2007.12.09	Suspicious file

Sophos	4.24.0	2007.12.09	Mal/EncPk-AX
Sunbelt	2.2.907.0	2007.12.07	VIPRE.Suspicious
Webwasher-Gateway	6.6.2	2007.12.08	Win32.Malware.gen!90 (suspicious)
Additional information□File size: 1812354 bytes□MD5: f68a4164e3e5005e871357eab9a08582□SHA1: df3c9c68bdb75ea80815aa990ededcb9275bcfac□PEiD: WinRAR 32-bit SFX Module□packers: UPX□packers: ASPack□packers: RAR, PecBundle, PECompact, UPX□packers: ASPack, ASPack, PE_Patch.PECompact, PecBundle, PECompact, UPX			

Where We Are Now

Automation has been developed to examine the proxy logs for suspect malicious code that has been downloaded by various hosts in the enterprise. Samples are obtained from the Internet source using the safe wget method described in the next section. A command line AV scan is performed on each .exe file which was downloaded from a site categorized as None. New detections are reported to the AV vendor via e-mail with an attached sample requesting a new AV signature DAT file. URLs found to be malicious are identified as such in a local database. Requests to change the content filtering category from None to Malicious

Sites are submitted to the filtering vendor automatically. The email text includes the complete URL showing the path to the .exe file for reference. Daily content filtering updates on the proxy servers ensure newly categorized sites are swiftly blocked.

Some automated scripts have been implemented that generate the following reports:

Regional Daily Top Malicious Site Access – This report identifies hosts reaching out to known malicious sites more than 2000 times per day. Each regional service desk receives a copy for their region and takes action to investigate and mitigate those hosts. This report has been well received and very successful. The threshold will be lowered to from 2000 to 1000 times per day as progress continues in identifying and mitigating hosts hitting known malicious sites.

Regional Top 15 Sites – This report identifies the most accessed web sites for each region. It has been useful in identifying misconfigured hosts found to be creating significant amounts of unneeded network traffic.

Regional Top 15 Source IP Addresses – This report shows the “top talkers” regarding web surfing.

Regional Top 15 AV Client Misconfigurations – This report shows hosts trying to download their latest AV signature DAT files from the Internet rather than a local repository.

This indicates a problem with either the host’s AV configuration or a network connectivity problem between the host and the internal AV DAT file repository. Action is taken by appropriate parties to remedy the misconfiguration.

Regional Top 15 Non-Standard AV User-Agents – This report reveals hosts configured with non-standard AV clients based on the User-Agent string in the proxy. For example, here is the User-Agent for a non-commercial AV product:

```
AVGINET-XPD%2075FREE%20AVI=269.17.1/1183%20BUILD=503%20LNG=US%20LIC=70FREE-TX-
L7Z2U-IB-P1-C01-SIJTY-QEN
```

Daily Top Malicious Sites – Based on internally developed threat model – still under development.

Daily Domain Risk – Based on internally developed threat model – still under development.

Proper wget Technique to Avoid Disaster

Your proxy log report tells you a client has downloaded notbogus_REALLY.exe from a nefarious looking site. I know what you are thinking: "I will do a wget to get a sample to work with!" That's good thinking, but be careful. Using wget with no user agent looks mighty suspicious to the nefarious group or individual who posted the malware. If they noticed this in their web logs and identified your IP Address space, you could become a victim of a Denial of

Service (Dos) Attack. Take for example the lengthy Distributed Denial of Service (DDoS) attacks by the Storm botnet against the Spamhaus Project and Surbl.org:

"Instead of pushing a huge stream of packets at their network to overwhelm their servers, the Storm botnet is flooding them with nonsensical URL requests. And this attack, which recently subsided, has been the longest attack they've ever had to repel -- lasting about two months."[2]

You would need to take measures to prevent this. First, use a user agent string in your wget command such as:

```
wget -U
"Mozilla/4.0%20(compatible;%20MSIE%206.0;%20Windows%20NT%205.1;%20SV1;%20.NET%20CLR
%201.1.4322;%20MAXTHON%202.0)" http://bad.biz/down/notbogus_REALLY.exe
```

This will cause the bad guy's web server log to log your request as if it were made from a web browser, cloaking your wget method. See the next section for more details on user agents.

Second, do not issue the wget command from IP Address space owned by your organization. Instead, purchase an outside shell account or install a 1U piece of hardware at a vendor providing co-location services. Search "1U colo" to find many vendors. Be sure to understand the vendor's Acceptable Use Policy about triggering DoS attacks against them. Although you are protecting your own organization against DoS attacks, triggering one against the vendor could create havoc for them and lawsuits for you

User-Agents

A User-Agent is actually an HTTP Request Header.   It provides information about the client software and is typically the name and version of the browser or other application making the request. It is used for server access statistic logging and also may be used to tailor how the server responds to the needs of different clients.[3]

One clever botnet herder named their User-Agent “007”, likely as homage to the infamous James Bond Agent 007 character. Be on the lookout for similar suspect “007” named User-Agents in your proxy logs. Here are some examples:

Your proxy logs show some unusual signs of client access to text files over TCP port 80 using a user agent of “Ms”. You may want to use the safe wget method described earlier to obtain a sample to investigate. If the destination URL is determined to be malicious, use your abilities to deny access to the site.

You discover a User-Agent named “bX” in your proxy logs, and as it turns out the traffic was denied at the proxies. The site is categorized as Malicious Sites. You may want to investigate the host generating the traffic with User-Agent “bX” that is hitting a known malicious site.

You discover a User-Agent named “dyalog.exe” that was seen downloading a text file called Now.txt from a site with just an IP address, no Fully Qualified Domain Name (FQDN). You try to get a sample of Now.txt, but the file is no longer available. You may want to investigate the host with the suspicious User-Agent “dyalog.exe”. A Google search of dyalog.exe determines it is probably related to some legitimate application, but being a good analyst you will need to validate that is indeed the case in this scenario.

Base64 Encoding Scenario - Smells Like Botnet

Your proxy logs show a high volume of traffic from a number of hosts to a particular Internet site and the URL string appears as follows:

```
        http://another.malicioussite.org/cgi-bin/

        dGM3N3ZtLXZ0SG9zdDE4JjAwLTBDLTIxLVoyLTEzLUsyJjE5Mi4xNjguMjAwLjEwMA==/viewuser
.cgi
```

The character string preceding the two equal sign characters looks like Base64, a method of encoding often used for obfuscating text[4]. You could copy paste the string including the equal signs into a Base64 decoder. A good one to use that offers a "Safe Decode As Text" feature is http://www.opinionatedgeek.com/dotnet/tools/Base64Decode. Doing so provides the following translated text:

```
        tc77vm-vtHost18&00-0C-21-Z2-13-K2&192.168.200.100
```

In this case, the translation is 3 fields separated by the ampersand "&" character:

Hostname, MAC Address, and IP Address. It's possible the host is infected with malware, and it is "phoning home" to report information about itself and/or it’s surrounding environment to a Botnet Command-and-Control Server (C&C). Another common character for separating fields in Base64 obfuscated URLs is the colon ":" character. It is not a requirement to have equal sign "=" characters in a Base64 string. The "=" characters are sometimes used as null padding to get the string to a number of characters divisible by 4, a requirement for Base64.

The hosts hitting the malicious site should be examined for malicious software, and samples of the suspect malware should be submitted to VirusTotal.org to look for detections. A good rule of thumb might be if more than 5 AV vendors detect the sample as malware, the sample should be submitted to your local AV vendor for a new AV signature file. The new signature file should be applied to the infected hosts and a full virus scan should be performed on the host. After validating the malicious software, now would be a good time to put controls in place preventing access to some.malicioussite.org for those "rogue" hosts on your network that could be infected but don't have AV installed or the latest AV signature files installed.

An Example of a Legitimate URL Containing Base64 Found in Proxy Logs

You may be perusing your proxy logs and discover what appears to be suspicious Base64 code. Take the following URL:

http://a.suspiciouslooking.com/jsapi?current_url=aHR0cDovL29ubGluZS53c2ouY29tL2FydGljbGUvU0IxMjAyMzM1Nzk5NjExNDg3NzEuaHRtbD9tb2Q9ZGplbWFsZXJ0TkVXUw==&action_ts=1202350222342&arrival_ts=1202350150548&methodName=addClickstream&publisher_key=6563391702&guid=SB120233579961148771

You decode the apparent Base64 string:

aHR0cDovL29ubGluZS53c2ouY29tL2FydGljbGUvU0IxMjAyMzM1Nzk5NjExNDg3NzEuaHRtbD9tb2Q9ZGplbWFsZXJ0TkVXUw==

And get:

http://online.wsj.com/article/SB120233579961148771.html?mod=djemalertNEWS

This is a legitimate news site - there is no obfuscation or nefarious activity going on here.

AdHoc Proxy Log Queries

After you get comfortable looking at your proxy logs, you may start to notice traffic to web servers hosting .txt files. The site names might look suspicious, like 1.looks-like-google-but-isnt.cn. The logs may reveal a client in the Asia Pacific region that downloaded a few different .txt files from the same Internet host on December 7th. When retrieving your samples to investigate, you may notice only one of the .txt files remains available. This .txt file contains one line: iesuper.exe,1;. That's a good clue to look for downloads of iesuper.exe. A quick Google search of iesuper.exe shows someone else has encountered a file with the same name and it was found to be malicious. Here's where an AdHoc query script can come in handy.

The AdHoc query script reads input from a search_terms.txt file and a date file and sends results to a file that can be massaged for analysis. The script relies on the log file naming convention for the dates and region. Here's the syntax for how easy it is to run the query:

```
$ echo 1207 > date_file                [which date to query]
$ echo iesuper\.exe > search_terms.txt [what to query, escaping the "."]
$ adhoc apac                           [which region to query]
Cleaning up previous search results...
Using output file: results.out (processing...)
Finished searching /logs/1207/apac01_1207.log.gz.
Finished searching /logs/1207/apac02_1207.log.gz.
0 matches found.
```

I can now say with confidence no downloads or download attempts of iesuper.exe occurred on December 7th from hosts in the Asia Pacific region. Using this standardized method ensures analysts produce accurate and repeatable results when investigating incidents and threat intelligence. This methodology may be expanded to mail log queries as well for investigating phishing and other incidents.

Script for Identifying .exe Files in Proxy Logs

The zgrep command allows searching compressed log files. You could use zgrep to look for all .exe files, but that would return many legitimate non-malicious files. Let's narrow it down to search for .exe downloads from sites categorized as "None", or no category. The URL string, including the filename, is configured on the proxies in this case to be the last field in each proxy log record, or row. The Blue Coat proxy log format includes a "Windows-like" carriage return at the end of each line. To account for this unusual carriage return when using zgrep, a ^M string is required, produced in Unix by typing ctrl V then ctrl M, while continuously holding down the ctrl key:

```
zgrep -i "get\ none.*\.exe^M" /proxylogs/0317/*
```

Note the "\" character is used to escape special characters including the space between get and none, along with the "." before the exe. The get\ none part of the string looks for the GET statement followed by the category NONE. The -i flag tells the zgrep command to ignore case.

Left alone in this raw format, the above command will return one row for each download that ends in .exe. Here's one row as an example:

```
1205745626.642 192.168.1.100 151 10.1.1.1 80 TCP_MISS/200 -
ip.somemalicioussite.com HTTP/1.1 24962 235 DIRECT/ip.somemalicioussite.com
```

application/octet-stream 10.1.1.1 - ICAP_NOT_SCANNED GET none OBSERVED Mozilla/4.0%20(compatible;%20MSIE%206.0;%20Windows%20NT%205.1;%20SV1;%20.NET%20CLR%201.1.4322;%20InfoPath.1) http://ip.somemalicioussite.com/1.exe

This time let's just grab the source IP and suspect malicious site including the URL. To do this, we can pipe to awk, printing out only the fields of interest which are the 2nd and 21st:

```
zgrep -i "get\ none.*\.exe^M" /proxylogs/0317/* | awk '{print $2;print $21}'
```

The same example result shown above is now reduced to:

192.168.1.100 http://ip.somemalicioussite.com/1.exe

The same method can be used for identifying suspicious text files, DLLs, PDFs, etc. Just replace .exe with .txt, .dll, or .pdf in the zgrep command.

Regarding the 1205745626.642 value from the log example, that is the Linux system time timestamp. The units for Linux system time are seconds elapsed since 00:00:00 January 1, 1970[5]. This value can be converted to human-readable date and time format. This example shows how to convert it if the logs are imported into Excel.

1. Make sure Unix timestamp is in column A.

2. Insert new column after column A.

3. Paste the following into the cells in column B:

=A1/86400+25569

4. Rt-click column B and change the format to Date, and the Type to 3/14/01 1:30 PM.

In this case the result is 3/16/08 11:20 PM.

Where We Are Going

- Continue refining automated reports.
- Automated comparison to multiple AV products.
- Automatic submission to behavioral analysis system.
- Scoring system for code analysis.

3. Recommendations

These are not given in any particular order. Using a layered approach, combining efforts will be of great benefit.

Reduce the number of Internet gateways at your organization. If there are 100 offices and each office has an Internet gateway with a unique architecture, you'll have malware a-plenty unless you lower the number and standardize the architecture. Recall the statement made earlier about saying with confidence a particular file was not downloaded. That is only true for hosts that are using a managed Internet gateway.

Rather than using the “allow all, deny known bad sites” approach to Internet surfing, adopt a “deny all, allow only needed sites” approach. If you have a group of users who need access to 10 web sites to do their work, allow only those 10 sites to that group. This act would be the most effective method for addressing the malware via surfing threat. If you need to throw in another 10 or so sites to allow for some work-life balance, allow them too, in a controlled fashion. This method would require some overhead to manage, however it would nearly eliminate the surfing malware vector and lower the amount of resources necessary for stamping out routine infestations. If a single user can cause your company great harm by visiting a web site and installing malicious software, while the multi-million dollar firewall implementation allows it to happen because TCP port 80 is allowed to almost any destination, it doesn’t make much sense.

Require laptop users to always use your Internet proxies whether they are on your network or not. This is entirely possible and would eliminate much malware introduced into business environments. Again, recall the earlier statement about saying with confidence a particular file was not downloaded. Laptop users do not apply to this statement if they are off the network and attaching to the Internet outside the control of your Internet proxies.

Use standardized web proxies with content filtering. One solution will be much easier to manage than 2 or 200.

Update filtering databases on proxies daily. In the current dynamic threat environment, anything less frequent than daily updates could expose clients to more malicious sites than necessary.

Consider surfing quotas and not allowing after hours web surfing. The less unnecessary surfing going on, the less likely malicious content will be introduced. If that workstation in Shipping is not required after 5 PM or on the weekends, don't allow surfing from it during those times. There are 3rd party solutions that can be employed for these preventative measures. If you implement surfing quotas, be prepared for users to appear with torches and pitchforks. This is where working within a secure badge access area can come in handy. On the other hand, productivity might improve, incidents would decline, and you might get an "attaboy" from someone not carrying a torch or pitchfork.

Ensure all machines on your network comply with your standards. If a shipping company wants to install their workstation on your network, don't allow it until it complies with the standards you've worked so hard to implement.

Use content filtering to prevent web-based email to sites like Yahoo and Hotmail. If you are going to allow web-based email, use security awareness to persuade users toward Gmail, as Gmail does not allow .exe file attachments, zipped or otherwise.

Use forensic tools for obtaining malware samples from your hosts. If you try to obtain a sample from a site identified in your proxy logs and the sample is no longer available, you may be able to get a sample from the workstation that downloaded the file. The forensic tool should allow getting the sample even if it has been deleted, unless of course that part of the media was overwritten by another file. EnCase Enterprise works well for this, although it's a pricey solution. EnCase allows sorting files by file type and date/time created. Correlating file create time with proxy log time stamps can be helpful in locating suspect downloaded malicious files on the host.

Correlate proxy logs with firewall logs. You may be surprised to see the traffic generated from some hosts that is being denied by the Internet firewall.

Use caution when using Windows systems while working with malware samples outside of a controlled lab environment.

Use a sandbox environment for testing malware samples. CWSandbox is a good tool that shows network connections and much more detail for digging under the hood.

Implement the Firefox web browser with the NoScript add-on to avoid compromises to your clients like those affected by the Dolphins Stadium site compromise. There might be some areas in your organization where this will go over better than others, as there is a bit of a learning curve in understanding how to use NoScript. By default, Internet Explorer and Firefox will run JavaScript. With NoScript, by default scripts will not run unless you allow them on a per site basis. Some users might grow tired of having to allow scripts on a per site basis and easily tweak NoScript to “Allow scripts globally” with a single click, so keep that in mind.

Use Windows Group Policy Objects (GPOs) to remove the ability to log into workstations with Administrator permissions. This would prevent installing .exe files in many cases.

Consider migrating to a non-Windows platform. In our findings, most of the malicious software posted for download from the Internet is applicable only to Windows systems. One analogy for this case would be about the U.S. bank robber, “Slick Willie” Sutton. When Sutton was asked why he robbed banks, he said simply, “That’s where the money is.”[6] So if you asked an attacker why they target Windows systems, they might say the same thing. If non-Windows systems were in the majority of households and companies, attackers would likely be targeting the non-Windows systems more.

Regardless of OS, ensure well-managed patching and AV systems are in place. A minimum frequency of monthly OS patches and daily AV DAT file updates should be employed. For Windows systems, LANDesk is a good tool for patch management of both OS level and application level patches.

Don’t adopt the Ron Popeil “Set it -- and forget it!” approach to AV and patching tools.

Audit sample sets of hosts to ensure they are really getting patched and really getting the latest AV DAT files.   Some AV products have reporting available that can notify administrators if a particular host stops "reporting in" to check for and download the latest signature files from an internal server. Use those reports and parse vetted results to the party that will visit the hosts to ensure they get the latest signatures. A good patching product should provide the same types of reports.

Responding to an incident is not the best time to find out a host is completely unpatched and has no AV installed, especially if patches or AV would have prevented the incident. Seek out and standardize hosts on your network that are missing your standard set of security tools.

4. Conclusions

If you are one of those organizations that says "We don't have security incidents", your organization is probably not looking at the web proxy logs. Having a firewall properly configured does offer protection, but not from your users surfing the Internet. Content filtering, AV protection, OS level and application level patching, bundled with good security awareness can help prevent incidents at your organization. Going the extra mile by proactively mining for malware and acting on the findings can not only prevent incidents at your organization, but throughout the entire network community.

5. References

1. Naraine, Ryan (2007) "Super Bowl stadium site hacked, seeded with exploits", ZDNet
2. Gaudin, Sharon (2007) "Storm Worm Botnet Attacks Anti-Spam Firms", Information Week
3. Kozierok, Charles (2005) “The TCP/IP Guide”, Chapter 82 – HTTP Message Headers
4. Spammer X (2004) “Inside the SPAM Cartel: Trade Secrets From the Dark Side”, Chapter 7 – Spam Filters: Detection and Evasion
5. Chirico, Mike (2004) “CLI Magic: It’s about time”, Linux.com
6. Duffy, Peter (2002) “City Lore; Willie Sutton, Urbane Scoundrel”, New York Times