



As the complexity of malicious software continues to evolve, knowing how to analyze malware is becoming increasingly important. However, sometimes we may not have the time or the necessity to perform an in-depth examination of malicious code. In this presentation I discuss several techniques and free tools that offer shortcuts for malware analysis. They speed up the process of establishing key characteristics of executable files and malicious websites.



Why even concern ourselves with malicious code? Malware is being increasingly used as a weapon in cyber-world. Here are just a few stories, carried by general media outlets, that I came across in the recent weeks:

**CNN: Gangs flooding the Web for prey.** This story covered tactics of using denial-of-service attacks for fraudulent purposes. At the heart of such attacks are bot networks—a popular form of malicious code.  
(<http://www.cnn.com/2006/TECH/internet/12/20/cybercrime>)

**The New York Times: Attack of the zombie computers is a growing threat.** This was yet another story covering the dangers of bot networks.  
(<http://www.nytimes.com/2007/01/07/technology/07net.html>)

**MSNBC: Don't get hooked this phishing season.** This story discussed the prevalence of phishing scams, which are typically driven by malicious software taking the form of phishing toolkits. (<http://www.msnbc.msn.com/id/16212289>)


**eWeek: Hackers selling Vista zero-day exploit.** This story reported the existence of an exploit that targeted an unpatched vulnerability in Windows Vista. Exploits, of course, are another popular form of malware.  
(<http://www.eweek.com/article2/0,1895,2073611,00.asp>)



As defenders of information systems, we're tasked with protecting IT resources against threats that often involve malware. To excel at this job, it helps to be able to learn, at least at a basic level, about the software that threatens our systems.


Consider a malicious executable you encounter on a desktop computer at your office. How do you determine its purpose, or assess where it may have come from or what its goals might be? Relying solely on your anti-virus engine is often insufficient. Anti-virus software may be unable to recognize the specimen as malicious. Sometimes, the anti-virus vendor's information about the malware specimen may be outdated or not applicable to the variant you discovered.

If you don't want to rely on someone else having produced the information about the malicious program, then you need to learn how to obtain that information yourself. That's why knowing how to analyze malware can be useful.



## In-depth analysis is not always necessary or practical.

Let's explore shortcuts to speed up analysis.



**SANS**  
INSTITUTE

© 2007 Lenny Zeltser

In a course that I teach at SANS Institute, called Reverse-Engineering Malware, we spend several days learning how to examine malicious programs. We examine their behavioral patterns and look at their code. Performing comprehensive analysis is often challenging, and is usually time-consuming. However, depending on your objectives, such in-depth analysis is not always necessary.

In this presentation, we take a look at a few shortcuts we can take to save ourselves time. These shortcuts are particularly useful in situations where we do not need to perform comprehensive analysis of the malicious program, but we'd still like to get a general sense about its functionality.



**Examine web-based malware components.**



© 2007 Lenny Zeltser

One of the most popular attack vectors is an attempt to exploit a vulnerability in client-side software, such as a web browser, to run the attacker's code on the victim's system. That's why I'd like to begin the discussion by looking at techniques that can help us examine web-based malware components.



**Let's use a real-world example for illustrative purposes.**

A website, possibly compromised, was distributing a suspicious executable via the following URL:

`http://www.pupinini.com.br/aspnet_client/static.htm?Imagen=145879652148962`



© 2007 Lenny Zeltser

Rather than simply showing you tools, I'd like to discuss them in the context of a real-world example. The particular incident I'd like to use is associated with a website that was distributing a malicious executable via the following URL:

`http://www.pupinini.com.br/aspnet_client/static.htm?Imagen=145879652148962`

(As of this writing, the web site no longer distributes the malicious executable at this URL.)

How was the website able to install the malicious program on the victim's computer? What were the program's capabilities? That's what I'd like to uncover by showing you the following tools and techniques.

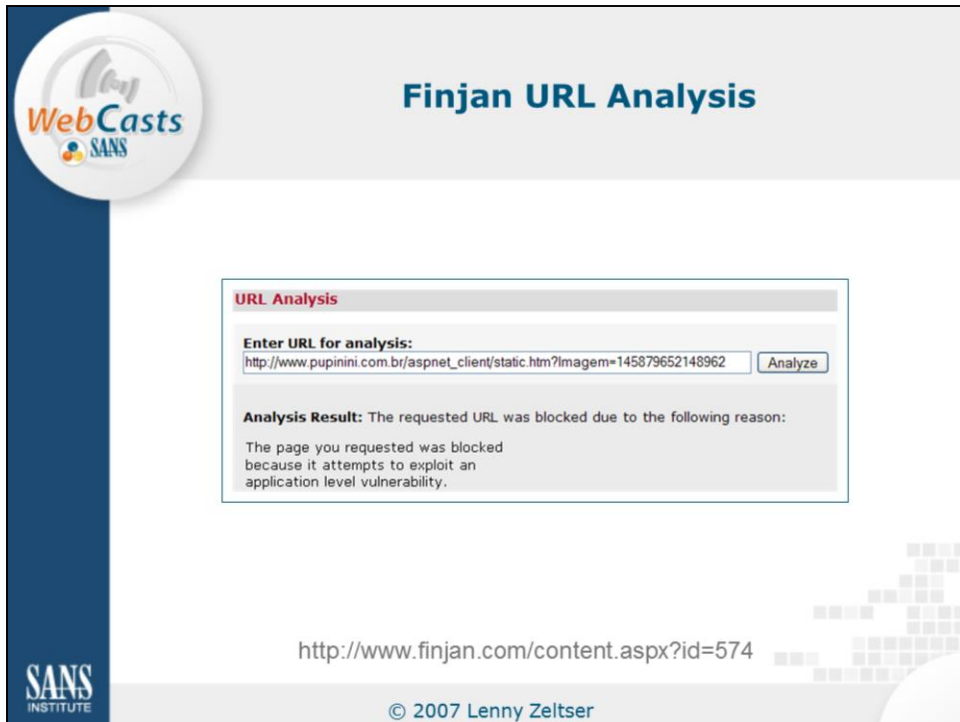


# Assess the posture of the suspicious web page with link scanners.



© 2007 Lenny Zeltser

A good place to start the analysis of the potential web-based malware incident is to assess the posture of the suspicious web page. What dangers could lie there should an unsuspecting user connect to it? There are several free sites that will assess the web page's threats for you. All you need to do is to supply the link to the suspicious site.

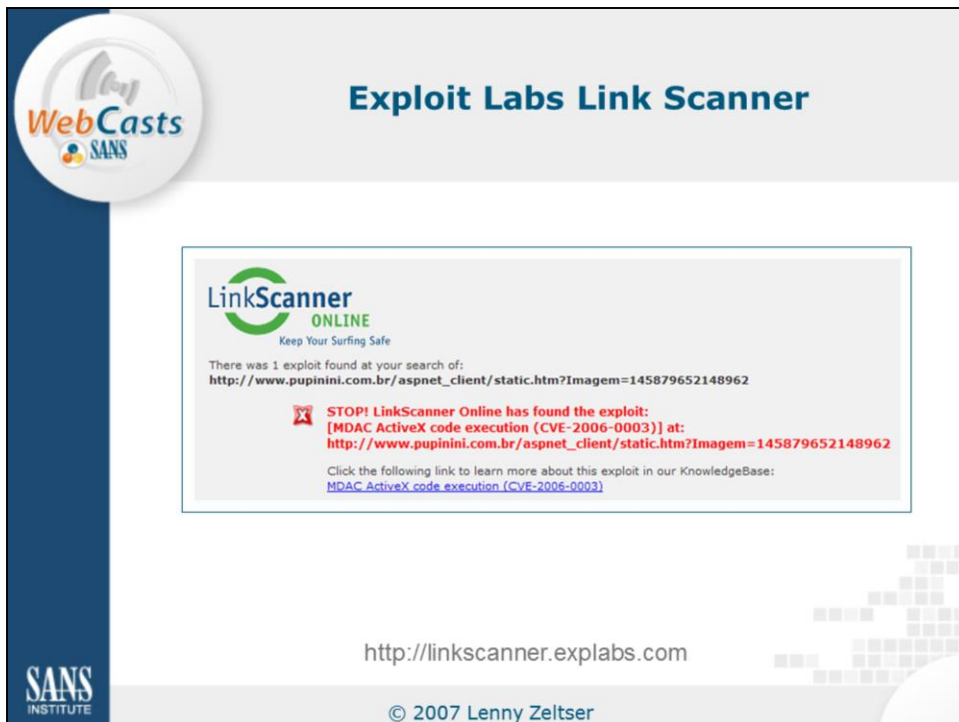


One of the free link scanners you should consider is Finjan URL Analysis, available freely at the following site:

<http://www.finjan.com/content.aspx?id=574>

A link scanner, such as Finjan URL Analysis, connects to the URL you supplied and scans it to locate malicious browser scripts, exploits, or other anomalies. It will not detect the presence of all malware, of course, but if it flags a page as malicious, there is a good chance it is right.





Exploit Labs Link Scanner is another URL scanner that is available for free and is worth your attention. It is available at:

<http://linkscanner.explabs.com>

My favorite aspect of this on-line tool is that if it detects malware on the scanned page, it is often able to tell you which particular exploit is being used on the malicious site.

In this case, Exploit Labs Link Scanner pointed out that the page I directed it to attempts to launch an MDAC ActiveX exploit (CVE-2006-0003) against the visitor's browser.

Both Finjan URL Analysis and Exploit Labs Link Scanner examine the site in real time, which makes them particularly convenient for malware analysis that involves a suspicious website.




After confirming that the suspicious web page is actually malicious, you may want to retrieve it, so that you have a local copy to examine at your leisure. If you decide to retrieve the page, you must exercise caution. Here are a couple of tools that can be of help.

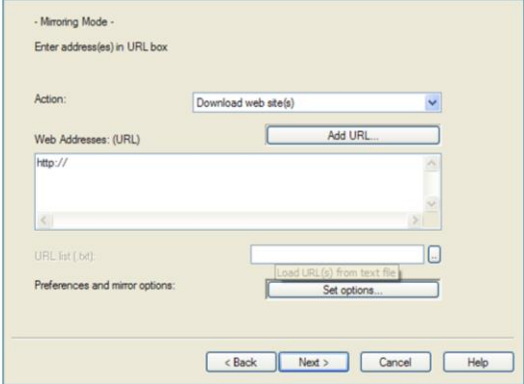
One of the reasons I think it's a good idea to retrieve a malicious web page as soon as you can is that the attacker may take it down if he or she notices someone "poking around." By having the page locally, you will have a copy of the page even if the attacker takes the live version of the page off line.

Note: Retrieving the malicious page involves connecting to a dangerous and unpredictable website. Therefore, I recommend doing so from within a sandbox, such as VMware or Sandboxie. We will discuss such tools a few slides later.





## Sometimes it's useful to mirror the whole site with HTTrack Copier.



<http://www.httrack.com>

© 2007 Lenny Zeltser

If you encounter a site with many potentially malicious pages or files, you may want to mirror the whole site. You can do this with `wget` via the “`--recursive`” command-line parameter.

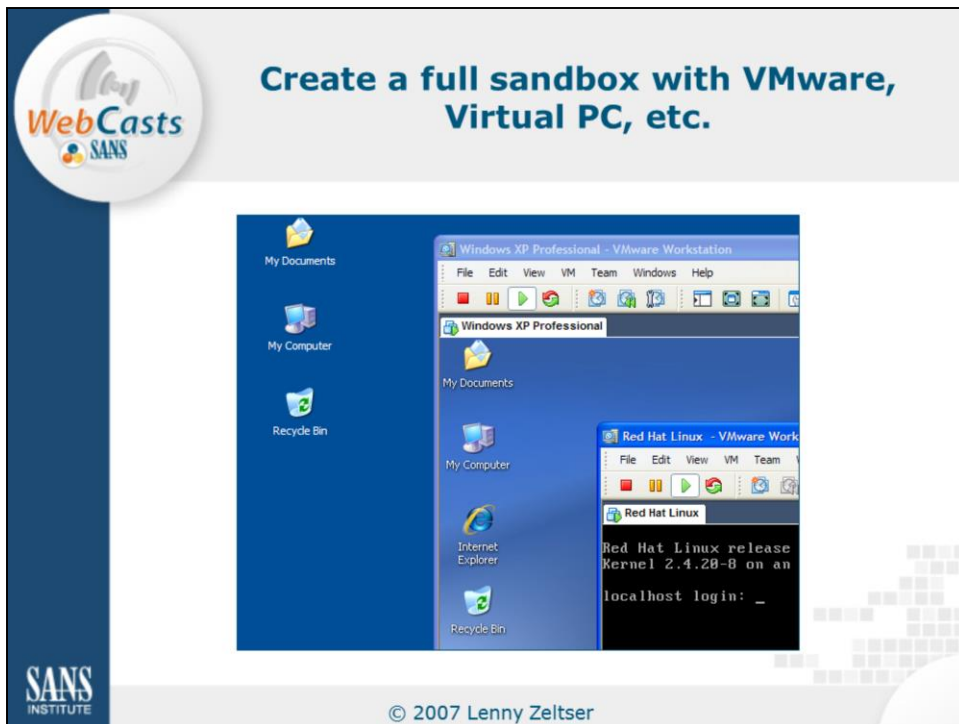
Alternatively, you may elect to use a more full-featured site mirroring tool called HTTrack Website Copier. This program offers a lot of options regarding with remote pages should be copied and how. It is available for free from:

<http://www.httrack.com>



Once you've copied the malicious pages to your local system, you can examine them to understand the threat. However, sometimes you will not have all the components of the attack that you need to sufficiently analyze it: there may be server-side scripts or other dependencies that you did not copy to your local system.

Therefore, it is often useful to interact with the malicious website when examining its functionality. This can be dangerous, since you'd be interacting with live malware, so you need to take extra precautions.




The most cautious way of interacting with malicious sites is probably to use a dedicated physical system that you'd wipe away after you've completed your analysis. That is rarely convenient, though. Most analysts prefer to use a more flexible mechanism provided by virtual systems, such as those you can set up with VMware or Virtual PC. Such virtual systems run at the same time as your main physical system, while providing a reasonably reliable sandbox. They're easy to store, because they have no physical manifestations, and they are very easy to restore to the previous state with a click of a button.

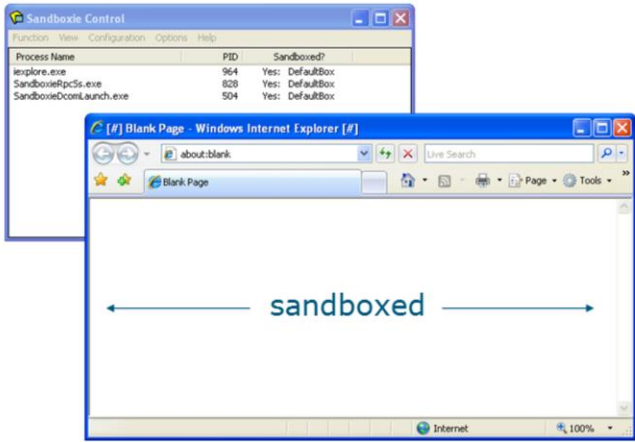
The screenshot on this slide shows VMware Workstation in action. In the background you see the desktop of my main physical system. Another window shows Windows XP Professional running as a virtual system. Also, a third window demonstrates a version of a Linux-based operating system running at the same time.

VMware Workstation is a commercial tool, although a free 30-day trial is available. The company also offers free versions of its virtualization products, in the form of VMware Player and VMware Server at the following site:


<http://www.vmware.com>



## Create a lightweight sandbox with Sandboxie.



<http://www.sandboxie.com>



© 2007 Lenny Zeltser

Fully virtualized systems, such as those you can build with VMware are nice, because all aspects of the machine are enclosed within the virtualized environment. This comprehensive nature of a virtual system often comes at the cost of performance—you need a reasonably powerful physical machine with lots of RAM for a product such as VMware.

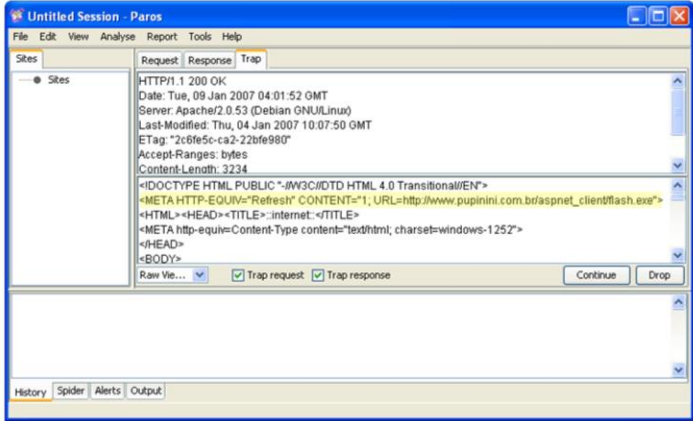
A more lightweight sandboxing option is available via a free tool called Sandboxie, which is available at:

<http://www.sandboxie.com>

Sandboxie lets you create an isolating sandbox around almost any program running on your computer, without having to create and run a full-blown virtual machine. On the screenshot on this slide, you see Sandboxie enclosing several processes associated with an instance of Internet Explorer. Internet Explorer is running within the sandbox. A visual clue to this is the presence of the “[#]” mark in the title of the Internet Explorer window. Now, we can quickly reverse any changes a malicious site may attempt making to our system via the web browser.



## Examine and modify the attack flow with Paros Proxy.



<http://www.parosproxy.org>  
 © 2007 Lenny Zeltser

Once the browser is running within a sandbox, you can begin interacting with the malicious website. This is where things may get tricky, because one of the biggest challenges to examining a web-based attack is understanding its flow: which scripts call which programs, and what data or code gets transferred as a result? To reverse-engineer the flow of an attack like this, a tool such as Paros Proxy comes in very handy.

Paros Proxy is a free application you can run on your system to intercept and even modify the requests your browser makes and the responses the browser receives from the malicious website. In this example, the malicious page is attempting to use a META tag to force the browser to download a suspicious executable called flash.exe. Paros Proxy allows me to observe this attempt and to remove the corresponding line from the HTTP stream, so that the browser on my system doesn't need to deal with this attempt. As you can see, this tool gives you the flexibility to modify the flow of the attack, so you can focus your examination on what is important to you.

You can download Paros Proxy from:

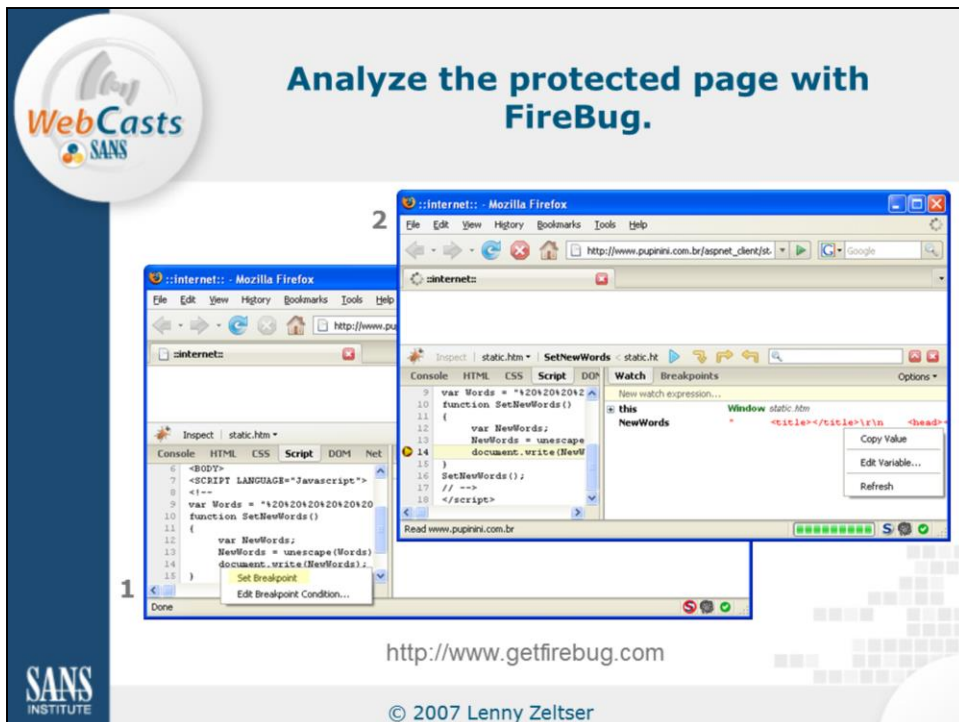
<http://www.parosproxy.org>





Attackers typically want to make the job of analyzing their malicious code difficult. As a result, many malicious web pages or the associated scripts are obfuscated. Fortunately, there are several mechanisms you can employ to strip away such protective measures.







FireBug is a free JavaScript debugger, which allows you to analyze many aspects of the malicious web page. FireBug operates as an extension for Firefox, and is available at:



<http://www.getfirebug.com>

My goal in this step of the analysis is to understand the contents of the Words variable, which seem to contain an obfuscated malicious script. What if the HTML Decoding Tools website could not decipher it? As you can see in the screenshot on this slide, I set a FireBug breakpoint on the line of the script that makes use of the variable, presumably after deciphering it. After the malicious script runs, I used FireBug to look and the contents of the variable via the Copy Value menu. The variable NewWords contained a fully deciphered version of the malicious script! The power of this approach is that it allows us to decipher scripts even if we do not know or understand the obfuscation algorithm.

I use FireBug in conjunction with another Firebox extension called NoScript, which prevents all scripts from running until I've set up the appropriate FireBug breakpoints.




**Identify general properties of the executable.**



© 2007 Lenny Zeltser

If you're analyzing an incident that involves a web-based infection vector, by this point in the analysis you may have a good idea of the vulnerability that the malicious site is attempting to exploit. You will also probably have a copy of a malicious executable that the remote site is attempting to install on the victim's computer. How can you quickly assess the general nature of the executable to understand what it's capable of doing?

The easiest manner of determining general characteristics of a malicious executable involves having someone else perform parts of the analysis for us. One way to accomplish that is to rely on the research already performed by anti-virus companies, who typically publish information about malware specimens on their websites. To make use of the information, you first need to identify the malware specimen in your possession. Let's see how to do that quickly...



## Use multiple anti-virus scanner: Virus Total.

Antivirus	Version	Update	Result
AntiVir	7.3.0.21	01.08.2007	TR/Delphi.Downloader.Gen
Authentium	4.93.8	12.30.2006	no virus found
Avast	4.7.892.0	01.08.2007	no virus found
AVG	386	01.08.2007	no virus found
BitDefender	7.2	01.09.2007	BehavesLike:Trojan.Downloader
CAT-QuickHeal	9.00	01.08.2007	no virus found
ClamAV	devel-20060426	01.09.2007	no virus found
DrWeb	4.33	01.08.2007	no virus found
eSafe	7.0.14.0	01.08.2007	no virus found
eTrust-InoculateIT	23.73.109	01.09.2007	no virus found
eTrust-Vet	30.3.3311	01.08.2007	no virus found
Ewido	4.0	01.08.2007	no virus found
Fortinet	2.82.0.0	01.09.2007	suspicious
F-Prot	3.16f	01.08.2007	no virus found
F-Prot4	4.2.1.29	01.09.2007	no virus found
Ikarus	T3.1.0.27	01.08.2007	no virus found
Kaspersky	4.0.2.24	01.09.2007	Trojan-Downloader.Win32.Small.eet
McAfee	4934	01.08.2007	no virus found
Microsoft	1.1904	01.09.2007	no virus found
MON32v2	1963	01.08.2007	probably a variant of Win32/TrojanDownloader.Banload.JM
Norman	5.80.02	12.31.2007	W32/Downloader
Panda	9.0.0.4	01.08.2007	Suspicious file
Prevx1	V2	01.09.2007	Malicious
Sophos	4.13.0	01.05.2007	no virus found
Sunbelt	2.2.907.0	01.05.2007	no virus found
TheHacker	6.0.3.146	01.08.2007	no virus found
UNA	1.83	01.06.2007	no virus found
VBA32	3.11.2	01.08.2007	no virus found
VirusBuster	4.3.19.9	01.08.2007	no virus found


<http://www.virustotal.com>

© 2007 Lenny Zeltser

If you rely on a single anti-virus engine to identify the malware specimen, you may receive inaccurate results or may not farther your quest at all if the anti-virus engine fails to detect anything malicious about the executable. Therefore, it's better to scan the specimen using multiple anti-virus engines. The VirusTotal website allows us to accomplish this without any fees via:

<http://www.virustotal.com>


After you upload the malicious specimen to VirusTotal, you will receive the results of the scan in almost real time. In my example, you can see that many anti-virus engines that VirusTotal uses failed to identify the executable. However, some labeled it with names that suggested it was a downloader of some kind. I could probably obtain additional information about the executable if I searched the corresponding anti-virus websites.



## Use multiple anti-virus scanner: Jotti's Malware Scan.

Scanner results	
Scan taken on 09 Jan 2007 04:29:45 (GMT)	
AntiVir	Found <b>TR/Delphi.Downloader.Gen</b>
ArcaVir	Found nothing
Avast	Found nothing
AVG Antivirus	Found nothing
BitDefender	Found <b>BehavesLike:Trojan.Downloader</b> (probable variant)
ClamAV	Found nothing
Dr.Web	Found nothing
F-Prot Antivirus	Found nothing
F-Secure Anti-Virus	Found <b>Trojan-Downloader.Win32.Small.eet</b>
Fortinet	Found nothing
Kaspersky Anti-Virus	Found <b>Trojan-Downloader.Win32.Small.eet</b>
NOD32	Found <b>probably a variant of Win32/TrojanDownloader.Banload.JM</b> (probable variant)
Norman Virus Control	Found <b>W32/DLoader.BOZX</b>
VirusBuster	Found nothing
VBA32	Found nothing


<http://virusscan.jotti.org>


© 2007 Lenny Zeltser

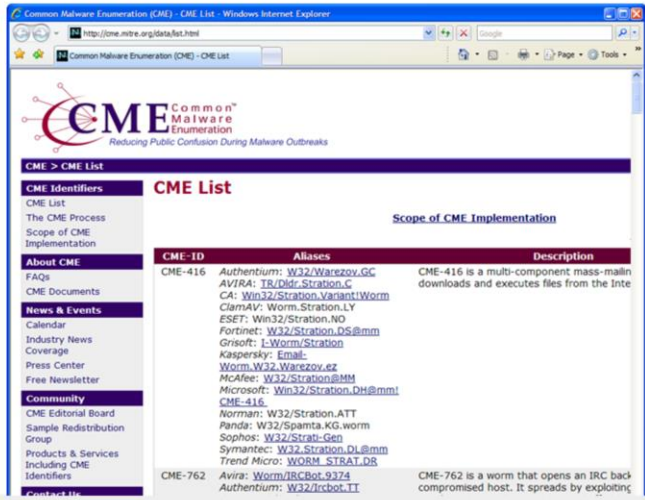
Another website that can scan a file using multiple anti-virus engines is Jotti's Malware Scan, available for free at:

<http://virusscan.jotti.org>

I like to use Jotti's Malware Scan when VirusTotal is not available for some reason, or just to get a second opinion on the malicious executable. As you can see in this example, the results of Jotti's Malware Scan closely match those of VirusTotal, even though Jotti's Malware Scan doesn't have quite as many anti-virus engines as VirusTotal.



## Make use of CME when possible.





CME-ID	Aliases	Description
CME-416	Authentium: W32/Warezov.GC AVIRA: TR/Dldr.Straton.G CA: W32/Straton.Variant(Worm) ClamAV: Worm.Straton.LY ESET: Win32/Straton.NO Fortinet: W32/Straton.DS@mm Gmsoft: L-Worm/Straton Kaspersky: Email: Worm.W32.Warezov.ez McAfee: W32/Straton@MM Microsoft: Win32/Straton.DH@mm! CME-416 Norman: W32/Straton.ATT Panda: W32/Spamta.XG.worm Sophos: W32/Strati-Gen Symantec: W32/Straton.DL@mm Trend Micro: WORM_STRAT.D8	CME-416 is a multi-component mass-mailer that downloads and executes files from the Internet.
CME-762	Avira: Worm/IRCbot-9374 Authentium: W32/Ircbot.TT	CME-762 is a worm that opens an IRC backdoor on a compromised host. It spreads by exploiting...

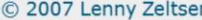

© 2007 Lenny Zeltser

Anti-virus vendors' websites often have a wealth of information that can significantly speed up your analysis. The most challenging aspect of using these sites is correlating the names of malware specimens. One site may refer to the specimen as ABC, while another could call it XYZ.

That's where Common Malware Enumeration (CME) comes in handy. CME is an effort to assign common names to malware specimens. Although not many specimens have been assigned CME identifiers, if the one you're analyzing has a CME identifier, you can search for it on anti-virus vendors' websites to quickly find the information you need.




**Assess key behavioral characteristics of the executable in a full sandbox.**



The information you may find on the web about the malware specimen you're analyzing is not always sufficient. Sometimes you need to perform some hands-on exploration in your lab. Behavioral analysis of this nature typically involves running the malware specimen in an isolated laboratory environment where you can observe how the malicious executable interacts with the local system and with network resources. In many cases, the laboratory environment is virtualized using tool such as VMware.

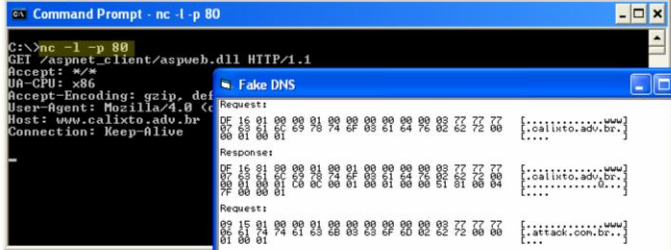
While a full-featured analysis would incorporate several steps, in this section I want to introduce you to a few useful tools and shortcuts.



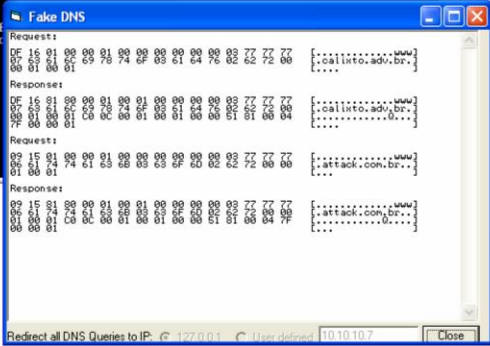


## Intercept DNS requests with fakeDNS.


1



2



<http://labs.idefense.com/software/malcode.php>




© 2007 Lenny Zeltser

Modern malware specimens often attempt to establish outbound network connections to get in touch with their author or operator. In an isolated laboratory environment it is useful to be able to redirect such connections to one of our laboratory systems, so we can determine what the specimen would do, had it been able to connect to the real system in the wild. A very convenient tool for redirecting such connections is fakeDNS, which is distributed as part of a free toolkit called Malcode Analysis Pack from:

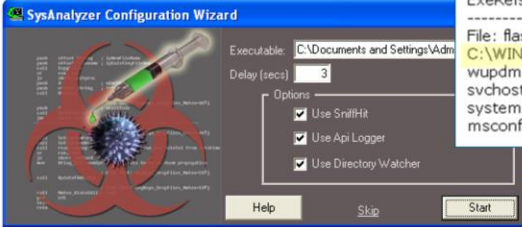
<http://labs.idefense.com/software/malcode.php>

This tool is a special DNS server that will respond to any DNS query with an IP address of your choosing. This way you can easily redirect a malicious network connection in your lab—you simply tell fakeDNS to resolve the hostname to one of your own systems.

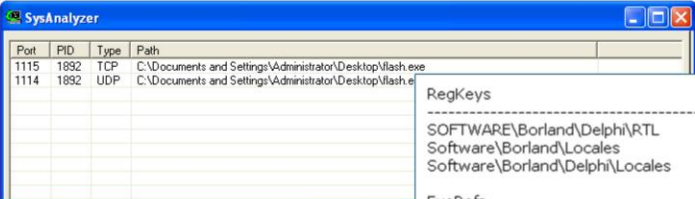
To determine what would happen once the specimen established its connection, you need to set up a listener on the necessary port. In my example, I suspected the specimen was trying to connect to a web server, so I set up Netcat to listen locally on port 80. As you can see, fakeDNS redirected connections to `www.calixto.adv.br` and `www.attack.com.br`, which allowed me to determine that the specimen wanted to retrieve `"/aspnet_client/aspweb.dll"`.



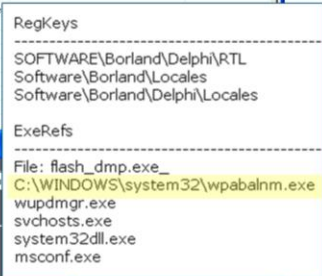
## Examine system interactions with SysAnalyzer.



1



2



3

<http://labs.idefense.com/software/malcode.php>


© 2007 Lenny Zeltser

Another convenient tool for quickly observing behavioral aspects of a malware specimen is SysAnalyzer, also available for free at:

<http://labs.idefense.com/software/malcode.php>

SysAnalyzer combines several miscellaneous utilities to let you observe the malicious executable as it runs on the laboratory system. It lets you detect some of the changes malware made to the system, which API calls it made, which systems it attempted connecting to, and so on. It even has a built-in program for dumping a packed executable from memory to disk. SysAnalyzer doesn't always work, but when it does, it can save you a lot of time.

In this example, I used SysAnalyzer to detect that the malicious program I'm analyzing attempted to make use of a file named "C:\WINDOWS\system32\wpabalm.exe" and that it attempted to listen to local UDP and TCP ports. If findings like these are not immediately useful during your analysis, you can use them to perform follow-up web searches and find additional information about program you're analyzing.



## Automate behavioral analysis with Norman Sandbox Live.

```
flash.exe : W32/Downloader (Signature: W32/DLoader.BOZX)

[ General information ]
* File length: 94208 bytes.
* MD5 hash: 45a31ae4473eae28f7cd72453a207c7.


[ Changes to filesystem ]
* Creates file C:\WINDOWS\system32\wpabalm.exe.
* Deletes file c:\windows\svchost.exe.
* Deletes existing software modules.

[ Network services ]
* Downloads file from http://www.calixto.adv.br/aspnet\_client/aspweb.dll as
C:\WINDOWS\system32\wpabalm.exe.

[ Security issues ]
* Starting downloaded file - potential security problem.

[ Process/window information ]
* Creates an event called .
* Enumerates running processes.
* Enumerates running processes several parses....
```

[http://sandbox.norman.no/live\\_4.html](http://sandbox.norman.no/live_4.html)

 © 2007 Lenny Zeltser

Another good resource for speeding up behavioral analysis of malware is Norman Sandbox Live, available as a free website at:

[http://sandbox.norman.no/live\\_4.html](http://sandbox.norman.no/live_4.html)

This tool performs automated analysis of the specimen's behavioral characteristics after you upload the malicious executable to the Norman Sandbox Live website. After the tool performs its analysis, you receive the resulting report via email.

As you can see on the slide, in my example Norman Sandbox Live provided several details about flash.exe, such as the files it attempts to create, delete, and download. I was able to obtain this information even without running the malicious executable on my own laboratory system. Very convenient.



## Analyze the newly-downloaded executable, if necessary.


Antivirus	Version	Update	Result
Antivir	7.3.0.21	01.08.2007	TR/Spy.Banker.754688
Authentium	4.93.8	12.30.2006	no virus found
Avast	4.7.892.0	12.30.2006	no virus found
AVG	386	01.08.2007	PSW.Banker2.XWJ
BitDefender	7.2	01.09.2007	no virus found
CAT-QuickHeal	9.00	01.08.2007	no virus found
ClamAV	devel-20060426	01.09.2007	no virus found
DrWeb	4.33	01.08.2007	no virus found
eSafe	7.0.14.0	01.08.2007	no virus found
eTrust-InoculateIT	23.73.109	01.09.2007	no virus found
eTrust-Vet	30.3.3311	01.08.2007	no virus found
Ewido	4.0	01.08.2007	Logger.Banker
Fortinet	2.82.0.0	01.09.2007	Spy/Banker
F-Prot	3.16f	01.08.2007	no virus found
F-Prot4	4.2.1.29	01.09.2007	no virus found
Ikarus	T3.1.0.27	01.08.2007	Trojan-Spy.Win32.Banker.ABG
Kaspersky	4.0.2.24	01.09.2007	Trojan-Spy.Win32.Banker.chz
McAfee	4934	01.08.2007	no virus found
Microsoft	1.1904	01.09.2007	no virus found
NOD32v2	1963	01.08.2007	probably a variant of Win32/Spy.Banker.BRY
Norman	5.80.02	12.31.2007	no virus found
Panda	9.0.0.4	01.08.2007	Suspicious file
Prevx1	V2	01.09.2007	no virus found
Sophos	4.13.0	01.05.2007	no virus found
Sunbelt	2.2.907.0	01.05.2007	VIPRE.Suspicious
TheHacker	6.0.3.146	01.08.2007	Trojan/Spy.Banker.chz
UNA	1.83	01.06.2007	no virus found
VBA32	3.11.2	01.08.2007	Trojan-Spy.Win32.Banker.chz
VirusBuster	4.3.19.9	01.08.2007	no virus found






© 2007 Lenny Zeltser


If the specimen you're analyzing attempts to download another program over the Internet, you can continue your analysis by downloading that program and examining its capabilities. In my example, I learned from Norman Sandbox Live that the original executable attempts to download another one over the web. The new executable looked like financially-focused spyware, based on the results of a VirusTotal scan shown on this slide.





## The tools and techniques we discussed help speed-up malware analysis.

-  Examine web-based malware components.
-  Identify general properties of the executable.
-  Assess key behavioral characteristics of the executable.



© 2007 Lenny Zeltser

This brings us to the end of this presentation. My goal was to demonstrate the user of several tools and techniques to speed up malware analysis. In the process, I took you through the steps of examining web-based malware components, identifying general properties of the executable, and assessing the malware specimen's key behavioral characteristics.



# The complexity of malware will continue to increase.




Knowing how to analyze malware allows you to take control of the incident.




© 2007 Lenny Zeltser

Malware analysis is an important skill to an information security specialist, because the complexity of malicious programs continues to increase. By having this skill we can take charge of the incident that involves malicious software, rather than relying on someone else to perform the analysis for us. As in many other situations, knowledge can be the source of our power when defending against attacks on information resources.







**Feel free to get in touch with any questions or suggestions.**



**Lenny Zeltser**  
InfoSec Practice Leader  
Gemini Systems, LLC  
[lenny@zeltser.com](mailto:lenny@zeltser.com)



© 2007 Lenny Zeltser

If you have any questions or comments on this presentation, I'd be glad to hear from you. You can reach me via the email address shown on this slide, and via my website:

<http://www.zeltser.com>

Thank you for your attention!