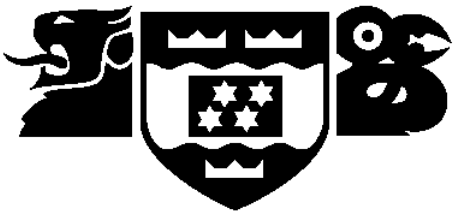
VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui



School of Mathematical and Computing Sciences
Computer Science

PO Box 600
Wellington
New Zealad

Tel: +64 4 463 5341, Fax: +64 4 463 5045
Email: Tech.Reports@mcs.vuw.ac.nz
http://www.mcs.vuw.ac.nz/research

Drive-by-downloads

Julia Narvaez, Barbara Endicott-Popovsky
julia.narvaez@viva-technology.com, endicott@u.washington.edu
University of Washington, Seattle, WA, USA

Christian Seifert, Ian Welch, Peter Komisarczuk
{cseifert,ian.welch,peter.komisarczuk@mcs.vuw.ac.nz}
Victoria University of Wellington, Wellington, New Zealand

Technical Report CS-TR-08/01
January 2008

Abstract

Client-side attacks are an emerging threat on the Internet today. Drive-by-downloads usually occur in which malware is pushed and executed on the client system without consent or notice of the user. An empirical evaluation of the malware with antivirus products is the focus of our research. Client honeypots, security devices that use virtualization to detect malicious web servers that launch these attacks on client system, are used to collect malware and evaluate it with various antivirus products. We show that applications that aim to defraud the victim are the primary malware type identified and show that antivirus products are only able to detect on average approximately 70% of any malware pushed in a drive-by-download attack.

Keywords: Invasive software (viruses, worms, Trojan horses), Client Honeypot, Virtualization, Antivirus

1 Introduction

Widespread attacks launched by hackers to flaunt their hacker skills, such as the SQL slammer or Code Red worm attacks, are now a thing of the past. Attackers today are part of organized crime, aiming to defraud their victims. Their goal is to deploy malware onto the victim's machine to gain complete control, so they can mine sensitive data or abuse the victim's computing resources. As the victim's exposed server services are increasingly protected by firewalls and automated patching mechanisms, the attackers turn to less protected paths to achieve their goals: client-side attacks.

Client-side attacks are attacks that target vulnerabilities within client applications. When a vulnerable client application interacts with a malicious server, this server could launch such a client-side attack. The client usually needs to initiate an interaction with a server to expose itself to a potential attack; passively running a vulnerable client application does not pose much risk.

In our research, we primarily observe client-side attacks launched by malicious web servers on vulnerable web browsers that retrieve a malicious web page from a server. During this process, a browser vulnerability is exploited and malware pushed and executed on the client machine. Such a client-side attack is also referred to as a drive-by-download. They usually happen without notice or consent of the user.

Client honeypots are security devices that use virtualization to detect malicious web servers on the Internet. With a client honeypot we have developed at Victoria University of Wellington, NZ: Capture-HPC 2.0 [5], we now not only have the ability to detect malicious web servers, but also to collect the malware pushed during drive-by-downloads for further examination. Today, no empirical information exists on what sort of malware is being pushed onto the victim's machine. An increased understanding is necessary to effectively protect against this threat.

Utilizing the collected malware and antivirus software, one of the most common security mechanisms used today, we answer questions around the effectiveness of antivirus software in detection and protection against drive-by-downloads. In addition, we take advantage of the classifying abilities of antivirus products and provide a quantitative description of the drive-by-downloads encountered on the web.

2. Background

As mentioned above, drive-by-downloads are a specific type of client-side attacks. Primarily, these drive-by-downloads are launched in the context of web browsing. In such a scenario, as shown in Figure 1 and Figure 2, a web browser requests web pages from a remote web server. As a response, the server returns a web page to the web browser, which contains attack code that exploits a web browser vulnerability (Step 1). Once the attack has been successful, malware is pushed or downloaded to and executed on the local workstation without the user's consent or notice (Step 2).

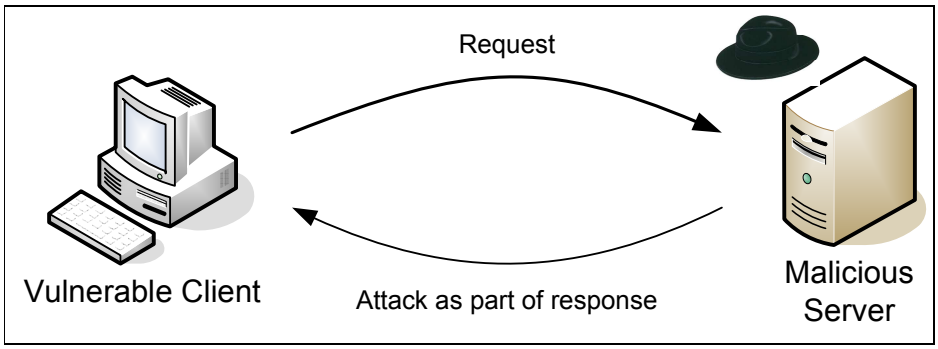


Figure 1 - Client-side attack - Step 1

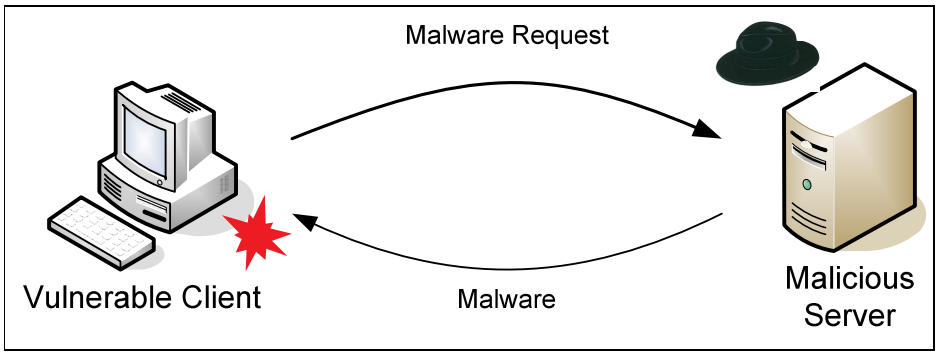


Figure 2 – Client-side attack - Step 2

Client honeypots are security devices that are able to detect such malicious web servers on the Internet. They are usually split into two functional areas: A client component and a controller. The client component is a virtualized vulnerable environment that is tasked with retrieving potentially malicious URLs, which are sent by the controller. As responses from the web servers are consumed by the client, it assesses whether the response is malicious. This assessment is made by monitoring the system for unauthorized state changes that occur after the interaction with the server. Since the client is a dedicated system with no other activity, unauthorized state changes such as a new process, newly installed files, etc., are reliable indications on whether the server that it just interacted with was malicious. Once a malicious web page is detected, the virtual machine is in a compromised state, at which point the controller will revert the virtual machine to a clean known state before sending the client additional URLs to inspect. An architectural overview of this system is shown in Figure 3.

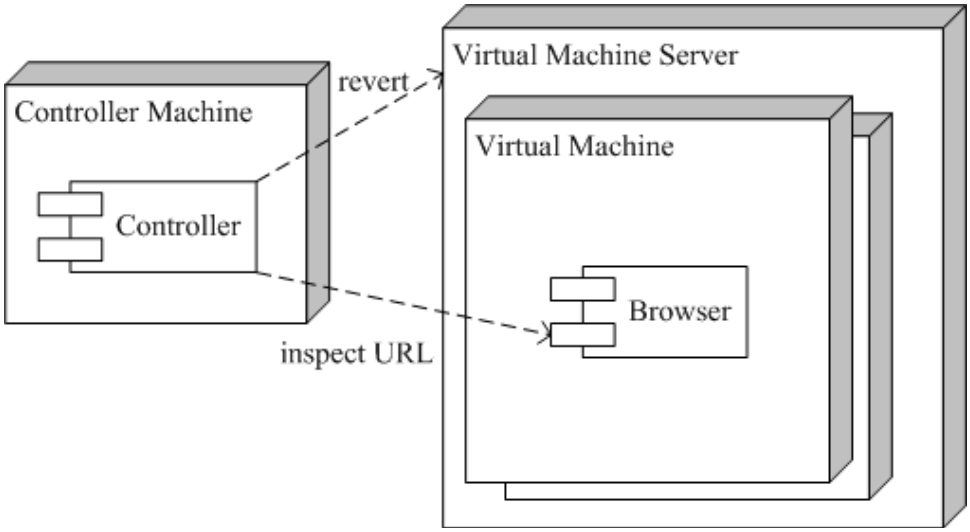


Figure 3 - Client Honeypot System Architecture

Client honeypots, however, are primarily used for research purposes to identify malicious web servers on the Internet. When it comes to protection on an end-user's machine, usually one finds antivirus software. It is one of the most common security mechanisms used [2] that provides a layer of protection while browsing the web. Antivirus software's scanning engines identify malware using various approaches. Primarily, they use static detection in which a so-called virus fingerprint is identified by an antivirus signature. To remain effective, an antivirus product needs to be provided with updated signatures as new malware emerges. These updates are usually provided by a subscription service by the antivirus vendors that automatically updates the signatures on the end-user's installation as they become available. Secondly, antivirus products attempt to monitor the system for suspicious behavior. For example, if an executable appends binary data to another executable, this might indicate viral behavior and is flagged accordingly.

The malware that the antivirus products are able to detect is plentiful. If the signature-based mechanism is used to detect the malware, the antivirus product is usually able to name it. Unfortunately, no common malware naming scheme is adopted by the various vendors. Malware that is named as "Spyware.ABC" by one antivirus product might be named as a generic "Trojan1" by the other vendor. The names themselves are not always descriptive and do not necessarily allow automatic mapping to a specific malware type, which is important for a quantitative analysis of the malware. For example, a product might classify two pieces of malware as Trojan.Peacomm.D and Spysheriff, both spyware. However, the naming of the malware does not allow one to automatically categorize this malware as spyware. Manual mappings need to be created.

We used the following malware classification for our mappings:

- Adware: Adware programs that facilitate delivery of advertising content to the user, and in some cases gather information from the user's computer, including information related to Internet browser usage or other computer habits.
- Bot: Robots used for malicious purposes such as the coordination and operation of automated attacks on networked computers, or to commit click fraud.
- Downloader: Programs that connect to the Internet and download other components or Trojan horses.
- Fraudulent or misleading application: Programs that are not hidden. They try to defraud the victim by incorrectly enticing him/her to purchase something of no value.
- Rootkit: Code that replaces or modifies the operating system and executable programs to hide or create backdoors.
- Spam: Software that involves sending unwanted messages, often unsolicited advertising, to a large number of recipients. Spam is a serious security concern as it can be used to deliver Trojan horses, viruses, worms, spyware, and targeted phishing attacks.
- Spyware: Programs that obtain various types of sensitive information from the victim's machine.
- Trojan horse: Programs that pretend to be useful while masking a hidden malicious purpose. Many of the above classifications are Trojans. We only classified a piece of malware as a Trojan horse if it was classified as a Trojan and no additional information was available to map it to a specific type.

3. Methodology

In order to collect a high number of malware samples for our study, we decided to collect data by inspecting adult sites. This has proven to be a highly infectious area of the web [6] and even a few thousand URLs would yield many malware samples. By submitting adult keywords to the Yahoo! search engine, we obtained several thousand URLs that point to adult sites. Using our high interaction client honeypot Capture-HPC v2.0, which was based on a clean Windows XP SP2 installation, we inspected these URLs from November 8 to November 14, 2007, and were able to obtain malware from 185 malicious web sites. Our client honeypot has a malware collection mechanism that works on the kernel level of the operating system, which allows us to collect any modified as well as deleted files.

Each bundle of malware per URL was scanned using several antivirus products using the provided command line interface. The following products were used:

Commercial:

- CA Anti-Virus 2007
- Kaspersky Lab V7.0
- Norton Internet Security 2007
- TrendMicro AntiVirus 2008

Free:

- ClamWin V 0.91.2

From the scan log files generated by each antivirus product, the following status results were considered as positive malware identification:

- CA: status "Infected"
- Kaspersky: status "Suspicion" and "Detected"
- Norton: entire content of the "Security risks" log
- TrendMicro: status "Found"
- ClamWin: status "Found"

The scan of the malware sample was executed twice. The first scan was done shortly after collection of the malware sample. Prior to the scan, the virus databases were updated and the scan was performed. This scan reflects the effectiveness of the antivirus product to identify malware when a computer with an updated antivirus database encounters a malicious web server. If one malware file was detected by the antivirus engine, the URL that contained that file was considered a match even if other malware files for that URL were undetected. The second scan was performed one month after the malware sample was collected. Again, the antivirus databases were updated, the scan was performed and the same methodology applied. This scan reflects the efficiency of antivirus products to respond to new threats and incorporate new virus signatures into their virus databases.

The second scan also served as the basis for our quantitative analysis of the drive-by-downloads encountered on the web. As already mentioned, an automated mapping between the malware name and the malware class was not feasible due to the lack of a reference collection or a central naming body [3]. Instead, we manually mapped each malware by its name and description to the malware types described above. Because malware descriptions were available for the classifications of the Norton product, we chose Norton's scan results to perform this mapping.

4. Results

From the URLs that were inspected, malware was obtained from 185 distinct URLs. A malicious web server that launched an attack modified, deleted, and usually pushed several files on the compromised machine. All those files were used as the basis for our analysis. First, we will present the results on the detection effectiveness of the various antivirus products followed by a quantitative analysis of malware types.

4.1 Detection effectiveness

Table 1 shows the results from the first scan on all malware collected by the 185 URLs. ClamWin, for example, was able to identify at least one of all malware files in 115 of the 185 URLs. That means that in 62% of the cases, ClamWin would have detected the malware in a drive-by-download attack from the URLs. The lowest detection effectiveness of the antivirus products is CA's with 61%, and the highest detection effectiveness is Kaspersky's with 91%. Note there is a statistical significance in the difference between Kaspersky's ability to detect malware pushed by drive-by-downloads and all the other vendors (chi-square test). However, no such statistical significance exists among the other vendors.

	CA	ClamWin	Kaspersky	Norton	TM
Malware detected	112	115	168	123	127
No malware detected	73	70	17	62	58
Total	185	185	185	185	185
Detection effectiveness %	61	62	91	66	69

Table 1 - Antivirus effectiveness rate in detecting drive-by-downloads

The results of Table 1 are complemented with the Venn diagram in Figure 4. This Venn diagram shows the malware detected by the various antivirus products and allows us to assess whether the effectiveness could be significantly increased if multiple antivirus products were to be combined on one machine. As shown in Figure 4, all antivirus engines are able to detect malware pushed by 94 of the 185 URLs. If all antivirus engines had been installed on one machine, malware from all but 6 of the 185 URLs would have been detected. This equals a 97% detection effectiveness; a significant gain (chi-square test).

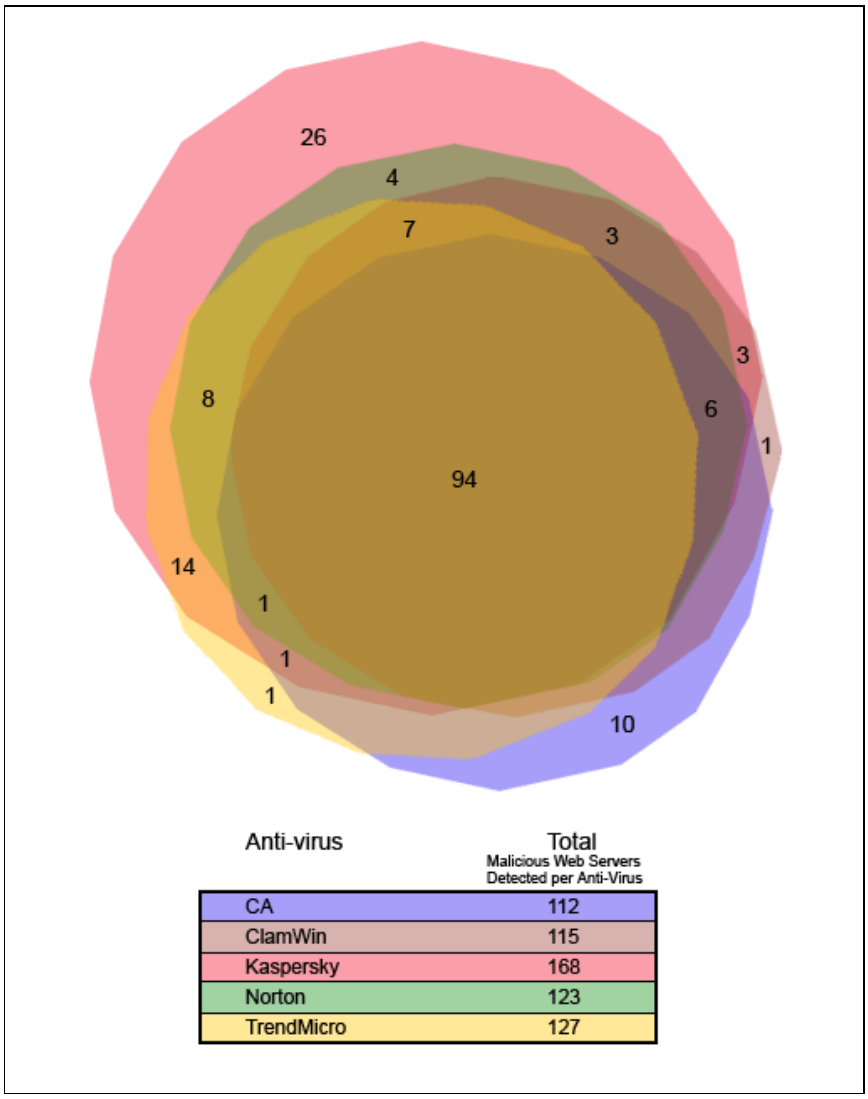


Figure 4 - Malicious web servers detected per antivirus Venn diagram

How well do antivirus products catch up with new malware? The answer of this question lies in Table 2, which shows the result from the second scan after one month has passed. All antivirus products show improvement in their detection accuracy after a month has passed. However, TM does not show a statistically significant improvement in its detection effectiveness (chi-square test). TM now shows the lowest detection effectiveness with 70% and Kaspersky still shows the highest detection effectiveness with 98%. While these numbers are better than the first scan, it is not likely that the antivirus products would perform that well in the real world. Just as the antivirus products had a chance to update their signatures, so did the attackers have a chance to update their malware to remain undetected.

	CA	ClamWin	Kaspersky	Norton	TM
Malware detected	149	138	181	166	129
No malware detected	36	47	4	19	56
Total	185	185	185	185	185
Detection effectiveness %	**81**	**75**	**98**	**90**	**70**

Table 2 - Antivirus effectiveness rate in detecting drive-by-downloads – One month later

4.2 Quantitative analysis of malware types

Most malicious web servers push more than one type of malware when responding to a single request. If an attacker is able to gain control of a system, the system is usually abused in many ways. A system could send out Spam as well as host a fraudulent application. Table 3 shows an example of the malware pushed in a drive-by-download attack from a single web server. It contains adware, Trojan, downloader, and spyware:

Antivirus product	Malware found for one example URL
CA	Win32/Chisyne.BV trojan., Win32/Filitop!generic trojan., ,Win32/SillyDl.YQ trojan., Win32/Zquest.G Trojan, Zquest trojan. Infected.
ClamWin	Adware.CommAd-1, Trojan.Downloader-15330, Trojan.Downloader-17408, Trojan.Downloader-2966, Trojan.Dropper-2852, Trojan.Vundo-349, Trojan.Vundo-845
Kaspersky	not-a-virus:AdWare.Win32.Agent.co, not-a-virus:AdWare.Win32.Agent.ta, not-a-virus:AdWare.Win32.Agent.tb, not-a-virus:AdWare.Win32.TTC.a, not-a-virus:AdWare.Win32.Virtumonde.ks, not-a-virus:Monitor.Win32.NetMon.a, Trojan.Win32.Pakes.akr, Trojan-Downloader.Win32.Small.buy, Trojan-Downloader.Win32.Small.gkh, Trojan-Downloader.Win32.Small.gll, Trojan-Downloader.Win32.VB.bqc, Trojan-Dropper.Win32.Agent.chq
Norton	Adware.Webbuy, Downloader, Spyware.ISearch, Trojan Horse, Trojan.Dropper, Trojan.Vundo
TrendMicro	Not detected

Table 3 - Classification of one sample URL per antivirus product

To determine the type of malware pushed in drive-by-downloads we used the log files from the second scan. By that time, the antivirus products had gained additional knowledge of malware they previously could not classify, providing us with a more complete picture of the various malware types. We mapped each identified malware to one of the classifications listed above, taking into account the malware name as classified by Norton and the description of the malware in Norton's virus database [1]. If we were dealing with a downloader or a Trojan whose purpose is to pull additional malware onto the machine, it was marked as a generic downloader without a payload. If we were dealing with anything else, we classified it as malware with a payload.

Figure 5 shows that downloaders are used in 84% of the attacks. In those attacks, a downloader is initially pushed to the machine. This is usually done because the permissible payload size does not support the payload that the attacker wants to push to the victim. Once the downloader is pushed and executed, a follow-on request is made and the actual payload downloaded. However, not all attacks follow this pattern and some attacks do not make use of a downloader. In this scenario, the payload is directly pushed.

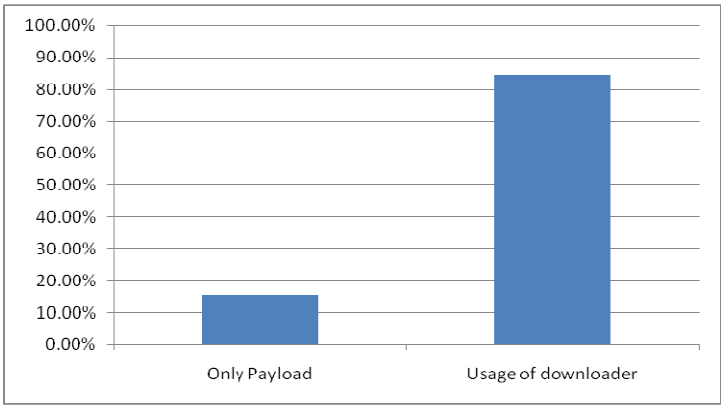


Figure 5 - Downloader usage

Taking a closer look at the actual payload yields additional insights. Of all malware detected by Norton, the generic Trojan class leads the field with 45%. Recall that this is the generic 'other' class if a malware was classified as a Trojan and no additional information was available. We therefore do not know the malicious intent of the software. However, all the other categories give us information about the malicious intent. In second place comes the fraudulent or misleading application (see side box for a detailed description of an example application). Spyware and adware is represented in 6%, rootkits in 2%, and bots and spam senders in 1% of the malware.

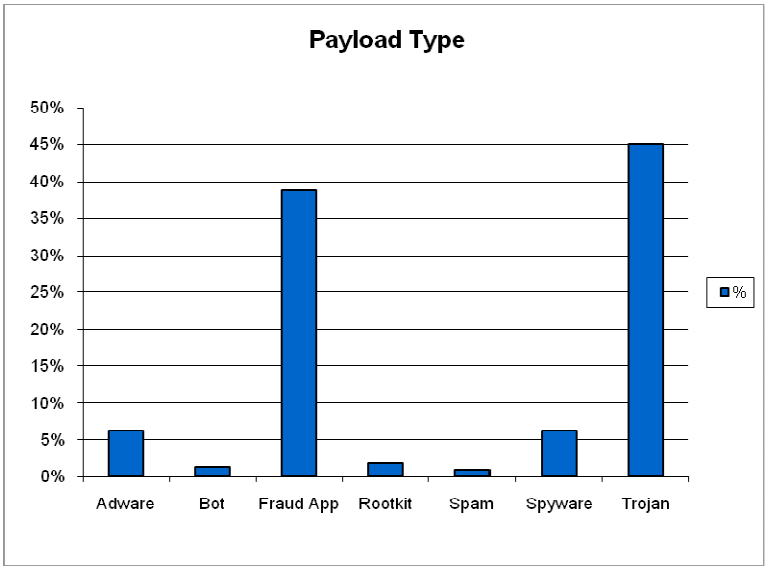


Figure 6 - Payload type

Interestingly, 47% of the analyzed servers seem to push the identical set of malware to the victim. It consisted of 18 files that include a generic Trojan, a downloader and a fraud application. This is indicative of a web exploitation framework that attackers use. (For more information about these frameworks, please refer to our KYE paper on web exploitation kits [4].) The criminals behind the attacks are most likely to purchase and use one of these frameworks rather than creating their own exploit and payload. As such, large quantities of attacks are likely to be canned attacks. This is good news in that it should make detection of those attacks easy and therefore antivirus providers should pick them up. The bad news is, considering the detection accuracy demonstrated in this study, antivirus vendors seem to still have a difficult time.

Fraudulent application

A fraudulent or misleading application, which is presented by 39% of the classified malware, is an application that is not hidden, but rather in full view of the victim. It usually aims to defraud the victim by incorrectly enticing them to purchase something of no value. In this side-box, we take a closer look at a fraudulent application that we frequently encountered: PestTrap.

PestTrap is a fraudulent application that applies social engineering strategy by disguising itself as anti-malware software, informing the user that malware exists on the machine, and then proceeds to entice the user to purchase a license of this "anti-malware software". Figure 6 shows the initial notification about malware existing on the user's machine. Note how the messaging mechanism resembles the messages of the Microsoft Security Center. Shortly after, the software proceeds to scan the machine and provide specific information about the malware that "exists" on the user's machine, as shown in Figure 8. Conveniently, a pop-up window suggests the user should purchase a license of this "anti-malware software" online. Major credit cards are accepted.

Figure 7 - Malware notification

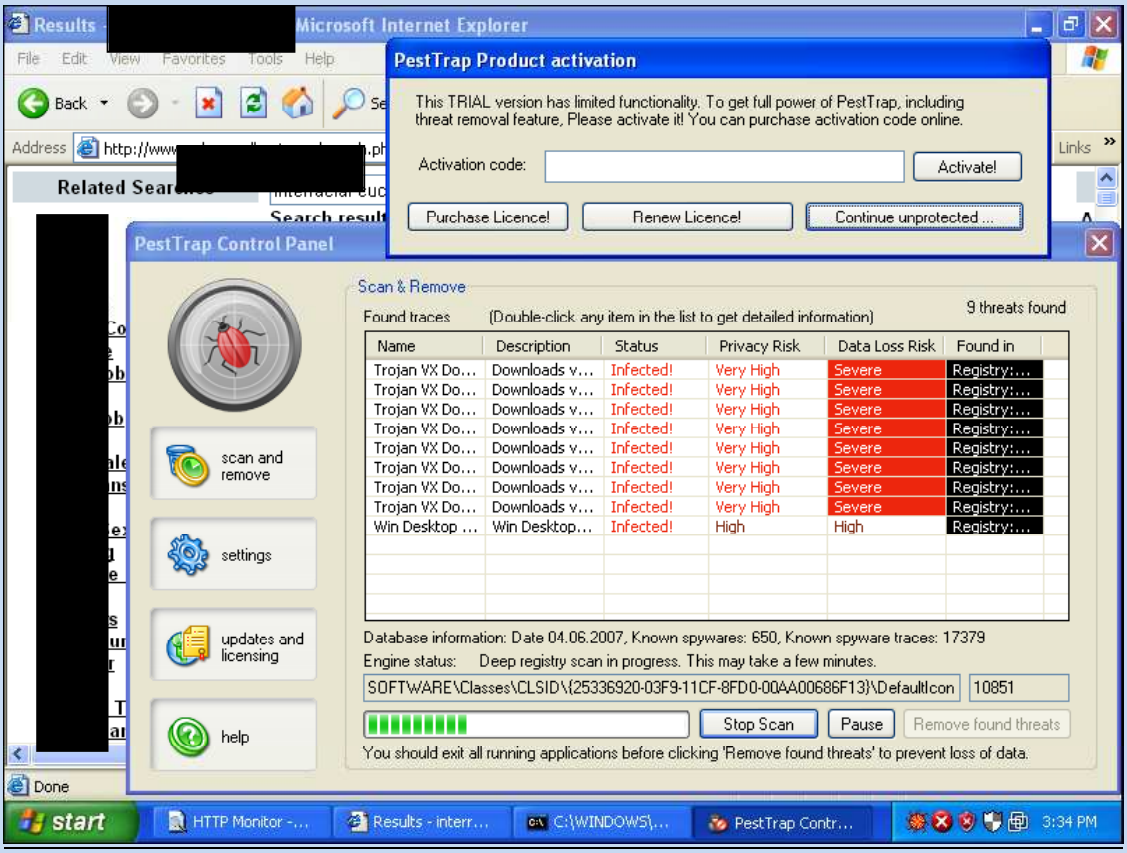


Figure 8 - PestTrap

5. Conclusion

We conclude that antivirus products are able to detect drive-by-downloads; however, the detection effectiveness varies significantly across products and when the scan occurs. While Kaspersky was able to detect 91% of the URLs that launch drive-by-download attacks, the average among the competitors' products was much lower at 64%. That means that for those products only 2 out of of 3 URLs that launch drive-by-download attacks would be detected. Even when signature subscriptions are purchased, antivirus products do not provide complete protection. This does not come as a surprise, but the level of detection effectiveness observed leaves much room for improvement. Again, this demonstrates that we are part of an arms race and it is difficult for the defensive measures to keep up.

The lesson learned here is that defense in depth and breadth is necessary to lower your risk significantly against drive-by-download attacks. In our recent study on malicious web servers [6], we evaluated additional defensive mechanisms against this threat: use of blacklists, patching, and using a non-mainstream browser are some of the measures that reduce risk further and are highly recommended. User education, of course, is another way to improve the situation. If users' awareness of drive-by-downloads and the antivirus detection effectiveness of such attacks is raised, users might be more cautious about the web pages they visit and links they click on.

A measure that can increase the detection effectiveness of antivirus products is to run multiple products in tandem. This is an example of defense in depth and breadth: if one antivirus product fails to detect malware, another antivirus product might detect it. However, antivirus vendors do not seem to like this idea as some purposefully claim exclusiveness on a machine. We encountered prompts from antivirus product's installation programs to first uninstall the existing antivirus products before continuing with the installation. However, some products do not seem to have this issue, showing that these incompatibilities are not rooted in technical causes.

The degree of protection that each antivirus product offers varies considerably if taking into account the time factor. An antivirus product is much more effective if provided with updated signatures. The average detection effectiveness of all five antivirus products increased from 70% to 83% one month after the malware was downloaded. If one assumes that malware also adjusts accordingly to evade detection of antivirus products, updating of virus signatures is imperative to remain effective. For instance, if the signatures were not updated immediately after our data collection completed and we would have collected drive-by-download malware samples again after a month, the average detection effectiveness would probably have decreased, because the malware would have adapted to evade detection in this time frame as well.

With respect to the classification of the malware by antivirus products, this has been a great aid in understanding the malware types pushed by drive-by-download attacks. From the classified malware, we know that a majority is represented by fraudulent applications that do not hide themselves from the user, but rather use social engineering to defraud the victim. This is valuable input to develop a defensive strategy against drive-by-download attacks. User education is a measure that would assist to reduce the risk.

However, antivirus classifications are not comprehensive; only a portion of the malware is detected. In addition, a lot of malware is placed into a generic bucket "Trojan", which could be anything. So while we observed 39% of malware identified to be a fraudulent application, the actual percentage is likely to be much lower if this shortcoming is taken into account. Classification is also still very labor intensive. While the antivirus products automate naming the malware, the naming schemes do not map directly to malware types, and additional classification work is required.

Overall, antivirus products are a good tool in the defensive arsenal against drive-by-downloads. They assist in defending against drive-by-downloads and they are valuable in researching and classifying malware. However, there are shortcomings with the tools we have used and we are looking forward to the advances the antivirus products will bring in the future to provide better protection.

References

[1] Symantec Corp., Security Response, 2007, Available from http://www.symantec.com/security_response/index.jsp; accessed on 20 December 2007.

[2] Gordon, L.A., Leob, M.P., Lucyshyn, W. and Richardson, R. CSI/FBI Computer Crime and Security Survey, Computer Security Institute, 2006.

[3] Gordon, S. What's in a Name? *SC Magazine*, *June 2002*.

[4] Seifert, C. Know Your Enemy: Behind the Scenes of Malicious Web Servers, The Honeynet Project, 2007, Available from http://www.honeynet.org/papers/wek; accessed on 7 November 2007.

[5] Seifert, C. and Steenson, R. Capture - Honeypot Client, Wellington, 2006, Available from https://www.client-honeynet.org/capture.html; accessed on 22 September 2007.

[6] Seifert, C., Steenson, R., Holz, T., Bing, Y. and Davis, M.A. Know Your Enemy: Malicious Web Servers, The Honeyney Project, 2007, Available from http://www.honeynet.org/papers/mws/; accessed on 25 September 2007.