

THE CHALLENGE OF DETECTING AND REMOVING INSTALLED THREATS

Jason Bruce

SophosLabs, Sophos Plc, The Pentagon,
Abingdon Science Park, Abingdon, OX14 3YP, UK

Tel +44 1235 544142

Email jason.bruce@sophos.com

ABSTRACT

The days when the competitiveness of an AV product was determined by the ability to detect a bucketful of samples will soon be behind us. New tests, driven by the requirement for AV products to deal with spyware, will measure the ability of an AV product to manage any given threat from detection to full removal.

Detecting and removing installed and active threats presents many challenges, particularly where multiple files, processes and registry components are involved. The ability for these components to be updated from the Internet at any time and with varying frequency only complicates the issue further.

This paper will discuss the challenges that are faced by AV vendors in modifying their products to move away from blindly detecting and deleting a given set of miscellaneous samples to detecting and removing samples in the context of the installed threat.

INTRODUCTION

When viruses first began infecting files the message from anti-virus products at the time was to label those files as infected. In some cases the files still worked as expected but they were performing additional functionality which they were not designed for and which may ultimately have had some detrimental affect on the whole system.

The solution for fixing an infected file was to replace it with a copy of the original, thus making absolutely certain that the files on the system were clean. This solution was acceptable in the days of small, uncluttered operating systems with a limited number of installed applications. As we moved through the 1990s operating systems became increasingly complex and on top of that technology improvements meant that users could install greater numbers of increasingly complex applications. The existing solution to a virus attack – of replacing infected files – became less and less practical, particularly for home users and small businesses, and as a result there was an increasing demand for anti-virus solutions to provide disinfection for virus-infected files.

Today a similarly significant change in demand is occurring. Threats are increasing in complexity to the point where it is no longer possible to provide simple instructions to assist with removal. The explosion of spyware has seen a rise in the number of threat types that install many components, such as files and registry entries. So whereas once, removing a trojan would have been a matter of killing a process, removing a file and deleting a registry entry, we now have threats that require the removal of many of each of these components. In addition

there has been a significant rise in the number of threats making use of stealthing and anti-removal technologies to further complicate removal procedures.

The complexity of removing threats with many components has led to customers demanding that security solutions manage the threat for them. *Microsoft* might have you believe that there are cases where this is just not possible [1]: ‘When you are dealing with rootkits and some advanced spyware programs, the only solution is to rebuild from scratch. In some cases, there really is no way to recover without nuking the systems from orbit.’ But as with the solution originally recommended for files infected with viruses, this is not currently a practical solution, particularly for home users and small businesses.

THE MULTI-COMPONENT THREAT

There are now many threats, particularly those associated with potentially unwanted applications such as adware, that add and modify multiple process, file and registry entries on the system. These multi-component threats present a number of challenges to security products attempting to reverse the changes that have been made to the system. Removing a multi-component threat with the intention of restoring a system to a stable and secure state may only involve reversing some of the changes resulting from the installation of that threat. However, removing a threat to the satisfaction of a customer means detecting and removing every installed component and restoring modified settings back to their original values.

Component classification

Installed threats can be thought of as containing two key categories of components: primary and secondary components.

Primary components

Primary components are the most significant of any threat. These are the ones that actually provide the threat with its functionality or cause that functionality to be loaded when the operating system starts up or a user logs in. Removing a threat’s primary components should, in most cases, be enough to neutralize the threat, preventing it from causing further damage, loss of information or error reports from the system. There are two categories of primary component.

Individual primary components: these are typically the executable files that provide the functional features of the threat, processes associated with those files and associated load points in the registry. Detection of these components will indicate that a system is affected by a particular threat, but simply removing them is not necessarily a sufficient solution to the infection.

Compound primary components: these are the components of a threat that are implemented by the modification and addition of multiple files and registry entries. Examples are services registered with the service control manager, layered service providers (LSP) hooked into the *Windows* TCP/IP handler, *Internet Explorer* browser helper objects (BHO) and other registered COM objects. Effective removal of compound components requires the modification of all the affected entries on the system. In some cases failing to fully manage all the changes made by a compound component will not result in any detrimental effects; in other cases significant damage can

be caused. Removing a file associated with an LSP for example, will result in the loss of network connectivity if the Winsock entries in the registry are not modified to remove that particular LSP.

Secondary components

Secondary components are typically comprised of registry entries and ancillary files such as data files, logs, configuration files, etc. If left on the system many of these components would remain benign as they are of no use without the primary components of the threat. However, for completeness they should ideally be removed or restored to a pre-infected state.

REMOVING THE INSTALLED THREAT

There are two significant steps in the removal of an installed threat. First of all the installed threat and all its components, both primary and secondary, need to be identified. Once that's completed the analysis of the threat can be used to define the actions that are required to remove the threat from the system so as to leave the system in a usable and secure state.

Detection phase

The objective of the detection phase is to build up a complete list of all the installed components of any threat found on the computer being scanned. With the consideration of multi-component threats to take into account we'll see that the use of two scanning techniques is required in order to effectively and efficiently build these lists of components.

Scanning techniques

Content scanning and context scanning are the two main static detection techniques that are generally used for detecting the presence of an installed threat and collating all the components of that threat.

Content scanning

Content scanning techniques, traditionally used by AV scanners, can be used for detecting primary file components such as the executable components of a threat. However, this technique is not necessarily the most suitable for detecting secondary file components, particularly data files, log files, etc. that are subject to frequent, unpredictable changes. In addition it is not always necessary to detect every component directly via content scanning, since once the scan has determined that a particular threat is installed there are more efficient methods, such as the context scanning technique discussed below, for detecting the remainder of the threat.

Context scanning

Contextual scanning techniques, more commonly relied upon by dedicated anti-spyware solutions, provide a method for detecting threats based on the known presence of a particular set of entries on the system being scanned. This method uses rules such as combinations of names

and locations of file and registry settings to determine whether a threat is installed on the system.

For example, consider the following set of entries on the system:

```
File: <system>\taskmon.exe
File: <system>\shimgapi.dll
Registry:
HKLM\Software\Microsoft\Windows\CurrentVersion\Run\Taskmon
= taskmon.exe
Registry: HKCR\CLSID\{E6FB5E20-DE35-11CF-9C87-00AA005127ED}\InProcServer32\Default= "shimgapi.dll"
```

Without scanning the contents of either of the files we can positively identify the system as being infected with W32/MyDoom-A.

Context scanning is not the most effective or practical technique when used on its own. For example, this scanning technique is not very effective at the gateway where no installed context rules can be applied.

There is also the complication of making a positive identification of a particular threat, where common file names or registry entries are being added or modified, leading to non-specific reports such as, 'this file and these registry entries are suspicious'. There is also a greater risk of false positive reports, particularly when relying on individual component attributes such as the names of files or registry entries.

For example, consider the following file and registry entry:

```
File: <windows>\system.exe
Registry key:
HKLM\Software\Microsoft\Windows\CurrentVersion\Run\system
= system.exe
```

At the very best this combination of components can be labelled as suspicious, but without scanning the contents of the file system.exe it is not possible to tie it down to any one particular threat, or even a threat at all.

Where context scanning has its most effective use is when used in combination with content scanning on the desktop to assist in collating all the components of an installed threat. The positive identification of one or more of a threat's primary components, e.g. files or processes, based on their contents, can be used to trigger context scanning rules. The context scan will use these rules to identify the components of the threat installed on the computer without relying on a content scanner to positively identify every individual component.

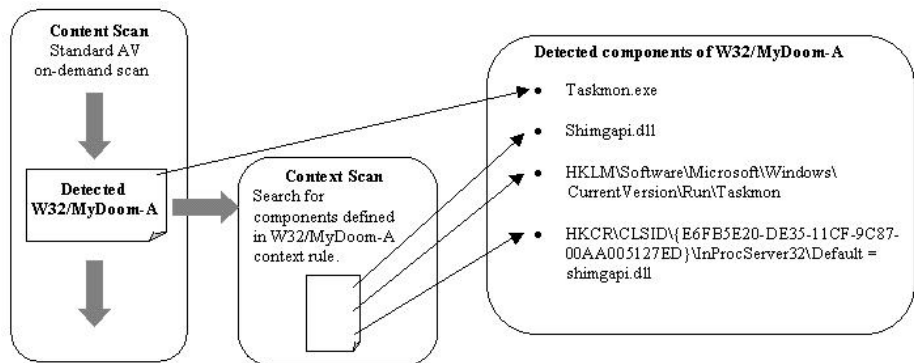


Figure 1: Scanning for W32/MyDoom-A components using a triggered context rule.

Consider the W32/MyDoom-A example above. If a scheduled scan of a system detects W32/MyDoom-A in taskmon.exe then this information can be used to trigger a context rule that seeks out the other components of W32/MyDoom-A (see Figure 1).

Scanning requirements

The majority of standard non-malicious applications can be considered to have a predictable, well behaved installation. The information that the application is installed on a computer is sufficient to determine the names and locations of all components, both primary and secondary.

There are many cases where this is true for malicious applications as well. Many trojans and worms are predictable between infections, and this in turn can make them easy and quick to remove, as detecting a single component is enough information to clean the whole of the malicious application from the system without scanning the whole system to detect each component explicitly. This is true for the W32/MyDoom-A example in Figure 1.

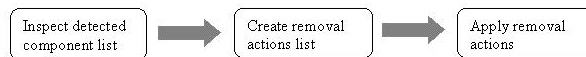
This predictability is not true of all malicious or potentially unwanted threats though. There are examples where more extensive scans of the system are required to ensure that all components of a threat have been identified correctly so that a successful cleanup of the system can be carried out (Figure 2). This is particularly true of threats such as the adware application Look2Me that contains randomly named components.

A further complication is added when threats use sophisticated stealthing techniques to hide components of their installation. Depending on the techniques used it may be possible to infer the presence of a stealthed threat using standard user mode content scanning and to use contextual techniques to build a full list of installed components. Otherwise more sophisticated rootkit detection techniques will be required.

Removal phase

Collecting information about all the installed components of a threat is the first step in removing a threat from a system. The second step is to carry out that removal based on the gathered information – but this is all too commonly becoming a far greater task than simply terminating, deleting or modifying those components.

The traditional line of attack is to remove the threat from memory, remove the infected file and registry entries and finish up by restoring any modified file and registry settings. While this is still where we ultimately want to get to, it is not always possible to achieve this without additional steps along the way involving actions ranging from the renaming of files and suspension of processes to a complete reboot of the operating system.



Defining the removal procedures for a specific threat is becoming increasingly dependent on a detailed analysis of that threat to determine how the threat behaves when removal actions are applied.

Anti-removal techniques

A complete technical paper could be written on anti-removal techniques, which is not my intention here. Instead I will refer you to Sergei Shevchenko’s article [4] and Eric Chien’s paper [5] for examples of the complications that can be involved. I will summarize here the considerations that security products must take into account when attempting to subvert the attempts by threats to prevent them from being removed.

Threats are increasingly employing techniques to complicate the removal of their software. Some of these techniques have been seen for many years in malware. One example is monitor or watchdog processes where one or more threads are set up to continually monitor the active status of the threat and its various components, as effectively implemented by W32/Chir-B back in 2002.

Another example is injecting code into system processes that cannot be restarted without shutting down the system, as used by W32/Lovgate in the following year. More complex techniques, however, have been coming from potentially unwanted applications such as adware. There is big money to be made out of these products and the vendors behind them often have teams of engineers to implement techniques that will prevent users from removing the applications. Techniques used include installing drivers to implement rootkit-style protection mechanisms, regularly renaming components of threats to subvert context detection and modifying privileges to the extent that even the administrator has no power to do anything about the threat.

In the more simple cases these techniques can be subverted by carefully considering the order in which components of the threat are removed from the system. In more complex cases it may not be possible to remove a threat in place and instead it

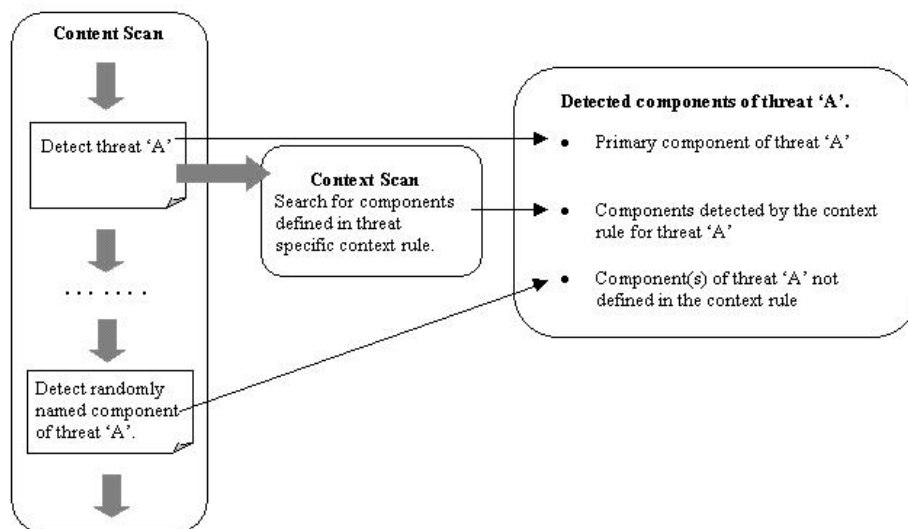


Figure 2: Updating the list of threat components when the context rules are unable to define all components.

is necessary to implement removal strategies that involve rebooting the operating system and completing removal actions during or after the boot up sequence.

Shared components

An important consideration to make when removing a threat from a system is identifying the components of a threat that are shared, or non-exclusive. These are components that may have been installed by the threat but are actually third-party applications or components used to provide additional functionality to the threat but not designed exclusively for that threat.

An example of a non-exclusive component is a language library. A threat may install a library to ensure that it runs on the targeted system but it's not possible to say, without a reliable snapshot history of the system whether that library file was on the system prior to installation of the threat. It is therefore unsafe to remove such components, however it should also be pointed out that they would not pose any increased security risk.

Pre-infection settings

Removal of a threat also involves the reversal of changes made to legitimate files or registry entries that exist on the affected system. This presents a challenge since it is usually the case that the security product carrying out the removal operation will be unaware of the state of the affected system immediately prior to infection. In cases where a threat-specific entry has been set, e.g. an *Internet Explorer* start-up entry, then a safe default can be set. However, if the changes are not threat-specific, e.g. *Internet Explorer* security zone settings, then it is not even possible to determine whether the changes made by the threat actually made any difference to the original settings.

There should arguably be some consensus in the industry to define default values for commonly modified entries on a system where it is unlikely that the pre-infected state is known.

CONCLUSION

In this paper I have described how security products such as anti-virus and anti-spyware software generally use two main static scanning techniques to assist with the detection of the components of an installed threat: the content scanning traditionally employed by anti-virus solutions and context scanning solutions often relied upon by dedicated anti-spyware solutions.

I have shown that neither scanning technique is the sole solution to the problem of identifying the components of a threat that should be flagged for removal. Instead, a combination of the two techniques needs to be implemented to provide the most effective and efficient solution. Having the two scanning techniques also provides an additional level of mitigation against the frequent updating we see occurring with some threats today.

Identifying all the components of an installed threat is not the sole requirement for removing that threat. Authors of malware and potentially unwanted applications are aware of the attempts by anti-virus and anti-spyware products to remove their applications. So just as malware authors have historically strived to create viruses that subvert detection the aim now is to implement techniques that prevent, or at least

complicate, the removal of unwanted applications. There are strategies for getting around these anti-removal techniques but it means that more often we are seeing threats that require removal identities that are as specific as the identities that detect the threat.

The varying effectiveness of the different security solutions offering removal of installed threats has led to the requirement for specific testing to provide objective comparisons of how well these security products perform at this task. However, unlike traditional anti-virus testing where there are only two results, detected and not detected, there is a certain amount of middle ground with tests that measure the effectiveness of removal. A system can be rendered stable and secure without necessarily removing or restoring every single component of a threat.

REFERENCES

- [1] Naraine, R. Microsoft says recovery from malware becoming impossible. eWeek.com. April 2006. <http://www.eweek.com/article/0,1895,1945808,00.asp>.
- [2] Paget, F. Adware & Spyware Free Detection/ Cleaning Tips and Techniques. AVAR Conference 2005.
- [3] Polischuk, A., Kovtun, A. Operating System Recovery by Anti Virus software. AVAR Conference 2005.
- [4] Shevchenko, S. Standing the privilege attack. Virus Bulletin, June 2005.
- [5] Chien, E. Techniques of adware and spyware. In Proceedings of the Virus Bulletin International Conference. 2005.