FEATURE 3

SYSTEM CLEANING: GETTING RID OF MALWARE FROM INFECTED PCS

Maik Morgenstern & Andreas Marx
AV-Test.org, Germany


Malware has evolved quite significantly in the decades it has been around. In the beginning, file-infecting viruses were the main threat (*although there were times in the very early days when boot-sector viruses caused the most infections - Ed*). Simple at first, they quickly evolved into more complex incarnations, using techniques such as self-encryption and eventually leading to polymorphic variants.

The anti-virus industry's response was also pretty simple at first. AV products were able to detect a virus and tell the user about it, but no cleaning routines were provided. Infected files had to be replaced with a clean version of the original in order to fix the problem. However, with the increasing complexity of operating systems and the ability of users to install more and more applications this soon became impractical. In response, AV vendors introduced disinfection capabilities to their products, which had to deal with more complex virus creations from year to year, covering not only executable files but also *Office* documents and other file types.

However, malware evolution did not stop at that point. Rather than simply infecting files, multi-component approaches affecting many parts of the system became the standard in malware and remain so today. This is especially true in the case of spyware, which traditionally makes a lot of changes to the file system as well as to the registry. Many other malware attacks also consist of several components that have to be dealt with by AV software. The easy cases with only one process, one file and one registry value are certainly getting rarer and the complexity of malware threats is increasing. This refers not only to the magnitude of changes to the system, but also to the techniques used and the overall behaviour of the malware. Rootkit techniques and anti-removal measures are some of the most challenging for detection software [1].

The AV industry responded to this new challenge and learned to remove the malicious components from the system. Simple cases were easy to deal with: terminate the process, remove the executable and maybe even handle corresponding registry entries. However, with the increase in volume and complexity of malware the removal of malicious components became more difficult. In order to remove a malicious item successfully from a system, it is necessary to know exactly what to remove. This means that some kind of disinfection routine must be in place, which in turn requires some analysis of the malware. This pretty much describes the way in which AV vendors traditionally did the job: AV researchers analysed the malware, identified the changes to the system and could provide a disinfection routine with the next update. While this approach worked well some years back with smaller volumes of slower-spreading malware, it has serious drawbacks now. Often, it just takes too long for a dedicated disinfection routine to become available.

The solution seems obvious: generic approaches for disinfection, which don't rely on an analysis from the AV vendor. While there are certainly promising attempts that can handle the simpler cases, the more complex cases still pose a problem. The components that are detected by static or dynamic mechanisms can usually be removed, however this is not always true for linked components, be they files and directories or registry entries. This means that some parts of the malware can indeed be successfully removed, but others which can still be a threat to the system remain. These findings and more details are available in [2].

With the above in mind it is clear that the testing of system cleaning capabilities is still a very valuable exercise. There are many variables that have to be considered by the AV vendors which could prevent successful cleaning. It is useful, therefore, to run tests that determine how well today's products are able to handle system disinfection and how well they can cope with special circumstances such as anti-removal techniques. We will describe the basic requirements of such tests, present some of the details of our testing procedures and look at the results of some of our recent tests.

SAMPLE SELECTION AND CREATING THE TEST SET

As for most tests, sample selection is one of the first and most important steps in the testing process. A wide variety and a large number of samples must be used in order for the results to have statistical relevance. Due to the complex nature of the tests, the test set cannot be as large as it would be for a static scan test, but other factors can still help ensure its relevance.

The basic requirement is that the samples are active and actually perform changes to the system. The likelihood of the products being able to detect the samples must also

be considered, because this will influence the disinfection process. There may be signature-based detection which could trigger a dedicated disinfection routine, proactive detection which might lead to a generic disinfection routine, or no detection, which obviously won't trigger any disinfection process.

Besides these basic requirements, different malware types and families should be chosen for the test set, to cover different behaviour and levels of complexity. The samples should also be currently spreading in the wild, to reflect real-world threats. Finally, the sample selection and analysis process must be performed on the same operating system and under the same conditions as those in which the test will be carried out. This is necessary to make sure the criteria that have been used for selecting the samples still apply when testing.

The tester needs to know exactly what changes to the system are performed by the malware. In order to determine this, an automated analysis tool is used, which records every change to the file system and registry and discovers newly created processes. This gives a comprehensive overview of the relevant malicious activities on the system and helps in the sample selection process.

The same tool can also be used to solve two common problems encountered when testing active malware: reproducibility and comparability. Since active malware may change its behaviour depending on several variables, including some that cannot be controlled by the tester, the actions of the malware – and therefore the changes to the system – may be different on every test run. This could prevent the tester from reproducing a test result, since the malware may never act as it did before. It could also prevent the tester from comparing the cleaning performance of one product against that of another, since they might have to cope with different malware behaviour and some may be easier and some harder to deal with. These issues are particularly likely to arise with malware which downloads additional components from the Internet.

Since the scope of this test extends only to the cleaning of an infected system and not the prevention of infection, the analysis tool can be used to help overcome these problems. The recorded system changes are saved in a special archive format (packages) which can be used to restore the whole infected state on any system at any time. This easily solves the two problems mentioned above: the package can be used to reproduce exactly the same infected system state as often as necessary. This in turn means that exactly the same conditions can be created for every product in the test and their cleaning performance can easily be compared.

PERFORMING THE TEST

The testing procedure is straightforward, especially with the help of the packages from the analysis tool. We use an image with an up-to-date installation of the AV product under test and turn off the on-access protection in order to be able to restore the infected system state. This is done by replaying the system changes recorded in the package. After this is finished and the system is in a known infected state a system scan is carried out using the default options. Whenever anything is detected, we let the AV product run its cleaning or disinfection routines to remove the malicious components. After allowing any required reboots and additional scanning and cleaning steps, the final system state is determined using the same tool as used in the analysis and preparation steps. Since we know exactly which changes to the system have been made by the malware, we can also determine exactly which components have been removed by the AV software and which components have been left behind.

This gives us the raw information as to what and how much has been detected and removed, but it does not represent the cleaning success. In order to assess this, the system changes must be categorized by risk level.

First, there are the changes that are clearly malicious, which must be removed, reverted or set to default settings. These include malicious executables and the linked start entries in the registry or file system, but also extend to modifications to the hosts file as well as altered security and browser settings in the registry.

The second category contains unpleasant or unwanted, but not actually dangerous, system changes. One example is pornographic images that accompany a lot of malware these days. This should certainly be handled by the AV product in corporate environments and home users will want them removed too, especially where children use the computer.

The last category contains changes that don't have any real effect but are visible on the system. These include directories, trash or 0-byte files or junk registry entries that are not used by the operating system.

In order to clean a system successfully, the bare minimum an AV product must be able to do is to handle the first category of changes and disable the malware effectively. This means the malicious processes must be terminated, the corresponding files and the start entries must be removed. Any changes to security and browser settings as well as modifications to the hosts file should at least be detected and reported to the user. Since the pre-infection settings are often unknown, it is not possible simply to reverse these changes, but reverting to the

Product	Version	Detection of inactive samples	Detection of active malware	Disabling of active malware	Removal of active malware
	Reference	5	5	5	5
Avira Antivir PersonalEdition Classic	7.06.00.270	5	4	4	3
BitDefender Antivirus 2008	11.0.0.15	5	4	4	2
BullGuard Internet Security 2008	8.0.0.1	5	4	4	2
F-Secure Anti-Virus 2008	8.00 build 101	5	4	4	2
G DATA AntiVirus 2008	18.3.7338.740	5	3	3	3
Kaspersky Anti-Virus 7.0	7.0.0.119	5	5	5	4
McAfee VirusScan Plus 2008	12.0 Build 176	5	4	4	3
Symantec Norton AntiVirus 2008	15.0.0.58	5	5	5	4
Panda AntiVirus 2008	3.00.00	5	4	4	3
Windows Live OneCare 2	2.0.2500.14	5	4	4	3

default settings is always an option that can be offered by the AV software.

What we often see is that only the malicious executables are handled. Additional dropped files, registry entries and other changes are not dealt with. This is critical for several reasons. The first has been explained above – many changes are themselves dangerous, e.g. in the case of changed browser settings, the user might be redirected to a malicious website that will infect the system with the latest version of the malware again. Another reason is the uncertainty in which the user is left when not all relevant components of the malware are removed. Especially in the case of an infection, a user might want to obtain a second opinion. This could lead to the detection of the left-over malware components by a second AV product and the user will most likely lose confidence in his original security software. The increasingly common 'light grey' software products that pose as security software but actually produce rather strange outputs may compound this problem [3]. These applications do not have any real eligibility to be on the market, but 'detecting' the left-over components from an incomplete system disinfection might just be what they were looking for as justification.

Besides handling the first category of changes, it would of course be very desirable to handle the other categories as well. Not doing this will not usually mean a failure in the test – as long as the malware is effectively disabled – but the product's failure to deal with all system changes will be reported.

SOME TEST RESULTS

In this section we will present a few small-scale test results, which illustrate some of the common problems encountered but also show that some products are able to handle the system cleaning task successfully. These results have been published in the German *ComputerBild* magazine [4].

The test was carried out at the beginning of 2008 on *Windows XP* (32-bit, SP2) and the products (in their most current versions) were updated and then frozen on 7 January 2008. The test was carried out as described above.

The results presented here are from tests run against five samples taken from the then current WildList – meaning that signature-based detection of the original sample should be guaranteed. There were three rather easy ones: Win32/Rbot!FB26, Win32/Spybot!ITW203 and Win32/Stration!69F2, as well as Win32/Feebs!8897, which uses rootkit techniques, and Win32/Rontokbro!E517, which tries to terminate AV software. The behaviour of the latter two samples complicated the cleaning process for some of the products.

While all products were able to detect the malware samples in an inactive state, there were some problems when they were already installed and active on the system. *G DATA* and *BullGuard* failed to detect the Win32/Feebs!8897 infection due to its use of rootkit technologies and were consequently not able to clean the system. All the others were able to detect and disable

this threat, however only *Kaspersky* and *Norton* achieved full removal. The remaining products didn't handle the 'ShellServiceObjectDelayLoad' registry entry that was used to restart the malware on reboot and could possibly cause false positives if not removed.

The other problematic sample was Win32/Rontokbro!E517, which terminated seven out of the ten tested AV products or prevented them from scanning. Only *BullGuard*, *Kaspersky* and *Norton* were able to deal with the sample and disable it. However, there were still some problems. The malware disabled the editing of the registry with the 'DisableRegistryTools' entry and none of the products dealt with this. While it is perfectly understandable for this entry not to simply be set back to the default value – which would allow editing of the registry again and may be different from the pre-infection state – it is not clear why this change was not reported to the user. An analysis of the sample in the lab certainly detected the change and it is also safe to assume that most users do not prevent access to their registry. This makes it pretty clear that the disabled registry would in most cases be the result of the malware behaviour and should therefore be reported.

Another issue was the modified hosts file. The *Norton* product did clean some parts of it, especially those that affected *Symantec* addresses, but it left a lot of other bad entries. The other two products that were able to handle this sample simply moved the file into the quarantine. While this effectively disables the malicious intent, it also removes user entries that may be necessary for the system to work as expected.

The other samples didn't pose any serious problems to the AV products: the Win32/Rbot!FB36 sample challenged *BitDefender*, *BullGuard* and *F-Secure* a little with its run registry entry that was left behind by these products, but Win32/Spybot!ITW203 and Win32/Stration!69F2 were both handled effectively by all products.

CONCLUSION

Preventing an infection when the malware sample is known is rather easy. Heuristic and generic detection as well as behaviour-based approaches are a big help in detecting unknown malware and preventing an infection. However, none of these approaches is 100% safe, and there is always the chance that new malware will remain undetected and infect systems. Also, we are well aware that some users do not use up-to-date AV software and only wake up when it is too late and discover an infection on their system. Then is the time for system cleaning routines.

As we have pointed out above, there are cases where AV products work perfectly well, not only disabling the threat but also removing all relevant parts. This is possible when a dedicated disinfection routine is available or if it is an easy case that can be handled by a generic routine. However, not every piece of malware is simple, and when a more complex piece of malware is encountered – such as one that tries to evade detection and removal and which clutters the system with lots of different components – some AV products show certain weaknesses.

The problems may even start with detection of the malware, because some products cannot handle rootkit techniques or because the malware terminates the security software. But even when it is detected, this does not mean that all parts of the malware will be disabled.

Finally, there is the removal of the malicious components – the performance of many current AV solutions in this area is disappointing in many ways. Registry entries are not handled or only some of them are removed, security and browser settings are ignored and the hosts file is only partially cleaned or simply quarantined. System cleaning involves a lot more than just detecting the malware process and removing the corresponding file. Depending on the complexity of the malware, many more steps might be necessary and must be taken carefully.

In order to solve some of the problems, there is always the option of using a bootable rescue media. Since the malware (and a possible included rootkit) is not active then, no scanner can be terminated. However, this does not replace the need for thorough analysis of current threats and the further development of better generic disinfection routines. Both of these are needed not only to disable (parts of) the malware, but also to remove all relevant components, to keep the user in a safe and confident state.

REFERENCES

[1] Bruce, J. The challenge of detecting and removing installed threats. Proceedings of the 16th Virus Bulletin International Conference, pp.61–64. 2006.

[2] Brosch, T.; Morgenstern, M. Malware removal – beyond content and context scanning. Proceedings of the 17th Virus Bulletin International Conference, pp.211–217. 2007.

[3] Schouwenberg, R. The (correct) detection of light grey software. Proceedings of the 16th Virus Bulletin International Conference, pp.52–55. 2006.

[4] Pursche, O.; Otten, M. Abserviren (to polish off malware). ComputerBild 06/2008, pp.60–67. http://www.computerbild.de/.