

Trojan Defence: A Forensic View Part II

The Trojan defence; “I didn’t do it, someone else did” – myth or reality? This two part article investigates the fascinating area of Trojan & network forensics and puts forward a set of processes to aid forensic practitioners in this complex and difficult area. Part I examined the Trojan defence, how Trojan horses are constructed and considered the collection of volatile data. PartII takes this further by investigating some of the forensic artefacts and evidence that may be found by a forensic practitioner and considers how to piece together the evidence to either support or refute a Trojan defence.

In the first part of this article the Trojan defence was described, and terms such as “Trojan” and “Backdoor” clarified. Amongst other areas, the authors examined the process of Trojan horse program creation, ways in which these may be configured to avoid detection and the importance of network evidence.

Part I also contained references to the potential evidential value of collecting data resident in volatile memory. Ideally, this type of information should be obtained whenever possible as it could significantly aid an ‘offline’ analysis. When conducting an offline analysis of a system, a forensic investigator will often discover encrypted or password protected files that could potentially contain additional evidence. A forensic analyst would traditionally utilise specialist password cracking applications to gain access to these files, with varying degrees of success; recovered passwords can prove to be extremely useful.

Some cases may require the analyst to link evidence to an individual (in the case of a shared system for example). The linking of evidence to an individual is not always an easy task; the use of passwords unique or particular to an individual can help demonstrate this link. Again, recovered passwords can help.

None of this is new and an experienced analyst would most likely examine a system for potential passwords; areas for examination may include the registry, configuration files and log files, amongst others. Many practitioners will also examine the virtual memory (such as the Windows page file) as this can contain a wealth of information, including passwords that were cached in memory and then swapped to disk. If virtual memory can yield so much useful information, then certainly a full memory dump could prove to be something of a gold mine.

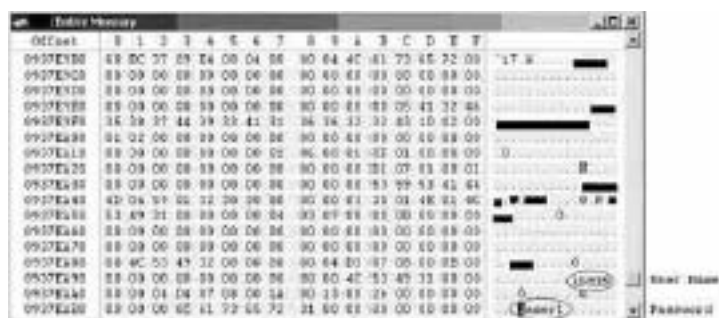


Figure 1 - Username and password in memory¹

Many applications will store passwords in memory or at least cache them for a period of time, including malicious backdoors. During the course of normal system usage, some of the memory will be swapped to disk. However, not all memory is swapped to disk; so where volatile data is not seized before the system is unplugged, a lot of potentially useful information could be lost. The value of the volatile information is not simply limited to password recovery, details about which network connections were active and which processes were loaded at the time the system is unplugged and seized can provide the forensic analyst with some additional useful insight.

For example, let us consider a system that has been seized as evidence as it has allegedly been used to access illegal material. The suspect claims that the images were probably placed on the system through a Trojan horse and backdoor combination.

As noted in part I, a Trojan is merely a delivery mechanism and isn't generally used to place incriminating data on a victim system. What is of more significance is the payload, which is what the Trojan horse program brings with it. As a backdoor is a common payload found within a Trojan, the forensic analyst should investigate this possibility.

A simple first step would be an anti-virus scan against the forensic copy of the seized disk. As pointed out in the part I of this article, although a backdoor antivirus signature may be detected on a disk image, this doesn't necessarily mean that the backdoor has been activated. It is entirely possible that the Trojan containing the backdoor has not yet been "run" and thus the backdoor has not actually been deployed. Determining whether or not a backdoor is merely detected on a system or is indeed active can sometimes be rather difficult to ascertain, particularly via static analysis.

Considerer the scenario where the anti-virus software doesn't identify a backdoor (confusingly also referred to as a Trojan by most AV). Does it mean that a backdoor is not installed on the system and that the suspect's claims are erroneous? Not necessarily, although one might then wonder what led the suspect to believe this to be the case! As AV software works by identifying 'known' signatures, it is continually playing catch-up with malicious software (malware) authors so that a new virus, Trojan or back door written fresh today may go undetected for a long time.

To help dispel this backdoor claim it is necessary to show that no backdoor ports were open at the time the incriminating evidence was created. This is not always easy to do and is often difficult to show with just an offline static analysis of a disk. Corroborating evidence supporting this fact may of course be found in personal firewall logs and external enforcement point logs (devices like firewalls or proxy servers), but not all systems maintain adequate logs or are protected by an external firewall or a personal firewall. This is where the volatile data and network captures may be useful.

Methodology

The volatile data should contain a list of running processes just prior to the system being unplugged, along with a list of listening and active ports. This is where our research on backdoor detection comes into play. Where the right tools have been used, the list should show which ports are associated with which process. By working through the list it is possible to eliminate all of the valid processes; the executables for each valid process should match a known good MD5 hash for example, making exclusion simple. If there are no processes left after eliminating the known good processes, then we can be fairly certain that a backdoor did not exist (or at least was not in operation) on the machine at the time the system was seized.

In the event that there are processes that could not be eliminated, it is necessary to analyse these processes further. Malware analysis and disassembling are beyond the scope of this article; there are however a few other techniques that can be used to help determine whether or not these remaining processes may be malicious.



In order for these processes to start, they require some kind of trigger mechanism – such as an entry in a start-up registry key, a script, a service or a modified system executable. The location of the start-up mechanism may provide a clue as to the nature of the process. For example, a modified system executable or an executable that is being started from the incorrect path requires careful scrutiny as the process is attempting to deceive the end-user. Similarly, unusual services or service names that aren't usually found on a system and are linked to open ports, for example the Windows Update Service or the Windows DNS Daemon, may indicate the presence of a backdoor. The executables for these services should be examined in greater detail.

Analysing the strings contained within the suspect executable files can sometimes yield useful information, such as ports, registry keys, passwords, notification options and other files that may be related to the program. Strings aren't always easily obtainable because they are hidden as a result of encryption and or compression from the packers (covered in part I). It is possible to unpack some executables if the correct unpacker can be found; after which strings analysis may be possible.

Running a suspect binary on a clean analysis box can also prove to be a fruitful exercise. By taking a before and after snapshot of the system, it is possible to determine changes that have been made to various parts of the system –providing useful information such as registry locations, created files and process names. The discovered artefacts can then be located on the suspect system. It is important to remember that malware can have multiple methods of hiding which means that it may be necessary to repeat the testing several times to get a clear understanding of what actions are taken by the malware. It is also worth noting that some malware is activated by a trigger – such as a reboot or a particular time or date. At 7Safe, we have found the use of virtual machines such as VMWare² to be most beneficial at this stage.

Network Analysis

Analysis of any captured network traffic obtained as part of the original evidence seizure may yield further clues. Network capture tools, or sniffers, are used to obtain this data. A free open source tool which can be used for sniffing is Ethereal³, which runs on all popular computing platforms including UNIX, Linux, and Windows. If the evidence in question was created during the time the network capture took place the network capture will clearly show which ports and protocols were used. If these ports are related to the suspect processes, then it is possible that a backdoor is in use; however, if the ports used relate to known good processes, then the backdoor claim may be refuted. It is also worth capturing traffic generated on the clean system when the suspect binary is introduced as this could provide clues as to the notification method used or other systems that may be compromised or otherwise involved. For example, some backdoors will use IRC channels for notification and a capture of the traffic would reveal the channel name and any password used for gaining access to the channel.



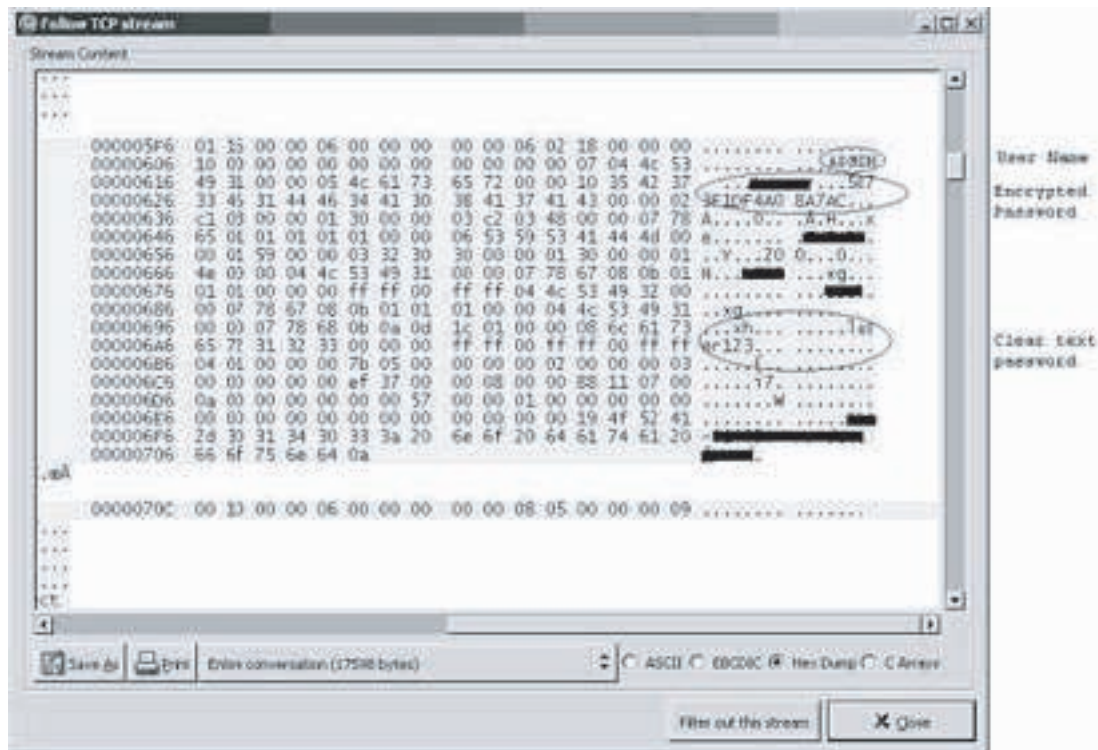


Figure 2 - Network traffic capture⁴

Internet search engines (especially Google⁵) can be excellent tools for hunting down malicious processes, particularly when armed with details such as the start-up mechanism, strings, process names port numbers and typical network traffic patterns. Establishing a good relationship with the anti-virus vendors is another worthwhile option, as they have the skills and resources required to analyse (potentially) malicious processes and report on the capabilities of the process.

Hide and seek

Open ports & active processes relating to malicious software can be hidden to the extent that using standard installed tools like netstat and Windows Task Manager (see figure below) will not reveal them during a live analysis. Imagine the implications if such evidence was not found in the course of an investigation yet a Trojan and backdoor combination (with no equivalent AV signature) had actually been used to implant illegal images on a victim machine. Of course, the reason that such evidence is hidden is that the malware author doesn't want the victim to find out what's happening during the period of the system compromise. Thankfully, specific tools such as those used during the 'Forensic Artefacts: Hands-On⁶' training course present the forensic investigator with an accurate analysis.

4 www.securitydocs.com/library/2885

5 www.google.com

6 www.7safe.com/computer_forensics_training_course.htm



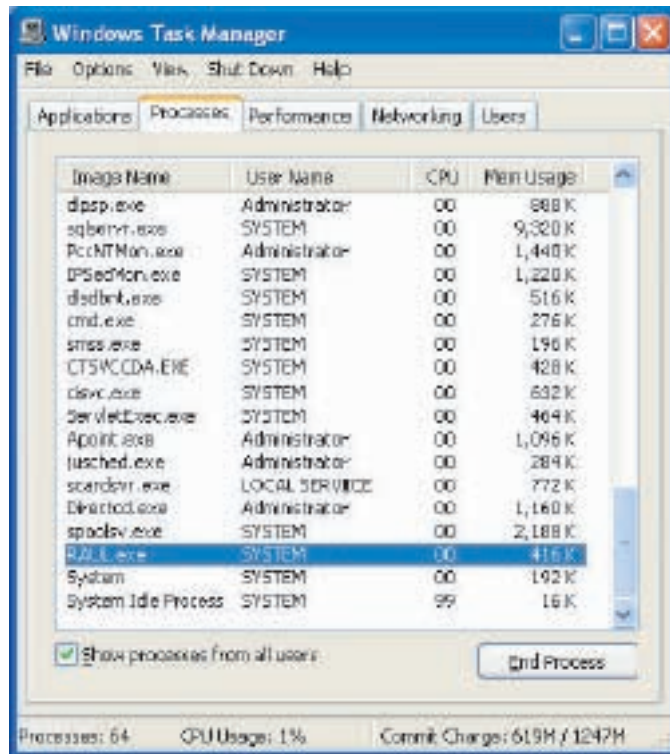


Figure 3: Task Manager – Don't be fooled

Conclusion

When the techniques that have been described in this article are combined with traditional host based computer forensics, it is clear that the forensic analyst is in a much stronger position to be able to prove or disprove a backdoor claim.

This article has explored various technical methods that can be used to help ascertain whether or not a backdoor was involved in the creation of evidence. While these techniques may help prove or refute the backdoor claim, it is important that we remember to consider the evidence in its entirety. A malicious process may be found along with an active port, but the personal firewall (if present) may have an entry that restricts access to that port from the Internet. Even if a backdoor is found and is accessible from the Internet, we have yet to see a backdoor that is capable of generating evidence, burning it in organised folders to CDs, placing printed colour labels on the CDs and filing these away in the attic!

About the authors: Byrne Ghalvalas (CSTP, CFIA, GCFA) is an experienced security practitioner and instructor in areas including computer forensic investigation and penetration testing. Alan Phillips (pictured) (BBus., CSTP, CFIA, MBCS) is a registered BCS Security practitioner and contributing author of IT security training courses, including Forensic Artefacts: Hands-On & the Hacking Insight series. Byrne and Alan work for 7Safe Ltd – an independent Information Security consultancy delivering an innovative portfolio of services including; Forensic Investigation, BS7799 Consulting, Penetration Testing & Information Security Training.

