A COLLABORATIVE DISTRIBUTED VIRTUAL PLATFORM FOR FORENSIC ANALYSIS OF MALICIOUS CODE

Leonard Shand[1] and Theodore Tryfonas[2]

[1]Independent e-commerce and e-forensics specialist
Monmouthshire
United Kingdom
leonard@lenshand.net

[2]Lecturer, Faculty of Engineering
University of Bristol
United Kingdom

theodore.tryfonas@gmail.com

ABSTRACT

Malicious software is prevalent in many forms with the potential for many types of malware to be downloaded while browsing the Internet using an unprotected system. The potential impact can be irreparable harm to a computer file system or even place a person in a situation where they could be charged for a criminal act, if the perpetrator assumes control of their system. Understanding contemporary forms of malware is crucial in order to prepare better defences against it as well as investigate related incidents and claims. Therefore forensic analysis of specific malware, requires specialised tools and techniques and is of significant importance for information security professionals.

In an effort to facilitate the process of forensic analysis of malicious and hostile code we intend to develop a system whereby specific malware can be identified, classified and the malware and detailed forensic analysis stored in a searchable database. The research results would assist computer forensics expert witnesses and infosecurity specialists, to determine the potential role, and impact on a case of certain malware types found to be present on a computer under examination.

To this end, we first research on different types of malware and obtain a selection of malware samples as a specimen to investigate. We create an environment containing suitable investigative tools with which to analyse malware and devise a virtual testing utility platform (containing networking settings, software etc.) to conduct examinations. Experts can use the virtual infrastructure provided to analyse malware and then log their analysis results, notes and experiences in a bespoke on-line collaborative web accessed database. In there experts can log their findings and further produce analytical aids including the behavioural profile of the malware inspected, and potentially be others analysing the same types of malicious code.

KEY WORDS

Malware analysis, computer forensic examination, virtual platform, knowledge management system

A COLLABORATIVE DISTRIBUTED VIRTUAL PLATFORM FOR FORENSIC ANALYSIS OF MALICIOUS CODE

1 INTRODUCTION

Malware has traditionally been associated with viruses and Trojans and is a software program designed to disrupt computer systems. Of late, however, malware has become more insidious and stealth-like and the means of detection have become more and more difficult. Programmers of malware are becoming increasingly adept at modifying malware and the proliferation of source code on the Internet has enabled them to inspect the operation of the malware in much greater detail [1].

When a forensic investigator investigates a computer system that has been used in a suspected criminal activity, he needs to determine, amongst other criteria, whether such activity was the result of the perpetrator's actions or whether the computer system and/or any related software could have been instrumental in causing the offence.

The purpose of the research presented in this paper is to assist the forensic investigator to be able to see at a glance whether there are indications/evidence that malware could be the cause of the suspected crime or whether the perpetrator is responsible. The overall project aims at exploring the following:

- The current status regarding malware and previous research into its analysis.
- The information required for determining whether malware could cause a criminal act to be perpetrated.
- Safe methods and environments for malware analysis.
- Investigating the initial requirements of a prototype tool for collaborative malware analysis and develop a web-based database application for it.

Our system is collaborative in the following sense: we assume the involvement of three distinct roles in malware analysis. That would be the anti-virus researcher/analyst, a person of highly technical orientation and skills, at least in the areas of operating systems and networking. Also, a computer forensics examiner, a person primarily competent in the use of forensic analysis applications such as EnCase and AccessData, as well as aware of current and relevant computer incident related legislation. Finaly, a systems administrator who is responsible for maintaining the web application. Facilitating the colaboration between the three roles and allowing for seamless exchange of information between them, and particularly towards the forensic examiner, is deemed essential. Of course the three roles mentioned here need not necessarily be distinctive individuals, but a malware analyst can also be a system administrator or forensic examiner and vice versa.

The paper then is structured as follows. Section two reviews the types of malware that a forensic examiner may come across during a computer examination as well as related concepts of malware impacts and protection. The third part describes the creation of a safe environment for the collection and analysis of malware. Specialist software was identified which could be used to analyse the malware in the safe environment and choices as to the most appropriate set of tools will be chosen in this report that will best suit the testing and analysis of the malware. Also in part three we discuss the design and implementation of a web-based application, the implementation of the safe environment and the analysis of malware. Part four examines the potential of this system in use and how it could assist a group of forensic investigators in collaborating and sharing knowledge on malware incidents. Finally we conclude the paper identifying areas for further development.

2 MALICIOUS CODE AND PROTECTION

2.1 Types, Impact and Controls

Under UK law, having malware in your possession is not necessarily an offence, but the dissemination of that material in any form is (the UK Computer Misuse Act, section 3). Using the malware in a concerted effort to do damage is a crime and as such is punishable under one or more of the pieces of legislation mentioned above. Apparently, knowing that your computer could possibly be used, or has been used in a malware threat could make you liable. It appears that by not properly protecting your computer

with all manner of anti-malware software, you could possibly be prosecuted criminally for recklessness and civilly for negligence. While this has never been tested in a court of law, the possibility is there.

The computer forensic investigator needs to determine what, if any, information on the system can be used as evidence for or against a user under investigation. The search for and inspection of malware is fraught with difficulty. There are three main categories of malware that any computer user could encounter in their life. The severity of malware is regarded as one of benign to destructive or dangerous, depending on the role of the malware. Malware can infect a system or the files on that system.

Viruses are the most common form of malware. A virus is a program which infects a computer system by installing themselves on it and then replicating. Most known viruses are caught by up to date antivirus software and are not as much of a threat as they used to be [2]. There are three distinct types of virus:

- File infectors: Two types of file infectors have been identified. The first is a virus that infects and attaches itself to files and then executes each time the file is run or opened. The second type does not change the file in any way, but alters the route in which a file is opened. Performance is variable as some viruses will actually impact on the performance of the entire system quite considerably.
- Boot sector: Boot sector viruses infect the boot sector of discs. They then replicate when booted from that disc. This type of virus may have no noticeable impact on the performance of the system.
- Macro: At present this is the most common form of virus [3]. Macro viruses use the program's own macro programming language (Visual Basic for Applications: VBA) to allow execution. The infection takes place by writing itself into the normal.dot file. Any time any Microsoft Office application is run, the virus is spread by infecting the document that is being created or read. Due to other applications' ability to open most formats of Office files, the virus can be spread to other computer platforms too. This is the case with Macintosh, DEC, Linux and Windows.

Worms are the oldest form of malware, even though they did not start out that way. The first implementation of a worm was by John F Shock and

Jon A Hupp, researchers at Xerox PARC (Palo Alto Research Center) in 1978. Its purpose was to search out other computer hosts, find idle processors on the network and assign them tasks, sharing the processing load, and so improving the 'CPU cycle use efficiency' across an entire network and then copy itself and self destruct after a programmed interval [4].

Worms travel through networks and the Internet by various means. They do not attach themselves to other programs. Worms almost always cause harm, even if it is only consuming bandwidth. While many worms are designed to spread, many have payloads. A payload is code designed to do more than just spread. Payloads can have the following actions: delete files on a host system, encrypt files in a cryptoviral extortion attack, send documents via e-mail, and install a backdoor in the infected computer to allow the creation of a "zombie" machine under control of the worm author.

When these machines are networked they are often referred to as botnets and are very commonly used by spam senders for sending junk email or to cloak their website's address. Backdoors can be exploited by other malware, including worms. While many worms are malevolent, there are a few that are used for good intent. One such example is those used to update or patch systems or applications such as Windows [5]. The biggest issue with worms is that they do their task without the explicit consent of the user and could generate considerable network traffic.

Trojan horses derive their name from Homer’s reference to the wooden horse of the Greeks in Troy. Its purpose is to infect a system under the disguise of a useful or required application or file. Most Trojans have a payload and are perpetrated from elsewhere to gain access to a system in order to gain full control of it as well as giving itself access to files and data on it. Trojans usually consist of two parts: a client and a server.

When the server is installed, it allows the remote client software to send commands to the server. This notifies the remote attacker, who can then upload and download files, can delete and create files and folders and can control most of the machine. Most Trojans will notify the remote attacker that the server is running. This action is mostly done via IRC (Internet Relay Chat) [6]. The Trojan infection process is explained in Figure 1.

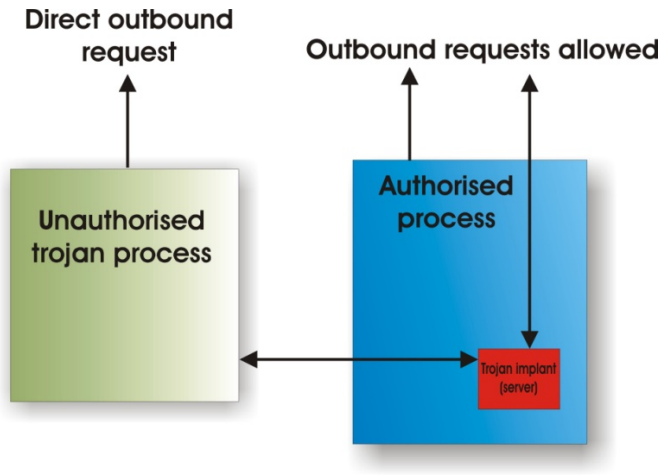

Figure 1 Trojan infection

Malware affects the system in many ways. Most notably are changes in system behaviour which include:

- attempt to connect to websites
- open file shares
- send email
- open other communication channels with remote systems
- launching new services and/or opening listening ports on a system that wait for remote commands
- modifying start-up settings to ensure that it will always run each time the system reboots
- modifying registry setting in Windows

In many cases the user is not even aware that malware exists on the system. It has been noted that even with antivirus software, a firewall and anti-spyware installed, malware can exist on the system without detection [7].

While a lot of users are computer literate enough to know that prevention is better than cure, many users still do not adequately protect their systems. A survey conducted by Schwartz Communications, Inc.

indicates that although many users have antivirus software installed, this software is not updated to an acceptable standard. And while it would be safe to assume that antivirus software will protect a user from most known viruses, for Trojans and spyware, this alone is not sufficient. An adequate firewall and dedicated anti-spyware software is also required. Of course, while this will provide protection for the user, it is not the panacea that most users will perceive as.

According to CERT [8] the following steps should be taken before connecting a new computer to any network:

- Connect the new computer behind a network (hardware-based) firewall or firewall router
- Turn on the software firewall included with the computer, if available
- Disable nonessential services, such as file and print sharing
- Download and install software patches as needed

While the above holds true, the user should attempt to download and install patches or service packs for the relevant operating system, before connecting it to the network by making use of an existing networked computer system. In this way, the user is assured that the system is up to date and that only software patches then need to be updated.

Anti-virus packages make use of various methods to detect malware. According to Aycock [9] there are three main tasks an anti-virus package should perform:

- Detection
- Identification
- Disinfection

Detection of malware is usually by its signature. This signature may be able to be in the form of a combination of bits or it could be a complete cryptographic payload. It is important that the anti-virus package correctly identify malware and to produce as few false positives as possible. Once the malware has been identified, the user should be presented with an option of whether the malware needs inoculation, quarantine or deletion.

2.2 The Need for Collaborative Forensic Analysis of Malware

In the case of Regina v Caffrey, Aaron Caffrey was acquitted only after a lengthy and costly court case. The Register [10] reported: “A forensic

examination of Caffrey's PC found attack tools but no trace of Trojan infection".

From the discussion on malware and of related court cases, it becomes apparent that, a tool that could assist forensic examiners in the task of identifying, analysing and reporting on malware found on a suspect machine, is deemed to be extremely useful. The application proposed here is a collaborative system in the form of a web-based database which will allow the forensic examiner to look up several aspects of malware, including:

- Name
- Aliases
- File names associated with the malware
- Exploit/means of attack
- Action of the malware
- Which parts of specific legislation could be used for an arrest
- Any registry keys potentially affected

With this information in hand, the examiner could be assisted in determining if the user knowingly had malware on the system at the inferred time or whether the malware came to be on the machine by other devious means.

3 SYSTEM DESIGN SPECIFICATIONS AND IMPLEMENTATION DETAILS

3.1 Virtual Analysis Environment

As malware can be detrimental in its operation, it is important that a safe method be devised to capture, analyse and disassemble malware. In this chapter we will consider the design specifications for our proposed tool based on the nature of malware and the requirements of collaboration, as discussed earlier.

While it would be easy to set up a computer system without any connectivity to the Internet or an internal network, many malware types attempt to dial out, scavenge the network or use subversive methods of informing a server or remote attacker. It is therefore necessary to implement a system that would have a form of network connectivity. Virtual environments have become quite widespread in the past decade. A virtual environment is one that runs on a computer system and allows the user to

create virtual machines within this environment. The virtual machine software allows a single computer to allow many more operating systems to be run in the same environment.

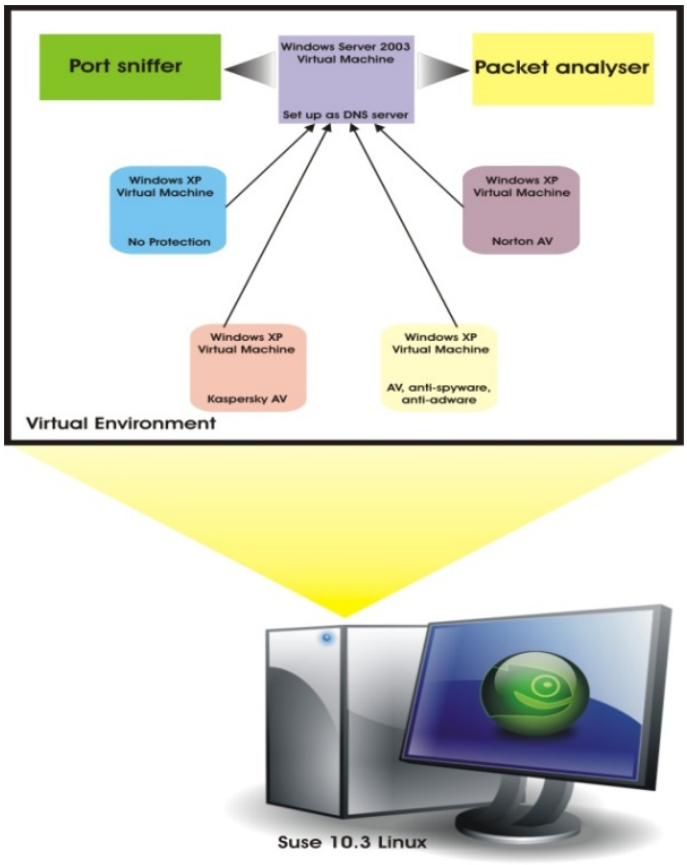

Figure 2 An envisaged virtual system

Using virtualisation, a number of technologies could be implemented within a single machine environment as detailed in Figure 2. By making use of a virtual machine system, licensing issues with proprietary software needs to be taken into account. While the software is running within a virtual environment the software must be appropriately licensed as should the operating system of the base machine. Each installation essentially needs its own license. In the case of Figure 2, The Microsoft Windows 2003 server

would require a license and its minimum 5 user connection license and the four Windows XP workstations would each require their own license.

Various software packages are required to analyse the malware. The software can be categorised as follows:

- System software – the base operating system
- Test bed software – Vmware
- Test bed system software – Windows Server 2003 and Windows XP Professional SP2
- Network analysis tools
- File analysis and disassembly tools
- Registry tools

There is a plethora of analysis tools that a forensic analyst could use for testing for the presence of malware, examining and disassembling malware and analysing malware behaviour. As this project is concerned with determining how the malware could have infested itself on a machine and what actions the malware will perform, only specific tools will be required.

Network analysis tools have been on the market as long as we have had networks. Most network administrators use tools every day to analyse their company's network. These tools would include, but are not limited to:

- Nessus – a vulnerability scanner (www.nessus.org)
- nMap – a utility for network exploration (http://insecure.org/nmap/)
- PRTG traffic grapher – monitors bandwidth usage (http://www.paessler.com/prtg)
- Ethereal – Now Wireshark - a protocol analyser (http://www.ethereal.com/)

As all of these programs are either free or can be downloaded and fully function as a trial, most of them will be used in the analysis to determine if the malware is making use of the network in any way.

File analysis tools are used to determine the code of the file. It is necessary to determine if the source code could be viewed in any form to

investigate how the malware operates. For this purpose OllyDBG can be used (available from http://www.ollydbg.de/ as shareware). Registration is free and once the registration form has been emailed, the program may be used freely. The IDA Pro disassembler and debugger has a graphical based interface and focuses on fundamental analysis of files (http://www.datarescue.com/idabase/idadown.htm).

Also, the Windows registry is a complex directory which stores information regarding the Windows operating system and installed applications. Many malwares update the registry in some fashion and a means of determining what has been changed and by which application is key. While regedit.exe could be used for searching the registry and manually changing keys, it is very difficult to determine changes. For this a snapshot of the registry is needed. A program that claims to be able to take snapshots of the registry is Registry Workshop from www.torchsoft.com.

All the above tools can be used to some extent for each piece of malware under analysis. As an innocent person could be wrongfully convicted by false information provided by the web-based application, it is necessary to determine exactly what the malware is capable of and how it operates. Therefore all means necessary to determine the modus operandi of the malware and possibly its origins are vital.

3.2 Content Management System

The overall system enables *malware analysts* to record their findings and enter those into a database. *Computer examiners* can then search the database for relevant information in a quick and efficient manner. The application allows an analyst to quickly search for an item and then attempt to identify it on the suspect system (within the safe environment), as one of its goals is to record malware, its associated behaviours and actions.

Reviewing the results of analysis of malware is crucial to the investigator in a criminal case. Not only does the information have to be current, it needs to be accurate and must also allow any one with some computer knowledge to be able to find the information – e.g. an investigative officer not necessarily expert in computing. The application allows the investigator to use various methods by which to search for specific items such as:

- Type of malware
- Name of malware
- Possible crime committed
- Registry key
- Infection method

Aside from the functional and non-functional requirements, it is critical to develop an application that is appealing and easy to use by the intended users. In order to achieve this, the system stores and displays all malware data in an appealing and usable manner for the user. This is achieved by attempting to make the system as easy to learn and use as possible, but also by providing the most possible detail in a single screen.

In the application the following tasks are provided for: record malware and its attributes, record malware behaviour and infection methods, record Acts of Parliament and statutes, record possible crimes, amend all the above, view malware and its attributes, view malware behaviour and infection methods, view Acts of parliament and statutes, view possible crimes and query data held by the system. Creating new records and amending existing records is available only to authorised skilled analysis personnel, while viewing and searching of the records is provided to authorised law enforcement agencies and personnel.

Finally, MySQL is used for the database implementation. The ER diagram of Figure 3 represents diagrammatically the initial structure of the database. Although there are two types of users, namely analysts who would record data into the database and users who would only perform searches, a decision was made to remove unnecessary redundancy from the database design. Therefore a single Users table has been created which has a 'level' for the user. This will allow only persons with the requisite authority to access the data input section of the web application. Figure 3 reflects the simplest structure of the database.

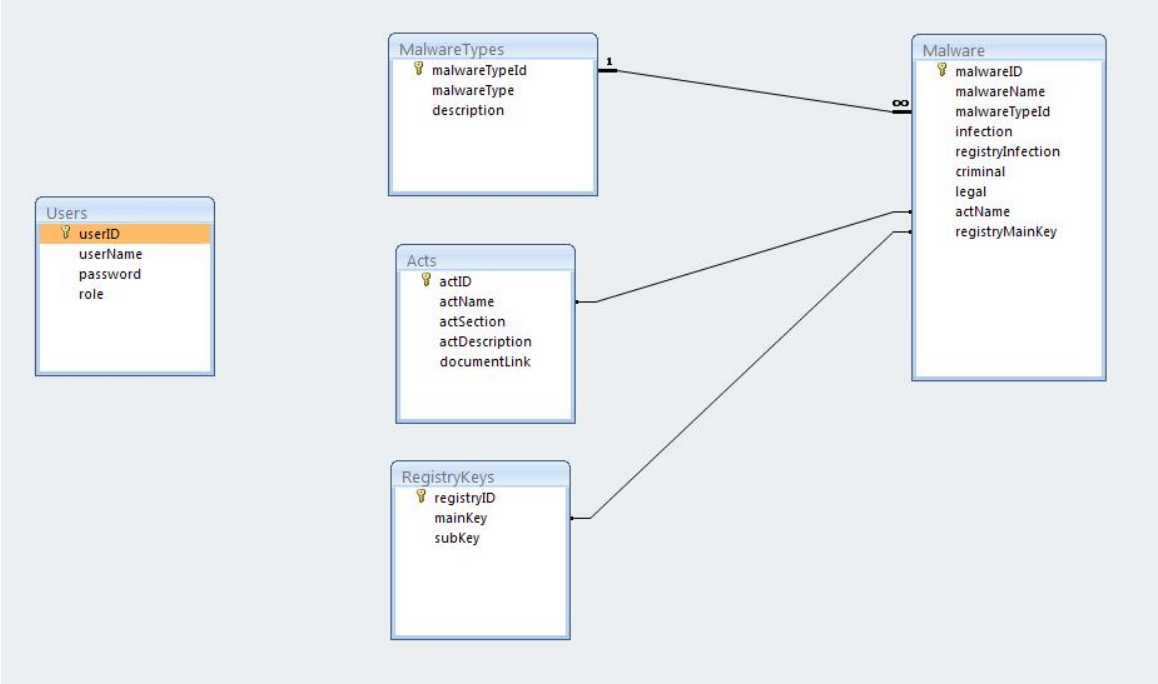

Figure 3 ER diagram for the web application database

4 USING THE SYSTEM PROTOTYPE

Our Malware Analysis Tool (MAT) is a prototype web application built upon a platform that matches the specifications as discussed in the previous part, which allows investigators to search for entries on items of malware (as for example it can be seen in figures 4 and 5) and use this information to assist them in determining whether there is malware present on the suspect machine.

The two key components of this environment are the content management system, enabling the sharing of knowledge, and the secure testing environment. The latter has been tested with various types of malware including Trojans and rootkits and appears to be solid with no leakage out of the environment. In this environment, the malware analyst is at the minute required to analyse the malicious code and subsequently manually input the data into the MAT analysis form which is then posted to the MAT database. Our ultimate goal is to implement a collection of scripts

and macros which automatically record the results as the analysis is taking place in the safe environment. Once the analysis is complete, the form can then be automatically uploaded to the MAT database.

Figure 4 Specific Malware listing

Figure 5 Listing of case under investigation

5 CONCLUSIONS AND FURTHER WORK

Not only is forensic examination of malware a problem for computer users, but also for law enforcement and forensic practitioners [11, 12]. As such, it shows that no sooner has one form of malware been discovered and analysed, then another shows up with distinctly differing properties. What was developed initially as a tool to assist researchers has been used by others for more sinister means. Malware is insidious at its best and performs its task without the users' consent, or claimed to be so [13, 14].

Finding, tracking, capturing and preventing malware infestations on a computer system are akin to war being waged. Therefore, when a crime is committed, it is up to the forensic investigator to search for the quickest means to determine who the guilty party is. Forensics analysts do not have the manpower to spend countless hours analysing and disassembling malware to come to a decision. A platform that would enable collaboration between malware analysts and forensic examiners could assist the latter in determining how the crime took place and whether it would be possible to convict a suspect. The objective would be to reduce the time spent searching for malware and create an easy reference for the investigator. To this end, the Malware Analysis Tool makes accessible all the information required by the investigator in a searchable format. Not only does it perform well as a reference tool for all types of malware, it also speeds up searches for registry keys, malware, affected APIs and a host of other Window's related areas.

As mentioned previously a future improvement to the manual process of analysis of malware and simultaneously inputting the data into the MAT would be to have a program which automatically records the results as the analysis is taking place, extracting the required data as the process continues. Once the analysis is complete, the form could be automatically uploaded to the MAT database. Ideally the format of the data form would also include a XML version, as this information would be accessible in more ways than just a web page.

6 REFERENCES

[1] Griffin, Brad (6 November 2006) An Introduction to Viruses and Malicious Code [online] available from <http://www.securityfocus.com/infocus/1188> [5 October 2007]

[2] Brain, Marshall (n. d.) How computer viruses work [online] available from <http://computer.howstuffworks.com/virus.htm/printable> [17 November 2007]
[3] Griffin, Brad (6 November 2000) An Introduction to Viruses and Malicious Code [online] available from http://www.securityfocus.com/infocus/1188, http://www.securityfocus.com/infocus/1189, http://www.securityfocus.com/infocus/1190 [5 October 2007]
[4] PARC (n. d.) Innovation Milestones [online] available from <http://www.parc.com/about/history/default.html> [18 November 2007]
[5] AntiVirusWorld.com (n. d.) Computer Worm [online] available from <http://www.antivirusworld.com/articles/computer-worm.php> [17 November 2007]
[6] GFi (n. d.) How do Trojans Work? [online] available from <http://kbase.gfi.com/showarticle.asp?id=KBID001671> [17 November 2007]
[7] PrevX (n. d.) MSSPA.EXE [online] available from <http://www.prevx.com/filenames/2190786876915996974-X1/MSSPA.EXE.html> [17 November 2007]
[8] CERT/CC (15 December 2003) Before You Connect a New Computer to the Internet [online] available from <http://www.cert.org/tech_tips/before_you_plug_in.html> [17 November 2007]
[9] Aycock, John (2006) Computer Viruses and Malware. New York: Springer
[10] Leyden, John (17 October 2003) Caffrey acquittal a setback for cybercrime prosecutions [online] available from <http://www.theregister.co.uk/2003/10/17/caffrey_acquittal_a_setback/> [17 November 2007]
[11] Bradbury D (2006), 'The metamorphosis of malware writers', Computers & Security, Volume 25, Number 2, 89-90.
[12] Forte D (2005), 'Spyware: more than a costly annoyance', Network Security, Volume 2005, Issue 12, 8-10.
[13] George E (2003), Case Note 'UK Computer Misuse Act- the Trojan virus defence Regina v Aaron Caffrey, Southwark Crown Court, 17 October 2003', Digital Investigation, Volume 1, Number 2, 89.

[14] Brenner S and Carrier B with Henninger J (2005), 'The Trojan Horse Defense In Cybercrime Cases', Santa Clara Computer and High Technology Law Journal, Vol 21, Issue 1.