Analysis and Detection of Cruising Computer Viruses

A Abimbola, JM Munoz and WJ Buchanan
School of Computing, Napier University, EH10 5DT, Scotland, UK
a.abimbola@myuni.ac.uk, +44 (0)131 455 2769

Abstract

Viral propagation is an important phenomenon of computer viruses. This paper focuses on viral movement and proposes the feasibility of a computer virus which is able to target a specific host, known as cruising. A novel component, based on target profiling of an intended victim, is introduced in a virus framework. This profile allows the virus to cruise to a specific intended target, which differs from normal virus propagation. To test the feasibility of this, a computer virus with an embedded target profile was designed, and experiments were conducted, which were compared with other 15 normal computer viruses. These experiments show that the cruising virus is able to target an intended victim, and induces a reduced overhead on Microsoft Outlook than other tested viruses, and consumes less network bandwidth. Finally, a method of overcoming the virus is presented. This uses system calls as application wrappers.

1. Introduction

Computer viruses remain a significant threat to modern networked computers systems. Despite the best efforts of those who develop anti-virus systems, new computer viruses, such as Microsoft Blaster (Engene, 2003) and others that implement hybrid exploitation techniques (ISS, 2002) are not dealt with by present anti-virus systems. In addition, the rate at which a computer virus can spread has risen dramatically with the increase in connectivity and also aided by the ease of accessing computer virus production toolkits (Markus, 2004).

Traditional anti-virus techniques typically focus on detecting static signatures of computer viruses. Whilst these techniques are effective, they do not address the dynamic nature of a computer virus infection within the context of the underlying system. For instance, polymorphic computer viruses (Harley, 2001) alter their instruction codes either by substitution or encryption methods to replicate a new computer viral instance.

A virus can be defined as a piece of code with two structural subroutines (Hoffman, 1990). One has the capability to reproduce, and the other has the ability to transfer replicated instances to other remote host. In addition, there is the payload or malicious act that may await a set of predetermined circumstances before being triggered.

This paper, introduces a novel subroutine into the computer virus framework called a *target profile*. This profile enables a computer virus propagate from one network terrain to another seeking a specific intended host. The target profile uses an algorithm consisting of pre-programmed logic conditions to seek-out suitable email addresses and specific target characteristics to determine if the current host is the target host, or otherwise propagates to the next suitable host. The notion of computer viruses being able to propagate to a specific intended host have been dismissed by the virus research community (Yang. S, 2004), as computer viruses *in the wild* have not been able to discriminate between friend and foe, and, as a result, are more suitable for mass terrorism. This idea has now been implemented in this paper by designing and implementing a computer virus with a target profile subroutine and experimentally comparing it with eight other computer viruses. An analysis is presented which investigates the feasibility of a computer virus propagating to a specific intended host or cruise. Along with this the bandwidth consumption of computer virus using the profile, or not, has been determine, along with the processing overhead on Microsoft Outlook.

2. Background

Computer virus movement can be classified into *self* and *non-self*. Non-self movement can be further divided into delivery and duplication, where delivery involves the storage media like floppy disk and duplication involves spreading medium like emailing, downloading, uploading just to name a few. This paper focuses on self-movement and its features, as these explain the propagation of computer viruses over varied network terrains. A common property of self-movement exhibited by most computer viruses is wandering, which is the random movement of computer viruses without a specific intended host. A cruising virus is a recent evolution of self-movement and is the most efficient movement from a remote source to a defined destination along the best path. Other definitions of cruising includes traits like the ability to target dynamic host and finite or infinite propagation.

There has been an increase research in computer viral in using computer viruses as direct weapons information warfare (Yang, 2004). A key drawback to these initiatives is that computer viruses have mainly wandering properties. As a result, computer viruses do not discriminate between friend and foe in executing their payload. This non-discriminatory issue, or lack of cruise-ability, makes computer viruses more suited as a terrorist weapon (Yang.S, 2004; Bontchev, 2004).

3. Motivation

Although it is important to understand the technology of computer virus, in order to understand the nature of the threat, it is also important to understand the motivation of those that launch the attacks. The following are examples of motivations:

- **Pride and power**. Some attackers are motivated by a desire to acquire (limited) power, and to show off their knowledge and ability to inflict harm on others (Honey, 2004).
- **Commercial advantage**. Since most developed countries have grown heavily dependent on computers for their day-to-day operation, international companies or organised crime members may participate in this type of attack, with a target range from specific companies to economic infrastructure.
- **Political Protest**. Groups wishing to use the Internet to publicize a specific message and to prevent others from publicizing theirs. As one example, the Yaha malicious software (McWilliams, 2004), was written as a tool of political protest by unknown parties claiming affiliation with Indian causes, to launch a DoS (Denial of Service) attack (Abimbola, 2003 “ISSA”) on Pakistan’s government website.
- **Terrorist**. Terrorist groups could employ computer viruses to meet some of their objectives. Since Internet-connected computers are a first world development, and major multinational concerns rely heavily on desktop machines for their day-to-day operation, payloads of an attacker could be selective to only execute in large network environments.

Different sorts of attackers will desire different payload to directly further their ends, using techniques such as:

- **Opening Backdoors**. Code Red opened a trivial-to use privileged backdoor on victim machines, giving anyone with a WWW browser the ability to execute arbitrary code (Phillips, 2004).
- **Remote DoS**. Code Red, Goner and Yaha have all contained DoS tools, either targeted at specific sites or retargeted under user control.
- **Data Harvesting**. Criminals are sometimes interested in identity theft, and significant subsets of the Blackhat community are involved in harvesting credit card detail (Cardcops, 2004).
- **Data Damage**. There have been many computer viruses and email worms, such as Chernobyl or Klez which contain a time-delay data eraser (Ferrie, 2004; Kasperkylabs, 2004).
- **Hardware damage** (Kasperkylabs, 2004).
- **Espionage** (CERT, 2004).

Typically anti-virus technology is divided into two approaches: a virus specific; and a heuristic/generic. In principle, the virus specific method requires knowledge of the computer viruses before they can be detected. With advances in technology these prerequisites are no longer entirely valid in many of the modern anti-viruses. A heuristic approach attempts to detect the virus by observing attributes or characteristics of all known viruses. For instances, integrity checkers detect viruses by checking for modification in executable files.

4. Related Work

The work on throttling viruses observes that during computer virus propagation, an infected machine will connect to as many different machines as quickly as possible (Williamson, 2002). An uninfected machine has a different behaviour: connections are made at a lower rate, and are locally correlated, where repeat connections to recently accessed machines are likely. The technique developed has two parts:

- **Connection invocation**. This determines whether a connection to a host is new, or not, using a short list of pass connections.
- **Connection rates**. This limits the rate of connections to new host by using a series of timeouts.

These methods are employed using a filter to monitor the network connections, where a sudden increase in the outgoing connections of a host can indicate an infection. This method, though, will not be able to detect a computer virus with a target profile, as the connection rate will be limited to few suitable host based on the pre-program logic conditions of the target profile.

Portable Executable Analysis Toolkit (PEAT) (Weber, 2002), is one of the most sophisticated tools that can determine whether malicious code has been inserted into an application after compilation. These tools rely on a structural feature of executable that is likely to indicate the presence of inserted malicious code. The underlying premise is that typical application programs are compiled into one binary, homogenous file from beginning to end with respect to certain structure features. Any disruption of this homogeneity is a strong indicator that the binary code has been tampered with. Experiments using PEAT to detect _BackOrifice 2000_ produces good results, although once the attacker knows the criteria of the logic, they can adapt the attack to circumvent the detection. The probability of actually knowing and detecting the files that have been infected in the case of a computer virus embedded with a target profile is low, as the computer virus only executes its payload at the target or intended host and infects only a handful of email addresses suitable for propagation to the target host. As a result, PEAT's methodology will have little material evidence to apply its premises on.

Research work being carried on MET - Malicious Email Tracking - (Stolfo, 2003) is designed to automatically report on the flow behaviour of malicious software delivered via email attachments both at local and global level. The core of MET's operation is a database of statistics about the trajectory of email attachments in and out of network systems, and the integration of these statistics across networks to present global view of the spread of malicious software. Similar research is MEF - malicious email filter (Schultz, 2001), which filters malicious attachments from emails using detection models obtained from data-mining over known malicious executable. This allows the detection of previously unseen, malicious attachments. The filter also allows for the automatic propagation of detection models from a central server and allows for monitoring and measuring of spread of malicious attachments. Both MET and MEF will be unable to detect a profile virus, as they are based around the detection of increases in connection rates.

5. Analysis of Target Profile

A target profile can be designed using Windows Script Host (WSH) (Esposito. D, 1999), as it provides access to most part of Microsoft Windows on the intended platform. The target profile is divided into two subroutines:

- **Profile search**. This includes script codes that search the Windows operating system for traits that validate that the current host is the target host. Searches include files and folders with varied names permutation like _targetnamecv_ in the main directory, opening and reading text files, reading registry entries, searching installed application configuration/installation settings in the registry, and also searching cookie details – as the main directory for likely website visited by the target and keywords that inform us of the user of the host. In using WSH, the following code samples checks if certain files and folders exists, opens and reads text files and registry content.

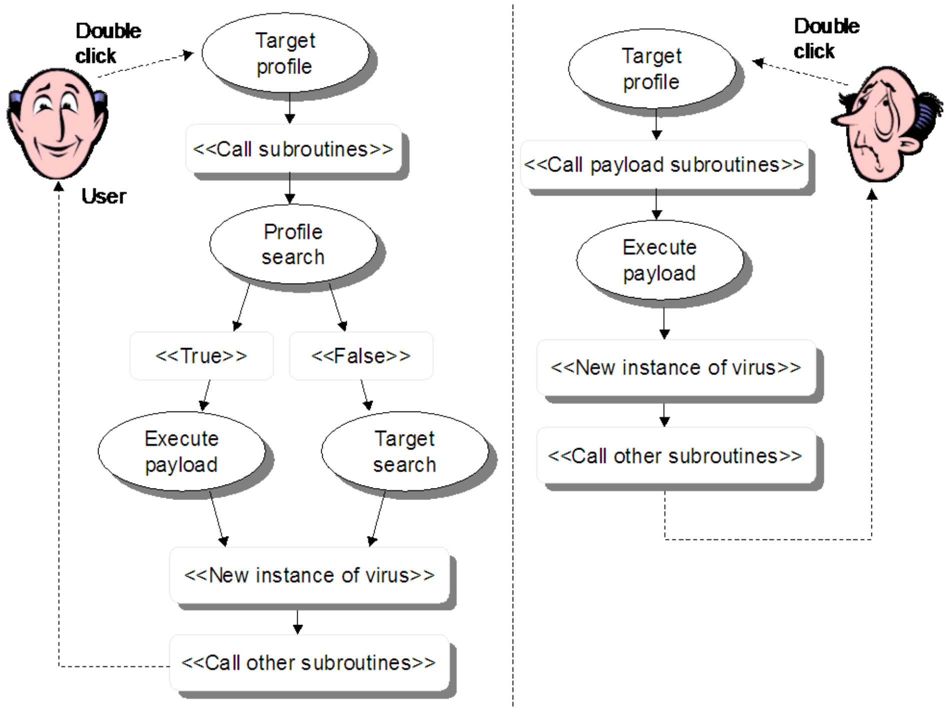
```
Dim fso, RegRead, Ts, Str
Set fso= createObject("Script.FileSystemObject")
Ts = fso.OpenTextFile(Filename, ForReading,
   false, FormatASCII)
RegRead= "HKEY_CURRENT_USER\….."
Set shell= CreateObject("Wscript.Shell")
Readregistry=shell.Regread("RegRead")
FolderExits("targetnamecv") and
   FileExits("targetnamecv")
```

  The main aim of the profile search routine is to determine if the current host is the intended target host. If it is then it executes the payload, otherwise it performs the target search subroutine.

- **Target search**. This performs a similar function to the one found in most viruses (Figure 1). The main differences are that a target search subroutine searches all folders in MS Outlook client, such as in the Inbox, Sent items, and so on, and seeks a specific sequence of email domain entries to infect, based on its pre-programmed logic conditions. A normal virus will only search the Contact folder for email addresses to infect, and afterwards infect one or more addresses in that folder, indiscriminately. As an example, consider a computer virus embedded with target profile seeking the host email address _targetme@myuni.ac.uk_. This will infect only email addresses with the following sequences: _*@*.*.uk_, _*@*.ac.uk_ and _*@myuni.ac.uk_, where * can be any alphanumerical word (Figure 2).

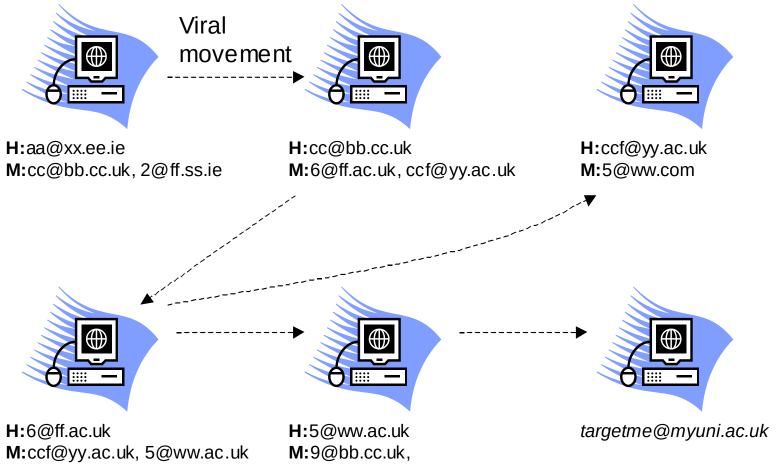
There are other subroutines implemented in our designed computer virus: the execution of a payload routine, an after-submit-delete routine; and an infection-indicator routine that prevents the re-infection of a host. The details of these subroutines can be found in the source code of computer viruses like _Loveletter_ and _Annakournikova_ (Virus, 2004). The main objective of the target search routine is to prevent the infection of email addresses that will not aid the propagation of our designed computer virus towards the intended target host. As a result, it provides a stealthy and minimal list of outgoing network connections to reach the host in a network before executing its payload at the intended target host. A sample of WSH source codes that checks for _*@myuni.ac.uk_ pattern in the Inbox of MS Outlook inbox folder is:

```
Set myunireg= New RegExp
myunireg.pattern= "@myuni.ac.uk"
Set outlook= CreateObject("Outlook.Application")
Set oNS=outlook.GetNamespace("MAPI")
Set oInbox=
   oNS.Session.GetDefaultFolder(o1FolderInbox)
Set inboxmsgs=oInbox.Items
If myunireg.test(inboxmsgs(1).To) Then ………..
```



Target profile virus **Normal virus**

Figure 1: UML Diagram of embedded target profile and Annaakournikova computer virus



H- Host email address, M - Email box content address

Figure 2: Profile targeting using email addresses

6. Experimental Details

The objectives of the experiments were:

- To investigate the feasibility of implementing and testing a computer virus embedded with a target profile.
- To investigate the overhead induced on MS Outlook by a computer virus embedded with a target profile and a normal computer virus.
- To investigate the connection rate and network overhead induced by a computer virus embedded with a target profile and a normal computer virus.

The experiment initially involved launching eight computer viruses (*Annakournikova*, *Tune*, *Loveletter*, *Shakirapics*, *Mawanella*, *Melissa*, *Homepage* and our designed computer virus) in an isolated local area network of over 200 hosts. The experimental tools used were MS Outlook client, Iris Network Monitor (Iris, 2004), MS Task Manager and Winmail Server (Winmail, 2004). The hosts settings are listed in table 1.

Table 1

Host No	*Settings/Configurations/Installed Tools*
1	Computer viruses, Iris Network Monitor and MS Task Manager.
1-198, 200	Windows 98, 128MB of memory and 8GB of disk space.
199	Mail Server

Each MS Outlook client in a network host contained six email addresses of other hosts in their respective folders. As defined in Table 1, Host 1 contained all the computer viruses to be launched, Iris the network analyser and the MS Task Manager. These computer viruses were then launched from host 1, and allowed to infect email addresses, replicate new computer viral strain and then propagate via the isolated local area network to their respective host. These new computer viral strains were then executed in their respective network hosts and the process repeated again. Iris, the network analyser, was used to measure the network traffic induced by the first computer viral strain generated by Host 1 and the results are shown in Figure 3. The overhead induced on Host 1's MS Outlook client by each computer virus launched was measured using MS task manager and the results are shown in Table 2.

Before the actual launch of these computer viruses, their respective payload were commented out from their source code for the following reasons:

- The overhead introduced on MS Outlook client by the computer viruses was to be measured, hence the infection phase and not the payload they were executing,
- Depending on the objectives of the computer virus's writer, the payload of either a normal virus or a computer virus embedded with our target profile could be the same; as a result it becomes a constant.

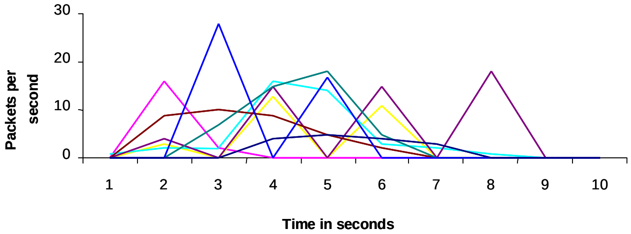


Figure 3: Bandwidth used by viruses

The experiments ignored the effects the protocols ARP, UDP, NETBIOS broadcast, UDP-BOOTPS and the MS Outlook agent on network bandwidth consumption. The following inferences were made from the experiments:

- A computer virus with embedded target profile was able to propagate towards an intended target and execute its payload using our target and profile search subroutine.
- The processes overhead introduced by our designed computer virus is roughly half of that that introduced by other computer viruses used.
- The average and peak bandwidth consumed by the target profile is significantly less than other computer viruses used.

Figure 3 shows the bandwidth consumed by all computer viruses experimented on and proofs that our designed computer virus with a target profile subroutine consumed less bandwidth on average and has the lowest peak network packet per second. Possibly our to its pre-programmed logic conditions that enable it select few email addresses for infection.

Table 2: CPU usage for differing viruses

Virus	*CPU usage (%)*
Annaakournikova	25
Loveletter	11
Tune	10
Shakirapics	9
Mawanella	21
Melissa	21
Homepage	18
Target Profile	8

The experimental results showed that the target profile computer virus is able to propagate to an intended target, and uses utilises less bandwidth and generates less overhead on MS Outlook during the infection phase, than other computer viruses tested.

7. Overcoming the threat

The key threat posed by the target profile virus is its low connection rate and the fact that, it only executes its payload at the intended target host. As a result, anti-virus technology that uses connection rate or prior-knowledge of a computer virus as a detection approach will likely fail. To prevent this threat, an intrusion system based on an active target host (Abimbola.A, 2004 "ESRCP"). This sandboxes an application using system calls as wrappers to protect the application, or host, from harmful processes. To determine malicious system calls, harmful processes are generated using malicious executables and their system calls are analysed for generic trends. These generic trends are then used to create rules sets, for comparison against daily system calls and alerts flagged if matches are detected.

There are several important intuitive advantages in auditing system calls, such as broad coverage and generality - for a given application. It may have wide application to detect a variety of novel malicious processes. However, there are several disadvantages, including performance cost introduced by tracing and analysing system calls, the adaptability and extensibility of wrappers question their practicality, and updates to an application may necessitate a complete retraining of the wrapper's system calls.

8. Conclusions

This paper has show that, contrary to current research, that computer viruses can cruise, that is, propagate to a specific intended target host. The normal understanding is that computer viruses only exhibit wandering property, where they propagate to all hosts on an entire network indiscriminately. In the design the target profile subroutine used email address domains to propagate towards the target mail server. It then uses target specific trends/characteristics in the Windows operating systems to determine if the current host is the target. The experiments have validated this theory by launching eight computer viruses and the target profile virus on an isolated computer network.

9. Reference

Abimbola.A, Shi.Q and Merabti.M (2003) "NetHost-Sensor: A Novel Concept In Intrusion Detection Systems", Eight IEEE International Symposiums on Computers and Communications, pp 232-240.

Abimbola.A, Munoz.J and Buchanan.W (2004), "NetHost-Sensor: Enhancing Intrusion Detection via an Active Target Host", to be submitted to ESORICS 2004.

Abimbola.A (2003) "Denial of Service Attack: What is Going on ?", ISSA Journal, November Issue.

Bontchev.V (2004) "Research & Writings/Future Trends in Virus Writing", Virus Test Center, University of Hamburg.

Cardcops (2004) http://www.cardcops.com.

CERT.CERT (2004) Advisory CA-2001-22 w32/Sircam Malicious Code, http://www.cert.org/advisories/ca-2001-22.html.

Espositor.D (1999), "Windows Script Host", Published by Wrox Press, ISBN:1-861002-65-3.

Eugene.E (2003) "The MSBlasterWorm: Going From Bad To Worse", Network Security, Vol. 2003, Iss No. 10, pp 4-8.

Ferrie.P (2004) "W32//Klez", http://toronto.vitrusbtn.com/magizine.archives/200207/Klez.xml.

Hoffman.L (1990) "Rogue Programs: Viruses, Worms, and Trojan Horses", Van Nostrand Reinhold, New York, NY.

The Honeynet Project (2004) "Know Your Enemy: Motives", http://project.honeynet.org/papers/motives/.

Iris Network Monitor (2004), www.Eeye.com.

ISS (2002) "Response Strategies for Hybrid Threats", www.Itsecurity.com.

Kasperkylabs (2004) "W95/CIH (a.k.a Chernobyl), http://www.vuruslist.com/eng/viruslist.html?id=3204.

Markus.S (2004) "Building Anna Kournikova:An Analysis of the VBSWG Worm Kit", http/www.online.securityfcus.com/infocus/1287.

McWilliams.B (2004) "Yaha Worm Takes out Pakistan Government's Site", http://online.security focus.com /news/501.

Phillips.S (2004) "Dasbistro.com Default.ida Responder",http://sunsite.bilkent.edu.tr/pub/inforsystem /% lph-pweb/default.txt.

Schultz.M et al (), "MEF: Malicious Email Filter A Unix Mail Filter that Detects malicious Windows Executables", USENIX Annual Technical Conference, pp 245-252.

Stolfo.J et al (), "A Behavior-Based Approach To Securing Email Systems", 1st International conference on Applied cryptography and Network Security, pp xx.

Virus source Code (2004), http://www. 62nds .co .nz/ cgi -bin/x/e4015.html.

Weber.M et al (2002), "A Toolkit for Detecting and Analyzing Malicious Software", 18th Annual Computer Security Application Conference, pp 423-431.

Williamson.M (2002), "Throttling Viruses: Restriction Propagation to Defeat Malicious Mobile Code", 18th Annual Computer Security Application Conference, pp 61-68.

Winmail Server (2004), http://www. Magic winmail.net/

Yang.S (2004) "Wandering and Cruise", http://www.tl.infi.net /~wtnewton/ vinfo/bs3.html.

Yang.S (2004) "Movement of Viruses ", http://www.intergate.bc.ca/personal/yang /movement.htm.

Yang.S (2004) "Behaviour Solution", http://www.tl.infi.net /~wtnewton/vinfo/bs3.html.

Yang.S (2004) "Autonomous Mobile Cyber Weapon", http://www.tl.infi.net / ~wtnewton /vinfo/bs3.html.