2006

A Preliminary Investigation into Malware Propagation on Australian ISP Networks using the mwcollect Malware Collector daemon

Craig Valli
Edith Cowan University

A Preliminary Investigation into Malware Propagation on Australian ISP Networks using the mwcollect Malware Collector daemon

C. Valli

School of Computer and Information Science
Edith Cowan University, Australia
E-mail: c.valli@ecu.edu.au

Abstract

This paper describes an initial investigation into the propagation of malicious software (malware) that allows for remote command and control of Internet connected machines using the Windows platform in the Australian ISP address space. The research as conducted utilised the mwcollect daemon which is a low interaction honeypot on the Linux platform, to collect the details about the activity. The program mwcollect works by emulation of vulnerable services on the target platform in this case Windows based computers. There were two collectors within the pilot collection system. The machines were running no other Internet services such as http or mail, and were not used by any person - they were simply connected to the Internet. The machines are located on two separate ISP networks and they both utilised high-speed ADSL connections connected to different segments of the Australian ISP network.

The malware collected is a variety of known exploits that allow for remote execution of code as well as known and unknown shellcodes that enabled attacks. General results from the initial scoping exercise are given and discussed.

Introduction

The use of the Internet for the spreading of viruses and worms is well documented. Propagators of malicious software (malware) are now compromising machines or entire networks to collect and control or 'own' them (Anonymous, 2005; Anonymous. 2006). These owned networks form what is referred to as 'botnets' meaning that the compromised computers (bots) can be controlled in a Master/Slave relationship. Control of the bots is can be achieved by utilizing various network protocols or services such as IRC, or via reverse command shell that allow for point and click command of the bots (Morring, 2005; Telecomworldwire, 2005).

The process of 'ownership' is relatively simple in execution once a potential vulnerable system is found. Ownership of the machine is achieved via an exploit either known such as MS03-26 or MS05-39 (Microsoft, 2003; Microsoft, 2005) or as yet unknown that allows for privilege escalation, remote procedure calls (RPC) and arbitrary execution of code. Malware is then pushed onto machines that allows for the partial or full control of the victim machine providing what is referred to as a 'backdoor'. This malware is often then left in an idle or 'dormant' state until the controller wishes to utilise the machine for their own purposes.

This research was conducted to gauge the level of activity on commodity ADSL based connections in Australian based ISP networks. The platform utilized to examine this was mwcollect, a malware collection daemon which a low interaction honeypot running on Linux Debian machines on two ADSL connections. This paper outlines the basic *modus operandii* of mwcollect, the deployment used and a discussion of the results from the two month study pilot.

ISSN 1445-3312 print/ISSN 1445-3347 online

Mwcollect – *modus operandii*

The program used in this study to collect data was mwcollect, which works by the emulation of vulnerable services on the target platform in this case the Windows operating system on the Intel platform. Mwcollect is a low interaction honeypot and highly modular in design and has four basic module types namely vulnerability, shellcode, download and submission modules.

The first type of module is a vulnerability module that emulates a specific vulnerability on common TCP/IP ports that are attacked for example, 135, 445, 1025. The types of vulnerability that the modules in mwcollect cover are MS03-26 (Microsoft 2003), MS04-11(Microsoft 2004) and MS05-39 (Microsoft 2005). Principally, the mwcollect daemon emulates systems that are vulnerable to the 'backdoor' genre which allows form remote control of the machines/bots. An example of this is the MS03-26 RPC vulnerability (Microsoft 2003). As stated by the programs developers "...the vulnerable service emulation is not very sophisticated, but functional: Often malware does not require an indistinguishable emulation of a real service but an approximation of it." (Freiling and Holz, 2005, p.11).

The second type of module parses shellcode (a small assembly language program which executes a command shell) which is received by one of the vulnerability modules. This module recursively detects XOR decoders, as many shellcodes are encrypted to avoid detection by intrusion detection systems or virus scanners. The unencrypted code is then decoded and searches are made in the extracted code for URLs that have linkage to a site containing the malware for download, for example, http://www.exploitedu.org/exploit.exe. Any detected URLs with these potential payloads are then parsed to the fetch module which then attempts download of the file (*ibid*, p.12). It should be noted that downloads in these cases are not always successful as the offending site is often no longer functional.

Fetch modules are the third type, and these simply download a file specified by a URL that is found as a result of a shellcode decode, or direct response to a vulnerable service or program. These utilize a variety of transfer protocols such as File Transfer Protocol (FTP), Trivial File Transfer Protocol (TFTP) and Hyper Text Transfer Protocol (HTTP).

The final module type is a submission module that handles any successfully downloaded files or shellcodes and writes them to disk or submits to a database. These modules can also submit upstream to another server for aggregation of results across a monitoring network. The modules also create a 'md5 hash' of the file and check for uniqueness before storing or uploading the resulting files or shellcodes. All of the activities perpetrated by the attacker or malware in addition to the storage of malware and new shellcodes are recorded to a text based log file. The following excerpt from a log file indicates the level of detail provided;

```
[2005-12-13 13:37:08] Stream subscription to :135 from 2XX.1XX.142.13:1287
dropped.
[2005-12-13 13:37:10] Got incoming stream connection to :135 from
2XX.1XX.142.13:1359.
[2005-12-13 13:37:11] Got successfully exploited via MS03-26 vulnerability!
[2005-12-13 13:37:11] Detected generic XOR Decoder, key is 12h, code is e1h
(e1h) bytes long.
[2005-12-13 13:37:11] Detected generic CreateProcess Shellcode: "tftp.exe -
i 2XX.1XX.142.13 get ntsfd.exe"
[2005-12-13 13:37:11] Malware Download issued.
[2005-12-13 13:37:11] Associated pairing to :0 from 2XX.1XX.142.13:69.
[2005-12-13 13:37:11] Stream subscription to :135 from 2XX.1XX.142.13:1359
dropped.
```

As can be seen from the log file output, the ability to forensically reconstruct given the low level of detail provided in the logfile is possibly questionable. It is possible however, to use

tools such as tcpdump, which creates a binary dump file that fully records network sessions on a given network interface that would significantly enhance forensic viability. However, in the case of this experiment a high level of forensic detail was not warranted and the standard log format was sufficient for the experimentation.

Method of collection

There were two collectors in the collection system each running Linux Debian 3.1 and mwcollect version 3.02. The machines are running no other services such as 'http' or 'mail' and are not used by any person: they are simply connected to the Internet via an ADSL modem. The machines are located on two separate ISP networks and they are both high-speed ADSL connections, which are connected to different segments of the Australian IP network. The start-up date for both collectors was the 13th December 2005 and the data collection for this analysis terminated on the 6th February 2006.

The resultant log files from both machines were coded and processed to determine simple statistical measures such as frequency and averages. The shellcodes and malware at this point in time have been collected but no detailed analysis such as disassembly has been conducted on them.

Exploit via Known Vulnerability Results

The mwcollect logs an exploit on one of the known vulnerability types as successful or possible. All of the confirmed successful exploits are totalled and presented in Table 1.

Exploit	Host A	Host B	Total	% Of Total
MS03-26	1371	0	1371	61.0%
MS03-26 HD Moore	66	0	66	2.9%
MS04-11	275	222	497	22.1%
MS04-11 House of Dabus	188	102	290	12.9%
MS05-39	12	7	19	0.8%
MS05-39 House of Dabus	4	2	6	0.3%
Total	1916	333	2249	

Table 1: Number of exploits by type for 2 target machines over a period of 56 days

One of the immediate and noticeable anomalies is that Host B did not experience any successful exploit of MS03-26 Remote Procedure Call (RPC) vulnerabilities (Microsoft 2003). What makes this behaviour more anomalous is that the MS03-26 exploit is the largest by category with 61% of overall attacks with Host A suffering these attacks. However, both hosts experienced exploit on the MS04-11 (Microsoft, 2004) and MS05-39 (Microsoft, 2005) variants.

The MS03-26 RPC vulnerability is achieved by sending a crafted packet to susceptible RPC based ports such as 135 and 445. The exploit attacks Microsoft's Distributed Component Object Model and allows full system privilege and the ability to run, install and delete software, privilege escalation and system account creation. This attack requires no other additional action on the attacker's part to achieve exploit other than the sending of the crafted packet.

The MS04-11 bulletin outlines a multitude of attack vectors and subsequent possible exploits on unpatched systems. The bulletin lists 14 exploits of which 8 allow for remote execution, 4 are privilege escalation and 2 denial of service. All of the detected exploits of the

vulnerability from the machines were types that would have allowed for remote execution or privilege escalation. It should be noted that granularity of logging by attack type here on this group of exploits would be complex due to the possible multiple vectors but could greatly aid investigation of the incidents.

MS05-39 attacks are perpetrated on both hosts but account for just over 1% of all attacks. This exploit, in terms of being known to the general public, was only 6 months old at time of the investigation and was little more than 3 months old at the start of the experiment. Likewise, this attack allows for remote code execution and subsequent command and control over the victim machine.

Unknown Shellcodes Results

The mwcollect daemon attempts to verify a shellcode against known codes in an internal database, and remote ones, if allowed by the user. It does this be creating a 'md5 hash' of the file as a check for uniqueness. If the shellcode is not known then the mwcollect daemon firstly stores it on the server into a directory with a unique (md5hash) filename. Upon successful storage it then interrogates the suspect shellcode to try and find any URLs contained within it. If any URLs are contained within the shellcode then they are interrogated to retrieve any malware items that are stored on the URL specified remote servers. The resultant downloaded malware is hashed using md5 for the purpose of uniqueness checking against known 'bads' for example, virus files. If the downloaded file is unknown it is then stored as a non-executable file in secured directories on the collecting server for later analysis. Each of the hosts collected unknown shellcodes the results are displayed in Table 2.

	Host A	Host B
Unknown shellcodes (stored)	2544	377
Successfully downloaded malware	754	8

Table 2: Shellcodes and Malware Downloaded by Host machines

Once again there is large discrepancy in the numbers in Table 2 given that the systems are configured the same. Even with the large discrepancies, each of the hosts indicates a high level of exploit. The successfully downloaded malware, as result of executing the shellcode in emulation, from each host is categorised by filename in Table 3.

The types of malware represented in Table 3 are extensive. There are various programs that are recognised as simple viral payload as well as more complex, multi-partite, viral code. Many of the detected malware programs themselves provide 'backdoor' or covert remote access to a compromised host. A further level of complexity was found that allows the use of the victim machine to compromise other vulnerable machines via remote control mechanisms. This allows an attacker to use captured bots to grow the network almost exponentially and with a limited exposure of their credentials or activity. As an example a few simple command lines issued via a remote control mechanism such as an IRC channel could have the bots scanning large network address spaces, only relaying data upon completion of the scan.

Filename.exe	Frequency
setup_"numeric"	210
lsass	109
winlogon	79
services	61
eraseme_"numeric"	57
msblast	35
start_msblast	35
mcafeeWALLX	31
win32ssr	13
bling	11
start_teekids	11
teekids	11
WinXP32	11
mslaugh	10
start_mslaugh	10
enbiei	9
ipconfg32	8
start_enbiei	8
"numeric"_upload	5
aa327c9	5

Table 3: Names of downloaded executable files as detected by mwcollect

It is interesting to note in Table 3 that there is specific software that is used to disable security countermeasures that a user may have in place such as firewalls and virus checkers. Not only do some of these malware programs disable countermeasures but also put in place keyboard loggers and other similar covert malware programs that defeat re-installed countermeasure or cryptographic countermeasures. These malware programs defeat countermeasures and cryptography by continuous logging of keystrokes or simply by waiting for a login event window to capture username and password information.

Network Results

In Table 4, the top 20 attacking IPs by connection are displayed. Where the HomeA or HomeB has been used, this attacking IP has originated within the ISP address space on which the two mwcollect hosts were connected. Where the hostname is marked 'ExtXX' this an external network to both known hosts.

It is interesting to note that both hosts sustained considerable attack and probing from within their own network space. That is, from within each ISPs own sub network. This level of probing/exploit is significant. Overall, for the top twenty attackers, Host A had a total of 3914 of 5188 (75%), Host B 2240 of 4189 (53%) attacking connections originating from the two ISP networks. It also interesting to note that Rank 13 on HostA attack was, in fact, a machine located in HostB's IP space possibly indicating this machine was used as a scanning bot.

Rank	Host A		Host B	
1	HomeA.254.183	732	HomeB.81.174	1,058
2	HomeA.72.141	656	HomeB.112.141	574
3	HomeA.8.164	426	Ext1.164.235.203	411
4	Ext2.50.252.4	422	Ext2.132.154.171	152
5	HomeA.122.72	285	HomeB.60.115	152
6	HomeA.13.103	284	Ext.254.234.110	152
7	HomeA.86.48	256	HomeB.83.231	148
8	HomeA.13.69	236	Ext3.193.137.2	136
9	HomeA.101.16	234	Ext1.70.56.156	136
10	HomeA.13.155	208	Ext1.33.167.90	136
11	HomeA.228.251	154	Ext1.36.223.211	136
12	Ext3.206.136.124	152	Ext9.144.35.61	136
13	HomeB.217.45	152	Ext10.50.111.122	134
14	HomeA.113.225	146	Ext4.90.111.214	128
15	HomeA.90.80	145	HomeB.54.82	125
16	Ext5.83.151.121	140	Ext4.81.43.76	120
17	Ext1.204.118.155	140	HomeB.111.9	93
18	ext6.113.51.7	140	HomeB.193.36	90
19	Ext7.239.207.29	140	Ext11.152.231.161	86
20	Ext8.38.55.149	140	Ext12.21.66.235	86

Table 4: Attacking Hosts by frequency of connection

In Table 5, a descriptive statistic is given by attack port used. As can be seen in the table, the majority of attacks originated on ports 135 or 445. This is consistent with the types of attack that would be perpetrated on vulnerable Windows machines based on the exploits being monitored for.

TCP/IP Port	HostA	Host B
135	10934	5,898
445	11266	7,965
1,025	0	40
Other	988	0
Total	23188	· 13903

Table 5: Connections based on inbound TCP port

Discussion

Of concern is the high level of exploits that would have occurred at the rate of approximately 40 compromises per day per machine. These are not insignificant numbers and show the real threat to vulnerable systems that are connected to the Internet from self-propagating malware. Furthermore, the longest any system survived as *virgo intacta* was a period of six hours before successful penetration and exploit of a vulnerable system. This grim statistic from this study concurs with other research about system survivability by other authors and sources (SANS, 2006).

Also, what was of concern was the high level of probing and compromise witnessed within both ISPs private subnets. This observed behaviour implies that there is little if any network monitoring and control of ISP internal subnets occurring. Much of the detected activity in this study could be readily combated by appropriate filtering and control of network packets within these subnets.

It would be reasonable to state that surveillance and appropriate incident response to malfeasance within these ISP networks is also poor given the consistent high level of probing and exploit by the other hosts within these networks. This initial data has major implications for network security at the individual business infrastructure level. If this behaviour is extrapolated across other ISPs it also has implications at a national critical information infrastructure level as well. If ISPs are not actively monitoring and responding to attackers within their networks, what chance is there to stop this phenomenon at a broader level? The initial evidence indicates that ISPs are possibly complicit in the propagation of these types of attacks by allowing these inappropriate behaviours and illegal activities to continue. These identified behaviours are the root cause of botnet and malware propagation on the Internet today.

Another problem indicated from the data is that the majority of attacks were MS03-26 based. The MS03-26 exploit is the use of a crafted network packet which a simple packet filter or intrusion detection rule set would deal with. What compounds this issue is that one ISP had no occurrences of MS03-26 detected at all, thereby indicating either they were extremely fortunate or had some level of filtering or countermeasure in place that dealt with these packets. Excuses of cost from ISPs are largely defeated when an open source tool such as the Snort IDS has rule sets that deal with or actively alert for these types of exploit.

This trend also highlighted that attackers are still deploying relatively aged exploits for example, MS03-26 which is over 3 years old on networks to attempt to achieve compromise of machines. There may be several reasons for this, one of which is that the tools attackers are using have old code bases and the programs probe for these regardless. The second possible reason is that they are using multi-partite attack approaches with code capable of performing a variety of probes or compromises. The final one is that the attackers are still finding machines that are able to be exploited via this old exploit code base. This would indicate that the education of end-users and businesses as to why they need regular installation of vendor supplied updates is failing or simply being ignored.

Some of the exploits attempted were naive or novice in nature, evidence of this was many of the IP addresses using A, B and C private network addresses that would not be routed on the Internet. The private IP address space decodes that were found in the data could mean that individuals were compromising internal hosts on a large corporate network. Then via covert or direct channels the malware then started propagation on the Internet. One explanation for this is a compromised laptop that is removed from a protected work environment and used at home or some place that did not have as strict network control. This placement of the laptop would then allow it to attempt to compromise other machines on the Internet. Another explanation is that it someone who is simply inept at modifying the code properly.

The data extracted from the logs also presented some IPs that no longer existed indicates possible removal or recovery of the offending machine. It would be a worthwhile exercise to determine if in the case of non-functioning IPs whether they ever truly existed that is, are they

Authors

Dr. Berna O. Dike-Anyiam has over 5 years of experience in network administration through her work in the industry and entrepreneurship. She is a co-author of *Data security in the electronic economy*. Dr. Dike-Anyiam has a Master of Science in Public Administration and Housing from Middlesex University, a Master of Business Administration in Computer Information Systems from De Sales University and a Doctor of Business Administration in Computer Information Systems from Argosy University.

Dhiraj Bhuyan is a CISSP certified Senior Security Researcher at British Telecommunications, in the United Kingdom. He has worked in many different areas of security including Voice over Internet, secure remote access, Wi-Fi technologies, Smart Cards, 3GPP IMSplatform, Liberty Alliance, Trusted Computing Platform, firewalls, botnets, malware, computer viruses, honeynets, Distributed Denial of Service attack mitigation and Broadband gateways.

Pertti Kuokkanen has received his Master of Science degree in Computer Science from the University of Helsinki, Finland, in 1999. He is a Doctoral student, and conducts his post-graduate research in computer science with primary interests in modelling of decision support applications. He is currently the Chief of the Kuopio Ordnance Depot at the Defence Forces Materiel Command.

Graeme Pye is an Associate Lecturer and doctoral candidate with the School of Information Systems, Deakin University, Australia. His research is focused on developing a practical model of Australian critical infrastructure and investigating the influence of relationships between associated infrastructures.

Professor Qamar Rehmani teaches in the Information Systems concentration at Argosy University. Dr. Rehmani has over 20 years of experience in information systems through his work in academia, industry, and entrepreneurship. He has worked on research projects on NASA's Space Shuttle. Dr. Rehmani has a B.S.E.E. from the Indian Institute of Technology, Kanpur, and earned his MBA and PhD from the University of Houston.

Dr. Craig Valli lectures as a Senior Lecturer in Computer and Network Security at Edith Cowan University. His active research profile includes honeypots, digital forensics and network security. He is completing a second doctoral study on applying honeypots as a countermeasure to insider malfeasance. Craig regularly consults to government and private industry on network security and forensics. He is founding Chair of the Australian Digital Forensics Conference and also founding Co-Chair of the Australian Information Security Management Conference. Craig is also active on various conference and journal committees.

Professor Matthew J. Warren is the Head of School and a Professor in the School of Information Systems, Deakin University, Australia. He has gained international recognition for his scholarly work in the areas of Information Security, Risk Analysis, Electronic Commerce and Information Warfare.

Suen Yek is a doctoral candidate within the School of Computer and Information Science at Edith Cowan University in Perth, Western Australia. Her research is on wireless honeynets and the application of deceptive mechanisms for countermeasures against network attacks. Suen holds a Bachelor of Science (Software Engineering) with Honours and a Bachelor of Business (Marketing).