Worm Charming:
Taking SMB Lure To The Next Level.

Martin Overton, IBM Global Services, UK
(Global Virus Emergency Response Team)

Email: ***overtonm@uk.ibm.com***
WWW: ***http://www.ibm.com/uk***

Tel: ***+44 (0) 2392 563442***

Abstract:

Over the last two years, worms have resurfaced as a major headache, especially for the companies that get hit by them. Worms aren't new; they have been around since almost the dawn of computing.

With the likes of Nimda, CodeRed, and last years quietly successful worm, Opaserv the rules have changed and the stakes are now significantly higher than ever before.

This paper will use the SMB Lure design as presented by John Morris of Nortel Networks at VB2002 as a staring point and cover how it can be extended to improve its usefulness, not just to corporates but also to researchers in the AV companies, these improvements will include:

- *Sample Capture, via custom scripts/tools.*
- *Sample Recognition, MD5 hashes and anti-virus tools and storage.*
- *Integration with other technologies, such as IDS, Integrity Checking, anti-virus and custom scripts and other useful tools.*
- *Automation*

By the time that VB2003 arrives a prototype system, based on the technologies and methodologies mentioned above will have been running for almost a year, so there should be some very interesting statistics as well as lessons learnt along the way to share……

Early statistics and information obtained using a very early version of this system was used in the article entitled "Are You Being [Opa]Serve[d]?" in the January 2003 edition of Virus Bulletin magazine.

This paper was written for, and presented at, the 2003 Virus Bulletin conference at the Royal Oak, Toronto, Canada on September 25th – 26th 2003.
I would welcome any constructive feedback on this paper and its content.
(Martin Overton 1st October 2003)

---

1 Introduction

This paper will use the SMB Lure design as presented at VB2002[1] by John Morris of Nortel Networks, and a fellow AVIEN member, as a staring point and cover how it can be extended to improve its usefulness, not just to corporates but also to researchers in the AV companies. This paper will not go into great depths on how to extend SMB-Lure as this is beyond the scope of the paper and the paper length imposed by the conference organizers.

At the time of writing this paper, a prototype 'extended' SMB-Lure has been running on the internet for 8 months. During this period, a number of brand new malware, and numerous variants have been trapped, analysed and then passed onto the AV companies for inclusion in their products.

The greatest success has been with new variants of the Opaserv family; over a dozen new variants were trapped first by this 'extended' SMB Lure design. However, it is not just share-aware worms that have been trapped using this prototype system, a number of Bots, Backdoors, RATS, and other malware have also been snared.

Finally, the real surprise with some of the trapped samples is that around 20 percent are carrying other malware with them, as a passenger – like some form of 'malware transit system'.

2 Definitions

Let's get a few basic definitions out of the way first so that we all understand what is meant by the following terms:

SMB:- *SMB, which stands for Server Message Block, is a generic protocol for sharing files, printers, serial ports and communications abstractions such as named pipes and mail slots between computers. Microsoft implements their own form of the SMB Protocol, to provide file and printer sharing in all versions of Window*[1]*s.*

Worm: - *A worm is a program that makes copies of itself, for example from one disk drive to another, or by copying itself using email or some other transport mechanism, such as the network. It may do damage and compromise the security of the computer, but it doesn't replicate by changing a hosts code or files."*

"Viruses infect, worms infest"[2]

Worm Charming (Original):- *"A quaint English sport started in 1961. Requires a 3 yard by 3 yard patch of grass (the pitch). The aim of the sport it to 'charm' or otherwise capture as many worms from the 'pitch' in the set time (usually 30 minutes), no digging, drugs or water allowed!"*[2]

The record is 511 worms, and is in the Guinness Book of Records

Worm Charming (Modern): - *"A quaint sport started in 2002. Requires two PCs, one to act as bait (the internet is the pitch), and one to charm (capture) any worms that take the bait. No imposed time limit. No garden forks or bending required!"*

The record is 13,024 samples in one month!
That's over 38 different worms (and variants)

[1] Source: http://hr.uoregon.edu/davidrl/samba/samba-intro.html#METHODOLOGY
[2] More information can be found here: http://mysite.freeserve.com/wormcharming/

3 What is SMB Lure?

"The SMB Lure is a network security sensor that remotely detects computers infected with file-share worms on a corporate network. After entering a corporate network, worms such as Funlove, Elkern, Klez, Nimda, Sircam, and Qaz exploit shared folders as a stealthy means of infecting more computers. The SMB-Lure actively attracts file-share worms to itself so that it can detect and identify the infected computers.

The SMB-Lure has been successfully used to detect hundreds of infected computers on both corporate and educational networks.

The SMB-Lure is built using a Samba file server specially configured to appear as a large number of computers in the Windows Network Neighbourhood. These virtual computers are positioned in strategic places, in the spread pattern of file-share worms, within the Network Neighbourhood.

The Samba server is configured to run in debug mode to provide extensive logging of each worm visit. The Samba file-share is baited as a honey-pot, containing a variety of interesting files and directories for the worms to interact with. [3]"

In short:-

- Originally created by John Morris of Nortel (a fellow AVIEN member). Instructions can be found here: http://smb-lure.dnsalias.com/smb-lure.htm
- It is a customised and specific SMB.CONF file for SAMBA running on Linux.
- It offers an 'open' share with Windows files and directories to other systems.
- Scripts to grep the SMB log files for signs of infections (actual and attempted)….These were created by Paul Schmehl (another AVIEN member) and can be found here: http://www.utdallas.edu/~pauls/EduTex/checklogs.perl or by following the links on John's SMB-Lure web page.
- Originally created for use on internal (private) IP networks.

Let's now look at how I've used it and extended its functionality and what I have found out 'on the journey'….

The prototype system created, builds on the original design from John Morris, but instead of using the SMB Lure on an internal network, mine was pointed at the Internet.

You could consider this to be an early-warning system, in some ways similar to an Intrusion Detection System that is placed in-front of your DMZ. This will pick up threats earlier, and may give you and your company more time to prepare your internal/gateway defences against these new threats.

Just like an Intrusion Detection System, an SMB Lure should be placed on the internet, as well as others on your internal networks. This will help to identify which threats are getting past your firewalls and other perimeter defences. Furthermore, in the event of an outbreak it will allow you to identify 'problem' areas/systems with greater speed than just relying on anti-virus tools or the end-users to report the problem. Also, just like an IDS, SMB-Lure will need some tuning and ongoing maintenance to keep it effective.

4 Parts of the Puzzle

This section will cover the hardware and software that is required to setup a SMB-Lure system, and also the storage and sample capture system. The following is a 'shopping' or 'ingredients' list to create a single 'extended' SMB-Lure. It would be possible to configure both systems on another system running VMWare and each system as a Virtual Machine (VM), however doing so may be less secure.

4.1.1 Ingredients

- Two 'old' Pentium class computer base units (or better).
- Two network cards.
- One hub/mau (depends on network flavour preferred)
- Cabling to match, both network and power requirements.
- One internet connection, preferably DSL or better.
- One Windows OS (98SE or later)
- One Linux OS (or other preferred *NIX flavour)
- One *NIX integrity checker, whatever is your preferred brand.
- SAMBA.
- Firewalling and/or NAT router
- Custom scripts.
- Custom 'monitoring' application(s)
- Disk space to suit.
- Remote management and maintenance tools, whatever flavours you prefer or have available.
- Optional: IDS and AV
- Optional: More systems for use on your internal networks.
- Encryption software, such as PGP Disk or DriveCrypt.

4.1.2 The Recipe for the WormBait System

The following is the recipe to create the 'WormBait' system. This is the one that has the open share and gets infected files dropped onto it.

- Take Linux box already configured with SAMBA and apply SMB-Lure SMB.CONF[3] (modify as required).
- Add network card and cabling as needed.
- Create the 'open share' directory, add sufficient rights (RW) and 'bait' with a suitable selection of tasty files and directories.
- Take baseline 'Integrity' snapshot of prepared 'open share'
- Add SMB-Lure scripts, and setup CRON job to report log matches.
- Add 'Integrity' CRON job to report ALL changes to the 'open share'
- Add SSH for remote management.
- Add firewalling to suit, allowing full internet access to 'open share' on ports 137/udp, 139/tcp and 445/udp and tcp. (If using a router that supports port forwarding, then forward the above ports to the WormBait system internal (Private) IP address.
- Add anti-virus if required to monitor the rest of the system (but not the 'open share', unless you really want to).
- Add custom scripts and other utilities to suit.

4.1.3 The Recipe for the WormCharmer System

The following is the recipe to create the 'WormCharmer' system. This is the one that monitors the 'WormBait' open share and captures the infected files dropped. It also does the samples processing, hashing and may also run other tools, such as WormCatcher, IDS and Anti-Virus.

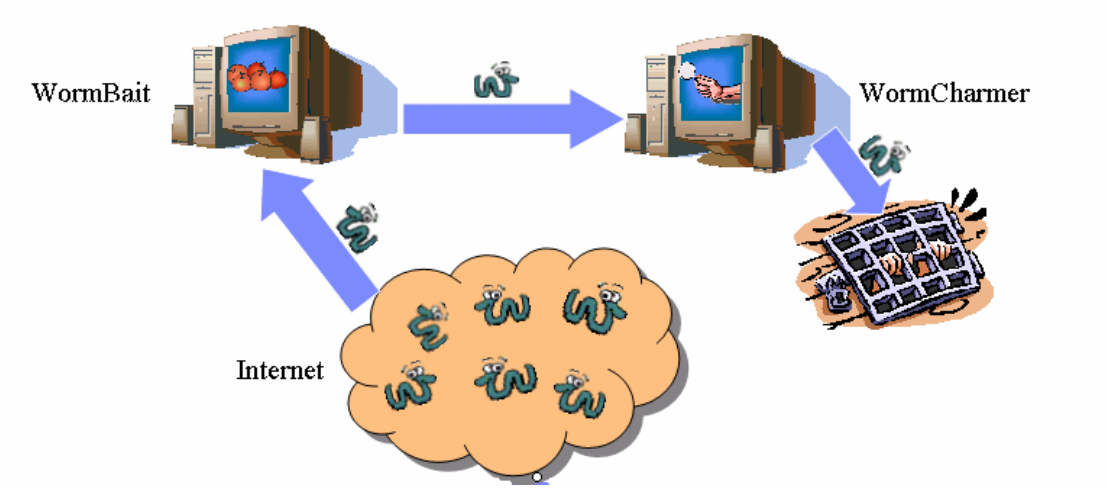
- Take Windows box already configured.
- Add network card and cabling as needed.
- Connect both WormBait and WormCharmer to hub/MAU

[3] See example in Appendix A.

- Connect to the ‘open share’ directory on the WormBait system (map drive to suitable letter.
- Create ‘worm-prison’ directory to store ‘charmed’ samples. Ideally should be encrypted.
- Add firewalling to suit, allowing read/write access to ‘open share’ on WormBait
- Add anti-virus if required to monitor the rest of the system (but not the ‘worm-prison’ directories).
- Add custom ‘Charmer’ application and configure as needed. Set it to check at regular intervals for dropped worms.
- Sprinkle ‘Roger Thompson's WormCatcher’ in to the mix to round out the detection capabilities.

5 Putting it all Together

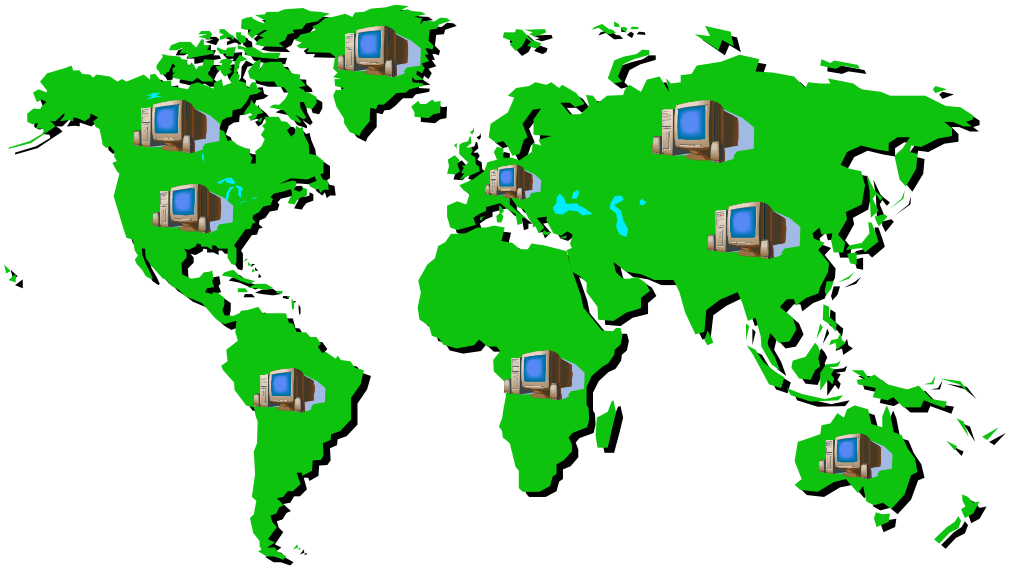
Take the systems created by following the ‘recipes’ above, and then connect the ‘WormBait’ system to the internet. Wait a short while, and the worms and other malware will soon appear.



The above is a basic representation of the completed (standalone) extended SMB-Lure system.

Similar systems could be deployed in strategic positions round your internal networks, such as one per geographic region; this will then allow you to identify the geographic point of entry into your network.

For example:



Let us say that the SMB-Lure system in Australia trapped a new share-aware piece of malware (or BOT being seeded). This information would allow the rest of the 'network of sensors' to be updated to specifically detect it, but more importantly, it would allow perimeter defences to be updated (either generically or by updated virus signatures from you preferred vendor(s)).

If we say that this new piece of malware uses a static filename, then it is very simple to add 'filename' blocking at other entry points on the network, such as e-mail or web access, as at this point in time we may not know what other methods of propagation this new malware uses, other than being share-aware.

At the same time, a sample should be sent to the AV community to ensure that any outbreak is minimized; alerts sent to mailing lists such as AVIEN and Bugtraq might also be advisable, especially if large numbers of samples are being trapped and no AV software currently detects it, even with heuristics switched to its most paranoid mode.

In a worst-case scenario, it could allow the infected network to be segregated from the rest of the corporate network, so that the outbreak can be isolated and starved once its 'local' victim pool is exhausted. The other option would be to automatically disconnect the infected system from the switch using John Morris's 'Network Isolator' design. However, there are two problems with this methodology:

1. The requirement that switches are used throughout your infrastructure and that they can be remotely managed down to port level. In many large companies this may not be a problem, but in small to medium organizations it may be insurmountable without great expense.

2. You could end up in a situation where you end up performing a DoS on your own network if strict guidelines are not followed or sufficient 'logic' added to the 'Network Isolator' software.

5.1 Sample Capture, via custom scripts/tools.

Now we have the 'extended SMB-Lure' up and running, what do we do with the malware that are being dropped to 'WormBait's' open-share?

First things we need to do are:

1. Monitor the output of the hourly 'Integrity Check' that was setup to watch over the open-share on 'WormBait'. This will pick up any discrepancies between the 'base-line' database, and the state of the open-share at the time it last ran.

An example AIDE cron job output looks like this:

```
From root  Fri May 23 12:08:25 2003
Date: Fri, 23 May 2003 12:08:24 +0100
From: root <root@localhost.localdomain>
To: me@localhost.localdomain
Subject: Aide Report

AIDE found differences between database and filesystem!!
Start timestamp: 2003-05-23 12:05:00
Summary:
Total number of files=2273,added files=3,removed files=0,changed
files=0

Added files:
added:/home/windows/malware.exe
added:/home/windows/marco!.bat
added:/home/windows/sample.pif
```

As you can see this shows that 3 new files have been dropped and currently not cleared (these are fictitious files). If for instance, an existing file was infected, then AIDE would also show that the file

was changed and the differing Hash values. This is the case with a Nimda infection via SMB, as it drops numerous *.NWS and *.EML files and will also infect a number of *.EXE files.

A useful tool to look at remote (and probably infected) systems connected to 'WormBait's' open share is the SMBSTATUS command. Below is an example of the output showing three connections that are from infected systems. The locked files section shows the files being dropped and/or modified by the remote infected systems.

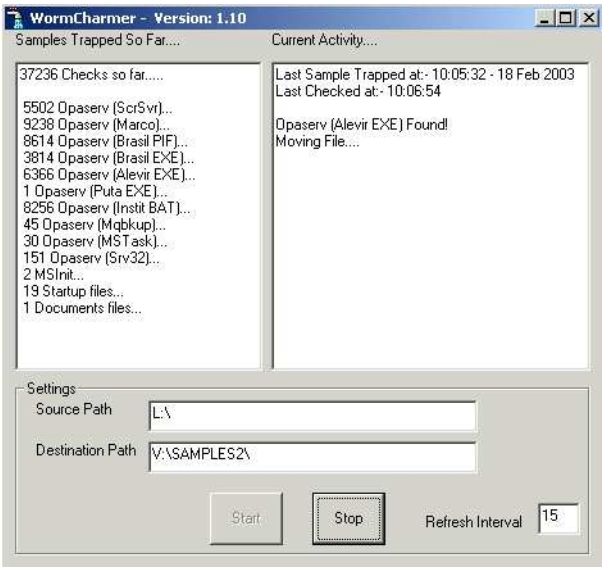
```
Samba version 2.2.8a
Service      uid      gid      pid     machine
----------------------------------------------
C            nobody   nobody   25217   snorty   (10.109.37.2)    Fri May  2 15:47:35 2003
C            nobody   nobody   26041   gustavo  (66.167.119.124) Fri May  2 16:15:16 2003
IPC$         nobody   nobody   25217   snorty   (10.109.37.2)    Fri May  2 16:12:52 2003
IPC$         nobody   nobody   28460   smbd     (216.50.4.38)    Fri May  2 07:06:33 2003
IPC$         nobody   nobody   26067   smbd     (218.62.11.250)  Fri May  2 16:38:27 2003


Locked files:
Pid    DenyMode   Access      R/W        Oplock           Name
--------------------------------------------------------------
25217  DENY_NONE  0x20089     RDONLY     EXCLUSIVE+BATCH  /home/wormbait/WIN.INI   Fri May
2 16:11:41 2003
26041  DENY_FCB 0x3           RDWR       NONE
/home/wormbait/WINDOWS/instit.bat   Fri May  2 16:15:16 2003
26056  DENY_FCB 0x3           RDWR       NONE
/home/wormbait/WINDOWS/alevir.exe   Fri May  2 16:20:50 2003
```

2. Monitor the 'open-share' for new files, to do this we need to create and install the 'WormCharmer' application, details below:

The next screenshot shows a custom tool which runs on the 'WormCharmer' system and performs the following tasks:

- Monitors the open-share on 'WormBait' for specific dropped files, and monitors specific directories for 'any' dropped files.
- If any dropped files are found, they are captured, details of the file, date, time, size, MD5 hash is generated and logged.
- Finally, the captured dropped files are stored in a directory structure, ready for processing, and any further research.



Screenshot of actual 'WormCharmer' application in use on the prototype 'Extended SMB-Lure' system

In the case above, the 'WormBait' open share (Source Path) has been mapped as drive 'L:' and all captured samples are sent to 'V:\SAMPLES2' (Destination Path) on the 'WormCharmer' system.

In the left hand window (Samples Trapped So Far), you can see the number of checks, the specific things that are monitored, as well as several directories.

The window on the right (Current Activity) shows the status, last sample captured, last check and the status of any captured files being processed.

This application (by default) will perform the checks every 15 seconds.

The above application was created in Visual Basic 6.0, although a similar tool could be written in Perl, Pascal or C/C++/C#.

5.2 Listening on Port 445 too

The SMB (Server Message Block) protocol is used among other things for file sharing in Windows 9x, NT, 2000 and XP. In Windows 9x/NT it runs on top of NBT (NetBIOS over TCP/IP), which uses the well-known ports 137, 138 (UDP) and 139 (TCP).[4]

The original design as created by John Morris uses SAMBA running in daemon mode. This is fine for many people as it listens for SMB traffic on port 137/udp and 139/tcp, which is perfect for catching Nimda, Opaserv[4], Qaz and many share-aware malware which uses NBT (NetBIOS over TCP/IP) as used by Windows 9x and NT 4.0.

However a number of new share-aware malware use SMB on port 445 (Microsoft-ds) instead, as supported by Windows 2000 and XP (200/XP will both fallback to NBT if needed and enabled). These new 'Microsoft-ds' share-aware malware include Nebiwo aka Deborm, Deloader and Slackor,

So, how do you get SAMBA to listen on other ports?

Well, after lots of searching and testing I came up with the following undocumented configuration.

Instead of running SAMBA in daemon mode (-D), you need to run it via Xinetd, this is the secret to getting SAMBA to listen on multiple ports. To enable this, you need to ensure that the Xinetd.conf file contains the following (adjust the path to smbd and nmbd to suit).

```
service netbios-ssn
{
        socket_type             = stream
        protocol                = tcp
        wait                    = no
        user                    = root
        server                  = /usr/sbin/smbd
        disable                 = no
}

service netbios-ns
{
        socket_type             = dgram
        protocol                = udp
        wait                    = yes
        user                    = root
        server                  = /usr/sbin/nmbd
        disable                 = no
}

service microsoft-ds
{
        socket_type             = stream
        protocol                = tcp
        wait                    = no
```

[4] In Windows 2000 and XP, Microsoft added the possibility to run SMB directly over TCP/IP, without the extra layer of NBT. For this they use TCP/UDP port 445.

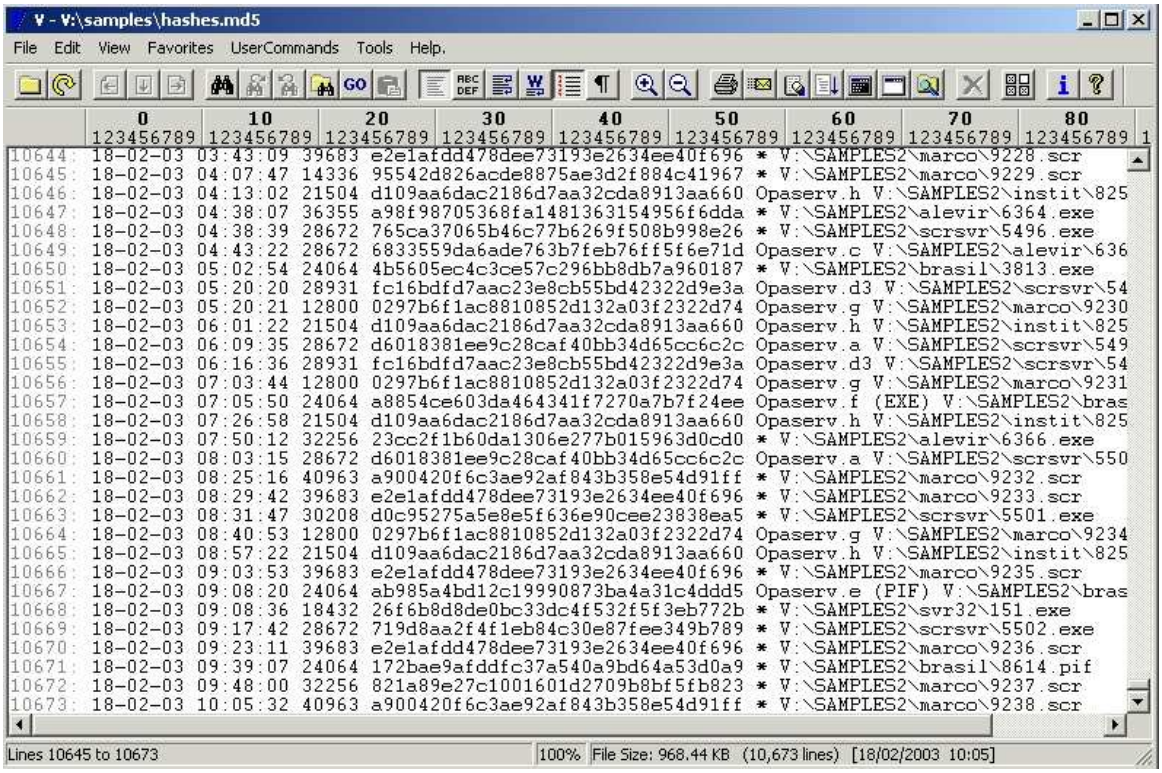
```
        user                    = root
        server                  = /usr/sbin/smbd
        disable                 = no
}

service microsoft-ds
{
        socket_type             = dgram
        protocol                = udp
        wait                    = yes
        user                    = root
        server                  = /usr/sbin/nmbd
        disable                 = no
}
```

5.3 Sample Recognition, MD5 hashes and anti-virus tools and storage.

The previous section covered how the custom 'WormCharmer' tool is used, but not how it works. This section will dig a little deeper and give a fuller understanding to what it is looking for and how it uses MD5 tables to identify what it has caught.

One of the first things added to the extended SMB-Lure 'WormCharmer' application was the ability to generate MD5 hashes for all trapped samples. This allowed, in most cases a simple way to identify certain malware and their variants. The screenshot below shows part of a log file generated by the 'WormCharmer' application. This records the date and time of capture, filesize (bytes), MD5 hash value for sample and finally if there is a match in the MD5 hash table it will indicate the malware and/or variant identified.



A '*' is used where an exact match has not been found. This usually means that further analysis of the file is required. In around 80% of these cases these samples are found to be either minor corruptions of the UPX headers (or other packer/compressor used) or cross-infected with another parasitic file infector. The remaining 20% are new variants or new malware.

Below is a table of MD5 hashes for a number of the malware and their variants:

Malware/Variant[5]	Filename (where 'static')	Size	MD5 Hash(es)
Opaserv.a	SCRSVR.EXE	28,672 32,256 26,624 26,112 36,355	d6018381ee9c28caf40bb34d65cc6c2c 5b859ce64a0efebeccde248dd5bad78b b5985cd86eaf5835dcccf27589a93e34 588de1e0e60be13127bb4bb6d2f8b44a afa22cbdadfac84f00a438e5f8a48d5d fd7615a5049e261e75a0104c0cca99d2 140c0e7d78e5849470a5f72a4e32e217 1ae26c957e439d70392b6bf548cf4a59
Opaserv.b	SCRSVR.EXE	28,672 28,419 32,515 32,256 27,648 26,624 32,259	038b783e70f58c12cb25fef8200ce741 58d3ffa4e6021fa0c819efc4bfdb09df 9282c8e399980c298f656e7f0938a585 74ec3fe41ac5ae3f0d8a3e2c2455b640 aecc8257b4bf32b975616df0bca0d46e 3bcc2817499284c047dcfb07acd35cee 73691090d6baf03348e3aa28a5756225 533fd3846358c1d7658023569df49400 4357adad60e5b2bb6dcde3d7682337e9 a0c1789524c663c5e6ce6d463e1b8062 3b5d1e7e9f1c3e564be7ecdeaf4df495 6656f0c0d3daed62b29527ff6e011bc4 53e45e2b592f8284693677a0d643f777
Opaserv.c	SCRSVR.EXE	28,672 32,256 32,515	6833559da6ade763b7feb76ff5f6e71d e473917d950d3b9950c554b848cba2f7 2583e8055e617f44d7de49562d3875c6 4dbf7f048868b58d699bdff0639c4e23 6876cf1ac29fbf5254a39ea05b8c33c2
Opaserv.d	SCRSVR.EXE	28,931 28,675 28,672 28,675 27,136	63a3d5f2cf63efcca61bd46d3b6eb843 72a16b83b22f6817acc2d8baf2e567f6 fc16bdfd7aac23e8cb55bd42322d9e3a 765ca37065b46c77b6269f508b998e26 6196a8207a0a9cf8000dabb36175c881 07bed8e868b459d6d75992d625310cbe 008f96c3142aa194413a3030e7bf0bf5 a76a7774ae473c36e7255447328c2c2c dbaf5164fb63a463982798575834c589 86eed6db33d3aa3d197ff8fd92bddb2c e79ba1d0b04128ad2396acca58195405 fccbf92d137b389b2f7119beaf699f30
Opaserv.e	BRASILF.PIF	24,064 24,576	ab985a4bd12c19990873ba4a31c4ddd5 3c712348c0833b3f5fbaeb82c881a447 9d5101a6616e481a25d6b4f55462ff16 a23671b3499e0aaac1d8f6190ade6d46 a0b28fd3d83eeecdfba750921faad797 6d5618f9dca73c5f88f39d048b0f520b 8f9e5aa60c4a6199fe19b332d6ab209f 9d9c87e9281bac115b458e4ed4275449 547109d53f25d1e1a65285590f7ec969 793bacd7a69cf649954578ce13175f56 b9799d58f53391270bd2c8a1b9aadbe6 2cc2814d9ccbc37dd2db634fda323617 172bae9afddfc37a540a9bd64a53d0a9 fe5c64b7d5231649143523759452dc1 b0493698e344777c24a26d47a53911aa
Opaserv.f	BRASIL.EXE	24,064 28,160 27,907 27,648 28,675 40,963 45,059	a8854ce603da464341f7270a7b7f24ee 41e38e27cddcb6b949004f39deabcc12 cf81fa30fc6ace6e97b52d40a48e667d 10a46cde3c4a4ceaf4dcaeee3d596ea8 ebd00514636b4a46816329e0f5d011b1 6eb9faf4ca670919d60e9fc33015460f 5efc330bda4aefbd7aa38d1b33b7d916 0550ce07d4c9e6d04ee55f3357da7efe e0f2e7e0ef3474652a3831d96755e8ba e8a3fa9cecdadfd1c92c4c7af09ea090 a3b24a113012ac683174aa68c31e570c 023e85be450dc9975196e614606d0025 f2f0845032100c6523e27d7b48960ce8 97ac7e8af98171e6762b51ae67a701cb 6735bc326c16d4487966a0677aed0301

[5] As detected/classified by McAfee

			06e3836f8750ddc0aa401d66c946b49f
Opaserv.g	ALEVIR.EXE	28,672 32,256 26,624 28,419 28,160 28,675 32,515	5e515609806f8f64365d184045cd9b9e 3934385c3ae1b2b5271361d43a6c6c5b 1aa6f4c294d0a5fde7f898ca7edd47e4 cdb9788ecccad4fd28564a414df0d6c5 9a8743be3b75fadeb4d56a4c4314624a 6833559da6ade763b7feb76ff5f6e71d ef6627d5a6329f7380716a1e82790320 e75f3d519c3dfd60c4ee3473cccef628 5e515609806f8f64365d184045cd9b9e 23cc2f1b60da1306e277b015963d0cd0 8465b32f71c28b529d839ed783ea0e25 2b6f1259aeb69370c1f6c0c424a08920 ba28eaac90aa36e66967157e6006c10a e54ed5baa202fcdd48ac31c901643199
Opaserv.h	MQBKUP.EXE	28,931 17,408 20,480	417e4e0bf1126e8d207a35195b2ac0ff e3ef743b6d341394bce081a1cf429abe
Opaserv.i	MARCO!.SCR	12,800 39,424-40,963 39,937 17,408 40,963 15,360 35,328 39,683 23,552 24,064 30,720 39,424 34,816 39,683 31,232 23,552 44,291	0297b6f1ac8810852d132a03f2322d74 a986be92b27fee190b2a20a4e28fb3d3 d974cb4579436f75fc110b627664d323 a900420f6c3ae92af843b358e54d91ff 67923b48e778598c4cf01272029cca18 f8f98da218cddff696c171ba0cb1a1d8 e2e1afdd478dee73193e2634ee40f696 1d0d6848b32d9da10dfd50a5298c45f7 cfee4a121948d78a1704b5a49b2fab64 139c7056dc3fcefd92a10b3f2b439547 9f0601378b5bf9b5315f5261d3623789 1130d8b74a65abbf030c00f33dcab80c e2e1afdd478dee73193e2634ee40f696 9c37c2b170e3d394017c569ab61ae7ef 087df75a043f20f2975900048d1e107d 060254cc34d544ba17d764e0dc269445 c8b26eb3d1b0539e5ff8b6bf78daf52c
Opaserv.j	SCRSVR.EXE	28672 45,056	cfa54843b9d4fbc9a9892ec279773747 029fe2a777d57c963626091837deac5c 8631f8abd785936b472e2f6eb093efd3
Opaserv.k	INSTIT.BAT	21,504	d109aa6dac2186d7aa32cda8913aa660 133f67fc73085c93647c235b7d40c047 7eb553f9bfa300bda1b36e4db65c4eda caebc888915922882159f2feb7e168cb f12cf2fc0d272aac86ea2d23423a4144 7b580b2484911b3b031605424f45d080 84cce5a599fb395c1511f650589f98cc
Opaserv.l	PUTA!!.EXE	29,065	a0a5ab3340a4f380baf663516593899e
Opaserv.m	MQBKUP.EXE	17,408 20,483 20,739 37,123 20,739	ae872d7d63e6aa8a6ecf1e2895713e7a 9420e7e54d8a14f825461bcab5733e88 19b19be237c940a6c806704410c18dce 2b82aceb633943049d2fc94c9c405f93 8a31f5647b90847a27612bf5041f2a0e 47d38fd308fe36553ca1313de8fc173b a145df831533eb5319db4013f2aaa63b d6d7500e9751e0c9b7404cfac20b65ec 6a83659d33ccfc92705a729b550aae3c d2b8ef6e4a609c2a7d780f6d0d95f361
Opaserv.n	MSTASK.EXE	20,480 20,739 17,408 18,853 17,920	6f771855c468a2d27c271ec03ed97e3d 1e31aa4d58ae72705dfd35f64b4d2004 c42114b126750ef1fa0ddf92ac6383b1 25f28f4d3145c66612586346O7a9b070 527bb7dfe5bd0027b3fa0ea05fbe5e85 d978b1624a482b15d952c41cdd62ba86 f2a004e08bdfa30fb3366529c69dab4f 4c50bbed4c1a86e06784d49ef04eff8d 544c652c2ef2523424e373268abae5eb bd7164103627005732165d5c4a8b676a c757e9c2e1e1aef246b21d4b93664684 d164a10fa91d82cd76320678e9a0c7ab c66602eeab38e08a176940908ac4bf44 70ebc879f795eb4fffe39be6e5b6e618
Opaserv.o	SRV32.EXE	18,432 22,528 24,579	dda445eed0ceb49af3ae56850a4fad94 934737c73a873b7dc133e52603d2e79f a7447883e8739f4820f41c65fbea7ebb 3805e9f2cd18758b944a5cdbf7cbec82 ecae3f43fe9b158ee502b6fbb35dd68c fecc3dd39991a9ed15309274169f3338

			f090da1567de1e1ce311d2764d22ef5c
Opaserv.p	BRASIL.EXE	47,616 53,251 50,176 41,472	282962bcade0a6c287e2b02599768143 79778cc4d39fb4c84c9d994fbeb97311 c02724bad5a3016934b3b25760baa8f5 522253d12892a11b675604c368499bc3 42996068bb2850f7f69509786aadc73e d007e86a1e69b19e361c9dddca9310a1 8983157ed40e2f1f9cef8a3f8f5d288f 9297227b7017366af74f8890df9d8174 c08f453ce0cc8af445b628d06adb5a20 0ea3ae224f3c59e9dfa4c72f78fd43a1 fa790133335ba47634d3d5980b932d20
Opaserv.q	SRV32.EXE	18,432 19,877 22,787	8c4ee9ad5795531652ef9edd49f72780 60895cc9e7c4eb9d271532e8e43e85dc 6f70cd80449df4b421f28da710ecd1e3 051828b578819c7aff98bb0fd17ca9fa 4728770eae84d7f380c254ea7d223481 1a81d053f68f48c344c4776f12b63a46 836bffebc29a4cab24895ceab345b6ef 1749b0db492a9a4ec496edcead2e8001 98b34af1127056b5cf0ba91c3b09fa2b 1976ac3fbcdfa415862b51dd9d7eb381 700545266b69e1787c11ef57d4671d77
Opaserv.r	MQBKUP.EXE	17,408 20,739 20,483 24,576 20,480 35,840	58c03d2485c09662735b9983e42f0143 16fe63154bfdf20088801c276470a13a 9ca6f452ceb337e769cdf6a753c8898d 8f1081599edc5d7e29be328d42dfd8bb b8f404249cbc4d6327be9996f06f1ff3 aafd84c1ee7d547aeb7c1729f3d3c633 e9cb0519fa80a41519d81bac1960ca96 c650d1a801d052af51d8c57cb1f44d32 a23aaa2024129a997d1f5acf6bb2d6ff 0d246aa67a7e4fe5cbd37572bd02d706 b0a7ddc9869c1b851d21315ee62b50a5 ecb17a5f837636767d2b220bb1df33c4
Opaserv.s	SRV32.EXE	18,432	8ab511ae9203eb8681daae8a49ae2e28 de0af623ff26e337c9a23786397aa753 979c1e064fc55fec7d979c4daed4fb6e
Opaserv.t	SRV32.EXE	18,432	!f8836aca2f7e40373bbb5b74c142def2 50eafbaa4ae2e9dd548fcad0e1a5aee6 2fd9543daaeb080b22710410b852bccf 160792a724ce836caa690f711315c760
Opaserv.u	SRV32.EXE	18,432	26f6b8d8de0bc33dc4f532f5f3eb772b 56f46c033dc684b1b3fcca181d0cb7ac b398a9d4a696438393f3d67019cf1dd9 9d4f2063337d4f05976a1b57171ba0be
Opaserv.v	SRV32.EXE	18,432	fda0f4e2819a02bd35de086d1a0065bf
Opaserv.w			52a1b7fd6dd709aab5d216e513e549e4
Nebiwo	Varies	56,320	82d72bbfbfbf98a60ebc2232e201b6d9
SpyBot		23,584	949cf295d96eee032f2133dd74d41eee
SpyBot		25,088	1ab5314dad67240f7a6b23ddfbfc7a36

You may say “But this isn’t required, as Opaserv variants have a fixed name”, to which I would answer, Yes that is true, but several variant use the same fixed filename, or it might just be a new piece of malware using the same filename as one of the Opaserv variants. The other answer is, it may be overkill for malware with fixed filenames, but what of those that use various or random filenames instead, such as Nebiwo aka Deborm?

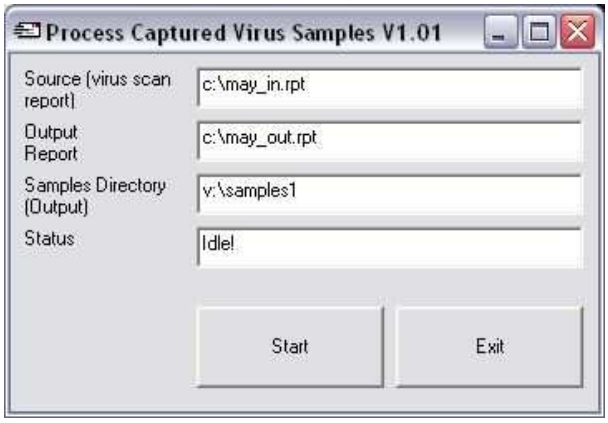
5.3.1 Sample processing:

This system has been capturing 5,000 samples a month, on average. So, how are they processed, identified, stored, and so on?

1. At regular intervals a virus scanner is used to identify all 'known' malware samples in the temporary storage area on 'WormCharmer'. The results are captured in a log file for further processing.

2. This log (from the scanned samples) is then parsed by another application. This application:

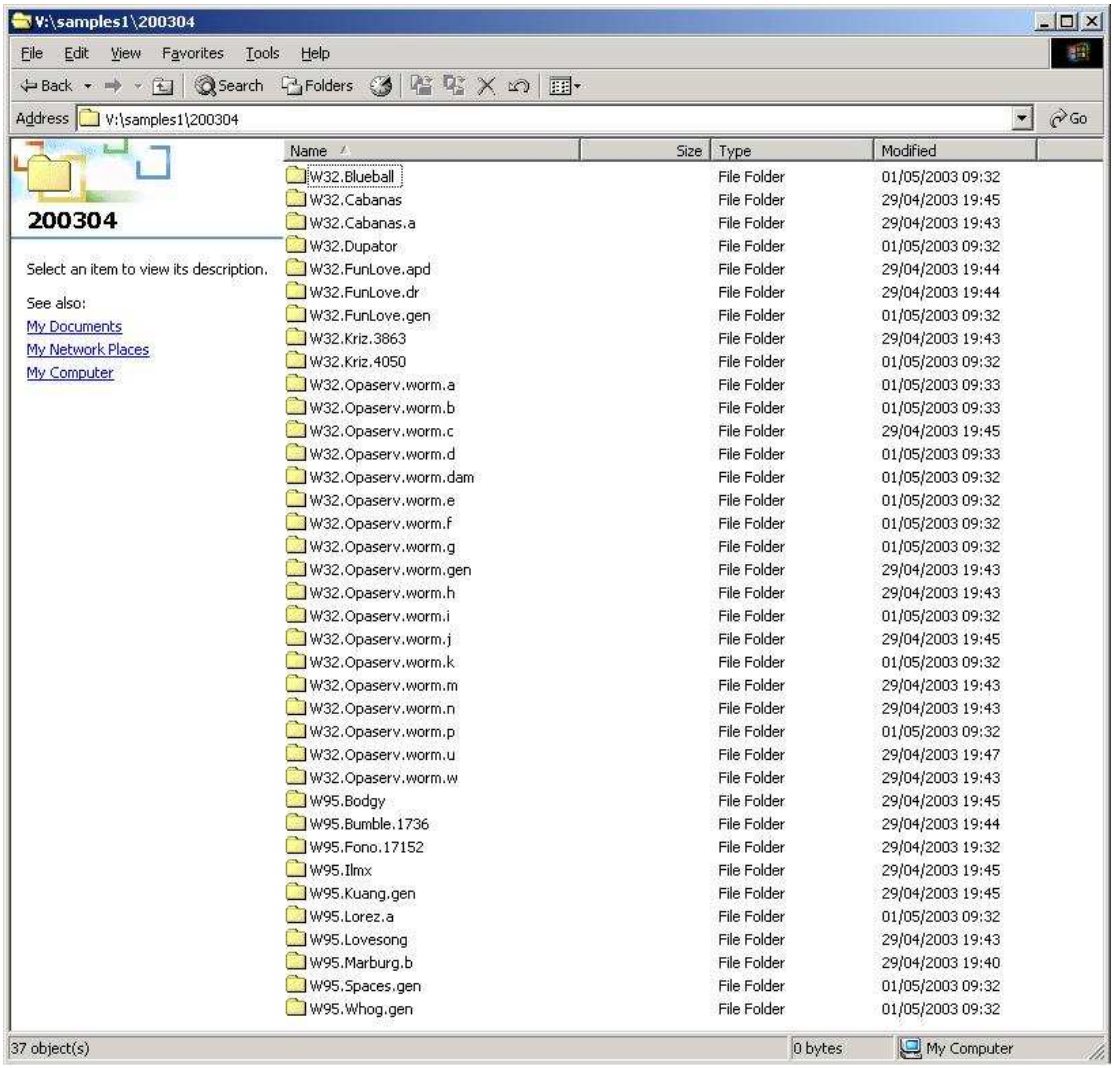
   Takes the virus name, sample name and path from log file
   If required creates, new directory for each virus/worm name/variant
   Also creates a new directory for month/year of capture
   Moves files from capture directory to correct virus/worm directory
   Outputs 'original' and 'moved' files MD5 hash to check integrity of samples



Screenshot of application described in point 2.

3. Results are sent to the WildList, along with any new malware/variant samples. They are also sent to Virus Bulletin for use in their Monthly Prevalence Table.

4. .gen (generic id) variants are sent to AV companies for further analysis, as they may be minor variants or minor corruptions.



Screenshot of actual results of the application described in point 2.

5.3.2 New Malware/Variants:

New malware and/or variants are handled in a slightly different way. These are usually analysed as soon as they appear on the SMB-Lure. Samples are handled using the following steps:

1. Sample(s) are analysed in a 'Virtual Network' to identify any other vectors such as e-mail, backdoor or Trojan functionality. This 'Virtual Network' is run on VMWare[6] with strict firewalling to stop malware escaping onto the host network. It is currently comprised of:

   3 x Windows 98SE VMs
   3 x Windows ME VMs
   3 x Windows 2000 VMs
   1 x Linux VM

2. Filesize, MD5 Hash, CRC32, Filename, Extension, Packer/Compressor used (if any) are noted, as are internal 'text strings' and other pertinent file format structure information.

3. Registry keys set/created/modified, dropped files, and other pertinent information are noted.

4. Once analysed, the samples are sent to anti-virus companies, along with any findings from the analysis phase.

[6] Details and trial software can be found here: http://www.vmware.com

5. The IBM Virus Emergency Response Team is then informed so that any preventative/remedial action can be taken to protect the internal network from the new threat(s).

6. Inform AVIEN membership and possibly send an alert to AVIEN and AVIEWS lists.

5.4 Integration with other technologies: IDS

Once the basic 'extended SMB-Lure' was up and running it was felt that other technologies should be investigated that could augment it's functionality, and even extend the detection capabilities further still. To this end, it was decided that adding 'Intrusion Detection' would be useful as it covers a wide gamut of risks/threats, not just those that travel via SMB (Server Message Block protocol[7]).

Unlike many traditional anti-malware tools, SNORT is not closed source, which allows you to create your own malware detection signatures/rules. These can be as simple as matching a 'filename', binary or MIME signatures for the actual malware you wish to detect.

At the end of the day how you decide to detect a threat is up to you, but it is the speed of deployment and the level of control of rules/signatures which can make the difference between a minor annoyance and dealing with a major outbreak of new malware, which is not yet detected by main steam AV software.

SNORT[8] was selected for several reasons, these include:

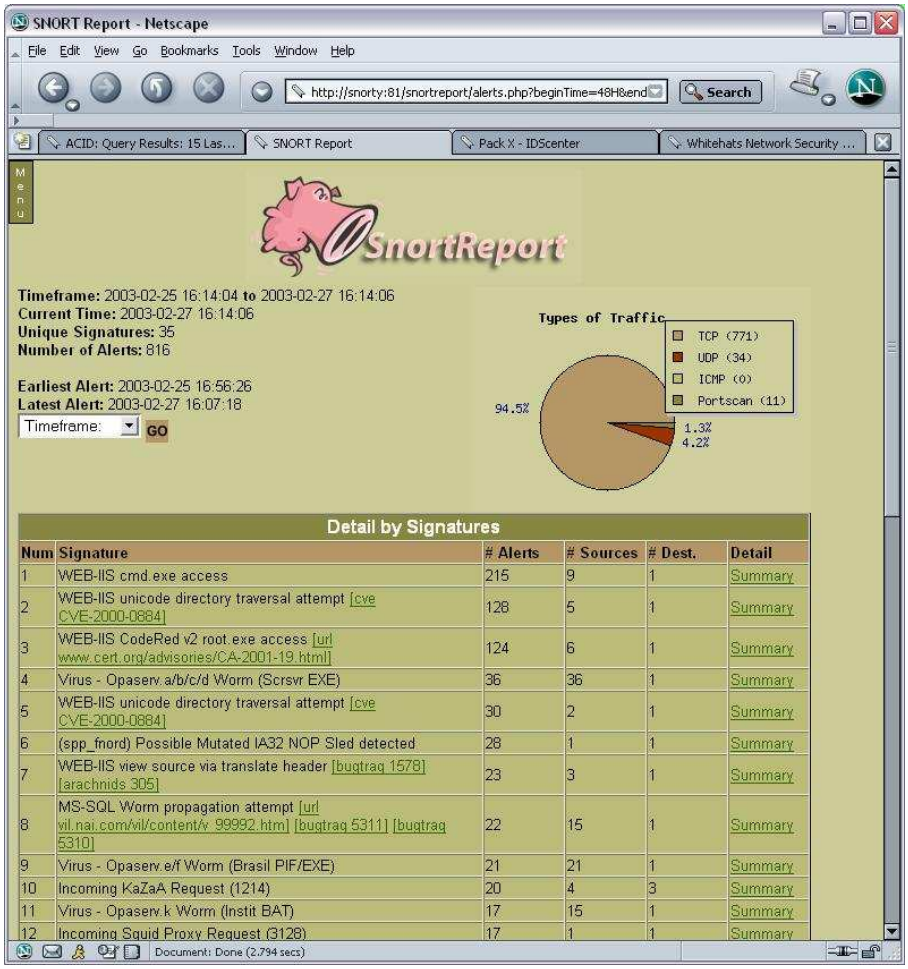
1. The author was already aware of, and using it anyway.
2. Its flexibility and power, especially with the rule language used.
3. It is Open Source.
4. Like SAMBA[9] it is GPL, and therefore very cost effective in any size of organization.
5. It runs on many *NIX flavours, as well as Windows NT.
6. Logging facilities include: Syslog, Database and many other formats.
7. Ease of rule/signature creation.

Also the benefit of running an IDS system with custom malware rules is that all traffic that matches a rule/signature is logged, including the originating IP address (and port). This is extremely useful, especially where you are using SMB-Lures on an internal network, as it can help speed up the removal of a malware infestation because it takes away a lot of the guess work on which machines are infected, even without up-to-date antivirus signatures/definitions. It is also not dependent on a member of the network management team to notice and/or mention' strange' or enlarged quantities of SMB traffic.

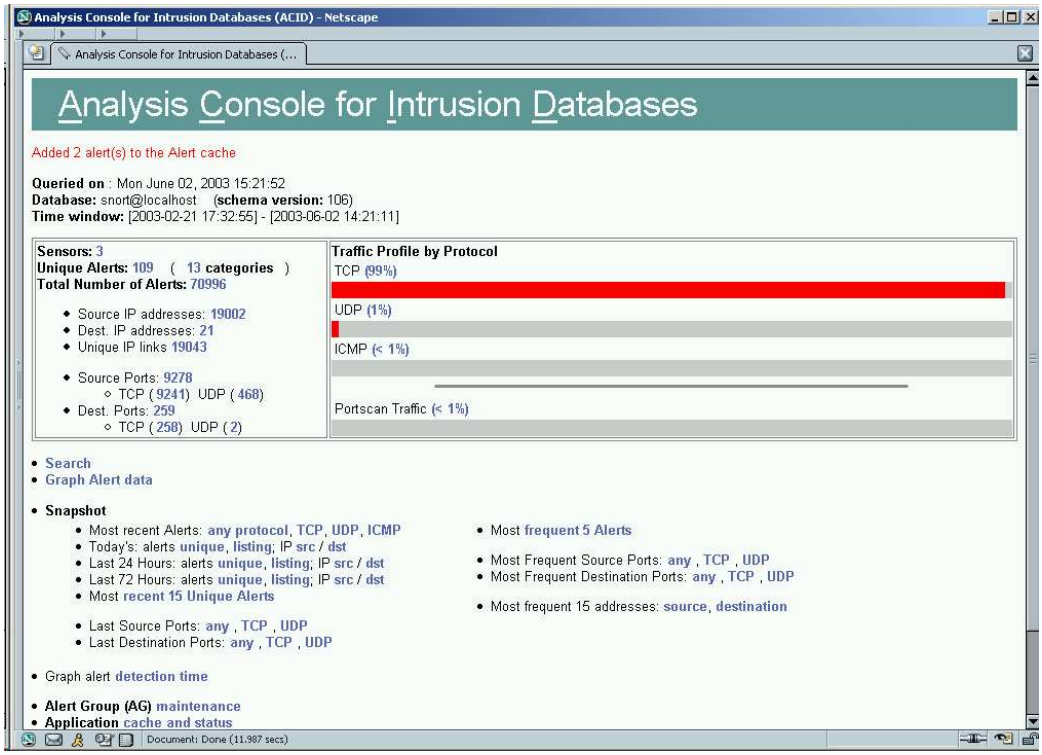
As I write this paper, the 'WormCharmer' system is running SNORT 2.0 on Windows 2000, using ACID (Analysis Console for Intrusion Databases)[10] for graphical output and alert database management, SnortReport[11] for a quick report of recent alerts (see screenshot overleaf), MySQL[12] as the database to store alerts, and lots of custom rules.

---

[7] See here for more details on SMB and how it works: http://linux-mag.com/2001-05/smb_01.html
[8] Details and software can be found here: http://www.snort.org
[9] Details and software can be found here: http://www.samba.org
[10] Details and software can be found here: http://www.cert.org/kb/aircert/
[11] Details and software can be found here: http://www.circuitsmaximus.com/download.html
[12] Details and software can be found here: http://ww.mysql.com



Screenshot of SnortReport output.



Screenshot of ACID

5.4.1 Custom Malware Rules

Below are a few examples of rules (signatures) for detecting malware via SNORT, most of these are 'binary' rules and designed to detect malware arriving via SMB, there are also some rules that will detect the same malware arriving MIME encoded[13]:

Opaserv Variants:-

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Opaserv.a/b/c/d Worm (Scrsvr
EXE)"; content: "|FF 32 C0 F2 AE B8 FE FF FF FF 2B C1 8B FA C3 FC 57 56 53 0B C0 74 44 0B
D2 74 40 8B D8 8B FA 32|"; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Opaserv.e/f Worm (Brasil
PIF/EXE)"; content: "|FF FF FF FF 78 56 06 12 BF 8C A4 F5 E0 59 B9 4C B6 F2 D1 C0 F9 18 4F
50 65 3E 8A E6 A0 AD A2 E5|"; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Opaserv.g Worm (Alevir SCR)";
content: "|41 6C 65 76 69 72 33 31 34 31 35 00 4B 45 52 4E 45 4C 33 32 2E 64 6C 6C 00 52
65 67 69 73 74 65|"; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Opaserv.h Worm (Mqbkup EXE)";
content: "|64 8F 64 68 2B 23 6C 57 AA 00 BF 8E 07 11 00 6D 71 62 6B 75 70 36 6E 31 05 01
4B 45 52 4E CC 4C|"; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Opaserv.i Worm (Marco SCR)";
content: "|6D 61 72 71 75 69 6E 68 1C 6F 73 21 03 4B 45 52 4E 98 4C 33 32 1C 2E 64 6C 78
7B 07 65 67 69 73|"; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Opaserv.j Worm"; content: "|57
56 53 0B C0 74 44 0B D2 74 40 8B D8 8B FA 32 C0 B9 FF FF FF FF F2 AE F7 D1 49 74 2E 8B F1
8B|"; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Opaserv.k Worm (Instit BAT)";
content: "|51 FD D8 73 68 AB 4F 9A 5C DF 50 F8 77 BF 7C C8 35 4B 5E 89 BE 66 F9 16 36 68
BA 06 00 79 6B EF|"; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Opaserv.l Worm (Puta!! EXE)";
content: "|5D 91 80 C9 94 1D DA 88 06 8B 5F 60 35 F9 B5 94 3F CC 42 4D 45 D7 C9 D4 4A 3C
AE 02 6B A8 5A 66|"; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Opaserv.m Worm"; content: "|57
C3 CB F3 A6 5F FB C8 E0 ED 8D 47 01 FF EB 02 33 C0 5B 5E FB 2A A4 56 47 18 AC 0A A6 0F 3C
1D|"; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Opaserv.n Worm (MStask EXE)";
content: "|85 C0 74 06 6A 63 01 64 FF D0 FB 4F 7B 1F 64 40 2E 7F 42 68 8C 01 93 2B 49 0F
19 8B F0 56 FF B7|"; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Opaserv.o Worm (Srv32 EXE)";
content: "|3E 8F BF 79 61 FD 54 A5 62 0D F6 05 D5 9F DA 89 33 89 50 36 F5 93 00 42 73 E1
08 CF C0 E4 56 9E|"; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Opaserv.p Worm"; content: "|57
34 FB 45 23 B9 49 47 01 CB 8A A6 D5 11 4C 60 A8 E8 B7 FC D0 0E 53 FD DE 05 2C 39 52 45 17
D2|"; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Opaserv.q Worm"; content: "|52
6B 67 4D A9 F9 FF AB CB 2A E2 57 26 31 EB 14 DF 61 AF BA 68 F0 5A D6 BF 8C F1 38 E9 1C 62
34|"; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Opaserv.r Worm"; content: "|8B
7D 55 73 03 52 EC 59 54 52 22 57 57 66 8C AE 01 78 E9 6A A9 93 2D 63 8B C2 14 A1 E9 89 06
DD|"; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Opaserv.t Worm"; content: "|B0
73 61 F5 B4 79 32 9E E2 7E 10 CC B0 AE 6A 09 8B EA 67 C8 A0 5F 7E 42 2F 42 23 D3 EA 01 FC
33|"; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Opaserv.u Worm"; content: "|38
A3 EC 64 03 12 C8 92 11 15 9F 12 F2 4F 95 C1 DC D5 C7 10 BC 64 70 DB BA 8B DF 69 BE 40 67
14|"; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Opaserv.v Worm"; content: "|DA
FA 8F 36 6D 78 4B B0 9D 86 DA DA D8 93 B7 BE FD 57 7B A6 EA 6F 6B 12 F8 27 63 00 86 A2 73
35|"; classtype: misc-activity;)
```

Yaha:-

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Yaha.g-1"; content:"|04 53 43
41 4D 33 32 FE BF 3F 77 07 49 52 43 57 49 4E 4B 37 5A 4F 4E 45 41 4C 41 52 4D DB FF F6|";
classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Yaha.g-2)"; content:"|7D FB FF
FF 8B 44 24 04 8B C8 8A 10 84 D2 74 0D 80 F2 BD 88 11 8A 51 01 41 0C 75 F3 C3 90 FF FF|";
classtype: misc-activity;)
```

Ultimax:-

[13] Many more can be found here: http://arachnid.homeip.net:81, please contact me for access.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 139 (msg:"BOT - Possible Ultimax.a"; content:
"rdvs.exe"; nocase; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 139 (msg:"BOT - Possible Ultimax.b"; content:
"rrddvvss.exe"; nocase; classtype: misc-activity;)
```

Acebot:-

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 139 (msg:"BOT - Possible Acebot (MSSG)"; content:
"mssg.exe"; nocase; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 139 (msg:"BOT - Possible Acebot (TSSG)"; content:
"tssg.exe"; nocase; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 139 (msg:"BOT - Possible Acebot (FFEN)"; content:
"ffen.exe"; nocase; classtype: misc-activity;)
```

ExploreZip:-

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - ExploreZip.h"; content: "|F7 FF
8B DF 3B EB 75 C2 8B D6 8B C5 14 10 84 9F 04 16 9E C9 BD 34 5A 5D 5F 88 A7 73 5E F8 7B
FB|"; nocase; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 139 (msg:"Virus - Possible ExploreZip"; content:
"_setup.exe"; nocase; classtype: misc-activity;)
```

SoBig:-

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"W32.SoBig.a - MIME"; content:
"HGdzB4ud6p6TSjUpADN483kA40VGLPSe2RRMvEnwXaS6+DYVCYO2hNX8z/eCUlcCsyESKP0WlLEgSqcQoA1oM9m";
nocase; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> any any (msg:"W32.SoBig.a@mm - SMB";content: "|71 9D CC 1E
2E 77 AA 7A 4D 28 D4 A4 00 CD E3 CD E4 03 8D 15 18 B3 D2 7B 64 51 32 F1 27 C1 76 92|";
nocase; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> any any (msg:"W32.SoBig.b@mm - SMB";content:"|2A F4 6D A0
15 3F 40 08 91 36 12 50 68 6F C4 78 4A EB 25 3B 1A 50 72 EB 13 AD 8F 21 1E 24 0E D4 F0 57
51 CB B0 7B 1D 7C 7F EB E0 23 5C 20 DB 12|"; classtype:misc-activity; sid:900100;rev:1;)
alert tcp $EXTERNAL_NET any -> any any (msg:"W32.SoBig.b@mm -
MIME";content:"MC42MwBGU0ghDAkCCe3ePJa5U8iZ1KgBAHW6AAAAkAEAJgoAv/9l"; classtype:misc-
activity; sid:900099;rev:1;)
alert tcp $EXTERNAL_NET any -> any any (msg:"W32.SoBig.c@mm -
MIME";content:"MC41NABDREQhDAkCCVr2ocZm5U6G3dgBANDSAAAAwAEAJgoAbP9l"; classtype:misc-
activity; sid:900099;rev:1;)
alert tcp $EXTERNAL_NET any -> any any (msg:"W32.SoBig.c@mm - SMB";content:"|08 4D DC FE
58 C3 FB 89 DA 49 14 07 45 83 7D E4 00 75 23 82 B9 45 C6 41 9C 3B EC 02 3B B0 ED C8|";
classtype:misc-activity; sid:900101;rev:1;)
```

Funlove:-

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Funlove"; content: "|CD 21 B0
F0 E6 64 EB FE 90 7E 46 75 6E 20 4C 6F 76 69 6E 67 20 43 72 69 6D 69 6E 61 6C 7E 0D 0D
0A|"; classtype: misc-activity;)
```

Spaces:-

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Spaces"; content: "|83 C5 20 FA
66 8B 75 06 C1 E6 10 66 8B 75 00 56 8D B3 59 10 40 00 66 89 75 00 C1 EE 10 66 89 75|";
classtype: misc-activity;)
```

Dupator:-

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Dupator"; content: "|EB F5 8B
F2 66 AD 66 3D 4D 5A 0F 85 3C 01 00 00 4E 4E 03 76 3C 66 AD 66 3D 50 45 0F 85 2B 01 00|";
classtype: misc-activity;)
```

Backdoor.b4:-

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Backdoor.b4
UPLOAD.EXE/MSAPP.EXE"; content: "|3F 05 64 4F A4 D9 9A 4A BD 90 76 C4 2B 88 D9 DF 3D 59 CD
49 8B D2 CE 5C BE EF E8 E7 00 00 00 00|"; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Backdoor.b4 MSWINSCK.OCX";
content: "|33 C0 EB F6 8B 54 24 04 8B C2 81 E2 FF 00 00 00 C1 E8 08 33 C2 33 D2 F7 71 18
8B C2 C2 04 00 56|"; classtype: misc-activity;)
```

Backdoor.AQM/ANF:-

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Backdoor-AQM MSPSXX.EXE";
content: "|A9 7D A3 6C 80 95 BE 8F D6 5F 46 D8 98 75 33 A0 4C 49 49 41 41 80 39 12 3A F7
4E C2 22 9D 75 EE|"; classtype: misc-activity;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Virus - Backdoor-ANF UNWISE.EXE";
```

```
content: "|08 57 8B C6 83 E6 FE 89 65 F4 83 E0 01 8B 0E C7 45 F8 00 10 40 00 56 89 45 FC
89 75 08 FF 51 04|"; classtype: misc-activity;)
```

Nebiwo:-

```
alert tcp $EXTERNAL_NET any -> any any (msg:"W32.Nebiwo@mm - SMB";content:"|DB C3 30 A4 41
AC C5 64 5B 33 C0 6D EE C7 2E EA E0 DA 56 E9 04 37 10 CD BC F6 B6 6E 85 28 01 60|";
classtype:misc-activity; sid:900100;rev:1;)
```

5.5 Integration with other technologies: Integrity Checking

In the prototype AIDE was chosen as the Integrity Checker to monitor the 'bait' directories and files. This can be found here: http://www.cs.tut.fi/~rammer/aide.html

Other products that may be suitable include: Tripwire.

Why use an integrity checker on 'WormBaits' open share? Not only will it spot any changed files, but also added and removed files too. This is not just useful for malware detection (or actions thereof) but also to identify if someone tries to turn your 'open share' into a warez site.

Below is an example of the output of AIDE running as a CRON job monitoring the 'open-share' on 'WormBait':

```
From root  Sun Jun  8 18:08:03 2003
Date: Sun, 8 Jun 2003 18:08:02 +0100
From: root <root@localhost.localdomain>
To: m155@localhost.localdomain
Subject: Aide Report

AIDE found differences between database and filesystem!!
Start timestamp: 2003-06-08 18:05:00
Summary:
Total number of files=2273,added files=6,removed files=0,changed files=1

Added files:
added:/home/wormbait/WINDOWS/hstlst
added:/home/wormbait/WINDOWS/marco!.scr
added:/home/wormbait/WINDOWS/scrsvr.exe
added:/home/wormbait/SYSTEM32
added:/home/wormbait/SYSTEM
added:/home/wormbait/drivers
Changed files:
changed:/home/wormbait/WINDOWS/win.ini
Detailed information about changes:

File: /home/wormbait/WINDOWS/win.ini
  Size     : 8379                            , 8400

  SHA1     : vwHpOB0q3O+/viVrpWZhDl60yB4=      , k3c1Vf3GDEBo5t2YuHk1wV/YmZw=
```

As you can see AIDE has identified three new directories, three new files (all are Opaserv dropped files) and also noted that the '/WINDOWS/win.ini' has been modified.

5.6 Integration with other technologies: AV

In the current prototype system integration with anti-virus scanners is fairly limited, more a semi-automatic scheduled scan and sort feature. However, I plan to extend this to embed calls to a virus scanner (maybe more than one) to identify malware strains and variants as they are deposited on the 'WormBait' system.

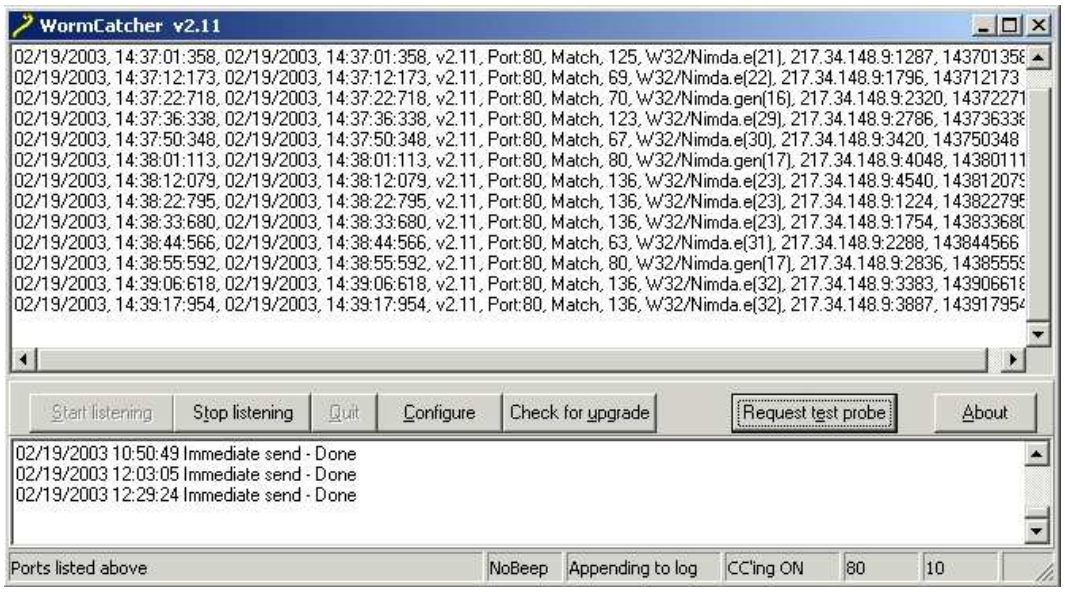
This feature will be added to the 'WormCharmer' application and will be fully integrated.

The more useful side-effect of this is that dropped files that are marked clean (uninfected), will be placed in another directory for immediate review, as it is likely to be something new and probably nasty to boot. The same will also be applied to samples that are flagged via 'Generic' signatures as they are possibly new minor variants of existing malware families.

5.7 WormCatcher

This is not part of the 'Extended SMB-Lure' but is a complementary tool which was created and is maintained by Roger Thompson. The data collected from this tool is collated and publicly posted here: http://wormwatch.org It is an open initiative, anyone can participate by downloading the 'WormCatcher' software and running it.

Below is a screenshot showing 'WormCatcher' running, and catching a Nimda attack.



5.8 Automation

As it currently stands the 'Extended SMB-Lure' can be considered to be automated, or at least semi-automated. Either way, there is still room for further automation, suggestions include:

1. The 'WormCharmer' application could be fully integrated with one or more scanners to help weed out potential 'new' malware, and alert the maintainer to the fact that something new (possibly malware) has arrived on the system. This would also allow semi-real-time statistics to be produced.

2. Automatic submission of statistics to the WildList, for inclusion in a possible 'real-time' version.

3. Files identified in point 1 could be automatically sent to AV companies for further analysis, or if an 'extended SMB-Lure' is deployed within an AV company, this could then feed the samples to their heuristic-based or other analysis systems for detailed investigation.

4. Cross correlation with IDS logs to allow ‘geographic’ infection fingerprinting of malware strains. This would be most useful where a geographic network of ‘extended SMB-Lure’ systems are all internet facing. The same could also be used on internal networks.

The shell script below was written to monitor files dropped to the SMB-Lure and to remove temporary files that were created with earlier versions of SAMBA. It is has been superceded by the ‘WormCharmer’ application, but it is still useful for ad-hoc monitoring.

```
#!/bin/sh

cd '/home/wormbait'
counter=0

while x=x

do
clear
echo "Monitoring WormLure for Temporary and NEW/Modified Files..."
echo "============================================================"
echo " "
tstamp=`date +%c`
tmps=`ls WINDOWS/VDM*.tmp 2> /dev/null`

    if [ "$tmps" != "" ]
                then
                echo "Temp files found...Deleting.... "
                rm -f WINDOWS/VDM*.tmp
    fi

tmps1=`find -mtime -2 -type f`
        if [ "$tmps1" != "" ]
                then
                echo " "
                echo "Modified File(s) Found:"
                echo "======================="
                find -mtime -6 -type f
        fi

tmps2=`find -mtime -2 -type d`
        if [ "$tmps3" != "" ]
                then
                echo " "
                echo "New Modified Dir(s) Found:"
                echo "=========================="
                find -name '[A-Z]*' -mtime -6 -type d
        fi

counter=`expr $counter + 1`
echo " "
echo "+============================================================="
echo "| "$counter "checks so far...."
echo "| Last Check -----> "$tstamp
echo "+============================================================="

sleep 15

done
```

5.8.1 Paul’s Scripts:

Paul Schmehl has created both shell scripts (bash) and Perl scripts to parse the SMB log files for specific indicators that log ‘hits’ for specific malware traffic. It is strongly suggested that you setup a CRON job for whichever script you decide to use.

Below is an example of the output of Paul's original shell script. This has now been superceded by a Perl script which has lots more features, including the ability to send pager alerts.

```
From root  Mon Jun  2 00:10:46 2003
Date: Mon, 2 Jun 2003 00:10:45 +0100
From: root <root@localhost.localdomain>
To: m155@localhost.localdomain
Subject: SMB Lure Logs

Opaserv (Marco) hits = 26.
50163099sp.logname
Log started at
50163099sp.logname
IP is unknown
User logged in was unknown

Opaserv (Brasil EXE) hits = 10.
Opaserv (Brasil PIF) hits = 21.
alevrius_.logname
Log started at
alevrius_.logname
IP is unknown
User logged in was unknown

Opaserv (Instit) hits = 22.
gustavo.logname
Log started at
gustavo.logname
IP is unknown
User logged in was unknown

Opaserv.A hits = 61.
Opaserv (Alevir) hits = 33.
localhost.logname
Log started at
localhost.logname
IP is unknown
User logged in was unknown
```

Paul has also created a shell script to populate empty directories on a 'WormBait' systems open-share. This works by using the *NIX 'touch' command on the input of two lists of 'Windows files' to create 0 byte files.

5.9 Extras

Chkrootkit[14] is a tool for use on *NIX systems to detect the presence of rootkits, backdoors and some worms. This is a useful tool that should be installed and scheduled to run via CRON. It may well tip you off if your 'WormBait' system becomes rooted or backdoored. Should be used alongside a good *NIX virus scanner.

The screenshot below shows part of the report from a CRON job running chkrootkit.

```
From root  Mon Jun  2 00:31:58 2003
Date: Mon, 2 Jun 2003 00:31:58 +0100
From: root <root@localhost.localdomain>
To: m155@localhost.localdomain
Subject: ChkRootKit Report

ROOTDIR is `/'
Checking `amd'... not infected
Checking `basename'... not infected
Checking `biff'... not found
Checking `chfn'... not infected
Checking `chsh'... not infected
Checking `cron'... not infected
Checking `date'... not infected
Checking `du'... not infected
```

[14] See here for more details and software: http://www.chkrootkit.org/

```
Checking `dirname'... not infected
Checking `echo'... not infected
Checking `egrep'... not infected
Checking `env'... not infected
Checking `find'... not infected
Checking `fingerd'... not infected
Checking `gpm'... not infected
Checking `grep'... not infected
Checking `hdparm'... not infected
Checking `su'... not infected
Checking `ifconfig'... not infected
Checking `inetd'... not infected
Checking `inetdconf'... not found
Checking `identd'... not infected
Checking `killall'... not infected
Checking `ldsopreload'... not infected
Checking `login'... not infected
Checking `ls'... not infected
Checking `lsof'... not infected
Checking `mail'... not infected
Checking `mingetty'... not infected
```

6 The Big (Wormy) Picture

It seems that worms are back in fashion once more. Just like the 'Internet/Morris worm' many of its modern counterparts use multiple attack vectors, not just propagating but using known holes in target operating systems and components. These are now more commonly referred to as 'Blended/Cocktail Threats'.

So what does this mean to the 'end-user' community? Simply this, they need to ensure that they follow safe computing practices, and that includes using up-to-date anti-virus, personal firewall/IDS and a lot less inclination to click on attachments or open 'questionable' e-mails. No downloading cracks, keygens, pirated software or other 'ripped-off' copyright material as they may well get more (or less) than they bargained for. Finally they need to be less gullible, as more and more miscreants are learning that the weakest link in most company's security is the human (the 'Wetware') behind the keyboard. In summation we (all of us) need to take security seriously, it is not (as many of them believe) someone else's problem, it is a problem that we all need to address. If you are not part of the solution then you are part of the problem.

6.1 The Future?

What of the future? Hmmmmm…..crystal ball gazing is fraught with danger, but unfortunately I see no end to the problem as long as we wish to have 'usable' systems. We will see a further merging of the hacking/malware underground and the offspring of this unholy marriage will be more and more complex malware with more and more attack vectors being added to extend the current unholy trinity; E-mail, SMB, and P2P[15][16]

On the anti-malware side of the coin, I believe that we will see more integrated solutions that use the full gamut of security technologies, not just anti-virus engines and heuristics, but IDS and/or firewall, integrity management and other 'more obscure' features, such as tarpiting, baiting, luring or honeypots. The other option will probably be that outsourcing companies may well offer this type of 'solution' as a managed 'anti-malware'service.

[15] P2P covers not just file-sharing tools like KaZaA, WinMX, Gnutella, etc. but also Instant Messaging clients and IRC too.

[16] There are other worms that fall outside this 'unholy trinity' but they are 'true' file-less worms like CodRed, SQLSlammer, and their kin.

6.2 Solutions?

Is there anything that can be done to slow or hobble 'SMB' aware worms and other malware?

Below are some suggestions that may be useful[5].

- Patch!
  Many worms will use known exploits to breach the security of a victim, so ensure that all systems are patched as soon as is feasible.
- Don't Share.
  Don't use null session or open shares on Windows systems. If you must share part of your system then share just the specific folder, rather than the whole hard drive, and never share the 'system[17]'directories.
- If you must Share, use good complex passwords, at least eight characters and include non-alphanumeric characters as well as alphanumeric. Do not use dictionary words.
  A number of recent worms use password/userid lists to perform a dictionary style attack on systems and password protected shares.
  The same rules should be applied to 'default administrator' shares and 'all user accounts too.
- If you must Share, use 'User-Level access control' rather than 'Share-Level access control'.
- If you must Share, make the share readable only, not writeable!
- Unbind Netbui/Netbios[18].
  If you don't need 'Windows Networking' then disable it, and use other network protocols, such as IPX instead. This is particularly important on systems that use modems, cable or xDSL connections into the corporate network. Otherwise you may become 'patient zero' for a new share-aware worm attacking you from the internet, and then travelling to your corporate network via your remote session, even if it is a VPN.
- Block SMB (137/udp, 139/tcp, 445/tcp and udp).
  Add rules to your corporate firewall to disallow inbound and outbound SMB traffic to/from your internal network. This will help to protect you from malware SMB traffic from the 'internet' If you use personal firewalls on your end-user systems, set a default rule to disable inbound SMB.
- Deploy SMB-Lures.
  This is a useful and relatively low-maintenance early-warning-system for new share-aware malware. It can even trap BOTs, Backdoors and other RATs. This should be seen as just one part of your anti-malware defences.
- Modify the SAMBA source-code to 'throttle'[6] the SMB communication process down, effectively turning it into an SMB tarpit.
- Educate your staff about the changes in Malware that they don't just get in via e-mail (as many of them think this is the only vector now).

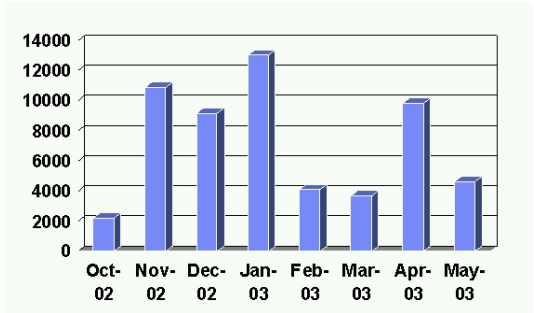
[17] Such as 'Windows', 'WinNT' and also directories that are above them (%WINDIR%).
[18] See http://www.mikeshardware.com/howtos/howto_disable_netbios.html

7 Statistics

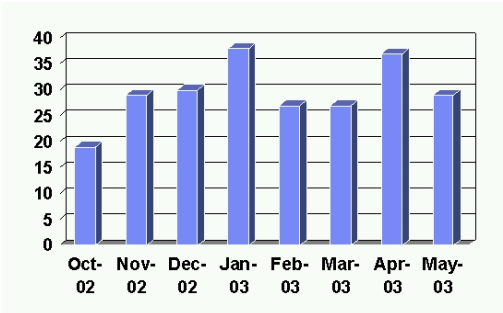
The following statistics cover the period from October 2002 to the end of May 2003. When the paper is presented I will have further data up to at least the end of August 2003.

Nevertheless, some interesting patterns and findings have come to light within the existing data. These are covered below:

(1) Total Samples Trapped Per Month

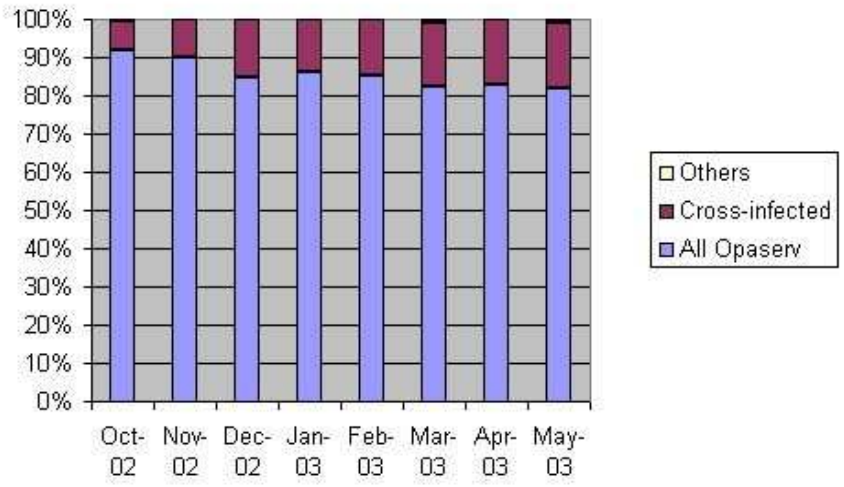


(2) Distinct Malware/Variants Trapped Per Month



The first pair of graphs show the totals for 'all samples trapped (1)' and the number of distinct worms (or other malware) and variants (2) thereof. A you can see the peak month was January 2003 with over 13,000 samples trapped which were comprised of 38 distinct malware strains/variants.

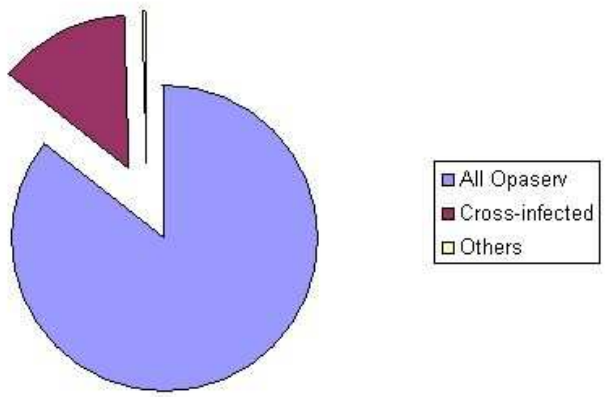
(3) Total Samples Trapped –By Type and Month



As you can see from (3) the number of cross-infected samples (by another parasitic infector) arriving has been steadily rising. In May it peaked at 19 percent. By far the greatest carriers were the many, many Opaserv variants.

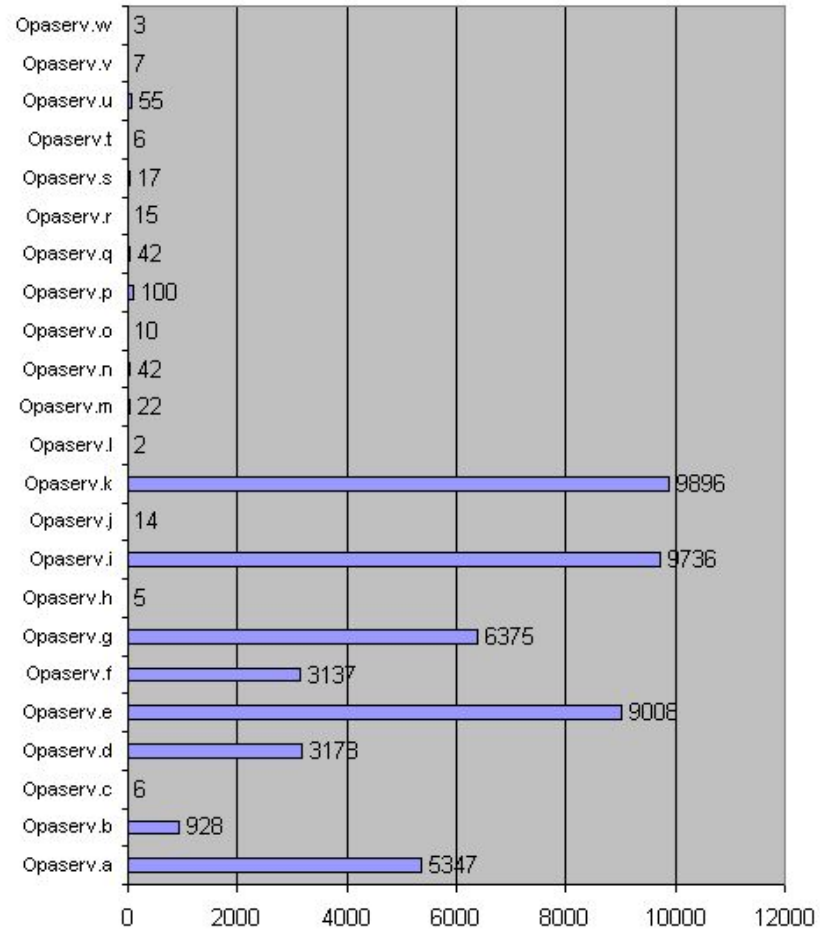
Another view of the data can be seen below in (4) which show the percentage for each type for the whole period.

(4) Total Samples Trapped –By Type (Oct 2002-May 2003)



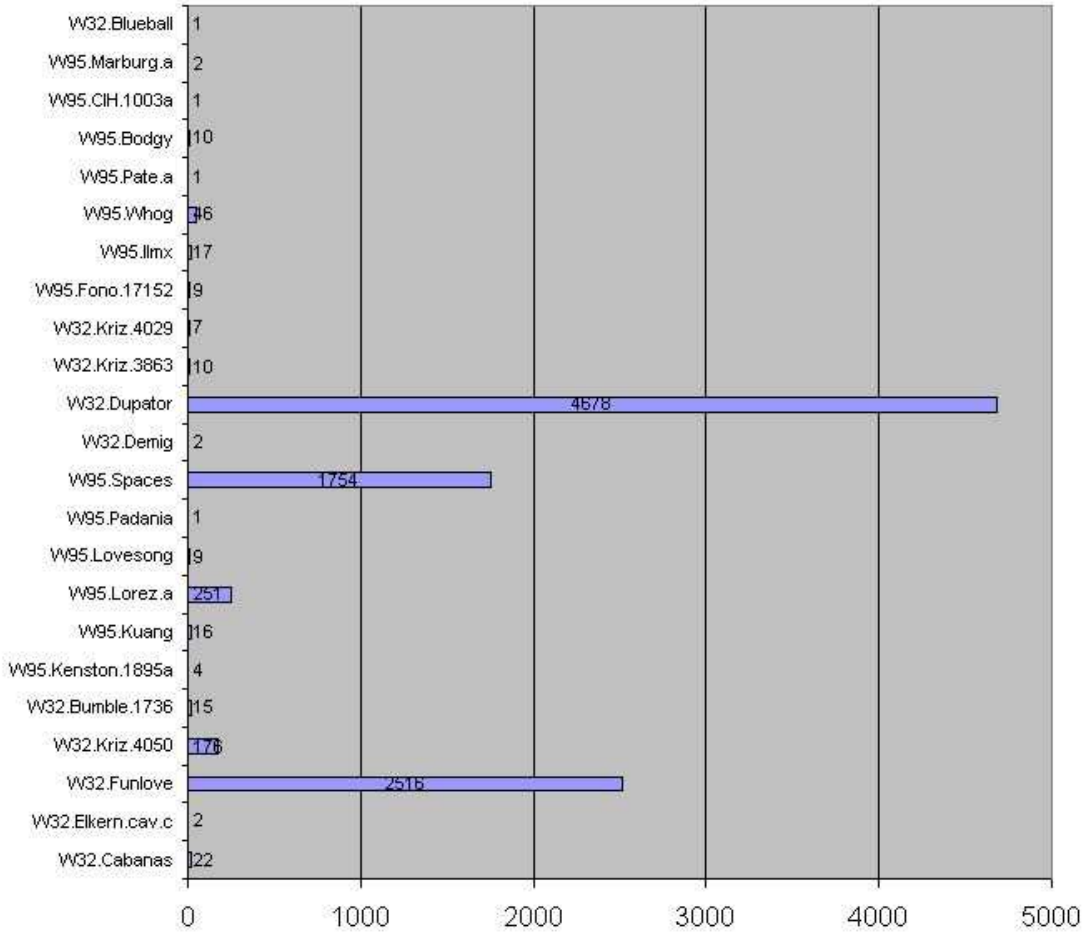
The following graph (5) shows the breakdown of all Opaserv variants trapped as of the end of May 2003. As you can see the most successful variants are E, I and K. The second most successful variants were the A and G variants.

(5) Opaserv Variants (Totals) Captured (Oct-02 to May-03) 47,951 samples



A number of the malware samples trapped have been real surprises, such as those that are not supposed to be 'in-the-wild'. All passenger malware appear in the graph below (6). As can be seen the most common passengers are: W32.Dupator, W32.Funlove and W95.Spaces.

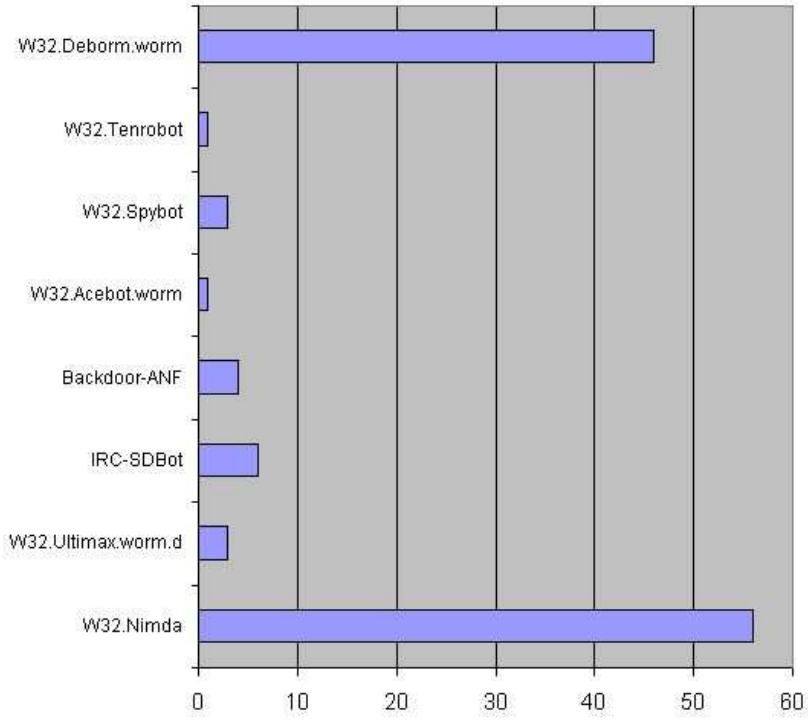
(6) Passenger malware on the 'Opaserv' malware transit system (Totals) Captured (Oct-02 to May-03) 9,550 samples



Recently a number of BOTS, Backdoors, Trojans, and other malware have started to be deployed via open Windows shares, this is a worrying trend as the possibilities for BOT armies to be quickly built is increased as is the use of Backdoors for 'information gathering' for corporate or industrial espionage.

The captured BOTS, Backdoors and Trojans can be seen in the graph below (7)

(7) Other Malware (Totals) Captured (Oct-02 to May-03) 120 samples



8 Lessons Learnt

- Use ‘real’ files rather than ‘touch’ created files.
- Integrity check the ‘bait’ files on a regular basis, say once an hour. This will identify files that do not have their ‘create’ or ‘last modified’ dates/times changed when infected as well as malware not covered by the SMB.LOG scripts.
- Use MD5 hash tables for static, rather than polymorphic worms.
- Use anti-virus to confirm dropped/modified files on a regular basis to confirm infection.
- Constantly review bait directory structure and file types and names to ensure maximum number of worms can be attracted.
- Can be used to capture not just worms, but Bots, Backdoors and other malware too.
- Keep a ‘clean’ backup of bait files and directory structure, so that the active ‘open share’ can be rebuilt quickly and easily.
- Make the ‘open share’ readable and writeable otherwise no files will be infected/dropped. Just because the SMB.CONF says read/write on the share doesn’t change the *NIX rights.
- Run SAMBA via Xinetd rather than as daemons (-D) as this will allow you to run it on multiple ports.
- Stuck files - With version 2.1.x of SAMBA running in daemon mode, files would get occasionally ‘stuck’. Manually ‘un-stick’ files by killing relevant PID listed in SMBSTATUS. This ‘feature’ seems to be fixed in versions 2.2.7 or later, certainly not a problem when it is run via Xinetd rather than as a daemon.
- Add IDS and other tools to the mix to round out detection of new threats.
- Excellent as one of a many early warning systems. Can give you a head-start on the next ‘big’ malware threat. The quicker you see it the faster you can block it.

9 Conclusions

- SMB-Lure (either the original or extended flavour) is well worth using as part of an early-warning system or for collecting new 'in-the-wild' share-aware malware and blended-threats that have Windows share capabilities.

- Useful for collecting statistics of infected systems, but the data should not be taken in isolation, must be merged with data from other vectors, such as e-mail, P2P, IRC, and other vectors to give a more accurate picture of the malware problem and which threats are most widespread.

- Can be integrated with other security/anti-malware technologies, such as Anti-Virus, Intrusion Detection System and Integrity management with very little work, and to great effect.

- Can be easily extended to become a semi or fully automated sample capture tool.

- At the time of writing this paper, over 60,000 samples had been captured using the prototype system. This includes a number of previously unknown malware strains/variants; it is believed that some were caught within a few hours of their initial deployment by their author(s).

- When used on an internal network, a web of geographically placed SMB-Lures will provide granularity on the entry point and spread of new share-aware malware.

- Multi-vector threats and worm-enabled BOTs and their kin are here to stay for the foreseeable future, so best be prepared.

- Once the system has both in-line virus scanning and near-real-time statistics then its usefulness will be increased tenfold.

10 Thanks and Feedback

I would like to thank John Morris for his work on the original SMB-Lure design, Paul Schmehl for his excellent shell and Perl scripts, Dr. Igor Muttik of NAI for his help in supplying some of the MD5 hash information for the Opaserv variants and also processing new generic samples and new variants as quickly as I trapped them. Thank also go to all AVIEN (and AVIEWS) members for their support and the IBM Virus Emergency Response Team.

All constructive feedback and suggestions for further features and improvements of the prototype design presented in this paper will be warmly received.

11 Appendix A: Example Internet Facing SMB.CONF file

```
# Samba config file created using SWAT
# from Wormfood (127.0.0.1)
# Date: 2002/10/17 20:41:32

# Global parameters

[global]
        workgroup = LURE
        netbios name = WORMBAIT
        netbios aliases = 000-InfectMe 000-DEAD 000-DoS-Box 000-EatMe 000-DrinkMe 000-
wormfood A00-wormfood B00-wormfood C00-wormfood D00-wormfood E00-wormfood W0rmF00d
M00-wormfood Z00-wormfood
        server string = Worm Detector - PLEASE DO NOT TOUCH
        security = SHARE
        debug level = 3
        log file = /var/log/samba/%m.log
        max log size = 0
        announce version = 4.0
        socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
        preferred master = Yes
        domain master = Yes
        dns proxy = No
        wins support = Yes

[C]
        comment = Worm Lure C, Please DON'T Touch
        path = /home/wormbait
        writeable = Yes
        guest ok = Yes

[C$]
        comment = Worm Lure C$, Please DON'T Touch
        path = /home/wormbait
        writeable = Yes
        guest ok = Yes


[ADMIN$]
        comment = Worm Lure ADMIN$, Please DON'T Touch
        path = /home/wormbait
        writeable = Yes
```

12 Appendix B – 'Bait' File and Directory List

This is not included in the paper as the list is over thirty pages long. To obtain this list, either e-mail me, or the list will be made available at: http://archnid.homeip.net:81 Please contact me for access to the site.

References:

[1] Fighting Worms in a Large Corporate Environment: A Design for a Network Anti-Worm Solution – Proceedings of the 12th Virus Bulletin International Conference 2002, pp 56 - 66
[2] Attributed to Simon Widlake
[3] Taken from John Morris's web page on SMB-Lure.
[4] See Frederic Perriot's article in Virus Bulletin December 2002 pp 6 – 8 entitled 'Crack Addict' for a good analysis of Opaserv.a
[5] Some of these are taken from my article 'Are You Being [Opa]Serve[d]? – Virus Bulletin January 2003, pp 10 - 13
[6] See the Virus Bulletin March 2003 pp 8 - 11article entitled 'Virus Throttling' by Mathew Williamson, et al.