

Windows Everywhere and Still Can't See a Thing!

This paper is designed to assist network security personnel in identifying what specific Windows Operating System (OS) servers and workstations are connected to the internal network without relying on or needing authenticated access to the individual systems. This will also help to identify any rogue or otherwise unauthorized Windows systems that are connected to the internal network.

We will:

- 1) Talk briefly about why it is important to know what systems are connected to the internal network (duh),
- 2) Discuss the tools that are available that uniquely identify Windows OS versions, go over the pros and cons for each tool, review the circumstances that may limit or restrict their results and
- 3) Provide a script for the tools that look most promising which can serve as a starting point upon which to build.

In the end, we will learn there is no single freeware tool available that can scan a range of IP addresses and without any authentication, provide a list of Windows systems with their OS version and differentiate whether they are a server or a workstation. This paper will provide sufficient knowledge to execute some of the tools that are available today and show how to use them (with or without a script).

Background:

Many companies have large networks which give them competitive advantage by increasing information delivery capabilities and reducing the time it takes to complete the delivery of critical company data. Some of these networks are very complex, span multiple Class B network address ranges and are physically located in many different places around the world.

As networks become larger, more complex and increasingly remote, it is much more difficult to identify unauthorized servers and systems. Sometimes these systems don't get our attention until they start causing problems on the network or to other systems. For example, if/when an unauthorized/unknown system gets infected with Nimda, it may take up all the bandwidth for the firewall which would subsequently affect Public Internet access for valid authorized internal systems. Therefore, for this and many other reasons, it is important to know what systems are connected to the network and the risks they pose to the network at large and other network-connected hosts. This paper will focus on enumerating network-connected Windows platforms.

There are several tools that have OS fingerprinting capabilities. With all of the tools that perform OS fingerprinting, we would think determining which version of Windows is running on a particular system would not be that hard. Actually, this is not very easy

and we will discuss why later in this document. When someone says “OS fingerprinting” most people will say, “huh”. The rest of us say, Nmap. As such, this paper will describe some of the more popular tools (Nmap and XProbe2) and some that provide Windows specific functionality that close gaps that Nmap and XProbe2 have (NBTScan, NBTStat and BrowStat).

Fingerprinting Tools and Descriptions:

Nmap:

The Nmap tool is a full fledged network discovery and enumeration scanner. Here is an excerpt from the insecure.org (the sponsor web site for Nmap):

Nmap ("Network Mapper") is an open source utility for network exploration or security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (ports) they are offering, what operating system (and OS version) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. Nmap runs on most types of computers, and both console and graphical versions are available. Nmap is free software, available with full source code under the terms of the GNU GPL.¹

For the purpose of this paper, I will not be going into details about how Nmap can be used except to discuss how to use Nmap for OS fingerprinting, its strengths and weaknesses. For more details on what port scanning is and how to use Nmap, Tim Corcoran has written an excellent paper on this topic called “An Introduction to NMAP”² at the SANS Reading Room.

OS fingerprinting functionality is provided by using the Nmap `-O` switch. The Nmap man page for this switch follows:

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

If Nmap is unable to guess the OS of a machine, and conditions are good (e.g. at least one open port), Nmap will provide a URL you can use to submit the fingerprint if you know (for sure) the OS running on the machine. By doing this you contribute to the pool of operating systems known to nmap and thus it will be more accurate for everyone.

1. Fyodor. “Nmap”.
2. Corcoran, Tim. “An Introduction to NMAP”.

The -O option also enables TCP Sequence Predictability Classification. This is a measure that describes approximately how hard it is to establish a forged TCP connection against the remote host. This is useful for exploiting source-IP based trust relationships (rlogin, firewall filters, etc) or for hiding the source of an attack. The actual difficulty number is based on statistical sampling and may fluctuate. It is generally better to use the English classification such as "worthy challenge" or "trivial joke".³

Nmap ports to most UNIX platforms and requires root privilege for several commands. Because OS fingerprinting uses many critical kernel interfaces (e.g. raw sockets), OS fingerprinting required root level authority on the system. Nmap uses many different techniques for making its guess for the OS. They include: The FIN probe, The BOGUS flag probe, TCP ISN Sampling, IPID sampling, TCP Timestamp, Don't Fragment bit, TCP Initial Window, ACK Value, ICMP Error Message Quenching, ICMP Message Quoting, ICMP Error message echoing footprints for over 500 integrity, Type of Service, Fragmentation Handling and TCP Options.⁴ Nmap can identify numerous versions of UNIX, routers, printers and other network connected devices.

Here is an example of a valid OS fingerprinting Nmap command and the resulting output:

```
[root@#####]# nmap -sS -O 192.168.x1.x1

Starting nmap V. 2.54BETA7 ( www.insecure.org/nmap/ )
Interesting ports on test1.sample.com (192.168.x1.x1):
(The 1530 ports scanned but not shown below are in state: closed)
Port      State      Service
21/tcp    open      ftp
135/tcp   open      loc-srv
139/tcp   open      netbios-ssn
3389/tcp  open      msrdp

TCP Sequence Prediction: Class=random positive increments
                        Difficulty=12443 (Worthy challenge)
Remote operating system guess: Windows 2000 Professional, Build 2183 (RC3)

Nmap run completed -- 1 IP address (1 host up) scanned in 4 seconds
```

Of note, the particular system scanned above is running Windows 2000 Advanced Server, but Nmap reports the Windows 2000 Professional class system. This is one of the drawbacks of Nmap regarding identification of Windows systems.

Nmap results are even more generic for older versions of Windows (Windows 95, 98 and NT 4) as the OS guesses are all lumped together. For background and information, here is an example of the results for a Windows NT 4 server:

3. Fyodor. "Nmap network security scanner man page".
4. Fyodor. "Remote OS Detection via TCP/IP Fingerprinting".

```
[root@#####]# nmap -sS -O 192.168.x2.x2

Starting nmap V. 2.54BETA7 ( www.insecure.org/nmap/ )
Interesting ports on test2.sample.com (192.168.x2.x2):
  (The 1524 ports scanned but not shown below are in state: closed)
Port      State      Service
80/tcp    open       http
135/tcp   open       loc-srv
139/tcp   open       netbios-ssn
443/tcp   open       https
1032/tcp  open       iad3
2301/tcp  open       compaqdiag

TCP Sequence Prediction: Class=trivial time dependency
                        Difficulty=2 (Trivial joke)
Remote operating system guess: Windows NT4 / Win95 / Win98

Nmap run completed -- 1 IP address (1 host up) scanned in 4 seconds
```

According to the Fyodor, Nmap's author, the NT stack remained essentially unchanged from Windows 95 to early versions of Windows NT 4.

... nmap is unable to distinguish between the TCP stacks of Win95, WinNT, or Win98. This is rather surprising, especially since Win98 came out about 4 years after Win95. You would think they would have bothered to improve the stack in some way (like supporting more TCP options) and so we would be able to detect the change and distinguish the operating systems. Unfortunately, this is not the case. The NT stack is apparently the same crappy stack they put into '95. And they didn't bother to upgrade it for '98.⁴

Summary:

Nmap provides very robust OS fingerprinting functionality except for deciphering between different versions of Windows platforms. For example, if the need is to differentiate between Windows NT 4 workstation and Windows NT 4 server, Nmap is probably not the best tool to use.

Xprobe2:

Xprobe2 is a cutting edge tool whose primary function was designed to perform OS fingerprinting using only ICMP. The tool uses "fuzzy" logic, similar to the logic used in Optical Character Recognition (OCR) technology, to perform statistical calculations and other mathematical algorithms to assign weighting factors for the OS guess.⁵

This tool is ideal for hackers since many systems are configured to respond to ICMP with little to no logging and some firewalls or router access controls lists (ACL) are configured to allow ICMP traffic to pass. Therefore, a hacker could send a single ICMP echo, probably not be logged, possibly get past firewalls or router ACL's and still obtain some information regarding the OS running on the target host.

4. Fyodor. "Remote OS Detection via TCP/IP Fingerprinting".

5. Arkin, Ofir and Yarochkin, Fyodor. "Xprobe v2.0".

Xprobe2 ports to most UNIX platforms and also requires root privilege access to run successfully. The tool can be obtained at <http://www.sys-security.com/>. It has built-in footprints in its database (xprobe2.conf) for 18 different server platforms with the instructions and functionality to easily add more to suit individual needs.⁶

Here is sample output for a Redhat Linux 7.0 target host:

```
[root@#####]# xprobe2 -v 192.168.x3.x3

XProbe2 v.0.1 Copyright (c) 2002 fygrave@tigerteam.net, ofir@sys-security.com
[+] Target is 192.168.x3.x3
[+] Loading modules.
[+] Following modules are loaded:
[x]ICMP echo (ping)
[x]TTL distance
[x]ICMP echo
[x]ICMP Timestamp
[x]ICMP Address
[x]ICMP Info Request
[x]ICMP port unreachable
[+] 7 modules registered
[+] Initializing scan engine
[+] Running scan engine
[+] Host: 192.168.1.200 is up (Guess probability: 100%)
[+] Target: 192.168.1.200 is alive
[+] Primary guess:
[+] Host 192.168.1.200 Running OS: "Microsoft Windows 2000/2000SP1/2000SP2"
(Guess probability: 68%)
[+] Other guesses:
[+] Host 192.168.1.200 Running OS: "Microsoft Windows XP Professional" (Guess
probability: 68%)
[+] Host 192.168.1.200 Running OS: "Microsoft Windows ME" (Guess probability:
63%)
[+] Host 192.168.1.200 Running OS: "Microsoft Windows NT 4 Service Pack 4 and
Above" (Guess probability: 59%)
[+] Host 192.168.1.200 Running OS: "NetBSD 1.5.2" (Guess probability: 59%)
[+] Cleaning up scan engine
[+] Modules deinitialized
[+] Execution completed.
```

For some systems, the number of guesses provided can be quite numerous.

Summary:

Xprobe2 is a new, cutting edge tool that can provide stealthy OS fingerprinting functionality. The results are, if you pardon the pun, fuzzy and not very precise. Therefore Xprobe2 is probably not the best tool to use for the circumstances which we are looking.

6. Arkin, Ofir and Yarochkin, Fyodor. "Sys-Security.com"

NBTStat:

NBTStat is a Microsoft provided troubleshooting tool that is available on all Windows platforms. NBTStat queries a single system for the NetBIOS name information and displays the results. There is not a port of this tool to other non-Windows platforms.

NBTStat is a work horse for managing the NetBIOS for a single system. For background and information, here is the help screen for NBTStat that lists all of the available switches and options:

Displays protocol statistics and current TCP/IP connections using NBT (NetBIOS over TCP/IP).

```
NBTSTAT [ [-a RemoteName] [-A IP address] [-c] [-n]
          [-r] [-R] [-RR] [-s] [-S] [interval] ]
```

```
-a (adapter status) Lists the remote machine's name table given its
                        name
-A (Adapter status) Lists the remote machine's name table given its
                        IP address.
-c (cache)           Lists NBT's cache of remote [machine] names and
                        their IP addresses
-n (names)           Lists local NetBIOS names.
-r (resolved)        Lists names resolved by broadcast and via WINS
-R (Reload)          Purges and reloads the remote cache name table
-S (Sessions)        Lists sessions table with the destination IP
                        addresses
-s (sessions)        Lists sessions table converting destination IP
                        addresses to computer NETBIOS names.
-RR (ReleaseRefresh) Sends Name Release packets to WINS and then,
                        starts Refresh
```

RemoteName Remote host machine name.

IP address Dotted decimal representation of the IP address.

interval Redisplays selected statistics, pausing interval seconds between each display. Press Ctrl+C to stop redisplaying statistics.

Here is a sample command line showing how to query for remote host (192.168.x4.x4) NetBIOS data from a source host (192.168.x1.x1) and what those results look like:

```
C:\>nbtstat -A 192.168.x4.x4
```

Local Area Connection:

Node IpAddress: [192.168.x1.x1] Scope Id: []

NetBIOS Remote Machine Name Table

| Name | Type | Status |
|---------------|-------------|------------|
| ANTHONY | <00> UNIQUE | Registered |
| DIRECTNET | <00> GROUP | Registered |
| ANTHONY | <03> UNIQUE | Registered |
| ANTHONY | <20> UNIQUE | Registered |
| DIRECTNET | <1E> GROUP | Registered |
| DIRECTNET | <1D> UNIQUE | Registered |
| .._MSBROWSE_. | <01> GROUP | Registered |

MAC Address = 00-05-5D-xx-xx-xx

The Local Area Connection:/Node... section signifies which TCP/IP interface is being reported. Each TCP/IP interface has separate sections in the results with the output of the different NetBIOS queries from each interface listed in their respective sections.

NBTStat output lists the NetBIOS data in a table on separate lines for each NetBIOS service running on the target host. Each line in the table contains data for Name, Type and Status columns. These three fields combine to make up a designator for each NetBIOS service. The translation table and process is described in detail at <http://www.neohapsis.com/resources/wins.htm>. Here is a portion of the translation tables to assist in the example that follows:

| Name | Number Hex (0x) | Type U=Unique G=Group | Usage |
|-----------------|-----------------|-----------------------------|---------------------|
| <computer_name> | 00 | U | Workstation Service |

Table 1: Small Sample of Registered NetBIOS Names for WINS⁷

The Name field is made up of 16 bytes where the first 15 bytes are alphanumeric and signify either the Server name, Domain name or another special name. The 16th byte is a hex digit that identifies the resource type. The Type field will be one of 2 entries: Unique or Group. The Status field lists whether the NetBIOS service is registered and is not really germane to the topic at hand.⁷

Here is an example; In the output above, the line “ANTHONY <00> UNIQUE” signifies the Computer Name is “Anthony” and the “Workstation Service” is being published.

NBTStat can be used to query NetBIOS information for a single IP address. For example, NBTStat could be used to identify whether a system is a Domain Controller. This could be used to confirm a system is a server (since all Domain Controllers are servers), but can not be used to exclude systems from the server list (since all servers are not necessarily Domain Controllers).

Summary:

NBTStat can be used to query single IP addresses for NetBIOS information that provides some information that could lead to identifying whether a system is a server. The process is quite cumbersome (single IP address at a time) and manual (have to look up specific NetBIOS service descriptions manually) in nature. As such, NBTStat is probably not the best tool to use for the circumstances which we are looking.

7. Kastl, Rebecca. “WINS”.

NBTScan:

NBTScan is NBTStat on steroids. NBTScan can query NetBIOS name information for multiple systems and display the results in a useful format. NBTScan ports to several platforms, to include, Win32 (Windows NT 4, 2000, XP) and various UNIX platforms (FreeBSD 4.3, OpenBSD 2.8 and RedHat Linux 7.1).⁸ There are a couple of different versions of NBTScan that have been developed.

One was written by Steve Friedl of Unixwiz.net. Mr. Friedl wrote this tool “because the existing tools either didn't do what {he} wanted or ran only on the Windows platforms: {NBTScan} runs on just about everything.”⁹ More information can be found at and the tool can be downloaded from <http://www.unixwiz.net/tools/nbtscan.html>.⁹

The other was written by Alla Bezroutchko supports a few more features (output delimiter, bandwidth throttling, etc.) than Mr. Friedl's version, so this is the version that will be described below. More information can be found at and the tool can be downloaded from <http://www.inetcat.org/software/nbtscan.html>.⁸

For background and information, here is the help screen for NBTScan which lists all of the available switches and options:

```
NBTscan version 1.0.3. Copyright (C) 1999-2000 Alla Bezroutchko.
This is a free software and it comes with absolutely no warranty.
You can use, distribute and modify it under terms of GNU GPL.
```

Usage:

```
nbtscan [-v] [-d] [-e] [-l] [-t timeout] [-b bandwidth] [-r] [-q] [-s
separator] [-m retransmits] (-f filename)|(<scan_range>)
  -v          verbose output. Print all names received
              from each host
  -d          dump packets. Print whole packet contents.
  -e          Format output in /etc/hosts format.
  -l          Format output in lmhosts format.
              Cannot be used with -v, -s or -h options.
  -t timeout  wait timeout seconds for response.
              Default 1.

  -b bandwidth  Output throttling. Slow down output
              so that it uses no more that bandwidth bps.
              Useful on slow links, so that outgoing queries
              don't get dropped.
  -r          use local port 137 for scans. Win95 boxes
              respond to this only.
              You need to be root to use this option on Unix.
  -q          Suppress banners and error messages,
  -s separator Script-friendly output. Don't print column and
              record headers, separate fields with separator.
  -h          Print human-readable names for services.
              Can only be used with -v option.
  -m retransmits  Number of retransmits. Default 0.
  -f filename  Take IP addresses to scan from file filename
```

8. Bezroutchko, Alla. “NBTScan. NetBIOS Name Network Scanner.”.

9. Friedl, Steve. “nbtscan - NETBIOS nameserver scanner”.

<scan_range> what to scan. Can either be single IP like 192.168.1.1 or range of addresses in one of two forms: xxx.xxx.xxx.xxx/xx or xxx.xxx.xxx.xxx-xxx.

Examples:

```
nbtscan -r 192.168.1.0/24
Scans the whole C-class network.

nbtscan 192.168.1.25-137
Scans a range from 192.168.1.25 to 192.168.1.137

nbtscan -v -s : 192.168.1.0/24
Scans C-class network. Prints results in script-
friendly
format using colon as field separator.
Produces output like that:

192.168.0.1:NT_SERVER:00U
192.168.0.1:MY_DOMAIN:00G
192.168.0.1:ADMINISTRATOR:03U
192.168.0.2:OTHER_BOX:00U
...

nbtscan -f iplist
Scans IP addresses specified in file iplist.
```

Now let's look at some command line examples and the resulting output to see how this might help us in our quest:

```
C:\>nbtscan 192.168.x5.x5
Doing NBT name scan for addresses from 192.168.x5.x5
```

| IP address | NetBIOS Name | Server | User | MAC address |
|---------------|--------------|----------|---------|-------------------|
| 192.168.x5.x5 | ANTHONY | <server> | ANTHONY | 00-05-5d-xx-xx-xx |

This short output format gives the IP address of the target host, the server name, whether the system is running the server service and the MAC address of the system. This tool provides a method of detecting systems that are running the "Server" service within a particular IP address range. Since all Windows servers run this service, this is a good first step. The problem is all workstations that have File and Print Sharing enabled also have this service running.

To get detailed output, just add the -v switch:

```
C:\>nbtscan -v 192.168.x5.x5
Doing NBT name scan for addresses from 192.168.x5.x5
```

NetBIOS Name Table for Host 192.168.x5.x5:

| Name | Service | Type |
|-------------------|---------|--------|
| ANTHONY | <00> | UNIQUE |
| DIRECTNET | <00> | GROUP |
| ANTHONY | <03> | UNIQUE |
| ANTHONY | <20> | UNIQUE |
| DIRECTNET | <1e> | GROUP |
| DIRECTNET | <1d> | UNIQUE |
| •• __MSBROWSE__ • | <01> | GROUP |

Adapter address: 00-05-5d-xx-xx-xx

This information is very similar to the NetBIOS name information obtained using the NBTStat tool described above.

After some informal testing on the speed of this tool, it is pretty quick. NBTScan takes about 11 minutes to complete a scan of a Class B range of addresses (using no switches (i.e. short output), a 100 Mb connection and redirecting the output to a text file). It also appeared this was independent of the number of hosts found as well. Beware, results may vary! NBTScan is quick because it uses UDP to perform the scans which greatly improves speed and performance.⁸ Another advantage of NBTScan is its ability to throttle bandwidth utilization (see the `-b` switch on the NBTScan website) in case that is needed.

Summary:

NBTScan is getting closer to the tool we need since it provides fast scanning capabilities, has the ability to scan multiple hosts with a single command and can identify NetBIOS individual services that are operating. However, NBTScan still does not quite get the level of detail for OS identification that we need.

BrowStat:

In walks BrowStat. BrowStat is an NT 4 Microsoft Resource Kit tool that is powerful command-line browser monitoring and querying tool.¹⁰ We will use the view (VW) function of BrowStat to obtain a list of members in a domain along with OS details (which OS, what version) and the purpose of the system (PDC, BDC, Stand Alone Server, or none of these). This will help us find out what OS version is running and differentiate between whether a system is a server or a workstation. This is our ultimate goal.

For background and information, here is the help screen for BrowStat that lists the available switches and options:

```
Usage: BROWSTAT Command [Options | /HELP]
Where <Command> is one of:
```

```
ELECT      ( EL) - Force election on remote domain
GETBLIST   ( GB) - Get backup list for domain
GETMASTER ( GM) - Get remote Master Browser name (using NetBIOS)
GETPDC     ( GP) - Get PDC name (using NetBIOS)
LISTWFW    (WFW) - List WFW servers that are actually running browser
STATS      (STS) - Dump browser statistics
STATUS     (STA) - Display status about a domain
TICKLE     (TIC) - Force remote master to stop
VIEW       ( VW) - Remote NetServerEnum to a server or domain on
                  transport
```

8. Bezroutchko, Alla. "NBTScan. NetBIOS Name Network Scanner."

10. Microsoft, "158388 - Useful Resource Kit Utilities for Domain Administrators".

In server (or domain) list displays, the following flags are used:
W=Workstation, S=Server, SQL=SQLServer, PDC=PrimaryDomainController,
BDC=BackupDomainController, TS=TimeSource, AFP=AFPServer, NV=Novell,
MBC=MemberServer, PQ=PrintServer, DL=DialinServer, XN=Xenix,
NT=Windows NT, WFW=WindowsForWorkgroups, MFPN=MS Netware,
SS=StandardServer, PBR=PotentialBrowser, BBR=BackupBrowser,
MBR=MasterBrowser, DMB=DomainMasterBrowser, OSF=OSFServer,
VMS=VMSServer, W95=Windows95, DFS=DistributedFileSystem

Now let's look at some output and see how this can help us in our quest:

```
C:\>BrowStat vw \Device\NetBT_Tcpip_{103728A4-D1A6-454E-9936-27E0CAAB4115}
\\GO-TESTBB001
Remoting NetServerEnum to \\GO-TESTBB001 on transport
\Device\NetBT_Tcpip_{103728A4-D1A6-454E-9936-27E0CAAB4115} with flags
ffffffff
25 entries returned. 25 total. 1302 milliseconds

\\SAMPLE001      NT    04.00 (W,S,BDC,NT,BBR)
\\SAMPLE002      NT    04.00 (W,S,PDC,NT,MBR)
\\SAMPLE003      NT    05.00 (W,S,NT,SS,BBR,02000000)    Description1
\\SAMPLE004      NT    05.00 (W,S,NT,SS,BBR,MBR,02000000)
\\SAMPLE005      NT    04.00 (W,S,NT,SS,PBR)    Description2
\\SAMPLE006      NT    05.00 (W,S,NT,SS,MBR,02000000)
\\SAMPLE007      NT    05.00 (W,S,NT,SS,BBR,02000000)
\\SAMPLE008      NT    05.00 (W,S,NT,SS,BBR,02000000)
\\SAMPLE009      NT    05.00 (W,S,NT,SS,BBR,02000000)
\\SAMPLE010      NT    05.00 (W,S,NT,SS,BBR,MBR,02000000)
\\SAMPLE011      NT    05.00 (W,S,NT,SS,02000000)
\\SAMPLE012      NT    05.02 (S,NT,SS,DFS)
\\SAMPLE013      NT    05.00 (W,S,NT,SS,BBR,MBR,02000000)
\\SAMPLE014      NT    04.00 (W,S,NT,SS,PBR,MBR)
\\SAMPLE015      NT    05.00 (W,S,SQL,NT,SS,MBR,02000000)
\\SAMPLE016      NT    05.00 (W,S,SQL,NT,SS,BBR,02000000)
\\SAMPLE017      NT    05.00 (W,S,NT,SS,BBR,02000000)
\\SAMPLE018      NT    05.00 (W,S,NT,SS,BBR,02000000)
\\SAMPLE019      NT    04.00 (W,S,NT,SS,PBR)
\\SAMPLE020      NT    04.00 (W,S,NT,SS,PBR)
\\SAMPLE021      NT    05.00 (W,S,NT,SS,BBR,MBR,02000000)
\\SAMPLE022      NT    05.00 (W,S,NT,SS,BBR,02000000)
\\SAMPLE023      NT    04.00 (S,NT,SS)
\\SAMPLE024      NT    05.00 (W,S,DL,NT,SS,BBR,02000000)
\\SAMPLE025      NT    05.00 (W,S,NT,SS,BBR,02000000)
```

The output is quite voluminous for just giving a single system name. Here is a description of the pertinent portions of this output. The first couple of lines are information about the scan. The table section is segregated into 4 columns.

- The first is the system name (e.g. \\SAMPLE001).
- The second is the OS (e.g. NT, OS2, 95).
- The third is the version of the OS (e.g. 04.00 with an OS of NT would translate to Windows NT 4).
- The last column lists the flags for the system (see the help screen above for the entire list and what they mean).

The additional servers listed in the BrowStat output are those Windows systems that are in the same domain as the one for which we queried. This is both a bonus and a curse. Systems that are not part of a domain and do not respond to browser queries (see below for more on this topic) would not get detected by this tool. Additionally, when scanning systems on the network, two different systems that are in the same domain will respond with each system in their BrowStat output. This will create multiple entries in the BrowStat output for some systems.

The specific flags we are interested in are SS, PDC and BDC. These flags stand for Stand Alone server, Primary Domain Controller and Backup Domain Controller, respectively. Every server has one (and only one) of these flags set. Additionally, workstations do not have any of these flags set. This is how we will ultimately distinguish between server and workstation systems.

When a scan is performed on a system that does not respond to browser queries, error messages like this one are received:

```
C:\>browstat vw \Device\NetBT_Tcpip_{103728A4-D1A6-454E-9936-27E0CAAB41 15}
\\sample001
Remoting NetServerEnum to \\sample001 on transport \Device\NetBT_Tcpip_
{103728A4-D1A6-454E-9936-27E0CAAB4115} with flags ffffffff
Unable to remote API to \\sample001 on transport \Device\NetBT_Tcpip_{
103728A4-D1A6-454E-9936-27E0CAAB4115}: The service has not been started.
(130 milliseconds)
```

This indicates the system does not respond to browser queries. In most cases (I haven't found any exceptions yet), this would indicate the system is a workstation and not a server. Although we don't get any new OS data from the output, this is still useful data and can be used to assist in determining whether a system identified by another tool is a server or a workstation.

With all of this taken into consideration, there are still some problems that arise when trying to use this tool for our purposes. This tool is relatively slow compared to NBTScan. It would be impractical for an organization to scan a large network segment with BrowStat alone.

Summary:

BrowStat gives us the information we need (e.g. system name, OS, OS version and whether it is a server or a workstation). The data, however, is not in a usable in raw form (e.g. duplicate listings) or alone (e.g. missing systems). Additionally the scan itself is somewhat slow and not practical for large networks.

So Let's Pull this all Together:

So how do we get the data we need in a usable format? I wrote a Windows batch script that uses the NBTScan and BrowStat tools to scan an IP address range, manipulate the data and get it into a usable format. The script starts with NBTScan and feeds the output to BrowStat. The data is then scrunched and separated into server and workstation lists that are stored in separate files.

Why Batch, you ask? Frankly, that is the scripting language (I know – what scripting language) that I know best. The portion of the code that eliminates duplicates takes the longest time to run and is not very efficient. This (and probably the rest of the code) could probably be written to execute quicker in Perl. Again, I'm not that familiar with Perl, so I chose batch.

The batch code can be found in Addendum A. The script was developed and runs on Windows XP Professional with NBTScan version 1.0.3 and BrowStat from the NT 4 Resource Kit. What follows is a brief description of the flow of the script. If more detail is needed, please refer to the code itself. The code is commented quite heavily and should be pretty easy to follow.

Starting out, the script validates the input from the command line. If /? was passed, a help list is displayed. If /v was passed, the script status is displayed in detail as it executes. The IP address and netmask are checked to make sure they are valid. The base file name (variable "UserFile") to be used is also set early in the script. This name is based on today's date so multiple scan results across different days will not be overwritten by future scans. The validation routines will not catch all of the input errors, by any means, but should catch the most egregious.

Next the NBTScan get launched. The parameter "-s ," tells NBTScan to use the comma as the delimiter for the output and "-b 80000" sets the bandwidth to 80K (this may be changed/eliminated as desired). The results are ported to a temporary "1" file (Note: .ol is used as the file extension to stand for Output List – this may be changed based on personal desire).

The "1" file is parsed on comma's, greater than and lesser than signs. The < and > signs show up in the third and sometimes fourth field. They really play havoc on batch script processes. The options were to put quotes around the 3rd and 4th fields or just eliminate them altogether. The latter was chosen. At the same time, the data is segregated according to whether the 3rd field is "server" or not. If it is "server", the system is a "potential server" and is placed in a "PS" file. If not, it is definitely a workstation and placed in the "W1" file. After this is complete the "1" file is deleted.

Next the "PS" file is parsed and the 1st field is input into the BrowStat View command. The output is directed to the "BS" (BrowStat) temporary file. The next few sections perform some manipulation on the "BS" output. First it is sorted. This will be important to speed up the identification and elimination of duplicate system names. Next, header and error messages are stripped out of the "BS" output. Then duplicates are identified and discarded. The "T" file is used intermittently during these sections and deleted when no longer needed.

Now that we have a clean "BS" file, the "PS" file is crunched through again to look for systems that don't show up in the "BS" file. These systems are presumed to be workstations (placed in the W1 file) since they do not respond to browser queries. After this is complete the "PS" file is deleted.

Next each line in the "BS" file is searched for either "SS", "PDC" or "BDC". If the string is found, the system is a server and placed in the "S" file. If it is not present, the system is placed in the "W2" file. There are 2 different workstation output files because the data is in different formats (W1 is in NBTScan format and W2 is in BrowStat format). Combining them into could have been done with much more scripting, but this appeared to be sufficient.

The script then cleans up variables and exits.

All-in-all, the code is pretty kluge, but does do the job it was asked to do.

Conclusion:

There is no single freeware tool available that can scan a range of IP addresses and without any authentication, provide a list of Windows systems with their OS version and differentiate whether they are a server or a workstation. This paper should give sufficient knowledge of some of the tools that are available today and show how to use them (with or without a script) to assist in this endeavor. It also provides a starting point for scripting a scanning solution.

© SANS Institute 2003, Author retains full rights.

References

1. Fyodor. "Nmap -- Free Stealth Port Scanner For Network Exploration & Security Audits. Runs on Linux/Windows/UNIX/Solaris/FreeBSD/OpenBSD". 10 AUG 2002. <http://www.insecure.org/nmap/index.html> (23 DEC 2002).
2. Corcoran, Tim. "An Introduction to NMAP". 25 OCT 2001. <http://rr.sans.org/audit/nmap2.php> (23 DEC 2002).
3. Fyodor. "Nmap network security scanner man page". 2001. http://www.insecure.org/nmap/data/nmap_manpage.html (23 DEC 2002).
4. Fyodor. "Remote OS Detection via TCP/IP Fingerprinting". 11 JUN 2002. <http://www.insecure.org/nmap/nmap-fingerprinting-article.html> (23 DEC 2002).
5. Arkin, Ofir and Yarochkin, Fyodor. "Xprobe v2.0. A "Fuzzy" Approach to Remote Active Operating System Fingerprinting". AUG 2002. <http://www.sys-security.com/archive/papers/Xprobe2.pdf> (24 DEC 2002).
6. Arkin, Ofir and Yarochkin, Fyodor. "Sys-Security.com - Because Security is not Trivial". 2002. <http://www.sys-security.com/html/projects/X.html> (24 DEC 2002).
7. Kastl, Rebecca. "WINS". 5 JUL 1999. <http://www.neohapsis.com/resources/wins.htm> (24 DEC 2002).
8. Bezroutchko, Alla. "NBTScan. NetBIOS Name Network Scanner.". FEB 2002. <http://www.inetcat.org/software/nbtscan.html> (25 DEC 2002).
9. Friedl, Steve. "nbtscan - NETBIOS nameserver scanner". 27 MAY 2002. <http://www.unixwiz.net/tools/nbtscan.html> (26 DEC 2002).
10. Microsoft. "158388 - Useful Resource Kit Utilities for Domain Administrators". 28 MAR 2002. <http://support.microsoft.com/default.aspx?scid=kb;en-us;158388> (26 DEC 2002).

Addendum A

@Echo off

```
REM Author: Terry Hickman
REM
REM Revision History:
REM -----
REM 1.0 Wrote original job on 12/25/2002
REM -----
REM
REM Batch job to invoke NBTScan to scan a given IP address range for Windows
REM machines. All Windows hosts are identified. Limitations: This script
REM will NOT detect Windows systems that:
REM 1) Have Router ACL or a firewall restricting access to port 137
REM 2) Have RestrictAnonymous=2 in the registry
REM 3) Do not have a Gateway selected for their network connection AND the scanning
REM system is not on the same network segment.
REM
REM The output is provided in 3 files. Each contains a list of Windows platforms
REM that are connected to the network for a given IP address range. There are 2
REM Workstation file (W1 and W2) and one server file (S).
REM
REM Usage:
REM
REM WINDOWS ###.###.###.###[/**]
REM
REM /v Provide verbose output on the screen.
REM ###.###.###.### Specifies the IP address(es) to scan.
REM /** Optional netmask (1-31) for the IP address to scan.
REM
REM The output files will be located in the same directory as the batch file.
REM The file names are: %UserFile%W1.ol, %UserFile%W2.ol and
REM %UserFile%S.ol where the date will always be today's system date in
REM YYYYMMDD format.
REM
REM Sets Date environment variables for use in file names to keep different
REM days files separated.
Call :SetDate
REM
REM Establishes the temporary file name to be used during the batch.
Set UserFile=TEST%Year%%Month%%Day%
REM
REM Clears the temporary Date environment variables before exiting the batch.
Call :ClearDate
REM
REM Branch to help is switch is /?.
If "%1"=="-?" Goto :Help
If "%1"=="/?" Goto :Help
If "%1"=="-h" Goto :Help
If "%1"=="--help" Goto :Help
REM
REM Set verbose mode if the /v switch was passed.
If "%1"==" /v" Set Verbose=1
If "%Verbose%"=="1" Shift
REM
REM Test IP Address Input.
FOR /F "tokens=1-5 delims=/" %%a in ('Echo %1') Do Call :ValidateIP %%a %%b %%c %%d %%e
If "%Error%"=="1" Goto :Help
Set Error=
REM
REM Delete all old files that were generated for this day if they exist.
If Exist %UserFile%S.ol Del %UserFile%S.ol
If Exist %UserFile%W1.ol Del %UserFile%W1.ol
If Exist %UserFile%W2.ol Del %UserFile%W2.ol
REM
REM Run NBTScan using comma as the delimiter, throttle bandwidth to 80K
REM against the input IP address and pipe the output to an output file.
REM The output of NBTScan is comma delimited into the 5 fields. The fields are:
REM %%a - IP Address
```

```

REM %%b - NetBIOS Server Name
REM %%c - Is "<server>" is the server service is running.
REM %%d - The name of the user that was logged in at the time of the NBTSCAN.
REM %%e - MAC Address.
If "%Verbose%"=="1" Echo.
If "%Verbose%"=="1" Echo.
If "%Verbose%"=="1" Echo Performing NBTScan....
If "%Verbose%"=="1" Echo If any "Recvfrom failed: Connection reset by peer" errors are received,
If "%Verbose%"=="1" Echo they may be ignored. It is a "bug" in the NBTScan tool. See
If "%Verbose%"=="1" Echo http://www.inetcat.org/software/nbtscan.html for more information.
If "%Verbose%"=="1" Echo.
If "%Verbose%"=="1" Echo.
Echo.> %UserFile%1.ol
NbtScan -s , -b 80000 %1 >> %UserFile%1.ol
If "%Verbose%"=="1" Echo.
If "%Verbose%"=="1" Echo NBTScan Complete!

REM Parse the input and 1) Save every line that has the phrase <server> in it
REM to the "PS" (Potential Server) file and 2) Save every line that Does not have
REM the phrase <server> in it to the "W" (Workstation) file.
If "%Verbose%"=="1" Echo.
If "%Verbose%"=="1" Echo Separating NBTScan output....
Echo.> %UserFile%PS.ol
For /F "tokens=1-5 delims=,<>" %%a in (%UserFile%1.ol) Do If "%c"=="server" (Echo
%%a,%%b,%%c,%%d,%%e >> %UserFile%PS.ol) Else (Echo %%a,%%b,%%c,%%d >> %UserFile%W1.ol)
REM Clean up unneeded files.
If Exist %UserFile%1.ol Del %UserFile%1.ol
If "%Verbose%"=="1" Echo.
If "%Verbose%"=="1" Echo NBTScan output separated!
If "%Verbose%"=="1" Echo Systems that didn't have "<server>" in the third field have been placed
in
If "%Verbose%"=="1" Echo Workstation %UserFile%W1.ol file. All other systems have been placed in
If "%Verbose%"=="1" Echo the Potential Server %UserFile%PS.ol file.

REM Take the Potential Server input and send it to the BrowStat tool.
If "%Verbose%"=="1" Echo.
If "%Verbose%"=="1" Echo Performing BrowStat scan on potential servers....
Echo.> %UserFile%BS.ol
REM
REM
REM The Device will need to be modified to your settings.
REM You can determine the available Devices on your system by executing the
REM BrowStat STA
REM command.
REM
REM
FOR /F "tokens=1-5 delims=," %%a in (%UserFile%PS.ol) Do BrowStat VW
\Device\NetBT_Tcpip_{103728A4-D1A6-454E-9936-27E0CAAB4115} \\%%a >> %UserFile%BS.ol
If "%Verbose%"=="1" Echo.
If "%Verbose%"=="1" Echo BrowStat Scan complete!

REM Sort the data for easier handling and use.
If "%Verbose%"=="1" Echo.
If "%Verbose%"=="1" Echo Sorting the BrowStat data....
Sort %UserFile%BS.ol > %UserFile%T.ol
REM Swap the temporary file back to the BrowStat file.
Del %UserFile%BS.ol
Ren %UserFile%T.ol %UserFile%BS.ol
If "%Verbose%"=="1" Echo.
If "%Verbose%"=="1" Echo BrowStat data sorted!

REM Eliminate the erroneous output from BrowStat and dump the rest into a temporary file.
If "%Verbose%"=="1" Echo.
If "%Verbose%"=="1" Echo Cleaning up BrowStat scan data....
Echo.> %UserFile%T.ol
For /F "tokens=1-15* delims=," %%a in (%UserFile%BS.ol) Do If "%a" NEQ "Remoting" If "%a" NEQ
"Unable" If "%b" NEQ "entries" If "%b" NEQ "milliseconds)" Echo %%a %%b %%c
%%d~%%e~%%f~%%g~%%h~%%i~%%j~%%k~%%l~%%m~%%n~%%o >> %UserFile%T.ol
REM Swap the temporary file back to the BrowStat file.
Del %UserFile%BS.ol
Ren %UserFile%T.ol %UserFile%BS.ol

```

```

If "%Verbose%"=="1" Echo.
If "%Verbose%"=="1" Echo BrowStat scan data clean up complete!

REM Eliminate duplicates and place the output in a temporary file.
If "%Verbose%"=="1" Echo.
If "%Verbose%"=="1" Echo Eliminating duplicate systems from the list....
Echo.> %UserFile%T.ol
For /F "tokens=1-4* delims=,)" %%a in (%UserFile%BS.ol) Do Call :Process1 %%a %%b %%c %%d
Set Same=
REM Swap the temporary file back to the BrowStat file.
Del %UserFile%BS.ol
Ren %UserFile%T.ol %UserFile%BS.ol
If "%Verbose%"=="1" Echo.
If "%Verbose%"=="1" Echo Duplicate systems eliminated from the list!

REM Find systems that didn't have output in the BrowStat file and presume they
REM are Workstations.
If "%Verbose%"=="1" Echo.
If "%Verbose%"=="1" Echo Finding systems that aren't in the BrowStat output (presumed
Workstations)....
For /F "tokens=1-2* delims=, " %%a in (%UserFile%PS.ol) Do For /F "tokens=1* delims=: " %%x in
('Find /C "%b" %UserFile%BS.ol') Do If "%y"==" 0" Echo %%a,%%b,%%c >> %UserFile%W1.ol
REM Clean up unneeded files.
If Exist %UserFile%PS.ol Del %UserFile%PS.ol
If "%Verbose%"=="1" Echo.
If "%Verbose%"=="1" Echo Systems that weren't in the BrowStat output have been placed in the
If "%Verbose%"=="1" Echo Workstation %UserFile%W1.ol file.

REM Split out Servers to "S" and Workstations to "W2" files.
REM All servers will have either SS, PDC or BDC as a BrowStat flag.
If "%Verbose%"=="1" Echo.
If "%Verbose%"=="1" Echo Finding all entries in the BrowStat output that are servers....
FindStr "~SS ~PDC ~BDC" %UserFile%BS.ol > %UserFile%S.ol
If "%Verbose%"=="1" Echo.
If "%Verbose%"=="1" Echo Systems found by BrowStat that are servers have been placed in the
Server
If "%Verbose%"=="1" Echo %UserFile%S.ol file.
If "%Verbose%"=="1" Echo.
If "%Verbose%"=="1" Echo Finding all entries in the BrowStat output that are workstations....
FindStr /V "~SS ~PDC ~BDC" %UserFile%BS.ol > %UserFile%W2.ol
If "%Verbose%"=="1" Echo.
If "%Verbose%"=="1" Echo Systems found by BrowStat that are not servers have been placed in the
If "%Verbose%"=="1" Echo Workstation %UserFile%W2.ol file.
REM Clean up unneeded files.
If Exist %UserFile%BS.ol Del %UserFile%BS.ol

REM Processing in this program is completed... Exiting.
Set UserFile=
If "%Verbose%"=="1" Echo.
If "%Verbose%"=="1" Echo Processing complete.
Set Verbose=
Goto :EOF

:help
Echo Provides 3 output files containing a list of Windows platforms
Echo that are connected to the network for a given IP address range.
Echo.
Echo WINDOWS [/v] ###.###.###.###[/**]
Echo.
Echo /v Provide verbose output on the screen.
Echo ###.###.###.### Specifies the IP address(es) to scan.
Echo /** Optional netmask (1-31) for the IP address to scan.
Echo.
Echo The output files will be located in the same directory as the batch file.
Echo The file names are: %UserFile%W1.ol, %UserFile%W2.ol and
Echo %UserFile%S.ol where the date will always be today's system date in
Echo YYYYMMDD format. The W1 and W2 files contain a list of Workstations
Echo found and the S file contains the list of Servers found.
Echo.
Goto :EOF

```

```
:Process1
If "%1" NEQ "%Same%" Echo %1 %2 %3 %4) >> %UserFile%T.ol
Set Same=%1
Goto :EOF

:ValidateIP
REM The first 4 numbers should be 0-255.
REM The last number should be 0-32.
Set Error=0
If %1 GTR 255 Set Error=1
If %1 LSS 0 Set Error=1
If %2 GTR 255 Set Error=1
If %2 LSS 0 Set Error=1
If %3 GTR 255 Set Error=1
If %3 LSS 0 Set Error=1
If %4 GTR 255 Set Error=1
If %4 LSS 0 Set Error=1
If %5 GTR 31 Set Error=1
If %5 LSS 1 Set Error=1
If "%Error%"=="1" Echo Error: The IP address provided was invalid.
If "%Error%"=="1" Echo.
Goto :EOF

:SetDate
REM Assumes MM-DD-YYYY format.
REM Assumes date separator is "/".
For /F "tokens=1-4 delims=/" %i in ('Date /T') Do Call :SetEnvVars %i %j %k %l
Goto :EOF

:SetEnvVars
Set DayOfWeek=%1
Set Month=%2
Set Day=%3
Set Year=%4
Set Date=%1 %2/%3/%4
Goto :EOF

:ClearDate
Set DayOfWeek=
Set Month=
Set Day=
Set Year=
Set Date=
Goto :EOF
```

© SANS Institute 2003, Author retains full rights.