Interested in learning
more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

System Identification for Vulnerability Assessment

An independent method of taking an inventory of the devices currently attached to your network must be developed in order to adequately secure the entire network. You must know what you are defending to adequately defend it. This paper is a description of one company's journey using existing software utilities to identify the hardware and software that placed their network at risk. The writer says: "I am hopeful that the tools and ideas used here will help others construct a suite of tools that suits their needs and wi...

Copyright SANS Institute
Author Retains Full Rights

AD

ForeScout

System identification for vulnerability assessment

By Michael C. Harris

Many sources exist within the security and hacking community that discus knowing the enemy, but that is only half the equation. Many security gurus have forgotten that they must also know them selves. They must know the systems and resources they are to protect.

Sun Tzu - Art of War

Sun Tzu said:

Know not the other and know not yourself
Every fight is certain defeat.

Know not the other and yet know yourself
One victory for one defeat.

Know the other and know yourself
Fight one hundred battles without danger.

(http://www.mailsbroadcast.com/the.artofwar.3.htm) (2)

Reasons for creating the tools

One of the largest burdens for those of us tasked with assessing the risk of hack attack or computerized intrusion is identifying the clients and hosts that are currently connected to our network. “Knowing ourselves”.

Identifying systems is an especially difficult task in large corporate, educational or public networks. These networks usually have distributed asset management and technical support functions. As wired network access had become commonplace and wireless network access grows in popularity, it becomes more and more difficult to keep an up to date listing of the systems that comprise your network. Keeping tabs on legitimate connections to the network becomes increasingly difficult in large networks with decentralized inventory and asset management and varying levels of support. Manual asset and inventory systems usually have little detail about network connectivity or software. Even in organizations that have formal centralized inventories, both operating systems and network connectivity details are either not

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

part of the data available or simply not populated. In some cases even when databases of systems detail do exist they are not available to the risk assessors or intrusion detection staff.

One of the highest risks is a rogue machine on any network. These are the machines that are usually unknown to management and data security personnel and are outside the usual **Zones of risk** (1). These machines are many times outside the standard maintaince and support methods even though they have been setup by those very same support individuals. Because they are setup and forgotten about, these are the machines that are in many cases running out-of-date or vulnerable software. Many times running services like FTP, Mail, or ICQ that do not belong there. Worse yet running some remote control package like PCAnywhere, VNC, BackOraface or Sub7 and is being used as a back door gateway into your network. With this in mind an independent method of taking an inventory of the devices currently attached to your network must be developed.

If ignorant both of your enemy and yourself, you are certain to be in peril. (Sun Tzu anchient art of war) (1)

You must know what you are defending to adequately defend it. Many of these rogue machines are installed by the most technically savvy users and IT support personnel you have in your organization. Usually these systems are created as a workaround to policies and procedures that are not fully understood or enforced. What follows is a description of our journey over the past several months using existing software utilities to identify the hardware and software that places our network at risk. I am hopeful that the tools and ideas used here will help others construct a suite of tools that suits their needs and will save them some of the headaches and redesign time we have spent. We did not have a clear vision of the end product when we started, by publishing our journey I hope to save others some of that pain.

Using existing tools to build a better mousetrap

What we initially set out to do is produce a system using commonly available tools that will identify the systems on our network, document some of the security risks, and identify who has responsibility for those systems. The elements of importance for us were:

Figure 0.

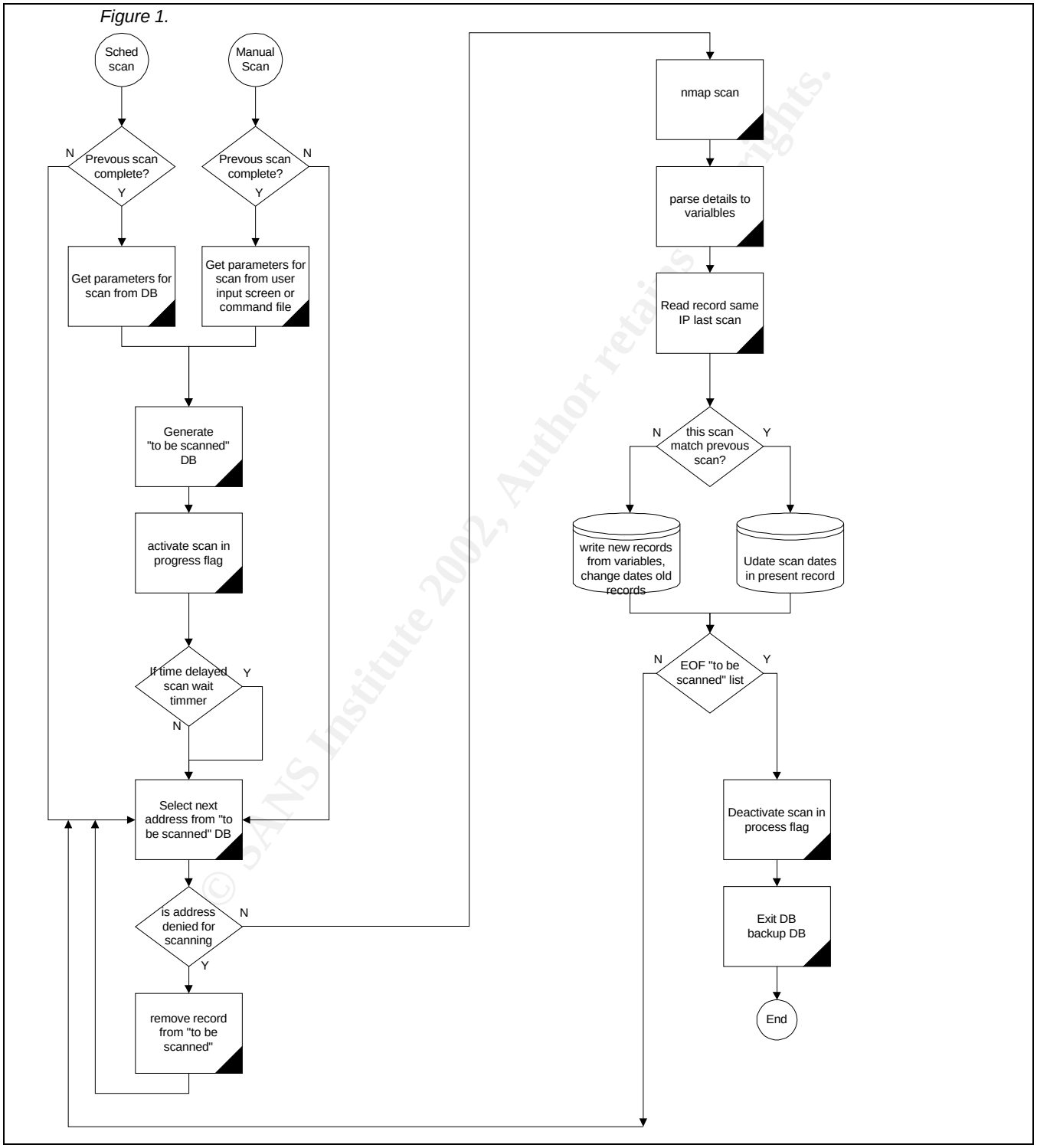
IP address
MAC address
DNS name
NetBIOS machine name
Network user ID
Operating system
Applications (services running)
User name
User department
User location
User phone
Support representative (entered as a function of department)
Security contact (entered as a function of department)

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

Vulnerability details

We have tried to use a model that is OS independent and can be hosted on either Linux or Windows NT systems. (Other Unix flavors probably also work well but we chose Redhat because it was what we were familiar with.) The basic flow of the original system is illustrated in figure.1.

Figure 1.



© SANS Institute 2002, Author retains full rights.

The entire system can be constructed on a single machine, but I would recommend having the SQL database on one machine, and the scanning software on another. The primary reason for this is speed. Both the database and scanning engine are processor and IO intensive processes and seem to work faster on multiple smaller x86 machines rather than a single large (fast) one. In theory many machines could be setup as scanning engines each assigned to a small physical or logical portion of the network, funneling data back to a central database. This would decrease scan times significantly and also allow some freedom within segmented or firewalled networks to scan addresses not reachable from a single point. This also allows the database and scanning engine to use different operating systems, in our case we had a production grade MS SQL server available to us and could then run the scanning engine Nmap (3), Samba and Nessus (10) on Linux. I believe the scanning tools run faster and give slightly better results when run under Linux because of the flexibility in the underlying IP services. The opposite can be run. MySQLx86™ for the database on Linux and the scanning can be Windows based using NmapNT (6) and Nessus (10) for Win32. These scanning tools are not as mature for the windows platform and these scanners are usually limited by the Microsoft IP stack. Activestate ™ Perl can be used on both platforms and has worked well for us. Add-ins for MS-SQL, MySQL™ and LDAP database connectivity are also available from the Activestate ™ website or can be linked to from there.

Our original scans with Nmap

I am indebted to H D Moore for his work on Nlog (5) and his scripts for putting Nmap(3) output into database format, they were the beginning of this thought process for me. A listing of our early database structure is provided in figure 2. Using an Nlog like structure for storage, we began doing repetitive Nmap scans of our Class B network and writing the output to a database for comparison against future scans. A list of tables is available at the end of this document.

FIGURE 2.

Master_DB_IP	Sub_DB_Port	Note
IP	Port	One exclusive master record for each IP.
IP1 octet 1	Port	Many port records, one for each port found
IP2 octet 2	Port status	for each IP address. (one to many)
IP3 octet 3	Port type	This has evolved into db table 6 IP Master
IP4 octet 4	start date	
DNS	last update	
OS detected	end date	
Start date		
Last update		
End date		

We soon found that Nmap scanning causes problems with several older versions of AIX, it also makes older HP Jet Direct cards loose communication with the network and causes older versions of PCAnywhere loose control of its IP based service. Many other similar anomalies have since been reported on the Nmap list (4). All of these faults required a reset of the host service. This was something our department network and server administrators were not very

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

happy about, especially since we were scanning seven days a week, round the clock. This brought about the next link in the chain, a database of addresses to be excluded from scanning. This does have some risk, because if a system is compromised in that group of exclusions we may not see the change in services. This risk was deemed to be less than the pain our administrators were experiencing from our scans breaking things. Keep in mind that any of the machines that were open to the outside Internet were subject to being scanned by others with similar consequence. Supplemental scans of the restricted addresses were scheduled with each administrator. The structure for the addresses to be excluded is illustrated in Figure 3. including start, stop, and update fields so we could add and remove systems for limited times based upon system administrator requests.

Figure 3.

Restricted_addresses_db	
IP	Originally we thought we could selectively scan
Port	some ports and not others for each IP. This has
Start date	proven to be a problem to pass the list to Nmap
Last update	for each address. Now we just drop the entire IP
End date	from the list. (Evolved into table 5 Exclusion list)

We then continued scanning, skipping the addresses in the restricted list, and producing reports of the types of services found within our network. Several of the most telling reports were listings of:

Listings of windows machines vulnerable to NetBIOS compromise (port 139 open) (7)
Number of HTTP servers (port 80),
Number FTP servers (port 21),
Anonymous FTP servers
Servers with more than 20/50/100 ports open,
Machines infected with known trojans etc.
And others

These reports were very helpful in our overall understanding of the vulnerability of our network (9) because many of the recipients of these reports were unaware of the services running in their area. This was compounded by many servers having default services running. A good example of this is HP Jet Direct print servers that have HTTP, FTP, SNMP open on them by default.

As we began working down through the lists of machines with problems, we quickly addressed issues on our primary servers. In addition we found that for many of the departmental servers there was no good way to get this reporting to the administrator of the server that was vulnerable. Worse yet, there was no good way to isolate the owner of a particular IP address, or where it was located other than knowing the location of the up-line hub or switch it was connected to. In our case the network infrastructure was well documented, and we had good location information for all of these up line devices. Our location problems were compounded because some of the machines exhibiting the highest number of open ports were not servers at all but individual user

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

workstations. DHCP complicates this even more because DHCP addressing pulls addresses from an open pool of available addresses so that the link of class C network number to the geographical department originally planned in our environment was lost.

Evolution of the tools

What we needed was a better tool; one that could scan the network like Nmap but that could also identify users and details about the physical location of the machine being scanned. Figure 4. illustrates the mix of tools we chose to help collect a better set of vulnerability data about each host.

Scans can be scheduled or can be triggered manually as needed. For manual scans the command variables are chosen and stored these include IP range to scan, ranges or individual IP's to exclude, and which scanning tools to run. For scheduled scans databases containing the range of addresses to be scanned, and the addresses to be exempt from scanning are used. Databases are also available containing addresses and ports attributed to known vulnerabilities, trojan and viri for use in reporting. Lists of the database tables are available at the end of this document.

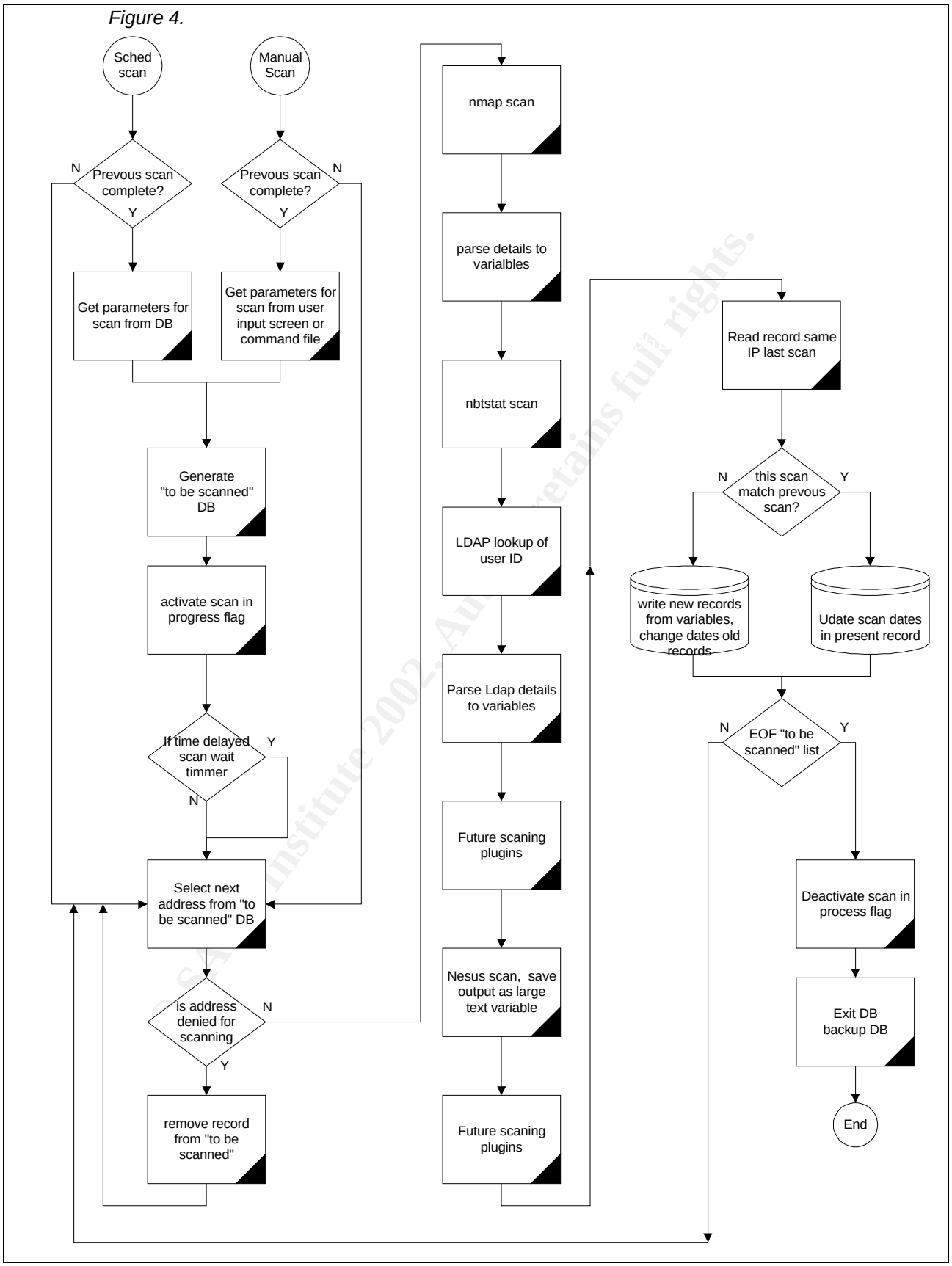
Once the controlling criteria are set, a list of IP addresses to be scanned is generated and saved to a "to be scanned database" and a flag is set to allow restart in case of program failure. This has become a necessity because several environmental issues can cause Nmap, Nessus or other scanners to lockup and error in ways that cannot be recovered from programmatically. On Linux derivatives of our system a process monitoring script can automatically increment the address to be scanned database and run another copy of the scanning software upon failure.

Once the initial criteria for the scan are setup the program checks the address about to be scanned against a list of "denied" addresses. Our environment contains medical devices that can connect via IP, like heart monitors, drug dose dispensing, and lab equipment. These devices are segregated into physically or logically separate networks. Some of our sample scanning points were selected to test the ability to reach these devices. These devices are scanned for vulnerability under test conditions rather than interfere with them in any way during their clinical usage. These ranges are denied from normal scanning and outside connection.

An Nmap scan is run and the elements in Figure 0. are collected to variables. There is a one to many relationship of the IP address master record to the Port records. Much like the text output of Nmap a Port record is created for each detected port and the status of open or closed is also recorded. These IP master and Port detail records are marked with activity dates do changes of the open ports can be reported on over time. Also reports of the increase or reduction of open services can be created for quarterly or yearly risk analysis. If no change is detected when evaluated against previous scans, only the checked dates are updated in the database.

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

Figure 4.



© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

The next scan to take place is actually issuing the MS DOS / Samba command NBTstat which collects NetBIOS and Microsoft domain details about the IP the command is issued against. (8) This populates the MAC address, MS_domain and MS_shares rows of the IP detail database for the selected IP address. The MS_user row of the User detail database is also populated from the output of this command. A sample NBTstat output is displayed in figure 6.

Figure 6. NetBIOS Remote Machine Name Table

```
  Name                  Type        Status
-----------------------------------------------------
MAC Address = 00-60-08-A4-DF-4B

MY-MACHINE-NT1   <00>   UNIQUE
MYDOMAIN         <00>   GROUP
MY-MACHINE-NT1   <20>   UNIQUE
MYDOMAIN         <1E>   GROUP
MYDOMAIN         <1D>   UNIQUE
HARRISMC         <03>   UNIQUE
..__MSBROWSE__.  <01>   GROUP
```

An explanation of the coding used for the output of this command is available at http://support.microsoft.com /support/kb/articles/Q163/4/09.asp and a description of enumeration based upon command line usage of NBTstat is available in Hacking Exposed 2nd edition page 77 and 78 (8). Keep in mind that only Windows and Samba clients (other NetBIOS aware systems) will respond to this command. We have had about an 80% positive hit rate within our network.

The next step is only available to Exchange users who populate their LDAP service with the Microsoft domain login names. This is, I believe, a default setting and easily publishable within Exchange, but many organizations that publicly publish their LDAP services do not want to include the logon ID for obvious security reasons. We take the MS logon name found from the NBTstat and look it up in our exchange LDAP service this gives us the data required to populate the matching rows in the User detail table as described in figure 7.

Figure 7.
User Detail (linked to IP master via IP)

IP	
MS_user	This is a duplication of data for what is redialy available
User_Name	from LDAP. We keep previous entries based upon scan
User_department	dates because no historical data is kept by LDAP and
User_address	department changes as users move from one department
User_phone	were lost. Many vulnerabilities and bad behaviors follow
Start_date	users as they move. This has evolved into table 9 as listed
Checked_date	at the end of this document.
End_date	

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

With this process complete we have a database source of the active IP addresses in our network, the services running on those machines and the operators of those machines with physical department locations and phone numbers if needed to contact them. The remaining 20% of the devices with IP addresses are network gear (switches, hubs, routers etc.) supported by an institution wide network support group or dedicated lab equipment. Nmap is able to detect many of both of these types of devices through its OS detection feature leaving only a small number of addresses to be manually investigated.

The next stage of scanning is more a matter of vulnerability assessment than machine or user identification. A Nessus scan is run against the target IP address and the output is placed as a single large text entry in the IP detail database in the Vulnerabilities_checked row. Each text block is parsed for the text strings in the Danger table to determine if any of the restricted or dangerous items are found. These items are specifically called out in policy and procedure and are reported upon on a quarterly basis. When found certain suggested and mandatory actions must be taken. These individually reported vulnerabilities are: Password less MS shares, anonymous FTP, unencrypted Telnet, mail forwarding, and SNMP public strings. A database of dangerous ports has also been created to check when reporting. Structure of that database is listed in table 4 at the end of the document.

Once the Nessus scan is complete data stored in variables throughout the scanning process is compared with previous detail records. If new details were found a new record is created and the begin and end dates are set. If no changes are found the existing records are retained and the Checked date for each is updated. Figure 7 is a complete list of the table structure for the application.

Other scans for your network can be added within this structure. One of the scans we have added due to the Code Red Worm outbreaks is daily targeted scans of smaller ranges. This is mostly because this type of scanning is slow, it takes several days to scan a complete a scan of a class B network. This limited our response time for new vulnerabilities as they were found. We have added the ability to prefeed lists of addresses to the “to be scanned” list based upon what services were found active in previous scans. As an example specific scans of only machines that have port 80 (HTTP) open are scanned for CGI and web server vulnerabilities.

Items we have slated for future development

Next we will be developing a dedicated web based front end for initiating the scanning process and a web based report selection tool. At present both of these processes are done manually by our security staff. One future developments planned is a link to our intrusion detection databases. Snort creates a very good database of the traffic target to or from a specific IP address. Having a link to this data when reporting vulnerability would be of great value during assessment, to be able to make a determination that a vulnerability was already being exploited is our goal.

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

Unresolved Problems

One of the problems we have not resolved is multiple reportings of the same vulnerability on multi-homed machines. (1) We are working to eliminate these duplicates by using a batch cleanup process that searches for duplicate MAC addresses.

Another problem is that of non-Microsoft NetBIOS aware devices. There are many devices that do not respond to NetBIOS calls. Much of the network gear and alternate operating systems do not respond to NBTstat therefore we do not receive user data or location details about them. We are currently adding location details manually to the database about non-NetBIOS responsive devices.

Summary

With the number of intrusive or malicious events on our networks growing at a frightening rate and external attack from the Internet on the rise, we have to begin thinking like the intruders that are looking toward our systems to exploit their resources. We must be equally diligent about modem pools and any side channels that can be used to infiltrate our networks. This drives the need for good information about the systems that we have within our networks and the vulnerabilities exist within them. Simple inventories on paper or in databases are not enough. The users and devices on your network are dynamic and constantly changing. With the constant threat there must be a constant vigilance on our part to keep our networks secure.

© SANS Institute 2002, Author retains full rights.

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

References

Special thanks to James Cutts III, Project Manager, ITS RES, University of Missouri Health Center for his assistance in database planning and authoring of the Perl scripts that is the glue of this project.

1) "Zone of risk" concept originally from *Thinking About Firewalls* Marcus J. Ranum originally presented at SANSII in Washington, DC, 1993 http://www.linuxsecurity.org/resource_files/firewalls/ThinkingFirewalls/ThinkingFirewalls.html

2) For the quote and graphic on the starting page, http://www.mailsbroadcast.com/the.artofwar.3.htm Copyright May 2001 MailsBroadcast dotcom - All rights reserved. Reprinted with permission from http://www.mailsbroadcast.com

3) Nmap http://www.insecure.org/nmap/ My greatest appreciation for Fyodor and the team developing Nmap without Nmap this project would not have been possible. Initial product reference. Man pages referenced.

4) *Nmap* Moderated list for announcements, patches, and light discussion regarding the Nmap scanner http://lists.insecure.org

5) Nlog by H D Moore http://www.digitaloffense.net/nlog/nlog-1.6.0/BUGTRAQ referenced as the original idea and basis for this project.

6) Special thanks to the folks at eEye for recompiling Nmap for Win32, allowing our detectors to be both Windows and Linux based http://www.eeye.com/html/Research/Tools/nmapNT.html

7) *Hacking exposed 2nd ed*.Scambray McClure & Kurtz Osborrne, Mcgraw Hill 2001 http://www.hackingexposed.com , page 50 & 51 re: understanding risk from Nmap scanning

8) *Hacking exposed 2nd ed*.Scambray McClure & Kurtz Osborrne, Mcgraw Hill 2001 http://www.hackingexposed.com , Page 77 &78 re: NetBios enumeration.

9) *Hacking exposed 2nd ed*.Scambray McClure & Kurtz Osborrne, Mcgraw Hill 2001 http://www.hackingexposed.com , Page 44 re: Port Scanning to determine OS and applications.

10) *Nessus* remote security scanner http://www.nessus.org/intro.html

11) *Sun Tzu: The Art of Warfare*, translated by Roger Ames, Ballantine Books, New York, 1993 http://vikingphoenix.com/public/SunTzu/suntzu.htm

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

Data Base Tables

DB table 1 Command
Range_Start
Range_end
Start_date
Start_time
Stop_date
Stop_time
Completed
Restart_halted_scan
OM *logfile*name
Port_list
Port_type
Source_IP_address
Host_timeout
Ldap_lookup
Nmap_string_used
Nessus
Nessus_string_used

DB table 2 IP to Scan
IP
Status

DB table 3 Ports to Scan
Ports_to_scan
Port_type

DB table 4 Danger
IP
Port
Port_desc
Status
Status_desc
Sugested_action
Manditory_action

DB table 5 Exclusion List
IP
Start_date
End_date
Reason_excluded
Ports_allowed
Ports_excluded
Auth_by
Notes

© SANS Institute 2002, Author retains full rights.

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

Nmap Nbtstat Nessus Owner Owner_group

DB table 6 IP Master IP IPa IPb IPc IPd Port_data Start_date Checked_date End_date Owner Owner_group

DB table 7 IP Detail IP IPa IPb IPc IPd OS Domain_name MS_domain MAC Vulnerabilities_detected MS_shares MS_passless_shares FTP Telnet Smtp_fwd Snmp_pub Ports_open_count Owner Owner_group

DB table 8 Port Detail IP Port Port_type Port_stat Port_desc

© SANS Institute 2002, Author retains full rights.

Danger Start_date Checked_date End_date Owner Owner_group

DB table 9 User Detail IP MS_user User_name User_department User_Address User_phone Start_date End_date Owner Owner_group

DB table 10 Owners Owner Owner_group Start_date End_date

Questions

1.
2.
3.
4.
5.
True / False
6.
7.
8.
9.
10

© SANS Institute 2002, Author retains full rights.

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

Last Updated: September 1st, 2015

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Crystal City 2015	Crystal City, VAUS	Sep 08, 2015 - Sep 13, 2015	Live Event
SANS Network Security 2015	Las Vegas, NVUS	Sep 12, 2015 - Sep 21, 2015	Live Event
SANS Seoul 2015	Seoul, KR	Sep 14, 2015 - Sep 19, 2015	Live Event
Data Breach Investigation Summit & Training	Dallas, TXUS	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS Baltimore 2015	Baltimore, MDUS	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS Perth 2015	Perth, AU	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS Tallinn 2015	Tallinn, EE	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS ICS Amsterdam 2015	Amsterdam, NL	Sep 22, 2015 - Sep 28, 2015	Live Event
SANS Bangalore 2015	Bangalore, IN	Sep 28, 2015 - Oct 17, 2015	Live Event
SANS DFIR Prague 2015	Prague, CZ	Oct 05, 2015 - Oct 17, 2015	Live Event
SANS Seattle 2015	Seattle, WAUS	Oct 05, 2015 - Oct 10, 2015	Live Event
SANS Tysons Corner 2015	Tysons Corner, VAUS	Oct 12, 2015 - Oct 17, 2015	Live Event
SOS: SANS October Singapore 2015	Singapore, SG	Oct 12, 2015 - Oct 24, 2015	Live Event
SANS Gulf Region 2015	Dubai, AE	Oct 17, 2015 - Oct 29, 2015	Live Event
SANS Cyber Defense San Diego 2015	San Diego, CAUS	Oct 19, 2015 - Oct 24, 2015	Live Event
SANS Tokyo Autumn 2015	Tokyo, JP	Oct 19, 2015 - Oct 31, 2015	Live Event
SANS South Florida 2015	Fort Lauderdale, FLUS	Nov 09, 2015 - Nov 14, 2015	Live Event
SANS Sydney 2015	Sydney, AU	Nov 09, 2015 - Nov 21, 2015	Live Event
SANS London 2015	London, GB	Nov 14, 2015 - Nov 23, 2015	Live Event
Pen Test Hackfest Summit & Training	Alexandria, VAUS	Nov 16, 2015 - Nov 23, 2015	Live Event
SANS Hyderabad 2015	Hyderabad, IN	Nov 24, 2015 - Dec 04, 2015	Live Event
SANS Cape Town 2015	Cape Town, ZA	Nov 30, 2015 - Dec 05, 2015	Live Event
SANS San Francisco 2015	San Francisco, CAUS	Nov 30, 2015 - Dec 05, 2015	Live Event
SANS Milan 2015	OnlineIT	Sep 07, 2015 - Sep 12, 2015	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced