|  |  |  |  |
| --- | --- | --- | --- |
| Not every company or state agency can afford the Solarwinds Toolset for SNMP scanning. Solarwinds does make a nice set of tools for scanning the network for hosts running SNMP, conducting dictionary attacks, and downloading configuration files from routers and switches. It even decrypts the ridiculously weak Cisco Type 7 encryption. This tutorial will show you how to audit a network for SNMP enumeration and vulnerabilities without the use of an expensive application. | | | |
| Command | Description | Link | Appendix |
| Nmap | Nmap ("Network Mapper") is a free and open source utility for network exploration or security auditing. | <http://nmap.org> | Linux [Install](#installnmap) |
| onesixtyone | onesixtyone takes advantage of the fact that SNMP is a connectionless protocol and sends all SNMP requests as fast as it can. Then the scanner waits for responses to come back and logs them, in a fashion similar to Nmap ping sweeps. | <http://www.phreedom.org/solar/onesixtyone/> | Linux [Install](#install161) |
| snmpenum.pl | Simple Perl script to enumerate devices and servers and grab information using common Management information bases (MIBs). |  | Linux [Install](#installsnmpenum) |
| snmpwalk | **snmpwalk** is an SNMP application that uses SNMP GETNEXT requests to query a network entity for a tree of information. | <http://www.net-snmp.org/docs/man/snmpwalk.html> |  |
|  |  |  |  |
|  |  |  |  |

| **Task** | **Steps and Description** | **Initials** | **Date** | **Linked Results** |
| --- | --- | --- | --- | --- |
| 1 | Scan for hosts running SNMP default UDP port 161. Parse the scan results and create a hosts file (host.txt) for use in later tasks.  #nmap -sU -p161 -oG snmp.log <ip\_address\_range> |  |  | [EV1](#EV1) |
| 2 | Create a file of hosts found with SNMP port open.  #grep open snmp.log | awk ‘{print $2}’ > hosts.txt |  |  | [EV2](#EV2) |
| 3 | Run an SNMP scan against all hosts identified using a small dictionary file of common SNMP community names. The tool onesixtyone is a fast SNMP scanner that will be used for this step. Use the default dict.txt file that comes with onesixtyone.  # onesixtyone -i hosts.txt -c dict.txt -d -o 161.log |  |  | [EV3](#EV3) |
| 4 | Parse onesixtyone results to produce a file of all hosts with the name of the community string identified and the operating system identified. If multiple community strings worked there will be multiple files.  Create a file with the following words can call it os.txt:  cisco  linux  windows  Create a file of community names found per Operating system (Windows, Linux/Unix, Cisco)  # for OS in $(cat os.txt);do for COMMUNITY in $(grep -i $OS 161.log | awk '{print $2}' |sed 's/.\(.\*\)./\1/'| sort -u);do grep $COMMUNITY 161.log |grep -i $OS | awk '{print $1}' |sort -u >> $OS.$COMMUNITY.txt; done; done  See [Appendix B](#task4loop) for details on what this long command does. |  |  | [EV4](#EV4) |
| 5 | Take the files from step 4 and use the perl script snmpenum.pl to obtain detailed information from each host. The script comes with three configuration files for Cisco, Windows, & Linux that contain common SNMP MIB strings to pull detailed information regarding the host.  Usage: perl snmpenum.pl <IP-address> <community> <configfile>  The files created from Task 4 contain all the information you would provide to the snmpenum.pl script. The file name is the format OS.community.txt and in the file are the actual ip addresses to scan. Got another for loop sandwich you can run to get the information you need.  #for OS in $(cat os.txt);do for COMMUNITY in $(ls $OS.\* |awk 'BEGIN { FS = "." } ; { print $2 }');do for IP in $(cat $OS.$COMMUNITY.txt);do perl snmpenum/snmpenum.pl $IP $COMMUNITY snmpenum/$OS.txt >> $OS.$COMMUNITY.$IP.txt; done; done; done  NOTE: you will probably have to edit the windows.txt file that comes with snmpenum.pl. The DOMAIN MIB will hang forever. It is best to remove that entry.  See [Appendix B](#task5loop) for details on what this long command does. |  |  | [EV5](#EV5) |
| 6 | Advanced  Walk the SNMP tree and obtain all information using SNMPWalk  Using SNMPWalk to obtain all SNMP information from a host. The snmpenum.pl Perl script does a great job of obtaining the most useful information from a Windows, Linux server, or Cisco device. However, you may want to view all the information that is available or you are dealing with a device that is not Windows, Linux, or Cisco. For example, you may want to obtain useful information from a print server or Nortel device. |  |  | [EV6](#EV6) |

EV1 – Nmap scan results (snmp.log) (example) ([Task 1](#task1))

# Nmap 5.50 scan initiated Tue Mar 22 16:24:14 2011 as: nmap -sU -vv -p161 -oG snmp.log 192.168.0.0/24

# Ports scanned: TCP(0;) UDP(1;161) SCTP(0;) PROTOCOLS(0;)

Host: 192.168.0.1 () Status: Up

Host: 192.168.0.1 () Ports: 161/closed/udp//snmp///

Host: 192.168.0.16 (j-laptop.lan) Status: Up

Host: 192.168.0.16 (j-laptop.lan) Ports: 161/closed/udp//snmp///

Host: 192.168.0.156 (android\_7792117e15d40a5f.lan) Status: Up

Host: 192.168.0.156 (android\_7792117e15d40a5f.lan) Ports: 161/closed/udp//snmp///

Host: 192.168.0.172 () Status: Up

Host: 192.168.0.172 () Ports: 161/open|filtered/udp//snmp///

Host: 192.168.0.214 (win2k3.lan) Status: Up

Host: 192.168.0.214 (win2k3.lan) Ports: 161/open|filtered/udp//snmp///

Host: 192.168.0.225 (e-ubuntu.lan) Status: Up

Host: 192.168.0.225 (e-ubuntu.lan) Ports: 161/open/udp//snmp///

Host: 192.168.0.242 (w2kserver.lan) Status: Up

Host: 192.168.0.242 (w2kserver.lan) Ports: 161/open|filtered/udp//snmp///

Host: 192.168.0.254 () Status: Up

Host: 192.168.0.254 () Ports: 161/open/udp//snmp///

# Nmap done at Tue Mar 22 16:24:23 2011 -- 256 IP addresses (8 hosts up) scanned in 9.37 secondsEV2 – Hosts file create - ([Task 2](#task2))

192.168.0.172

192.168.0.214

192.168.0.225

192.168.0.242

192.168.0.254 EV3 – ([Task 3](#task3))

|  |  |
| --- | --- |
| Sample Screen Output. NOTE: This is with debug output. You can even be more verbose with –dd. | Sample Log file Output. |
| root@e-ubuntu:~# onesixtyone -i hosts.txt -c dict.txt -d -o 161.log  Debug level 1  Reading hosts from input file hosts.txt  5 hosts read from file  Using community file dict.txt  Logging to file 161.log  49 communities: 1234 2read 4changes CISCO IBM OrigEquipMfr SNMP SUN access admin agent all cisco community default enable field guest hello ibm manager mngt monitor netman network none openview pass password private proxy public read read-only read-write root router secret security snmp snmpd solaris sun switch system tech test world write  Waiting for 10 milliseconds between packets  Scanning 5 hosts, 49 communities  Trying community 1234  Trying community 2read  Trying community 4changes  Trying community CISCO  Trying community IBM  Trying community OrigEquipMfr  Trying community SNMP  Trying community SUN  Trying community access  Trying community admin  Trying community agent  Trying community all  Trying community cisco  192.168.0.254 [cisco] Cisco Internetwork Operating System Software IOS (tm) C1700 Software (C1700-Y-M), Version 12.1(3), RELEASE SOFTWARE (fc1) Copyright (c) 1986-2000 by cisco Systems, Inc. Compiled Wed 05-Jul-00 17:07 by cmong  Trying community community  Trying community default  Trying community enable  Trying community field  Trying community guest  Trying community hello  Trying community ibm  Trying community manager  Trying community mngt  Trying community monitor  Trying community netman  Trying community network  Trying community none  Trying community openview  Trying community pass  Trying community password  Trying community private  192.168.0.214 [private] Hardware: x86 Family 6 Model 37 Stepping 5 AT/AT COMPATIBLE - Software: Windows Version 5.2 (Build 3790 Uniprocessor Free)  192.168.0.242 [private] Hardware: x86 Family 6 Model 5 Stepping 5 AT/AT COMPATIBLE - Software: Windows 2000 Version 5.0 (Build 2195 Uniprocessor Free)  Trying community proxy  Trying community public  192.168.0.214 [public] Hardware: x86 Family 6 Model 37 Stepping 5 AT/AT COMPATIBLE - Software: Windows Version 5.2 (Build 3790 Uniprocessor Free)  192.168.0.225 [public] Linux e-ubuntu 2.6.32-24-generic #39-Ubuntu SMP Wed Jul 28 06:07:29 UTC 2010 i686  192.168.0.242 [public] Hardware: x86 Family 6 Model 5 Stepping 5 AT/AT COMPATIBLE - Software: Windows 2000 Version 5.0 (Build 2195 Uniprocessor Free)  192.168.0.254 [public] Cisco Internetwork Operating System Software IOS (tm) C1700 Software (C1700-Y-M), Version 12.1(3), RELEASE SOFTWARE (fc1) Copyright (c) 1986-2000 by cisco Systems, Inc. Compiled Wed 05-Jul-00 17:07 by cmong  Trying community read  Trying community read-only  Trying community read-write  Trying community root  Trying community router  Trying community secret  Trying community security  Trying community snmp  Trying community snmpd  Trying community solaris  Trying community sun  Trying community switch  Trying community system  Trying community tech  Trying community test  Trying community world  Trying community write  All packets sent, waiting for responses.  done. | 192.168.0.254 [cisco] Cisco Internetwork Operating System Software IOS (tm) C1700 Software (C1700-Y-M), Version 12.1(3), RELEASE SOFTWARE (fc1) Copyright (c) 1986-2000 by cisco Systems, Inc. Compiled Wed 05-Jul-00 17:07 by cmong  192.168.0.214 [private] Hardware: x86 Family 6 Model 37 Stepping 5 AT/AT COMPATIBLE - Software: Windows Version 5.2 (Build 3790 Uniprocessor Free)  192.168.0.242 [private] Hardware: x86 Family 6 Model 5 Stepping 5 AT/AT COMPATIBLE - Software: Windows 2000 Version 5.0 (Build 2195 Uniprocessor Free)  192.168.0.214 [public] Hardware: x86 Family 6 Model 37 Stepping 5 AT/AT COMPATIBLE - Software: Windows Version 5.2 (Build 3790 Uniprocessor Free)  192.168.0.225 [public] Linux e-ubuntu 2.6.32-24-generic #39-Ubuntu SMP Wed Jul 28 06:07:29 UTC 2010 i686  192.168.0.242 [public] Hardware: x86 Family 6 Model 5 Stepping 5 AT/AT COMPATIBLE - Software: Windows 2000 Version 5.0 (Build 2195 Uniprocessor Free)  192.168.0.254 [public] Cisco Internetwork Operating System Software IOS (tm) C1700 Software (C1700-Y-M), Version 12.1(3), RELEASE SOFTWARE (fc1) Copyright (c) 1986-2000 by cisco Systems, Inc. Compiled Wed 05-Jul-00 17:07 by cmong |

EV4 - ([Task 4](#task4)).

root@e-ubuntu:~# for OS in $(cat os.txt);do for COMMUNITY in $(grep -i $OS 161.log | awk '{print $2}' |sed 's/.\(.\*\)./\1/'| sort -u);do grep $COMMUNITY 161.log |grep -i $OS | awk '{print $1}' |sort -u >> $OS.$COMMUNITY.txt; done; done

root@e-ubuntu:~# cat cisco.cisco.txt

192.168.0.254

root@e-ubuntu:~# cat cisco.public.txt

192.168.0.254

root@e-ubuntu:~# cat windows.public.txt

192.168.0.214

192.168.0.242

root@e-ubuntu:~# cat linux.public.txt

192.168.0.225

root@e-ubuntu:~# cat windows.private.txt

192.168.0.214

192.168.0.242EV5 – ([Task 5](#task5)).

Example Results

<http://www.jedge.com/docs/windows.public.192.168.0.214.txt>

<http://www.jedge.com/docs/cisco.public.192.168.0.254.txt>

<http://www.jedge.com/docs/linux.public.192.168.0.226.txt>

EV6 – ([Task 6](#task6)).

**Appendix A: Command Installation Help**

**Installing NMAP**

$mkdir ~/source

$cd ~/source

$wget http://nmap.org/dist/nmap-5.51.tar.bz2

$tar jxvf nmap-5.51.tar.bz2

$cd nmap-5.51

$./configure

$make

$sudo make install

**Installing onesixtyone**

#wget http://www.phreedom.org/solar/onesixtyone/onesixtyone-0.3.2.tar.gz

#tar zxvf onesixtyone-0.3.2.tar.gz

#cd onesixtyone-0.3.2/

#make

#cp onesixtyone /usr/local/bin

**Installing snmpenum.pl**

#apt-get install libnet-snmp-perl

#mkdir ~/tools

#cd ~/tools

#wget http://www.jedge.com/utilities/snmpenum.tar.gz

#tar zxvf snmpenum.tar.gz

**Installing copy-router-config.pl**

# perl -MCPAN -e 'install Cisco::CopyConfig'

# wget http://www.jedge.com/utilities/copy-router-config.tar.gz

# tar zxvf copy-router-config.tar.gz

# chmod 777 copy-router-config.pl

Appendix B – Command Help

**Task 4 Nested For Loop [BASH]**

This section will break out the nested for loop found in [task 4](#task4).

1 for OS in $(cat os.txt)

2 do

3 for COMMUNITY in $(grep -i $OS 161.log | awk '{print $2}' |sed 's/.\(.\*\)./\1/'| sort -u)

4 do

5 grep $COMMUNITY 161.log | grep -i $OS | awk '{print $1}' |sort -u >> $OS.$COMMUNITY.txt

6 done

7 done

Cycle through the os.txt file and for each word (device) we will search 161.log for devices found and get a list of community names that work for those devices. Then a file is created of all ip addresses that match that device and community name.

**Task 5 Nested For Loop [BASH]**

This section will break out the nested for loop found in [task 5](#task5).

1 for OS in $(cat os.txt);

2 do

3 for COMMUNITY in $(ls $OS.\* |awk 'BEGIN { FS = "." } ; { print $2 }');

4 do

5 for IP in $(cat $OS.$COMMUNITY.txt);

6 do

7 perl snmpenum/snmpenum.pl $IP $COMMUNITY snmpenum/$OS.txt >> $OS.$COMMUNITY.$IP.txt

8 done

9 done

10 done

Cycle through the os.txt file and for each word (device) we will list (ls) the directory looking for the files created from the previous task with the first word of the file being the device name we pulled from os.txt. We use awk to grab the second word from the file. This is the community name. Lastly we use this information to list the contents (IP address) of the file (os.community.txt). We take all this information and feed it to the perl script snmpenum.pl to grab the information from all hosts enumerated.