



Security Tool Review and Howto:
scanssh

by Christian Houle CISSP, TICSA
christian@localareasecurity.com

Introduction

As security conscious administrators or information security professionals we must remain up to date in the world of vulnerabilities. Knowing where these vulnerabilities lie in your I.T. infrastructure is an essential step to successful risk mitigation. This section will be dedicated to reviewing tools that will help us in identifying and locating various security vulnerabilities.

Overview

<i>scanssh</i> v.1.6b	http://www.monkey.org/~provos/scanssh/
<i>scanssh</i> is a multithreaded protocol scanner that can probe a list of addresses and networks for running SSH & HTTP servers and displays their version numbers.	
<i>scanssh</i> supports random selection of IP addresses from large network ranges and is useful for gathering statistics on the deployment of SSH protocol servers in a company or the Internet as whole.	
<i>scanssh</i> is distributed under a BSD-license and completely free for any use.	
<i>scanssh</i> [-VIERh] [-n port] [-e excludefile] [-b hostalias] [-p ifaddr] <IP address/network>...	
-V	Prints the version number.
-I	Does not send an identification string. This causes the IP number of the scanning host to be logged.
-E	Exit the program, if the file containing the addresses for exclusion can not be found.
-R	If addresses are generated at random, this flag causes the program to ignore excluded addresses from the exclude file. The default behavior is to always exclude addresses.
-h	Displays the usage of the program.
-n port	Specifies the port number to scan. Only port 22/ssh or port 80/http are supported. Port number 22 scans for SSH server versions and port number 80 scans for HTTP server versions. The default is 22.
-e excludefile	Specifies the file that contains the addresses to be excluded from the scan. The syntax is the same as for the addresses on the command line.
-b hostalias	Specifies the ip address of an interface alias from which connections should be attempted. If -p is being used, both ip addresses should be the same.
-p ifaddr	Specifies the address of the local interface. This is used to speed up the scanning by pre-probing the addresses with TCP-SYN packets.
	Please note that you will have to be root (under Linux) to use this option.
Advanced	Please consult the man pages for more details on the advance options.

The following command scans the class C network 192.168.1.0 - 192.168.1.255:

```
root@1[root]# scanssh 192.168.1.0/24
```

The next command uses SYN probes to speed up the scan

```
root@1[root]# scanssh -p 10.0.0.1 192.168.0.0/16
```

The next command does not send the id string & excludes the scanning of the ip's in file exip.txt:

```
root@1[root]# scanssh -I -e exip.txt 192.168.1.0/24
```

Sample Output:

```
192.168.1.25 <refused>  
192.168.1.193 <refused>  
192.168.1.248 SSH-1.99-OpenSSH_3.5p1  
192.168.1.177 SSH-2.0-OpenSSH_3.1p1  
192.168.1.254 <timeout>  
192.168.1.235 <timeout>
```

More Information on scanssh:

<http://www.citi.umich.edu/techreports/reports/citi-tr-01-13.pdf>

Review

One of the services that I commonly use for remote system administration is secure shell a.k.a. SSH. Its encryption capabilities, tunneling features and support for various methods of advance authentication make it one of the best remote administration tools available. Various commercial and open source implementations of SSH have had their share of vulnerabilities. In September 2000, *scanssh* was released in order to identify vulnerable SSH servers thru version identification.

scanssh is a simple tool that probes a single or multiple systems for the SSH or HTTP service. While scanning, it accurately displays the version of each service. This permits us to identify and eventually update the outdated SSH or HTTP services. *scanssh* can also perform multiple probes quickly due to its capacity to use multiple threads and pre-probe the addresses with tcp-syn packets.

Unfortunately, it's limited to reviewing only port tcp/22 and port tcp/80 for the SSH and HTTP service respectively. A non-standard configuration and implementation of these services can drastically limit the tools overall use in an organization.

A number of tools can perform the same basic functions. We tested and discovered that nmap 3.48 was able to perform the review of a class C subnet faster & at the same level of accuracy using the new -sV switch. Both of the tools are available on the new LAS bootable distribution.

scanssh v.1.6b (<http://www.monkey.org/~provos/scanssh/>)
nmap version 3.48 (<http://www.insecure.org/nmap/>)

I believe *scanssh* is a good tool but the constant additions and improvements to nmap's version detection feature are slowly making this non-maintained tool (*scanssh*) obsolete.

Test Results:

A few problems came up when attempting the use scanssh with the `-p` option & specifying a class C as target. This was attempted on 2 different systems & networks.

The 3 tools all captured the same & correct versions of the running SSH servers in a specific class C range. An interesting difference was the output which I found the simplest to read & search thru in scanssh.

Usage:

Command	Time to Completion
scanssh 192.168.1.0/24	2 mins
Sample Output: 192.168.1.222 <refused> 192.168.1.193 <refused> 192.168.1.248 SSH-1.99-OpenSSH_3.5p1 192.168.1.199 <refused> 192.168.1.253 <timeout> 192.168.1.255 <timeout> 192.168.1.177 SSH-2.0-OpenSSH_3.1p1 192.168.1.254 <timeout> 192.168.1.241 <timeout> 192.168.1.235 <timeout> 192.168.1.244 <timeout> 192.168.1.234 <timeout> 192.168.1.243 <timeout>	
scanssh -p 192.168.1.102 192.168.1.0/24	N/A
No Output/Error Encountered	
nmap -sS -sV -p 22 192.168.1.0/24	35 secs

Sample Output:

Interesting ports on (192.168.1.176):
PORT STATE SERVICE VERSION
22/tcp closed ssh

Interesting ports on (192.168.1.177):
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 3.1p1 (protocol 2.0)

Interesting ports on (192.168.1.183):
PORT STATE SERVICE VERSION
22/tcp closed ssh

hackbot.pl -s 192.168.1.0/24

12mins. 15secs

Sample Output:

Checking 192.168.1.177 ...

Checking for SSH

SSH-2.0-OpenSSH_3.1p1

Checking 192.168.1.178 ...

Checking for SSH

Checking 192.168.1.179 ...

Checking for SSH

Checking 192.168.1.180 ...

Checking for SSH

Manpage

scanssh(1) BSD General Commands Manual scanssh(1)

NAME

scanssh - scans the Internet for SSH server versions

SYNOPSIS

scanssh [-VIERh] [-n port] [-e excludefile] [-b hostalias] [-p ifaddr]
addresses...

DESCRIPTION

scanssh scans the given addresses and networks for running SSH servers. It will query their version number and displays the results in a list.

The addresses can be either specified as an IPv4 address or an CIDR like IP prefix, ipaddress/masklength.

Additionally, the following two commands can be prefixed to the address:

random(n[,seed])/ The random command selects random address from the address range specified. The arguments are as follows: n is the number of address to randomly create in the given network and seed is a seed for the pseudo random number generator.

split(s,e)/ The split command is used to split the address range in several unique components. This can be use to scan from serveral hosts in parallel. The arguments are as follows: e specifies the number of hosts scanning in parallel and s is the number of the host this particular scan runs on.

The options are as follows:

- V Causes scanssh to print its version number.
- I Does not send an identification string. This causes the IP number of the scanning host to be logged.
- E Exit the program, if the file containing the addresses for exclusion can not be found.
- R If addresses are generated at random, this flag causes the program to ignore excluded addresses from the exclude file. The default behaviour is to always exclude addresses.
- h Displays the usage of the program.
- n port Specifies the port number to scan. Currently, only 22 or 80 are supported. Port number 22 scans for SSH server versions and port number 80 scans for HTTP server versions. The default is 22.
- e excludefile Specifies the file that contains the addresses to be

excluded from the scan. The syntax is the same as for the addresses on the command line.

-b hostalias Specifies the ip address of an interface alias from which connections should be attempted. If **-p** is being used, both ip addresses should be the same.

-p ifaddr Specifies the address of the local interface. This is used to speed up the scanning by pre-probing the addresses with TCP-SYN packets.

Please note that you will have to be root (under Linux at any rate) to use this option.

The output from `scanssh` contains only IP addresses. However, the IP addresses can be converted to names with the `logresolve(8)` tool included in the Apache webserver.

EXAMPLES

The following command scans the class C network 10.0.0.0 - 10.0.0.255:

```
scanssh 10.0.0.0/24
```

The next command uses SYN probes to speed up the scan

```
scanssh -p 10.0.0.1 192.168.0.0/16
```

The following command can be used in a parallel scan. Two hosts scan the specified networks randomly, where this is the first host:

```
scanssh -p 10.0.0.1 'random(0,rsd)/split(1,2)/(192.168.0.0/16 10.1.0.0/24)'
```

BUGS

At the moment, `scanssh` leaves a one line entry in the log file of the ssh server. It is probably not possible to avoid that.

VERSION

This man page documents `scanssh` version 1.60b (Debian GNU/Linux package of version 1.6b)

BSD

July 17, 2000

BSD