# **InfraGard Technology Tool Kit Disclaimer Notice:**

# Created by Deborah Watson, Technology Tool Kit Program Development Manager, InfraGard - Houston Chapter; adapted from content created by Dave Armstrong, dave@bindshell.net

# IN NO EVENT SHALL INFRAGARD BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND/OR DOCUMENTATION, EVEN IF INFRAGARD HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

INFRAGARD SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE AND/OR ACCOMPANYING DOCUMENTATION, IF ANY, PROVIDED HEREUNDER IS PROVIDED "AS IS". INFRAGARD HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

# **Scanline**

Command line port scanner.

ScanLine is a command-line port scanner for all Windows platforms. It can perform traditional ICMP "pinging", optional additional ICMP TimeStamp scanning, can show host response times and number of hops, do TCP scanning, simple UDP scanning, banner grabbing and hostname resolving. Scanning is performed in a fast highly parallel fashion without resorting to using multiple threads. It can handle huge numbers and ranges of IP addresses without a problem.

This is the usage line as reported by typing "sl" or "sl -?"

ScanLine (TM) 1.01  
Copyright (c) Foundstone, Inc. 2002

sl [-?bhijnprsTUvz]  
[-cdgmq <n>]  
[-flLoO <file>]  
[-tu <n>[,<n>-<n>]]  
IP[,IP-IP]

-? - Shows this help text  
-b - Get port banners  
-c - Timeout for TCP and UDP attempts (ms). Default is 4000  
-d - Delay between scans (ms). Default is 0  
-f - Read IPs from file. Use "stdin" for stdin  
-g - Bind to given local port  
-h - Hide results for systems with no open ports  
-i - For pinging use ICMP Timestamp Requests in addition to Echo Requests  
-j - Don't output "-----..." separator between IPs  
-l - Read TCP ports from file  
-L - Read UDP ports from file  
-m - Bind to given local interface IP  
-n - No port scanning - only pinging (unless you use -p)  
-o - Output file (overwrite)  
-O - Output file (append)  
-p - Do not ping hosts before scanning  
-q - Timeout for pings (ms). Default is 2000  
-r - Resolve IP addresses to hostnames  
-s - Output in comma separated format (csv)  
-t - TCP port(s) to scan (a comma separated list of ports/ranges)  
-T - Use internal list of TCP ports  
-u - UDP port(s) to scan (a comma separated list of ports/ranges)  
-U - Use internal list of UDP ports  
-v - Verbose mode  
-z - Randomize IP and port scan order

**Example:** sl -bht 80,100-200,443 10.0.0.1-200

This example would scan TCP ports 80, 100, 101...200 and 443 on all IP addresses from 10.0.0.1 to 10.0.1.200 inclusive, grabbing banners from those ports and hiding hosts that had no open ports.

**Detailed option descriptions**  
**-?**  
Shows the usage of the program as in the above text.

**-b**  
Attempts to read the responses from the scanned ports and displays the results. If any part of the read text contains non-printable characters these will be replaced with spaces and multiple spaces reduced to single spaces. A maximum of 2 lines are displayed.

**-c**  
This is the connection timeout value for TCP ports and also the timeout value to wait for responses from UDP ports. For TCP scans, if no connection to the current port has been made to the remote host after this amount of time then it is assumed that there is no open port. For UDP scans, if no data response  
has been received from the remote host in the given time period or, with the case of the "extended" UDP scanning modes no ICMP port unreachable messages have been received after this time has elapsed it is assumed that the port is active.

**-d**  
Specifies a delay (in milliseconds) between each port scanned. This applies to all sections of the scan - ICMP, TCP and UDP. Use this option to perform a "drip" scan so as to help avoid detection by an IDS or if you have a slow network connection such as with a modem.

**-f**  
Reads IP addresses from the given file. You can successfully use the output file from a previous ScanLine session as input for this switch. However, any text file that contains valid IP addresses or address ranges can be used. Duplicates are removed before scanning.

**-g**  
Binds the local IP endpoint to a given port number. This is used for "source port" scanning and can sometimes successfully bypass firewall rules that allow packets through with low numbered source ports (53 and 88 for example).

**-h**  
Hides output from systems that have no open ports. Without this option all discovered live hosts will be displayed regardless of whether they had any open ports that were scanned for. Use this option to trim excess unwanted output.

**-i**  
When looking for "live" hosts, ScanLine will normally use a standard ICMP Echo Request packet. Providing this option will cause ScanLine to use an additional host discovery pass using ICMP Timestamp Requests instead of Echo Requests. Some systems will block Echo Requests but will respond to Timestamp Requests.

**-j**  
The output from ScanLine is normally made clearer by separating the information obtained from each IP address with a line of dashes. Use this option if you don't want the line separators to appear in the output.

**-l**  
Specifies a TCP port list text file to read ports from. Ports in the file are in lines in the same format as specified on the command line i.e. 79,80-88.

**-L**  
Specifies a UDP port list text file to read ports from. Ports in the file are in lines in the same format as specified on the command line i.e. 137,80-88.

**-m**  
Binds the local IP endpoint to a given IP interface. If your system has multiple network cards or interfaces you can tell ScanLine to use a preferred IP address to bind to. Note that you can only specify the IP address of an active interface on your own system - you can't "spoof" packets by using an arbitrary IP address here.

**-n**  
Specifies that no port scanning will take place. If all you want to do is discover live systems (ones that respond to ICMP) use this option. You can also specify -p (no pinging) in conjunction with -n if for example you simply wanted to resolve a list of IP addresses into hostnames.

**-o**  
Specifies a file that the output from the program is sent to. Note that output will also appear on the screen at the same time. If the file already exists it will be overwritten. If you want to append output to the file without overwriting it then use the "-O" option.

**-O**  
Same as -o but appends the program output to the text file instead of overwriting it.

**-p**  
Don't "ping" each host before scanning. ScanLine will make no attempt to discover live hosts for scanning if this is used - it will simply go ahead and port scan all hosts regardless if they exist on the network or not. You could use this option to scan for systems that had certain ports open but were blocking ICMP packets.

**-q**  
Specifies the maximum time that the program waits for a response from a ping. If no ping reply is received after this amount of the time the remote host will be regarded as "dead" and won't be scanned.

**-r**  
Resolve IP addresses to hostnames. An attempt will be made to resolve each live IP address into a hostname. Even though the scan process in ScanLine is single threaded, up to 8 threads will be used for resolving hostnames if this option is specified

**-s**  
Output in csv format. CSV format is usually used when you want to import a file into a spreadsheet program for further analysis. The format of the line that ScanLine outputs when this option is used is  
<IP>,<hostname>,<ping response time>,<hops>, <responds with ICMP unreachable>,<open TCP ports>,<open UDP ports>. No port banners are included in this output and if -b is specified on the command line it will be ignored when used in conjunction with -s.

**-t**  
TCP port numbers to scan. The port numbers can be single numbers separated by commas and/or port ranges separated by the "-" character. For example sl -t 60,70,80-100 10.1.2.3 would scan TCP ports 60, 70, and 80 through 100 inclusive. No spaces are allowed between any of the numbers or ranges.

**-T**  
Use the built-in TCP port list. If you don't explicitly specify any ports to scan on the command line, ScanLine will go ahead and use its built in list of ports. If you DO specify ports and you want to also include ScanLine's internal list of TCP ports use -T. For example, if you wanted to scan for UDP  
port 137 together will all built-in TCP ports you'd specify sl 10.1.2.3 -u 137 -T

**-u**  
UDP port numbers to scan. The port numbers can be single numbers separated by commas and/or port ranges separated by the "-" character. For example sl -u 60,70,80-100 10.1.2.3 would scan UDP ports 60, 70, and 80 through 100 inclusive. No spaces are allowed between any of the numbers or ranges. This UDP scanning option uses known data probes for many common UDP services and are designed to produce a response from the scanned system. This ensures total accuracy of the UDP scanning; for the remote system to respond with a UDP packet the port that was scanned must by definition be open.   
Under normal circumstances sending a UDP packet to a closed port will cause an ICMP Destination Port Unreachable message to be sent back. I say "under normal circumstances" because often a system will be filtered in some way such that it never responds in this fashion. This would lead to lots of false  
positive open ports. So what ScanLine does with it's UDP scanning is to first send a UDP probe to a "known closed port" on the target to see if it actually responds with an ICMP message. For this purpose ScanLine uses UDP port 1. If we do get an ICMP response the program will then assume that all further ports scanned that do NOT generate an ICMP message are open.

\*\*\*\*\* Windows 95/98/ME systems do not register ICMP destination port  
\*\*\*\*\* unreachable messages via Winsock due to subtle differences in the way  
\*\*\*\*\* Winsock is implemented when compared to Windows 2000 and XP. Thus  
\*\*\*\*\* you will never see any open UDP ports detected using this technique  
\*\*\*\*\* on these operating systems.

**-U**  
Use the built-in UDP port list. If you don't explicitly specify any ports to scan on the command line, ScanLine will go ahead and use its built in list of ports. If you DO specify ports and you want to also include ScanLine's internal list of UDP ports use -U. For example, if you wanted to scan for TCP port 80 together will all built-in UDP ports you'd specify sl 10.1.2.3 -t 80 -U

**-v**  
Verbose mode. Additional information will be shown if you set the program to verbose mode.

**-z**  
Use this to randomize the order that IP addresses and ports will be scanned in. Normally ScanLine will scan the provided IPs and ports in numerical order. By using the "-r" option the IPs and port numbers will be "shuffled" into a random ordering. This can help to some extent in avoiding detection by intrusion detection systems.

**Default Port Lists**  
The internal port lists used in ScanLine are as follows:

**UDP ports**

7 9 11 53 67-69 111 123 135 137 138 161 191 192 256 260 407 445 500 514 520  
1009 1024 1025 1027 1028 1030 1033 1034 1035 1037 1041 1058 1060 1091 1352  
1434 1645 1646 1812 1813 1900 1978 2002 2049 2140 2161 2301 2365 2493 2631  
2967 3179 3327 3456 4045 4156 4296 4469 4802 5631 5632 11487 31337  
32768-32790 43981

**TCP ports**

7 9 11 13 15 19 21 22 23 25 43 49 53 66-68 70 79 80 81 88 89 98 109 110 111  
113 118 119 135 139 143 150 156 179 256-259 264 389 396 427 443 445 457 465  
512-515 524 540 563 587 593 636 691 799 900-901 1024-1031 1080 1100 1214  
1243 1313 1352 1433 1494 1498 1521 1524-1525 1529 1541 1542 1720 1723 1745  
1755 1813 1944 2000 2001 2003 2049 2080 2140 2301 2447 2766 2779 2869 2998  
3128 3268 3300 3306 3372 3389 4000 4001 4002 4045 4321 4444 4665 4899 5000  
5222 5556 5631 5632 5678 5800 5801 5802 5900 5901 6000 6112 6346 6347 6588  
6666-6667 7000 7001 7002 7070 7100 7777 7947 8000 8001 8010 8080-8081 8100  
8383 8888 9090 10000 12345 20034 27374 30821 32768-32790

**IP address and hostnames**  
Any item on the command line that is not associated with a "-" character is assumed to be an IP address or hostname. The addresses can be single numbers separated by commas and/or IP ranges separated by the "-" character. For example

sl -t 80 10.1.2.3,10.1.2.4,10.1.2.5-10.1.2.20

would scan TCP port 80 on the machines 10.1.2.3, 10.1.2.4 and 10.1.2.5 through 10.1.2.20 inclusive. No spaces are allowed between any of the numbers or ranges.

To make life easier you can also specify ranges in a number of ways :-

10.1.2.2-254  
would add all IPs from 10.1.2.2 to 10.1.2.254 inclusive.

10.1.2.2-3.254  
would add all IPs from 10.1.2.2 to 10.1.3.254 inclusive.

Hostnames can be used in place of IP address, although obviously they do not  
make sense as part of an address range e.g.

sl -p 2-200 www.microsoft.com

**To best illustrate the use of ScanLine here are a series of examples.**  
Example #1:   
Scan machine at 10.0.2.2 for all TCP ports from 1 to 200.

sl -t 1-200 10.0.2.2

Example #2:  
Scan machines from 10.0.2.2 to 10.0.2.254 for all TCP ports in the range 1 to 65535 sending the output to the file "out.txt"

sl -t 1-65535 10.0.2.2-254 -o out.txt

Example #3:  
Scan machines from 10.0.2.2 to 10.0.2.20 for TCP ports 80, 81, 88, 8000 and 8080 and UDP ports 31337 without pinging and append the output to the file "out.txt"

sl -pt 80,81,88,8000,8080 -u 31337 10.0.2.2-20 -O out.txt

Notice the use of the combined options "pt". This is perfectly legitimate.

Example #4:  
Scan machines 10.0.2.2 to 10.0.2.254 for TCP ports 21 and 25 and show the banner responses. Include extra output information.

sl -vbt 21,25 10.0.2.2-254

Example #5:  
Scan machines 10.0.2.1 to 10.0.2.254 for TCP ports 1 to 65535, showing the banner responses and with a delay of 5 seconds between each port scan. Include extra output information.

sl -vbt 1-65535 10.0.2.1-254 -d 5000

Example #6:  
Scan machines 10.0.2.1 to 10.0.2.254 with no port scanning, only pinging. Resolve the IPs into hostnames.

sl -nr 10.0.2.1-254

Example #7:  
Scan machines 10.0.2.1 to 10.0.2.254 with no port scanning or pinging. Resolve the IPs into hostnames.

sl -npr 10.0.2.1-254

Example #8:  
Scan machines 10.0.2.1 to 10.0.2.254 for open UDP ports.

sl -U 10.0.2.1-254

Example #9:  
Scan for systems that have open TCP ports 80 or 443, not caring if they respond to pings, then having found those perform a full port scan on only those systems.

sl -hpt 80,443 10.0.2.1-254 | sl -f "stdin"