

## **Nmap scanning tool**

Asianux 2.0 provides a network exploring tool and security scanner ---- Nmap.

Scanner is a kind of program that automatically examines security weak of long-distance or local host computers. Use scanner, we can obtain plenty of information from long-distance server. Through this information, we can know about security problem of long-distance host, so that we can mend hidden trouble of system in time. At the same time, scanner can provide much convenience for attacker, it can simplify their work.

Generally, scanner firstly send request to long-distance TCP/IP port, record the reply of target host, and then analyze replied information. Through this method, it can search for variety of useful information about target host. Such as: whether port is open, provided services, software version and whether it can log in anonymously.

*Scanner generally is not a direct network leak attacker program; it only can help us to find some inner weak of target machine.*

### **1 Brief introduction of Nmap**

Nmap is a port scanning tool, it provides many scanning technologies and many advanced functions.

Nmap can scan a host computer, and it also can scan a net section. To scan a host computer, it only needs to appoint host name or IP address. To scan a net section, it can have several methods, they have equal effect: 192.168.1.1/24, 192.168.1.\* or 192.168.1.0-254.

Users can select the type of network that Nmap scans, Nmap can scan TCP, UDP and TCP synchronization, it can use ping command to examine whether the host computer is open or not. Nmap can also scan one or several special ports. For example, set scanning port number to be 20-30, 80 or 6000-, It can scan ports that its number it 20-30, 80 and above 6000.

The running results of Nmap is the port list of scanned host computers (host group), usually it gives port number, service name and state.

*If you want to know more information about Nmap and advanced functions, please refer to its main page: <http://www.insecure.org/nmap/>.*

### **2 User guide of Nmap**

The grammar of Nmap commands is very simple. Different options of Nmap and -s sign compose of different scanning type, for example: a Ping-scan command is "-sP". After confirmed target host and network, it can start to scan. If we use root to run Nmap, its will have more powerful functions, because super user can establish customized data packet that convenient for Nmap to use.

The grammar format of Nmap commands is as following:

```
nmap [scan type] [option] host or network
```

#### **2.1 Description**

The original intention of designing Nmap is that allowing administrator to examine which host contained by a large network system and which kind of services are running on it. It supports scan of several kinds of protocols, such as: UDP, TCP connect (), TCP SYN (half open), ftp proxy (bounce attack), Reverse-ident, ICMP (ping sweep), FIN, ACK sweep, Xmas Tree, SYN sweep

and Null.

Nmap also provides some advanced functions, such as distinguish OS type through TCP/IP, secretly scanning, dynamically delay, resend, parallel scanning, detect underlying host through parallel Ping, fraudulent scanning, port filtering detecting, direct RPC scanning, distributed scanning, agile object selecting and description of port. After running Nmap, we usually get a port list about scanned host computer. Nmap always displays this server's name, port number, state and protocol. It has three states: open, filtered and unfiltered. Open means that target machine will accept your connecting request at this port; filtered means that there has firewall, filtering advice or other network obstacles that obstruct Nmap from finding out whether port is opened or not; unfiltered only appear when most scanning port are at the state of filtered.

After using some parameters and options, Nmap can also report the following characteristics of long-distance host: used system, TCP continuity, name of application user that is used on each port, DNS name, whether host is a smurf address and some other functions.

## 2.2 Option and parameter

The following options all can be used assembly. Use parameter can accurately define a scanning mode. Nmap will give hints if there has some un-canonical parameter compages.

*If hurry to start, refer to [section 3.3: Examples of using Nmap](#).*

- Scan type

-sT

TCP connect () scan: this is basic detecting form to TCP. This connect () detects corresponding port on target host, is this port is monitoring, this connecting is successful, otherwise, this port can not arrive. The advantage of this technology is that it is unnecessary for to have any especial right and this command can be used freely by any user under most UNIX system.

This kind of detecting is easy to be caught and recorded by target host. It will display a group of information about connection and errors to server that connected by accept (), and make it close at once.

-sS

TCP SYN scan: this kind of technology doesn't need to open an entire TCP connection, it is usually considered as a "half open" scan. When you send a SYN information packet, it seems like opening a real connection and waiting for reply of opposite side. A SYN|ACK (reply) will indicate that this port is open for monitoring and a RST means that this port has not been monitored. If it returns SYN|ACK answer, then send a RST packet at once to intermit this connection. This biggest advantage of this scan is that only few sites will record it, but it needs root popedom to customize these SYN packets.

-sF -sX -sN

Stealth FIN, Xmas Tree or Null scan mode: sometime, even SYN is not covert enough. Some firewall and packet filter will guard at important port and SYN will be intercepted. Some application such as Synlogger and Courtney are good at detecting this kind of scanning. So, it needs further scan that can pass them in the situation that does not meet troubles.

New thought is that the closed port will return a RST to the detecting information packet sent by you. And the opened port will not ignore it (Please refer to RFC 793pp64). FIN scanning uses a null FIN information packet as a detector; Xmas tree scanning uses FIN, URG and PUSH sign; Null scanning does not use any sign. But unfortunately, Microsoft decides to completely ignore this standard. So, this scan type can not work under WINDOWS 9X and NT.

Seeing from positive aspect, actually this is a good method to distinguish two kinds of platform. If it finds open port, we can know that this machine does not run WINDOWS. If all `-sF`, `-sX` and `-sN` scanning displays that all ports are closed, but a SYN (`-sS`) scanning displays that there has open port, then we can conclude that it is WINDOWS platform.

This is just a simple application. Now, Nmap has more thorough method to distinguish operating system (of course, its theory is similar with above methods), this platform includes Cisco, BSDI, HP/UX, MVS and IRIX.

`-sP`

Ping scan: sometime we just want to know which host is open on network; Nmap can make it through sending ICMP echo request information packet to appointed IP address. If it has replay, this host is open.

But, some sites such as microdoft.com set obstacle to echo request packet. This way, Nmap also can send a TCP ack packet to 80 port, if it receives RST return, it means this machine is open; the third method is that send a SYN information packet and wait for RST or SYN/ACK responds. This can be used by non-root users.

To root user, defaulted Nmap uses ICMP and ACK methods to scan. Of course, we also can modify `-P` option.

It is better to ping host at first, only the host that responds is necessary to be scanned, you can search active host before actual deeply scanning.

`-sU`

UDP scan: this method is used to confirm which UDP (User Datagram Protocol, RFC 768) port on host is open. This technology will send 0-byte UDP packet to each port of target computer, if we have received a replay that ICMP port can not be reached, then this port is closed, otherwise, we consider that it is open.

But, firewall always blocks the message that port's unreachable. This leads that port seems to be open. Sometime, a ISP just blocks several special dangerous ports such as 31337 (back door) and 139 (Windows NetBIOS), let these ports that easy to be attacked seems to be open. Unfortunately, distinguishing real open UDP port and these initiatively filtered ports is not always easy.

Some people think that UDP scan is meaningless. Rpcbind will hide in a informal UDP port on some 32770, so it is inessential to carry out firewall filtering on 111. But have you ever found that the port above 30,000 are monitoring? Using UDP scan can be easy to make it. Think about Back Orifice horse that produced by cDc, it can configure a UDP port in Windows computer, let alone so many services that can use UDP and are easy to be attacked, such as snmp, tftp, NFS etc.

But we have to mention, because most host carry out a suggestion of limiting ICMP error information rate according to RFC 1812, sometime UDP scan is very slow. For example, Linux kernel program (`net/ipv4/icmp.h`) limits to create 80 unreachable information each 4 seconds (every time create 1/4 second's delay); Solaris has more strict limit (about every second has two

delays), so it will cost much time on scanning. Nmap will detect this limit and slower its speed, but not fill in network with large quantity of information that is useless and will be ignored by target computers.

Microsoft ignores suggestions of RFC, and seems like that it does not set any rate limit on Win95 and NT computers. This way, we can scan ports that are almost 65K on Windows computer with high speed.

-sO

IP protocol scan: this method is used to confirm the IP protocol that a host supports. This technology will send raw IP packet (does not include any protocol head) to every appointed protocol of target computer. If we receive an unreachable message of an ICMP protocol, this means this protocol has not been used. Otherwise, we assume that it is open. Please notice: some host (AIX, HP-UX, and Digital UNIX) and firewall may not send unreachable messages of protocols. This will lead that all protocols seem like “Open”.

Because that performed technology is similar with UDP port scan, rate limit of ICMP may work. But IP protocol domain is only 8 bits, so most 256 protocols can be detected in reasonable time.

-sI <zombie host [:probeport]>

Idlescan: this advanced scan method allows scanning a real convert TCP port on target computer (this means that there is no packet that is sent to target host from your real IP address). Contrarily, a unique channel access attack will use predictable “subsection ID of IP” on zombie host to collect information about open ports of target machine. IDS system will display and scan ports that seem like coming from appointed zombie computer (it must have good capacity and accord with some special standard).

Except excessive concealing, this scan type allows mapping credit relationship that based on IP between two machines. Port list displays open port in zombie host picture. So, you can try to scan an target computer by using different zombies that you consider it can be trusted (through router/packet filter principle). Apparently, these information is very important when distinguishing attacking object. Otherwise, your detecting may not help to costing much resource to “own” a “middle” system, but only to find that its IP even can not be trust by target host/network that followed by you.

If you hope to detect an especial port for IPID modification on zombie host, you can add a colon after a port number. Otherwise, Nmap will use defaulted port of “tcp ping”.

-sA

ACK scan: this advanced method is usually used to mapping principle muster of firewall. Especially, it can help to confirm whether firewall is static or just a simple packet filter that prevent SYN from entering.

This scan type sends an ACK packet to appointed port. If it returns a RST, this port will be classified to be “unfiltered”; if it does not return any information or it returns unreachable information of ICMP, this port will be classified to be “filtered”. Please notice: Nmap usually do not print “unfiltered” port, so if receiving “no port” display, it usually means that all detectors have passed (and return RST). Obviously, this scan will never display port using “open” state.

-sW

Window scan: except that sometime it can examine open port, and because of filtered/nonfiltered that led by OS's report of abnormality in TCP window, this advanced scan is similar with ACK scan on other aspects. Systems that are easy to be attacked at least include: AIX, Amiga, BeOS/BSDI, Cray, Tru64 UNIX, DG/UX, OpenVMS, Digital UNIX, FreeBSD, HP-UX, OS/2, IRIX, MacOS, NetBSD, OpenBSD, OpenStep, QNX, Rhapsody, SunOS 4.Ultrix, VAX and some versions of VxWorks.

-sR

RPC scan: this method is one mode that combines with several nmap scan. It obtains all TCP/UDP open ports and fills in with NULL command of SunRPC program; its aim is trying to confirm whether they are RPC ports, if it is RPC port, what program is running on it and which version. This way, even if target host hides behind firewall or defended by TCP wrappers, it can obtain information whose effect is similar with 'rpcinfo -p'. But now Decoys can not work normally under RPC scan, in future it may add decoy support to UDP RPC scan.

-sL

List scan. This method simply creates and prints a IP/name list, do not carry out actual ping or port scan. Unless using -n, otherwise, DNS name interpreting will not be carried out.

-b <ftp relay host>

FTP bounce attack: an interesting characteristic of FTP protocol is that it supports proxy FTP connection (RFC 959). In other words, we can connect to a FTP Server target.com from evil.com and require target host to send file to "anywhere" on Internet! After this RFC was written in 1985, this character was put in force. But on recent day's network, we do not allow people to arbitrarily "hijack" ftp server and attack others. When Hobbit wrote the bug of this protocol in 1995, he said: "It can send mails, news and aggressive behaviors that are actually hard to be traced from many sites; fill in your hard disc, try to make firewall disabled or more boring and meaningless trouble". We use it in order to scan TCP port from proxy FTP Server; this way, we can connect to a FTP Server after firewall, and then scan ports that may be blocked (139 is a good example). If FTP Server allows reading and writing some records (such as /incoming), it can send any data to the open port that you found (even if nmap will not do this).

When using "b" option to make host be your proxy, standard URL format is: username:password@server:port, any content is optional except server.

- General option

These options are not necessary, but some of them may be useful.

-P0

Do not try to PING host before scan; it is used to scan hosts or networks that do not allow ICMP echo request (or response). microsoft.com is an example, it must use -P0 or -PT80 to scan port of microsoft.com.

-PT [portlist]

Use ping of TCP to confirm whether host is opened, but not the mode that send ICMP echo request packet and wait for responding, we can send plenty of TCP ACK packets to target network (or single computer) and wait for response bit by bit, opened host will return a RST. This option can let you scan a network/host with high efficiency all the same when ping information packet is blocked. To non-root user, we use connect (), using following format to set target detector: -PT <port1> [,port2][...], defaulted port is 80. This is because that this port is not filtered usually. Please notice that this option can accept several port numbers that are separated by comma.

-PS [portlist]

This option is used by root user, use SYN (connection request) packet to substitute for ACK packet. Opened host will return a RST (or SYN|ACK, but singularly). You can set target port by using the same mode with -PT.

-PU [portlist]

This option sends UDP detecting to the appointed UDP. If host is running well, it will obtain an unreachable information packet of ICMP port (or if this port is open, it may be a UDP response). Because that many UDP services will not respond to a blank packet, it is best to consider that this blank packet is sent to a prospective closed port, but not open port.

-PE

This option uses a real ping (ICMP echo request) packet. It finds open host and search for all broadcast addresses on this subnet ---- this broadcast address is the address that can receive and interpret IP packets. If you find that they can allow large quantity of denied service attack (Smurf is most hackneyed), you should delete them.

-PP

Use an ICMP timestamp request (serial number is 13) to search for monitor host.

-PM

Except using network mask request (ICMP number is 17), other aspects are similar with -PE and -PP.

-PB

Defaulted ping type. It uses ACK (-PT) and ICMP echo request (-PE) to carry parallel attack. This node can pass packet filter or firewall. Target port of TCP detector can be set by same mode with -PT. Please notice that because pingtype sign now can be combined, this sign is objected. So, you should use "PE" and "PT" to get same effect.

-O

This option distinguishes OS type of long-distance host through TCP/IP fingerprint identifying. In other words, it uses a series of packet to detect that the scanned host locates at correlative stack information of OS and distinguish their refined difference, in this way to distinguish OS. It uses found information to set up a "fingerprint", this fingerprint is used to compare with known OS fingerprint identifying (nmap-os-fingerprint file) database, and in this way there has basis to

distinguish OS.

If Nmap can not speculate OS of computer, but there has good condition (for example, at least one port is open), then if you want to know exactly that OS is running on computer, Nmap will provide a URL that you can use it to submit fingerprint identification. This provides useful information for OS identifying base known by Nmap, in this way, this function will be more exact. Please notice that if you left an IP address in table, computer may be scanned when we add fingerprint identifying (to confirm it is running).

-O option also can start other test. One test is measuring “normal running time”, this test uses TCP timestamp option (RFC 1323) to speculate when a computer restart for the last time. This information only report to the computer that provides this information.

Another test started by -O is predictable classifying of TCP list. This measure approximatively describes how hard it is to build a spurious TCP connection to counterwork long-distance host. This is useful for credit relationship based on source IP (long-distance login command, firewall filter etc.) or concealing an attack source.

When miscellaneous mode starting with -O, IPID list will also be reported. Most computer are in “incremental” class, its meaning is that they add “ID” domain for each packet in IP head. This will make them easy to be attacked by several advanced information searching and cheat.

-6

This option starts IPv6 support. If using this option, all target hosts must be IPv6, and they can be appointed through routine DNS name (AAAA record), or they can be appointed as a literal IP address such as 3ffe:501:4819:2000:210:f3ff:fe03:4d0. Now, connect ( ) TCP scan and TCP connect ( ) scan are supported.

-I

This will start reverse ident scan. As Dave Goldsmith said in Bugtraq in 1996, ident protocol (rfc 1413) allows get username of process through TCP connection ---- even if this connection is not started by this process. So you can connect to http port, and then use identd to confirm whether server is running by using root. This only can be accomplished by an entire TCP connection to target port (such as -sT scan option). Use -I option, identd of long-distance host accept connection inquiring at open port, apparently, if host does not run identd, it can not work normally.

-f

This option configuration implements scan request of SYN, FIN, XMAS or NULL through fine IP fragment. This thought is that respectively put TCP head into different information packets, let packet filter and inbreak detecting hard to carry out, and then you can crash into system and do anything you want. But you have to notice that it is difficult for some program to deal with these fine packets. For example, my favorite sniffer subsection appears some problem when received the first 36-byte information. Later, another 24-byte! When packet filter and firewall that can arrange IP fragment don't obtain this order (like CON-FIG\_IP\_ALWAYS\_DEFRAG option in Linux kernel), some network system can not reflect that the object is found and quit.

Remember that this parameter may not always work well on any system, it is normal under my Linux, FreeBSD and OpenBSD. Of course that some one says it can work under parts of different

\*NIX environment.

-v

Detailed mode. This is an intensively commended option. It can publicize more information of executing operation. You can reuse it to gain larger effect. If you want to page screen largely, please use `-d` twice.

-h

A shortcut help option, it can display using method of nmap on screen. As you have noticed, this man page is not a “quick accidence reference”.

-oN <logfilename>

This option is used to appoint a file which record scan result; this result is convenient for reading.

-oX <logfilename>

This option will use XML format to record scan result to appointed file, this will allow program to conveniently catch and explain Nmap result. You can send output to stdout (shell pipeline etc.) by giving variable “-”. In this situation, routine output will be restrained. If you use this option, please notice nearly to error message. At the same time, please notice “-v”, it may lead that some additional message will be printed. DTD that defines XML output frame can be found in <http://www.insecure.org/nmap/nmap.dtd>.

-oG <logfilename>

This option records scan result to appointed file by using grepable format. This simple format provides all information in line (in this way, it can be easy to search for port or OS information by using `grep` command, and see all IP). This usually is preferred mechanism of program that interacts with Nmap, but we recommend using XML output (-oX). Information that is included in this simple format may not so much as other format. You can send output to stdout (shell pipeline etc.) by giving variable “-”. In this situation, routine output will be restrained. If you use this option, please notice nearly to error message. At the same time, please notice “-v”, it may lead that some additional message will be printed.

-oA <basefilename>

This option will tell all main format of Nmap (routine, grepable and XML). It can give a base for filename; the output file will be `base.nmap`, `base.gnmap` and `base.xml`.

--resume <logfilename>

If network scan is canceled because of control-C or network intermitting and so on, it can use this option to continue. Failed scan record file must be a routine (-oN) record or grepable (-oG) record. It will not give other options (they are same with failed scan record). Nmap will start to scan from the last successfully scanned record in log file.

--append\_output

Add Nmap scan result into any appointed output file, but not rewrite those files.

-iL <inputfilename>

Read data from appointed file but not command-line. This file can store a list of host or network, and separate them by space, TAB key or newline. If you hope to read from standard output device (file) ---- such as end of pipeline prompt, you must use hyphen (-) in filename. Please refer to object standard to get more information about using expression to fill in file.

-iR <num hosts>

This option tells Nmap to create self host through simply gather arbitrary data to carry out scan. When given IP was scanned ---- after using 0 on a nonstop scan, this operation will never end. This option is useful for statistic swatch of Internet estimating all kinds of things. If feel boring, please try `nmap -sS -PS80 -iR 0 -P 80` to find some network server and check.

-p <port range>

Appoint the port you would like to scan. For example, “-p 23” will only detect 23 port of target host. “-p 20-30, 139, 60000-” will scan 20 to 30, 139 and all ports above 60000. The ports that it defaults to scan is 1 to 1024, or ports that listed in services file of nmap. To IP protocol scan, this option will appoint the protocol number (0-255) that you want to scan.

When scanning TCP and UDP ports, it can appoint a special protocol through the above port number of “T:” or “U:”. Limitation prompt will not disappear until another limitation prompt is appointed. For example, variable “-p U:53, 111, 137, T:21-25, 80, 139, 8080” will scan port 53, 111 and 137, and listed TCP ports. Please notice that in order to scan UDP and TCP, it needs to appoint -sU and at least one TCP scan type (such as -sS, -sF or -sT). If it does not appoint limitation prompt of protocol, port number will be added into all protocol lists.

-F fast scan mode

Appoint that only want to scan ports that listed in services file (or protocol file of -sO) in nmap. This will obviously faster than scanning all 65535 ports.

-D <decoy1 [, decoy2] [, ME],...>

This is a scan that have inveiglement mode, it will record all inveigling addresses that you have appointed in connection record of long-distance host. In this way, their data memorizer will display that some port scan is started from some IP, but they can not distinguish which one is the real IP and which one is used as coverture, this can frustrate some track behavior through route, so it is an applied technology to conceal your IP.

Use comma to separate each inveigling address, you can put “ME” to any place where you want to display real IP, if you put “ME” at the sixth bit or even the end, some port scanning recorder (such as Solar Designer’s excellent scanlogd) may not display your IP, if you don’t use “ME”, nmap will put it randomly.

Remember that the host you use to inveigle must be open or you can scan your object. Because that it is very easy to distinguish which one is the real intruder from many actually useless IP addresses. You may also use IP address to substitute for name, in this way, it will not record you in name server logs of inveigled host.

You have to remember that some (stupid) “quot; port scanning detector” will deny port scanning attempt that arrives host. This will unconsciously lead that it will lose connection between host

that you scanned and “inveigled host”, it may bring a big problem ---- if this “inveigled host” is a gateway or even it is its local host, its connection will also be cut! So you must be careful to use this parameter.

This kind of inveiglement can be used in initial ping scan and actual port state scan, it can also be used in long-distance OS distinguishing (-O).

Of course, it is useless if you write too many inveigling addresses; this can just slow down scanning and decrease its precision. And some command dealing system may filter your inveigling packet, though most (almost all) will not put any limitation to inveigling packet.

`-S <IP_Address>`

In some situation, nmap can not make sure your source address ---- in this situation nmap will have hint, now you have to use `-s` and IP address to label.

Another possibility is to use it to inveigle object to make it consider that someone is scanning it. Think about that some company finds that it is scanned by competitor continually; this is usage that is not supported, or we can say that it is not the main aim. I just use it to mention people that don't just censure when find a port scanner, it may be inculpable. `-e` can explain general usage of this parameter.

`-e <interface>`

Tell nmap to use which interface to send or receive packets. nmap can detect it automatically, if it can not make it, there will have hint.

`-g <portnumber>`

Set source port number that used in scanning. Many “innocent” firewall and packet filter will filter all connections except the connection that they established to allow packets of DNS (53) or FTP-DATA (20). This is obviously imprudent, because intruder can edit a source port that comes from FTP or DNS easily. For example, if you can not get information from host:port of host through TCP ISN, then through using `-g` command, nmap will change source port and try again.

Please notice that this is just a request ---- nmap will allow doing this only when it can do this. For example, you can not carry out all samplings of TCP ISN from one host:port to another host:port. So even if you have used `-g`, nmap will change source port also.

You have to know that there may have short delay if you use this option, because sometime it needs to store some useful information in these source port numbers.

`--data_length <number>`

Generally, Nmap will send a lowest request packet that only contains caption head. So, its TCP packet is usually 40 bytes, ICMP echo request is 28 bytes. This option will tell Nmap that most packets that have been sent to it have an additional 0 full bit of given number. OS detecting packet will not be influenced, but most ping and port scanning packets will be influenced. This will lead the speed slow down, but not very obvious.

`-n`

Tell Nmap never to carry out inverted DNS interpreting in active IP address it found. Because that DNS is generally slow, this will help it to speed up.

-R

Tell Nmap always to carry out inverted DNS interpreting in target IP address. Generally, it will do this only when there just one computer is found to be active.

-r

Tell Nmap not to arbitrarily arrange order of scanned port.

-ttl <value>

Send packet to given value to set IPv4 time for active domain.

--randomize\_hosts

Tell Nmap to disarrange every group into 2048 host before starting to scan. This will make scan seem not so obvious in different network monitoring systems, especially when using it with slow timing option.

-M <max sockets>

Set biggest defaulted quantity of sockets that used to carry out parallel TCP connect ( ) scan. This will slow down scan so as to avoid breakdown of long-distance host. Another approach is to use -sS, this is generally easy to deal with for computer.

--packet\_trace

Tell Nmap to display all sent and received packets by using format that is similar with tcpdump. This option is useful for debugging, and it is a very good learning tool.

--datadir [directoryname]

When Nmap is running, we can get some special data from files named nmap-services, map-protocols, nmap-rpc and nmap-os-fingerprints. Nmap will search for these files in directory options first. Any file that has not been found here will be searched in directory that is appointed by NMAPDIR environment variable; the next is ~/nmap, then the location where the program is edited, such as /usr/share/nmap; the last method is that let Nmap search in current directory.

- Time option

Though Nmap can accomplish scan task as quickly as possible in running time generally, but occasionally it can not detect some hosts or ports, this may because that the defaulted time strategy of Nmap is different from your target, the following are some options to control scanning time.

-T <Paranoid|Sneaky|Polite|Aggressive|Insane>

This is a parameter setting that conveniently expresses Nmap time strategy priority. Paranoid mode scans by very slow speed to avoid being monitoring by number record system, it makes scan continuous but not parallel and usually has to wait for at least five minutes to send a packet. Sneaky is similar, but it sends a packet every 15 seconds. Polite mode is used to lighten network load so that to decrease the possibility of breakdown, it sends probe continuously and waits for 0.4 second between the intervals of two packets. Normal is the routine usage of Nmap, it scans as

quickly as possible ---- unless it lost host or port connection. Aggressive mode is to set five minutes' timeout for each probe, and wait for probe response for not more than 1.25 seconds. Insane mode is to adapt very fast network or you don't care that it may lose some information, its timeout is set to be 75 seconds and just waits for response for 0.3 second, it allows "sweeping" a fast network system.

You can also use number (0-5) to delegate parameters, for example, "-T0" means Paranoid mode, and "-T5" means Insane mode.

--host\_timeout <milliseconds>

Concretely appoint time gross that Nmap scans some IP, it will give up dealing with if time is over, and default is not to set.

--max\_rtt\_timeout <milliseconds>

Appoint the maximum time that Nmap respond to a detector from long-distance object, and default is 9000.

--min\_rtt\_timeout <milliseconds>

When target host starts to set up a responding mode, Nmap will shorten time that given to each detector. It will speed up scanning, but if it costs more time than average to respond, it will lose packet. Use this parameter, it can ensure that Nmap will wait for a given time before giving up to a detector.

--initial\_rtt\_timeout <milliseconds>

Appoint timeout of initial detector, generally it is effective when using -P0 to scan host that is protected by firewall, Nmap will get a good RTT evaluation through ping and initial detector information. Default is 6000.

--max\_parallelism <number>

Appoint maximum quantity of parallel scan that nmap allows, set to be 1 means Nmap scans one port every time, it will also influence other scans such as ping sweep, RPC etc.

--min\_parallelism <number>

Tell Nmap to scan ports' quantity that is given by using parallel mode. This can speed up the scan that arrack some special firewall host through a value, but you have to notice that if it carry out too fast, the result will be irresponsible.

--scan\_delay <milliseconds>

Appoint the minimum time that Nmap have to wait between two detecting. This is an effective method that can lighten network load and make scan not so conspicuous in integrated data storing record.

- Target criterion

All options that do not carry parameter will be considered as target host description of nmap. The simplest example is that only list out single host name or IP address in command line. If you want to scan a subnet, you can add "/mask" to host name or IP address. Mask must be between 0 (scan

whole network)-32 (special single host). Use /24 to scan a C class address, and use /16 to scan B class address.

Nmap also has some more useful signs that are used to appoint all kinds of network addresses. For example, you want to scan some B class network address, then you can use “128.210.\*.\*” or “128.210.0-255.0-255” or even “128.210.1-50, 51-255.1, 2, 3, 4, 5-255” to express. Of course, it can also use mask to express: ‘128.210.0.0/16’, these are all equal.

Another interesting usage is that it can “separate” the whole network, for example, we can use “\*.\*.5.6-7” to scan all IP addresses that ended with .5.6 or .5.7. If you want to get more information, please refer to the net section: [Example of Using Nmap](#).

### 3 Example of using Nmap

Because that the function of Nmap is powerful, and options are too many, so there needs some experience if you want to use it expertly. The following are some example about how to use Nmap:

```
nmap -v target.example.com/24
```

Scan all TCP ports on host target.example.com. -v means using detailed mode. In default situation, nmap will use -sT scan mode.

```
nmap -sS -O target.example.com/24
```

Start a SYN half-open scan, its target is the C class net where target.example.com locates, it also try to make sure what kind of system is running on it. Because that it uses half-open scan and system detecting, it needs to use root right.

```
nmap -sX -p 22, 53, 110, 143, 4564 198.116.*.1-127
```

Send a Xmas tree scan to half of the range of B class subnet that 198.116 locates, we will detect whether system runs sshd, DNS, pop3d, imapd or port 4564. Please notice, because Microsoft TCP stack is not perfect, Xmas scan can not run successfully on its platform, the same problem may exist in CISCO, IRIX, HP/UX and BSDI.

```
nmap -v - - randomize_hosts -p 80 *.*.2.3-5'
```

This is a mode that locates a net domain (it will separate the whole network to many small parts) first and scans then, the addresses scanned here are IP addresses that ended with .2.3, .2.4 or .2.5. If it is root, it can also use -sS.

```
nmap -I company.com | cut '-d' -f 4 | ./nmap -v -iL -
```

Use a DNS domain conversion to look for host in company.com and send IP address to nmap.