Nmap

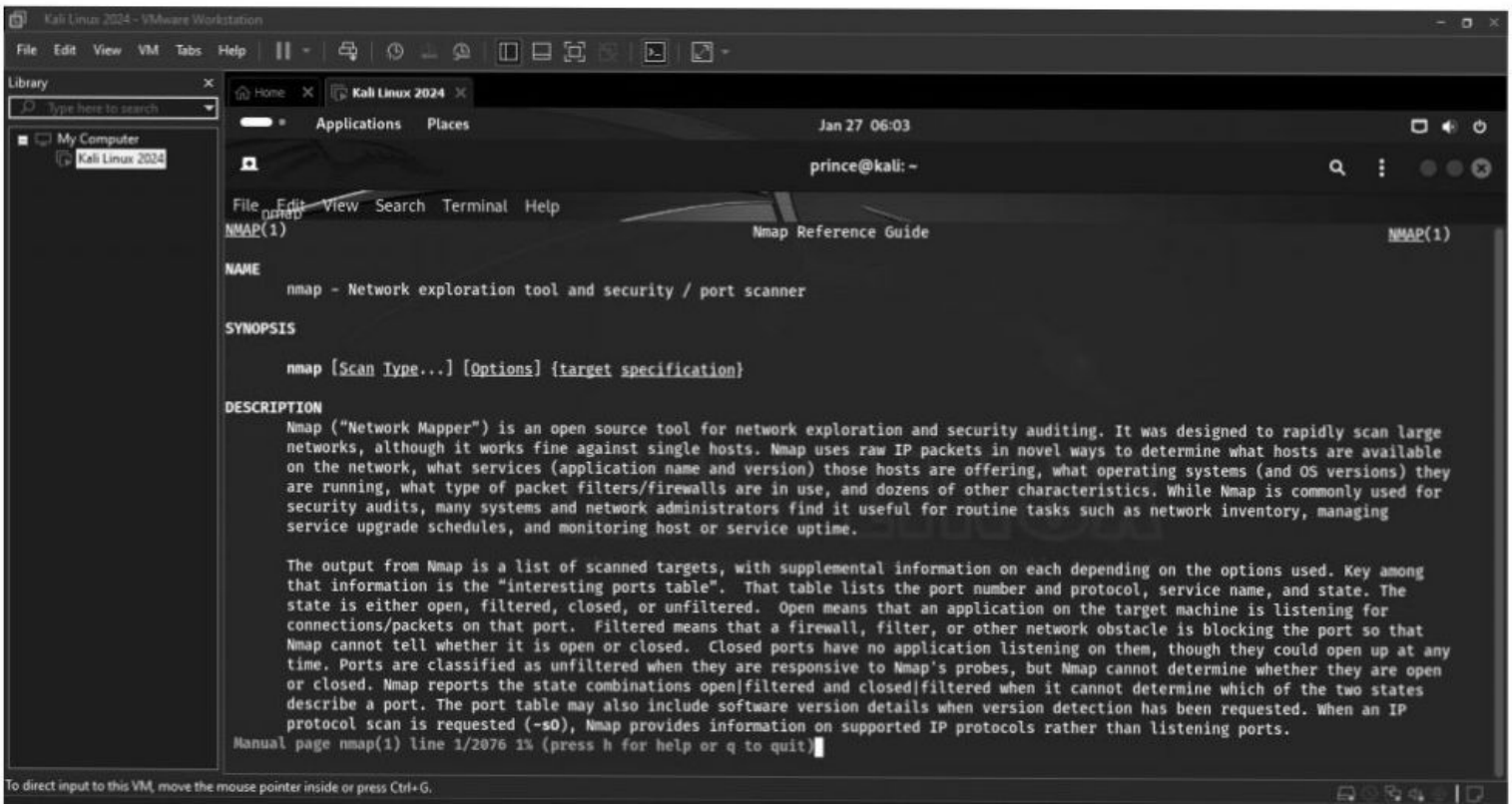
Submitted by

Zahed Hosen

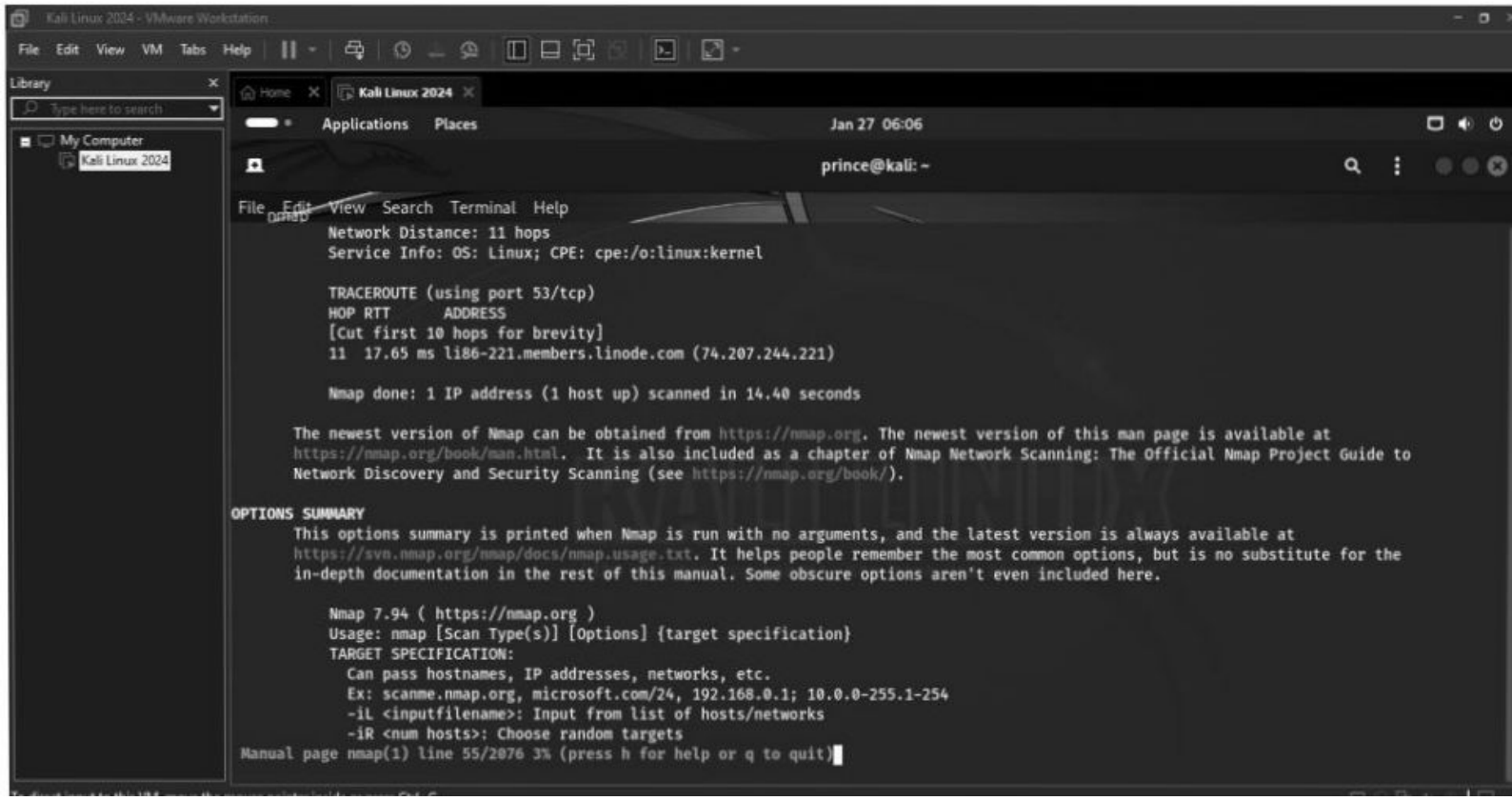
MC223104

OutPut

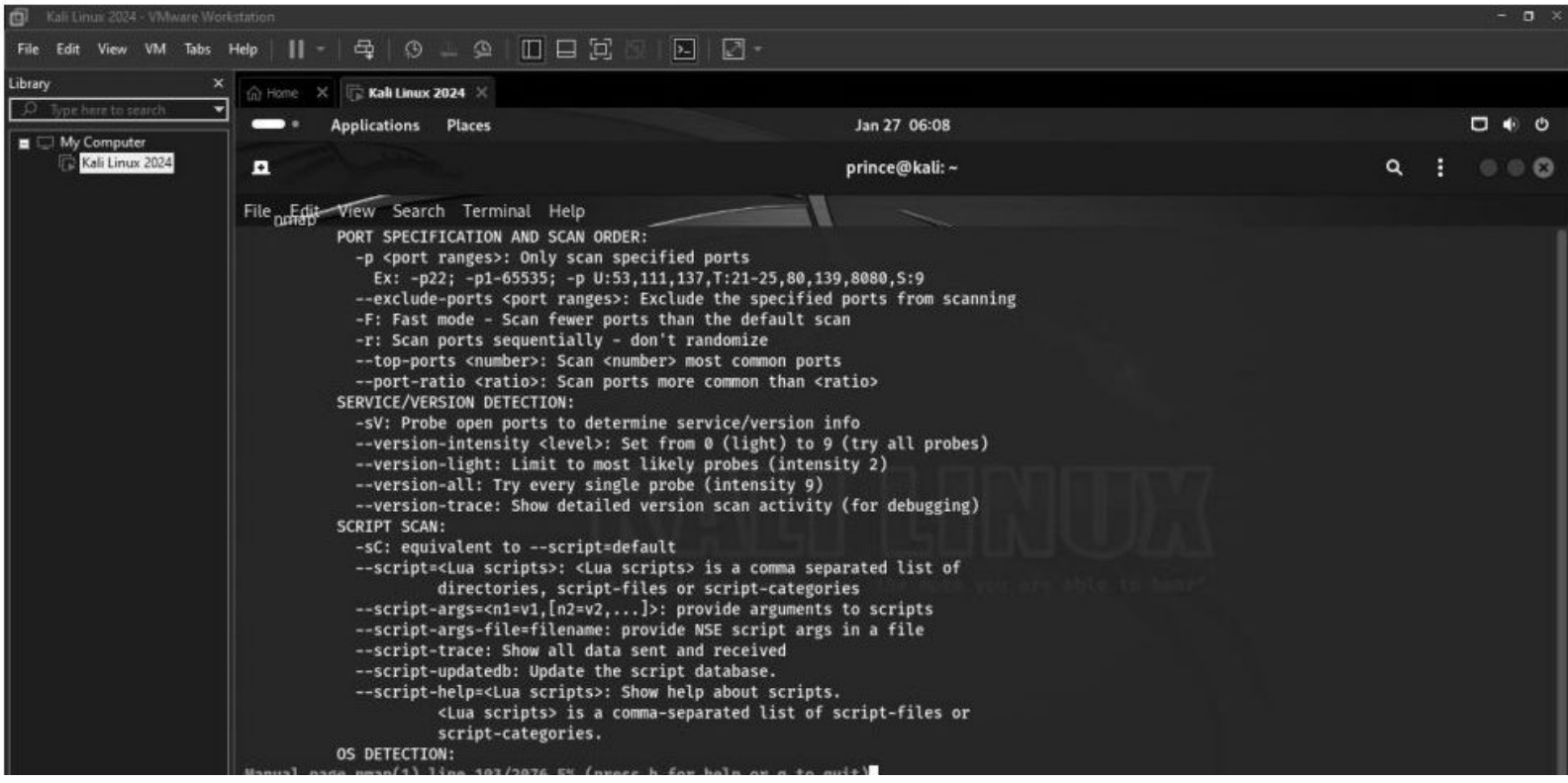

Home
Kali Linux 2024
Applications
Places
Jan 27 05:54
prince@kali: ~
File Edit View Search Terminal Help
nmap
┌──(prince㉿kali)-[~]
└─$ nmap --version
Nmap version 7.94SVN (https://nmap.org)
Platform: x86_64-pc-linux-gnu
Compiled with: liblua-5.4.6 openssl-3.0.11 libssh2-1.11.0 libz-1.2.13 libpcre2-10.42 libpcap-1.10.4 nmap-libdnet-1.12 ipv6
Compiled without:
Available nsock engines: epoll poll select
┌──(prince㉿kali)-[~]
└─$


Kali Linux 2024 - VMware Workstation
File Edit View VM Tabs Help
Library
My Computer
Kali Linux 2024
Home
Kali Linux 2024
Applications
Places
Jan 27 06:03
prince@kali: ~
File Edit View Search Terminal Help
nmap
NMAP(1) Nmap Reference Guide NMAP(1)
NAME
nmap - Network exploration tool and security / port scanner
SYNOPSIS
nmap [Scan Type...] [Options] {target specification}
DESCRIPTION
Nmap ("Network Mapper") is an open source tool for network exploration and security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. While Nmap is commonly used for security audits, many systems and network administrators find it useful for routine tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime.
The output from Nmap is a list of scanned targets, with supplemental information on each depending on the options used. Key among that information is the "interesting ports table". That table lists the port number and protocol, service name, and state. The state is either open, filtered, closed, or unfiltered. Open means that an application on the target machine is listening for connections/packets on that port. Filtered means that a firewall, filter, or other network obstacle is blocking the port so that Nmap cannot tell whether it is open or closed. Closed ports have no application listening on them, though they could open up at any time. Ports are classified as unfiltered when they are responsive to Nmap's probes, but Nmap cannot determine whether they are open or closed. Nmap reports the state combinations open|filtered and closed|filtered when it cannot determine which of the two states describe a port. The port table may also include software version details when version detection has been requested. When an IP protocol scan is requested (-sO), Nmap provides information on supported IP protocols rather than listening ports.
Manual page nmap(1) line 1/2076 1% (press h for help or q to quit)
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

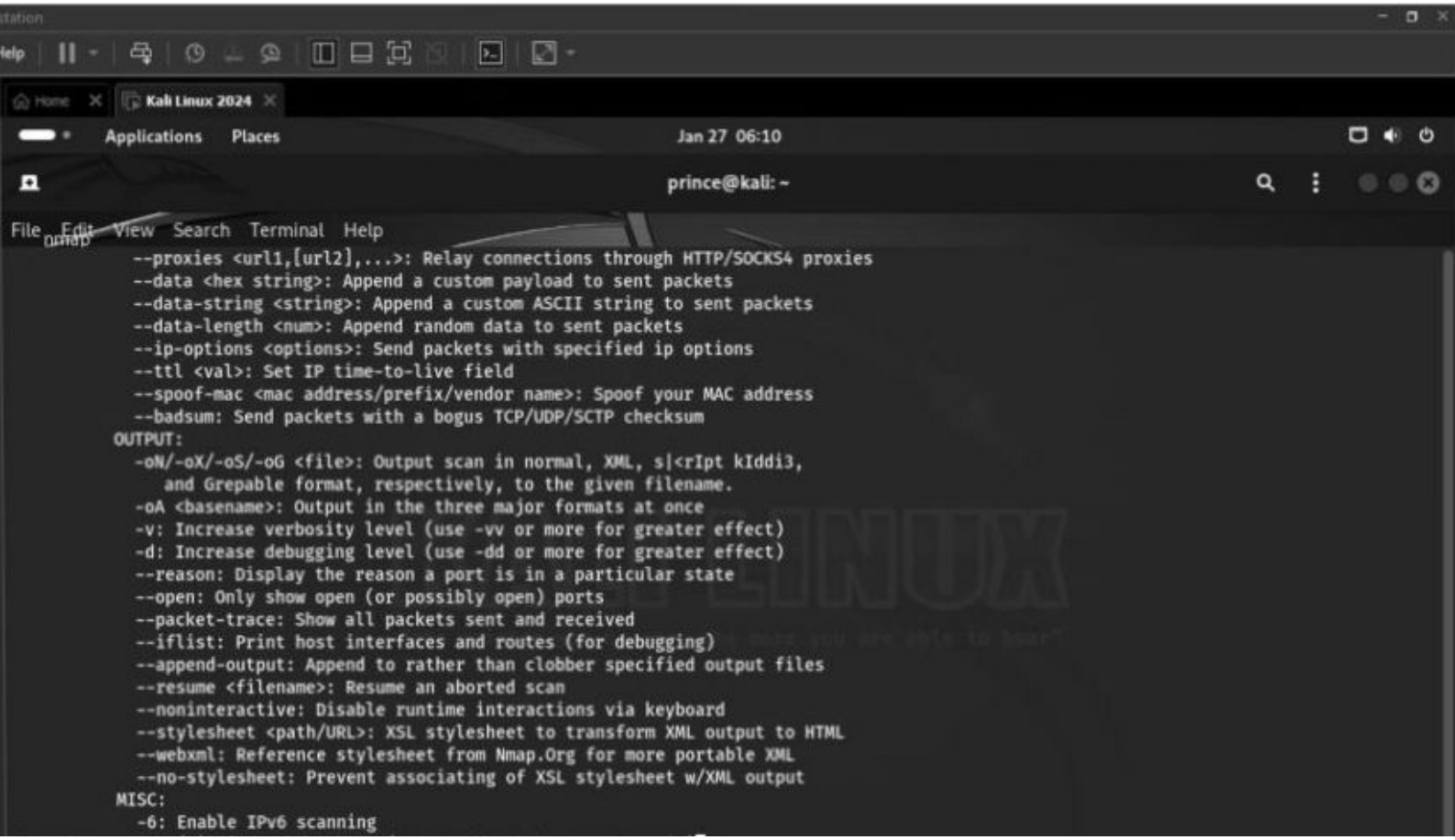

Kali Linux 2024
prince@kali: ~
File Edit View Search Terminal Help
In addition to the interesting ports table, Nmap can provide further information on targets, including reverse DNS names, operating system guesses, device types, and MAC addresses.
A typical Nmap scan is shown in Example 1. The only Nmap arguments used in this example are -A, to enable OS and version detection, script scanning, and traceroute; -T4 for faster execution; and then the hostname.
Example 1. A representative Nmap scan
nmap -A -T4 scanme.nmap.org
Nmap scan report for scanme.nmap.org (74.207.244.221)
Host is up (0.029s latency).
rDNS record for 74.207.244.221: li86-221.members.linode.com
Not shown: 995 closed ports
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 5.3p1 Debian 3ubuntu7 (protocol 2.0)
| ssh-hostkey: 1024 8d:60:f1:7c:ca:b7:3d:0a:d6:67:54:9d:69:d9:b9:dd (DSA)
|_2048 79:f8:09:ac:d4:e2:32:42:10:49:d3:bd:20:82:85:ec (RSA)
80/tcp open http Apache httpd 2.2.14 ((Ubuntu))
|_http-title: Go ahead and ScanMe!
646/tcp filtered ldp
1720/tcp filtered H.323/Q.931
9929/tcp open nping-echo Nping echo
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.39
Manual page nmap(1) line 28/2076 2% (press h for help or q to quit)
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
17°C Haze
Search
8:04 PM 1/27/2024

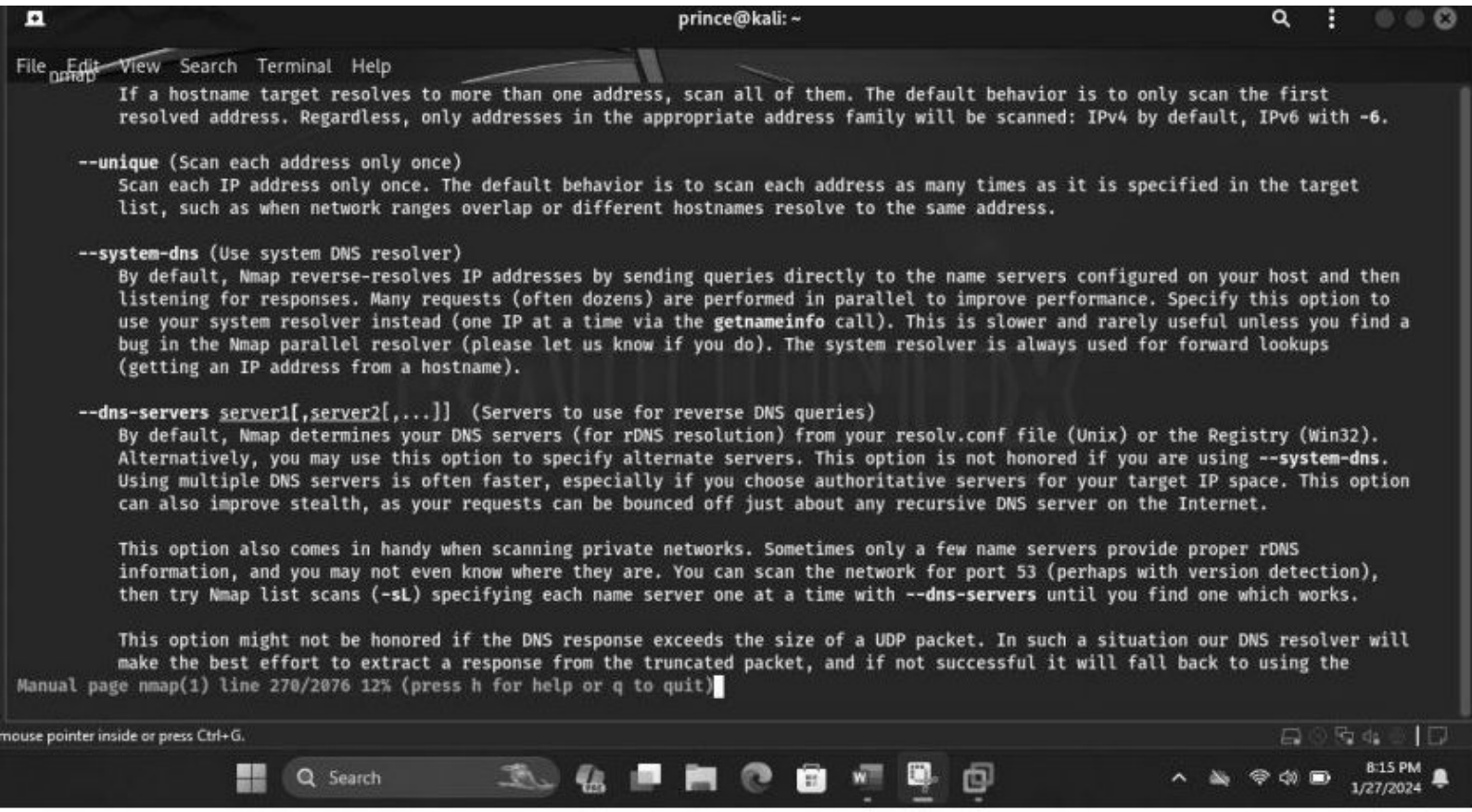

Kali Linux 2024 - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
Kali Linux 2024
Home
Kali Linux 2024
Applications Places
Jan 27 06:06
prince@kali: ~
File Edit View Search Terminal Help
Network Distance: 11 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:kernel
TRACEROUTE (using port 53/tcp)
HOP RTT ADDRESS
[Cut first 10 hops for brevity]
11 17.65 ms li86-221.members.linode.com (74.207.244.221)
Nmap done: 1 IP address (1 host up) scanned in 14.40 seconds
The newest version of Nmap can be obtained from https://nmap.org. The newest version of this man page is available at https://nmap.org/book/man.html. It is also included as a chapter of Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning (see https://nmap.org/book/).
OPTIONS SUMMARY
This options summary is printed when Nmap is run with no arguments, and the latest version is always available at https://svn.nmap.org/nmap/docs/nmap.usage.txt. It helps people remember the most common options, but is no substitute for the in-depth documentation in the rest of this manual. Some obscure options aren't even included here.
Nmap 7.94 (https://nmap.org)
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
Can pass hostnames, IP addresses, networks, etc.
Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
-iL <inputfilename>: Input from list of hosts/networks
-iR <num hosts>: Choose random targets
Manual page nmap(1) line 55/2076 3% (press h for help or q to quit)


File Edit View Search Terminal Help
-iL <inputfilename>: Input from list of hosts/networks
-iR <num hosts>: Choose random targets
--exclude <host1[,host2][,host3],...>: Exclude hosts/networks
--excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
-sL: List Scan - simply list targets to scan
-sn: Ping Scan - disable port scan
-Pn: Treat all hosts as online -- skip host discovery
-PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
-PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
-PO[protocol list]: IP Protocol Ping
-n/-R: Never do DNS resolution/Always resolve [default: sometimes]
--dns-servers <serv1[,serv2],...>: Specify custom DNS servers
--system-dns: Use OS's DNS resolver
--traceroute: Trace hop path to each host
SCAN TECHNIQUES:
-sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
-sU: UDP Scan
-sN/sF/sX: TCP Null, FIN, and Xmas scans
--scanflags <flags>: Customize TCP scan flags
-sI <zombie host[:probeport]>: Idle scan
-sY/sZ: SCTP INIT/COOKIE-ECHO scans
-sO: IP protocol scan
-b <FTP relay host>: FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
-p <port ranges>: Only scan specified ports
Manual page nmap(1) line 79/2076 4% (press h for help or q to quit)
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
17°C Haze
Search
8:07 PM 1/27/2024

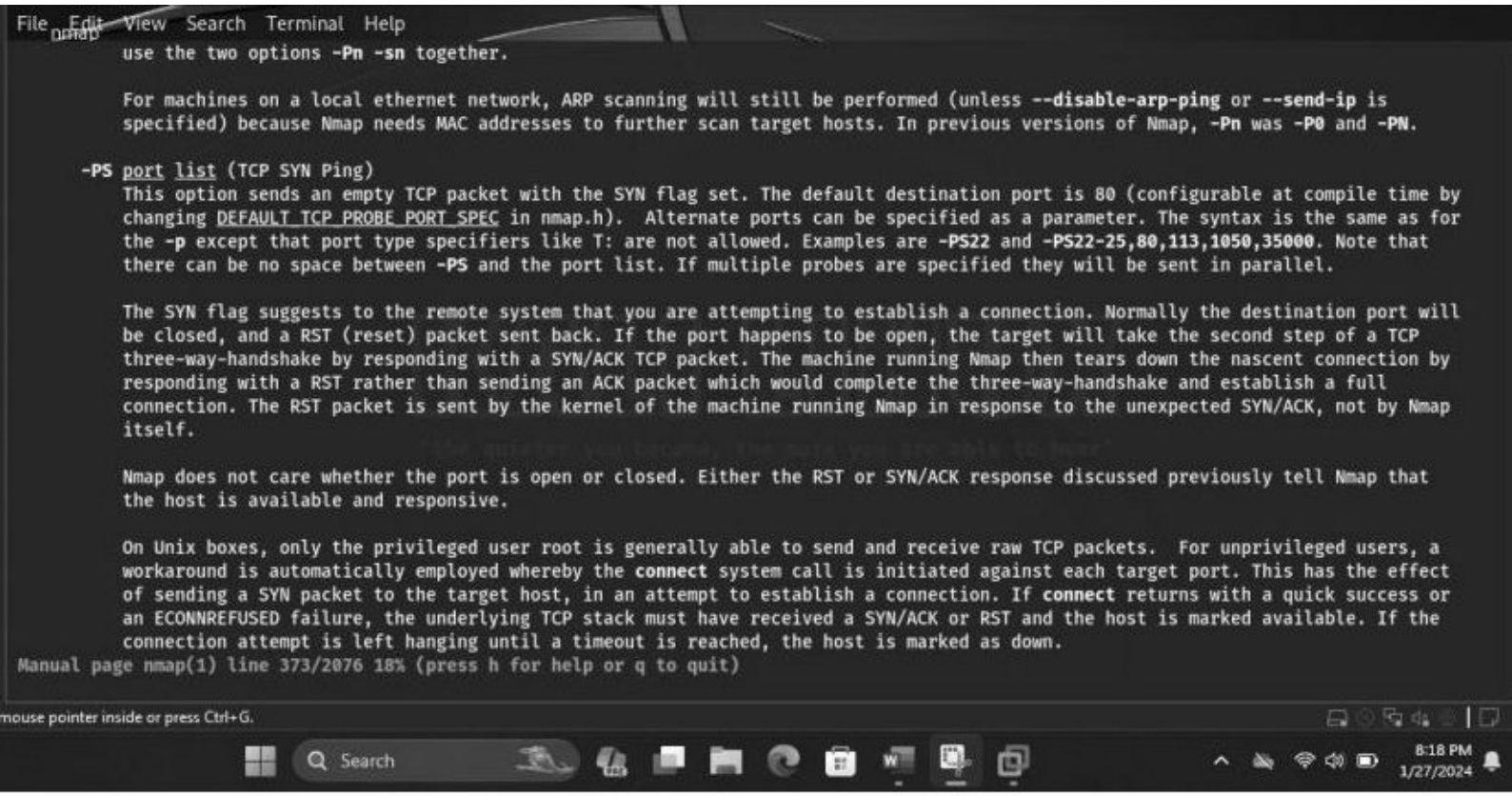

Kali Linux 2024 - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
Kali Linux 2024
Home
Kali Linux 2024
Applications Places
Jan 27 06:08
prince@kali: ~
File Edit View Search Terminal Help
PORT SPECIFICATION AND SCAN ORDER:
-p <port ranges>: Only scan specified ports
Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9
--exclude-ports <port ranges>: Exclude the specified ports from scanning
-F: Fast mode - Scan fewer ports than the default scan
-r: Scan ports sequentially - don't randomize
--top-ports <number>: Scan <number> most common ports
--port-ratio <ratio>: Scan ports more common than <ratio>
SERVICE/VERSION DETECTION:
-sV: Probe open ports to determine service/version info
--version-intensity <level>: Set from 0 (light) to 9 (try all probes)
--version-light: Limit to most likely probes (intensity 2)
--version-all: Try every single probe (intensity 9)
--version-trace: Show detailed version scan activity (for debugging)
SCRIPT SCAN:
-sC: equivalent to --script=default
--script=<Lua scripts>: <Lua scripts> is a comma separated list of
directories, script-files or script-categories
--script-args=<n1=v1,[n2=v2,...]>: provide arguments to scripts
--script-args-file=filename: provide NSE script args in a file
--script-trace: Show all data sent and received
--script-updatedb: Update the script database.
--script-help=<Lua scripts>: Show help about scripts.
<Lua scripts> is a comma-separated list of script-files or
script-categories.
OS DETECTION:

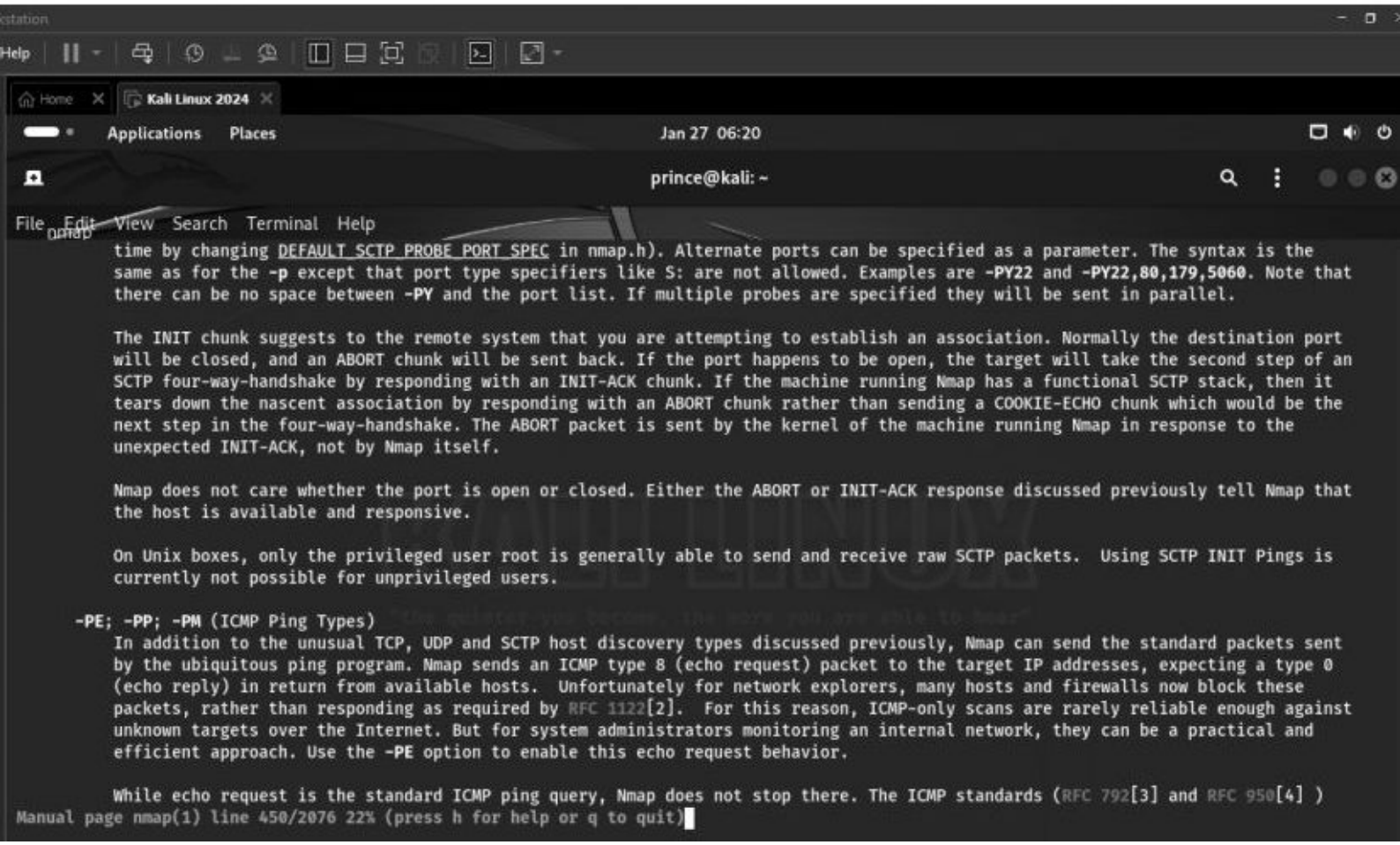

File Edit View Search Terminal Help
nmap
script-categories.
OS DETECTION:
-O: Enable OS detection
--osscan-limit: Limit OS detection to promising targets
--osscan-guess: Guess OS more aggressively
TIMING AND PERFORMANCE:
Options which take <time> are in seconds, or append 'ms' (milliseconds),
's' (seconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).
-T<0-5>: Set timing template (higher is faster)
--min-hostgroup/max-hostgroup <size>: Parallel host scan group sizes
--min-parallelism/max-parallelism <numprobes>: Probe parallelization
--min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time>: Specifies
probe round trip time.
--max-retries <tries>: Caps number of port scan probe retransmissions.
--host-timeout <time>: Give up on target after this long
--scan-delay/--max-scan-delay <time>: Adjust delay between probes
--min-rate <number>: Send packets no slower than <number> per second
--max-rate <number>: Send packets no faster than <number> per second
FIREWALL/IDS EVASION AND SPOOFING:
-f; --mtu <val>: fragment packets (optionally w/given MTU)
-D <decoy1,decoy2[,ME],...>: Cloak a scan with decoys
-S <IP_Address>: Spoof source address
-e <iface>: Use specified interface
-g/--source-port <portnum>: Use given port number
--proxies <url1,[url2],...>: Relay connections through HTTP/SOCKS4 proxies
--data <hex string>: Append a custom payload to sent packets
Manual page nmap(1) line 127/2076 5% (press h for help or q to quit)
ouse pointer inside or press Ctrl+G.
Search
8:09 PM
1/27/2024

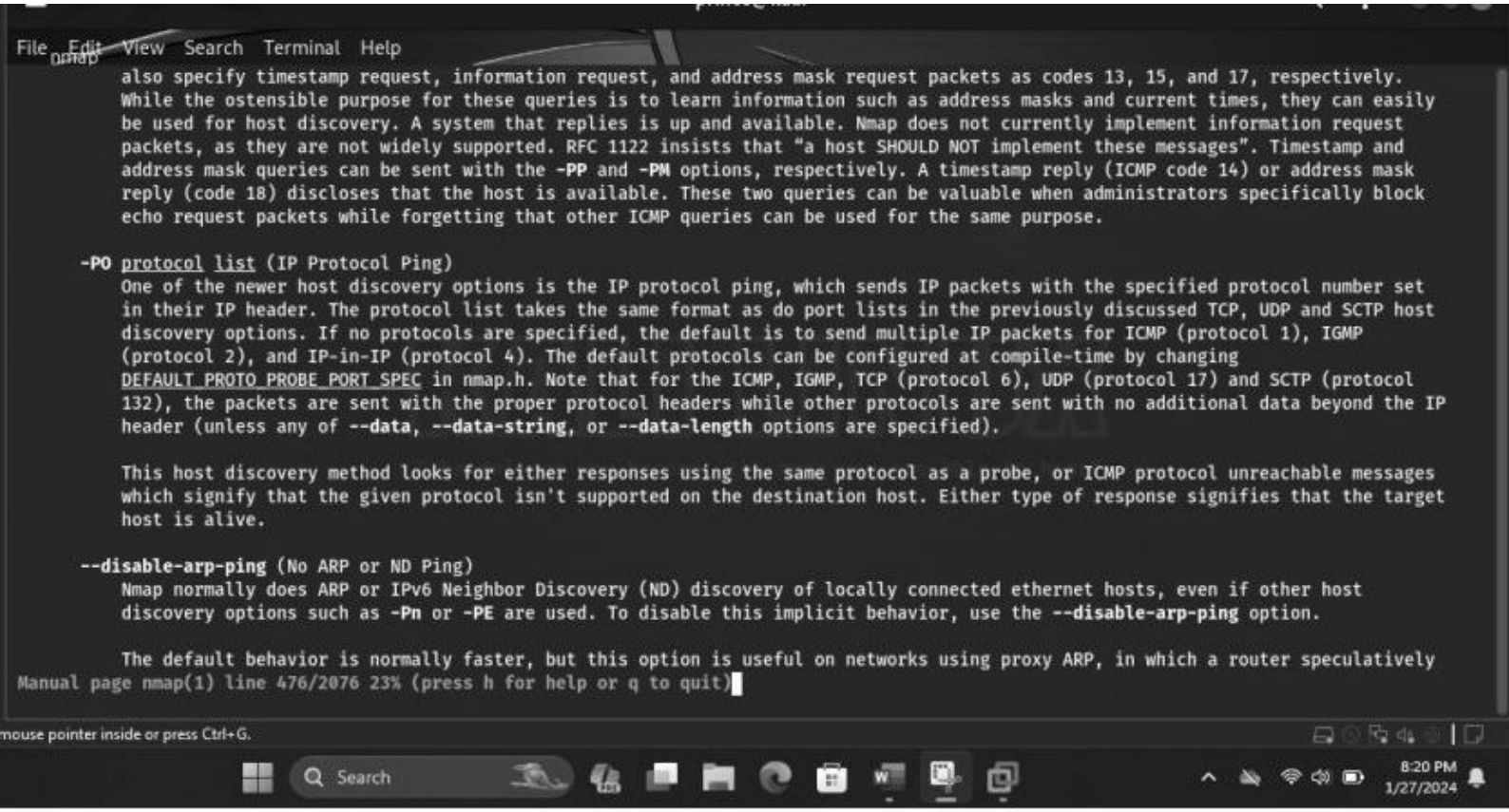

Home
Kali Linux 2024
Applications
Places
Jan 27 06:10
prince@kali: ~
File Edit View Search Terminal Help
nmap
--proxies <url1,[url2],...>: Relay connections through HTTP/SOCKS4 proxies
--data <hex string>: Append a custom payload to sent packets
--data-string <string>: Append a custom ASCII string to sent packets
--data-length <num>: Append random data to sent packets
--ip-options <options>: Send packets with specified ip options
--ttl <val>: Set IP time-to-live field
--spoof-mac <mac address/prefix/vendor name>: Spoof your MAC address
--badsum: Send packets with a bogus TCP/UDP/SCTP checksum
OUTPUT:
-oN/-oX/-oS/-oG <file>: Output scan in normal, XML, s|<rIpt kIddi3,
and Grepable format, respectively, to the given filename.
-oA <basename>: Output in the three major formats at once
-v: Increase verbosity level (use -vv or more for greater effect)
-d: Increase debugging level (use -dd or more for greater effect)
--reason: Display the reason a port is in a particular state
--open: Only show open (or possibly open) ports
--packet-trace: Show all packets sent and received
--iflist: Print host interfaces and routes (for debugging)
--append-output: Append to rather than clobber specified output files
--resume <filename>: Resume an aborted scan
--noninteractive: Disable runtime interactions via keyboard
--stylesheet <path/URL>: XSL stylesheet to transform XML output to HTML
--webxml: Reference stylesheet from Nmap.Org for more portable XML
--no-stylesheet: Prevent associating of XSL stylesheet w/XML output
MISC:
-6: Enable IPv6 scanning

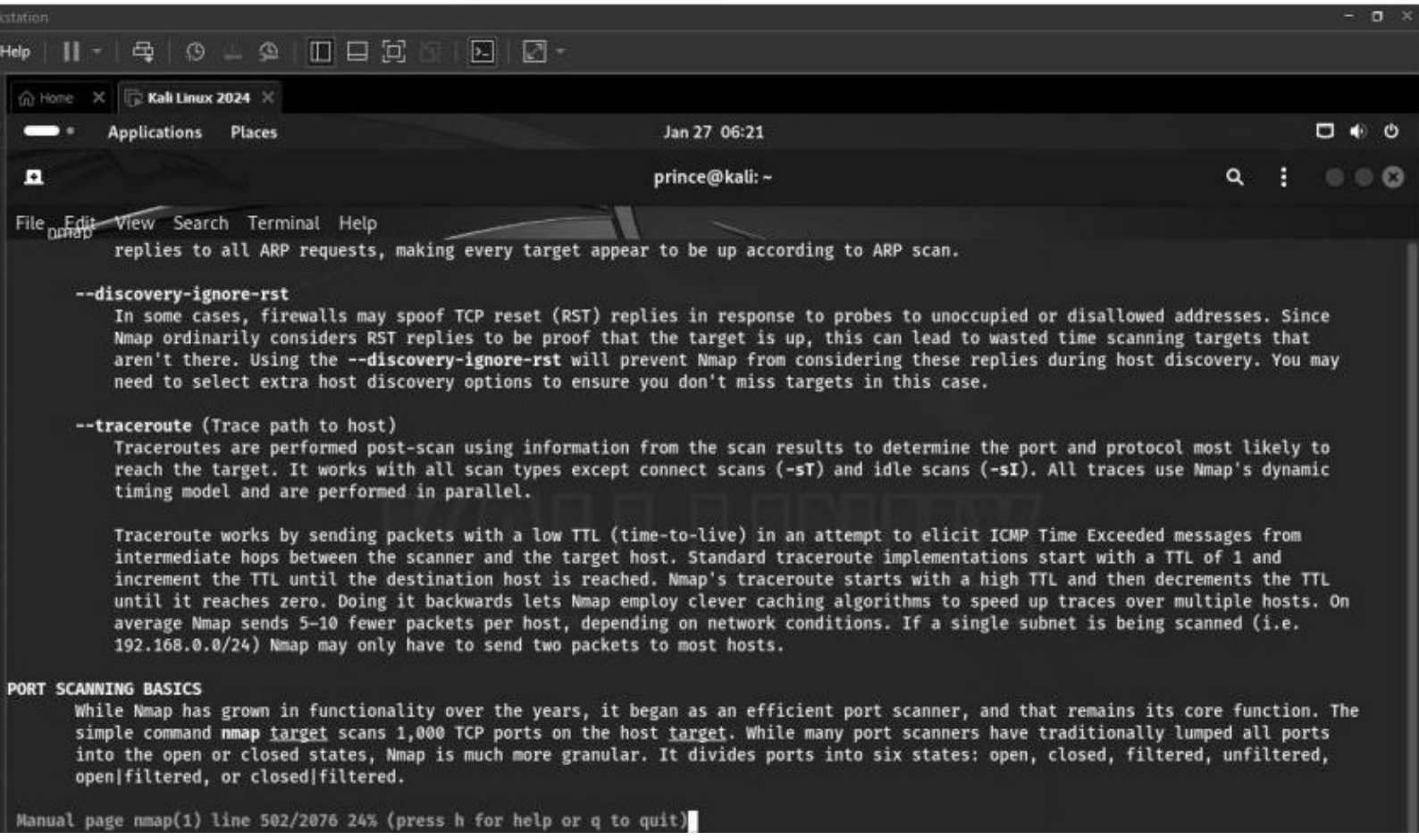

prince@kali: ~
File Edit View Search Terminal Help
nmap
If a hostname target resolves to more than one address, scan all of them. The default behavior is to only scan the first resolved address. Regardless, only addresses in the appropriate address family will be scanned: IPv4 by default, IPv6 with -6.
--unique (Scan each address only once)
Scan each IP address only once. The default behavior is to scan each address as many times as it is specified in the target list, such as when network ranges overlap or different hostnames resolve to the same address.
--system-dns (Use system DNS resolver)
By default, Nmap reverse-resolves IP addresses by sending queries directly to the name servers configured on your host and then listening for responses. Many requests (often dozens) are performed in parallel to improve performance. Specify this option to use your system resolver instead (one IP at a time via the getnameinfo call). This is slower and rarely useful unless you find a bug in the Nmap parallel resolver (please let us know if you do). The system resolver is always used for forward lookups (getting an IP address from a hostname).
--dns-servers server1[,server2[,...]] (Servers to use for reverse DNS queries)
By default, Nmap determines your DNS servers (for rDNS resolution) from your resolv.conf file (Unix) or the Registry (Win32). Alternatively, you may use this option to specify alternate servers. This option is not honored if you are using --system-dns. Using multiple DNS servers is often faster, especially if you choose authoritative servers for your target IP space. This option can also improve stealth, as your requests can be bounced off just about any recursive DNS server on the Internet.
This option also comes in handy when scanning private networks. Sometimes only a few name servers provide proper rDNS information, and you may not even know where they are. You can scan the network for port 53 (perhaps with version detection), then try Nmap list scans (-sL) specifying each name server one at a time with --dns-servers until you find one which works.
This option might not be honored if the DNS response exceeds the size of a UDP packet. In such a situation our DNS resolver will make the best effort to extract a response from the truncated packet, and if not successful it will fall back to using the
Manual page nmap(1) line 270/2076 12% (press h for help or q to quit)
mouse pointer inside or press Ctrl+G.
Search
8:15 PM
1/27/2024

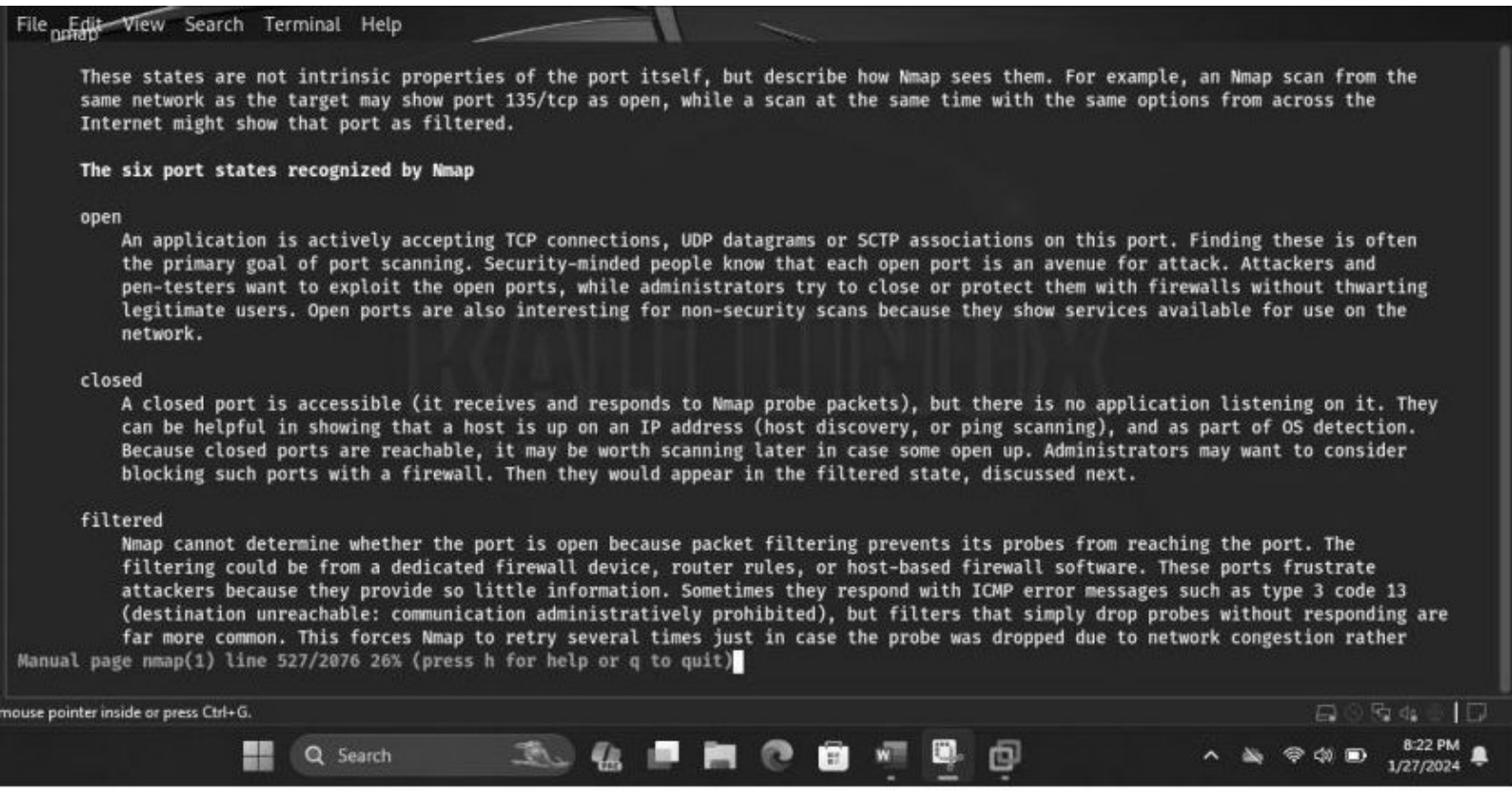

Help
Home
Kali Linux 2024
Applications
Places
Jan 27 06:17
prince@kali: ~
File Edit View Search Terminal Help
nmap
The -P* options (which select ping types) can be combined. You can increase your odds of penetrating strict firewalls by sending many probe types using different TCP ports/flags and ICMP codes. Also note that ARP/Neighbor Discovery is done by default against targets on a local Ethernet network even if you specify other -P* options, because it is almost always faster and more effective.
By default, Nmap does host discovery and then performs a port scan against each host it determines is online. This is true even if you specify non-default host discovery types such as UDP probes (-PU). Read about the -sn option to learn how to perform only host discovery, or use -Pn to skip host discovery and port scan all target addresses. The following options control host discovery:
-sL (List Scan)
The list scan is a degenerate form of host discovery that simply lists each host of the network(s) specified, without sending any packets to the target hosts. By default, Nmap still does reverse-DNS resolution on the hosts to learn their names. It is often surprising how much useful information simple hostnames give out. For example, fw.chi is the name of one company's Chicago firewall.
Nmap also reports the total number of IP addresses at the end. The list scan is a good sanity check to ensure that you have proper IP addresses for your targets. If the hosts sport domain names you do not recognize, it is worth investigating further to prevent scanning the wrong company's network.
Since the idea is to simply print a list of target hosts, options for higher level functionality such as port scanning, OS detection, or host discovery cannot be combined with this. If you wish to disable host discovery while still performing such higher level functionality, read up on the -Pn (skip host discovery) option.
-sn (No port scan)
This option tells Nmap not to do a port scan after host discovery, and only print out the available hosts that responded to the host discovery probes. This is often known as a "ping scan", but you can also request that traceroute and NSE host scripts be
Manual page nmap(1) line 321/2076 15% (press h for help or q to quit)

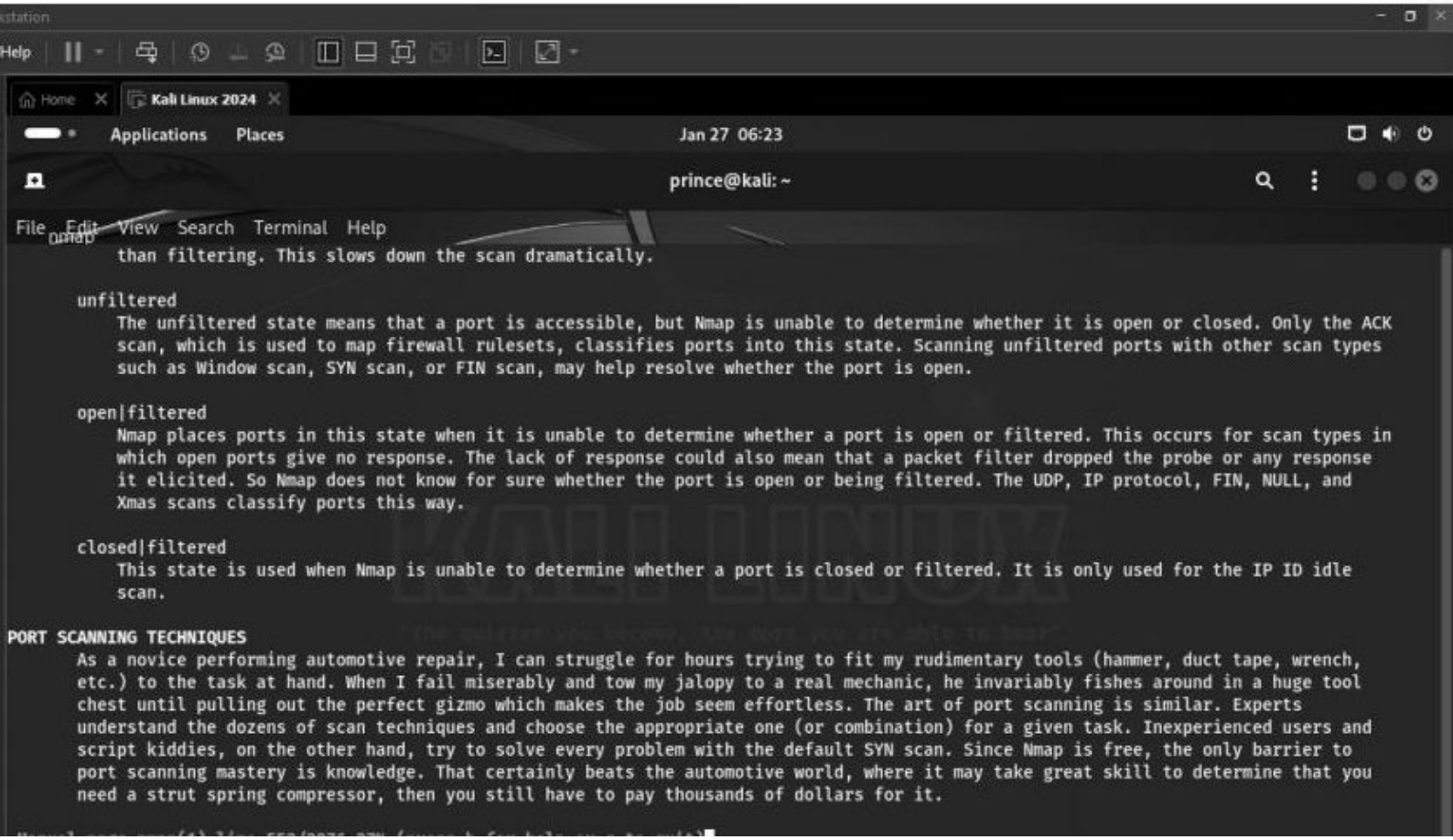

File Edit View Search Terminal Help
use the two options -Pn -sn together.
For machines on a local ethernet network, ARP scanning will still be performed (unless --disable-arp-ping or --send-ip is specified) because Nmap needs MAC addresses to further scan target hosts. In previous versions of Nmap, -Pn was -P0 and -PN.
-PS port list (TCP SYN Ping)
This option sends an empty TCP packet with the SYN flag set. The default destination port is 80 (configurable at compile time by changing DEFAULT_TCP_PROBE_PORT_SPEC in nmap.h). Alternate ports can be specified as a parameter. The syntax is the same as for the -p except that port type specifiers like T: are not allowed. Examples are -PS22 and -PS22-25,80,113,1050,35000. Note that there can be no space between -PS and the port list. If multiple probes are specified they will be sent in parallel.
The SYN flag suggests to the remote system that you are attempting to establish a connection. Normally the destination port will be closed, and a RST (reset) packet sent back. If the port happens to be open, the target will take the second step of a TCP three-way-handshake by responding with a SYN/ACK TCP packet. The machine running Nmap then tears down the nascent connection by responding with a RST rather than sending an ACK packet which would complete the three-way-handshake and establish a full connection. The RST packet is sent by the kernel of the machine running Nmap in response to the unexpected SYN/ACK, not by Nmap itself.
Nmap does not care whether the port is open or closed. Either the RST or SYN/ACK response discussed previously tell Nmap that the host is available and responsive.
On Unix boxes, only the privileged user root is generally able to send and receive raw TCP packets. For unprivileged users, a workaround is automatically employed whereby the connect system call is initiated against each target port. This has the effect of sending a SYN packet to the target host, in an attempt to establish a connection. If connect returns with a quick success or an ECONNREFUSED failure, the underlying TCP stack must have received a SYN/ACK or RST and the host is marked available. If the connection attempt is left hanging until a timeout is reached, the host is marked as down.
Manual page nmap(1) line 373/2076 18% (press h for help or q to quit)
mouse pointer inside or press Ctrl+G.
Search
8:18 PM
1/27/2024

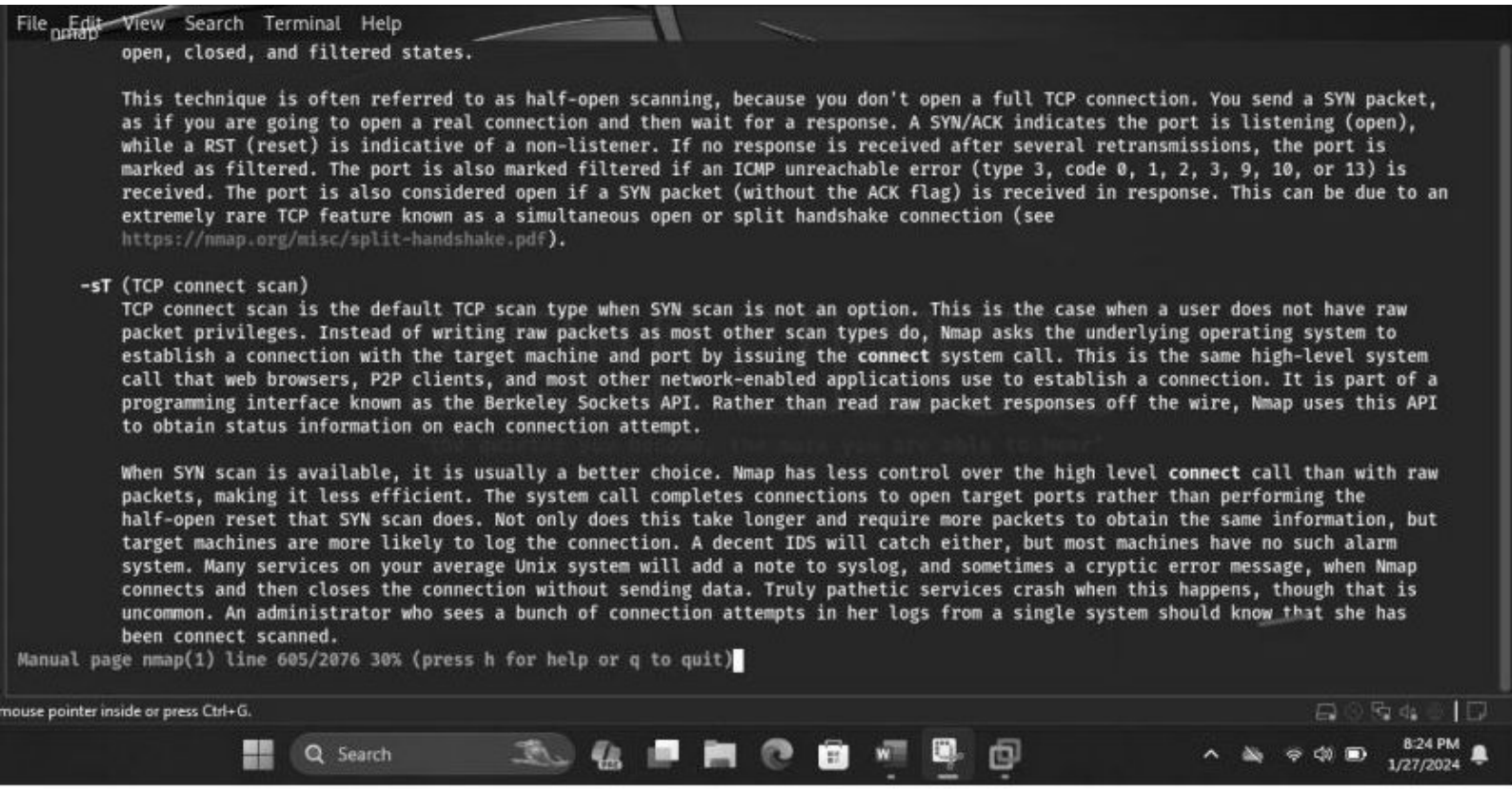

Help
Home
Kali Linux 2024
Applications
Places
Jan 27 06:20
prince@kali: ~
File Edit View Search Terminal Help
time by changing DEFAULT_SCTP_PROBE_PORT_SPEC in nmap.h). Alternate ports can be specified as a parameter. The syntax is the same as for the -p except that port type specifiers like S: are not allowed. Examples are -PY22 and -PY22,80,179,5060. Note that there can be no space between -PY and the port list. If multiple probes are specified they will be sent in parallel.
The INIT chunk suggests to the remote system that you are attempting to establish an association. Normally the destination port will be closed, and an ABORT chunk will be sent back. If the port happens to be open, the target will take the second step of an SCTP four-way-handshake by responding with an INIT-ACK chunk. If the machine running Nmap has a functional SCTP stack, then it tears down the nascent association by responding with an ABORT chunk rather than sending a COOKIE-ECHO chunk which would be the next step in the four-way-handshake. The ABORT packet is sent by the kernel of the machine running Nmap in response to the unexpected INIT-ACK, not by Nmap itself.
Nmap does not care whether the port is open or closed. Either the ABORT or INIT-ACK response discussed previously tell Nmap that the host is available and responsive.
On Unix boxes, only the privileged user root is generally able to send and receive raw SCTP packets. Using SCTP INIT Pings is currently not possible for unprivileged users.
-PE; -PP; -PM (ICMP Ping Types)
In addition to the unusual TCP, UDP and SCTP host discovery types discussed previously, Nmap can send the standard packets sent by the ubiquitous ping program. Nmap sends an ICMP type 8 (echo request) packet to the target IP addresses, expecting a type 0 (echo reply) in return from available hosts. Unfortunately for network explorers, many hosts and firewalls now block these packets, rather than responding as required by RFC 1122[2]. For this reason, ICMP-only scans are rarely reliable enough against unknown targets over the Internet. But for system administrators monitoring an internal network, they can be a practical and efficient approach. Use the -PE option to enable this echo request behavior.
While echo request is the standard ICMP ping query, Nmap does not stop there. The ICMP standards (RFC 792[3] and RFC 950[4])
Manual page nmap(1) line 450/2076 22% (press h for help or q to quit)

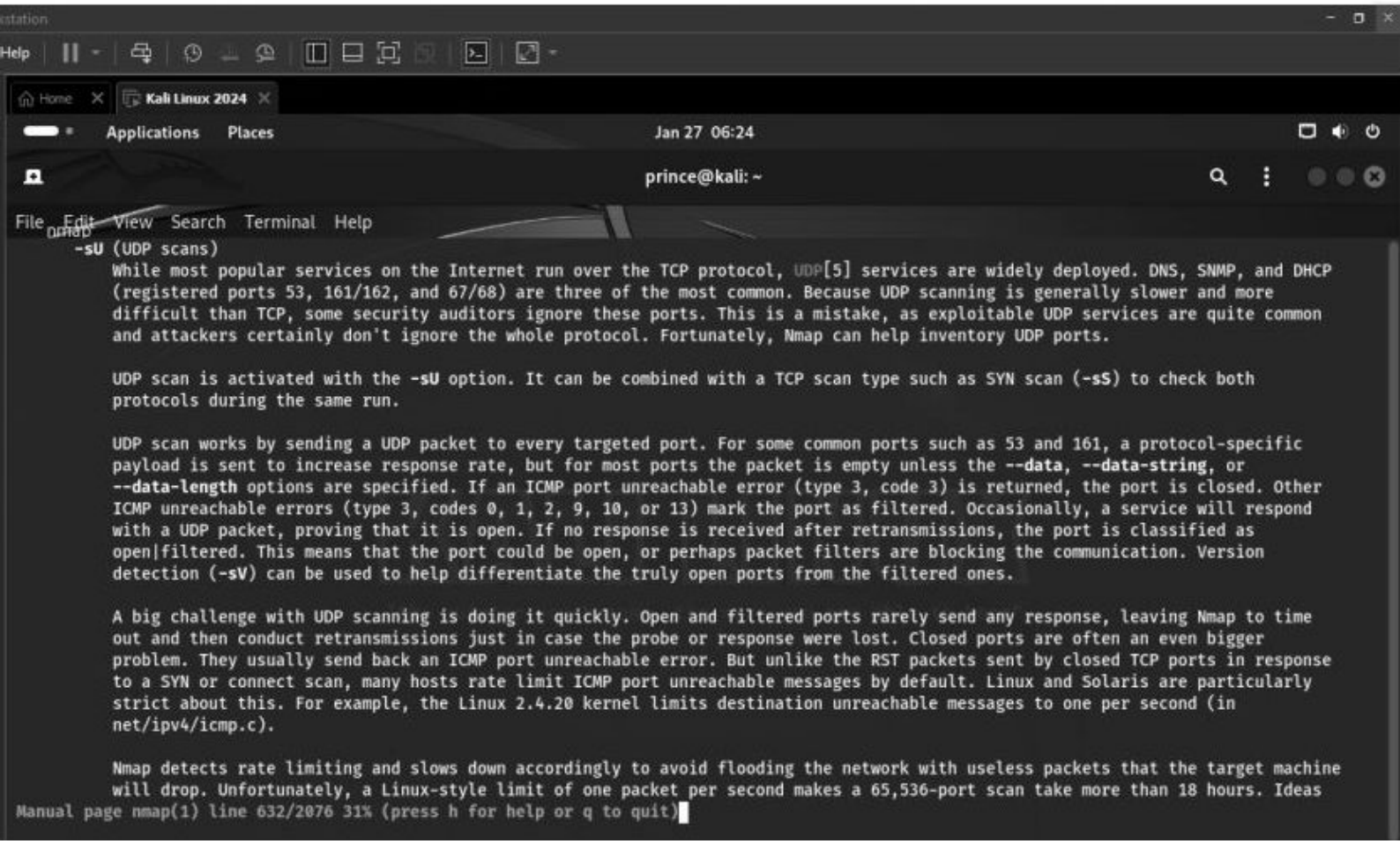

File Edit View Search Terminal Help
also specify timestamp request, information request, and address mask request packets as codes 13, 15, and 17, respectively. While the ostensible purpose for these queries is to learn information such as address masks and current times, they can easily be used for host discovery. A system that replies is up and available. Nmap does not currently implement information request packets, as they are not widely supported. RFC 1122 insists that "a host SHOULD NOT implement these messages". Timestamp and address mask queries can be sent with the -PP and -PM options, respectively. A timestamp reply (ICMP code 14) or address mask reply (code 18) discloses that the host is available. These two queries can be valuable when administrators specifically block echo request packets while forgetting that other ICMP queries can be used for the same purpose.
-PO protocol list (IP Protocol Ping)
One of the newer host discovery options is the IP protocol ping, which sends IP packets with the specified protocol number set in their IP header. The protocol list takes the same format as do port lists in the previously discussed TCP, UDP and SCTP host discovery options. If no protocols are specified, the default is to send multiple IP packets for ICMP (protocol 1), IGMP (protocol 2), and IP-in-IP (protocol 4). The default protocols can be configured at compile-time by changing DEFAULT_PROTO_PROBE_PORT_SPEC in nmap.h. Note that for the ICMP, IGMP, TCP (protocol 6), UDP (protocol 17) and SCTP (protocol 132), the packets are sent with the proper protocol headers while other protocols are sent with no additional data beyond the IP header (unless any of --data, --data-string, or --data-length options are specified).
This host discovery method looks for either responses using the same protocol as a probe, or ICMP protocol unreachable messages which signify that the given protocol isn't supported on the destination host. Either type of response signifies that the target host is alive.
--disable-arp-ping (No ARP or ND Ping)
Nmap normally does ARP or IPv6 Neighbor Discovery (ND) discovery of locally connected ethernet hosts, even if other host discovery options such as -Pn or -PE are used. To disable this implicit behavior, use the --disable-arp-ping option.
The default behavior is normally faster, but this option is useful on networks using proxy ARP, in which a router speculatively
Manual page nmap(1) line 476/2076 23% (press h for help or q to quit)
mouse pointer inside or press Ctrl+G.
Search
8:20 PM
1/27/2024

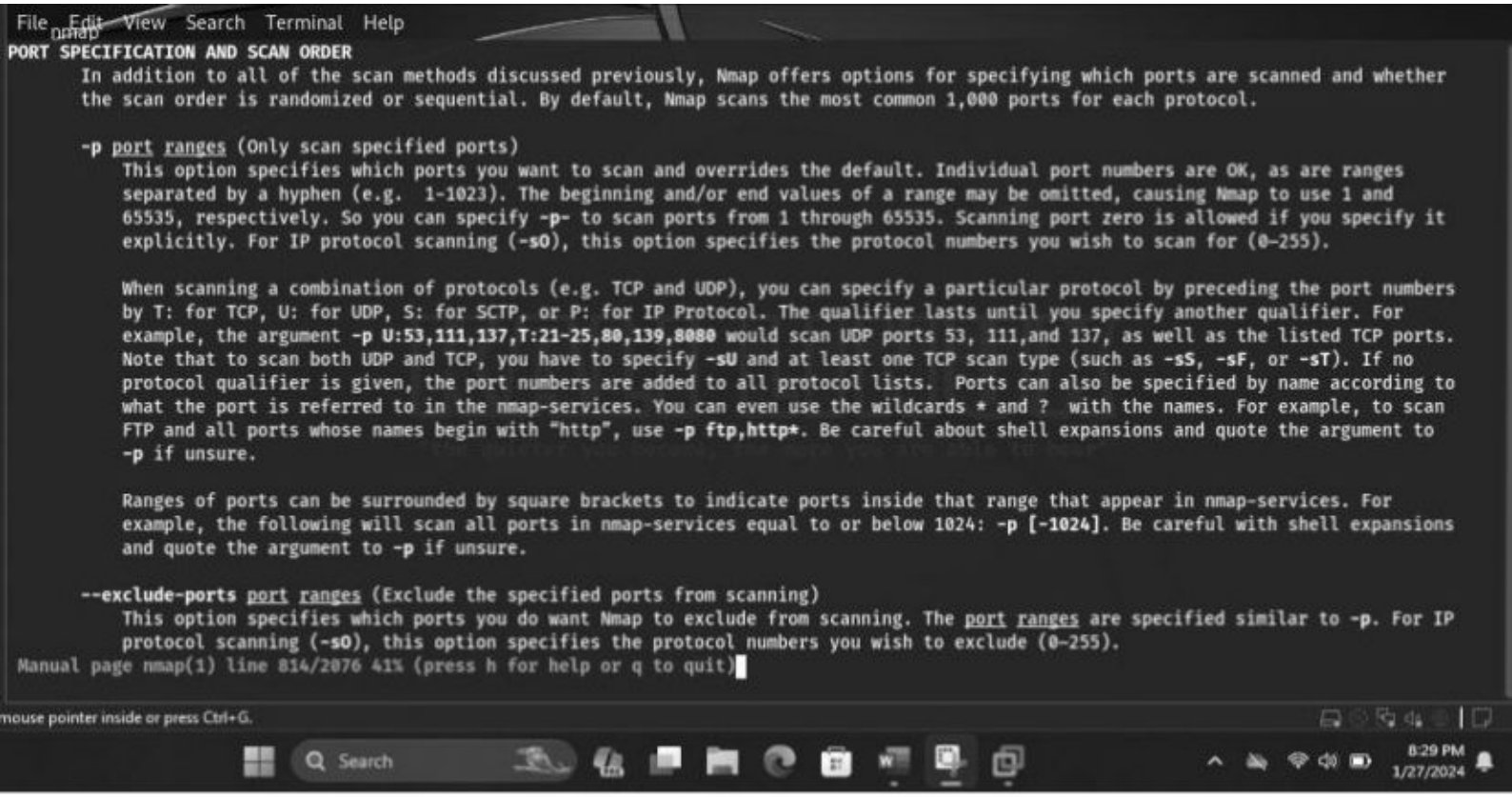

Help
Home
Kali Linux 2024
Applications
Places
Jan 27 06:21
prince@kali: ~
File Edit View Search Terminal Help
replies to all ARP requests, making every target appear to be up according to ARP scan.
--discovery-ignore-rst
In some cases, firewalls may spoof TCP reset (RST) replies in response to probes to unoccupied or disallowed addresses. Since Nmap ordinarily considers RST replies to be proof that the target is up, this can lead to wasted time scanning targets that aren't there. Using the --discovery-ignore-rst will prevent Nmap from considering these replies during host discovery. You may need to select extra host discovery options to ensure you don't miss targets in this case.
--traceroute (Trace path to host)
Traceroutes are performed post-scan using information from the scan results to determine the port and protocol most likely to reach the target. It works with all scan types except connect scans (-sT) and idle scans (-sI). All traces use Nmap's dynamic timing model and are performed in parallel.
Traceroute works by sending packets with a low TTL (time-to-live) in an attempt to elicit ICMP Time Exceeded messages from intermediate hops between the scanner and the target host. Standard traceroute implementations start with a TTL of 1 and increment the TTL until the destination host is reached. Nmap's traceroute starts with a high TTL and then decrements the TTL until it reaches zero. Doing it backwards lets Nmap employ clever caching algorithms to speed up traces over multiple hosts. On average Nmap sends 5–10 fewer packets per host, depending on network conditions. If a single subnet is being scanned (i.e. 192.168.0.0/24) Nmap may only have to send two packets to most hosts.
PORT SCANNING BASICS
While Nmap has grown in functionality over the years, it began as an efficient port scanner, and that remains its core function. The simple command nmap target scans 1,000 TCP ports on the host target. While many port scanners have traditionally lumped all ports into the open or closed states, Nmap is much more granular. It divides ports into six states: open, closed, filtered, unfiltered, open|filtered, or closed|filtered.
Manual page nmap(1) line 502/2076 24% (press h for help or q to quit)

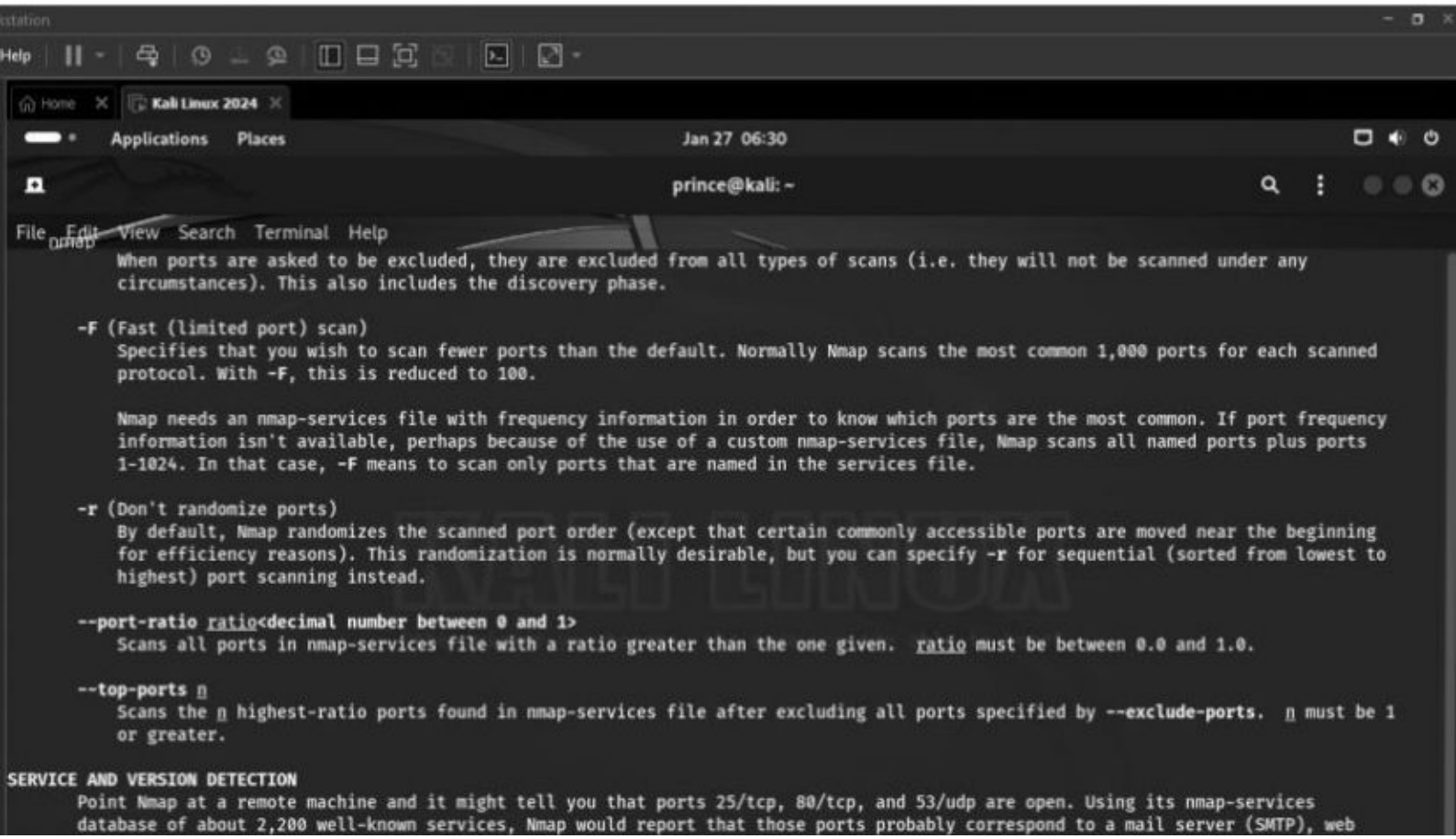

File Edit View Search Terminal Help
These states are not intrinsic properties of the port itself, but describe how Nmap sees them. For example, an Nmap scan from the same network as the target may show port 135/tcp as open, while a scan at the same time with the same options from across the Internet might show that port as filtered.
The six port states recognized by Nmap
open
An application is actively accepting TCP connections, UDP datagrams or SCTP associations on this port. Finding these is often the primary goal of port scanning. Security-minded people know that each open port is an avenue for attack. Attackers and pen-testers want to exploit the open ports, while administrators try to close or protect them with firewalls without thwarting legitimate users. Open ports are also interesting for non-security scans because they show services available for use on the network.
closed
A closed port is accessible (it receives and responds to Nmap probe packets), but there is no application listening on it. They can be helpful in showing that a host is up on an IP address (host discovery, or ping scanning), and as part of OS detection. Because closed ports are reachable, it may be worth scanning later in case some open up. Administrators may want to consider blocking such ports with a firewall. Then they would appear in the filtered state, discussed next.
filtered
Nmap cannot determine whether the port is open because packet filtering prevents its probes from reaching the port. The filtering could be from a dedicated firewall device, router rules, or host-based firewall software. These ports frustrate attackers because they provide so little information. Sometimes they respond with ICMP error messages such as type 3 code 13 (destination unreachable: communication administratively prohibited), but filters that simply drop probes without responding are far more common. This forces Nmap to retry several times just in case the probe was dropped due to network congestion rather
Manual page nmap(1) line 527/2076 26% (press h for help or q to quit)
mouse pointer inside or press Ctrl+G.
Search
8:22 PM
1/27/2024

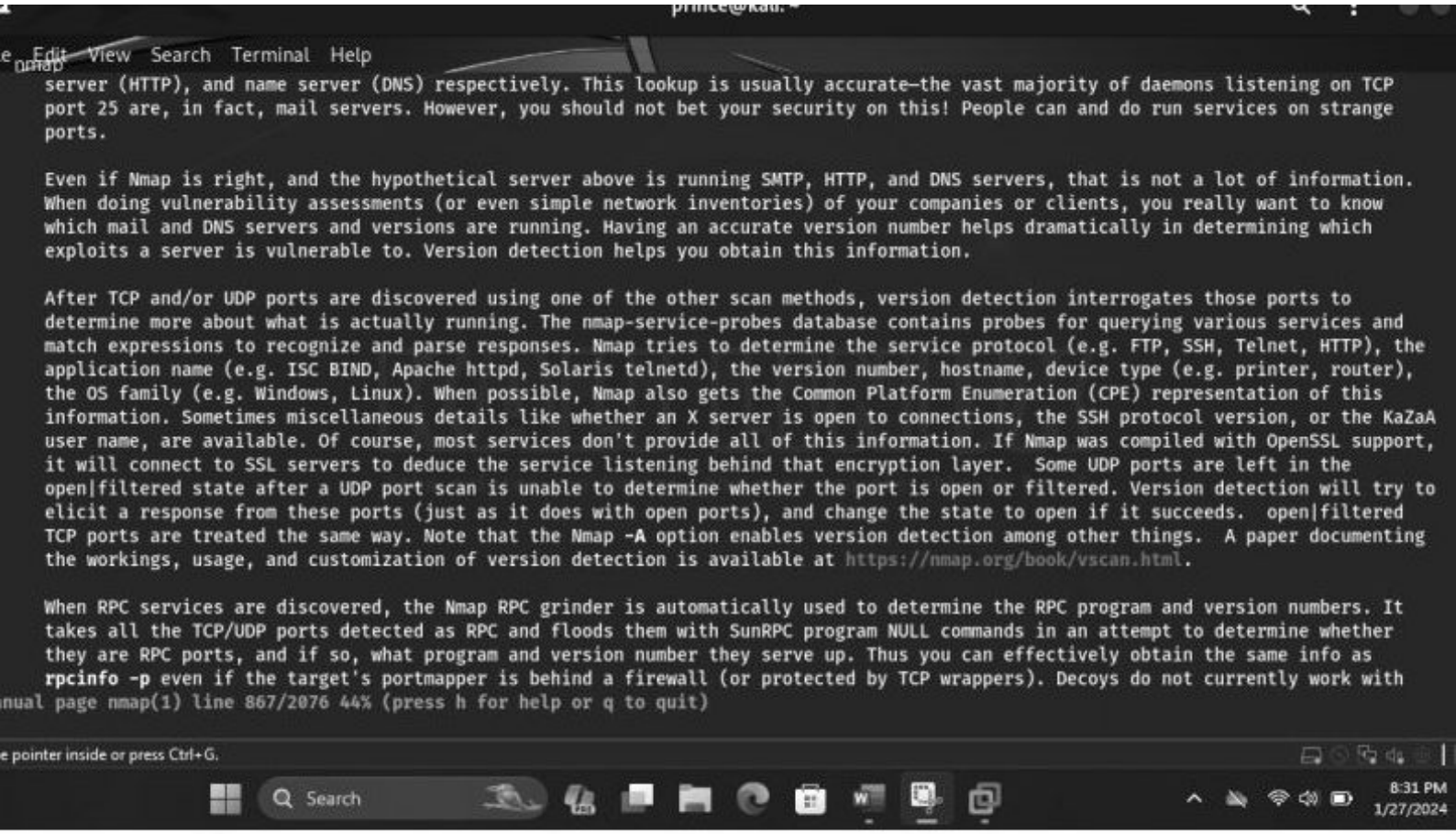

Help
Home
Kali Linux 2024
Applications
Places
Jan 27 06:23
prince@kali: ~
File Edit View Search Terminal Help
than filtering. This slows down the scan dramatically.
unfiltered
The unfiltered state means that a port is accessible, but Nmap is unable to determine whether it is open or closed. Only the ACK scan, which is used to map firewall rulesets, classifies ports into this state. Scanning unfiltered ports with other scan types such as Window scan, SYN scan, or FIN scan, may help resolve whether the port is open.
open|filtered
Nmap places ports in this state when it is unable to determine whether a port is open or filtered. This occurs for scan types in which open ports give no response. The lack of response could also mean that a packet filter dropped the probe or any response it elicited. So Nmap does not know for sure whether the port is open or being filtered. The UDP, IP protocol, FIN, NULL, and Xmas scans classify ports this way.
closed|filtered
This state is used when Nmap is unable to determine whether a port is closed or filtered. It is only used for the IP ID idle scan.
PORT SCANNING TECHNIQUES
As a novice performing automotive repair, I can struggle for hours trying to fit my rudimentary tools (hammer, duct tape, wrench, etc.) to the task at hand. When I fail miserably and tow my jalopy to a real mechanic, he invariably fishes around in a huge tool chest until pulling out the perfect gizmo which makes the job seem effortless. The art of port scanning is similar. Experts understand the dozens of scan techniques and choose the appropriate one (or combination) for a given task. Inexperienced users and script kiddies, on the other hand, try to solve every problem with the default SYN scan. Since Nmap is free, the only barrier to port scanning mastery is knowledge. That certainly beats the automotive world, where it may take great skill to determine that you need a strut spring compressor, then you still have to pay thousands of dollars for it.

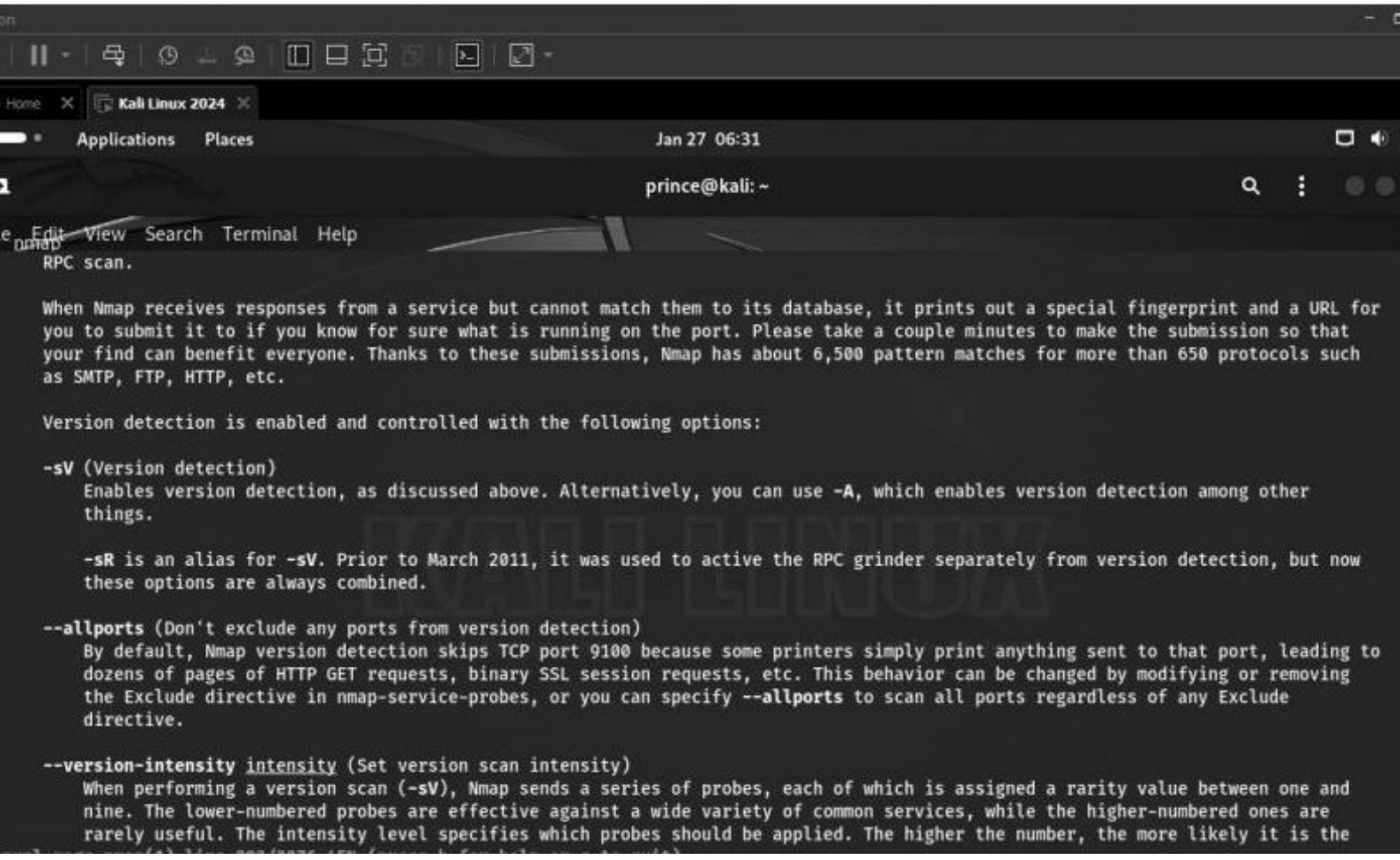

File Edit View Search Terminal Help
nmap
open, closed, and filtered states.
This technique is often referred to as half-open scanning, because you don't open a full TCP connection. You send a SYN packet, as if you are going to open a real connection and then wait for a response. A SYN/ACK indicates the port is listening (open), while a RST (reset) is indicative of a non-listener. If no response is received after several retransmissions, the port is marked as filtered. The port is also marked filtered if an ICMP unreachable error (type 3, code 0, 1, 2, 3, 9, 10, or 13) is received. The port is also considered open if a SYN packet (without the ACK flag) is received in response. This can be due to an extremely rare TCP feature known as a simultaneous open or split handshake connection (see https://nmap.org/misc/split-handshake.pdf).
-sT (TCP connect scan)
TCP connect scan is the default TCP scan type when SYN scan is not an option. This is the case when a user does not have raw packet privileges. Instead of writing raw packets as most other scan types do, Nmap asks the underlying operating system to establish a connection with the target machine and port by issuing the connect system call. This is the same high-level system call that web browsers, P2P clients, and most other network-enabled applications use to establish a connection. It is part of a programming interface known as the Berkeley Sockets API. Rather than read raw packet responses off the wire, Nmap uses this API to obtain status information on each connection attempt.
When SYN scan is available, it is usually a better choice. Nmap has less control over the high level connect call than with raw packets, making it less efficient. The system call completes connections to open target ports rather than performing the half-open reset that SYN scan does. Not only does this take longer and require more packets to obtain the same information, but target machines are more likely to log the connection. A decent IDS will catch either, but most machines have no such alarm system. Many services on your average Unix system will add a note to syslog, and sometimes a cryptic error message, when Nmap connects and then closes the connection without sending data. Truly pathetic services crash when this happens, though that is uncommon. An administrator who sees a bunch of connection attempts in her logs from a single system should know that she has been connect scanned.
Manual page nmap(1) line 605/2076 30% (press h for help or q to quit)
mouse pointer inside or press Ctrl+G.
Search
8:24 PM
1/27/2024

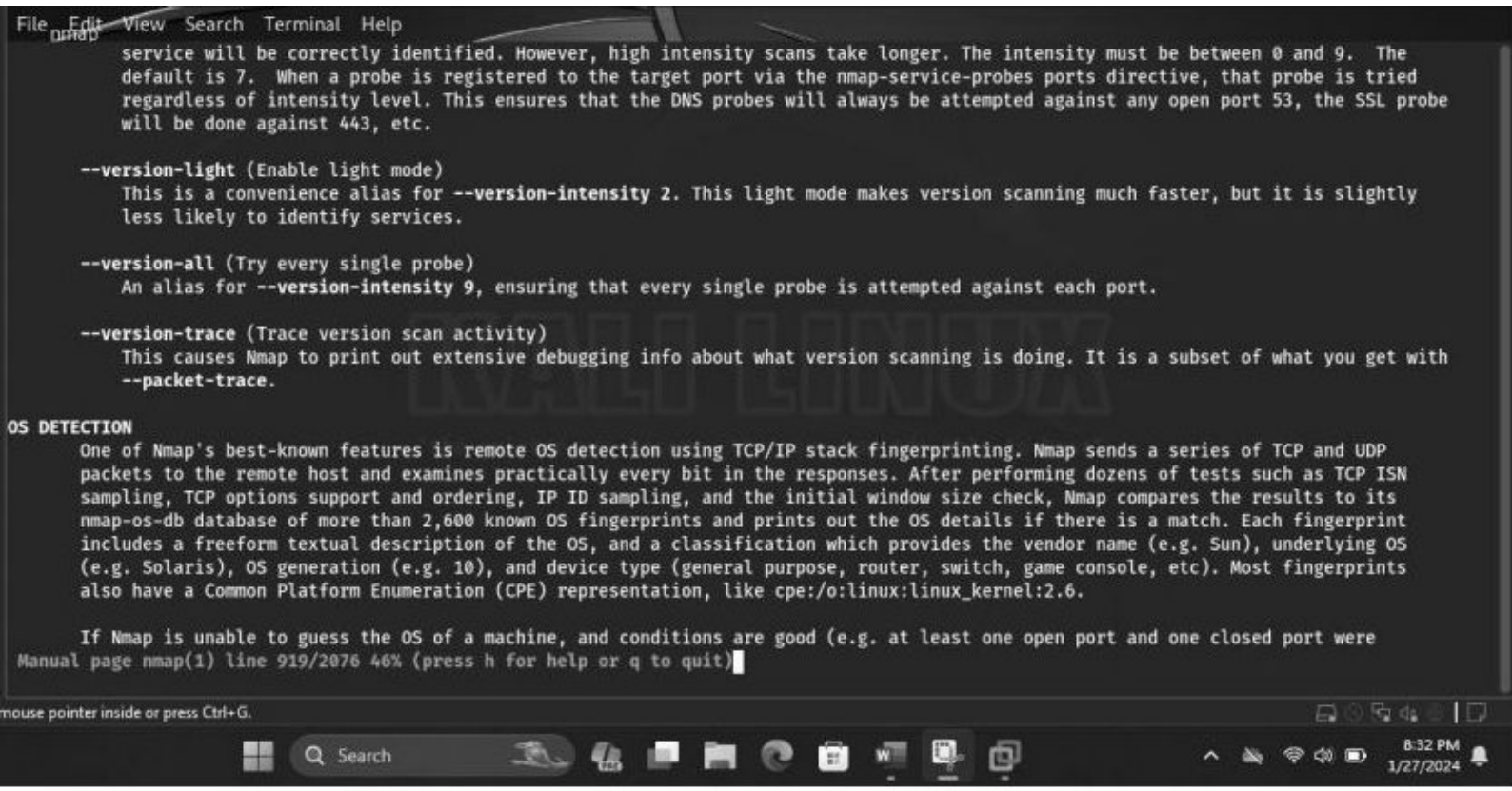

Help
Home
Kali Linux 2024
Applications
Places
Jan 27 06:24
prince@kali: ~
File Edit View Search Terminal Help
nmap
-sU (UDP scans)
While most popular services on the Internet run over the TCP protocol, UDP[5] services are widely deployed. DNS, SNMP, and DHCP (registered ports 53, 161/162, and 67/68) are three of the most common. Because UDP scanning is generally slower and more difficult than TCP, some security auditors ignore these ports. This is a mistake, as exploitable UDP services are quite common and attackers certainly don't ignore the whole protocol. Fortunately, Nmap can help inventory UDP ports.
UDP scan is activated with the -sU option. It can be combined with a TCP scan type such as SYN scan (-sS) to check both protocols during the same run.
UDP scan works by sending a UDP packet to every targeted port. For some common ports such as 53 and 161, a protocol-specific payload is sent to increase response rate, but for most ports the packet is empty unless the --data, --data-string, or --data-length options are specified. If an ICMP port unreachable error (type 3, code 3) is returned, the port is closed. Other ICMP unreachable errors (type 3, codes 0, 1, 2, 9, 10, or 13) mark the port as filtered. Occasionally, a service will respond with a UDP packet, proving that it is open. If no response is received after retransmissions, the port is classified as open|filtered. This means that the port could be open, or perhaps packet filters are blocking the communication. Version detection (-sV) can be used to help differentiate the truly open ports from the filtered ones.
A big challenge with UDP scanning is doing it quickly. Open and filtered ports rarely send any response, leaving Nmap to time out and then conduct retransmissions just in case the probe or response were lost. Closed ports are often an even bigger problem. They usually send back an ICMP port unreachable error. But unlike the RST packets sent by closed TCP ports in response to a SYN or connect scan, many hosts rate limit ICMP port unreachable messages by default. Linux and Solaris are particularly strict about this. For example, the Linux 2.4.20 kernel limits destination unreachable messages to one per second (in net/ipv4/icmp.c).
Nmap detects rate limiting and slows down accordingly to avoid flooding the network with useless packets that the target machine will drop. Unfortunately, a Linux-style limit of one packet per second makes a 65,536-port scan take more than 18 hours. Ideas
Manual page nmap(1) line 632/2076 31% (press h for help or q to quit)

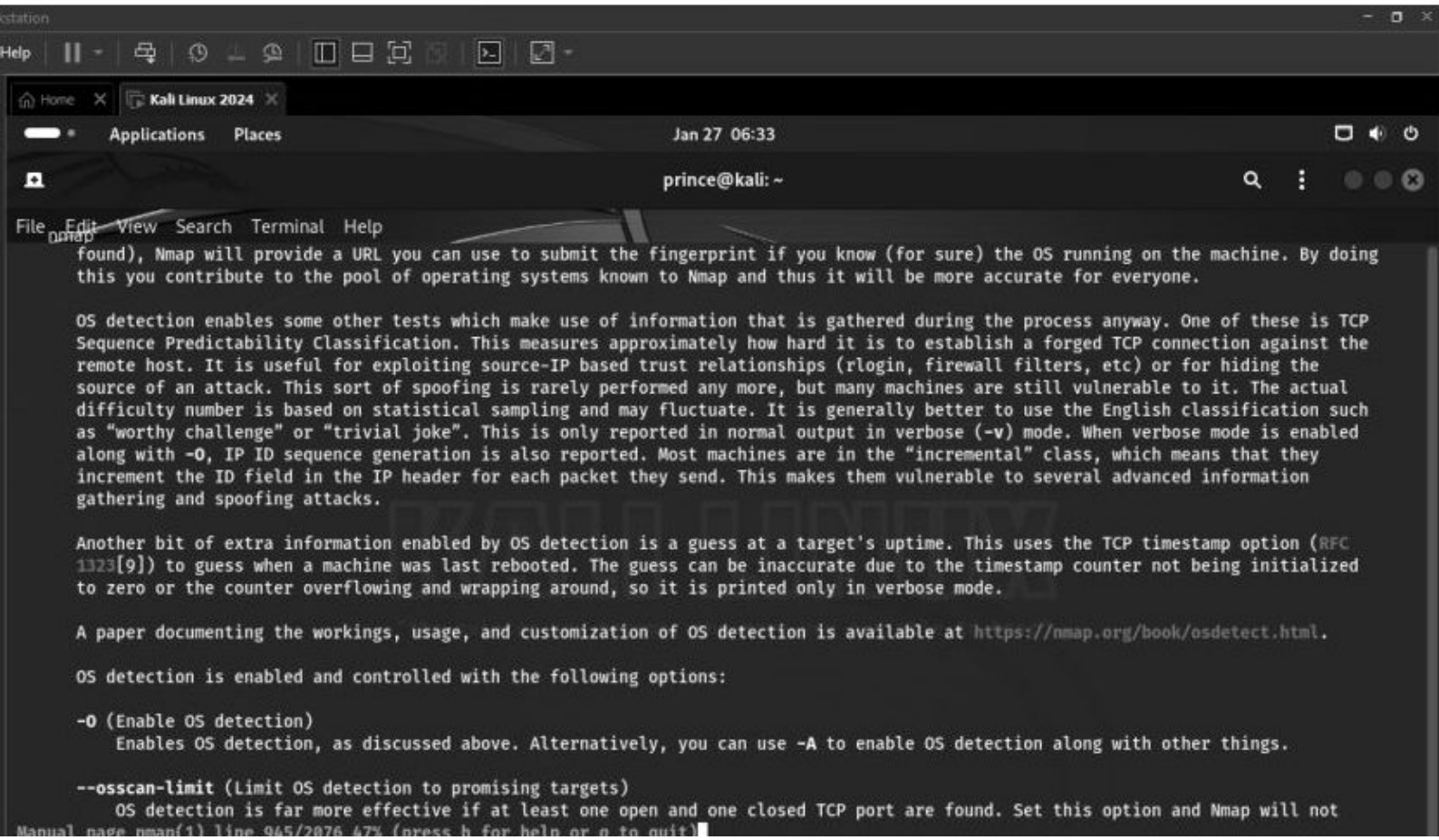

File Edit View Search Terminal Help
nmap
PORT SPECIFICATION AND SCAN ORDER
In addition to all of the scan methods discussed previously, Nmap offers options for specifying which ports are scanned and whether the scan order is randomized or sequential. By default, Nmap scans the most common 1,000 ports for each protocol.
-p port ranges (Only scan specified ports)
This option specifies which ports you want to scan and overrides the default. Individual port numbers are OK, as are ranges separated by a hyphen (e.g. 1-1023). The beginning and/or end values of a range may be omitted, causing Nmap to use 1 and 65535, respectively. So you can specify -p- to scan ports from 1 through 65535. Scanning port zero is allowed if you specify it explicitly. For IP protocol scanning (-sO), this option specifies the protocol numbers you wish to scan for (0-255).
When scanning a combination of protocols (e.g. TCP and UDP), you can specify a particular protocol by preceding the port numbers by T: for TCP, U: for UDP, S: for SCTP, or P: for IP Protocol. The qualifier lasts until you specify another qualifier. For example, the argument -p U:53,111,137,T:21-25,80,139,8080 would scan UDP ports 53, 111,and 137, as well as the listed TCP ports. Note that to scan both UDP and TCP, you have to specify -sU and at least one TCP scan type (such as -sS, -sF, or -sT). If no protocol qualifier is given, the port numbers are added to all protocol lists. Ports can also be specified by name according to what the port is referred to in the nmap-services. You can even use the wildcards * and ? with the names. For example, to scan FTP and all ports whose names begin with "http", use -p ftp,http*. Be careful about shell expansions and quote the argument to -p if unsure.
Ranges of ports can be surrounded by square brackets to indicate ports inside that range that appear in nmap-services. For example, the following will scan all ports in nmap-services equal to or below 1024: -p [-1024]. Be careful with shell expansions and quote the argument to -p if unsure.
--exclude-ports port ranges (Exclude the specified ports from scanning)
This option specifies which ports you do want Nmap to exclude from scanning. The port ranges are specified similar to -p. For IP protocol scanning (-sO), this option specifies the protocol numbers you wish to exclude (0-255).
Manual page nmap(1) line 814/2076 41% (press h for help or q to quit)
mouse pointer inside or press Ctrl+G.
Search
8:29 PM
1/27/2024

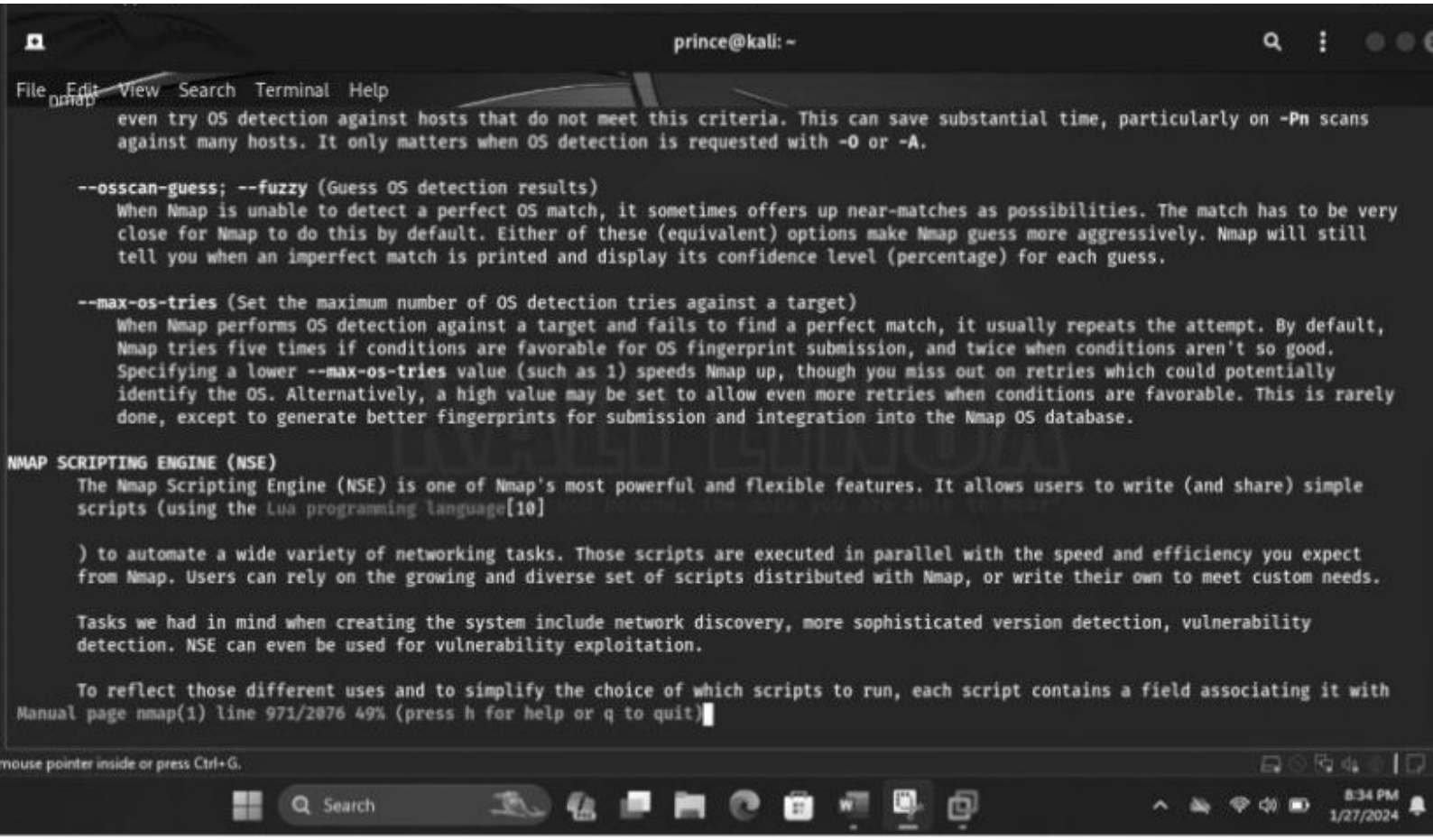

Help
Home
Kali Linux 2024
Applications
Places
Jan 27 06:30
prince@kali: ~
File Edit View Search Terminal Help
nmap
When ports are asked to be excluded, they are excluded from all types of scans (i.e. they will not be scanned under any circumstances). This also includes the discovery phase.
-F (Fast (limited port) scan)
Specifies that you wish to scan fewer ports than the default. Normally Nmap scans the most common 1,000 ports for each scanned protocol. With -F, this is reduced to 100.
Nmap needs an nmap-services file with frequency information in order to know which ports are the most common. If port frequency information isn't available, perhaps because of the use of a custom nmap-services file, Nmap scans all named ports plus ports 1-1024. In that case, -F means to scan only ports that are named in the services file.
-r (Don't randomize ports)
By default, Nmap randomizes the scanned port order (except that certain commonly accessible ports are moved near the beginning for efficiency reasons). This randomization is normally desirable, but you can specify -r for sequential (sorted from lowest to highest) port scanning instead.
--port-ratio ratio<decimal number between 0 and 1>
Scans all ports in nmap-services file with a ratio greater than the one given. ratio must be between 0.0 and 1.0.
--top-ports n
Scans the n highest-ratio ports found in nmap-services file after excluding all ports specified by --exclude-ports. n must be 1 or greater.
SERVICE AND VERSION DETECTION
Point Nmap at a remote machine and it might tell you that ports 25/tcp, 80/tcp, and 53/udp are open. Using its nmap-services database of about 2,200 well-known services, Nmap would report that those ports probably correspond to a mail server (SMTP), web

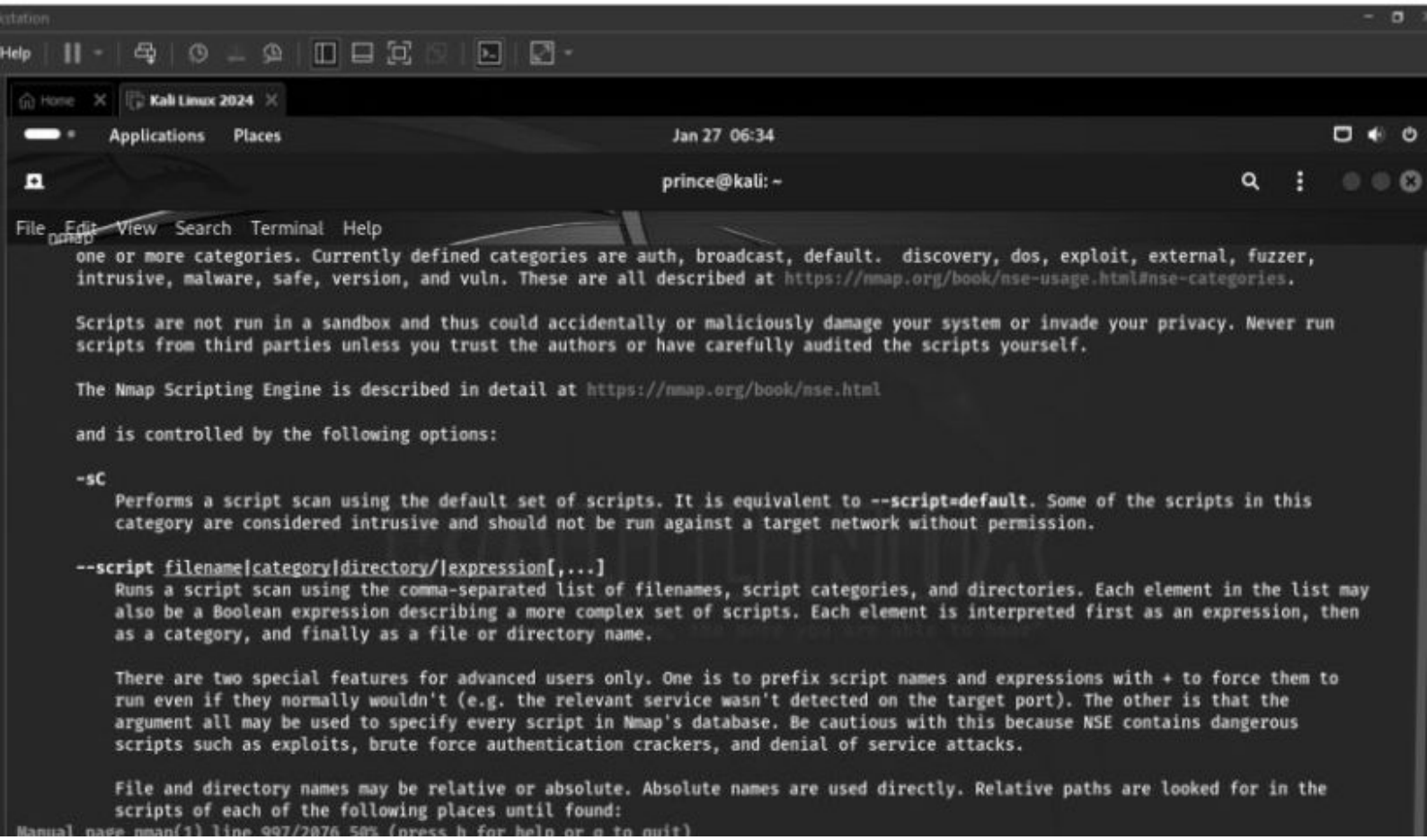

prince@kali: ~
e Edit View Search Terminal Help
server (HTTP), and name server (DNS) respectively. This lookup is usually accurate—the vast majority of daemons listening on TCP port 25 are, in fact, mail servers. However, you should not bet your security on this! People can and do run services on strange ports.
Even if Nmap is right, and the hypothetical server above is running SMTP, HTTP, and DNS servers, that is not a lot of information. When doing vulnerability assessments (or even simple network inventories) of your companies or clients, you really want to know which mail and DNS servers and versions are running. Having an accurate version number helps dramatically in determining which exploits a server is vulnerable to. Version detection helps you obtain this information.
After TCP and/or UDP ports are discovered using one of the other scan methods, version detection interrogates those ports to determine more about what is actually running. The nmap-service-probes database contains probes for querying various services and match expressions to recognize and parse responses. Nmap tries to determine the service protocol (e.g. FTP, SSH, Telnet, HTTP), the application name (e.g. ISC BIND, Apache httpd, Solaris telnetd), the version number, hostname, device type (e.g. printer, router), the OS family (e.g. Windows, Linux). When possible, Nmap also gets the Common Platform Enumeration (CPE) representation of this information. Sometimes miscellaneous details like whether an X server is open to connections, the SSH protocol version, or the KaZaA user name, are available. Of course, most services don't provide all of this information. If Nmap was compiled with OpenSSL support, it will connect to SSL servers to deduce the service listening behind that encryption layer. Some UDP ports are left in the open|filtered state after a UDP port scan is unable to determine whether the port is open or filtered. Version detection will try to elicit a response from these ports (just as it does with open ports), and change the state to open if it succeeds. open|filtered TCP ports are treated the same way. Note that the Nmap -A option enables version detection among other things. A paper documenting the workings, usage, and customization of version detection is available at https://nmap.org/book/vscan.html.
When RPC services are discovered, the Nmap RPC grinder is automatically used to determine the RPC program and version numbers. It takes all the TCP/UDP ports detected as RPC and floods them with SunRPC program NULL commands in an attempt to determine whether they are RPC ports, and if so, what program and version number they serve up. Thus you can effectively obtain the same info as rpcinfo -p even if the target's portmapper is behind a firewall (or protected by TCP wrappers). Decoys do not currently work with
nual page nmap(1) line 867/2076 44% (press h for help or q to quit)
e pointer inside or press Ctrl+G.
Search
8:31 PM
1/27/2024

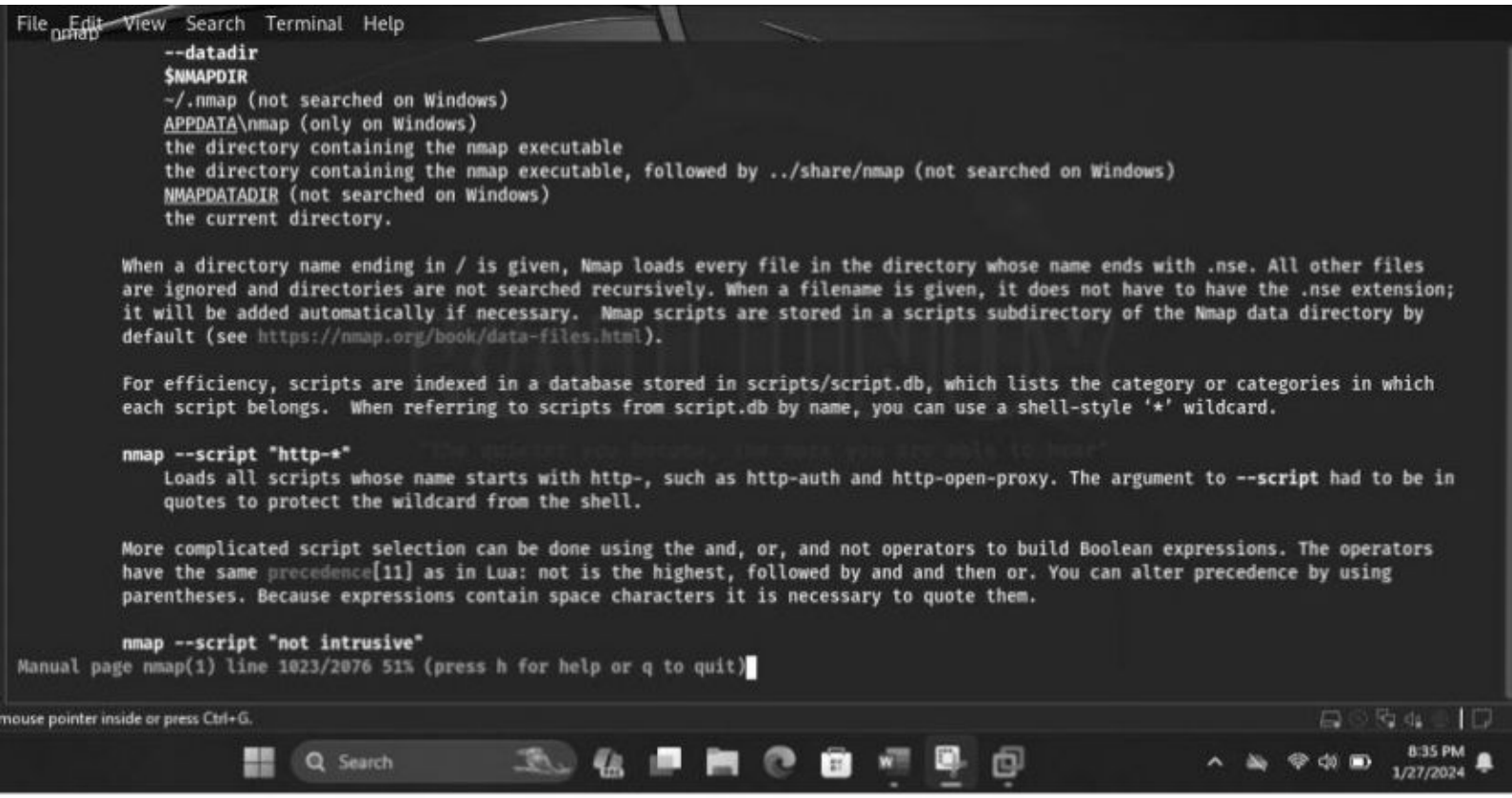

Home
Kali Linux 2024
Applications
Places
Jan 27 06:31
prince@kali: ~
e Edit View Search Terminal Help
RPC scan.
When Nmap receives responses from a service but cannot match them to its database, it prints out a special fingerprint and a URL for you to submit it to if you know for sure what is running on the port. Please take a couple minutes to make the submission so that your find can benefit everyone. Thanks to these submissions, Nmap has about 6,500 pattern matches for more than 650 protocols such as SMTP, FTP, HTTP, etc.
Version detection is enabled and controlled with the following options:
-sV (Version detection)
Enables version detection, as discussed above. Alternatively, you can use -A, which enables version detection among other things.
-sR is an alias for -sV. Prior to March 2011, it was used to active the RPC grinder separately from version detection, but now these options are always combined.
--allports (Don't exclude any ports from version detection)
By default, Nmap version detection skips TCP port 9100 because some printers simply print anything sent to that port, leading to dozens of pages of HTTP GET requests, binary SSL session requests, etc. This behavior can be changed by modifying or removing the Exclude directive in nmap-service-probes, or you can specify --allports to scan all ports regardless of any Exclude directive.
--version-intensity intensity (Set version scan intensity)
When performing a version scan (-sV), Nmap sends a series of probes, each of which is assigned a rarity value between one and nine. The lower-numbered probes are effective against a wide variety of common services, while the higher-numbered ones are rarely useful. The intensity level specifies which probes should be applied. The higher the number, the more likely it is the

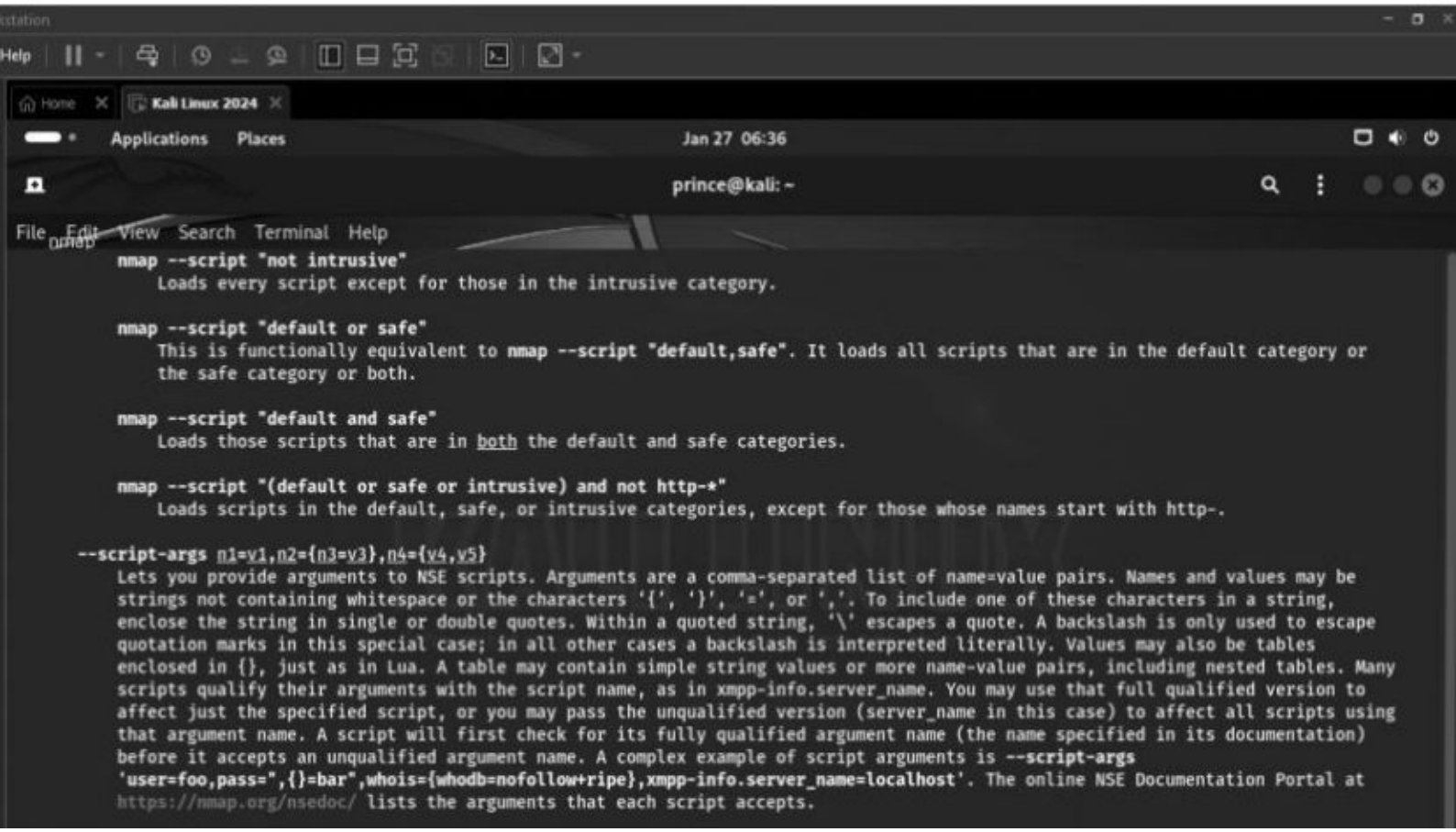

File Edit View Search Terminal Help
service will be correctly identified. However, high intensity scans take longer. The intensity must be between 0 and 9. The default is 7. When a probe is registered to the target port via the nmap-service-probes ports directive, that probe is tried regardless of intensity level. This ensures that the DNS probes will always be attempted against any open port 53, the SSL probe will be done against 443, etc.
--version-light (Enable light mode)
This is a convenience alias for --version-intensity 2. This light mode makes version scanning much faster, but it is slightly less likely to identify services.
--version-all (Try every single probe)
An alias for --version-intensity 9, ensuring that every single probe is attempted against each port.
--version-trace (Trace version scan activity)
This causes Nmap to print out extensive debugging info about what version scanning is doing. It is a subset of what you get with --packet-trace.
OS DETECTION
One of Nmap's best-known features is remote OS detection using TCP/IP stack fingerprinting. Nmap sends a series of TCP and UDP packets to the remote host and examines practically every bit in the responses. After performing dozens of tests such as TCP ISN sampling, TCP options support and ordering, IP ID sampling, and the initial window size check, Nmap compares the results to its nmap-os-db database of more than 2,600 known OS fingerprints and prints out the OS details if there is a match. Each fingerprint includes a freeform textual description of the OS, and a classification which provides the vendor name (e.g. Sun), underlying OS (e.g. Solaris), OS generation (e.g. 10), and device type (general purpose, router, switch, game console, etc). Most fingerprints also have a Common Platform Enumeration (CPE) representation, like cpe:/o:linux:linux_kernel:2.6.
If Nmap is unable to guess the OS of a machine, and conditions are good (e.g. at least one open port and one closed port were
Manual page nmap(1) line 919/2076 46% (press h for help or q to quit)
mouse pointer inside or press Ctrl+G.
Search
8:32 PM
1/27/2024

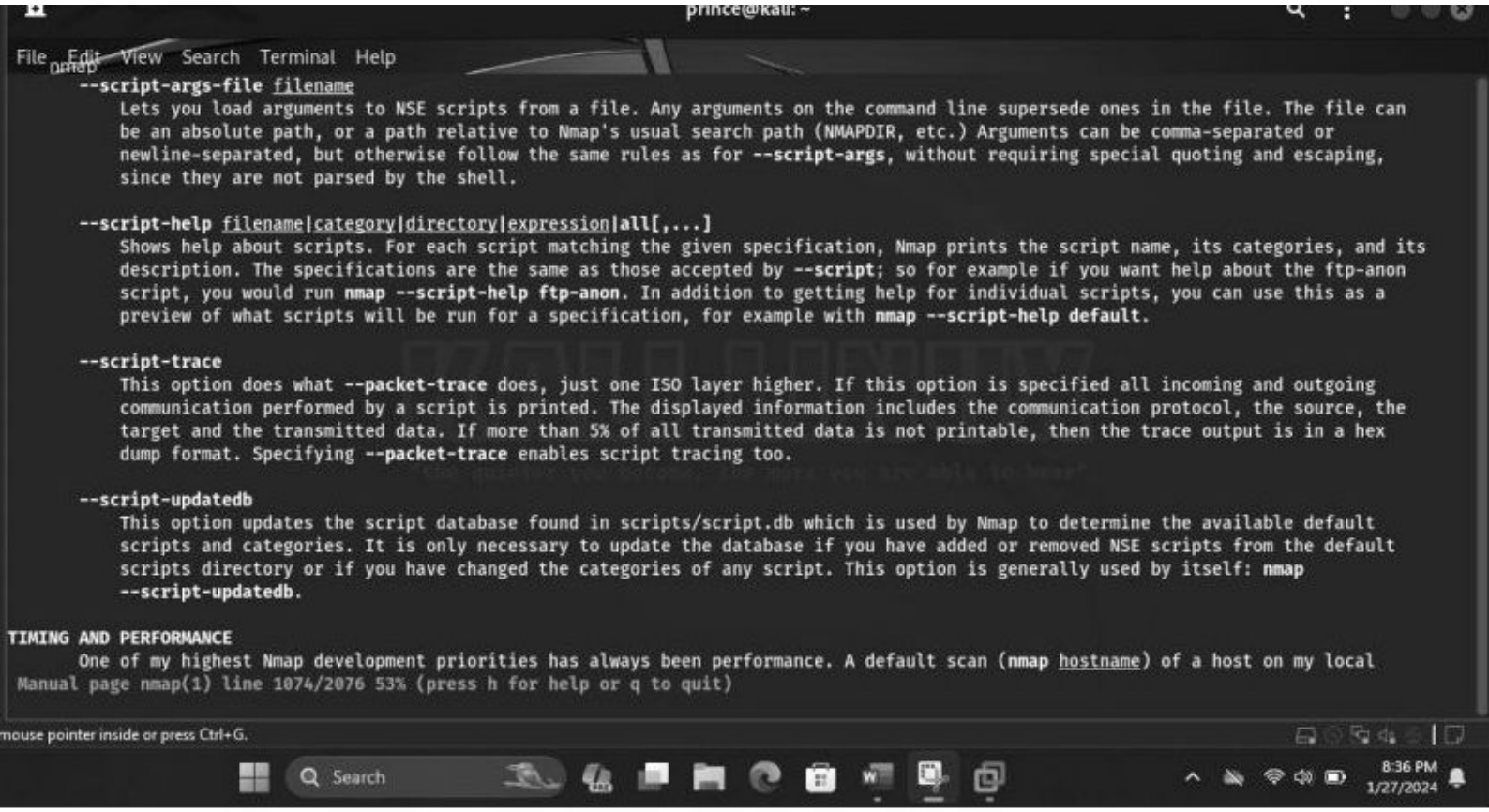

Help
Home
Kali Linux 2024
Applications
Places
Jan 27 06:33
prince@kali: ~
File Edit View Search Terminal Help
found), Nmap will provide a URL you can use to submit the fingerprint if you know (for sure) the OS running on the machine. By doing this you contribute to the pool of operating systems known to Nmap and thus it will be more accurate for everyone.
OS detection enables some other tests which make use of information that is gathered during the process anyway. One of these is TCP Sequence Predictability Classification. This measures approximately how hard it is to establish a forged TCP connection against the remote host. It is useful for exploiting source-IP based trust relationships (rlogin, firewall filters, etc) or for hiding the source of an attack. This sort of spoofing is rarely performed any more, but many machines are still vulnerable to it. The actual difficulty number is based on statistical sampling and may fluctuate. It is generally better to use the English classification such as "worthy challenge" or "trivial joke". This is only reported in normal output in verbose (-v) mode. When verbose mode is enabled along with -O, IP ID sequence generation is also reported. Most machines are in the "incremental" class, which means that they increment the ID field in the IP header for each packet they send. This makes them vulnerable to several advanced information gathering and spoofing attacks.
Another bit of extra information enabled by OS detection is a guess at a target's uptime. This uses the TCP timestamp option (RFC 1323[9]) to guess when a machine was last rebooted. The guess can be inaccurate due to the timestamp counter not being initialized to zero or the counter overflowing and wrapping around, so it is printed only in verbose mode.
A paper documenting the workings, usage, and customization of OS detection is available at https://nmap.org/book/osdetect.html.
OS detection is enabled and controlled with the following options:
-O (Enable OS detection)
Enables OS detection, as discussed above. Alternatively, you can use -A to enable OS detection along with other things.
--osscan-limit (Limit OS detection to promising targets)
OS detection is far more effective if at least one open and one closed TCP port are found. Set this option and Nmap will not
Manual page nmap(1) line 945/2076 47% (press h for help or q to quit)


prince@kali: ~
File Edit View Search Terminal Help
even try OS detection against hosts that do not meet this criteria. This can save substantial time, particularly on -Pn scans against many hosts. It only matters when OS detection is requested with -O or -A.
--osscan-guess; --fuzzy (Guess OS detection results)
When Nmap is unable to detect a perfect OS match, it sometimes offers up near-matches as possibilities. The match has to be very close for Nmap to do this by default. Either of these (equivalent) options make Nmap guess more aggressively. Nmap will still tell you when an imperfect match is printed and display its confidence level (percentage) for each guess.
--max-os-tries (Set the maximum number of OS detection tries against a target)
When Nmap performs OS detection against a target and fails to find a perfect match, it usually repeats the attempt. By default, Nmap tries five times if conditions are favorable for OS fingerprint submission, and twice when conditions aren't so good. Specifying a lower --max-os-tries value (such as 1) speeds Nmap up, though you miss out on retries which could potentially identify the OS. Alternatively, a high value may be set to allow even more retries when conditions are favorable. This is rarely done, except to generate better fingerprints for submission and integration into the Nmap OS database.
NMAP SCRIPTING ENGINE (NSE)
The Nmap Scripting Engine (NSE) is one of Nmap's most powerful and flexible features. It allows users to write (and share) simple scripts (using the Lua programming language[10]
) to automate a wide variety of networking tasks. Those scripts are executed in parallel with the speed and efficiency you expect from Nmap. Users can rely on the growing and diverse set of scripts distributed with Nmap, or write their own to meet custom needs.
Tasks we had in mind when creating the system include network discovery, more sophisticated version detection, vulnerability detection. NSE can even be used for vulnerability exploitation.
To reflect those different uses and to simplify the choice of which scripts to run, each script contains a field associating it with
Manual page nmap(1) line 971/2076 49% (press h for help or q to quit)
mouse pointer inside or press Ctrl+G.
Search
8:34 PM
1/27/2024


Help
Home
Kali Linux 2024
Applications
Places
Jan 27 06:34
prince@kali: ~
File Edit View Search Terminal Help
one or more categories. Currently defined categories are auth, broadcast, default. discovery, dos, exploit, external, fuzzer, intrusive, malware, safe, version, and vuln. These are all described at https://nmap.org/book/nse-usage.html#nse-categories.
Scripts are not run in a sandbox and thus could accidentally or maliciously damage your system or invade your privacy. Never run scripts from third parties unless you trust the authors or have carefully audited the scripts yourself.
The Nmap Scripting Engine is described in detail at https://nmap.org/book/nse.html
and is controlled by the following options:
-sC
Performs a script scan using the default set of scripts. It is equivalent to --script=default. Some of the scripts in this category are considered intrusive and should not be run against a target network without permission.
--script filename|category|directory/|expression[,...]
Runs a script scan using the comma-separated list of filenames, script categories, and directories. Each element in the list may also be a Boolean expression describing a more complex set of scripts. Each element is interpreted first as an expression, then as a category, and finally as a file or directory name.
There are two special features for advanced users only. One is to prefix script names and expressions with + to force them to run even if they normally wouldn't (e.g. the relevant service wasn't detected on the target port). The other is that the argument all may be used to specify every script in Nmap's database. Be cautious with this because NSE contains dangerous scripts such as exploits, brute force authentication crackers, and denial of service attacks.
File and directory names may be relative or absolute. Absolute names are used directly. Relative paths are looked for in the scripts of each of the following places until found:
Manual page nmap(1) line 997/2076 50% (press h for help or q to quit)

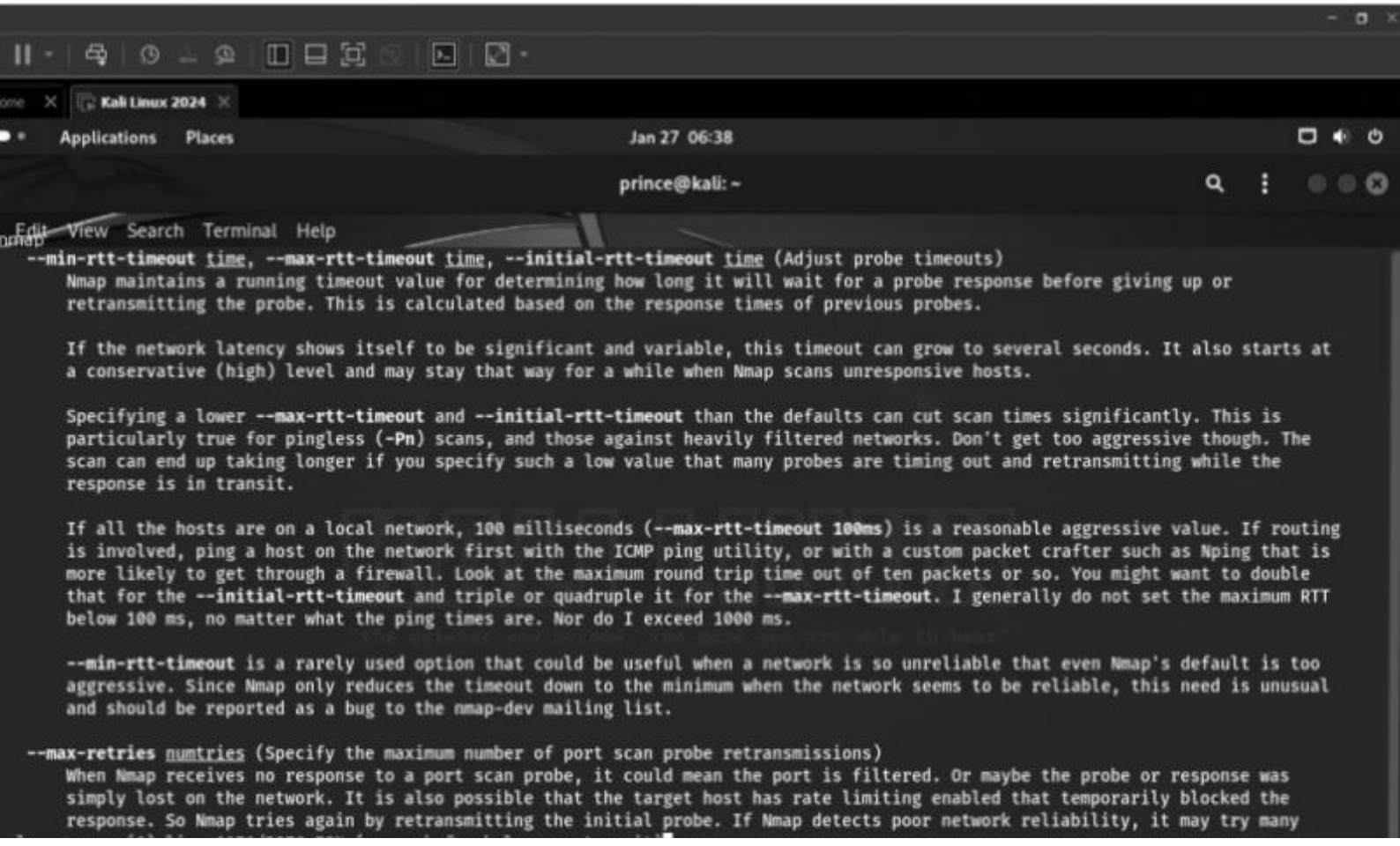

File Edit View Search Terminal Help
--datadir
$NMAPDIR
~/.nmap (not searched on Windows)
APPDATA\nmap (only on Windows)
the directory containing the nmap executable
the directory containing the nmap executable, followed by ../share/nmap (not searched on Windows)
NMAPDATADIR (not searched on Windows)
the current directory.
When a directory name ending in / is given, Nmap loads every file in the directory whose name ends with .nse. All other files are ignored and directories are not searched recursively. When a filename is given, it does not have to have the .nse extension; it will be added automatically if necessary. Nmap scripts are stored in a scripts subdirectory of the Nmap data directory by default (see https://nmap.org/book/data-files.html).
For efficiency, scripts are indexed in a database stored in scripts/script.db, which lists the category or categories in which each script belongs. When referring to scripts from script.db by name, you can use a shell-style '*' wildcard.
nmap --script "http-*"
Loads all scripts whose name starts with http-, such as http-auth and http-open-proxy. The argument to --script had to be in quotes to protect the wildcard from the shell.
More complicated script selection can be done using the and, or, and not operators to build Boolean expressions. The operators have the same precedence[11] as in Lua: not is the highest, followed by and and then or. You can alter precedence by using parentheses. Because expressions contain space characters it is necessary to quote them.
nmap --script "not intrusive"
Manual page nmap(1) line 1023/2076 51% (press h for help or q to quit)
mouse pointer inside or press Ctrl+G.
Search
8:35 PM
1/27/2024


Help
Home
Kali Linux 2024
Applications
Places
Jan 27 06:36
prince@kali: ~
File Edit View Search Terminal Help
nmap --script "not intrusive"
Loads every script except for those in the intrusive category.
nmap --script "default or safe"
This is functionally equivalent to nmap --script "default,safe". It loads all scripts that are in the default category or the safe category or both.
nmap --script "default and safe"
Loads those scripts that are in both the default and safe categories.
nmap --script "(default or safe or intrusive) and not http-*"
Loads scripts in the default, safe, or intrusive categories, except for those whose names start with http-.
--script-args n1=v1,n2={n3=v3},n4={v4,v5}
Lets you provide arguments to NSE scripts. Arguments are a comma-separated list of name=value pairs. Names and values may be strings not containing whitespace or the characters '{', '}', '=', or ','. To include one of these characters in a string, enclose the string in single or double quotes. Within a quoted string, '\' escapes a quote. A backslash is only used to escape quotation marks in this special case; in all other cases a backslash is interpreted literally. Values may also be tables enclosed in {}, just as in Lua. A table may contain simple string values or more name-value pairs, including nested tables. Many scripts qualify their arguments with the script name, as in xmpp-info.server_name. You may use that full qualified version to affect just the specified script, or you may pass the unqualified version (server_name in this case) to affect all scripts using that argument name. A script will first check for its fully qualified argument name (the name specified in its documentation) before it accepts an unqualified argument name. A complex example of script arguments is --script-args 'user=foo,pass=",{}=bar",whois={whodb=nofollow+ripe},xmpp-info.server_name=localhost'. The online NSE Documentation Portal at https://nmap.org/nsedoc/ lists the arguments that each script accepts.


prince@kali: ~
File Edit View Search Terminal Help
nmap
--script-args-file filename
Lets you load arguments to NSE scripts from a file. Any arguments on the command line supersede ones in the file. The file can be an absolute path, or a path relative to Nmap's usual search path (NMAPDIR, etc.) Arguments can be comma-separated or newline-separated, but otherwise follow the same rules as for --script-args, without requiring special quoting and escaping, since they are not parsed by the shell.
--script-help filename|category|directory|expression|all[,...]
Shows help about scripts. For each script matching the given specification, Nmap prints the script name, its categories, and its description. The specifications are the same as those accepted by --script; so for example if you want help about the ftp-anon script, you would run nmap --script-help ftp-anon. In addition to getting help for individual scripts, you can use this as a preview of what scripts will be run for a specification, for example with nmap --script-help default.
--script-trace
This option does what --packet-trace does, just one ISO layer higher. If this option is specified all incoming and outgoing communication performed by a script is printed. The displayed information includes the communication protocol, the source, the target and the transmitted data. If more than 5% of all transmitted data is not printable, then the trace output is in a hex dump format. Specifying --packet-trace enables script tracing too.
--script-updatedb
This option updates the script database found in scripts/script.db which is used by Nmap to determine the available default scripts and categories. It is only necessary to update the database if you have added or removed NSE scripts from the default scripts directory or if you have changed the categories of any script. This option is generally used by itself: nmap --script-updatedb.
TIMING AND PERFORMANCE
One of my highest Nmap development priorities has always been performance. A default scan (nmap hostname) of a host on my local
Manual page nmap(1) line 1074/2076 53% (press h for help or q to quit)
mouse pointer inside or press Ctrl+G.
Search
8:36 PM
1/27/2024


Help
Home
Kali Linux 2024
Applications
Places
Jan 27 06:37
prince@kali: ~
File Edit View Search Terminal Help
nmap
network takes a fifth of a second. That is barely enough time to blink, but adds up when you are scanning hundreds or thousands of hosts. Moreover, certain scan options such as UDP scanning and version detection can increase scan times substantially. So can certain firewall configurations, particularly response rate limiting. While Nmap utilizes parallelism and many advanced algorithms to accelerate these scans, the user has ultimate control over how Nmap runs. Expert users carefully craft Nmap commands to obtain only the information they care about while meeting their time constraints.
Techniques for improving scan times include omitting non-critical tests, and upgrading to the latest version of Nmap (performance enhancements are made frequently). Optimizing timing parameters can also make a substantial difference. Those options are listed below.
Some options accept a time parameter. This is specified in seconds by default, though you can append 'ms', 's', 'm', or 'h' to the value to specify milliseconds, seconds, minutes, or hours. So the --host-timeout arguments 900000ms, 900, 900s, and 15m all do the same thing.
--min-hostgroup numhosts; --max-hostgroup numhosts (Adjust parallel scan group sizes)
Nmap has the ability to port scan or version scan multiple hosts in parallel. Nmap does this by dividing the target IP space into groups and then scanning one group at a time. In general, larger groups are more efficient. The downside is that host results can't be provided until the whole group is finished. So if Nmap started out with a group size of 50, the user would not receive any reports (except for the updates offered in verbose mode) until the first 50 hosts are completed.
By default, Nmap takes a compromise approach to this conflict. It starts out with a group size as low as five so the first results come quickly and then increases the groupsize to as high as 1024. The exact default numbers depend on the options given. For efficiency reasons, Nmap uses larger group sizes for UDP or few-port TCP scans.
When a maximum group size is specified with --max-hostgroup, Nmap will never exceed that size. Specify a minimum size with --min-hostgroup and Nmap will try to keep group sizes above that level. Nmap may have to use smaller groups than you specify if


prince@kali: ~
Edit View Search Terminal Help
there are not enough target hosts left on a given interface to fulfill the specified minimum. Both may be set to keep the group size within a specific range, though this is rarely desired.
These options do not have an effect during the host discovery phase of a scan. This includes plain ping scans (-sn). Host discovery always works in large groups of hosts to improve speed and accuracy.
The primary use of these options is to specify a large minimum group size so that the full scan runs more quickly. A common choice is 256 to scan a network in /24 sized chunks. For a scan with many ports, exceeding that number is unlikely to help much. For scans of just a few port numbers, host group sizes of 2048 or more may be helpful.
--min-parallelism numprobes; --max-parallelism numprobes (Adjust probe parallelization)
These options control the total number of probes that may be outstanding for a host group. They are used for port scanning and host discovery. By default, Nmap calculates an ever-changing ideal parallelism based on network performance. If packets are being dropped, Nmap slows down and allows fewer outstanding probes. The ideal probe number slowly rises as the network proves itself worthy. These options place minimum or maximum bounds on that variable. By default, the ideal parallelism can drop to one if the network proves unreliable and rise to several hundred in perfect conditions.
The most common usage is to set --min-parallelism to a number higher than one to speed up scans of poorly performing hosts or networks. This is a risky option to play with, as setting it too high may affect accuracy. Setting this also reduces Nmap's ability to control parallelism dynamically based on network conditions. A value of 10 might be reasonable, though I only adjust this value as a last resort.
The --max-parallelism option is sometimes set to one to prevent Nmap from sending more than one probe at a time to hosts. The --scan-delay option, discussed later, is another way to do this.
--min-rtt-timeout time, --max-rtt-timeout time, --initial-rtt-timeout time (Adjust probe timeouts)
ual page nmap(1) line 1126/2076 56% (press h for help or q to quit)
pointer inside or press Ctrl+G.
Search
8:37 PM
1/27/2024


Kali Linux 2024
Applications Places
Jan 27 06:38
prince@kali: ~
Edit View Search Terminal Help
--min-rtt-timeout time, --max-rtt-timeout time, --initial-rtt-timeout time (Adjust probe timeouts)
Nmap maintains a running timeout value for determining how long it will wait for a probe response before giving up or retransmitting the probe. This is calculated based on the response times of previous probes.
If the network latency shows itself to be significant and variable, this timeout can grow to several seconds. It also starts at a conservative (high) level and may stay that way for a while when Nmap scans unresponsive hosts.
Specifying a lower --max-rtt-timeout and --initial-rtt-timeout than the defaults can cut scan times significantly. This is particularly true for pingless (-Pn) scans, and those against heavily filtered networks. Don't get too aggressive though. The scan can end up taking longer if you specify such a low value that many probes are timing out and retransmitting while the response is in transit.
If all the hosts are on a local network, 100 milliseconds (--max-rtt-timeout 100ms) is a reasonable aggressive value. If routing is involved, ping a host on the network first with the ICMP ping utility, or with a custom packet crafter such as Nping that is more likely to get through a firewall. Look at the maximum round trip time out of ten packets or so. You might want to double that for the --initial-rtt-timeout and triple or quadruple it for the --max-rtt-timeout. I generally do not set the maximum RTT below 100 ms, no matter what the ping times are. Nor do I exceed 1000 ms.
--min-rtt-timeout is a rarely used option that could be useful when a network is so unreliable that even Nmap's default is too aggressive. Since Nmap only reduces the timeout down to the minimum when the network seems to be reliable, this need is unusual and should be reported as a bug to the nmap-dev mailing list.
--max-retries numtries (Specify the maximum number of port scan probe retransmissions)
When Nmap receives no response to a port scan probe, it could mean the port is filtered. Or maybe the probe or response was simply lost on the network. It is also possible that the target host has rate limiting enabled that temporarily blocked the response. So Nmap tries again by retransmitting the initial probe. If Nmap detects poor network reliability, it may try many

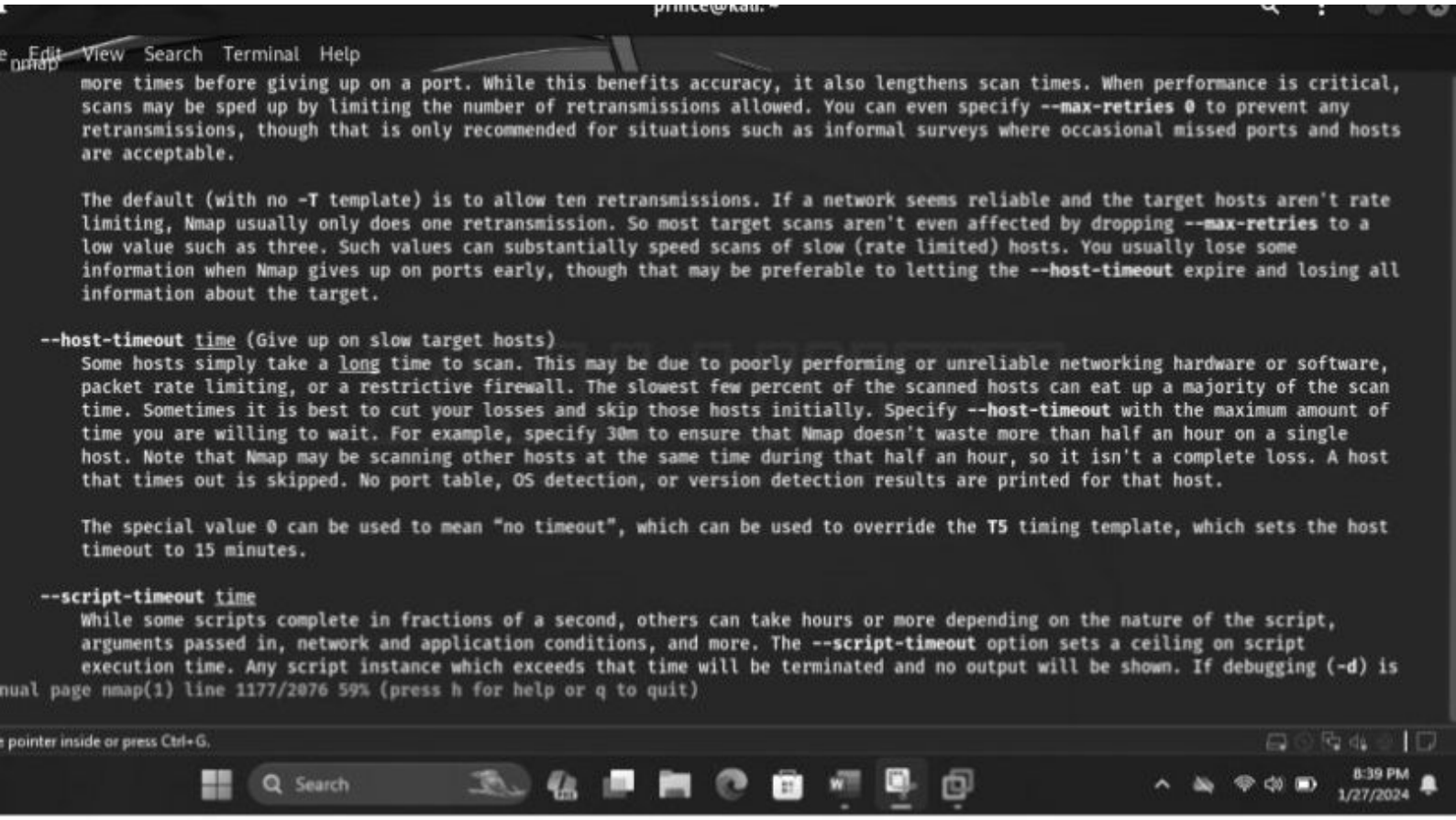
more times before giving up on a port. While this benefits accuracy, it also lengthens scan times. When performance is critical, scans may be sped up by limiting the number of retransmissions allowed. You can even specify **--max-retries 0** to prevent any retransmissions, though that is only recommended for situations such as informal surveys where occasional missed ports and hosts are acceptable.

The default (with no **-T** template) is to allow ten retransmissions. If a network seems reliable and the target hosts aren't rate limiting, Nmap usually only does one retransmission. So most target scans aren't even affected by dropping **--max-retries** to a low value such as three. Such values can substantially speed scans of slow (rate limited) hosts. You usually lose some information when Nmap gives up on ports early, though that may be preferable to letting the **--host-timeout** expire and losing all information about the target.

--host-timeout time (Give up on slow target hosts)
Some hosts simply take a long time to scan. This may be due to poorly performing or unreliable networking hardware or software, packet rate limiting, or a restrictive firewall. The slowest few percent of the scanned hosts can eat up a majority of the scan time. Sometimes it is best to cut your losses and skip those hosts initially. Specify **--host-timeout** with the maximum amount of time you are willing to wait. For example, specify 30m to ensure that Nmap doesn't waste more than half an hour on a single host. Note that Nmap may be scanning other hosts at the same time during that half an hour, so it isn't a complete loss. A host that times out is skipped. No port table, OS detection, or version detection results are printed for that host.

The special value **0** can be used to mean "no timeout", which can be used to override the **T5** timing template, which sets the host timeout to 15 minutes.

--script-timeout time
While some scripts complete in fractions of a second, others can take hours or more depending on the nature of the script, arguments passed in, network and application conditions, and more. The **--script-timeout** option sets a ceiling on script execution time. Any script instance which exceeds that time will be terminated and no output will be shown. If debugging (**-d**) is



enabled, Nmap will report on each timeout. For host and service scripts, a script instance only scans a single target host or port and the timeout period will be reset for the next instance.

The special value **0** can be used to mean "no timeout", which can be used to override the **T5** timing template, which sets the script timeout to 10 minutes.

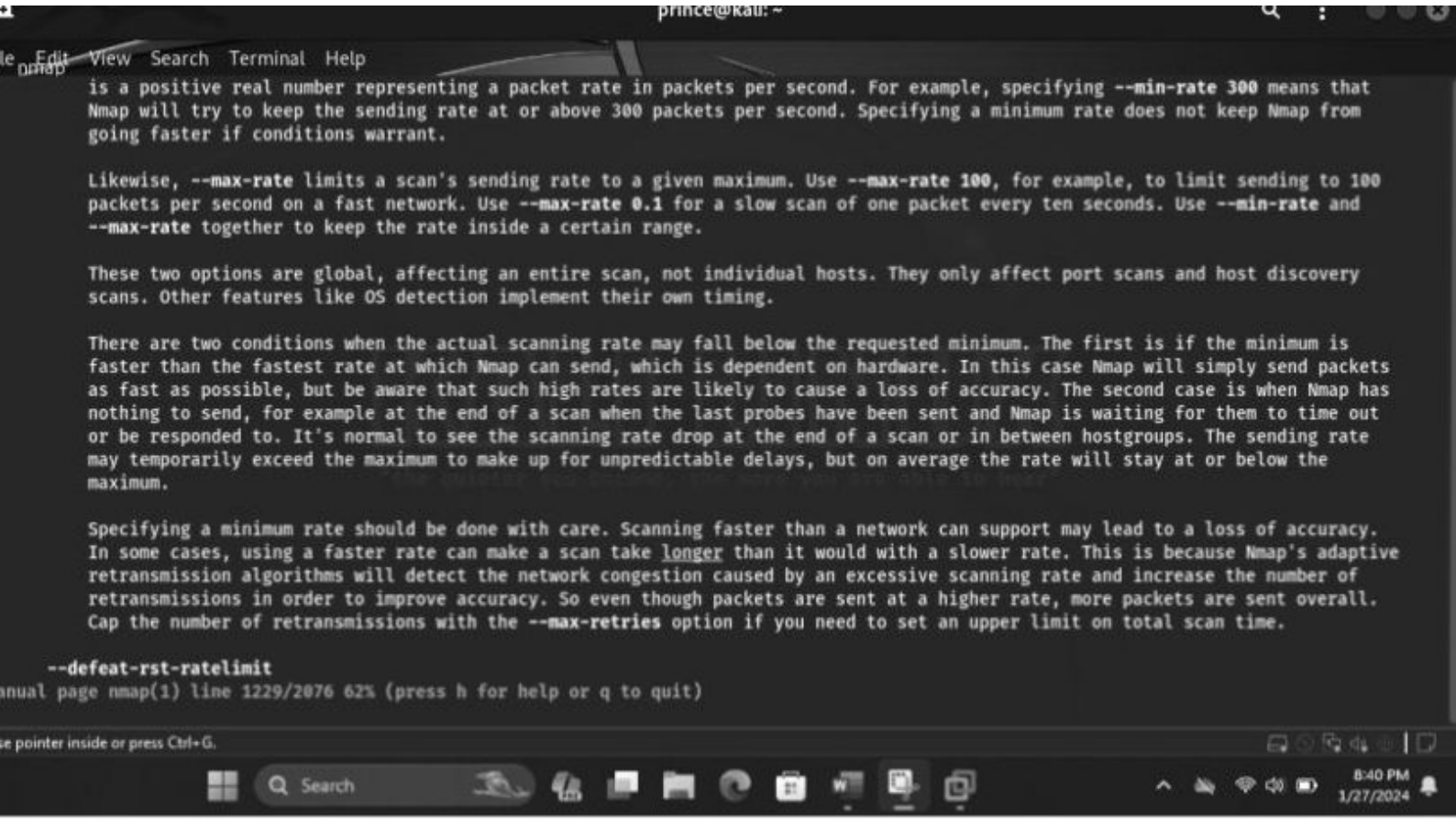
--scan-delay time; **--max-scan-delay** time (Adjust delay between probes)
This option causes Nmap to wait at least the given amount of time between each probe it sends to a given host. This is particularly useful in the case of rate limiting. Solaris machines (among many others) will usually respond to UDP scan probe packets with only one ICMP message per second. Any more than that sent by Nmap will be wasteful. A **--scan-delay** of 1s will keep Nmap at that slow rate. Nmap tries to detect rate limiting and adjust the scan delay accordingly, but it doesn't hurt to specify it explicitly if you already know what rate works best.

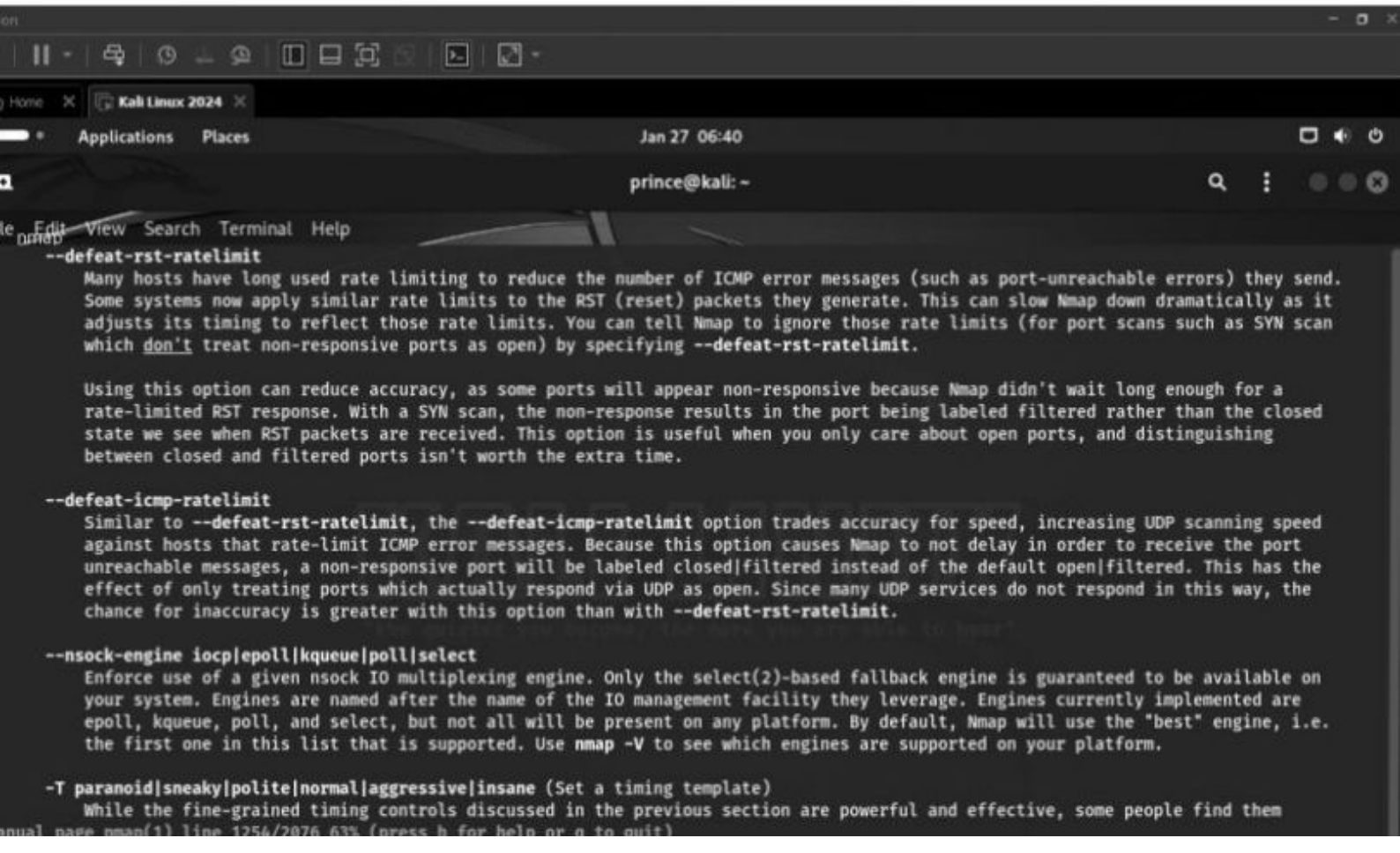
When Nmap adjusts the scan delay upward to cope with rate limiting, the scan slows down dramatically. The **--max-scan-delay** option specifies the largest delay that Nmap will allow. A low **--max-scan-delay** can speed up Nmap, but it is risky. Setting this value too low can lead to wasteful packet retransmissions and possible missed ports when the target implements strict rate limiting.

Another use of **--scan-delay** is to evade threshold based intrusion detection and prevention systems (IDS/IPS).

--min-rate number; **--max-rate** number (Directly control the scanning rate)
Nmap's dynamic timing does a good job of finding an appropriate speed at which to scan. Sometimes, however, you may happen to know an appropriate scanning rate for a network, or you may have to guarantee that a scan will be finished by a certain time. Or perhaps you must keep Nmap from scanning too quickly. The **--min-rate** and **--max-rate** options are designed for these situations.

When the **--min-rate** option is given Nmap will do its best to send packets as fast as or faster than the given rate. The argument


prince@kali: ~
File Edit View Search Terminal Help
nmap
is a positive real number representing a packet rate in packets per second. For example, specifying --min-rate 300 means that Nmap will try to keep the sending rate at or above 300 packets per second. Specifying a minimum rate does not keep Nmap from going faster if conditions warrant.
Likewise, --max-rate limits a scan's sending rate to a given maximum. Use --max-rate 100, for example, to limit sending to 100 packets per second on a fast network. Use --max-rate 0.1 for a slow scan of one packet every ten seconds. Use --min-rate and --max-rate together to keep the rate inside a certain range.
These two options are global, affecting an entire scan, not individual hosts. They only affect port scans and host discovery scans. Other features like OS detection implement their own timing.
There are two conditions when the actual scanning rate may fall below the requested minimum. The first is if the minimum is faster than the fastest rate at which Nmap can send, which is dependent on hardware. In this case Nmap will simply send packets as fast as possible, but be aware that such high rates are likely to cause a loss of accuracy. The second case is when Nmap has nothing to send, for example at the end of a scan when the last probes have been sent and Nmap is waiting for them to time out or be responded to. It's normal to see the scanning rate drop at the end of a scan or in between hostgroups. The sending rate may temporarily exceed the maximum to make up for unpredictable delays, but on average the rate will stay at or below the maximum.
Specifying a minimum rate should be done with care. Scanning faster than a network can support may lead to a loss of accuracy. In some cases, using a faster rate can make a scan take longer than it would with a slower rate. This is because Nmap's adaptive retransmission algorithms will detect the network congestion caused by an excessive scanning rate and increase the number of retransmissions in order to improve accuracy. So even though packets are sent at a higher rate, more packets are sent overall. Cap the number of retransmissions with the --max-retries option if you need to set an upper limit on total scan time.
--defeat-rst-ratelimit
anual page nmap(1) line 1229/2076 62% (press h for help or q to quit)
se pointer inside or press Ctrl+G.
Search
8:40 PM
1/27/2024


Home
Kali Linux 2024
Applications Places
Jan 27 06:40
prince@kali: ~
File Edit View Search Terminal Help
nmap
--defeat-rst-ratelimit
Many hosts have long used rate limiting to reduce the number of ICMP error messages (such as port-unreachable errors) they send. Some systems now apply similar rate limits to the RST (reset) packets they generate. This can slow Nmap down dramatically as it adjusts its timing to reflect those rate limits. You can tell Nmap to ignore those rate limits (for port scans such as SYN scan which don't treat non-responsive ports as open) by specifying --defeat-rst-ratelimit.
Using this option can reduce accuracy, as some ports will appear non-responsive because Nmap didn't wait long enough for a rate-limited RST response. With a SYN scan, the non-response results in the port being labeled filtered rather than the closed state we see when RST packets are received. This option is useful when you only care about open ports, and distinguishing between closed and filtered ports isn't worth the extra time.
--defeat-icmp-ratelimit
Similar to --defeat-rst-ratelimit, the --defeat-icmp-ratelimit option trades accuracy for speed, increasing UDP scanning speed against hosts that rate-limit ICMP error messages. Because this option causes Nmap to not delay in order to receive the port unreachable messages, a non-responsive port will be labeled closed|filtered instead of the default open|filtered. This has the effect of only treating ports which actually respond via UDP as open. Since many UDP services do not respond in this way, the chance for inaccuracy is greater with this option than with --defeat-rst-ratelimit.
--nsock-engine iocp|epoll|kqueue|poll|select
Enforce use of a given nsock IO multiplexing engine. Only the select(2)-based fallback engine is guaranteed to be available on your system. Engines are named after the name of the IO management facility they leverage. Engines currently implemented are epoll, kqueue, poll, and select, but not all will be present on any platform. By default, Nmap will use the "best" engine, i.e. the first one in this list that is supported. Use nmap -V to see which engines are supported on your platform.
-T paranoid|sneaky|polite|normal|aggressive|insane (Set a timing template)
While the fine-grained timing controls discussed in the previous section are powerful and effective, some people find them
anual page nmap(1) line 1254/2076 63% (press h for help or q to quit)

File Edit View Search Terminal Help

confusing. Moreover, choosing the appropriate values can sometimes take more time than the scan you are trying to optimize. Fortunately, Nmap offers a simpler approach, with six timing templates. You can specify them with the **-T** option and their number (0–5) or their name. The template names are **paranoid** (**0**), **sneaky** (**1**), **polite** (2), **normal** (3), **aggressive** (4), and **insane** (**5**). The first two are for IDS evasion. Polite mode slows down the scan to use less bandwidth and target machine resources. Normal mode is the default and so **-T3** does nothing. Aggressive mode speeds scans up by making the assumption that you are on a reasonably fast and reliable network. Finally insane mode assumes that you are on an extraordinarily fast network or are willing to sacrifice some accuracy for speed.

These templates allow the user to specify how aggressive they wish to be, while leaving Nmap to pick the exact timing values. The templates also make some minor speed adjustments for which fine-grained control options do not currently exist. For example, **-T4** prohibits the dynamic scan delay from exceeding 10 ms for TCP ports and **-T5** caps that value at 5 ms. Templates can be used in combination with fine-grained controls, and the fine-grained controls that you specify will take precedence over the timing template default for that parameter. I recommend using **-T4** when scanning reasonably modern and reliable networks. Keep that option even when you add fine-grained controls so that you benefit from those extra minor optimizations that it enables.

If you are on a decent broadband or ethernet connection, I would recommend always using **-T4**. Some people love **-T5** though it is too aggressive for my taste. People sometimes specify **-T2** because they think it is less likely to crash hosts or because they consider themselves to be polite in general. They often don't realize just how slow **-T polite** really is. Their scan may take ten times longer than a default scan. Machine crashes and bandwidth problems are rare with the default timing options (**-T3**) and so I normally recommend that for cautious scanners. Omitting version detection is far more effective than playing with timing values at reducing these problems.

While **-T0** and **-T1** may be useful for avoiding IDS alerts, they will take an extraordinarily long time to scan thousands of machines or ports. For such a long scan, you may prefer to set the exact timing values you need rather than rely on the canned **-T0** and **-T1** values.

Manual page nmap(1) line 1280/2076 65% (press h for help or q to quit)

mouse pointer inside or press Ctrl+G.

Search

8:41 PM
1/27/2024

Help

Home | Kali Linux 2024

Applications Places

Jan 27 06:41

prince@kali: ~

File Edit View Search Terminal Help

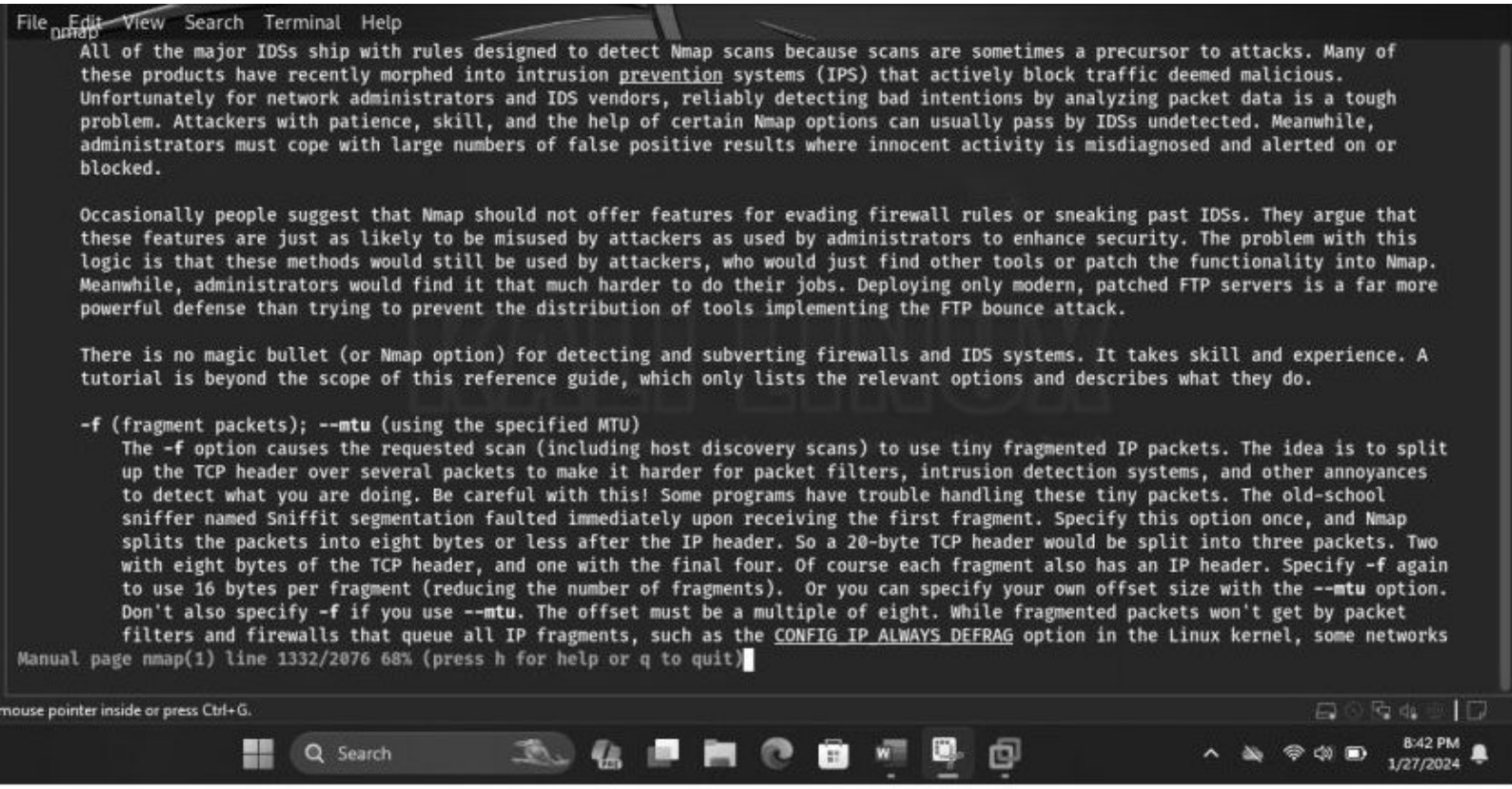
The main effects of **T0** are serializing the scan so only one port is scanned at a time, and waiting five minutes between sending each probe. **T1** and **T2** are similar but they only wait 15 seconds and 0.4 seconds, respectively, between probes. **T3** is Nmap's default behavior, which includes parallelization. **-T4** does the equivalent of **--max-rtt-timeout 1250ms --min-rtt-timeout 100ms --initial-rtt-timeout 500ms --max-retries 6** and sets the maximum TCP and SCTP scan delay to 10ms. **T5** does the equivalent of **--max-rtt-timeout 300ms --min-rtt-timeout 50ms --initial-rtt-timeout 250ms --max-retries 2 --host-timeout 15m --script-timeout 10m** as well as setting the maximum TCP and SCTP scan delay to 5ms. Maximum UDP scan delay is not set by **T4** or **T5**, but it can be set with the **--max-scan-delay** option.

FIREWALL/IDS EVASION AND SPOOFING

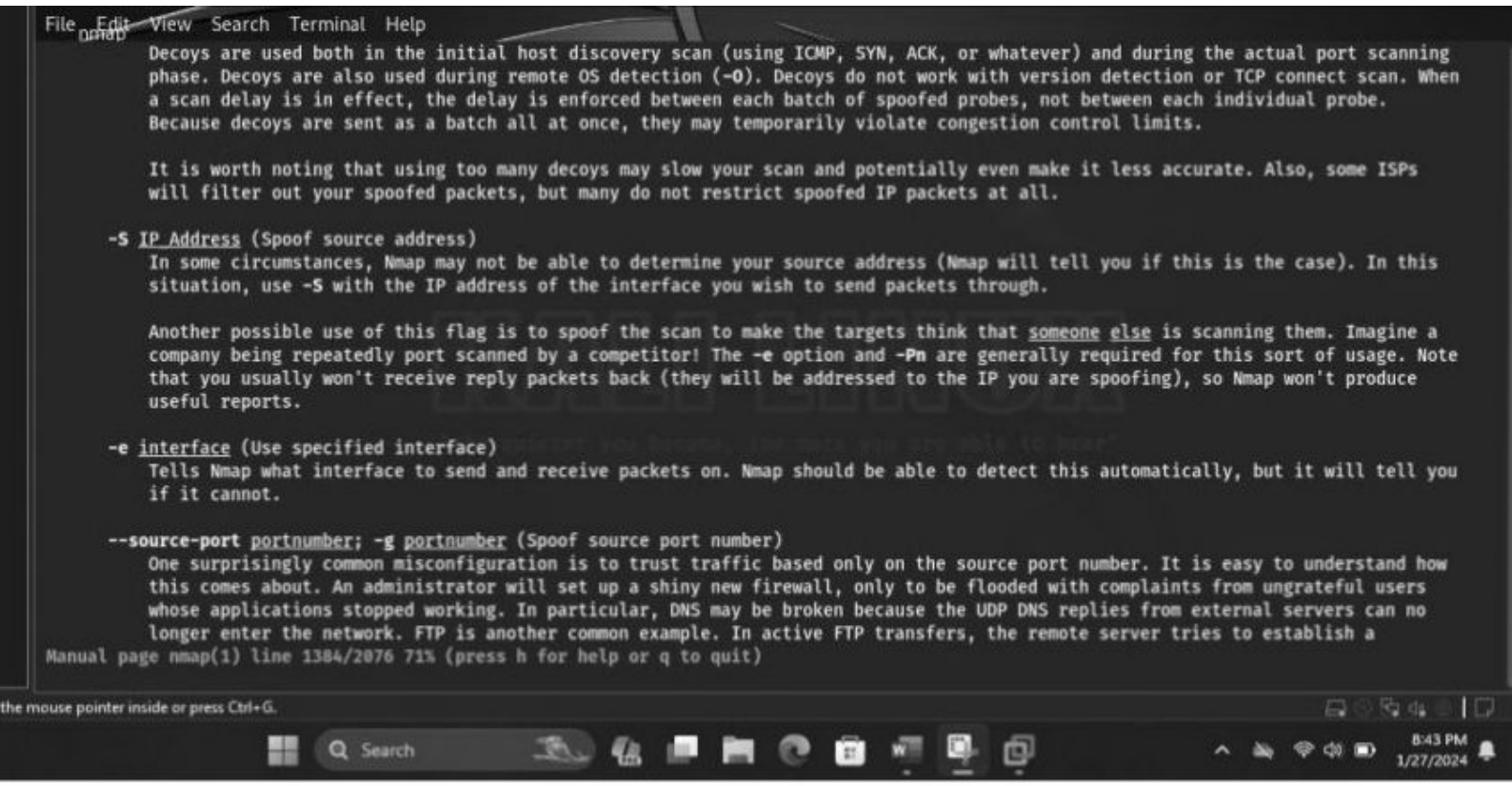
Many Internet pioneers envisioned a global open network with a universal IP address space allowing virtual connections between any two nodes. This allows hosts to act as true peers, serving and retrieving information from each other. People could access all of their home systems from work, changing the climate control settings or unlocking the doors for early guests. This vision of universal connectivity has been stifled by address space shortages and security concerns. In the early 1990s, organizations began deploying firewalls for the express purpose of reducing connectivity. Huge networks were cordoned off from the unfiltered Internet by application proxies, network address translation, and packet filters. The unrestricted flow of information gave way to tight regulation of approved communication channels and the content that passes over them.

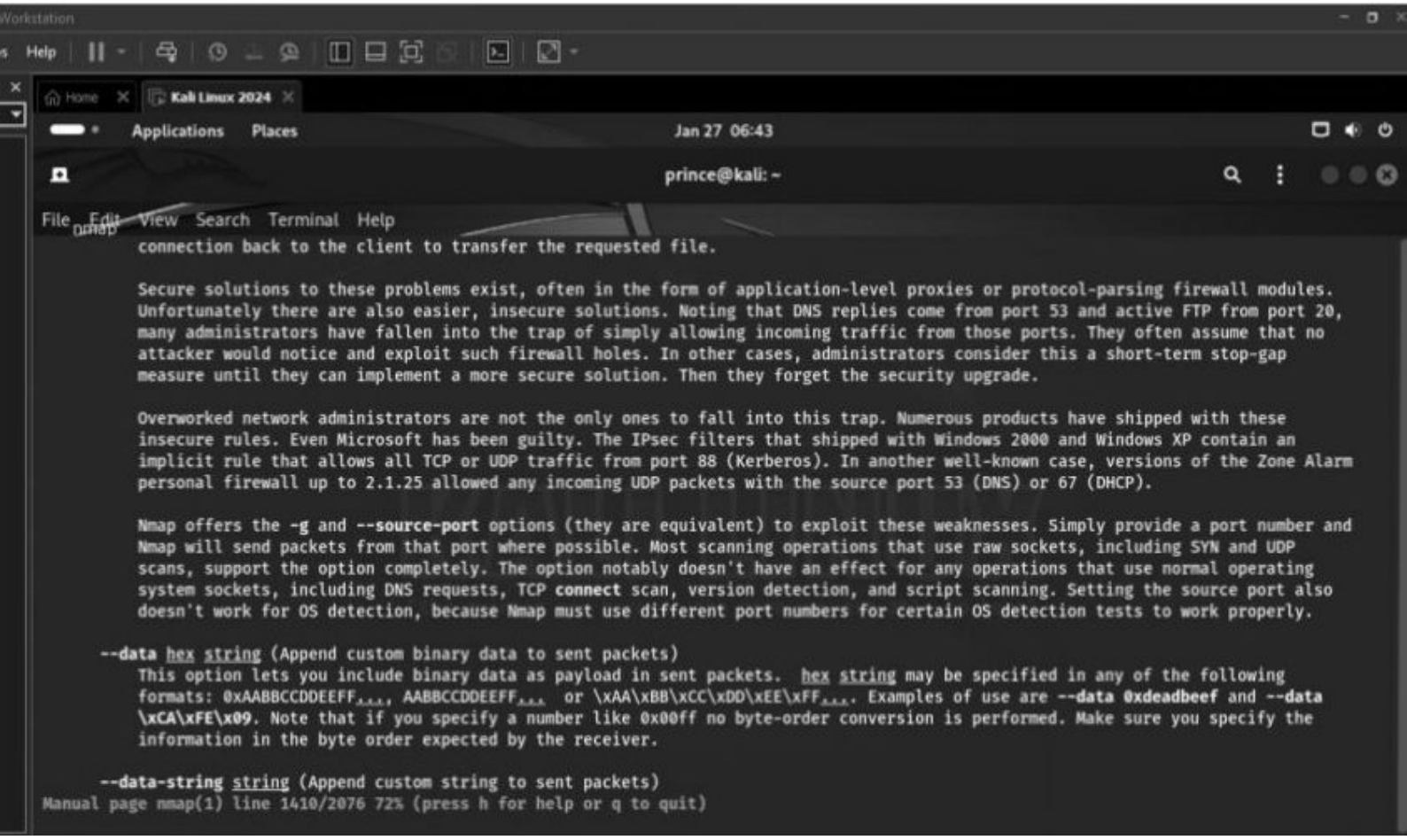
Network obstructions such as firewalls can make mapping a network exceedingly difficult. It will not get any easier, as stifling casual reconnaissance is often a key goal of implementing the devices. Nevertheless, Nmap offers many features to help understand these complex networks, and to verify that filters are working as intended. It even supports mechanisms for bypassing poorly implemented defenses. One of the best methods of understanding your network security posture is to try to defeat it. Place yourself in the mind-set of an attacker, and deploy techniques from this section against your networks. Launch an FTP bounce scan, idle scan, fragmentation attack, or try to tunnel through one of your own proxies.

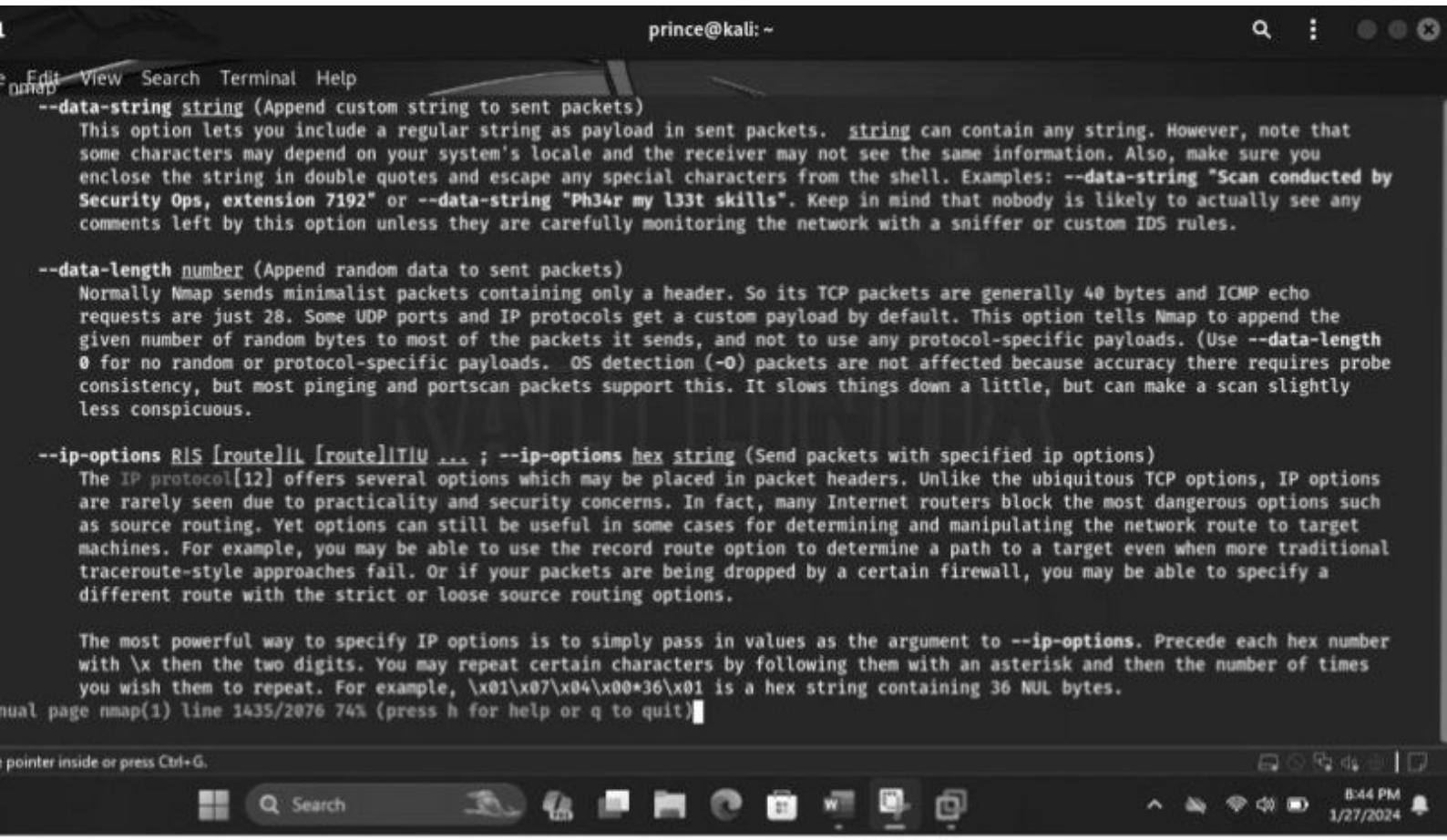
In addition to restricting network activity, companies are increasingly monitoring traffic with intrusion detection systems (IDS).


File Edit View Search Terminal Help
nmap
All of the major IDSs ship with rules designed to detect Nmap scans because scans are sometimes a precursor to attacks. Many of these products have recently morphed into intrusion prevention systems (IPS) that actively block traffic deemed malicious. Unfortunately for network administrators and IDS vendors, reliably detecting bad intentions by analyzing packet data is a tough problem. Attackers with patience, skill, and the help of certain Nmap options can usually pass by IDSs undetected. Meanwhile, administrators must cope with large numbers of false positive results where innocent activity is misdiagnosed and alerted on or blocked.
Occasionally people suggest that Nmap should not offer features for evading firewall rules or sneaking past IDSs. They argue that these features are just as likely to be misused by attackers as used by administrators to enhance security. The problem with this logic is that these methods would still be used by attackers, who would just find other tools or patch the functionality into Nmap. Meanwhile, administrators would find it that much harder to do their jobs. Deploying only modern, patched FTP servers is a far more powerful defense than trying to prevent the distribution of tools implementing the FTP bounce attack.
There is no magic bullet (or Nmap option) for detecting and subverting firewalls and IDS systems. It takes skill and experience. A tutorial is beyond the scope of this reference guide, which only lists the relevant options and describes what they do.
-f (fragment packets); --mtu (using the specified MTU)
The -f option causes the requested scan (including host discovery scans) to use tiny fragmented IP packets. The idea is to split up the TCP header over several packets to make it harder for packet filters, intrusion detection systems, and other annoyances to detect what you are doing. Be careful with this! Some programs have trouble handling these tiny packets. The old-school sniffer named Sniffit segmentation faulted immediately upon receiving the first fragment. Specify this option once, and Nmap splits the packets into eight bytes or less after the IP header. So a 20-byte TCP header would be split into three packets. Two with eight bytes of the TCP header, and one with the final four. Of course each fragment also has an IP header. Specify -f again to use 16 bytes per fragment (reducing the number of fragments). Or you can specify your own offset size with the --mtu option. Don't also specify -f if you use --mtu. The offset must be a multiple of eight. While fragmented packets won't get by packet filters and firewalls that queue all IP fragments, such as the CONFIG_IP_ALWAYS_DEFRAG option in the Linux kernel, some networks
Manual page nmap(1) line 1332/2076 68% (press h for help or q for quit)
mouse pointer inside or press Ctrl+G.
Search
8:42 PM
1/27/2024

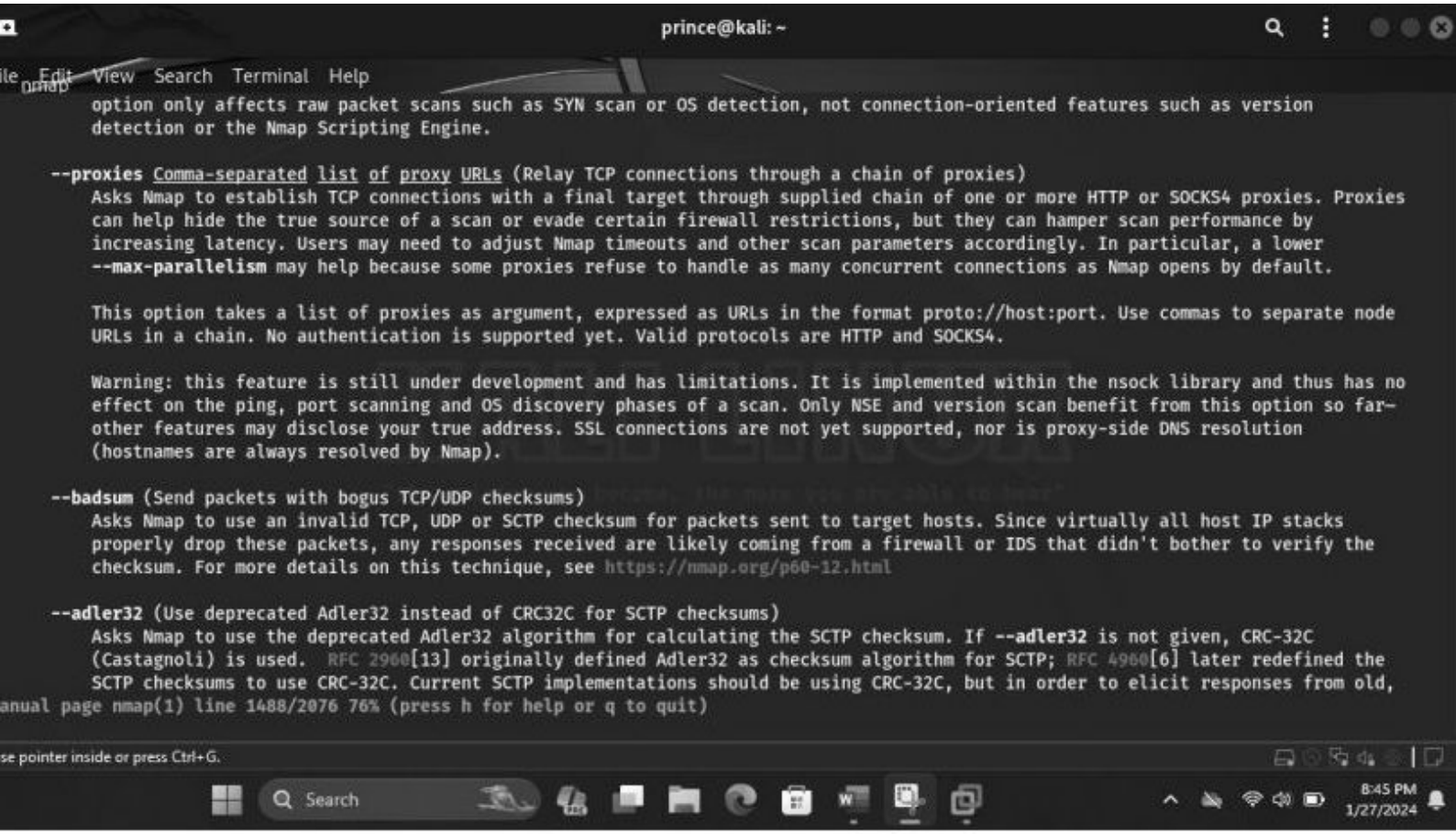

Help
Home
Kali Linux 2024
Applications Places
Jan 27 06:42
prince@kali: ~
File Edit View Search Terminal Help
nmap
can't afford the performance hit this causes and thus leave it disabled. Others can't enable this because fragments may take different routes into their networks. Some source systems defragment outgoing packets in the kernel. Linux with the iptables connection tracking module is one such example. Do a scan while a sniffer such as Wireshark is running to ensure that sent packets are fragmented. If your host OS is causing problems, try the --send-eth option to bypass the IP layer and send raw ethernet frames.
Fragmentation is only supported for Nmap's raw packet features, which includes TCP and UDP port scans (except connect scan and FTP bounce scan) and OS detection. Features such as version detection and the Nmap Scripting Engine generally don't support fragmentation because they rely on your host's TCP stack to communicate with target services.
-D decoy1[,decoy2][,ME][,...] (Cloak a scan with decoys)
Causes a decoy scan to be performed, which makes it appear to the remote host that the host(s) you specify as decoys are scanning the target network too. Thus their IDS might report 5–10 port scans from unique IP addresses, but they won't know which IP was scanning them and which were innocent decoys. While this can be defeated through router path tracing, response-dropping, and other active mechanisms, it is generally an effective technique for hiding your IP address.
Separate each decoy host with commas, and you can optionally use ME as one of the decoys to represent the position for your real IP address. If you put ME in the sixth position or later, some common port scan detectors (such as Solar Designer's excellent Scanlogd) are unlikely to show your IP address at all. If you don't use ME, Nmap will put you in a random position. You can also use RND to generate a random, non-reserved IP address, or RND:number to generate number addresses.
Note that the hosts you use as decoys should be up or you might accidentally SYN flood your targets. Also it will be pretty easy to determine which host is scanning if only one is actually up on the network. You might want to use IP addresses instead of names (so the decoy networks don't see you in their nameserver logs). Right now random IP address generation is only supported with IPv4


File Edit View Search Terminal Help
nmap
Decoys are used both in the initial host discovery scan (using ICMP, SYN, ACK, or whatever) and during the actual port scanning phase. Decoys are also used during remote OS detection (-O). Decoys do not work with version detection or TCP connect scan. When a scan delay is in effect, the delay is enforced between each batch of spoofed probes, not between each individual probe. Because decoys are sent as a batch all at once, they may temporarily violate congestion control limits.
It is worth noting that using too many decoys may slow your scan and potentially even make it less accurate. Also, some ISPs will filter out your spoofed packets, but many do not restrict spoofed IP packets at all.
-S IP Address (Spoof source address)
In some circumstances, Nmap may not be able to determine your source address (Nmap will tell you if this is the case). In this situation, use -S with the IP address of the interface you wish to send packets through.
Another possible use of this flag is to spoof the scan to make the targets think that someone else is scanning them. Imagine a company being repeatedly port scanned by a competitor! The -e option and -Pn are generally required for this sort of usage. Note that you usually won't receive reply packets back (they will be addressed to the IP you are spoofing), so Nmap won't produce useful reports.
-e interface (Use specified interface)
Tells Nmap what interface to send and receive packets on. Nmap should be able to detect this automatically, but it will tell you if it cannot.
--source-port portnumber; -g portnumber (Spoof source port number)
One surprisingly common misconfiguration is to trust traffic based only on the source port number. It is easy to understand how this comes about. An administrator will set up a shiny new firewall, only to be flooded with complaints from ungrateful users whose applications stopped working. In particular, DNS may be broken because the UDP DNS replies from external servers can no longer enter the network. FTP is another common example. In active FTP transfers, the remote server tries to establish a
Manual page nmap(1) line 1384/2076 71% (press h for help or q to quit)
the mouse pointer inside or press Ctrl+G.
Search
8:43 PM
1/27/2024


Workstation
s Help
Home
Kali Linux 2024
Applications Places
Jan 27 06:43
prince@kali: ~
File Edit View Search Terminal Help
nmap
connection back to the client to transfer the requested file.
Secure solutions to these problems exist, often in the form of application-level proxies or protocol-parsing firewall modules. Unfortunately there are also easier, insecure solutions. Noting that DNS replies come from port 53 and active FTP from port 20, many administrators have fallen into the trap of simply allowing incoming traffic from those ports. They often assume that no attacker would notice and exploit such firewall holes. In other cases, administrators consider this a short-term stop-gap measure until they can implement a more secure solution. Then they forget the security upgrade.
Overworked network administrators are not the only ones to fall into this trap. Numerous products have shipped with these insecure rules. Even Microsoft has been guilty. The IPsec filters that shipped with Windows 2000 and Windows XP contain an implicit rule that allows all TCP or UDP traffic from port 88 (Kerberos). In another well-known case, versions of the Zone Alarm personal firewall up to 2.1.25 allowed any incoming UDP packets with the source port 53 (DNS) or 67 (DHCP).
Nmap offers the -g and --source-port options (they are equivalent) to exploit these weaknesses. Simply provide a port number and Nmap will send packets from that port where possible. Most scanning operations that use raw sockets, including SYN and UDP scans, support the option completely. The option notably doesn't have an effect for any operations that use normal operating system sockets, including DNS requests, TCP connect scan, version detection, and script scanning. Setting the source port also doesn't work for OS detection, because Nmap must use different port numbers for certain OS detection tests to work properly.
--data hex string (Append custom binary data to sent packets)
This option lets you include binary data as payload in sent packets. hex string may be specified in any of the following formats: 0xAABBCCDDEEFF..., AABBCCDDEEFF... or \xAA\xBB\xCC\xDD\xEE\xFF.... Examples of use are --data 0xdeadbeef and --data \xCA\xFE\x09. Note that if you specify a number like 0x00ff no byte-order conversion is performed. Make sure you specify the information in the byte order expected by the receiver.
--data-string string (Append custom string to sent packets)
Manual page nmap(1) line 1410/2076 72% (press h for help or q to quit)

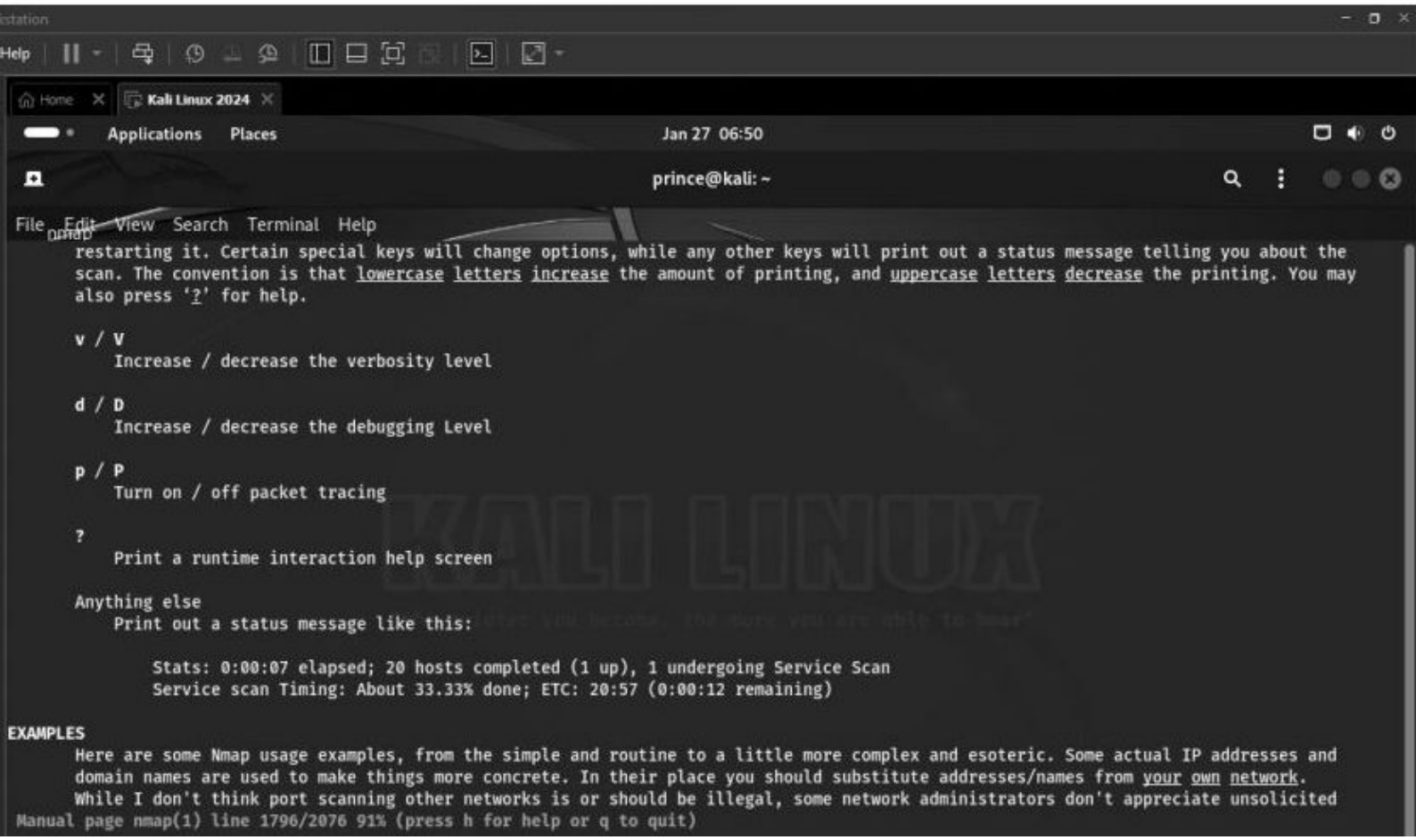

prince@kali: ~
Edit View Search Terminal Help
--data-string string (Append custom string to sent packets)
This option lets you include a regular string as payload in sent packets. string can contain any string. However, note that some characters may depend on your system's locale and the receiver may not see the same information. Also, make sure you enclose the string in double quotes and escape any special characters from the shell. Examples: --data-string "Scan conducted by Security Ops, extension 7192" or --data-string "Ph34r my l33t skills". Keep in mind that nobody is likely to actually see any comments left by this option unless they are carefully monitoring the network with a sniffer or custom IDS rules.
--data-length number (Append random data to sent packets)
Normally Nmap sends minimalist packets containing only a header. So its TCP packets are generally 40 bytes and ICMP echo requests are just 28. Some UDP ports and IP protocols get a custom payload by default. This option tells Nmap to append the given number of random bytes to most of the packets it sends, and not to use any protocol-specific payloads. (Use --data-length 0 for no random or protocol-specific payloads. OS detection (-O) packets are not affected because accuracy there requires probe consistency, but most pinging and portscan packets support this. It slows things down a little, but can make a scan slightly less conspicuous.
--ip-options RIS [route]|L [route]|T|U ... ; --ip-options hex string (Send packets with specified ip options)
The IP protocol[12] offers several options which may be placed in packet headers. Unlike the ubiquitous TCP options, IP options are rarely seen due to practicality and security concerns. In fact, many Internet routers block the most dangerous options such as source routing. Yet options can still be useful in some cases for determining and manipulating the network route to target machines. For example, you may be able to use the record route option to determine a path to a target even when more traditional traceroute-style approaches fail. Or if your packets are being dropped by a certain firewall, you may be able to specify a different route with the strict or loose source routing options.
The most powerful way to specify IP options is to simply pass in values as the argument to --ip-options. Precede each hex number with \x then the two digits. You may repeat certain characters by following them with an asterisk and then the number of times you wish them to repeat. For example, \x01\x07\x04\x00*36\x01 is a hex string containing 36 NUL bytes.
nual page nmap(1) line 1435/2076 74% (press h for help or q to quit)
pointer inside or press Ctrl+G.
Search
8:44 PM
1/27/2024

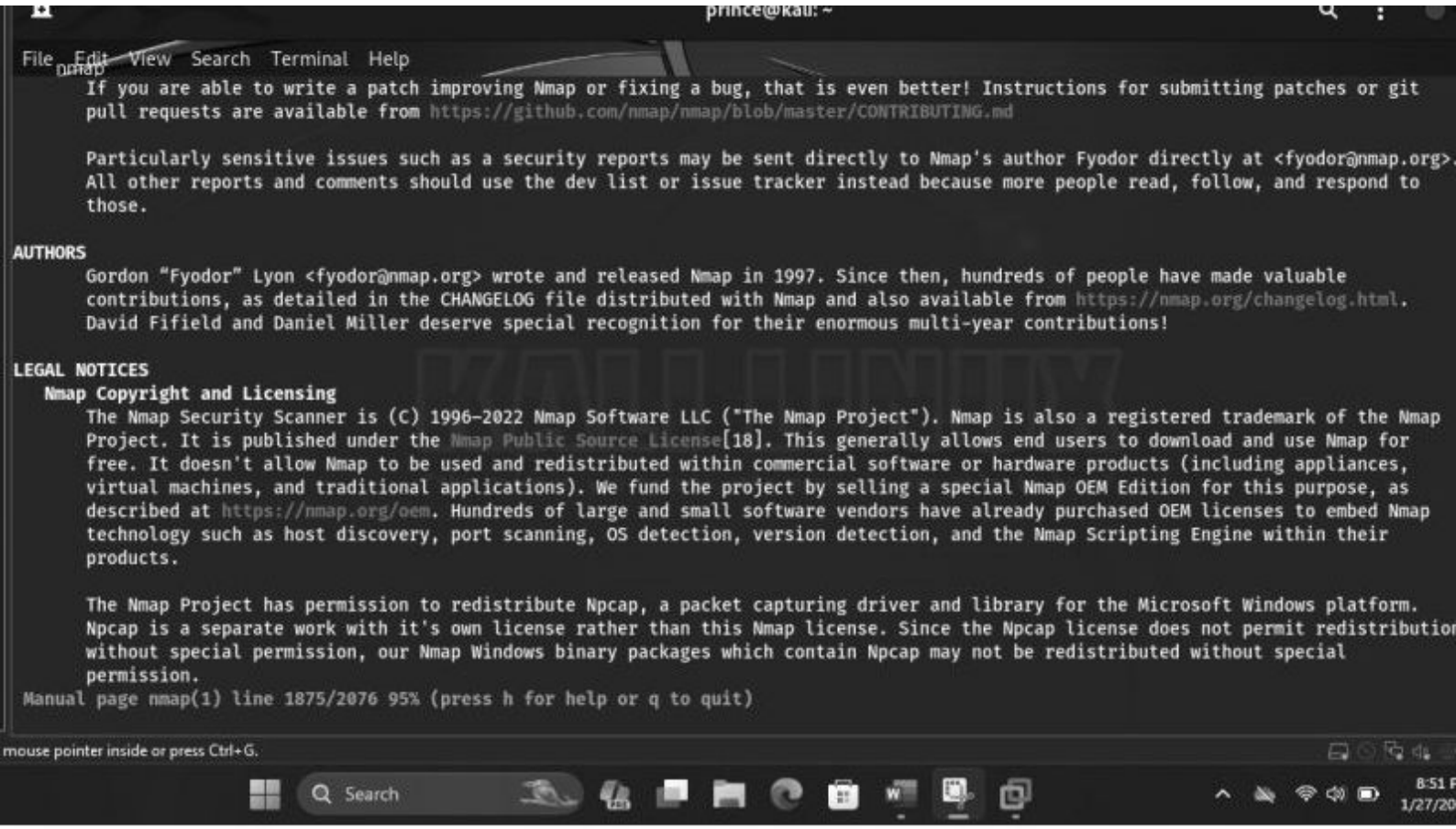

Home
Kali Linux 2024
Applications Places
Jan 27 06:44
prince@kali: ~
Edit View Search Terminal Help
Nmap also offers a shortcut mechanism for specifying options. Simply pass the letter R, T, or U to request record-route, record-timestamp, or both options together, respectively. Loose or strict source routing may be specified with an L or S followed by a space and then a space-separated list of IP addresses.
If you wish to see the options in packets sent and received, specify --packet-trace. For more information and examples of using IP options with Nmap, see https://seclists.org/nmap-dev/2006/q3/52.
--ttl value (Set IP time-to-live field)
Sets the IPv4 time-to-live field in sent packets to the given value.
--randomize-hosts (Randomize target host order)
Tells Nmap to shuffle each group of up to 16384 hosts before it scans them. This can make the scans less obvious to various network monitoring systems, especially when you combine it with slow timing options. If you want to randomize over larger group sizes, increase PING_GROUP_SZ in nmap.h and recompile. An alternative solution is to generate the target IP list with a list scan (-sL -n -oN filename), randomize it with a Perl script, then provide the whole list to Nmap with -iL.
--spoof-mac MAC address, prefix, or vendor name (Spoof MAC address)
Asks Nmap to use the given MAC address
for all of the raw ethernet frames it sends. This option implies --send-eth to ensure that Nmap actually sends ethernet-level packets. The MAC given can take several formats. If it is simply the number 0, Nmap chooses a completely random MAC address for the session. If the given string is an even number of hex digits (with the pairs optionally separated by a colon), Nmap will use those as the MAC. If fewer than 12 hex digits are provided, Nmap fills in the remainder of the six bytes with random values. If the argument isn't a zero or hex string, Nmap looks through nmap-mac-prefixes to find a vendor name containing the given string (it is case insensitive). If a match is found, Nmap uses the vendor's OUI (three-byte prefix) and fills out the remaining three bytes randomly. Valid --spoof-mac argument examples are Apple, 0, 01:02:03:04:05:06, deadbeefcafe, 0020F2, and Cisco. This
nual page nmap(1) line 1462/2076 75% (press h for help or q to quit)

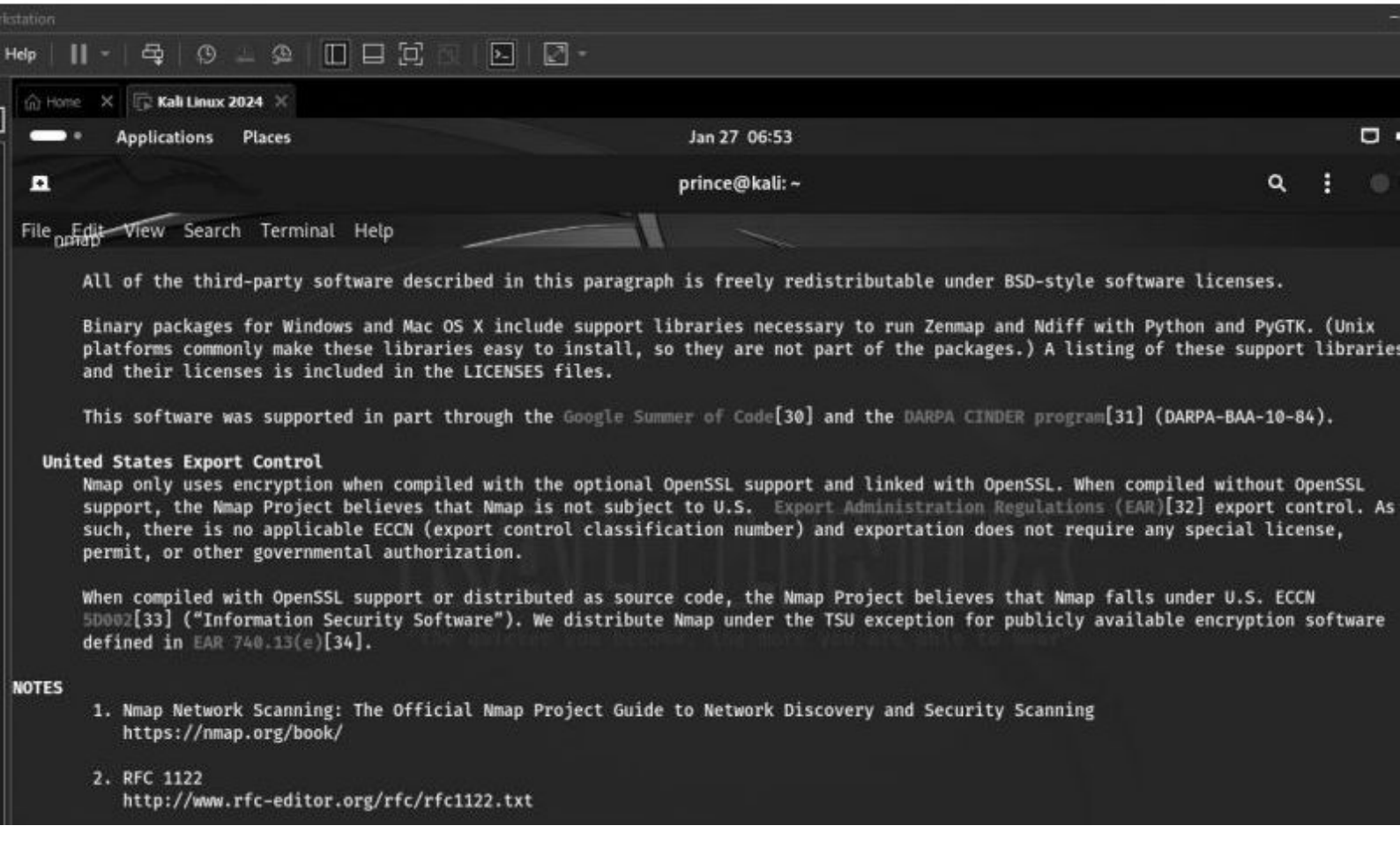

prince@kali: ~
File Edit View Search Terminal Help
option only affects raw packet scans such as SYN scan or OS detection, not connection-oriented features such as version detection or the Nmap Scripting Engine.
--proxies Comma-separated list of proxy URLs (Relay TCP connections through a chain of proxies)
Asks Nmap to establish TCP connections with a final target through supplied chain of one or more HTTP or SOCKS4 proxies. Proxies can help hide the true source of a scan or evade certain firewall restrictions, but they can hamper scan performance by increasing latency. Users may need to adjust Nmap timeouts and other scan parameters accordingly. In particular, a lower --max-parallelism may help because some proxies refuse to handle as many concurrent connections as Nmap opens by default.
This option takes a list of proxies as argument, expressed as URLs in the format proto://host:port. Use commas to separate node URLs in a chain. No authentication is supported yet. Valid protocols are HTTP and SOCKS4.
Warning: this feature is still under development and has limitations. It is implemented within the nsock library and thus has no effect on the ping, port scanning and OS discovery phases of a scan. Only NSE and version scan benefit from this option so far– other features may disclose your true address. SSL connections are not yet supported, nor is proxy-side DNS resolution (hostnames are always resolved by Nmap).
--badsum (Send packets with bogus TCP/UDP checksums)
Asks Nmap to use an invalid TCP, UDP or SCTP checksum for packets sent to target hosts. Since virtually all host IP stacks properly drop these packets, any responses received are likely coming from a firewall or IDS that didn't bother to verify the checksum. For more details on this technique, see https://nmap.org/p60-12.html
--adler32 (Use deprecated Adler32 instead of CRC32C for SCTP checksums)
Asks Nmap to use the deprecated Adler32 algorithm for calculating the SCTP checksum. If --adler32 is not given, CRC-32C (Castagnoli) is used. RFC 2960[13] originally defined Adler32 as checksum algorithm for SCTP; RFC 4960[6] later redefined the SCTP checksums to use CRC-32C. Current SCTP implementations should be using CRC-32C, but in order to elicit responses from old,
anual page nmap(1) line 1488/2076 76% (press h for help or q to quit)
se pointer inside or press Ctrl+G.
Search
8:45 PM
1/27/2024

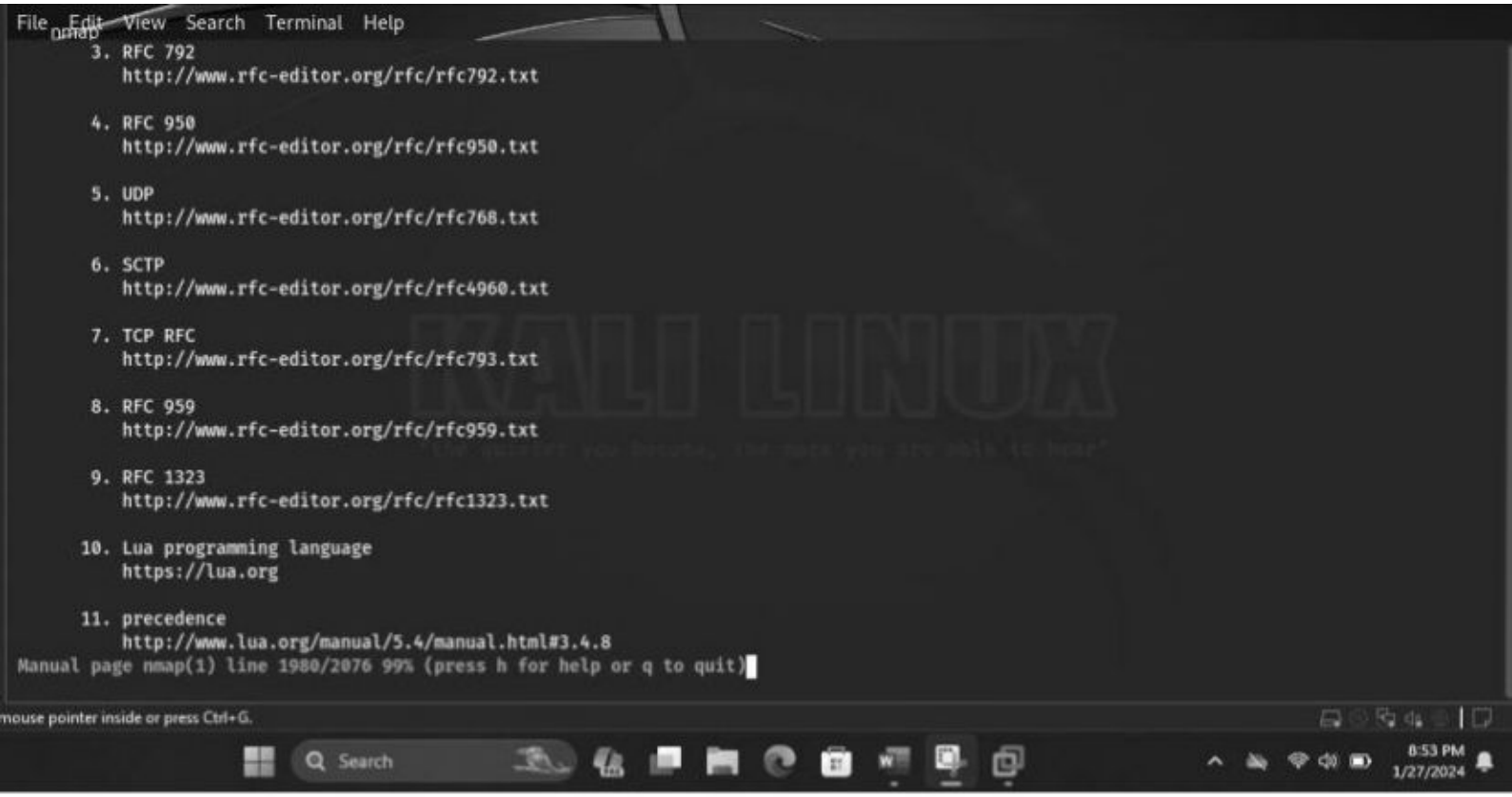

Home
Kali Linux 2024
Applications
Places
Jan 27 06:45
prince@kali: ~
File Edit View Search Terminal Help
legacy SCTP implementations, it may be preferable to use Adler32.
TPUT
Any security tool is only as useful as the output it generates. Complex tests and algorithms are of little value if they aren't presented in an organized and comprehensible fashion. Given the number of ways Nmap is used by people and other software, no single format can please everyone. So Nmap offers several formats, including the interactive mode for humans to read directly and XML for easy parsing by software.
In addition to offering different output formats, Nmap provides options for controlling the verbosity of output as well as debugging messages. Output types may be sent to standard output or to named files, which Nmap can append to or clobber. Output files may also be used to resume aborted scans.
Nmap makes output available in five different formats. The default is called interactive output, and it is sent to standard output (stdout). There is also normal output, which is similar to interactive except that it displays less runtime information and warnings since it is expected to be analyzed after the scan completes rather than interactively.
XML output is one of the most important output types, as it can be converted to HTML, easily parsed by programs such as Nmap graphical user interfaces, or imported into databases.
The two remaining output types are the simple grepable output which includes most information for a target host on a single line, and sCRiPt KiDDi3 0utPUt for users who consider themselves |<-r4d.
While interactive output is the default and has no associated command-line options, the other four format options use the same syntax. They take one argument, which is the filename that results should be stored in. Multiple formats may be specified, but each format may only be specified once. For example, you may wish to save normal output for your own review while saving XML of the same scan for programmatic analysis. You might do this with the options -oX myscan.xml -oN myscan.nmap. While this chapter uses the
anual page nmap(1) line 1514/2076 78% (press h for help or q to quit)


prince@kali: ~
File Edit View Search Terminal Help
nmap
-6 (Enable IPv6 scanning)
Nmap has IPv6 support for its most popular features. Ping scanning, port scanning, version detection, and the Nmap Scripting Engine all support IPv6. The command syntax is the same as usual except that you also add the -6 option. Of course, you must use IPv6 syntax if you specify an address rather than a hostname. An address might look like 3ffe:7501:4819:2000:210:f3ff:fe03:14d0, so hostnames are recommended. The output looks the same as usual, with the IPv6 address on the "interesting ports" line being the only IPv6 giveaway.
While IPv6 hasn't exactly taken the world by storm, it gets significant use in some (usually Asian) countries and most modern operating systems support it. To use Nmap with IPv6, both the source and target of your scan must be configured for IPv6. If your ISP (like most of them) does not allocate IPv6 addresses to you, free tunnel brokers are widely available and work fine with Nmap. I use the free IPv6 tunnel broker service at http://www.tunnelbroker.net. Other tunnel brokers are listed at Wikipedia[17]. 6to4 tunnels are another popular, free approach.
On Windows, raw-socket IPv6 scans are supported only on ethernet devices (not tunnels), and only on Windows Vista and later. Use the --unprivileged option in other situations.
-A (Aggressive scan options)
This option enables additional advanced and aggressive options. Presently this enables OS detection (-O), version scanning (-sV), script scanning (-sC) and traceroute (--traceroute). More features may be added in the future. The point is to enable a comprehensive set of scan options without people having to remember a large set of flags. However, because script scanning with the default set is considered intrusive, you should not use -A against target networks without permission. This option only enables features, and not timing options (such as -T4) or verbosity options (-v) that you might want as well. Options which require privileges (e.g. root access) such as OS detection and traceroute will only be enabled if those privileges are available.
--datadir directoryname (Specify custom Nmap data file location)
Manual page nmap(1) line 1720/2076 88% (press h for help or q to quit)
mouse pointer inside or press Ctrl+G.
Search
8:49 PM
1/27/2024


Help
Home
Kali Linux 2024
Applications
Places
Jan 27 06:50
prince@kali: ~
File Edit View Search Terminal Help
nmap
restarting it. Certain special keys will change options, while any other keys will print out a status message telling you about the scan. The convention is that lowercase letters increase the amount of printing, and uppercase letters decrease the printing. You may also press '?' for help.
v / V
Increase / decrease the verbosity level
d / D
Increase / decrease the debugging Level
p / P
Turn on / off packet tracing
?
Print a runtime interaction help screen
Anything else
Print out a status message like this:
Stats: 0:00:07 elapsed; 20 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 33.33% done; ETC: 20:57 (0:00:12 remaining)
EXAMPLES
Here are some Nmap usage examples, from the simple and routine to a little more complex and esoteric. Some actual IP addresses and domain names are used to make things more concrete. In their place you should substitute addresses/names from your own network. While I don't think port scanning other networks is or should be illegal, some network administrators don't appreciate unsolicited
Manual page nmap(1) line 1796/2076 91% (press h for help or q to quit)


prince@kali: ~
File Edit View Search Terminal Help
nmap
If you are able to write a patch improving Nmap or fixing a bug, that is even better! Instructions for submitting patches or git pull requests are available from https://github.com/nmap/nmap/blob/master/CONTRIBUTING.md
Particularly sensitive issues such as a security reports may be sent directly to Nmap's author Fyodor directly at <fyodor@nmap.org>. All other reports and comments should use the dev list or issue tracker instead because more people read, follow, and respond to those.
AUTHORS
Gordon "Fyodor" Lyon <fyodor@nmap.org> wrote and released Nmap in 1997. Since then, hundreds of people have made valuable contributions, as detailed in the CHANGELOG file distributed with Nmap and also available from https://nmap.org/changelog.html. David Fifield and Daniel Miller deserve special recognition for their enormous multi-year contributions!
LEGAL NOTICES
Nmap Copyright and Licensing
The Nmap Security Scanner is (C) 1996-2022 Nmap Software LLC ("The Nmap Project"). Nmap is also a registered trademark of the Nmap Project. It is published under the Nmap Public Source License[18]. This generally allows end users to download and use Nmap for free. It doesn't allow Nmap to be used and redistributed within commercial software or hardware products (including appliances, virtual machines, and traditional applications). We fund the project by selling a special Nmap OEM Edition for this purpose, as described at https://nmap.org/oem. Hundreds of large and small software vendors have already purchased OEM licenses to embed Nmap technology such as host discovery, port scanning, OS detection, version detection, and the Nmap Scripting Engine within their products.
The Nmap Project has permission to redistribute Npcap, a packet capturing driver and library for the Microsoft Windows platform. Npcap is a separate work with it's own license rather than this Nmap license. Since the Npcap license does not permit redistribution without special permission, our Nmap Windows binary packages which contain Npcap may not be redistributed without special permission.
Manual page nmap(1) line 1875/2076 95% (press h for help or q to quit)
mouse pointer inside or press Ctrl+G.
Search
8:51 P
1/27/20


Help
Home
Kali Linux 2024
Applications
Places
Jan 27 06:53
prince@kali: ~
File Edit View Search Terminal Help
nmap
All of the third-party software described in this paragraph is freely redistributable under BSD-style software licenses.
Binary packages for Windows and Mac OS X include support libraries necessary to run Zenmap and Ndiff with Python and PyGTK. (Unix platforms commonly make these libraries easy to install, so they are not part of the packages.) A listing of these support libraries and their licenses is included in the LICENSES files.
This software was supported in part through the Google Summer of Code[30] and the DARPA CINDER program[31] (DARPA-BAA-10-84).
United States Export Control
Nmap only uses encryption when compiled with the optional OpenSSL support and linked with OpenSSL. When compiled without OpenSSL support, the Nmap Project believes that Nmap is not subject to U.S. Export Administration Regulations (EAR)[32] export control. As such, there is no applicable ECCN (export control classification number) and exportation does not require any special license, permit, or other governmental authorization.
When compiled with OpenSSL support or distributed as source code, the Nmap Project believes that Nmap falls under U.S. ECCN 5D002[33] ("Information Security Software"). We distribute Nmap under the TSU exception for publicly available encryption software defined in EAR 740.13(e)[34].
NOTES
1. Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning
https://nmap.org/book/
2. RFC 1122
http://www.rfc-editor.org/rfc/rfc1122.txt


File Edit View Search Terminal Help
nmap
3. RFC 792
http://www.rfc-editor.org/rfc/rfc792.txt
4. RFC 950
http://www.rfc-editor.org/rfc/rfc950.txt
5. UDP
http://www.rfc-editor.org/rfc/rfc768.txt
6. SCTP
http://www.rfc-editor.org/rfc/rfc4960.txt
7. TCP RFC
http://www.rfc-editor.org/rfc/rfc793.txt
8. RFC 959
http://www.rfc-editor.org/rfc/rfc959.txt
9. RFC 1323
http://www.rfc-editor.org/rfc/rfc1323.txt
10. Lua programming language
https://lua.org
11. precedence
http://www.lua.org/manual/5.4/manual.html#3.4.8
Manual page nmap(1) line 1980/2076 99% (press h for help or q to quit)
mouse pointer inside or press Ctrl+G.
Search
8:53 PM
1/27/2024


Help
Home
Kali Linux 2024
Applications
Places
Jan 27 06:54
prince@kali: ~
File Edit View Search Terminal Help
nmap
12. IP protocol
http://www.rfc-editor.org/rfc/rfc791.txt
13. RFC 2960
http://www.rfc-editor.org/rfc/rfc2960.txt
14. Nmap::Scanner
http://sourceforge.net/projects/nmap-scanner/
15. Nmap::Parser
http://nmapparser.wordpress.com/
16. xsltproc
http://xmlsoft.org/XSLT/
17. listed at Wikipedia
http://en.wikipedia.org/wiki/List_of_IPv6_tunnel_brokers
18. Nmap Public Source License
https://nmap.org/npsl
19. Nmap OEM license
https://nmap.org/oem/
20. Creative Commons Attribution License
http://creativecommons.org/licenses/by/3.0/
Manual page nmap(1) line 2007/2076 99% (press h for help or q to quit)

```
┌──(prince㉿kali)-[~]
└─$ sudo nmap -sT -p 80,443 10.7.1.0/24
[sudo] password for prince:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-30 22:36 PST
Nmap scan report for 10.7.1.0
Host is up (0.0013s latency).

PORT    STATE    SERVICE
80/tcp  filtered http
443/tcp filtered https

Nmap scan report for 10.7.1.1
Host is up (0.00054s latency).

PORT    STATE    SERVICE
80/tcp  filtered http
443/tcp filtered https
```

```
prince@kali: ~

PORT    STATE    SERVICE
80/tcp  filtered http
443/tcp filtered https

Nmap scan report for 10.7.1.2
Host is up (0.00014s latency).

PORT    STATE    SERVICE
80/tcp  filtered http
443/tcp filtered https

Nmap scan report for 10.7.1.3
Host is up (0.00040s latency).

PORT    STATE    SERVICE
80/tcp  filtered http
443/tcp filtered https

Nmap scan report for 10.7.1.4
```

```
┌──(prince㉿kali)-[~]
└─$ sudo nmap -sT 10.7.1.226
[sudo] password for prince:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-30 23:05 PST
Nmap scan report for 10.7.1.226
Host is up (0.0018s latency).
All 1000 scanned ports on 10.7.1.226 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 22.50 seconds
```

Output

```
┌──(prince㉿kali)-[~]
└─$ sudo nmap -sS 10.7.1.226
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-07 00:21 PST
Nmap scan report for 10.7.1.226
Host is up (0.00077s latency).
All 1000 scanned ports on 10.7.1.226 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 53.48 seconds
```

Output

```
┌──(prince㉿kali)-[~]
└─$ sudo nmap 10.7.1.226
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-07 00:27 PST
Nmap scan report for 10.7.1.226
Host is up (0.012s latency).
All 1000 scanned ports on 10.7.1.226 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 51.54 seconds
```