Know Your Tools: Nmap

Anthony Kosednar

Slides….

- https://github.com/akosednar/presentations

My Background

- Education
  - Number Theory, Field Theory, Cryptography and Statistics @ ASU
  - Aerospace Engineering - Astronautics @ ASU
  - DHS Cybersecurity Training for Industrial Control Systems – SCADA, DCS, PLC, Embedded Control
  - GIAC Exploit Researcher and Advanced Penetration Tester - GXPN

- Career Highlights
  - Built a natural air-cooled datacenter in under 6 months
  - Lead encryption for a large fortune 50 financial institution
  - Prevented cyber attacks for large events (Arizona Cardinals, Arizona Board of Tourism, NFL, Super Bowl XLIX, Fiesta Bowl)

Nmap

What is it?

- At it’s basic level:
    - Network Scanner - Used to discover hosts and services on a network
    - Free & Open-source
    - Multi-platform (Linux, Windows, FreeBSD, OpenBSD, Solaris, Mac and more..)
    - Popular! Lots of tutorials and talks online

How do you install it?

- Kali
  - Included!
- Centos, Fedora & Redhat
  - yum install nmap
- Ubuntu & Debian
  - sudo apt-get install nmap
- For more options… https://nmap.org/download.html

What does the nmap package come with?

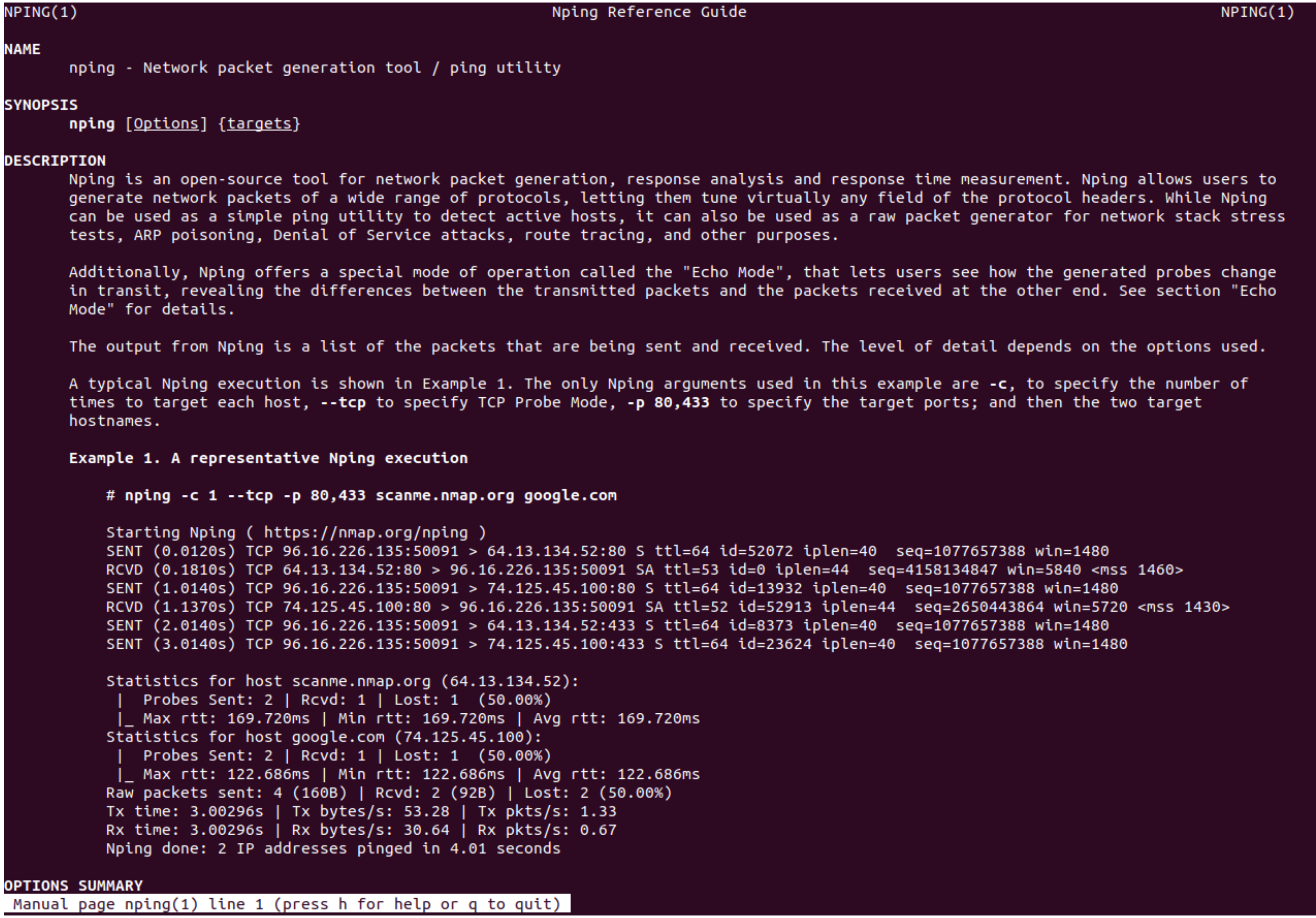
- nping – Packet generation/ping utility
- ncat – Concatenate and redirect sockets
- nmap – Network Mapper

nping!

- Packet generation/ping utility
- Helps us analyze packets in transit (firewall rules, packet corruption, NATs)

nping!

- > man nping



```
NPING(1)                                    Nping Reference Guide                                    NPING(1)

NAME
       nping - Network packet generation tool / ping utility

SYNOPSIS
       nping [Options] {targets}

DESCRIPTION
       Nping is an open-source tool for network packet generation, response analysis and response time measurement. Nping allows users to
       generate network packets of a wide range of protocols, letting them tune virtually any field of the protocol headers. While Nping
       can be used as a simple ping utility to detect active hosts, it can also be used as a raw packet generator for network stack stress
       tests, ARP poisoning, Denial of Service attacks, route tracing, and other purposes.

       Additionally, Nping offers a special mode of operation called the "Echo Mode", that lets users see how the generated probes change
       in transit, revealing the differences between the transmitted packets and the packets received at the other end. See section "Echo
       Mode" for details.

       The output from Nping is a list of the packets that are being sent and received. The level of detail depends on the options used.

       A typical Nping execution is shown in Example 1. The only Nping arguments used in this example are -c, to specify the number of
       times to target each host, --tcp to specify TCP Probe Mode, -p 80,433 to specify the target ports; and then the two target
       hostnames.

       Example 1. A representative Nping execution

           # nping -c 1 --tcp -p 80,433 scanme.nmap.org google.com

           Starting Nping ( https://nmap.org/nping )
           SENT (0.0120s) TCP 96.16.226.135:50091 > 64.13.134.52:80 S ttl=64 id=52072 iplen=40  seq=1077657388 win=1480
           RCVD (0.1810s) TCP 64.13.134.52:80 > 96.16.226.135:50091 SA ttl=53 id=0 iplen=44  seq=4158134847 win=5840 <mss 1460>
           SENT (1.0140s) TCP 96.16.226.135:50091 > 74.125.45.100:80 S ttl=64 id=13932 iplen=40  seq=1077657388 win=1480
           RCVD (1.1370s) TCP 74.125.45.100:80 > 96.16.226.135:50091 SA ttl=52 id=52913 iplen=44  seq=2650443864 win=5720 <mss 1430>
           SENT (2.0140s) TCP 96.16.226.135:50091 > 64.13.134.52:433 S ttl=64 id=8373 iplen=40  seq=1077657388 win=1480
           SENT (3.0140s) TCP 96.16.226.135:50091 > 74.125.45.100:433 S ttl=64 id=23624 iplen=40  seq=1077657388 win=1480

           Statistics for host scanme.nmap.org (64.13.134.52):
            |  Probes Sent: 2 | Rcvd: 1 | Lost: 1  (50.00%)
            |_ Max rtt: 169.720ms | Min rtt: 169.720ms | Avg rtt: 169.720ms
           Statistics for host google.com (74.125.45.100):
            |  Probes Sent: 2 | Rcvd: 1 | Lost: 1  (50.00%)
            |_ Max rtt: 122.686ms | Min rtt: 122.686ms | Avg rtt: 122.686ms
           Raw packets sent: 4 (160B) | Rcvd: 2 (92B) | Lost: 2 (50.00%)
           Tx time: 3.00296s | Tx bytes/s: 53.28 | Tx pkts/s: 1.33
           Rx time: 3.00296s | Rx bytes/s: 30.64 | Rx pkts/s: 0.67
           Nping done: 2 IP addresses pinged in 4.01 seconds

OPTIONS SUMMARY
 Manual page nping(1) line 1 (press h for help or q to quit)
```

nping!

- > man nping
  - Options
  - Targets

nping!

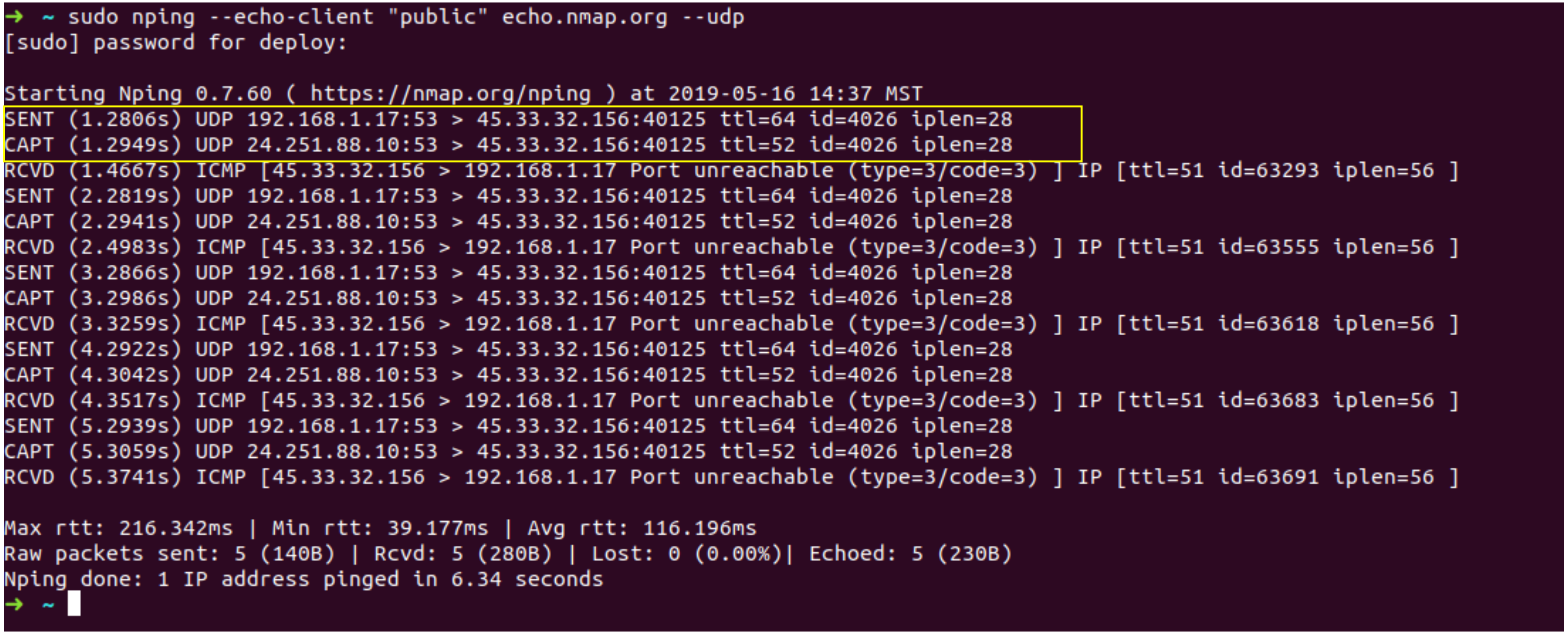
- > man nping
  - Options
    - Probe Mode (tcp, udp, icmp, arp, traceroute, “unprivledged tcp”)
    - Protocol options for all these probe modes
    - Echo mode!
      - Can setup a client and a server to see how packets change between two points

nping!

- > man nping
  - Options
    - Probe Mode (tcp, udp, icmp, arp, traceroute, “unprivledged tcp”)
    - Protocol options for all these probe modes
    - Echo mode!
      - Can setup a client and a server to see how packets change between two points

Example: nping – NAT Detection

We know there's a NAT between us and the other device because the sent and received packets differ.



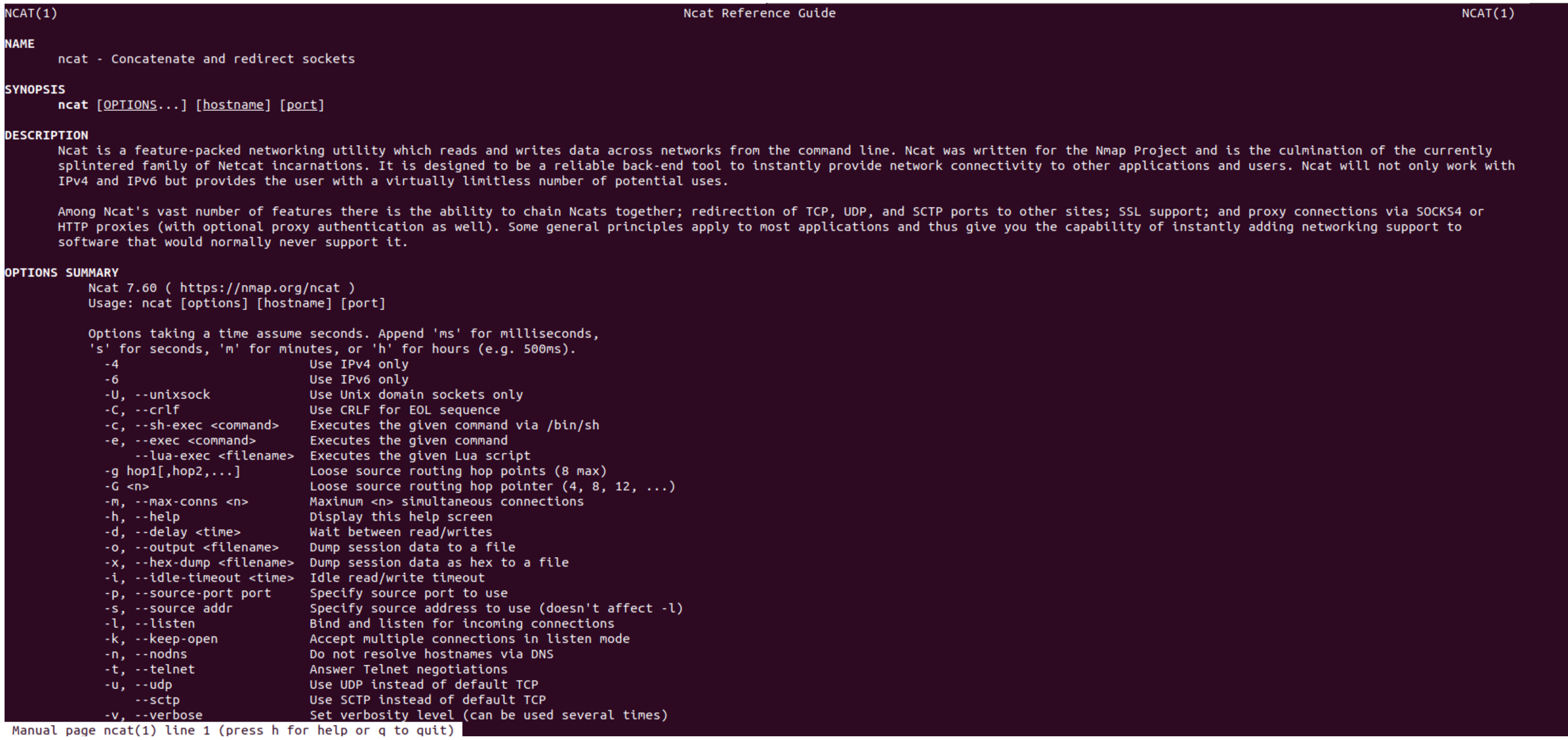
```
➜  ~ sudo nping --echo-client "public" echo.nmap.org --udp
[sudo] password for deploy:

Starting Nping 0.7.60 ( https://nmap.org/nping ) at 2019-05-16 14:37 MST
SENT (1.2806s) UDP 192.168.1.17:53 > 45.33.32.156:40125 ttl=64 id=4026 iplen=28
CAPT (1.2949s) UDP 24.251.88.10:53 > 45.33.32.156:40125 ttl=52 id=4026 iplen=28
RCVD (1.4667s) ICMP [45.33.32.156 > 192.168.1.17 Port unreachable (type=3/code=3) ] IP [ttl=51 id=63293 iplen=56 ]
SENT (2.2819s) UDP 192.168.1.17:53 > 45.33.32.156:40125 ttl=64 id=4026 iplen=28
CAPT (2.2941s) UDP 24.251.88.10:53 > 45.33.32.156:40125 ttl=52 id=4026 iplen=28
RCVD (2.4983s) ICMP [45.33.32.156 > 192.168.1.17 Port unreachable (type=3/code=3) ] IP [ttl=51 id=63555 iplen=56 ]
SENT (3.2866s) UDP 192.168.1.17:53 > 45.33.32.156:40125 ttl=64 id=4026 iplen=28
CAPT (3.2986s) UDP 24.251.88.10:53 > 45.33.32.156:40125 ttl=52 id=4026 iplen=28
RCVD (3.3259s) ICMP [45.33.32.156 > 192.168.1.17 Port unreachable (type=3/code=3) ] IP [ttl=51 id=63618 iplen=56 ]
SENT (4.2922s) UDP 192.168.1.17:53 > 45.33.32.156:40125 ttl=64 id=4026 iplen=28
CAPT (4.3042s) UDP 24.251.88.10:53 > 45.33.32.156:40125 ttl=52 id=4026 iplen=28
RCVD (4.3517s) ICMP [45.33.32.156 > 192.168.1.17 Port unreachable (type=3/code=3) ] IP [ttl=51 id=63683 iplen=56 ]
SENT (5.2939s) UDP 192.168.1.17:53 > 45.33.32.156:40125 ttl=64 id=4026 iplen=28
CAPT (5.3059s) UDP 24.251.88.10:53 > 45.33.32.156:40125 ttl=52 id=4026 iplen=28
RCVD (5.3741s) ICMP [45.33.32.156 > 192.168.1.17 Port unreachable (type=3/code=3) ] IP [ttl=51 id=63691 iplen=56 ]

Max rtt: 216.342ms | Min rtt: 39.177ms | Avg rtt: 116.196ms
Raw packets sent: 5 (140B) | Rcvd: 5 (280B) | Lost: 0 (0.00%)| Echoed: 5 (230B)
Nping done: 1 IP address pinged in 6.34 seconds
➜  ~
```

ncat!

- General purpose network connector
  - Act as a client to interact with a networked server
  - Act as a server for a network client
  - Redirect/proxy traffic
  - Even can act as a network gateway for the execution of system commands!

ncat!

- > man ncat



```
NCAT(1)                                              Ncat Reference Guide                                              NCAT(1)

NAME
       ncat - Concatenate and redirect sockets

SYNOPSIS
       ncat [OPTIONS...] [hostname] [port]

DESCRIPTION
       Ncat is a feature-packed networking utility which reads and writes data across networks from the command line. Ncat was written for the Nmap Project and is the culmination of the currently
       splintered family of Netcat incarnations. It is designed to be a reliable back-end tool to instantly provide network connectivity to other applications and users. Ncat will not only work with
       IPv4 and IPv6 but provides the user with a virtually limitless number of potential uses.

       Among Ncat's vast number of features there is the ability to chain Ncats together; redirection of TCP, UDP, and SCTP ports to other sites; SSL support; and proxy connections via SOCKS4 or
       HTTP proxies (with optional proxy authentication as well). Some general principles apply to most applications and thus give you the capability of instantly adding networking support to
       software that would normally never support it.

OPTIONS SUMMARY
           Ncat 7.60 ( https://nmap.org/ncat )
           Usage: ncat [options] [hostname] [port]

           Options taking a time assume seconds. Append 'ms' for milliseconds,
           's' for seconds, 'm' for minutes, or 'h' for hours (e.g. 500ms).
             -4                         Use IPv4 only
             -6                         Use IPv6 only
             -U, --unixsock             Use Unix domain sockets only
             -C, --crlf                 Use CRLF for EOL sequence
             -c, --sh-exec <command>    Executes the given command via /bin/sh
             -e, --exec <command>       Executes the given command
                 --lua-exec <filename>  Executes the given Lua script
             -g hop1[,hop2,...]         Loose source routing hop points (8 max)
             -G <n>                     Loose source routing hop pointer (4, 8, 12, ...)
             -m, --max-conns <n>        Maximum <n> simultaneous connections
             -h, --help                 Display this help screen
             -d, --delay <time>         Wait between read/writes
             -o, --output <filename>    Dump session data to a file
             -x, --hex-dump <filename>  Dump session data as hex to a file
             -i, --idle-timeout <time>  Idle read/write timeout
             -p, --source-port port     Specify source port to use
             -s, --source addr          Specify source address to use (doesn't affect -l)
             -l, --listen               Bind and listen for incoming connections
             -k, --keep-open            Accept multiple connections in listen mode
             -n, --nodns                Do not resolve hostnames via DNS
             -t, --telnet               Answer Telnet negotiations
             -u, --udp                  Use UDP instead of default TCP
                 --sctp                 Use SCTP instead of default TCP
             -v, --verbose              Set verbosity level (can be used several times)
 Manual page ncat(1) line 1 (press h for help or q to quit)
```

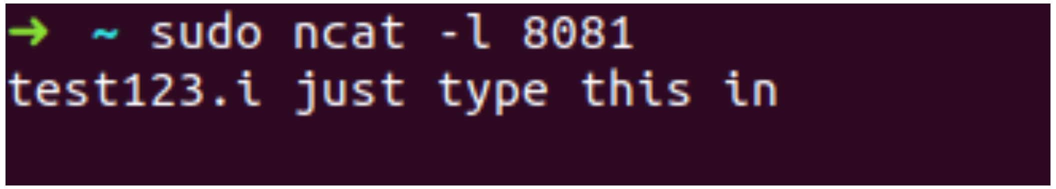
ncat!

- > man ncat
  - Options
  - Hostname
  - Port

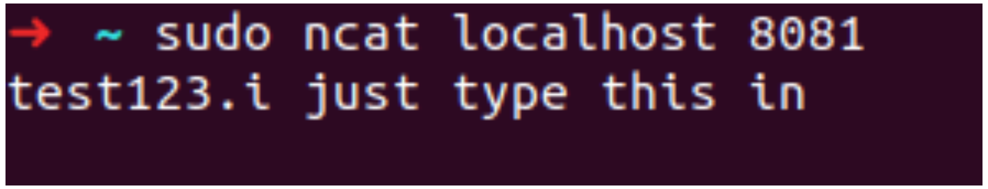
ncat!

- > man ncat
    - Options
        - Operates in one of two primary modes – connect or listen
        - Other modes act as a special case of the two (HTTP Proxy Server)
            - ….which we are going to save for a future talk ;)

Example: ncat – Connect & Listen Mode

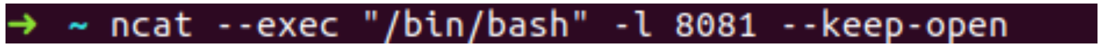


```
→  ~ sudo ncat -l 8081
test123.i just type this in
```

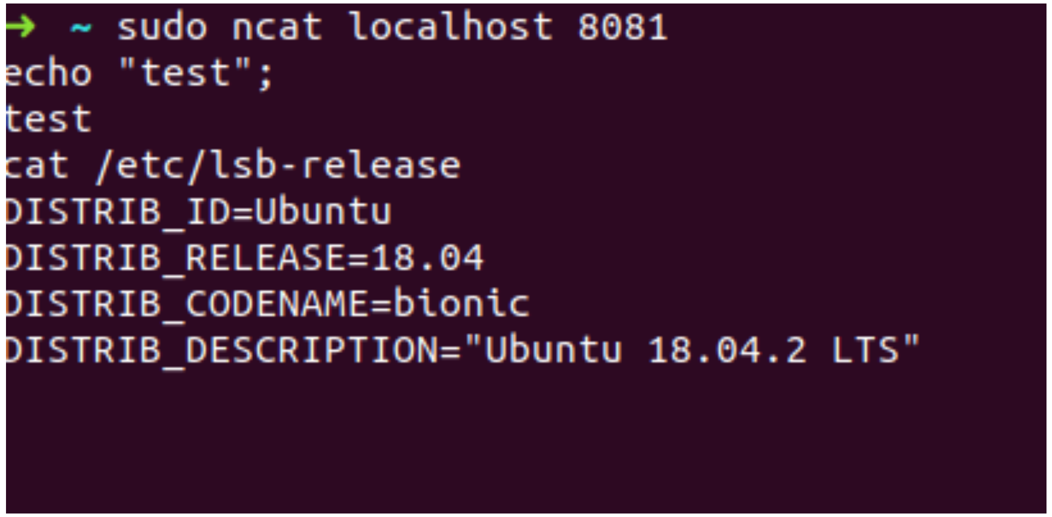


```
→  ~ sudo ncat localhost 8081
test123.i just type this in
```

Example: ncat – Remote Terminal (exec mode)



```
➜  ~ ncat --exec "/bin/bash" -l 8081 --keep-open
```

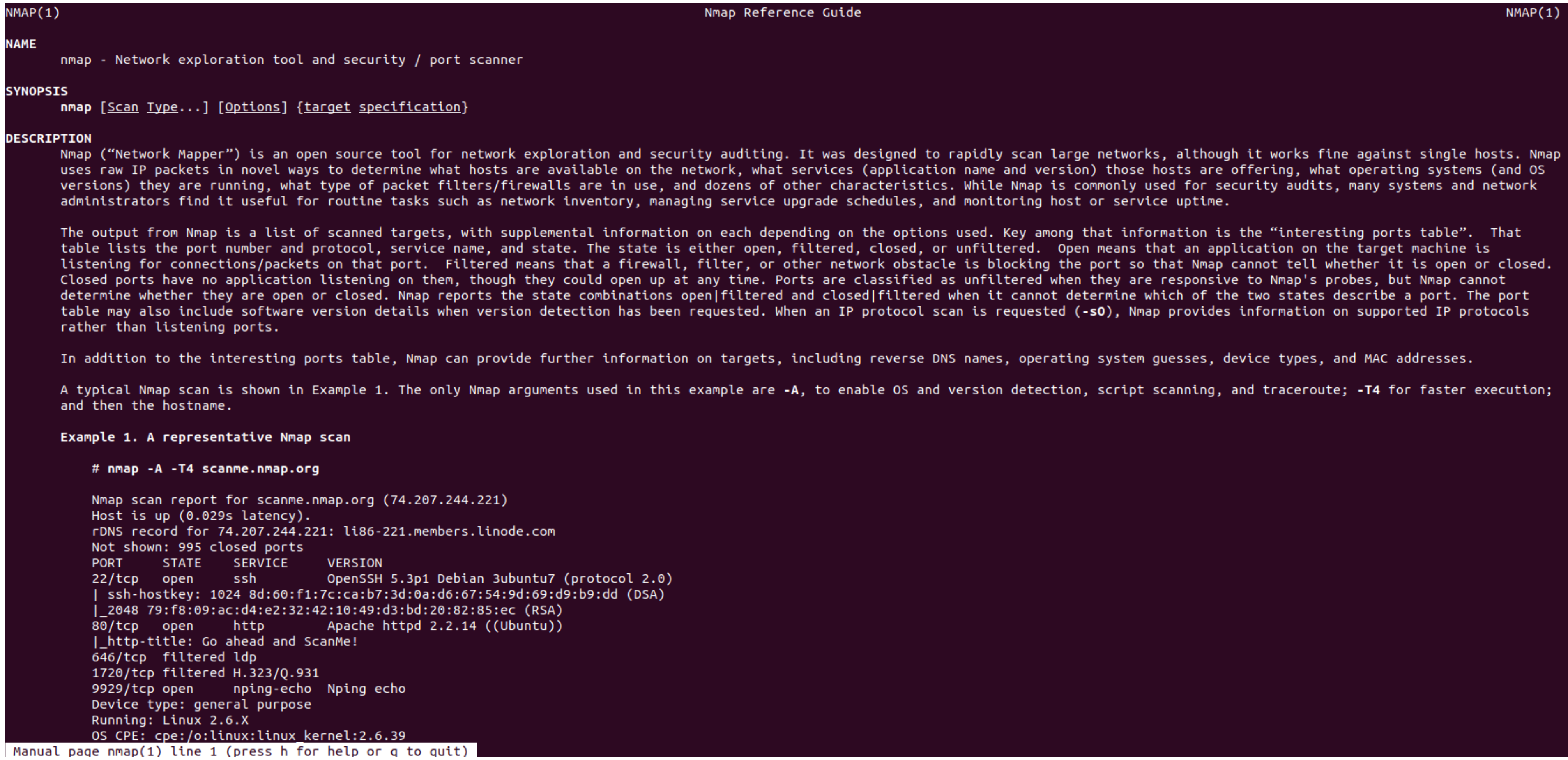


```
➜  ~ sudo ncat localhost 8081
echo "test";
test
cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=18.04
DISTRIB_CODENAME=bionic
DISTRIB_DESCRIPTION="Ubuntu 18.04.2 LTS"
```

Finally Nmap time!

nmap!

- > man 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.

       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 1 (press h for help or q to quit)
```

nmap!

- > nmap [SCAN TYPE] [OPTIONS] [TARGET!]

What’s our goal when using Nmap?

- Discover the hosts
- Scan the ports we are interested in to see if they are online
- Detect their services
- Detect the OS based upon these ports and services

- …while evading firewalls & IDSs

First we need a target!

What do we wan to scan?

> nmap [SCAN TYPE] [OPTIONS] [TARGET!]

nmap - Targets

- Can pass hostnames, IP addresses, networks (lists of them too!)
  - Example.com, example.com/24, 10.0.0.1, 10.0.0-255.1-254

- -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

nmap – Port Specification and Scan Order

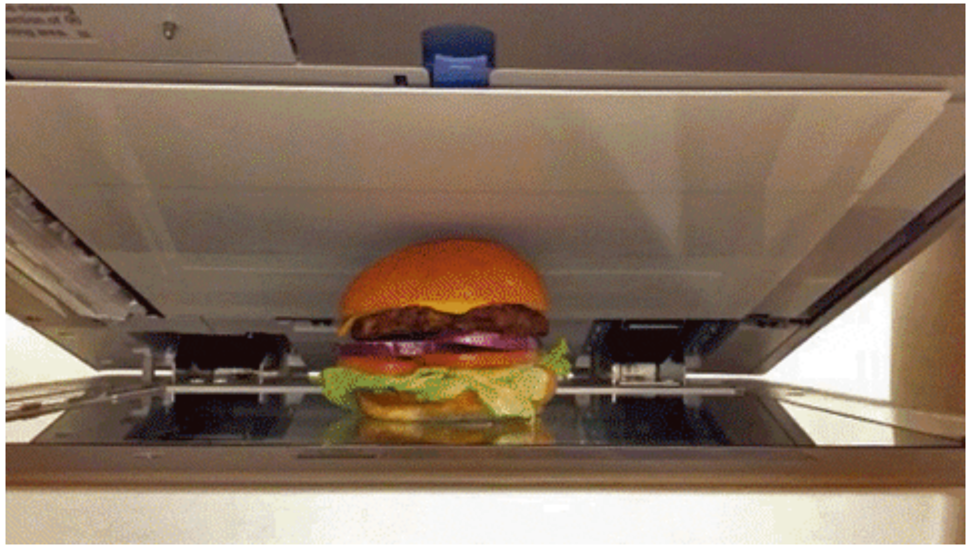
- -p <port ranges>: Only scan specified ports
  - Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9
- --exclude-ports <port ranges>: Exclude the specified ports from scanning
- -F: Fast mode - Scan fewer ports than the default scan
- -r: Scan ports consecutively - don't randomize
- --top-ports <number>: Scan <number> most common ports
- --port-ratio <ratio>: Scan ports more common than <ratio>

Next scan types

…and their options

How do we want to scan it?

```
> nmap [SCAN TYPE] [OPTIONS] [TARGET!]
```



nmap – Host Discovery Option

- Nmap can attempt to discover active hosts within the target to scan!

nmap – Host Discovery Option

- List Scan - simply list targets to scan
- Ping Scan - disable port scan
- TCP SYN/ACK, UDP or SCTP discovery to given ports
- ICMP echo, timestamp, and netmask request discovery probes
- IP Protocol Ping
- Disable DNS resolution
- Trace hop path to each host

nmap – Host Discovery Option

- -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

nmap – Service and Version Detection Option

- Probes open ports to determine service/version info
- Uses internal database that contains proves for querying various services and matching their responses to a service name

nmap – Service and Version Detection Option

- -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)

nmap – OS Detection Option

- Uses a series of techniques to detect the OS
- Also…
  - Does TCP sequence predictability classification (forged TCP connections)
    - Useful for exploiting source-IP based trust (ex: firewalls)
  - Guesses the target's uptime using TCP timestamp option

nmap – OS Detection Option

- -O: Enable OS detection
- --osscan-limit: Limit OS detection to promising targets
- --osscan-guess: Guess OS more aggressively

We have problem

All this is very noisy and can take too long!

We need better!

…stealth and performance

nmap – Firewall/IDS Evasion and Spoofing

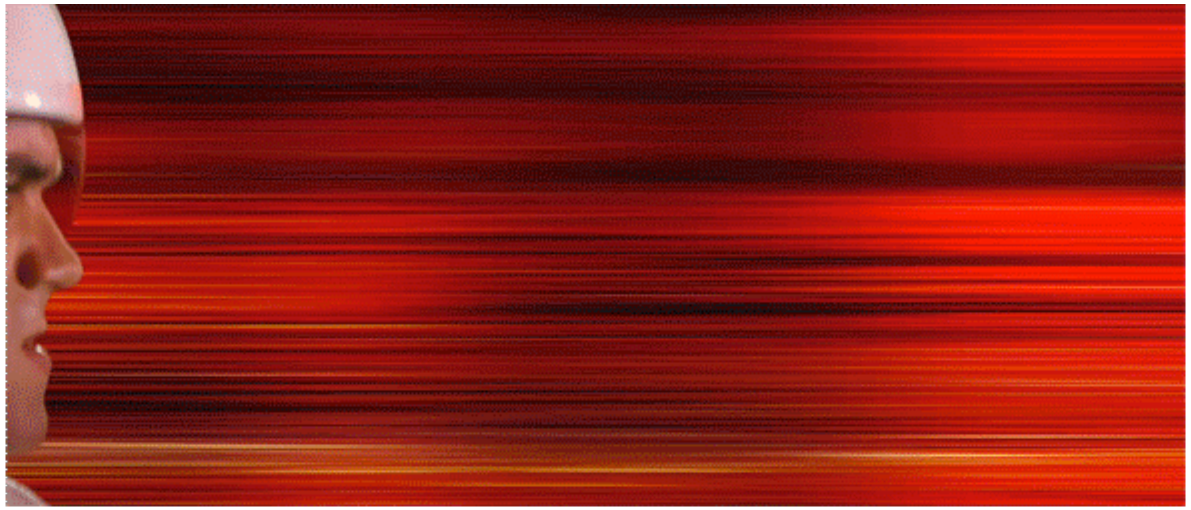
- Most modern systems ship with nmap scan detection techniques
- Getting around this is really its own talk
  - But….we will go through the tools nmap gives us

nmap – 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
- --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

nmap – Timing and Performance

- It's hard to scan a lot of systems
- Nmap gives us some options to increase performance



nmap – Timing and Performance

- -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

So that’s the important stuff!

Where are the scripts?

But….scripts!

Let me do my own things.

nmap – Scripting Engine (NSE)

- Allows the writing of simple scripts to automate networking tasks!
- On Kali, Ubuntu, and Debian they are located at /usr/share/nmap/scripts

nmap – Scripting Engine (NSE)

- Script types/phases:
  - Prerule – Run before any scan phase
  - Host – Run during the normal host discovery, port scanning, version detection and OS detection phase
  - Service – Run against specific services discovered to be listening
  - Postrule – Run after nmap has scanned all the target

nmap – Scripting Engine (NSE)

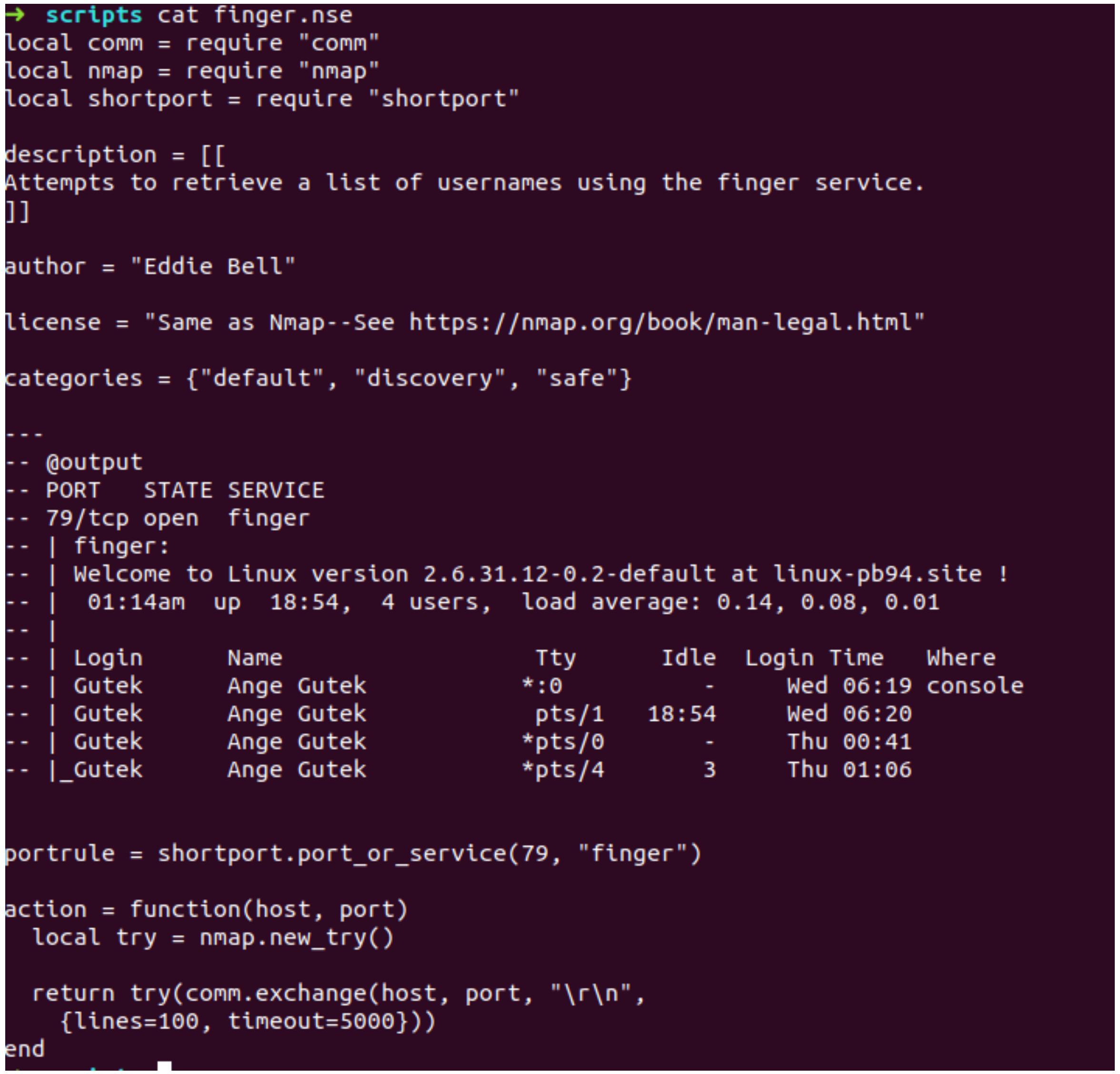
- Script categories:
    - Auth – credentials or bypassing them
    - Broadcast – host discovery via broadcast techniques
    - Brute – Brute force attacks
    - Default – scripts that run by default
    - Discovery – Discover network information (ex: SNMP querying, html-title inspection)
    - Dos – Denial of Service tests
    - Exploit – Aim to actively attempt to exploit vulnerabilities
    - External – Query an external db for more information (ex: whois-ip)
    - Fuzzer – Sends unexpected or randomized fields in packets to try to detect bugs
    - Intrusive – Likely to crash the system or cause issues (often bandwidth or cpu time)
    - Malware – Test if the platform has malware or backdoors
    - Safe – Scripts designed not to crash or cause issues
    - Version - Version detection scripts
    - Vuln – Check for known vulnerabilities

nmap – Scripting Engine (NSE)

- -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.

What does a NSE script look like?

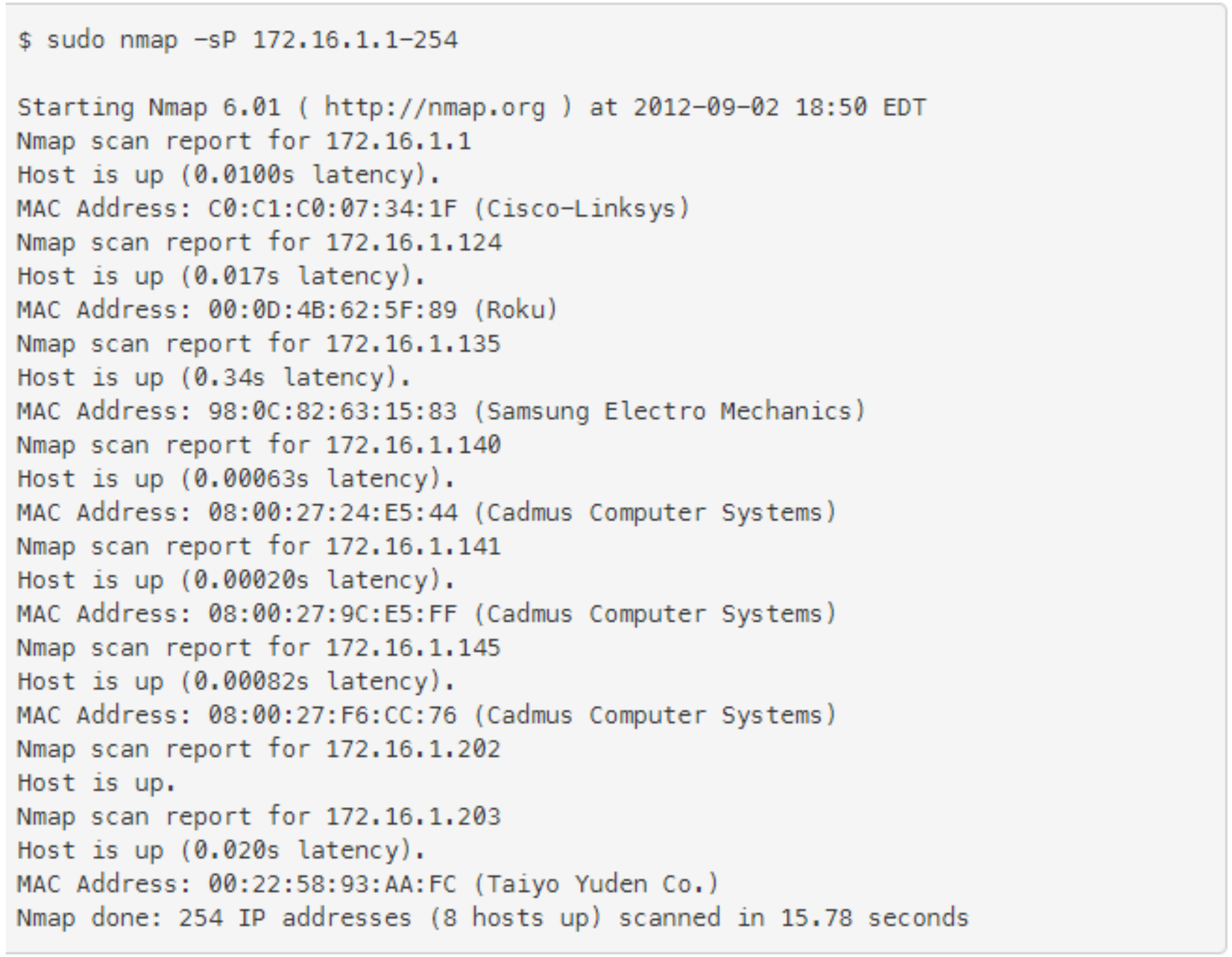
nmap – Scripting Engine (NSE) - finger.nse



```
→ scripts cat finger.nse
local comm = require "comm"
local nmap = require "nmap"
local shortport = require "shortport"

description = [[
Attempts to retrieve a list of usernames using the finger service.
]]

author = "Eddie Bell"

license = "Same as Nmap--See https://nmap.org/book/man-legal.html"

categories = {"default", "discovery", "safe"}

---
-- @output
-- PORT   STATE SERVICE
-- 79/tcp open  finger
-- | finger:
-- | Welcome to Linux version 2.6.31.12-0.2-default at linux-pb94.site !
-- |  01:14am  up  18:54,  4 users,  load average: 0.14, 0.08, 0.01
-- |
-- | Login      Name                  Tty      Idle  Login Time   Where
-- | Gutek      Ange Gutek           *:0          -     Wed 06:19 console
-- | Gutek      Ange Gutek            pts/1   18:54     Wed 06:20
-- | Gutek      Ange Gutek           *pts/0       -     Thu 00:41
-- |_Gutek      Ange Gutek           *pts/4       3     Thu 01:06


portrule = shortport.port_or_service(79, "finger")

action = function(host, port)
  local try = nmap.new_try()

  return try(comm.exchange(host, port, "\r\n",
    {lines=100, timeout=5000}))
end
```

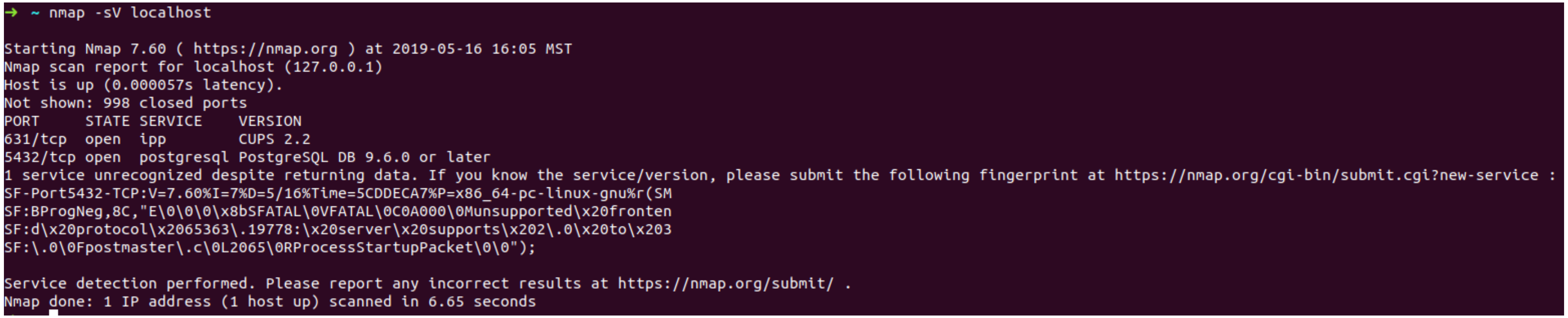
More Examples of Nmap!

Example: nmap – Host Discovery



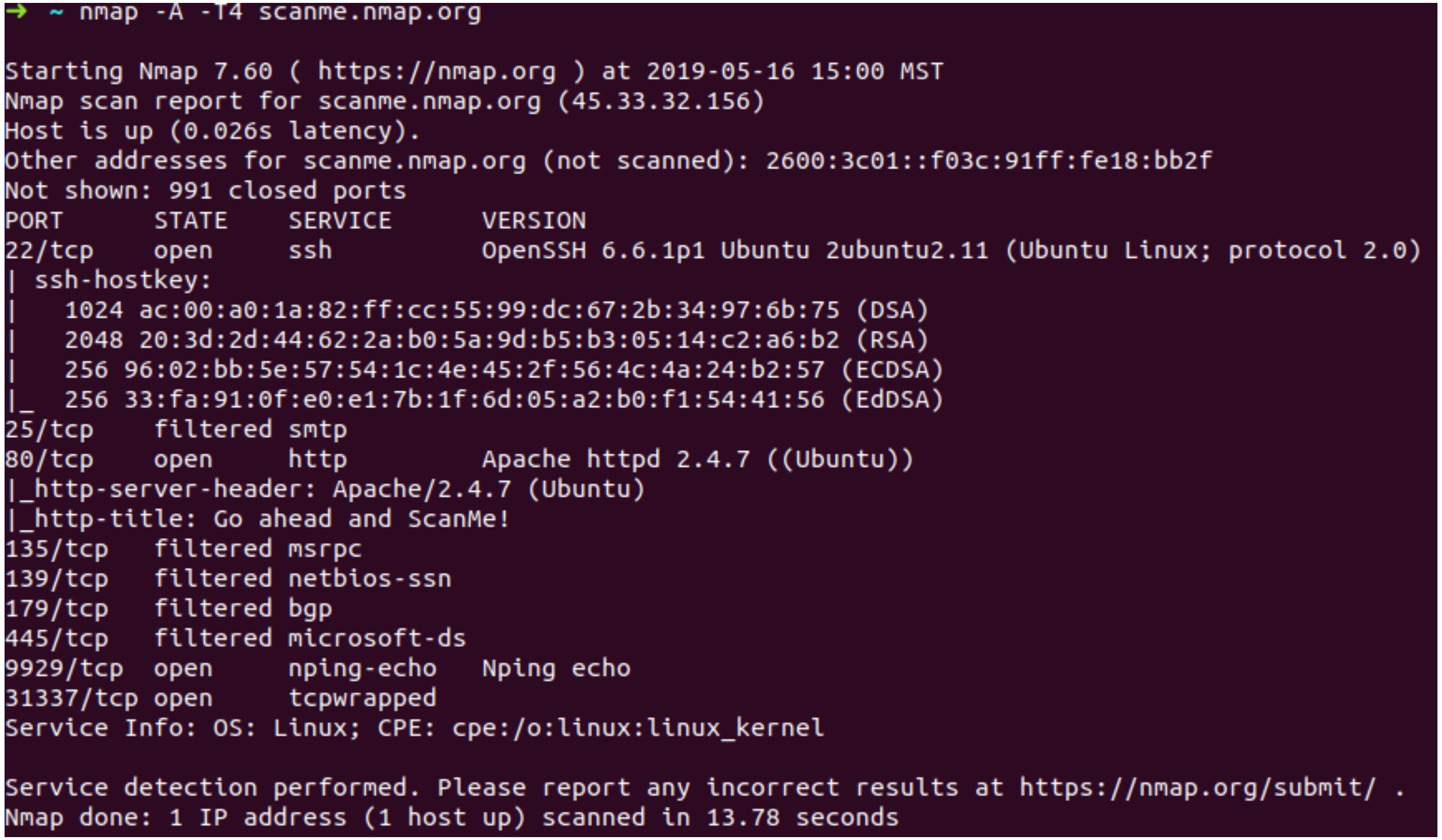
```
$ sudo nmap -sP 172.16.1.1-254

Starting Nmap 6.01 ( http://nmap.org ) at 2012-09-02 18:50 EDT
Nmap scan report for 172.16.1.1
Host is up (0.0100s latency).
MAC Address: C0:C1:C0:07:34:1F (Cisco-Linksys)
Nmap scan report for 172.16.1.124
Host is up (0.017s latency).
MAC Address: 00:0D:4B:62:5F:89 (Roku)
Nmap scan report for 172.16.1.135
Host is up (0.34s latency).
MAC Address: 98:0C:82:63:15:83 (Samsung Electro Mechanics)
Nmap scan report for 172.16.1.140
Host is up (0.00063s latency).
MAC Address: 08:00:27:24:E5:44 (Cadmus Computer Systems)
Nmap scan report for 172.16.1.141
Host is up (0.00020s latency).
MAC Address: 08:00:27:9C:E5:FF (Cadmus Computer Systems)
Nmap scan report for 172.16.1.145
Host is up (0.00082s latency).
MAC Address: 08:00:27:F6:CC:76 (Cadmus Computer Systems)
Nmap scan report for 172.16.1.202
Host is up.
Nmap scan report for 172.16.1.203
Host is up (0.020s latency).
MAC Address: 00:22:58:93:AA:FC (Taiyo Yuden Co.)
Nmap done: 254 IP addresses (8 hosts up) scanned in 15.78 seconds
```

Example: nmap – Service & Version Detection



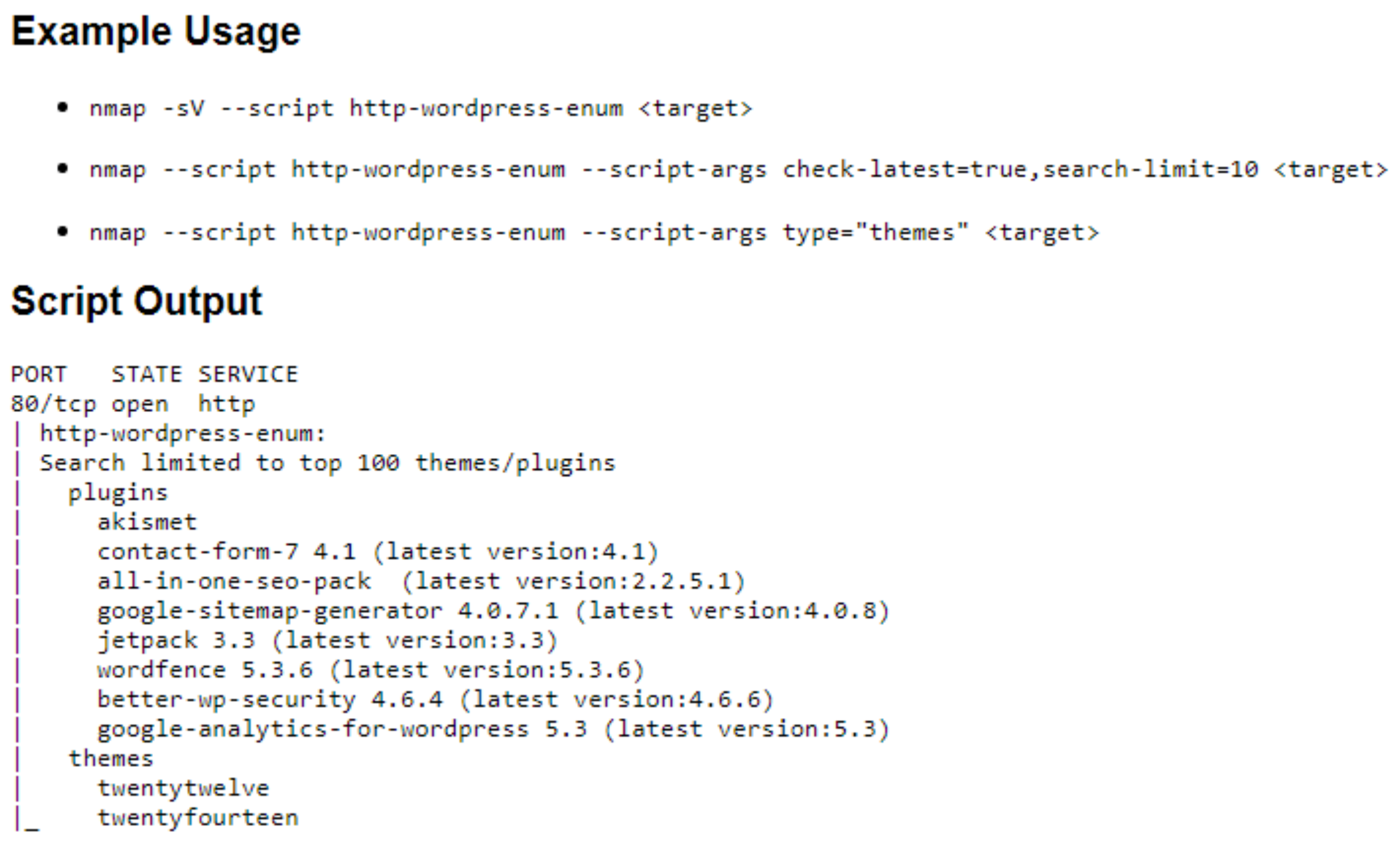
```
→ ~ nmap -sV localhost

Starting Nmap 7.60 ( https://nmap.org ) at 2019-05-16 16:05 MST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000057s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE    VERSION
631/tcp  open  ipp        CUPS 2.2
5432/tcp open  postgresql PostgreSQL DB 9.6.0 or later
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port5432-TCP:V=7.60%I=7%D=5/16%Time=5CDDECA7%P=x86_64-pc-linux-gnu%r(SM
SF:BProgNeg,8C,"E\0\0\0\x8bSFATAL\0VFATAL\0C0A000\0Munsupported\x20fronten
SF:d\x20protocol\x2065363\.19778:\x20server\x20supports\x202\.0\x20to\x203
SF:\.0\0Fpostmaster\.c\0L2065\0RProcessStartupPacket\0\0");

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.65 seconds
```

Example: nmap – OS Detection



```
➜ ~ nmap -A -T4 scanme.nmap.org

Starting Nmap 7.60 ( https://nmap.org ) at 2019-05-16 15:00 MST
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.026s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 991 closed ports
PORT      STATE    SERVICE     VERSION
22/tcp    open     ssh         OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.11 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 ac:00:a0:1a:82:ff:cc:55:99:dc:67:2b:34:97:6b:75 (DSA)
|   2048 20:3d:2d:44:62:2a:b0:5a:9d:b5:b3:05:14:c2:a6:b2 (RSA)
|   256 96:02:bb:5e:57:54:1c:4e:45:2f:56:4c:4a:24:b2:57 (ECDSA)
|_  256 33:fa:91:0f:e0:e1:7b:1f:6d:05:a2:b0:f1:54:41:56 (EdDSA)
25/tcp    filtered smtp
80/tcp    open     http        Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: Go ahead and ScanMe!
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
179/tcp   filtered bgp
445/tcp   filtered microsoft-ds
9929/tcp  open     nping-echo  Nping echo
31337/tcp open     tcpwrapped
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.78 seconds
```

Example: nmap – NSE: Wordpress Scanning



Example Usage

- `nmap -sV --script http-wordpress-enum <target>`
- `nmap --script http-wordpress-enum --script-args check-latest=true,search-limit=10 <target>`
- `nmap --script http-wordpress-enum --script-args type="themes" <target>`

Script Output

```
PORT   STATE SERVICE
80/tcp open  http
| http-wordpress-enum:
| Search limited to top 100 themes/plugins
|   plugins
|     akismet
|     contact-form-7 4.1 (latest version:4.1)
|     all-in-one-seo-pack  (latest version:2.2.5.1)
|     google-sitemap-generator 4.0.7.1 (latest version:4.0.8)
|     jetpack 3.3 (latest version:3.3)
|     wordfence 5.3.6 (latest version:5.3.6)
|     better-wp-security 4.6.4 (latest version:4.6.6)
|     google-analytics-for-wordpress 5.3 (latest version:5.3)
|   themes
|     twentytwelve
|_    twentyfourteen
```

Questions?

Anthony.Kosednar@gmail.com