www.hackingarticles.in

RUSTSCAN

SPEED UP YOUR SCAN

iGNITE Technologies

www.ignitetechnologies.in

Contents

A Detailed Guide on RustScan

In the realm of cybersecurity, network scanning tools play a vital role in reconnaissance and vulnerability assessment. Among the array of options available, Rustscan has emerged as a formidable contender, offering speed, efficiency, and versatility that distinguish it from traditional tools like Nmap.

What sets Rustscan apart?

Rustscan is an open-source network scanner developed in the Rust programming language. Its lightweight design, optimized algorithms, and user-friendly interface make it a preferred choice for both penetration testers and security professionals.

Advantages of Rustscan over Nmap:

Speed: It is renowned for its rapid scanning capabilities. Its multithreaded architecture and optimized algorithms enable it to scan large networks significantly faster than traditional scanners like Nmap.

Efficiency: It prioritizes efficiency, utilizing resources intelligently and minimizing overhead. This ensures that scanning tasks are completed swiftly without excessive resource consumption.

Ease of Use: With its intuitive interface and simplified command structure, Rustscan is accessible to users of all levels of expertise. Its design minimizes the learning curve typically associated with network scanning tools.

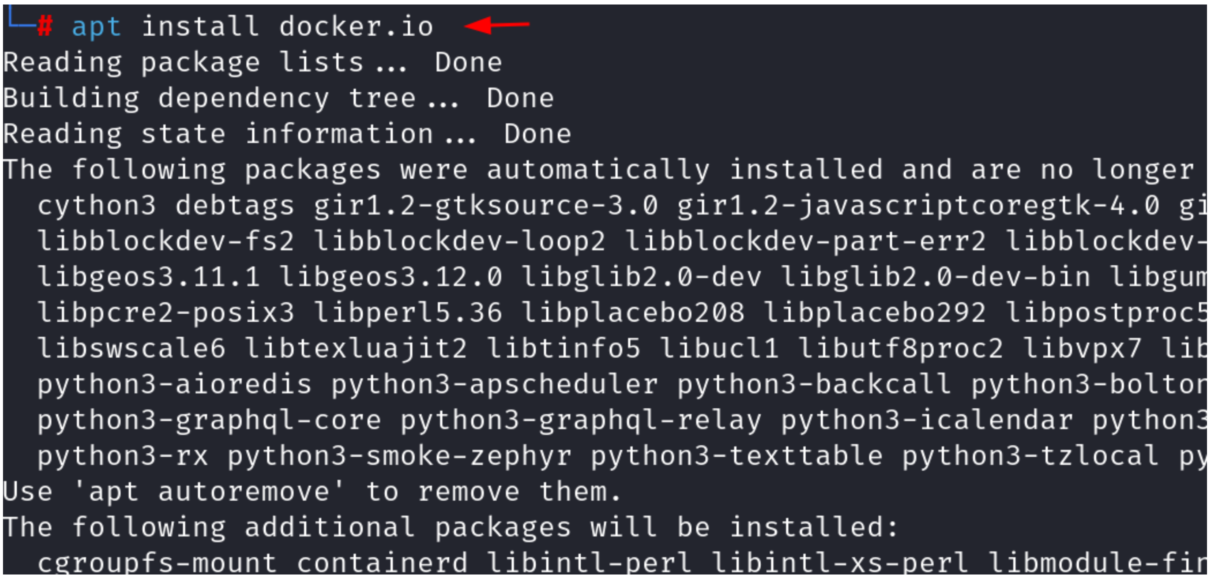
Versatility: It offers a broad range of features and customization options, allowing users to tailor their scans to specific requirements. Whether performing basic port scanning or comprehensive service enumeration, Rustscan delivers.

Usage (Docker)

Rustscan can run by pulling an image using docker. The installation guide is available here https://github.com/RustScan/RustScan/wiki/Installation-Guide

To install docker use the command:

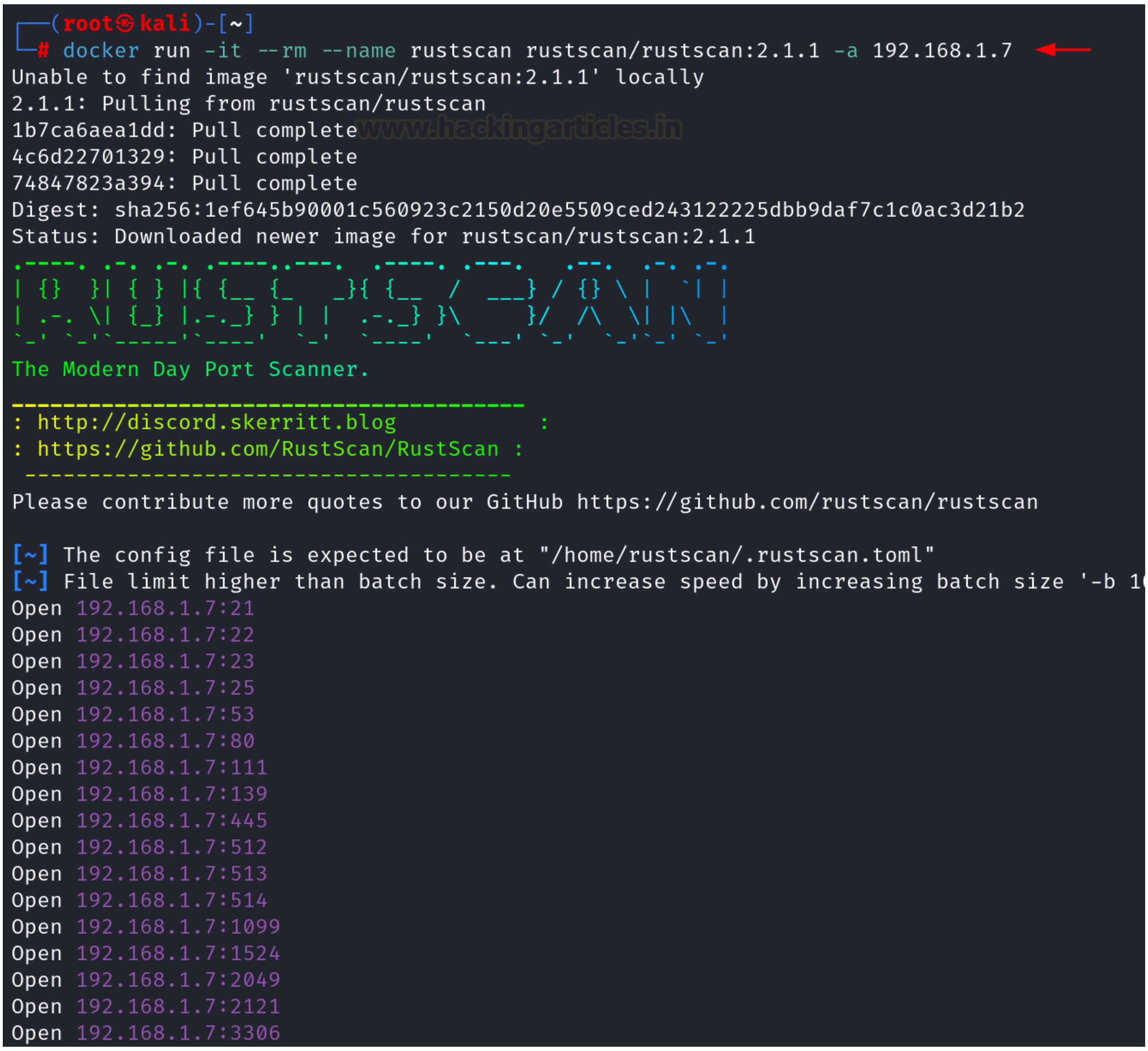
```
apt install docker.io
```



```
└─# apt install docker.io
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following packages were automatically installed and are no longer
  cython3 debtags gir1.2-gtksource-3.0 gir1.2-javascriptcoregtk-4.0 gi
  libblockdev-fs2 libblockdev-loop2 libblockdev-part-err2 libblockdev-
  libgeos3.11.1 libgeos3.12.0 libglib2.0-dev libglib2.0-dev-bin libgum
  libpcre2-posix3 libperl5.36 libplacebo208 libplacebo292 libpostproc5
  libswscale6 libtexluajit2 libtinfo5 libucl1 libutf8proc2 libvpx7 lib
  python3-aioredis python3-apscheduler python3-backcall python3-boltor
  python3-graphql-core python3-graphql-relay python3-icalendar python3
  python3-rx python3-smoke-zephyr python3-texttable python3-tzlocal py
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  cgroupfs-mount containerd libintl-perl libintl-xs-perl libmodule-fir
```

After the docker installation, rustscan can run from the following command:

```
docker run -it --rm --name rustscan rustscan/rustscan:2.1.1 -a 192.168.1.7
```

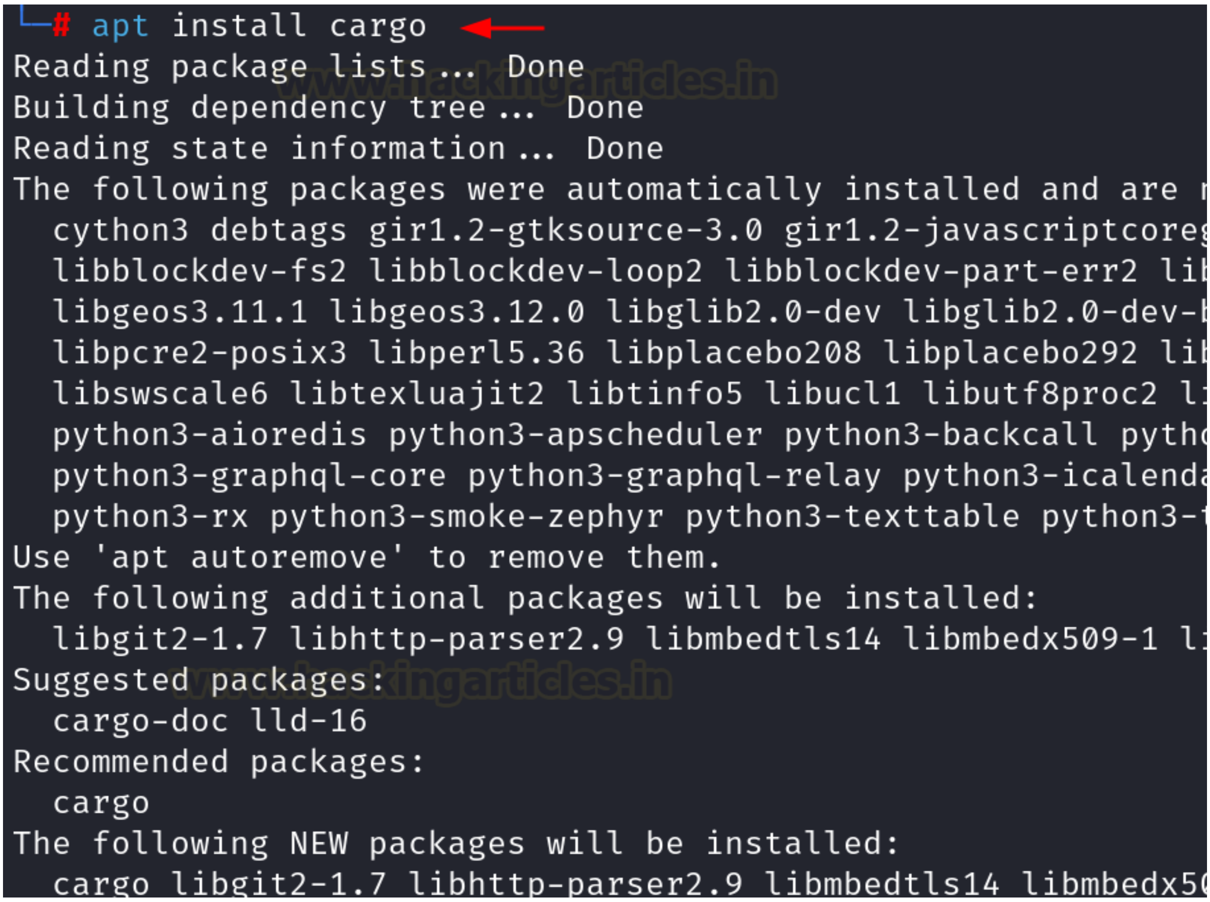


```
┌──(root㉿kali)-[~]
└─# docker run -it --rm --name rustscan rustscan/rustscan:2.1.1 -a 192.168.1.7
Unable to find image 'rustscan/rustscan:2.1.1' locally
2.1.1: Pulling from rustscan/rustscan
1b7ca6aea1dd: Pull complete
4c6d22701329: Pull complete
74847823a394: Pull complete
Digest: sha256:1ef645b90001c560923c2150d20e5509ced243122225dbb9daf7c1c0ac3d21b2
Status: Downloaded newer image for rustscan/rustscan:2.1.1
.----. .-. .-. .----..---.  .----. .---.   .--.  .-. .-.
| {}  }| { } |{ {__ {_   _}{ {__  /  ___} / {} \ |  `| |
| .-. \| {_} |.-._} } | |  .-._} }\     }/  /\  \| |\  |
`-' `-'`-----'`----'  `-'  `----'  `---' `-'  `-'`-' `-'
The Modern Day Port Scanner.

________________________________________
: http://discord.skerritt.blog           :
: https://github.com/RustScan/RustScan :
 --------------------------------------
Please contribute more quotes to our GitHub https://github.com/rustscan/rustscan

[~] The config file is expected to be at "/home/rustscan/.rustscan.toml"
[~] File limit higher than batch size. Can increase speed by increasing batch size '-b 1
Open 192.168.1.7:21
Open 192.168.1.7:22
Open 192.168.1.7:23
Open 192.168.1.7:25
Open 192.168.1.7:53
Open 192.168.1.7:80
Open 192.168.1.7:111
Open 192.168.1.7:139
Open 192.168.1.7:445
Open 192.168.1.7:512
Open 192.168.1.7:513
Open 192.168.1.7:514
Open 192.168.1.7:1099
Open 192.168.1.7:1524
Open 192.168.1.7:2049
Open 192.168.1.7:2121
Open 192.168.1.7:3306
```

Installation and Usage (Standalone)

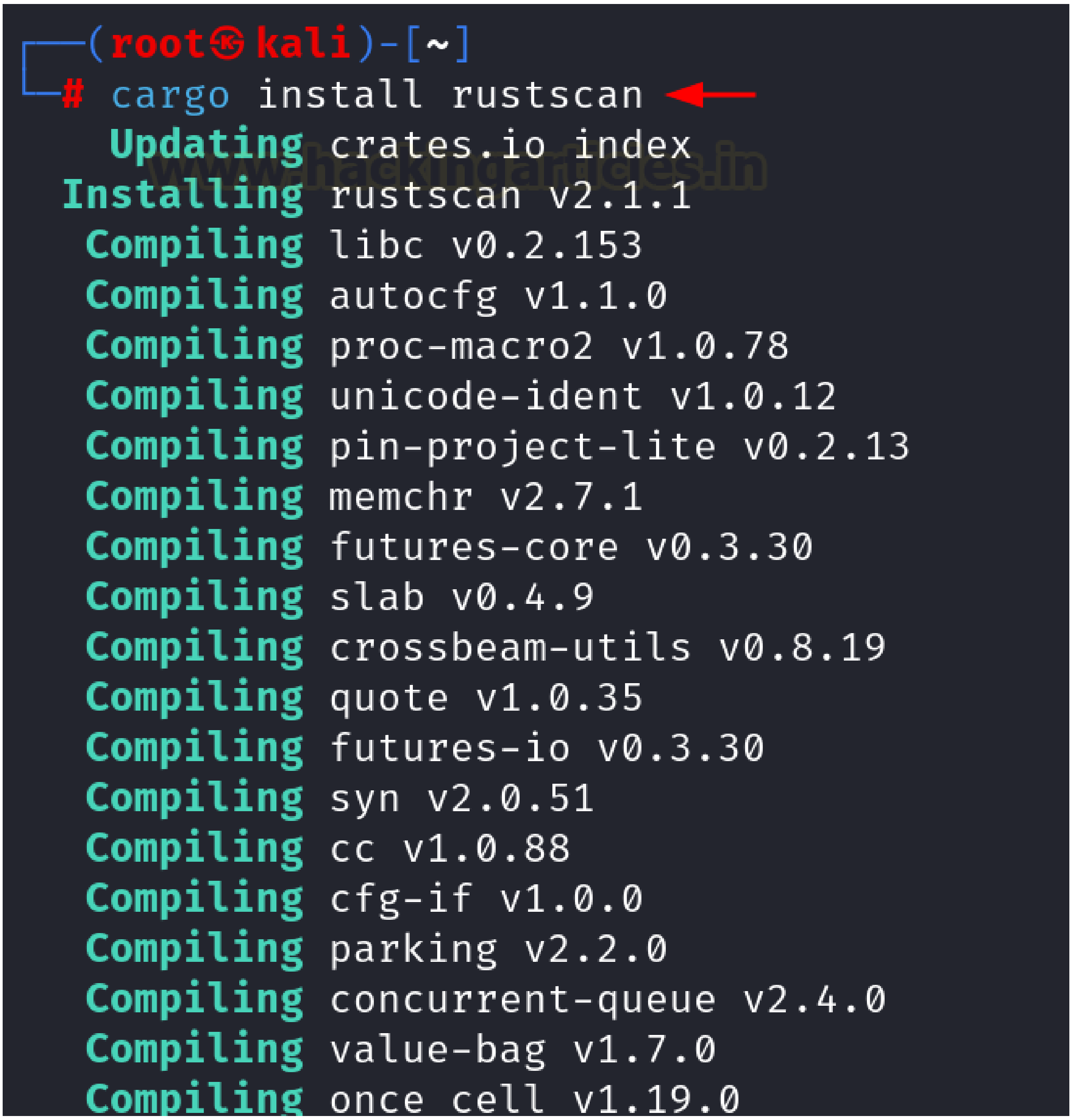
Installation of Rustscan can be performed using cargo, the following command can be used:

```
apt install cargo
```



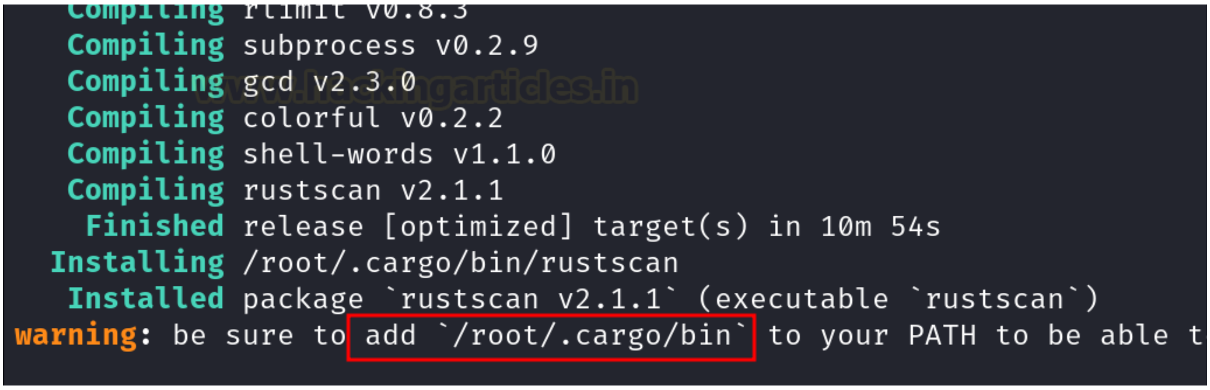
```
└─# apt install cargo
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are r
  cython3 debtags gir1.2-gtksource-3.0 gir1.2-javascriptcoreg
  libblockdev-fs2 libblockdev-loop2 libblockdev-part-err2 lib
  libgeos3.11.1 libgeos3.12.0 libglib2.0-dev libglib2.0-dev-b
  libpcre2-posix3 libperl5.36 libplacebo208 libplacebo292 lib
  libswscale6 libtexluajit2 libtinfo5 libucl1 libutf8proc2 li
  python3-aioredis python3-apscheduler python3-backcall pytho
  python3-graphql-core python3-graphql-relay python3-icalenda
  python3-rx python3-smoke-zephyr python3-texttable python3-t
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  libgit2-1.7 libhttp-parser2.9 libmbedtls14 libmbedx509-1 li
Suggested packages:
  cargo-doc lld-16
Recommended packages:
  cargo
The following NEW packages will be installed:
  cargo libgit2-1.7 libhttp-parser2.9 libmbedtls14 libmbedx50
```

```
cargo install rustscan
```



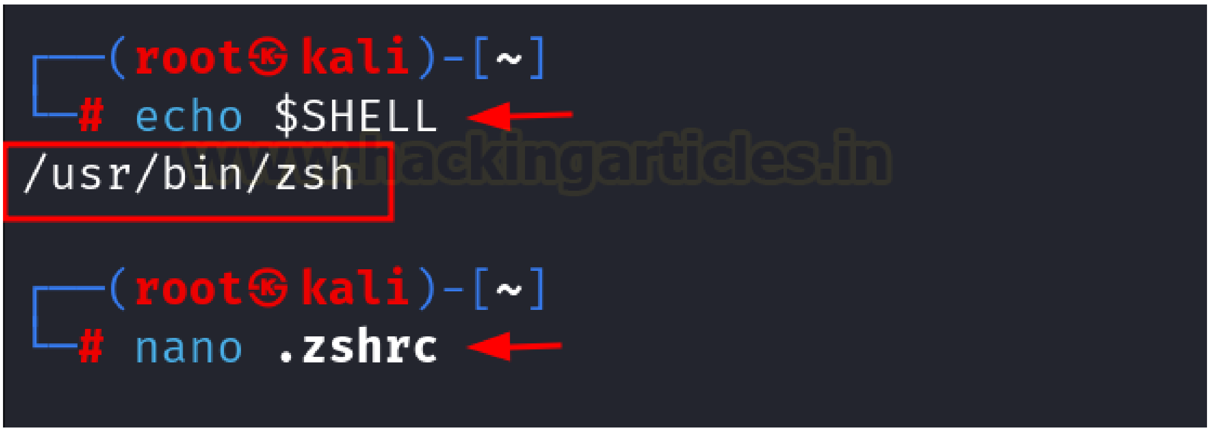
```
┌──(root㉿kali)-[~]
└─# cargo install rustscan
    Updating crates.io index
  Installing rustscan v2.1.1
   Compiling libc v0.2.153
   Compiling autocfg v1.1.0
   Compiling proc-macro2 v1.0.78
   Compiling unicode-ident v1.0.12
   Compiling pin-project-lite v0.2.13
   Compiling memchr v2.7.1
   Compiling futures-core v0.3.30
   Compiling slab v0.4.9
   Compiling crossbeam-utils v0.8.19
   Compiling quote v1.0.35
   Compiling futures-io v0.3.30
   Compiling syn v2.0.51
   Compiling cc v1.0.88
   Compiling cfg-if v1.0.0
   Compiling parking v2.2.0
   Compiling concurrent-queue v2.4.0
   Compiling value-bag v1.7.0
   Compiling once_cell v1.19.0
```

Make sure to add the **/root/.cargo/bin** to the path



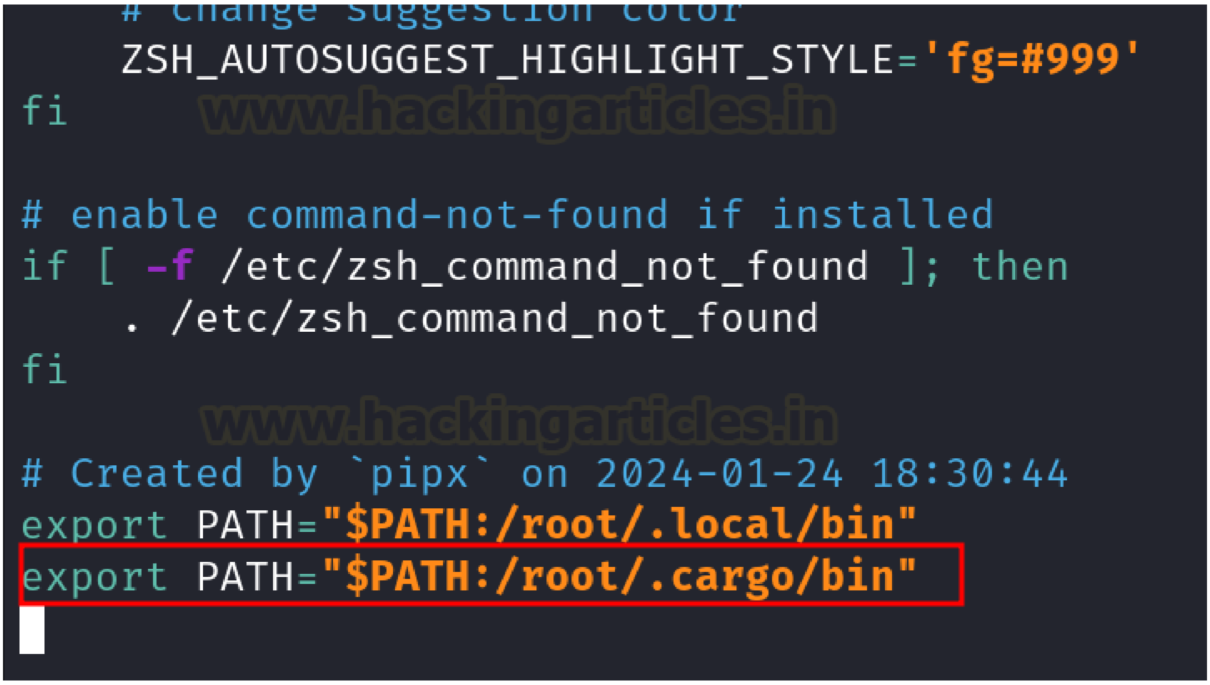
```
   Compiling rlimit v0.8.3
   Compiling subprocess v0.2.9
   Compiling gcd v2.3.0
   Compiling colorful v0.2.2
   Compiling shell-words v1.1.0
   Compiling rustscan v2.1.1
    Finished release [optimized] target(s) in 10m 54s
  Installing /root/.cargo/bin/rustscan
   Installed package `rustscan v2.1.1` (executable `rustscan`)
warning: be sure to add `/root/.cargo/bin` to your PATH to be able t
```

```
echo $SHELL
nano .zshrc
```



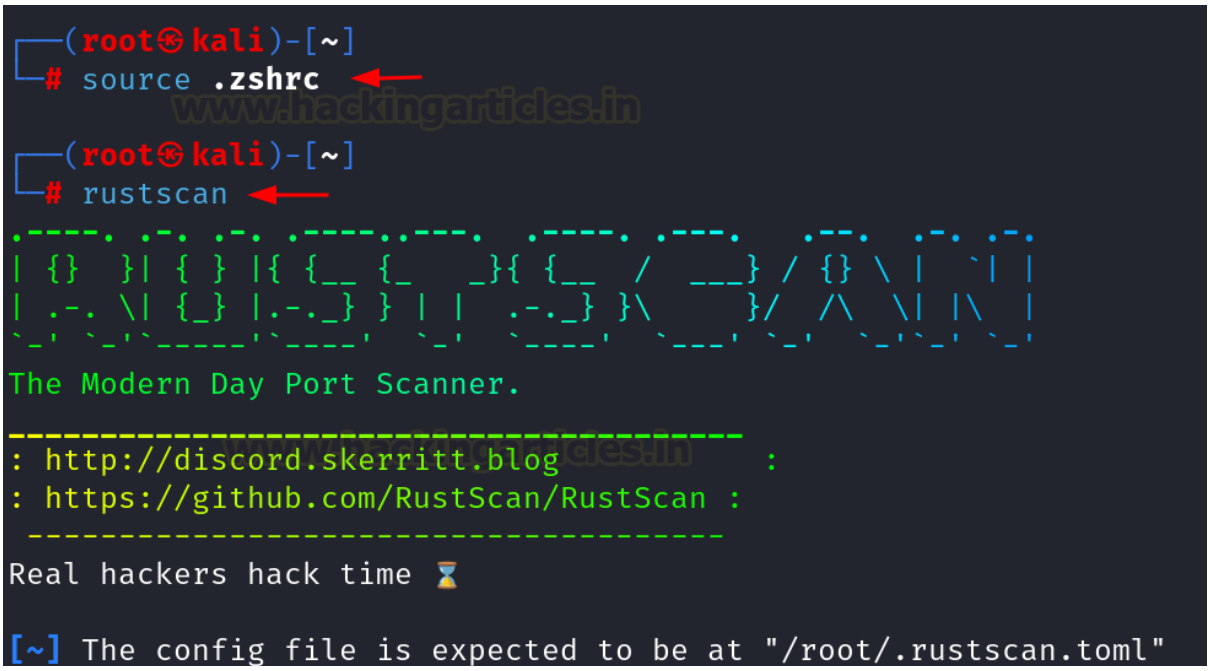
```
┌──(root㉿kali)-[~]
└─# echo $SHELL
/usr/bin/zsh

┌──(root㉿kali)-[~]
└─# nano .zshrc
```

Add the **/root/.cargo/bin** as export PATH.



```
    # change suggestion color
    ZSH_AUTOSUGGEST_HIGHLIGHT_STYLE='fg=#999'
fi

# enable command-not-found if installed
if [ -f /etc/zsh_command_not_found ]; then
    . /etc/zsh_command_not_found
fi

# Created by `pipx` on 2024-01-24 18:30:44
export PATH="$PATH:/root/.local/bin"
export PATH="$PATH:/root/.cargo/bin"
```

After installation success, Rustscan is now ready to run.



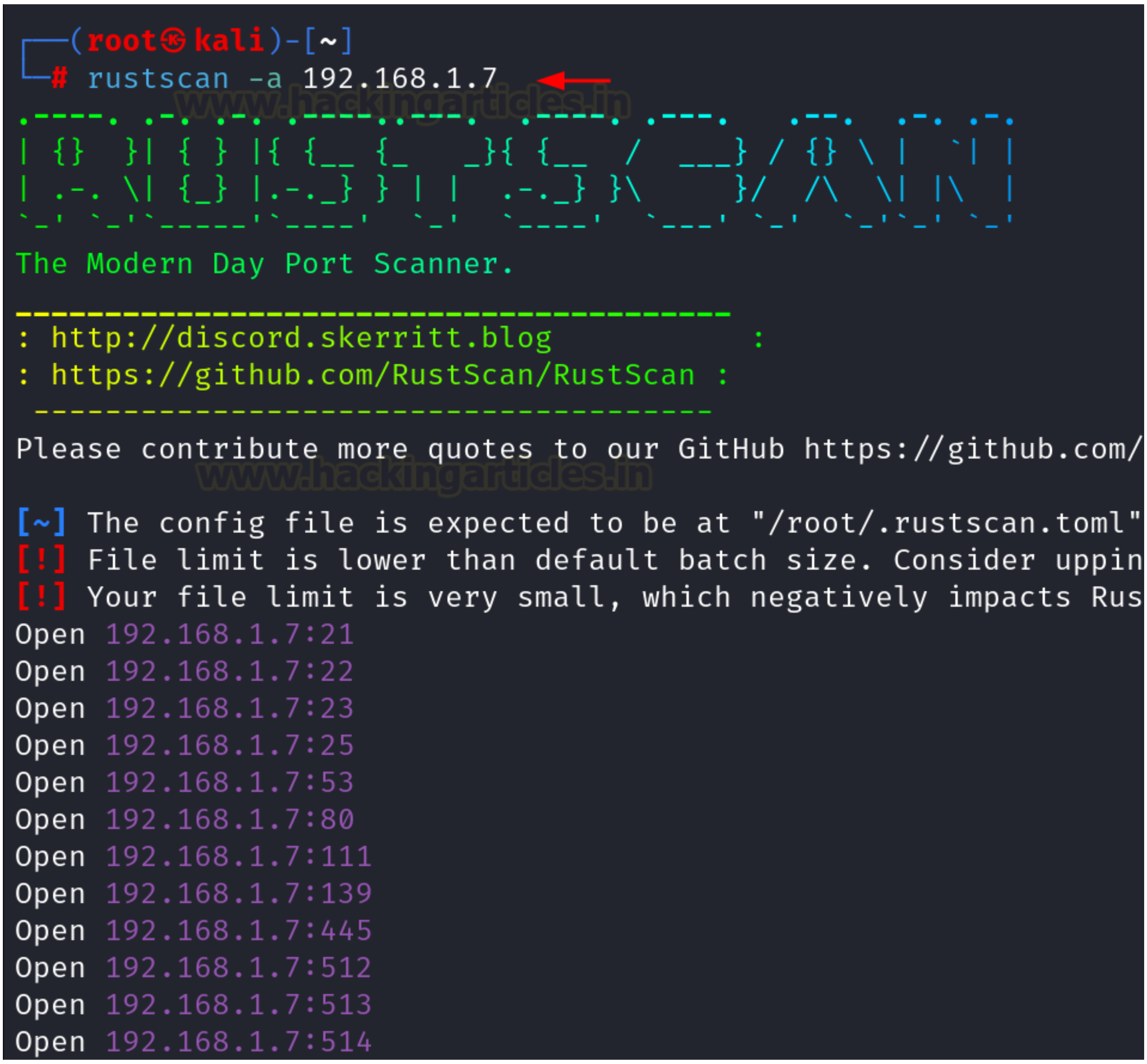
```
┌──(root㉿kali)-[~]
└─# source .zshrc

┌──(root㉿kali)-[~]
└─# rustscan
.----. .-. .-. .----..---.  .----. .---.   .--.  .-. .-.
| {}  }| { } |{ {__ {_   _}{ {__  /  ___} / {} \ |  `| |
| .-. \| {_} |.-._} } | |  .-._} }\     }/  /\  \| |\  |
`-' `-'`-----'`----'  `-'  `----'  `---' `-'  `-'`-' `-'
The Modern Day Port Scanner.
________________________________________
: http://discord.skerritt.blog           :
: https://github.com/RustScan/RustScan :
 --------------------------------------
Real hackers hack time ⌛

[~] The config file is expected to be at "/root/.rustscan.toml"
```

Rustscan flags

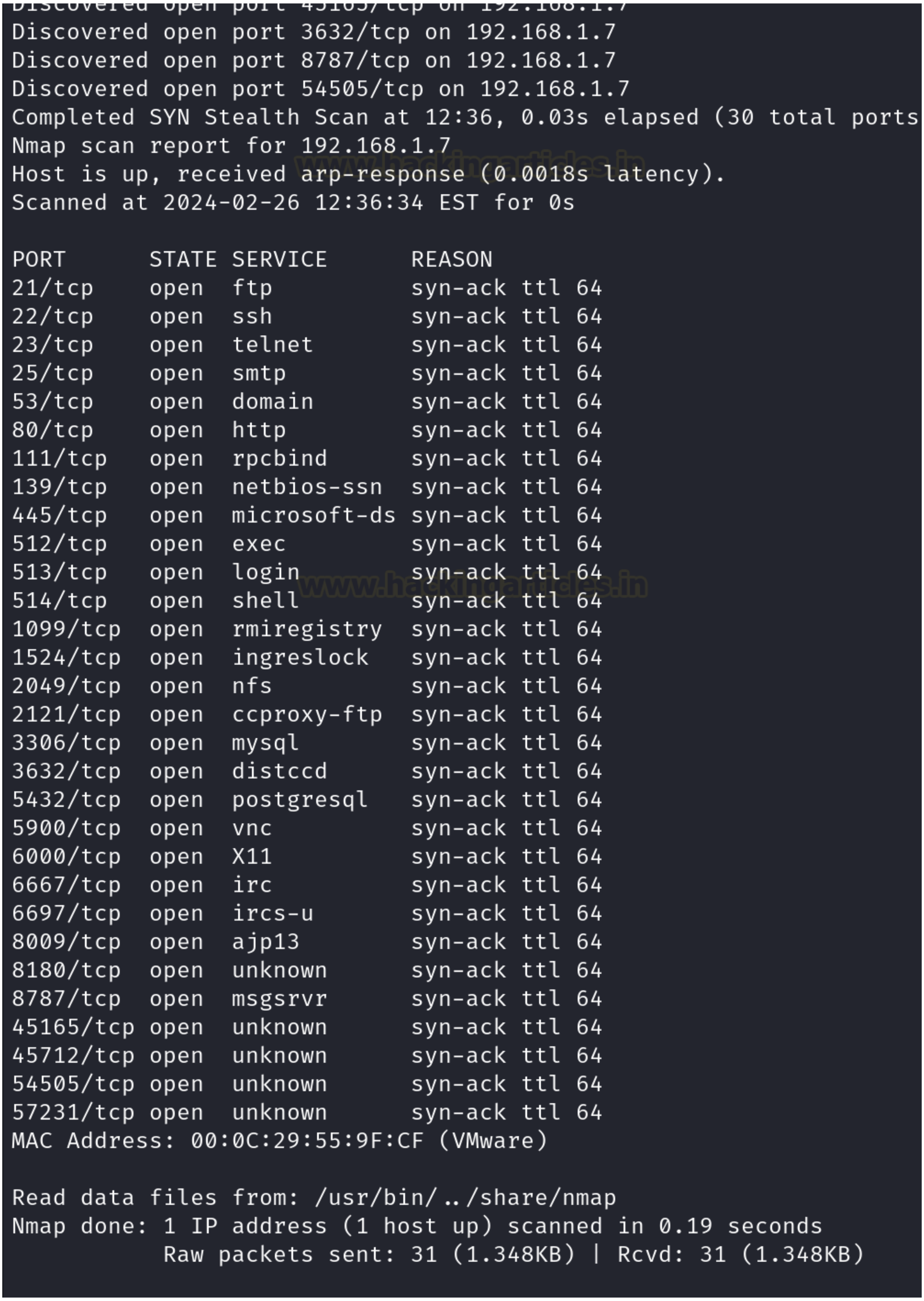
There are a number of operations which can be performed using Rustscan, below listed are the flags to perform respective operation in Rustscan.

-a : To perform a comprehensive scan of all TCP ports.

```
rustscan -a 192.168.1.7
```



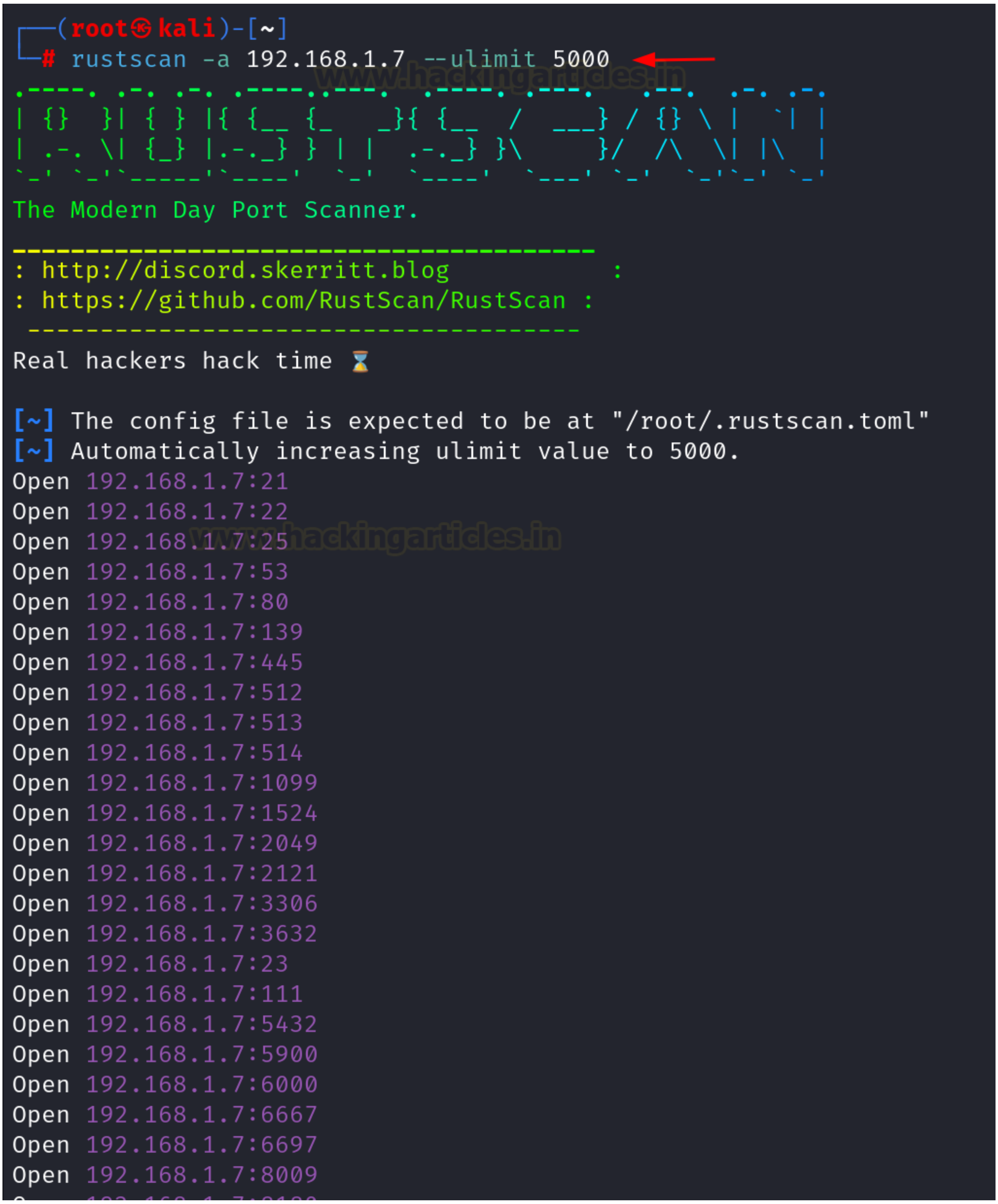
```
┌──(root㉿kali)-[~]
└─# rustscan -a 192.168.1.7
.----. .-. .-. .----..---.  .----. .---.   .--.  .-. .-.
| {}  }| { } |{ {__ {_   _}{ {__  /  ___} / {} \ |  `| |
| .-. \| {_} |.-._} } | |  .-._} }\     }/  /\  \| |\  |
`-' `-'`-----'`----'  `-'  `----'  `---' `-'  `-'`-' `-'
The Modern Day Port Scanner.

----------------------------------------
: http://discord.skerritt.blog           :
: https://github.com/RustScan/RustScan :
 --------------------------------------
Please contribute more quotes to our GitHub https://github.com/

[~] The config file is expected to be at "/root/.rustscan.toml"
[!] File limit is lower than default batch size. Consider uppin
[!] Your file limit is very small, which negatively impacts Rus
Open 192.168.1.7:21
Open 192.168.1.7:22
Open 192.168.1.7:23
Open 192.168.1.7:25
Open 192.168.1.7:53
Open 192.168.1.7:80
Open 192.168.1.7:111
Open 192.168.1.7:139
Open 192.168.1.7:445
Open 192.168.1.7:512
Open 192.168.1.7:513
Open 192.168.1.7:514
```



```
Discovered open port 45165/tcp on 192.168.1.7
Discovered open port 3632/tcp on 192.168.1.7
Discovered open port 8787/tcp on 192.168.1.7
Discovered open port 54505/tcp on 192.168.1.7
Completed SYN Stealth Scan at 12:36, 0.03s elapsed (30 total ports
Nmap scan report for 192.168.1.7
Host is up, received arp-response (0.0018s latency).
Scanned at 2024-02-26 12:36:34 EST for 0s

PORT      STATE SERVICE      REASON
21/tcp    open  ftp          syn-ack ttl 64
22/tcp    open  ssh          syn-ack ttl 64
23/tcp    open  telnet       syn-ack ttl 64
25/tcp    open  smtp         syn-ack ttl 64
53/tcp    open  domain       syn-ack ttl 64
80/tcp    open  http         syn-ack ttl 64
111/tcp   open  rpcbind      syn-ack ttl 64
139/tcp   open  netbios-ssn  syn-ack ttl 64
445/tcp   open  microsoft-ds syn-ack ttl 64
512/tcp   open  exec         syn-ack ttl 64
513/tcp   open  login        syn-ack ttl 64
514/tcp   open  shell        syn-ack ttl 64
1099/tcp  open  rmiregistry  syn-ack ttl 64
1524/tcp  open  ingreslock   syn-ack ttl 64
2049/tcp  open  nfs          syn-ack ttl 64
2121/tcp  open  ccproxy-ftp  syn-ack ttl 64
3306/tcp  open  mysql        syn-ack ttl 64
3632/tcp  open  distccd      syn-ack ttl 64
5432/tcp  open  postgresql   syn-ack ttl 64
5900/tcp  open  vnc          syn-ack ttl 64
6000/tcp  open  X11          syn-ack ttl 64
6667/tcp  open  irc          syn-ack ttl 64
6697/tcp  open  ircs-u       syn-ack ttl 64
8009/tcp  open  ajp13        syn-ack ttl 64
8180/tcp  open  unknown      syn-ack ttl 64
8787/tcp  open  msgsrvr      syn-ack ttl 64
45165/tcp open  unknown      syn-ack ttl 64
45712/tcp open  unknown      syn-ack ttl 64
54505/tcp open  unknown      syn-ack ttl 64
57231/tcp open  unknown      syn-ack ttl 64
MAC Address: 00:0C:29:55:9F:CF (VMware)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
           Raw packets sent: 31 (1.348KB) | Rcvd: 31 (1.348KB)
```

--ulimit : To adjust the ulimit for file descriptors to handle large-scale scans. If the scan is running slow adding this flag with a value of 5000 will make it run faster.

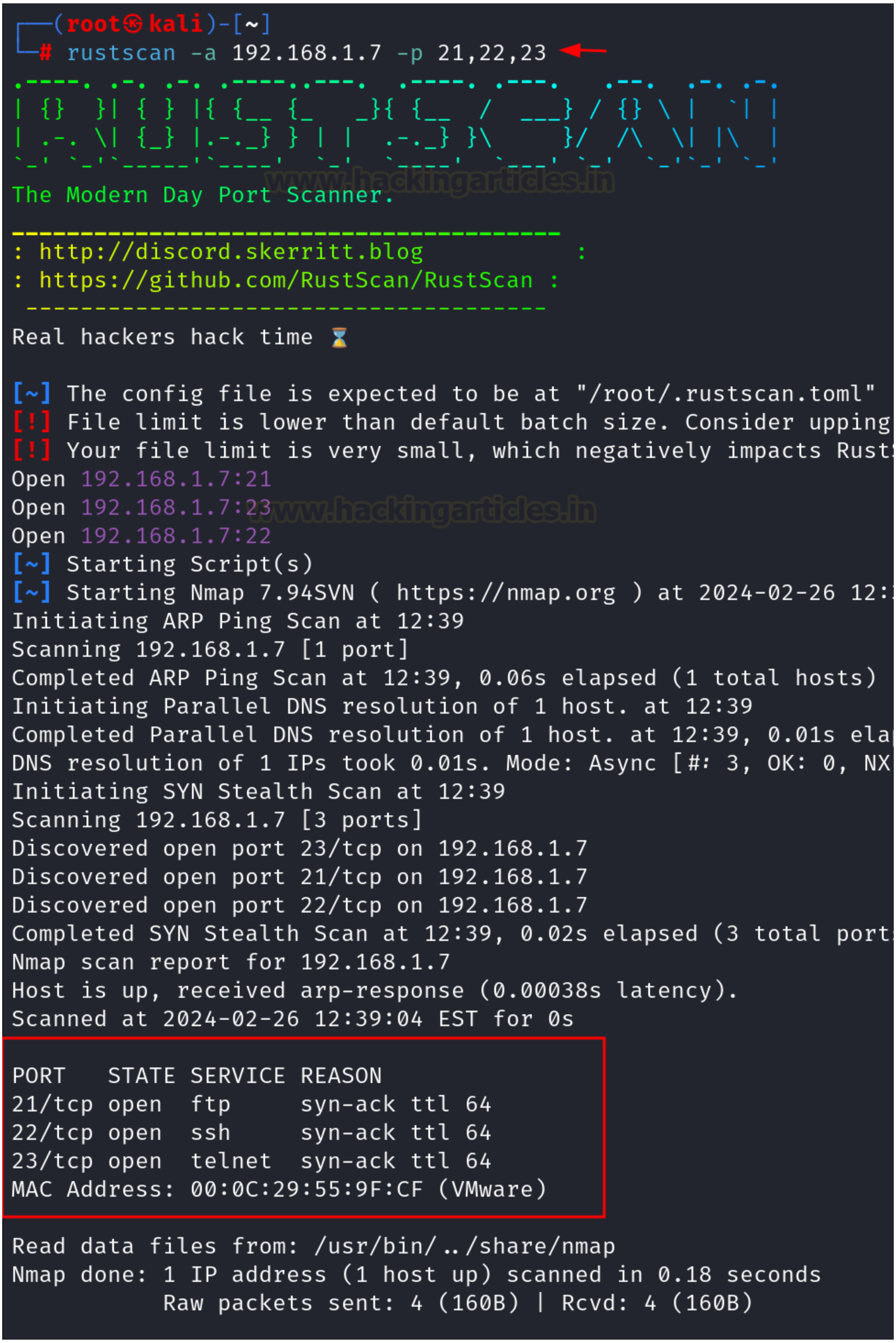
```
rustscan -a 192.168.1.7 --ulimit 5000
```



```
┌──(root㉿kali)-[~]
└─# rustscan -a 192.168.1.7 --ulimit 5000
.----. .-. .-. .----..---.  .----. .---.   .--.  .-. .-.
| {}  }| { } |{ {__ {_   _}{ {__  /  ___} / {} \ |  `| |
| .-. \| {_} |.-._} } | |  .-._} }\     }/  /\  \| |\  |
`-' `-'`-----'`----'  `-'  `----'  `---' `-'  `-'`-' `-'
The Modern Day Port Scanner.

________________________________________
: http://discord.skerritt.blog           :
: https://github.com/RustScan/RustScan :
 --------------------------------------
Real hackers hack time ⌛

[~] The config file is expected to be at "/root/.rustscan.toml"
[~] Automatically increasing ulimit value to 5000.
Open 192.168.1.7:21
Open 192.168.1.7:22
Open 192.168.1.7:25
Open 192.168.1.7:53
Open 192.168.1.7:80
Open 192.168.1.7:139
Open 192.168.1.7:445
Open 192.168.1.7:512
Open 192.168.1.7:513
Open 192.168.1.7:514
Open 192.168.1.7:1099
Open 192.168.1.7:1524
Open 192.168.1.7:2049
Open 192.168.1.7:2121
Open 192.168.1.7:3306
Open 192.168.1.7:3632
Open 192.168.1.7:23
Open 192.168.1.7:111
Open 192.168.1.7:5432
Open 192.168.1.7:5900
Open 192.168.1.7:6000
Open 192.168.1.7:6667
Open 192.168.1.7:6697
Open 192.168.1.7:8009
```

-p : To define specific ports to be scanned.

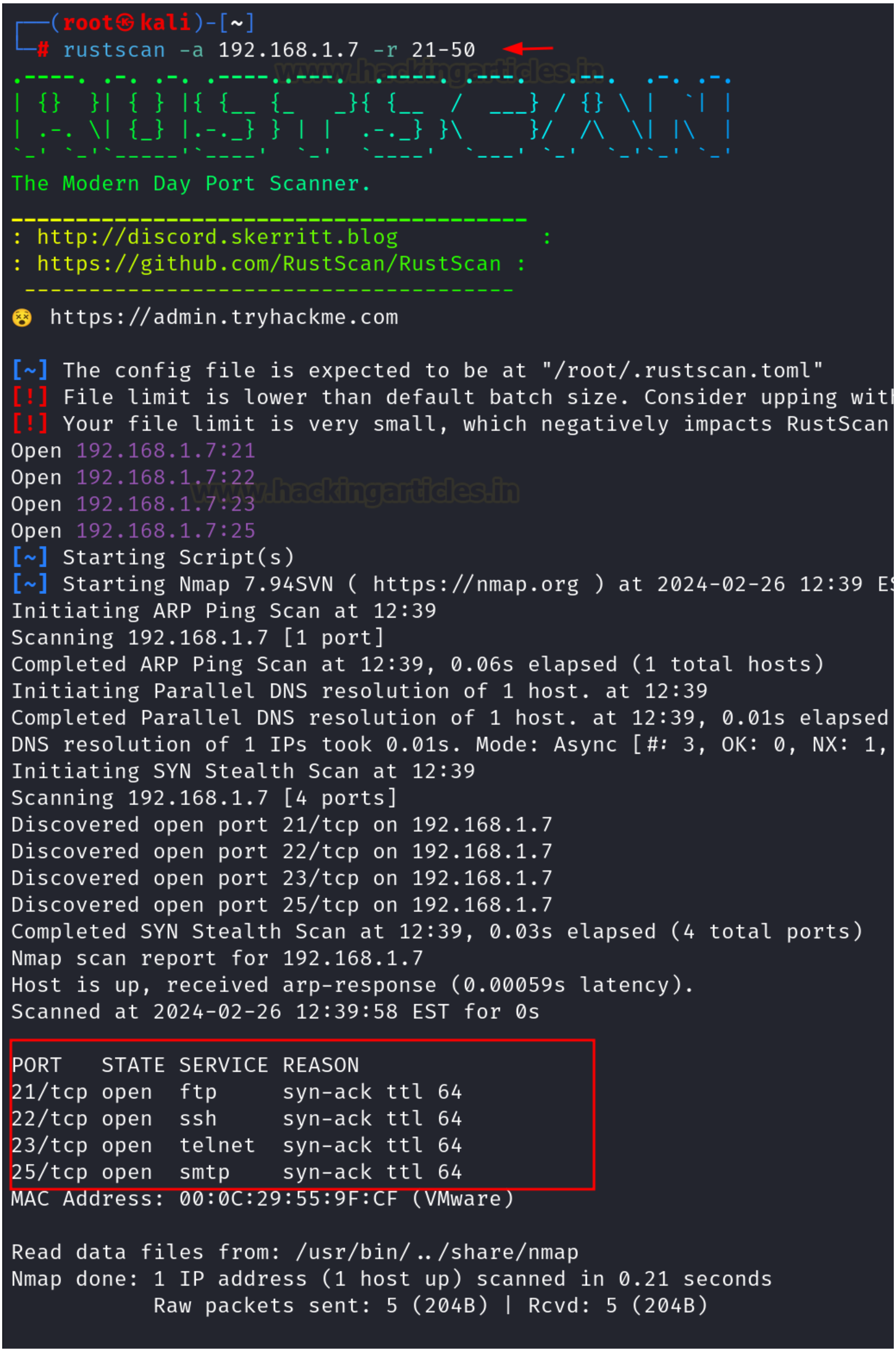
```
rustscan -a 192.168.1.7 -p 21,22,23
```



```
┌──(root㉿kali)-[~]
└─# rustscan -a 192.168.1.7 -p 21,22,23
.----. .-. .-. .----..---.  .----. .---.   .--.  .-. .-.
| {}  }| { } |{ {__ {_   _}{ {__  /  ___} / {} \ |  `| |
| .-. \| {_} |.-._} } | |  .-._} }\     }/  /\  \| |\  |
`-' `-'`-----'`----'  `-'  `----'  `---' `-'  `-'`-' `-'
The Modern Day Port Scanner.

________________________________________
: http://discord.skerritt.blog           :
: https://github.com/RustScan/RustScan :
 --------------------------------------
Real hackers hack time ⌛

[~] The config file is expected to be at "/root/.rustscan.toml"
[!] File limit is lower than default batch size. Consider upping
[!] Your file limit is very small, which negatively impacts Rust
Open 192.168.1.7:21
Open 192.168.1.7:23
Open 192.168.1.7:22
[~] Starting Script(s)
[~] Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-26 12:
Initiating ARP Ping Scan at 12:39
Scanning 192.168.1.7 [1 port]
Completed ARP Ping Scan at 12:39, 0.06s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 12:39
Completed Parallel DNS resolution of 1 host. at 12:39, 0.01s ela
DNS resolution of 1 IPs took 0.01s. Mode: Async [#: 3, OK: 0, NX
Initiating SYN Stealth Scan at 12:39
Scanning 192.168.1.7 [3 ports]
Discovered open port 23/tcp on 192.168.1.7
Discovered open port 21/tcp on 192.168.1.7
Discovered open port 22/tcp on 192.168.1.7
Completed SYN Stealth Scan at 12:39, 0.02s elapsed (3 total port
Nmap scan report for 192.168.1.7
Host is up, received arp-response (0.00038s latency).
Scanned at 2024-02-26 12:39:04 EST for 0s

PORT   STATE SERVICE REASON
21/tcp open  ftp     syn-ack ttl 64
22/tcp open  ssh     syn-ack ttl 64
23/tcp open  telnet  syn-ack ttl 64
MAC Address: 00:0C:29:55:9F:CF (VMware)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.18 seconds
           Raw packets sent: 4 (160B) | Rcvd: 4 (160B)
```

-r : To perform a range scan, specific range of ports will be scanned.

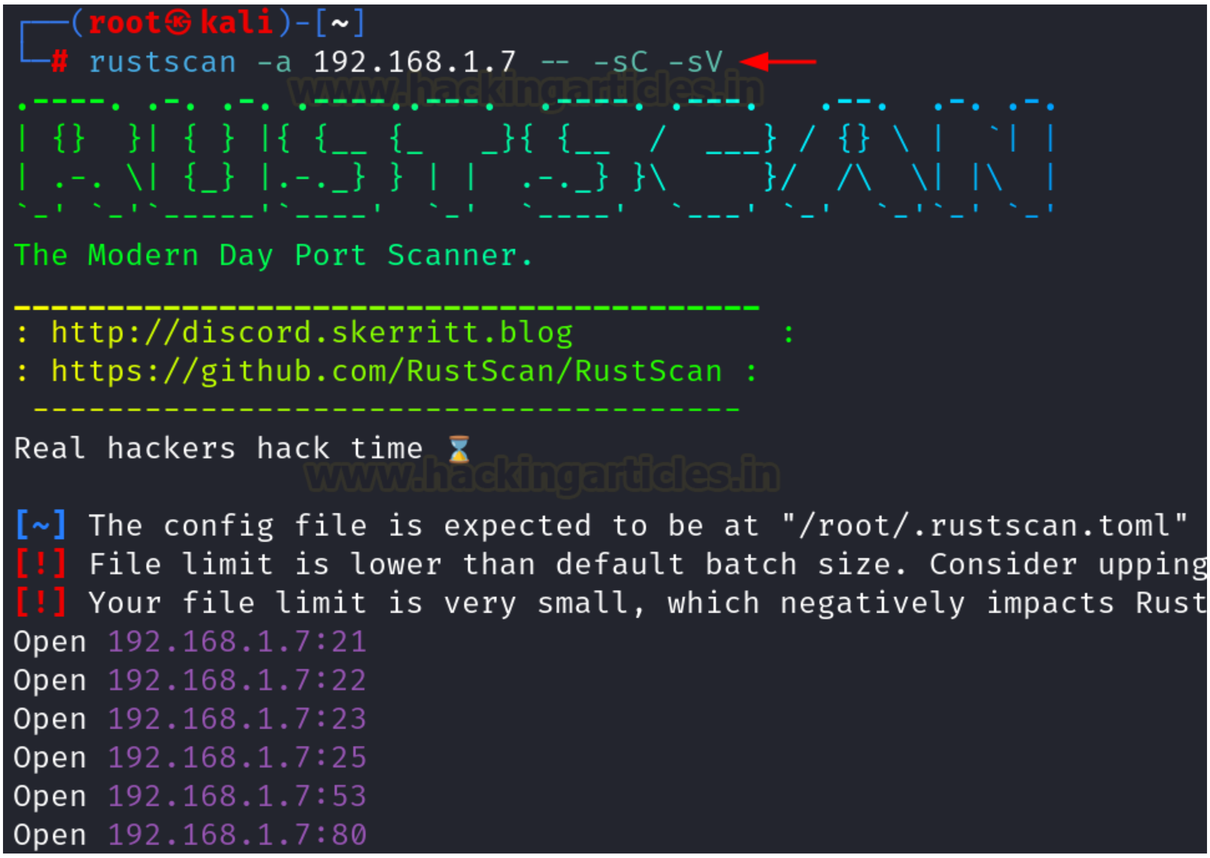
```
rustscan -a 192.168.1.7 -r 21-50
```



```
┌──(root㉿kali)-[~]
└─# rustscan -a 192.168.1.7 -r 21-50
.----. .-. .-. .----..---.  .----. .---.   .--.  .-. .-.
| {}  }| { } |{ {__ {_   _}{ {__  /  ___} / {} \ |  `| |
| .-. \| {_} |.-._} } | |  .-._} }\     }/  /\  \| |\  |
`-' `-'`-----'`----'  `-'  `----'  `---' `-'  `-'`-' `-'
The Modern Day Port Scanner.

________________________________________
: http://discord.skerritt.blog           :
: https://github.com/RustScan/RustScan :
 --------------------------------------
😵 https://admin.tryhackme.com

[~] The config file is expected to be at "/root/.rustscan.toml"
[!] File limit is lower than default batch size. Consider upping wit
[!] Your file limit is very small, which negatively impacts RustScan
Open 192.168.1.7:21
Open 192.168.1.7:22
Open 192.168.1.7:23
Open 192.168.1.7:25
[~] Starting Script(s)
[~] Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-26 12:39 ES
Initiating ARP Ping Scan at 12:39
Scanning 192.168.1.7 [1 port]
Completed ARP Ping Scan at 12:39, 0.06s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 12:39
Completed Parallel DNS resolution of 1 host. at 12:39, 0.01s elapsed
DNS resolution of 1 IPs took 0.01s. Mode: Async [#: 3, OK: 0, NX: 1,
Initiating SYN Stealth Scan at 12:39
Scanning 192.168.1.7 [4 ports]
Discovered open port 21/tcp on 192.168.1.7
Discovered open port 22/tcp on 192.168.1.7
Discovered open port 23/tcp on 192.168.1.7
Discovered open port 25/tcp on 192.168.1.7
Completed SYN Stealth Scan at 12:39, 0.03s elapsed (4 total ports)
Nmap scan report for 192.168.1.7
Host is up, received arp-response (0.00059s latency).
Scanned at 2024-02-26 12:39:58 EST for 0s

PORT   STATE SERVICE REASON
21/tcp open  ftp     syn-ack ttl 64
22/tcp open  ssh     syn-ack ttl 64
23/tcp open  telnet  syn-ack ttl 64
25/tcp open  smtp    syn-ack ttl 64
MAC Address: 00:0C:29:55:9F:CF (VMware)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
           Raw packets sent: 5 (204B) | Rcvd: 5 (204B)
```

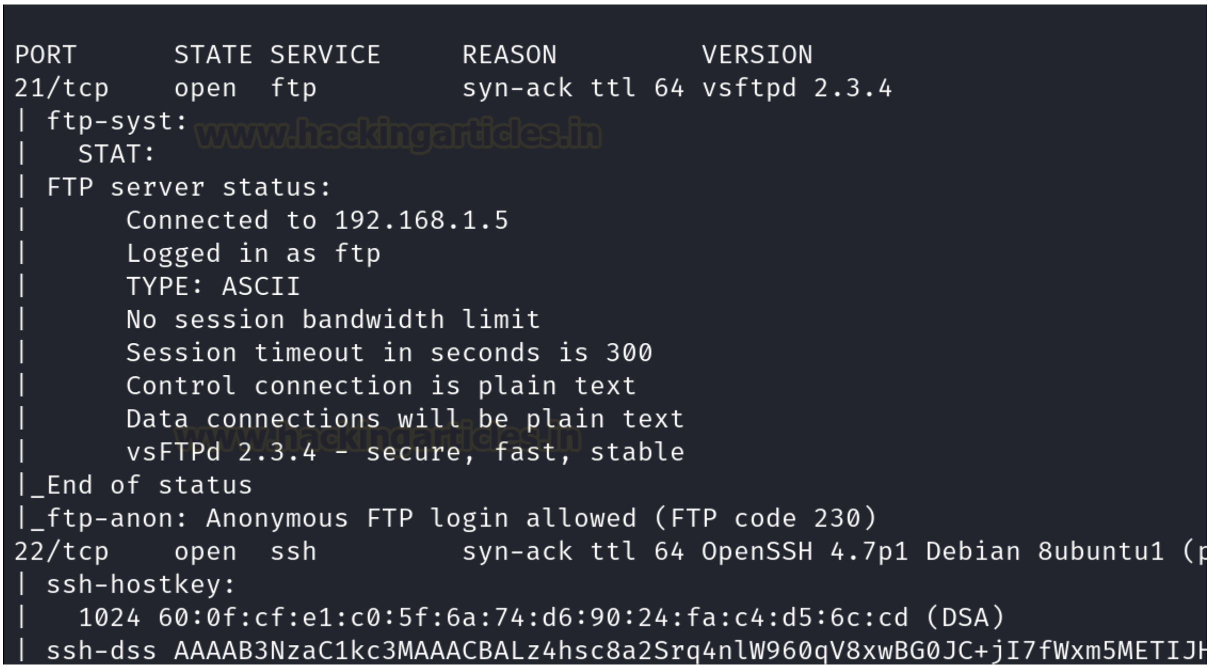
-sC -sV : To perform default script scan and service version scan.

```
rustscan -a 192.168.1.7 -- -sC -sV
```



```
┌──(root㉿kali)-[~]
└─# rustscan -a 192.168.1.7 -- -sC -sV
.----. .-. .-. .----..---.  .----. .---.   .--.  .-. .-.
| {}  }| { } |{ {__ {_   _}{ {__  /  ___} / {} \ |  `| |
| .-. \| {_} |.-._} } | |  .-._} }\     }/  /\  \| |\  |
`-' `-'`-----'`----'  `-'  `----'  `---' `-'  `-'`-' `-'
The Modern Day Port Scanner.
________________________________________
: http://discord.skerritt.blog           :
: https://github.com/RustScan/RustScan :
 --------------------------------------
Real hackers hack time ⌛

[~] The config file is expected to be at "/root/.rustscan.toml"
[!] File limit is lower than default batch size. Consider upping
[!] Your file limit is very small, which negatively impacts Rust
Open 192.168.1.7:21
Open 192.168.1.7:22
Open 192.168.1.7:23
Open 192.168.1.7:25
Open 192.168.1.7:53
Open 192.168.1.7:80
```

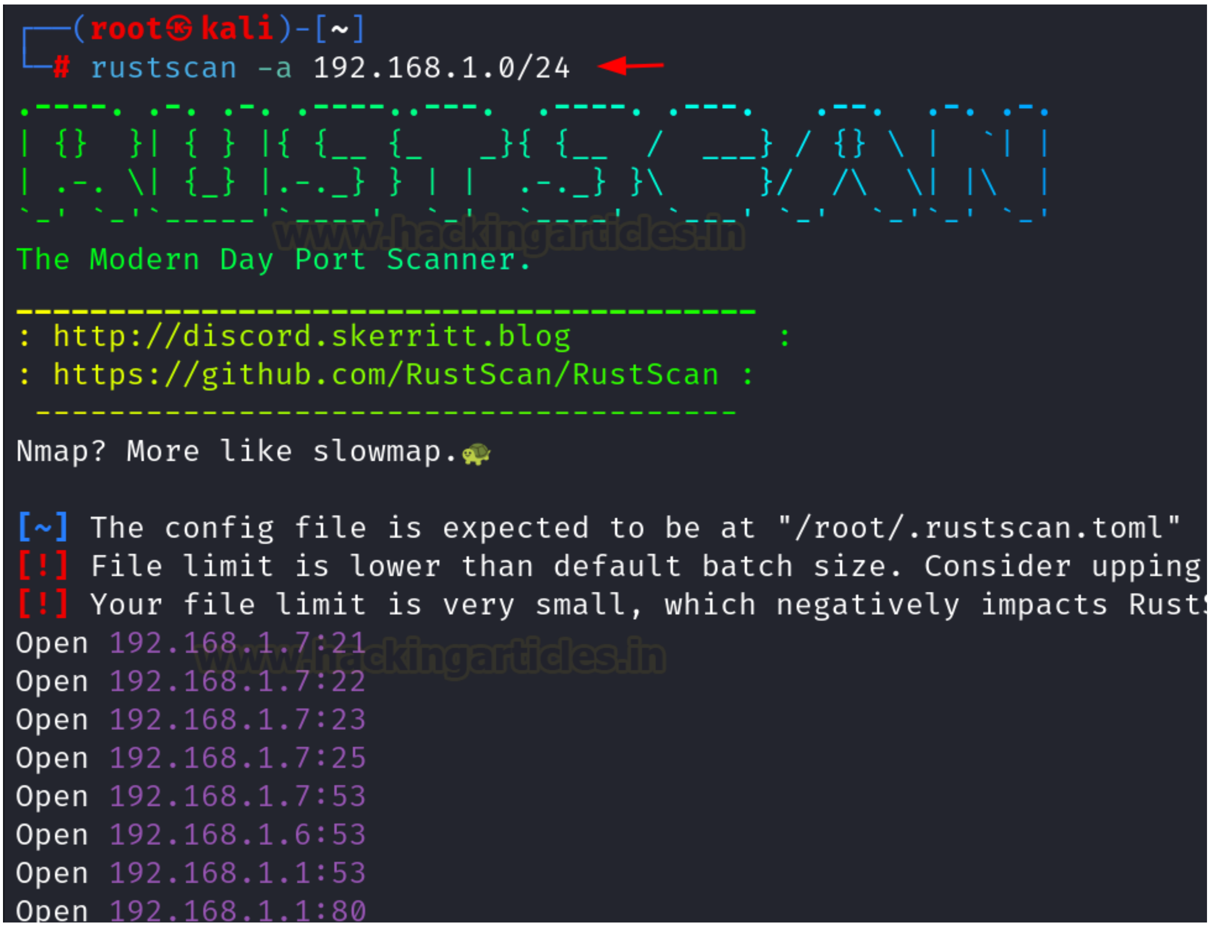
Results of service version and default script scan can be seen below.



```
PORT      STATE SERVICE     REASON         VERSION
21/tcp    open  ftp         syn-ack ttl 64 vsftpd 2.3.4
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.1.5
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      vsFTPd 2.3.4 - secure, fast, stable
|_End of status
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp    open  ssh         syn-ack ttl 64 OpenSSH 4.7p1 Debian 8ubuntu1 (p
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
| ssh-dss AAAAB3NzaC1kc3MAAACBALz4hsc8a2Srq4nlW960qV8xwBG0JC+jI7fWxm5METIJH
```

Rustscan can also be used to scan the entire subnet IP addresses by just adding a **/24** after the IP address.

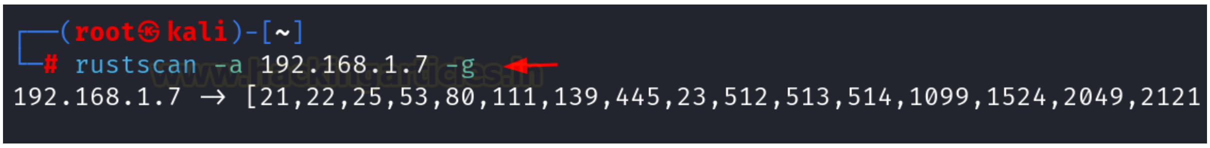
```
rustscan -a 192.168.1.0/24
```



```
┌──(root㉿kali)-[~]
└─# rustscan -a 192.168.1.0/24
.----. .-. .-. .----..---.  .----. .---.   .--.  .-. .-.
| {}  }| { } |{ {__ {_   _}{ {__  /  ___} / {} \ |  `| |
| .-. \| {_} |.-._} } | |  .-._} }\     }/  /\  \| |\  |
`-' `-'`-----'`----'  `-'  `----'  `---' `-'  `-'`-' `-'
The Modern Day Port Scanner.

________________________________________
: http://discord.skerritt.blog           :
: https://github.com/RustScan/RustScan :
 --------------------------------------
Nmap? More like slowmap.🐢

[~] The config file is expected to be at "/root/.rustscan.toml"
[!] File limit is lower than default batch size. Consider upping
[!] Your file limit is very small, which negatively impacts Rust
Open 192.168.1.7:21
Open 192.168.1.7:22
Open 192.168.1.7:23
Open 192.168.1.7:25
Open 192.168.1.7:53
Open 192.168.1.6:53
Open 192.168.1.1:53
Open 192.168.1.1:80
```

-g : To enable the "greppable" output format for easy parsing and analysis.

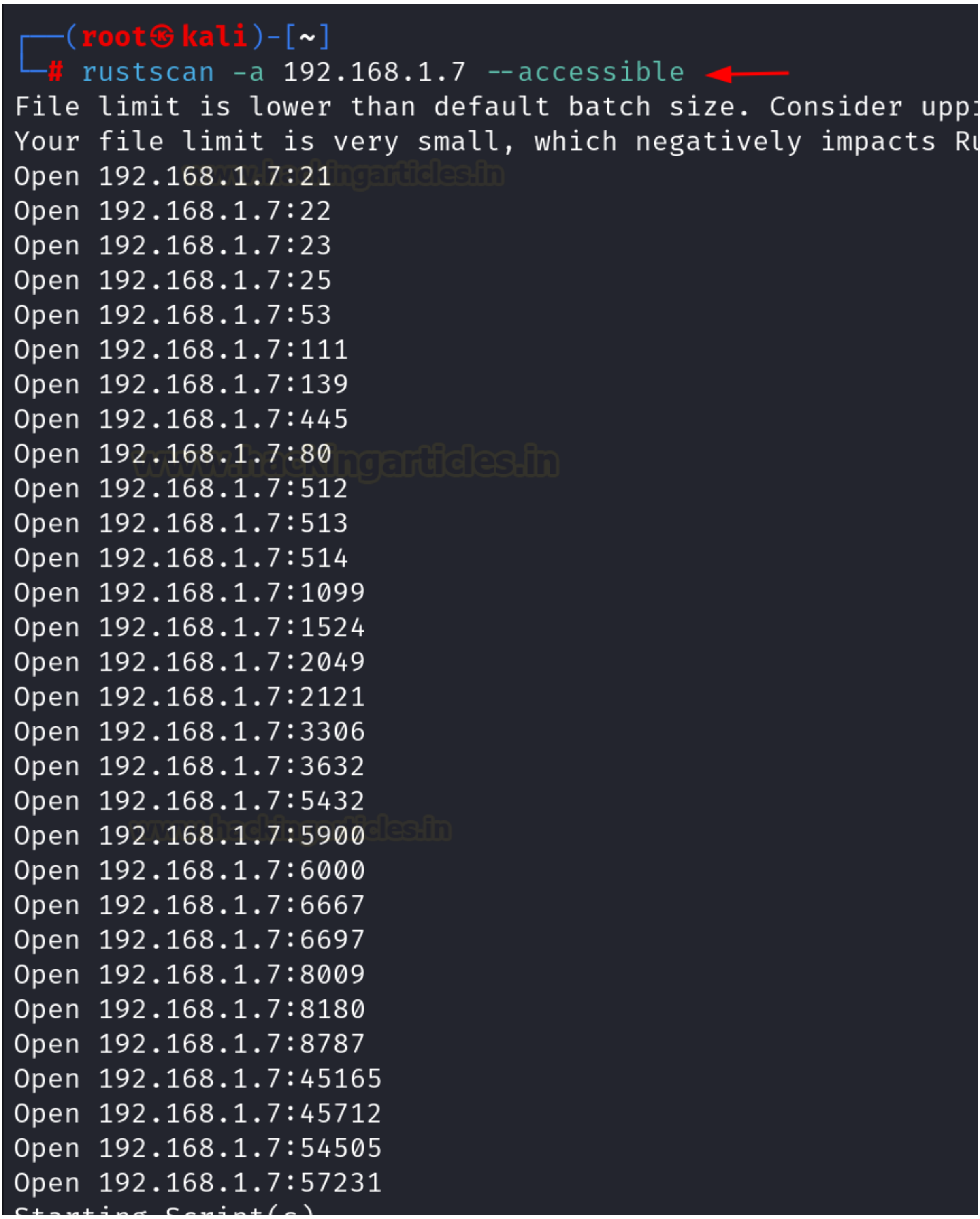
```
rustscan -a 192.168.1.7 -g
```



```
┌──(root㉿kali)-[~]
└─# rustscan -a 192.168.1.7 -g
192.168.1.7 -> [21,22,25,53,80,111,139,445,23,512,513,514,1099,1524,2049,2121
```

--accessible : Turn on accessible mode, does not print ASCII art. Also does not print very large blocks of text, as this can cause some pain with screenreaders. This reduces the information you get.

```
rustscan -a 192.168.1.7 --accessible
```



```
┌──(root㉿kali)-[~]
└─# rustscan -a 192.168.1.7 --accessible
File limit is lower than default batch size. Consider upp
Your file limit is very small, which negatively impacts Ru
Open 192.168.1.7:21
Open 192.168.1.7:22
Open 192.168.1.7:23
Open 192.168.1.7:25
Open 192.168.1.7:53
Open 192.168.1.7:111
Open 192.168.1.7:139
Open 192.168.1.7:445
Open 192.168.1.7:80
Open 192.168.1.7:512
Open 192.168.1.7:513
Open 192.168.1.7:514
Open 192.168.1.7:1099
Open 192.168.1.7:1524
Open 192.168.1.7:2049
Open 192.168.1.7:2121
Open 192.168.1.7:3306
Open 192.168.1.7:3632
Open 192.168.1.7:5432
Open 192.168.1.7:5900
Open 192.168.1.7:6000
Open 192.168.1.7:6667
Open 192.168.1.7:6697
Open 192.168.1.7:8009
Open 192.168.1.7:8180
Open 192.168.1.7:8787
Open 192.168.1.7:45165
Open 192.168.1.7:45712
Open 192.168.1.7:54505
Open 192.168.1.7:57231
Starting Script(s)
```

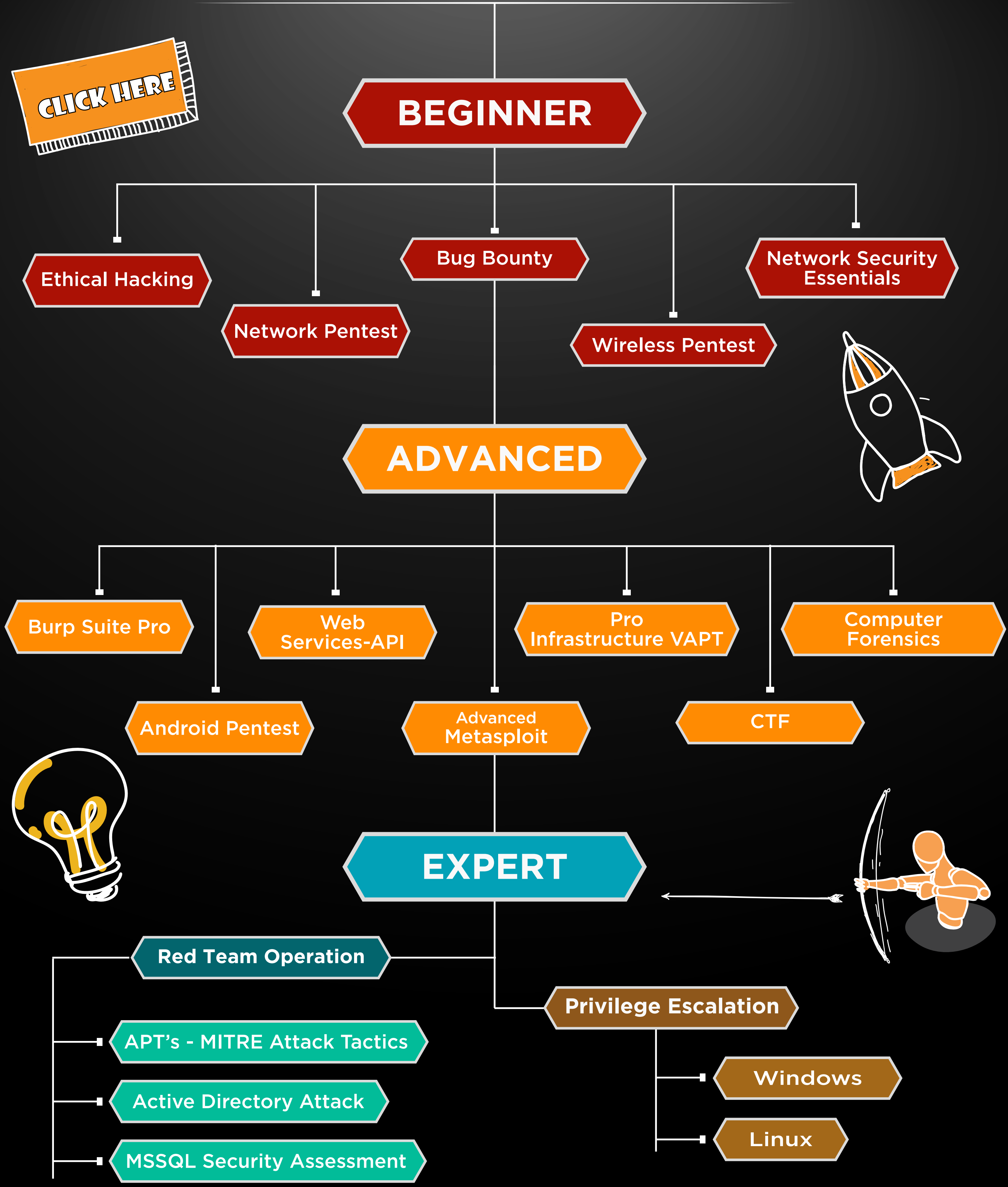
Conclusion

Rustscan represents a significant advancement in network scanning technology. Its speed, efficiency, and versatility make it an invaluable tool for cybersecurity professionals. Whether conducting routine network audits or hunting for vulnerabilities, Rustscan is a must-have in your toolkit.

iGNITE
Technologies

JOIN OUR TRAINING PROGRAMS

CLICK HERE



BEGINNER

- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials

ADVANCED

- Burp Suite Pro
- Android Pentest
- Web Services-API
- Advanced Metasploit
- Pro Infrastructure VAPT
- CTF
- Computer Forensics

EXPERT

- Red Team Operation
  - APT's - MITRE Attack Tactics
  - Active Directory Attack
  - MSSQL Security Assessment
- Privilege Escalation
  - Windows
  - Linux

www.ignitetechnologies.in