Bug Hunting Methodology and Enumeration

Summary

- Passive Recon
  - Shodan
  - Wayback Machine
  - The Harvester
- Active Recon
  - Network discovery
  - RPCClient
  - Enum4all
- List all the subdirectories and files
  - Gobuster
  - Backup File Artifacts Checker
- Web Vulnerabilities
  - Repository Github
  - Burp
  - Web Checklist
  - Nikto
  - Payment functionality

Passive recon

- Using Shodan (https://www.shodan.io/) to detect similar app

```
can be integrated with nmap (https://github.com/glennzw/shodan-hq-nse)
nmap --script shodan-hq.nse --script-args 'apikey=<yourShodanAPIKey>,target=
<hackme>'
```

- Using The Wayback Machine (https://archive.org/web/) to detect forgotten endpoints

```
look for JS files, old links
curl -sX GET "http://web.archive.org/cdx/search/cdx?url=
<targetDomain.com>&output=text&fl=original&collapse=urlkey&matchType=prefix"
```

- Using The Harvester (https://github.com/laramies/theHarvester)

```
python theHarvester.py -b all -d domain.com
```

Active recon

- Network discovery with masscan, nmap etc.

- rpcclient

```
$ rpcclient -U '%' [target host]
rpcclient $> querydominfo
Domain: WORKGROUP
Server: METASPLOITABLE
Comment: metasploitable server (Samba 3.0.20-Debian)
Total Users: 35

rpcclient $> enumdomusers
user:[games] rid:[0x3f2]
user:[nobody] rid:[0x1f5]
user:[bind] rid:[0x4ba]
```

- enum4linux

```
enum4linux v0.8.9 (http://labs.portcullis.co.uk/application/enum4linux/)
Usage: ./enum4linux.pl [options] ip
-U        get userlist
-M        get machine list*
-S        get sharelist
-P        get password policy information
-G        get group and member list
-d        be detailed, applies to -U and -S
-u user   specify username to use (default “”)
-p pass   specify password to use (default “”
-a        Do all simple enumeration (-U -S -G -P -r -o -n -i).
-o        Get OS information
-i        Get printer information
==============================
|   Users on XXX.XXX.XXX.XXX |
==============================
index: 0x1 Account: games Name: games Desc: (null)
index: 0x2 Account: nobody Name: nobody Desc: (null)
index: 0x3 Account: bind Name: (null) Desc: (null)
index: 0x4 Account: proxy Name: proxy Desc: (null)
index: 0x5 Account: syslog Name: (null) Desc: (null)
index: 0x6 Account: user Name: just a user,111,, Desc: (null)
index: 0x7 Account: www-data Name: www-data Desc: (null)
index: 0x8 Account: root Name: root Desc: (null)
```

- Zone Transfer

```
host -t ns domain.local
domain.local name server master.domain.local.

host master.domain.local
master.domain.local has address 192.168.1.1

dig axfr domain.local @192.168.1.1
```

List all the subdirectories and files

- Using BFAC (Backup File Artifacts Checker): An automated tool that checks for backup artifacts that may disclose the web-application's source code.

```
git clone https://github.com/mazen160/bfac

Check a single URL
bfac --url http://example.com/test.php --level 4

Check a list of URLs
bfac --list testing_list.txt
```

- Using DirBuster or GoBuster

```
./gobuster -u http://buffered.io/ -w words.txt -t 10
-u url
-w wordlist
-t threads

More subdomain :
./gobuster -m dns -w subdomains.txt -u google.com -i

gobuster -w wordlist -u URL -r -e
```

- Using a script to detect all phpinfo.php files in a range of IPs (CIDR can be found with a whois)

```
#!/bin/bash
for ipa in 98.13{6..9}.{0..255}.{0..255}; do
wget -t 1 -T 3 http://${ipa}/phpinfo.php; done &
```

- Using a script to detect all .htpasswd files in a range of IPs

```
#!/bin/bash
for ipa in 98.13{6..9}.{0..255}.{0..255}; do
wget -t 1 -T 3 http://${ipa}/.htpasswd; done &
```

Looking for Web vulnerabilities

- Look for private information in GitHub repos with GitRob

```
git clone https://github.com/michenriksen/gitrob.git
gitrob analyze johndoe --site=https://github.acme.com --
endpoint=https://github.acme.com/api/v3 --access-tokens=token1,token2
```

- Explore the website with a proxy (ZAP/Burp Suite)

1. Start proxy, visit the main target site and perform a Forced Browse to discover files and directories
2. Map technologies used with Wappalyzer and Burp Suite (or ZAP) proxy
3. Explore and understand available functionality, noting areas that correspond to vulnerability types

```
Burp Proxy configuration on port 8080 (in .bashrc):
alias set_proxy_burp='gsettings set org.gnome.system.proxy.http host
"http://localhost";gsettings set org.gnome.system.proxy.http port 8080;gsettings
set org.gnome.system.proxy mode "manual"'
alias set_proxy_normal='gsettings set org.gnome.system.proxy mode "none"'

then launch Burp with : java -jar burpsuite_free_v*.jar &
```

- WAHH Task Checklist copied from http://mdsec.net/wahh/tasks.html

- Subscribe to the site and pay for the additional functionality to test

- Launch a Nikto scan in case you missed something

```
nikto -h http://domain.example.com
```

- Payment functionality - @gwendallecoguic

> if the webapp you're testing uses an external payment gateway, check the doc to find the test credit numbers, purchase something and if the webapp didn't disable the test mode, it will be free

From https://stripe.com/docs/testing#cards : "Use any of the following test card numbers, a valid expiration date in the future, and any random CVC number, to create a successful payment. Each test card's billing country is set to U.S. " e.g :

Test card numbers and tokens

NUMBER	BRAND	TOKEN
4242424242424242	Visa	tok_visa
4000056655665556	Visa (debit)	tok_visa_debit
5555555555554444	Mastercard	tok_mastercard

International test card numbers and tokens

NUMBER	TOKEN	COUNTRY	BRAND
4000000400000008	tok_at	Austria (AT)	Visa
4000000560000004	tok_be	Belgium (BE)	Visa
4000002080000001	tok_dk	Denmark (DK)	Visa
4000002460000001	tok_fi	Finland (FI)	Visa
4000002500000003	tok_fr	France (FR)	Visa

References

- [BugBounty] Yahoo phpinfo.php disclosure - Patrik Fehrenbach
- Nmap CheatSheet - HackerTarget