Jesse Enriquez

Vulnerability Scanning Lab

Table of contents

Overview:

In this lab, we practiced the basics of vulnerability scans first by practicing with Kioptrix 2, with step-by-step instructions, then worked with kioptrix 3 as a homework assignment. We used multiple types of vulnerability scans such as Nmap, Nikto, and OWASP Zap, to assist us in areas where a website could be attacked. The intention of this was to identify potential access points. This would be used when being asked by an employer where a site could be vulnerable to an attack.

Effectiveness of Applications:

The potential effects could be damaging to multiple different institutions. Companies that work in finance, government, or commerce should be some of the first to instantly have a cybersecurity team run some kind of vulnerability scan on their networks. These sectors are trusted with

critical information that many trust them with. If their networks happen to be vulnerable and leave some kind of information that would gain access to a private space, that trust many had could quickly disappear. A Cyber Security team would quickly cut the loose ends of the security system put into place and only reinforce it for the better.

Potential Vulnerabilities:

Nmap*:* Will instantly reveal which ports are unsecured, which only reveals a vulnerability and a place to start. Provides information on what kind of attacks can potentially work such as MySQL, CentOS, Apache, and more if revealed.

Kioptrix 3 Website: Cannot successfully reveal a vulnerability by triggering access codes.

***Kioptrix 3 Source Page*:** Reveals potential vulnerabilities such as the programs used which could be used as a starting point to gain access to the system.

Nikto: Picks up server information, in this case, reveals the use of Apache. Reveals that the anticlickingjack X header is not present. Reveals that the X-XSS Protection Header is not defined. Shows that certain HTTP requests could also allow access to the network.

Owasp Zap: Used to perform automated scans that perform attacks that will provide all sorts of data and ways into the system. The intention of this system is to reveal vulnerabilities that would allow others access to the system.

Dangerous Website Extensions:

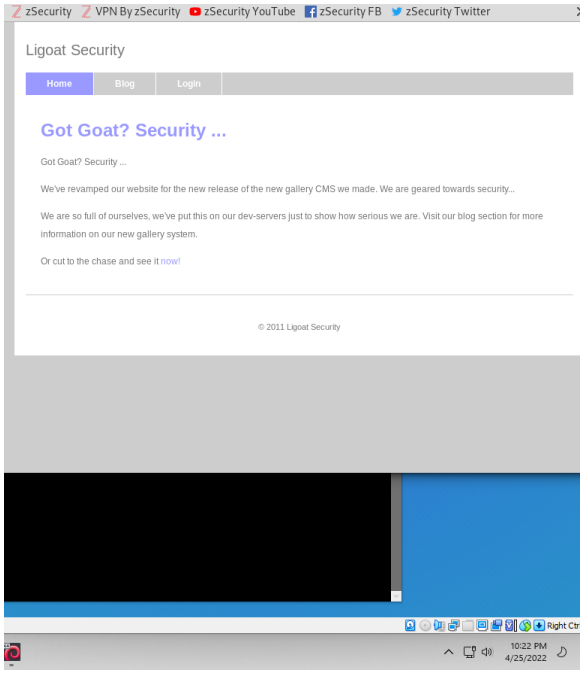
SQL Injection: This attack inserts an SQL script, which is a language used by most databases to perform quarry operations. This attack enables most attackers to pass many things such as login screenshots and access sensitive databases. Most apps are protected against attacks like this but when this vulnerability is found, it could be malicious.

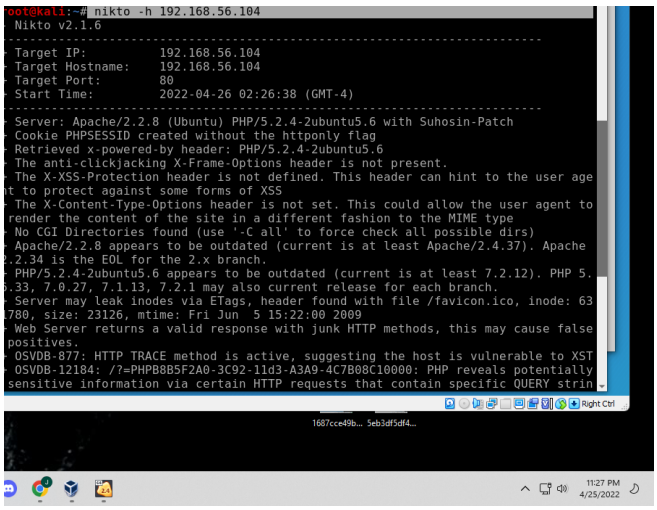
Code Injection: One of the most common attacks to take place, attackers must know the programming language that the program is written in. They can inject code into certain areas to force the web observer they are looking for. This attack requires someone to not only know the network they are trying to access quite well but the programming language as well.

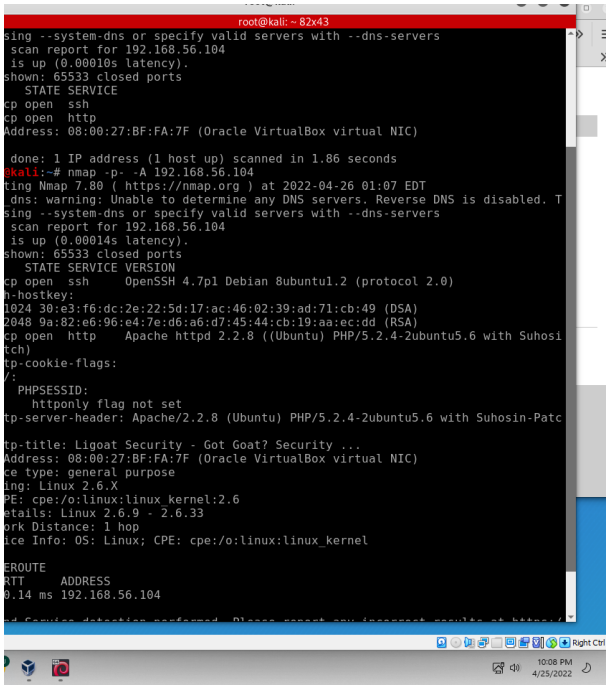
Defensive Strategy: When it comes to defensive strategies, you really have to build up a system around what you are trying to prevent. For SQL scripts, it is very important for the system not to be developed to be penetrated by SQL scripts. Most programs are far past being disturbed by my SQL scripts. As for code injections, it is crucial for the code writers to be tight with their code. If you write code that has multiple loose ends that could simply be dissected by a person that knows the same coding language as you, it's probably a satisfactory job.

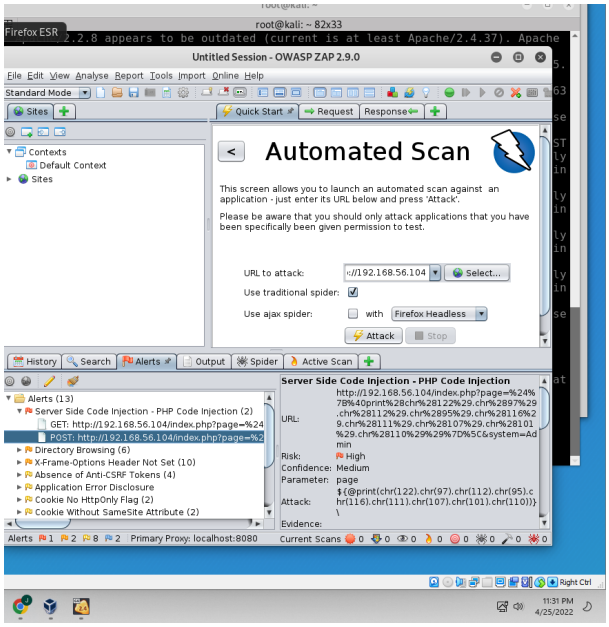
Summary: In conclusion, vulnerability testing is a crucial role in the cybersecurity world. There are many tests we can run that will reveal what kind of vulnerabilities systems and networks expose themselves to for attacks. This lab taught us a few of the tests we might run so the vulnerabilities could expose themselves.

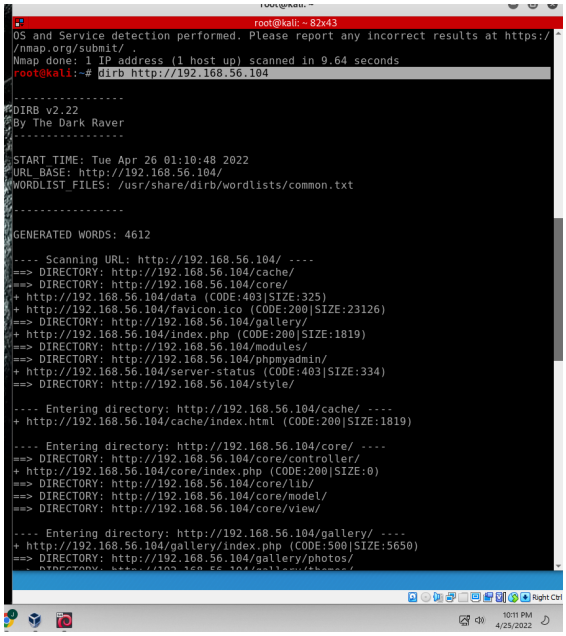
Screenshots:

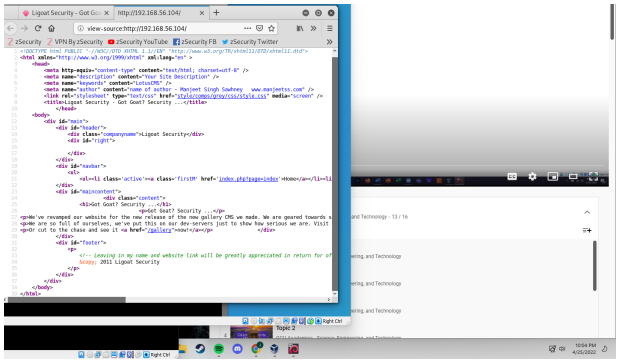

zSecurity
VPN By zSecurity
zSecurity YouTube
zSecurity FB
zSecurity Twitter
Ligoat Security
Home
Blog
Login
Got Goat? Security ...
Got Goat? Security ...
We've revamped our website for the new release of the new gallery CMS we made. We are geared towards security...
We are so full of ourselves, we've put this on our dev-servers just to show how serious we are. Visit our blog section for more information on our new gallery system.
Or cut to the chase and see it now!
© 2011 Ligoat Security


root@kali:~# nikto -h 192.168.56.104
Nikto v2.1.6
Target IP: 192.168.56.104
Target Hostname: 192.168.56.104
Target Port: 80
Start Time: 2022-04-26 02:26:38 (GMT-4)
Server: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.6 with Suhosin-Patch
Cookie PHPSESSID created without the httponly flag
Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.6
The anti-clickjacking X-Frame-Options header is not present.
The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
No CGI Directories found (use '-C all' to force check all possible dirs)
Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
PHP/5.2.4-2ubuntu5.6 appears to be outdated (current is at least 7.2.12). PHP 5.6.33, 7.0.27, 7.1.13, 7.2.1 may also current release for each branch.
Server may leak inodes via ETags, header found with file /favicon.ico, inode: 6380, size: 23126, mtime: Fri Jun 5 15:22:00 2009
Web Server returns a valid response with junk HTTP methods, this may cause false positives.
OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
OSVDB-12184: /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strin


root@kali: ~ 82x43
sing --system-dns or specify valid servers with --dns-servers
scan report for 192.168.56.104
is up (0.00010s latency).
shown: 65533 closed ports
STATE SERVICE
cp open ssh
cp open http
Address: 08:00:27:BF:FA:7F (Oracle VirtualBox virtual NIC)
done: 1 IP address (1 host up) scanned in 1.86 seconds
@kali:~# nmap -p- -A 192.168.56.104
ting Nmap 7.80 (https://nmap.org) at 2022-04-26 01:07 EDT
dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. T
sing --system-dns or specify valid servers with --dns-servers
scan report for 192.168.56.104
is up (0.00014s latency).
shown: 65533 closed ports
STATE SERVICE VERSION
cp open ssh OpenSSH 4.7p1 Debian 8ubuntu1.2 (protocol 2.0)
h-hostkey:
1024 30:e3:f6:dc:2e:22:5d:17:ac:46:02:39:ad:71:cb:49 (DSA)
2048 9a:82:e6:96:e4:7e:d6:a6:d7:45:44:cb:19:aa:ec:dd (RSA)
cp open http Apache httpd 2.2.8 ((Ubuntu) PHP/5.2.4-2ubuntu5.6 with Suhosi
tch)
tp-cookie-flags:
/:
PHPSESSID:
httponly flag not set
tp-server-header: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.6 with Suhosin-Patc
tp-title: Ligoat Security - Got Goat? Security ...
Address: 08:00:27:BF:FA:7F (Oracle VirtualBox virtual NIC)
ce type: general purpose
ing: Linux 2.6.X
PE: cpe:/o:linux:linux_kernel:2.6
etails: Linux 2.6.9 - 2.6.33
ork Distance: 1 hop
ice Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
EROUTE
RTT ADDRESS
0.14 ms 192.168.56.104


Firefox ESR
root@kali: ~ 82x33
Untitled Session - OWASP ZAP 2.9.0
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request
Response
Contexts
Default Context
Sites
Automated Scan
This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.
Please be aware that you should only attack applications that you have been specifically been given permission to test.
URL to attack:
Use traditional spider:
Use ajax spider:
with
Firefox Headless
Attack
Stop
History
Search
Alerts
Output
Spider
Active Scan
Alerts (13)
Server Side Code Injection - PHP Code Injection (2)
GET: http://192.168.56.104/index.php?page=%24
POST: http://192.168.56.104/index.php?page=%2
Directory Browsing (6)
X-Frame-Options Header Not Set (10)
Absence of Anti-CSRF Tokens (4)
Application Error Disclosure
Cookie No HttpOnly Flag (2)
Cookie Without SameSite Attribute (2)
Server Side Code Injection - PHP Code Injection
URL:
Risk: High
Confidence: Medium
Parameter: page
Attack:
Evidence:
Alerts 1 2 8 2 Primary Proxy: localhost:8080
Current Scans


root@kali: ~ 82x43
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.64 seconds
root@kali:~# dirb http://192.168.56.104
DIRB v2.22
By The Dark Raver
START_TIME: Tue Apr 26 01:10:48 2022
URL_BASE: http://192.168.56.104/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
GENERATED WORDS: 4612
---- Scanning URL: http://192.168.56.104/ ----
==> DIRECTORY: http://192.168.56.104/cache/
==> DIRECTORY: http://192.168.56.104/core/
+ http://192.168.56.104/data (CODE:403|SIZE:325)
+ http://192.168.56.104/favicon.ico (CODE:200|SIZE:23126)
==> DIRECTORY: http://192.168.56.104/gallery/
+ http://192.168.56.104/index.php (CODE:200|SIZE:1819)
==> DIRECTORY: http://192.168.56.104/modules/
==> DIRECTORY: http://192.168.56.104/phpmyadmin/
+ http://192.168.56.104/server-status (CODE:403|SIZE:334)
==> DIRECTORY: http://192.168.56.104/style/
---- Entering directory: http://192.168.56.104/cache/ ----
+ http://192.168.56.104/cache/index.html (CODE:200|SIZE:1819)
---- Entering directory: http://192.168.56.104/core/ ----
==> DIRECTORY: http://192.168.56.104/core/controller/
+ http://192.168.56.104/core/index.php (CODE:200|SIZE:0)
==> DIRECTORY: http://192.168.56.104/core/lib/
==> DIRECTORY: http://192.168.56.104/core/model/
==> DIRECTORY: http://192.168.56.104/core/view/
---- Entering directory: http://192.168.56.104/gallery/ ----
+ http://192.168.56.104/gallery/index.php (CODE:500|SIZE:5650)
==> DIRECTORY: http://192.168.56.104/gallery/photos/
Right Ctrl
10:11 PM
4/25/2022


Ligoat Security - Got Goa
http://192.168.56.104/
view-source:http://192.168.56.104/
zSecurity
VPN By zSecurity
zSecurity YouTube
zSecurity FB
zSecurity Twitter
Right Ctrl
10:04 PM
4/25/2022