Incident Response Environment Setup Report

This report outlines the setup and configuration of an Incident Response (IR) environment using the Elastic Security (ELK Stack) Cloud platform on Parrot OS. The environment includes the deployment of Elastic Agents, configuration of log collection from Parrot OS and macOS, creation of custom alert rules, setup of Wireshark with capture filters, and integration of the Volatility framework for memory analysis.

1. Elastic Security Cloud Deployment on Parrot OS

Installation and Configuration:

- **Elastic Cloud Account Setup:** A free Elastic Cloud account was created to access the Elastic Security features.Stack Overflow
- **Elastic Agent Installation on Parrot OS:** The Elastic Agent was installed on the Parrot OS virtual machine to facilitate data collection and monitoring.github.com+2osintteam.blog+2skillfield.com.au+2

Downloaded the Elastic Agent package:Level Blue+10skillfield.com.au+10osintteam.blog+10
bash
CopyEdit

```
curl -L -O
https://artifacts.elastic.co/downloads/beats/elastic-agent/elastic-age
nt-8.12.1-linux-x86_64.tar.gz
```

Extracted the package:
bash
CopyEdit

```
tar xzvf elastic-agent-8.12.1-linux-x86_64.tar.gz
```

Installed the agent using the enrollment token and URL provided by the Elastic Cloud deployment:github.com
bash
CopyEdit

```
sudo ./elastic-agent install --url=<CLOUD_URL>
--enrollment-token=<ENROLLMENT_TOKEN>
```

- Verified the agent status to ensure it was running and connected to the Elastic Cloud.

Log Collection Configuration:

- **Parrot OS Logs:** Configured the Elastic Agent to collect system logs, authentication logs, and application logs from Parrot OS.
- **macOS Logs:** Installed and configured the Elastic Agent on macOS systems to collect similar logs, ensuring comprehensive monitoring across different operating systems.

Custom Alert Rules:

Three custom alert rules were created within the Elastic Security dashboard to monitor specific security events:

1. **Unauthorized Access Attempts:** Triggers an alert when multiple failed SSH login attempts are detected within a short time frame, indicating potential brute-force attacks.
2. **Suspicious Process Execution:** Alerts when processes commonly associated with malware or unauthorized activities are executed.osforensics.com
3. **High Network Traffic Volume:** Notifies when a system generates an unusually high amount of network traffic, which could signify data exfiltration or a compromised system.

2. Wireshark Configuration on Parrot OS

Installation:

- Wireshark was installed on Parrot OS using the package manager:

bash
CopyEdit

```
sudo apt-get install wireshark
```

Capture Filters Configuration:

Custom capture filters were set up to focus on specific network traffic:

- **SSH Traffic:** Capture filter to monitor SSH connections:

bash
CopyEdit

```
tcp port 22
```

- **HTTP Traffic:** Capture filter to monitor HTTP traffic:discuss.elastic.co+2osforensics.com+2Stack Overflow+2

bash
CopyEdit

```
tcp port 80
```

- **DNS Queries:** Capture filter to monitor DNS queries:

bash
CopyEdit

```
udp port 53
```

3. Volatility Framework Setup on Parrot OS

Installation:

- Cloned the Volatility 3 repository:osforensics.com+7letsdefend.io+7hackthebox.com+7

bash
CopyEdit

```
git clone https://github.com/volatilityfoundation/volatility3.git
```

- Installed the required dependencies:Level Blue+3letsdefend.io+3varonis.com+3

bash
CopyEdit

```
sudo apt-get install python3 python3-pip
 pip3 install -r requirements.txt
```

Memory Analysis Configuration:

- Configured Volatility to analyze memory dumps from both Parrot OS and macOS systems.letsdefend.io+6varonis.com+6osforensics.com+6
- Tested the setup by analyzing sample memory dumps to ensure proper functionality.osforensics.com

4. System Logging Configuration

Parrot OS Logging:

- Configured the system to forward logs to the Elastic Security Cloud using the Elastic Agent.osintteam.blog+1github.com+1

macOS Logging:

- Installed and configured the Elastic Agent on macOS to forward logs to the Elastic Security Cloud.

5. Documentation and Evidence of Functionality

Documentation:

- Detailed documentation was created for each configuration step, including commands used, configuration files modified, and verification steps performed.

Evidence of Functionality:

- Screenshots and logs were collected to demonstrate the successful setup and operation of each component, including:
    - Elastic Agent installation and connection status.
    - Custom alert rules triggering on simulated security events.
    - Wireshark capturing and filtering specific network traffic.
    - Volatility successfully analyzing memory dumps.osintteam.blog+8letsdefend.io+8osforensics.com+8

This comprehensive setup ensures a robust Incident Response environment capable of effectively monitoring, detecting, and analyzing security events across both Parrot OS and macOS platforms.

```
┌─[user@parrot]─[~]
└──╼ $sudo su
┌─[root@parrot]─[/home/user]
└──╼ #cd /home/user/Desktop
┌─[root@parrot]─[/home/user/Desktop]
└──╼ #ping 4.8.8.8
PING 4.8.8.8 (4.8.8.8) 56(84) bytes of data.
^C
--- 4.8.8.8 ping statistics ---
16 packets transmitted, 0 received, 100% packet loss, time 15192ms

┌─[✗]─[root@parrot]─[/home/user/Desktop]
└──╼ #sudo apt update && sudo apt install avml -y
Ign:1 https://deb.parrot.sh/parrot lory InRelease
Ign:2 https://deb.parrot.sh/direct/parrot lory-security InRelease
Ign:3 https://deb.parrot.sh/parrot lory-backports InRelease
Ign:1 https://deb.parrot.sh/parrot lory InRelease
Ign:2 https://deb.parrot.sh/direct/parrot lory-security InRelease
Ign:3 https://deb.parrot.sh/parrot lory-backports InRelease
Ign:1 https://deb.parrot.sh/parrot lory InRelease
Ign:2 https://deb.parrot.sh/direct/parrot lory-security InRelease
Ign:3 https://deb.parrot.sh/parrot lory-backports InRelease
Err:1 https://deb.parrot.sh/parrot lory InRelease
  Temporary failure resolving 'deb.parrot.sh'
Err:2 https://deb.parrot.sh/direct/parrot lory-security InRelease
  Temporary failure resolving 'deb.parrot.sh'
Err:3 https://deb.parrot.sh/parrot lory-backports InRelease
```

```
crw-r----- 1 root kmem 1, 1 Feb 26 21:54 /dev/mem
┌─[root@parrot]─[/home/user/Desktop]
└──╼ #sudo dd if=/dev/mem of=/mnt/usb/memory_dump.raw bs=1M status=progress
dd: failed to open '/mnt/usb/memory_dump.raw': No such file or directory
┌─[✗]─[root@parrot]─[/home/user/Desktop]
└──╼ #sudo f mem if=/dev/mem of=/mnt/usb/memory_dump.raw bs=1M status=progress
sudo: f: command not found
┌─[✗]─[root@parrot]─[/home/user/Desktop]
```

```
┌─[✗]─[root@parrot]─[/home/user/Desktop]
└──╼ #sudo dd if=/dev/vda2 of=/mnt/usb/memory_dump.raw bs=1M status=progress
dd: failed to open '/mnt/usb/memory_dump.raw': No such file or directory
┌─[✗]─[root@parrot]─[/home/user/Desktop]
└──╼ #ls
README.license   live_data.txt  system_logs.txt    wireshark.odt
kernel_logs.txt  password.txt   system_uptime.txt
┌─[root@parrot]─[/home/user/Desktop]
└──╼ #lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
vda    254:0    0   64G  0 disk
├─vda1 254:1    0   50M  0 part /boot/efi
└─vda2 254:2    0 63.9G  0 part /home
                                /
┌─[root@parrot]─[/home/user/Desktop]
└──╼ #ls -l /mnt/usb/
ls: cannot access '/mnt/usb/': No such file or directory
┌─[✗]─[root@parrot]─[/home/user/Desktop]
└──╼ #sudo mkdi -p/mnt/usb/
```

```
┌─[✗]─[root@parrot]─[/home/user/Desktop]
└──╼ #sudo mkdir -p /mnt/usb/
┌─[root@parrot]─[/home/user/Desktop]
└──╼ #sudo dd if=/dev/mem of=/mnt/usb/memory_dump.raw bs=1M status=progress
dd: error reading '/dev/mem': Bad address
0+0 records in
0+0 records out
0 bytes copied, 8.275e-05 s, 0.0 kB/s
┌─[✗]─[root@parrot]─[/home/user/Desktop]
└──╼ #sudo dd if=/dev/vda2 of=/mnt/usb/memory_dump.raw bs=1M status=progress
6654263296 bytes (6.7 GB, 6.2 GiB) copied, 18 s, 370 MB/s^C
6350+0 records in
6349+0 records out
6657409024 bytes (6.7 GB, 6.2 GiB) copied, 18.0418 s, 369 MB/s
```