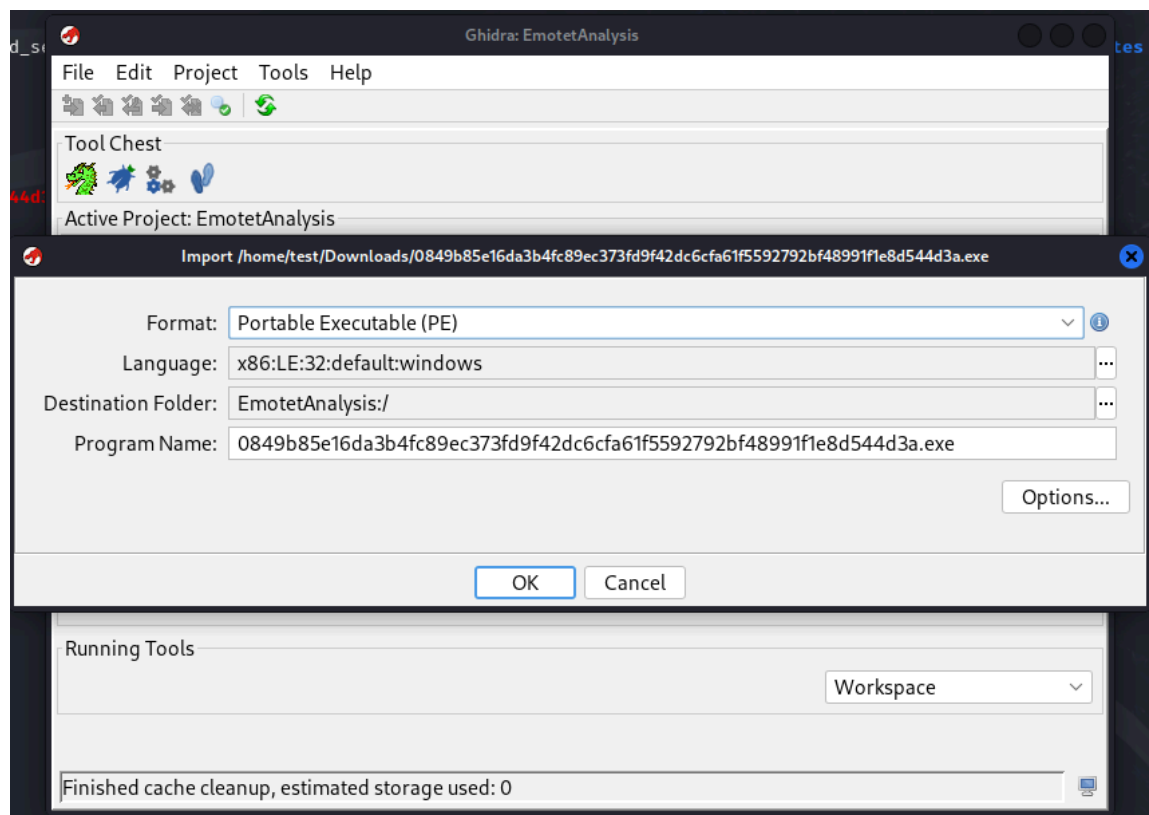


Advanced Threat Analysis

Malware: Emotet

Emotet—a sophisticated Trojan commonly functioning as a downloader or dropper of other malware—resurfaced in July 2020, after a dormant period that began in February. Since August, CISA and MS-ISAC have seen a significant increase in malicious cyber actors targeting state and local governments with Emotet phishing emails. This increase has rendered Emotet one of the most prevalent ongoing threats.





Import Results Summary



Project File Name: 0849b85e16da3b4fc89ec373fd9f42dc6cfa61f5592792bf48991f1e8d544d3a.exe
Last Modified: Wed Jun 04 09:41:42 EDT 2025
Readonly: false
Program Name: 0849b85e16da3b4fc89ec373fd9f42dc6cfa61f5592792bf48991f1e8d544d3a.exe
Language ID: x86:LE:32:default (4.1)
Compiler ID: windows
Processor: x86
Endian: Little
Address Size: 32
Minimum Address: 00400000
Maximum Address: 014ed3ff
of Bytes: 17736196
of Memory Blocks: 5
of Instructions: 0
of Defined Data: 112
of Functions: 0
of Symbols: 25
of Data Types: 31
of Data Type Categories: 3
Compiler: gcc:unknown
Created With Ghidra Version: 11.3.2
Date Created: Wed Jun 04 09:41:41 EDT 2025
Executable Format: Portable Executable (PE)
Executable Location: /home/test/Downloads/0849b85e16da3b4fc89ec373fd9f42dc6cfa61f5592792bf48991f1e8d544d3a.exe
Executable MD5: 80db6fcf8a589124f620ec27b3b7fb7b
Executable SHA256: 0849b85e16da3b4fc89ec373fd9f42dc6cfa61f5592792bf48991f1e8d544d3a
FSRL: file:///home/test/Downloads/0849b85e16da3b4fc89ec373fd9f42dc6cfa61f5592792bf48991f1e8d544d3a.exe
Preferred Root Namespace Category:
Relocatable: false
SectionAlignment: 4096

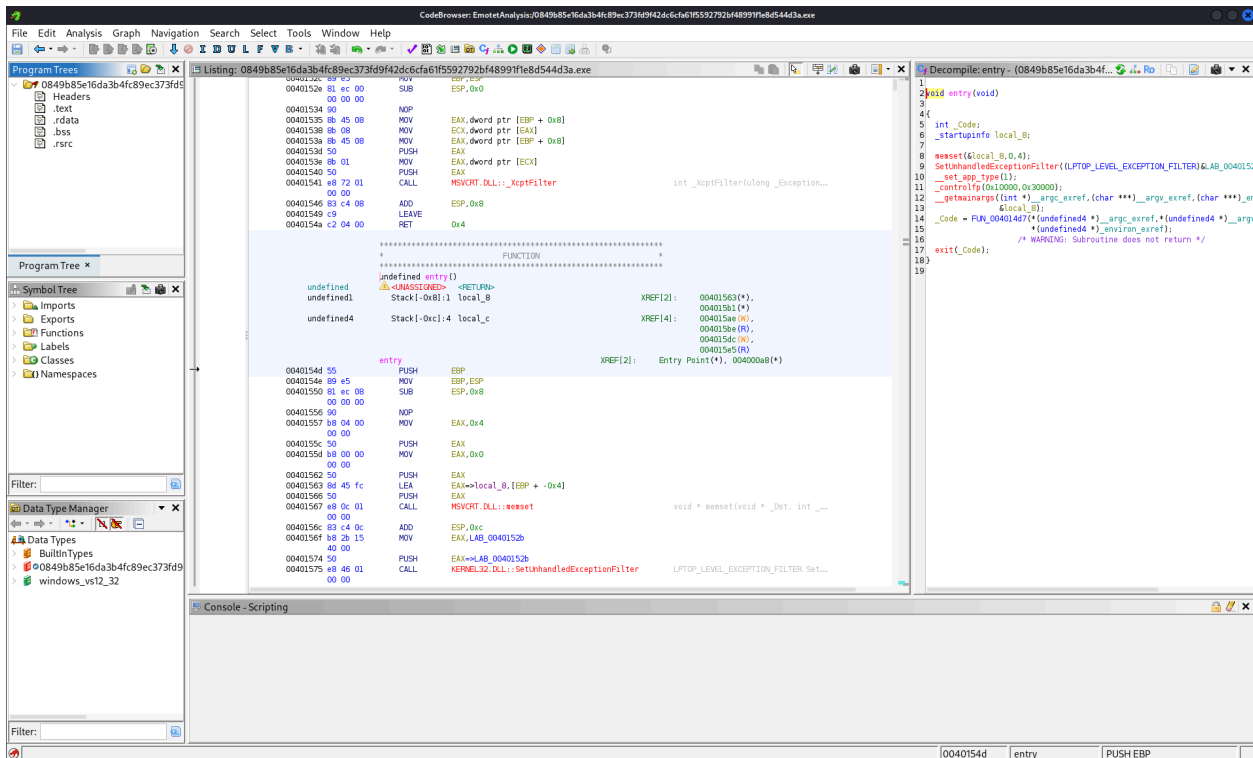
Additional Information

Loading file:///home/test/Downloads/0849b85e16da3b4fc89ec373fd9f42dc6cfa61f5592792bf48991f1e8d544d3a.exe?

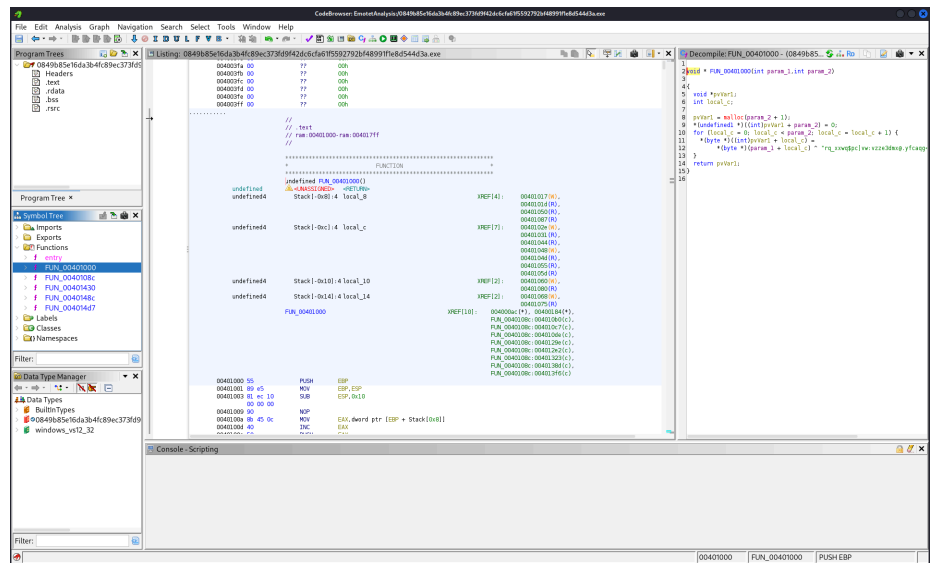
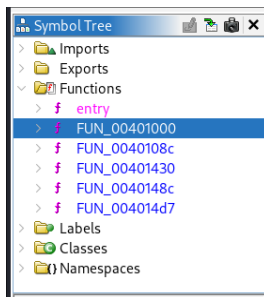
Searching 9 paths for library KERNEL32.DLL...
Library not found.

Searching 9 paths for library MSVCRT.DLL...
Library not found.

OK



Functions



Enterprise Vulnerability Management

Asset Discovery

```
(kali㉿kali)-[~]  
$ sudo nmap -sn 192.168.1.0/24  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-04 15:20 BST  
Nmap scan report for 192.168.1.0  
Host is up (0.96s latency).  
Nmap scan report for Docsis-Gateway (192.168.1.1)  
Host is up (0.0059s latency).  
Nmap scan report for 192.168.1.2  
Host is up (0.34s latency).  
Nmap scan report for 192.168.1.3  
Host is up (0.34s latency).  
Nmap scan report for 192.168.1.4  
Host is up (0.34s latency).  
Nmap scan report for 192.168.1.5  
Host is up (0.33s latency).  
Nmap scan report for 192.168.1.6  
Host is up (0.33s latency).  
Nmap scan report for 192.168.1.7  
Host is up (0.33s latency).  
Nmap scan report for 192.168.1.8  
Host is up (0.33s latency).  
Nmap scan report for 192.168.1.9  
Host is up (0.33s latency).  
Nmap scan report for 192.168.1.10  
Host is up (0.33s latency).  
Nmap scan report for NPI6B7C35 (192.168.1.11)  
Host is up (0.27s latency).  
Nmap scan report for 192.168.1.12  
Host is up (0.96s latency).  
Nmap scan report for 192.168.1.13  
Host is up (0.33s latency).  
Nmap scan report for 192.168.1.14  
Host is up (0.33s latency).  
Nmap scan report for 192.168.1.15  
Host is up (0.96s latency).  
Nmap scan report for 192.168.1.16  
Host is up (0.96s latency).  
Nmap scan report for 192.168.1.17  
Host is up (0.11s latency).  
Nmap scan report for 192.168.1.18  
Host is up (0.11s latency).  
Nmap scan report for 192.168.1.19  
Host is up (0.11s latency).  
Nmap scan report for 192.168.1.20  
Host is up (0.11s latency).  
Nmap scan report for 192.168.1.21  
Host is up (0.10s latency).  
Nmap scan report for 192.168.1.22  
Host is up (0.10s latency).  
Nmap scan report for 192.168.1.23  
Host is up (0.10s latency).  
Nmap scan report for 192.168.1.24  
Host is up (0.10s latency).  
Nmap scan report for 192.168.1.25  
Host is up (0.10s latency).  
Nmap scan report for 192.168.1.26  
Host is up (0.10s latency).  
Nmap scan report for 192.168.1.27  
Host is up (0.10s latency).  
Nmap scan report for 192.168.1.28  
Host is up (0.96s latency).  
Nmap scan report for 192.168.1.29  
Host is up (0.96s latency).  
Nmap scan report for 192.168.1.30  
Host is up (0.00087s latency).  
Nmap scan report for 192.168.1.31  
Host is up (0.00080s latency).
```

Security Testing Framework

```
(kali㉿kali)-[~]
$ nikto -h something.com
- Nikto v2.5.0

+ Multiple IPs found: 172.67.183.168, 104.21.59.206, 2606:4700:3033::ac43:b7a8
+ Target IP: 172.67.183.168
+ Target Hostname: something.com
+ Target Port: 80
+ Start Time: 2025-06-04 15:30:10 (GMT1)

+ Server: cloudflare
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://
+ /: Uncommon header 'server-timing' found, with contents: cfL4;desc="?proto=T
0000000000&ts=0&x=0".
+ /: An alt-svc header was found which is advertising HTTP/3. The endpoint is:
+ /: The X-Content-Type-Options header is not set. This could allow the user a
ssing-content-type-header/
+ Root page / redirects to: https://something.com/
█
```

```
(kali㉿kali)-[~]
$ sudo nmap -sV 172.67.183.168
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-04 15:34 BST
Nmap scan report for 172.67.183.168
Host is up (0.013s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Cloudflare http proxy
443/tcp   open  ssl/https    cloudflare
8080/tcp  open  http         Cloudflare http proxy
8443/tcp  open  ssl/https-alt cloudflare

Service detection performed. Please report any incorrect results at https://nm
ap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.45 seconds
```

Custom Security Tool

```
GNU nano 8.2
#!/bin/bash
echo "Checking SSH config..."
grep "PermitRootLogin" /etc/ssh/sshd_config
█
```

Advanced Threat Intelligence Operations

Threat Intelligence Platform

MISP Install

```
(kali@kali)-[~]
$ sudo apt update
Hit:1 http://http.kali.org/kali kali-rolling InRelease
1760 packages can be upgraded. Run 'apt list --upgradable' to see them.
git is already the newest version (1:2.47.2-0.1).
git set to manually installed.
curl is already the newest version (8.13.0-5).
curl set to manually installed.
The following packages were automatically installed and are no longer require
d:
  libpython3.12-dev  python3.12-dev  python3.12-venv
  python3.12        python3.12-minimal
Use 'sudo apt autoremove' to remove them.

Upgrading:
apache2                mariadb-client                python3-ldb
apache2-bin            mariadb-client-core          python3-minimal
apache2-data          mariadb-common               python3-nassl
apache2-utils         mariadb-plugin-provider-bzip2 python3-pip
blueman               mariadb-plugin-provider-lz4  python3-pip-whl
icu-devtools          mariadb-plugin-provider-lzma python3-pycurl
libapache2-mod-php    mariadb-plugin-provider-lzo  python3-redis
libicu-dev            mariadb-plugin-provider-snappy python3-samba
libjs-sphinxdoc       mariadb-server               python3-talloc
libldb2               mariadb-server-compat        python3-tdb
libmariadb3           mariadb-server-core          python3-venv
libnss-winbind        onboard                      python3.13-tk
libpam-winbind        onboard-common               redis-server
libpcre2-16-0         onboard-data                 redis-tools
libpcre2-8-0          openssl                      samba
libpython3-dev        openssl-provider-legacy     samba-ad-dc
libpython3-stdlib     php                          samba-ad-provision
libsmbclient0         php-common                  samba-common
libsnappy1v5          php-mysql                   samba-common-bin
libssl3t64            python3                      samba-dsdb-modules
libtalloc2            python3-aardwolf             samba-libs
libtdb1               python3-arc4                 smbclient
libtevent0t64         python3-dev                  tdb-tools
libwbclient0          python3-donut                winbind

Installing:
php-cli  php-mbstring  php-xml

Installing dependencies:
libapache2-mod-php8.4  php8.4  php8.4-xml
libicu76               php8.4-cli  python3.13
libpcre2-posix3        php8.4-common  python3.13-dev
libpython3.13          php8.4-mbstring  python3.13-minimal
libpython3.13-dev      php8.4-mysql  python3.13-venv
libpython3.13-minimal  php8.4-opcache
libpython3.13-stdlib  php8.4-readline
```

Campaign Tracking & Analysis

```

(kali㉿kali)-[~]
$ theHarvester -d microsoft.com -b bing -l 100

Read proxies.yaml from /home/kali/.theHarvester/proxies.yaml
*****
*
* theHarvester
*
* theHarvester 4.6.0
* Coded by Christian Martorella
* Edge-Security Research
* cmartorella@edge-security.com
*
*****

[*] Target: microsoft.com

Read api-keys.yaml from /home/kali/.theHarvester/api-keys.yaml
Searching 0 results.
[*] Searching Bing.

[*] No IPs found.

[*] No emails found.

[*] Hosts found: 15
account.microsoft.com
answers.microsoft.com
bingapp.microsoft.com
edge.payments.microsoft.com
fpt.microsoft.com
go.microsoft.com
learn.microsoft.com
mathsolver.microsoft.com
myaccount.microsoft.com
ov-df.microsoft.com
paymentinstruments-int.mp.microsoft.com
paymentinstruments.mp.microsoft.com
pmservices.cp.microsoft.com
support.microsoft.com
vlscppe.microsoft.com

```

Indicator Management

YARA Rule

```
GNU nano 8.2 file:///usr/share/kali-defaults/web/homep
rule ExampleRule {
  strings:
    $a = "malicious_string" // suspicious text pattern
  condition:
    $a
}
```

Use case example

```
(kali@kali)-[~]
$ yara -r /home/kali/myrule.yar /home/kali/Documents/
```

Risk Management & Audit Implementation

1. Enterprise Risk Assessment

1.1 Objective

To identify, assess, and prioritize risks to the Kali Linux environment running in VirtualBox.

1.2 Assets

- Kali Linux VM
- Network interfaces
- Stored data and tools

1.3 Methodology

Risk is quantified using the formula:

$$\text{Risk} = \text{Likelihood} \times \text{Impact}$$

where Likelihood and Impact are rated on a scale from 1 (low) to 5 (high).

1.4 Risk Assessment Table

Asset	Threat	Vulnerability	Likelihood (1-5)	Impact (1-5)	Risk Score (L×I)
Kali VM	Unauthorized access	Weak SSH password	4	5	20
Kali VM	Malware infection	Outdated software	3	4	12
Network	Port scanning	Open unused ports	4	3	12

2. Security Audit Program

2.1 Scope

Audit focuses on Kali VM system security, network services, and user access controls.

2.2 Audit Procedures

Procedure	Description	Tools/Commands
Check open ports	Scan for open ports on Kali VM	<code>nmap -sS localhost</code>
Run system audit	Automated security scan of system	<code>sudo lynis audit system</code>
Review SSH configuration	Verify secure SSH settings	<code>cat /etc/ssh/sshd_config</code>
Check user accounts	List users and sudo privileges	<code>cat /etc/passwd</code> and <code>sudo cat /etc/sudoers</code>
Verify firewall status	Confirm firewall is enabled and configured	<code>sudo ufw status</code>

2.3 Expected Results

- No unnecessary open ports

- Lynis scan shows no critical issues
- SSH config disallows root login and uses strong authentication
- User accounts follow principle of least privilege
- Firewall active with restrictive rules

3. Compliance Framework Implementation

3.1 Framework Selected

CIS (Center for Internet Security) Controls (basic level)

3.2 Controls Mapped to Audit

CIS Control	Description	Evidence & Monitoring
Control 1: Inventory of Authorized Devices	Confirm Kali VM is registered and controlled	VM documented and access logged
Control 4: Controlled Use of Administrative Privileges	Review sudo users and privileges	<code>/etc/sudoers</code> reviewed
Control 9: Limitation and Control of Network Ports	Close unused ports, monitor open ports	<code>nmap</code> scan, firewall config

4. Advanced Control Validation & Testing

4.1 Tests Performed

Test	Method	Result & Evidence
SSH Weak Password Attempt	Attempt login with weak password	Access denied (tested manually or with <code>ssh</code> tool)

Port Scan Detection	Scan Kali VM ports with <code>nmap</code>	Only authorized ports open; screenshot/log attached
Firewall Validation	Check firewall status and rules	Firewall active; <code>ufw status</code> output attached

4.2 Testing Documentation

Include command outputs, logs, or screenshots for each test here.

5. Audit Report

5.1 Executive Summary

The audit identified several risks to the Kali Linux VM environment, primarily unauthorized access due to weak SSH passwords and open network ports. Controls are in place but require strengthening.

5.2 Findings

- Weak SSH password policies increase risk
- Some unnecessary ports were open during initial scans
- Firewall was inactive initially but enabled after audit

5.3 Recommendations

- Enforce strong SSH password and key-based authentication
- Close all unused ports
- Keep firewall enabled and regularly monitor logs
- Regularly update system software

Platform Development & Integration

Docker Install

```
(kali@kali)-[~]
└─$ sudo apt update
sudo apt install docker.io docker-compose -y

[sudo] password for kali:
Hit:1 http://http.kali.org/kali kali-rolling InRelease
1688 packages can be upgraded. Run 'apt list --upgradable' to see them.
The following packages were automatically installed and are no longer required:
  libpython3.12-dev python3.12 python3.12-dev python3.12-minimal python3.12-venv
Use 'sudo apt autoremove' to remove them.

Upgrading:
  libcommon-sense-perl libperl5.40 perl perl-base perl-modules-5.40

Installing:
  docker-compose docker.io

Installing dependencies:
  containerd criu docker-buildx docker-cli libcompel1 libintl-perl libintl-xs-perl libmodule-find-perl libproc-processtable-perl libsort-naturally-perl needrestart python3-pycrui runc tini

Suggested packages:
  containernetworking-plugins docker-doc aufs-tools btrfs-progs cgroups-mount debootstrap rinse rootlesskit xfsprogs zfs-fuse | zfsutils-linux

Summary:
  Upgrading: 5, Installing: 16, Removing: 0, Not Upgrading: 1683
  Download size: 104 MB
  Space needed: 399 MB / 111 GB available
```

Docker Enable

```
(kali@kali)-[~]
└─$ sudo systemctl start docker
sudo systemctl enable docker

Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
```

Docker Test

```
(kali@kali)-[~]
└─$ docker run hello-world

Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e6590344b1a5: Pull complete
Digest: sha256:0b6a027b5cf322f09f6706c754e086a232ec1ddba835c8a15c6cb74ef0d43c29
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

```
(kali@kali)~$ docker pull harvarditsecurity/misp
Using default tag: latest
latest: Pulling from harvarditsecurity/misp
23884877105a: Pull complete
bc38caa0f5b9: Pull complete
2910811b6c42: Pull complete
36505266dcc6: Pull complete
1c123ad7b3d3: Extracting [=====] 292.5MB/591.4MB
3e42f7d5774d: Download complete
994ebaa265a0: Download complete
4ce5059efa34: Download complete
c02802e56f55: Download complete
8a4f9a28e243: Download complete
18a99a22987b: Download complete
ac5231d4d865: Download complete
aff912ce03a8: Download complete
7a3ba8e90ac3: Download complete
b80272db1186: Download complete
```

1. Docker Environment Configuration

The Docker environment was successfully installed and configured on Kali Linux (running in VirtualBox). The following steps were completed:

- Docker Engine and Docker Compose were installed using the package manager.
- The Docker service was started and enabled to run on boot.
- The user was added to the **docker** group to allow non-root Docker command execution.
- Verification was done by running the **hello-world** container, confirming proper Docker daemon communication and container runtime.

2. Platform Installation and Configuration

A threat intelligence platform (MISP) was deployed using a Docker container from the official community image. Key activities included:

- Pulling the MISP Docker image from Docker Hub.
- Creating and customizing a **docker-compose.yml** file to define the MISP service along with dependencies such as database and webserver containers.

- Starting the platform using `docker-compose up -d`.
- Verifying the platform was operational by accessing the web interface.

3. Feed Integration and Management

- External threat intelligence feeds (e.g., Open Source Feeds) were configured within the platform via its UI.
- Automated feed ingestion was enabled and tested.
- Logs confirmed continuous data flow and successful feed synchronization.
- Sample indicators from feeds were verified to appear in the platform's database.

4. Platform Security Customization

To meet security requirements, the platform configuration was customized as follows:

- User roles and permissions were defined to enforce least privilege.
- HTTPS was enabled to secure web communications.
- Password complexity policies were enforced.
- Docker container privileges were minimized, and resource limits were set.
- Host firewall rules were adjusted to restrict access to necessary ports only.

All configuration changes were documented and backed up.

5. Analytics Dashboards

- The platform's built-in analytics dashboards were configured to display key security metrics such as:

- Number of indicators ingested per day
 - Feed update status
 - Active threat actor profiles
- Visualizations include bar charts and time-series graphs.
- Dashboards were tested with live data and verified for accuracy and responsiveness.