

## Project 2: Data Breaching Investigation

Using TryhackMe: <https://tryhackme.com/>

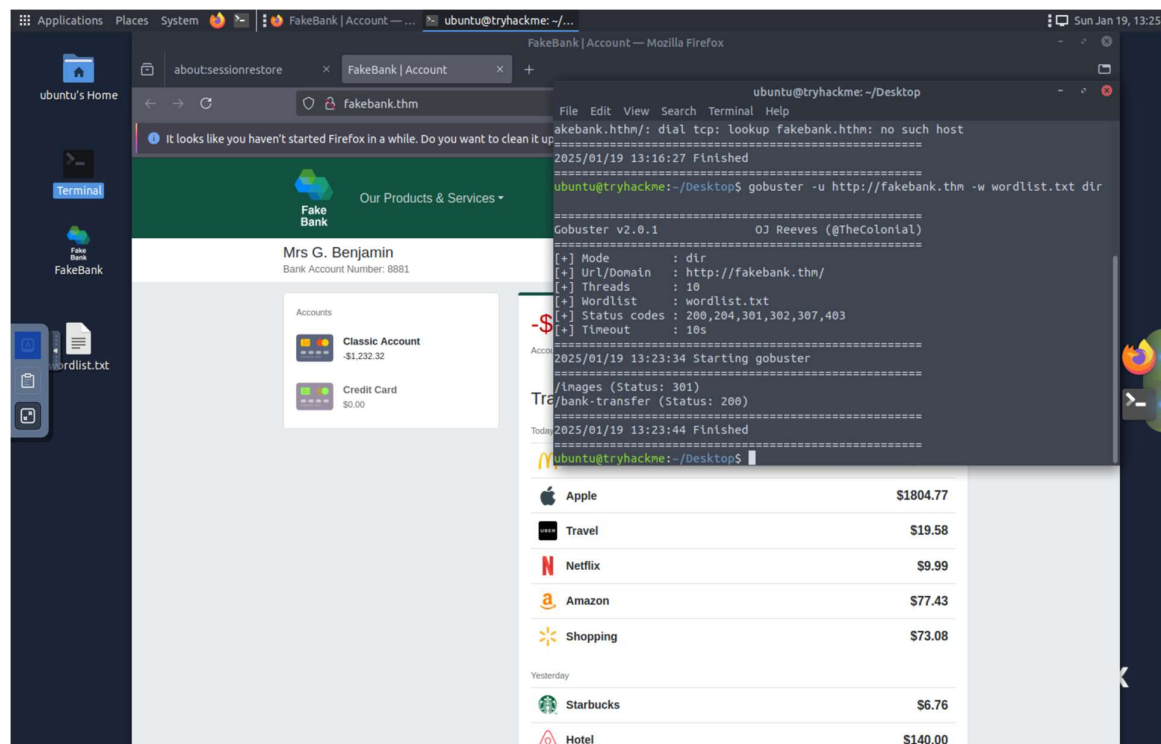
**Offensive security:** which aims to identify and exploit system vulnerabilities to enhance security measures.

Includes: Exploiting software bugs | Leveraging Insecure setups | taking advantages unenforced access control policies

Here, “gobuster” is used to identify potential hidden pages where the confidential data can be accessed!!, TryhackMe applied the gobuster on “fake bank | account”.

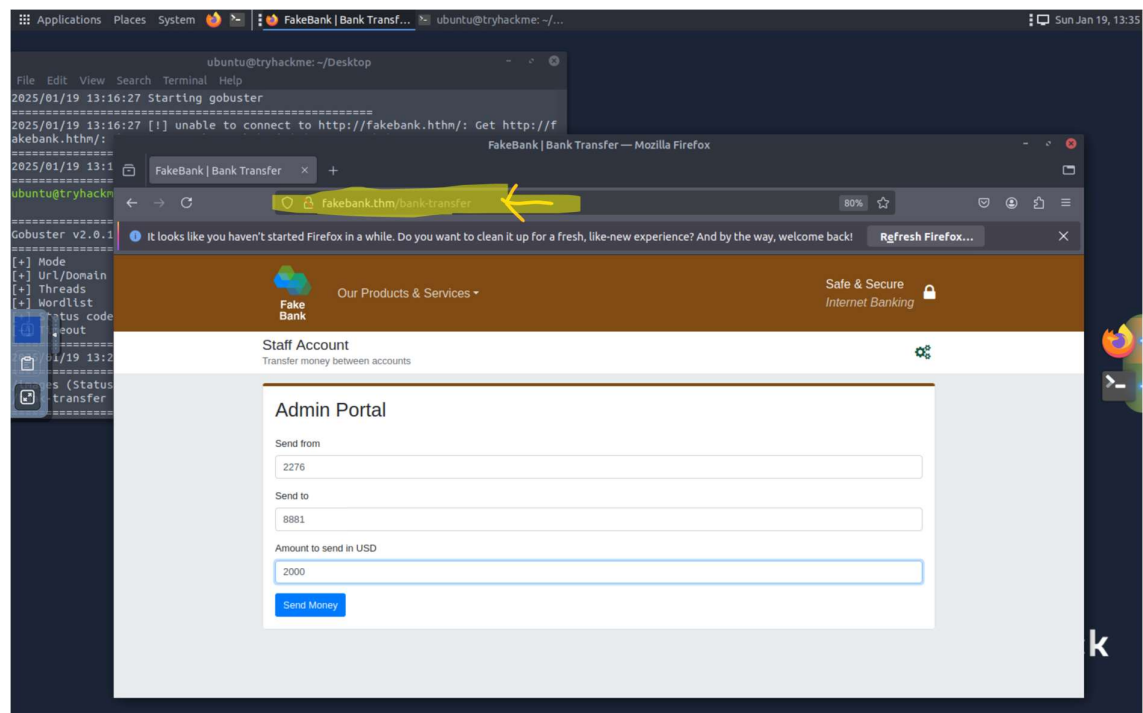
STEPS:

1. In CLI: “gobuster -u <http://fakebank.thm> -w wordlist.txt dir”  
Gobuster found out the pages in which the one is “/bank-transfer” [secret bank transfer page] , let’s hack the bank.

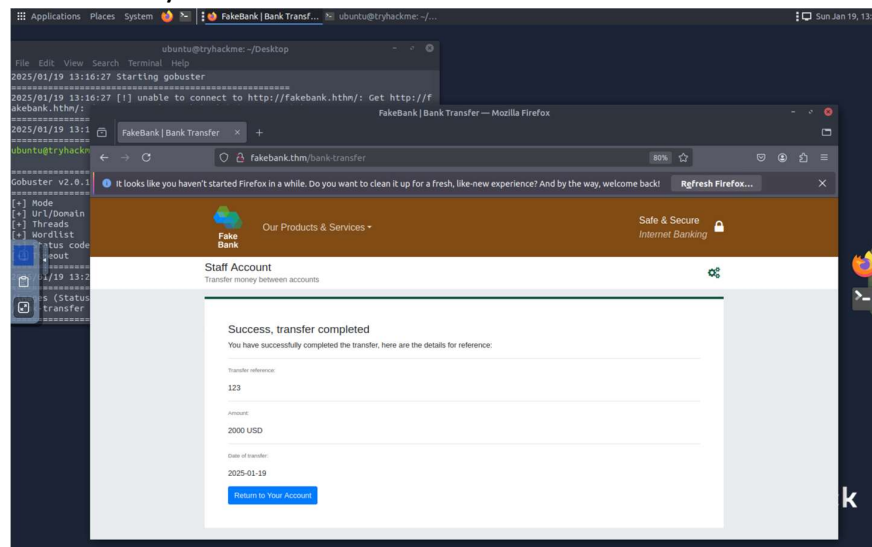


2. Now in the Firefox go to the fakebank.htm and add “/bank-transfer” in last.

The hacker has authorized access and can steal money from bank account.



here, the hacker has transferred the \$2000 in it's bank account successfully!



**Defensive Security:** Introducing defensive security and related topics, such as Threat Intelligence, SOC, DFIR, Malware Analysis, and SIEM.

Includes: Preventing intrusions from occurring | Detecting intrusions when they occur and responding properly.

*Blue teams* are part of Defensive security landscape

## **Security Operations Center (SOC):**

A *Security Operations Center* (SOC) is a team of cyber security professionals that monitors the network and its systems to detect malicious cyber security events. Some of the main areas of interest for a SOC are:

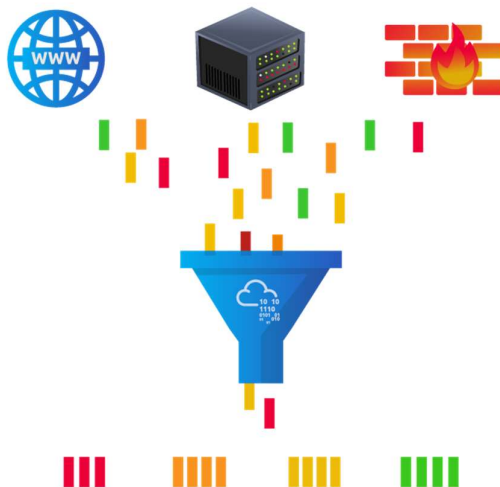
- **Vulnerabilities:** Whenever a system vulnerability (weakness) is discovered, it is essential to fix it by installing a proper update or patch. When a fix is unavailable, the necessary measures should be taken to prevent an attacker from exploiting it. Although remediating vulnerabilities is vital to a SOC, it is not necessarily assigned to them.
- **Policy violations:** A security policy is a set of rules required to protect the network and systems. For example, it might be a policy violation if users upload confidential company data to an online storage service.
- **Unauthorized activity:** Consider the case where a user's login name and password are stolen, and the attacker uses them to log into the network. A SOC must detect and block such an event as soon as possible before further damage is done.
- **Network intrusions:** No matter how good your security is, there is always a chance for an intrusion. An intrusion can occur when a user clicks on a malicious link or when an attacker exploits a public server. Either way, when an intrusion occurs, we must detect it as soon as possible to prevent further damage.



Security operations cover various tasks to ensure protection; one such task is “threat intelligence”.

### **Threat Intelligence:**

- In this context, *intelligence* refers to information you gather about actual and potential enemies. A *threat* is any action that can disrupt or adversely affect a system.



- Different companies have different adversaries. Some adversaries might seek to steal customer data from a mobile operator; however, other adversaries are interested in halting the production in a petroleum refinery.
- **Intelligence needs Data:** Can be sourced out from local sources such as forums and then the data will be processed and analysed.

## **Digital Forensics and Incident Response (DFIR)**

- **Digital Forensics:** To investigate crimes and establish facts, such as intellectual property theft, cyber espionage, and possession of unauthorized content.
  - File System: Analyzing a digital forensics image (low-level copy) of a system's storage reveals much information, such as installed programs, created files, partially overwritten files, and deleted files.
  - System memory: If the attacker runs their malicious program in memory without saving it to the disk, taking a forensic image (low-level copy) of the system memory is the best way to analyze its contents and learn about the attack.
  - System logs: Each client and server computer maintain different log files about what is happening. Log files provide plenty of information about what happened on a system. Even if the attacker tries to clear their traces, some traces will remain.
  - Network logs: Logs of the network packets that have traversed a network would help answer more questions about whether an attack is occurring and what it entails.

- **Incident Response:** An *incident* usually refers to a data breach or cyber-attack however, in some cases, it can be something less critical, such as a misconfiguration, an intrusion attempt, or a policy violation.

Examples of a cyber-attack include an attacker making our network or systems inaccessible, defacing (changing) the public website, and data breach (stealing company data).

The four major phases of the incident response process are:

1. **Preparation:** This requires a team trained and ready to handle incidents. Ideally, prevents incidents to occur.
2. **Detection and Analysis:** The team has the necessary resources to detect any incident; moreover, it is essential learn about its severity.
3. **Containment, Eradication, and Recovery:** Once an incident is detected, it is crucial to stop it from affecting other systems, eliminate it, and recover the affected systems. For instance, when we notice that a system is infected with a computer virus, we would like to stop (contain) the virus from spreading to other systems, clean (eradicate) the virus, and ensure proper system recovery.
4. **Post-Incident Activity:** After a successful recovery, a report is produced, and the lesson learned is shared to prevent similar future incidents.



➤ **Malware Analysis:** Malware stands for malicious software. It includes many types such as:

- A virus is a piece of code (part of a program) that attaches itself to a program. It is designed to spread from one computer to another and works by altering, overwriting, and deleting files once it infects a computer. The result ranges from the computer becoming slow to unusable.
- Trojan Horse is a program that shows one desirable function but hides a malicious function underneath. For example, a victim might download a video player from a shady website that gives the attacker complete control over their system.
- Ransomware is a malicious program that encrypts the user's files. Encryption makes the files unreadable without knowing the encryption password. The attacker offers the user the encryption password if the user is willing to pay a "ransom."
- There are two ways to learn about malicious attacks.
  - Static analysis works by inspecting the malicious program without running it. This usually requires solid knowledge of assembly language (the processor's instruction set, i.e., the computer's fundamental instructions).
  - Dynamic analysis works by running the malware in a controlled environment and monitoring its activities. It lets you observe how the malware behaves when running.

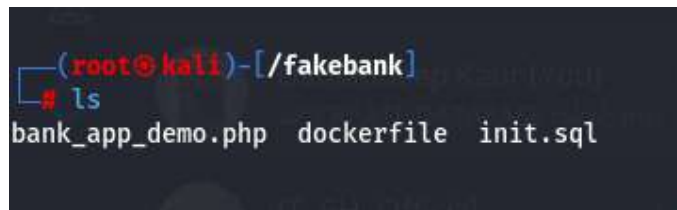
# Practical on the Defensive Security

- In virtual box: created Forensic VM contains kali los Debian 64 bit.
- In root terminal: `sudo apt-get upgrade` [takes longer time]
- Will Install Following things

- Docker:

- `Sudo apt-get install -y docker.io`
- `Sudo Systemctl start docker`
- `Sudo Systemctl enable docker`
- `Sudo docker --version`

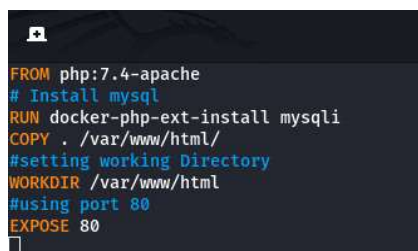
- Now, setting up FakeBank, Creating fakebank directory{folder}



```
(root@kali)-[/fakebank]
# ls
bank_app_demo.php  dockerfile  init.sql
```

- To locate Directory from terminal: `cd /file name/`  
open .

- Vi dockerfile



```
FROM php:7.4-apache
# Install mysql
RUN docker-php-ext-install mysqli
COPY . /var/www/html/
#setting working Directory
WORKDIR /var/www/html
#using port 80
EXPOSE 80
```

- Vi bank\_demo\_app.php



```
<?php
$conn = new mysqli("fakebank-mysql","root","mysql_password","fakebank");

if($conn->connect_error){
    die("Connection failed:" . $conn->connect_error);
}

$user_input = $_GET['id'];
$sql = "SELECT * FROM customers WHERE id = '$user_input'";
$result = $conn->query($sql);

if($result->num_rows > 0){
    while($row = $result->fetch_assoc()){
        echo "Name:" . $row['name'] . " -Account:" .
            $row['account_number'] . " -Balance:" .
            $row['balance'] . "<br>";
    }
}else{
    echo "0 results";
}
$conn->closed();
?>
```



- Vi init.sql

```
CREATE DATABASE fakebank;
USE fakebank;

CREATE TABLE customers{

    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    account_number VARCHAR(200),
    balance DECIMAL(10,2)

};

INSERT INTO customers(name, account_number,balance)VALUES
('John Doe','1234567890',1000.00),
('Peter Smith','0987654321',3400.00);
```

- Setting up MYSQL Container:

```
(root@kali)-[/]
└─# sudo docker network create fakebank-network
0339b81042d976650ab1eb575ef9a98228f3e46106396ab8d2c5d6957b61f139

(root@kali)-[/]
└─# sudo docker run -d --name fakebank-mysql --network fakebank-network -e MYSQL_ROOT_PASSWORD=fakebank -v $(pwd)/init.sql:/docker-entrypoint-initdb.d/init.sql mysql 5.7
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
2c0a233485c3: Downloading [=====] 34.13MB/49.1MB
cb5a6a8519b2: Download complete
570d30cf82c5: Download complete
a841bff36f3c: Download complete
80ba30c57782: Download complete
5e49e1f26961: Download complete
ced670fc7f1c: Downloading [=====] 28.04MB/48.21MB
0b9dc7ad7f03: Download complete
cd0d5df9937b: Downloading [=====] 29.66MB/69.09MB
1f87d67b89c6: Waiting
```

- In fakebank directory make docker containers:

- sudo docker build -t fakebank .
- sudo docker network create fakebank-network
- sudo docker run -d --name fakebank-mysql --network fakebank-network -e MYSQL\_ROOT\_PASSWORD=mypassword -v \$(pwd)/init.sql:/docker-entrypoint-initdb.d/init.sql:ro mysql:5.7
- sudo docker run -d --name fakebank-web --network fakebank-network -p 8080:80 fakebank
- Run the url on firefox:  
[http://localhost:8080/bank\\_app\\_demo.php?id=1](http://localhost:8080/bank_app_demo.php?id=1)

- Troubleshooting steps: problem arised at fakebank-web
  - o Sudo docker logs fakebank-web: to check is there any error.
  - o Edit the dockerfile as its suggesting to set the 'server name' directive globally to supress this message.  
# Set ServerName to localhost RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf  
under the fakebank directory-> vi dockerfile
  - o Now Redo the sudo docker build -t fakebank in the directory only.
  - o Remove fakebank-web name docker and again make it.  
sudo docker run -d --name fakebank-web --network fakebank-network -p 8080:80 fakebank
  - o Handling Deprecation Warning
    - Install Docker buildx sudo apt-get install -y docker-buildx
  - o sudo docker build -t fakebank .
  - o sudo docker stop fakebank-web
  - o sudo docker rm fakebank-web
  - o sudo docker run -d --name fakebank-web --network fakebank-network -p 8080:80 fakebank

#### 1: Initial Setup and Vulnerability Assessment

└─ Install Nessus - Identify vulnerabilities

#### 2: Forensic Analysis and Evidence Collection

└─ Install Seluth Key [TSK] - Collect digital evidence

#### 3: Containment

└─ Integrity Check with Tripwire - Verify data integrity

#### 4: Regulatory Compliance and Communication

└─ Privacy Regulations - Ensure GDPR and CCPA compliance

└─ Notification Templates - Inform affected stakeholders

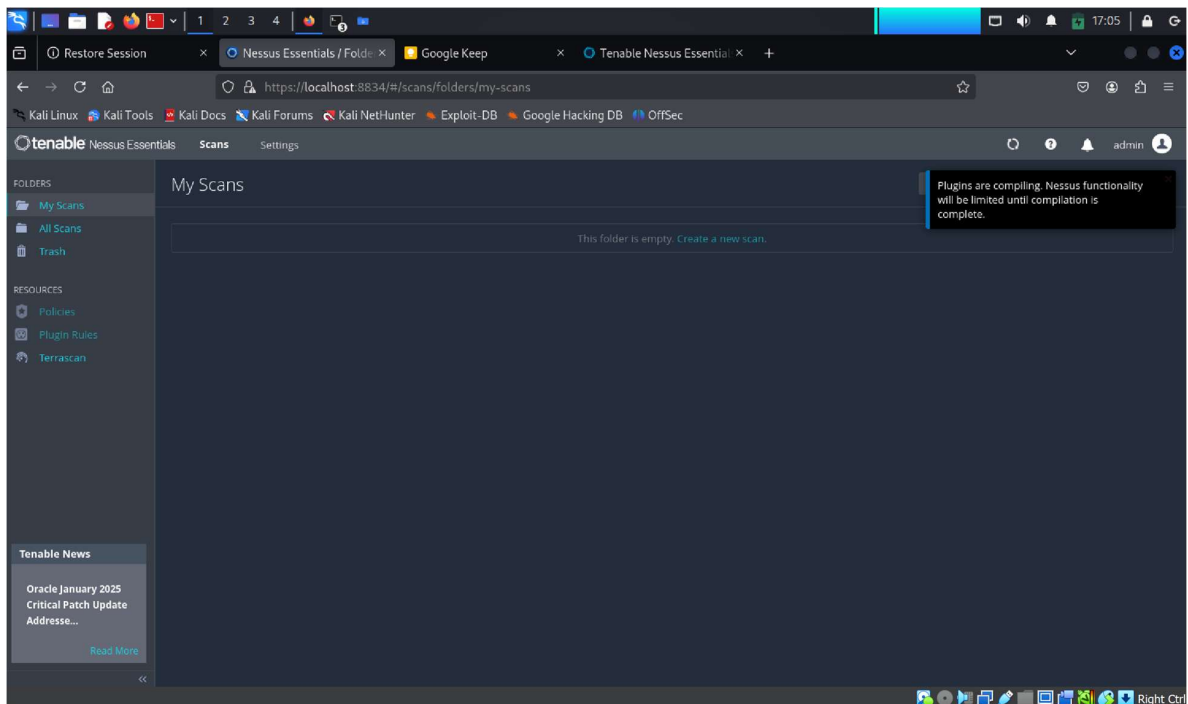
#### 5: Post-Incident Review and Recommendations

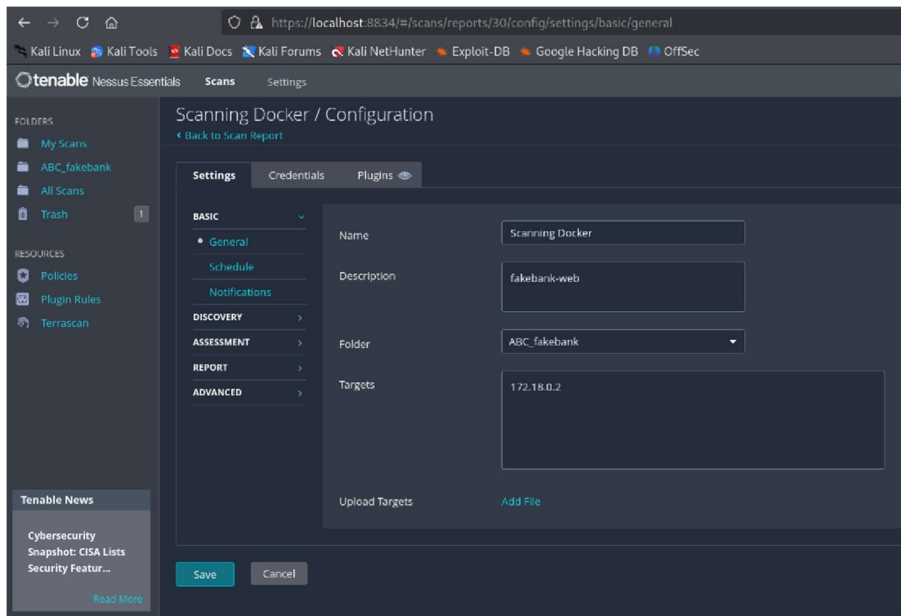
└─ Run Nessus Assessment - Confirm all vulnerabilities fixed

└─ Conduct RiskWatch Analysis- Comprehensive risk analysis

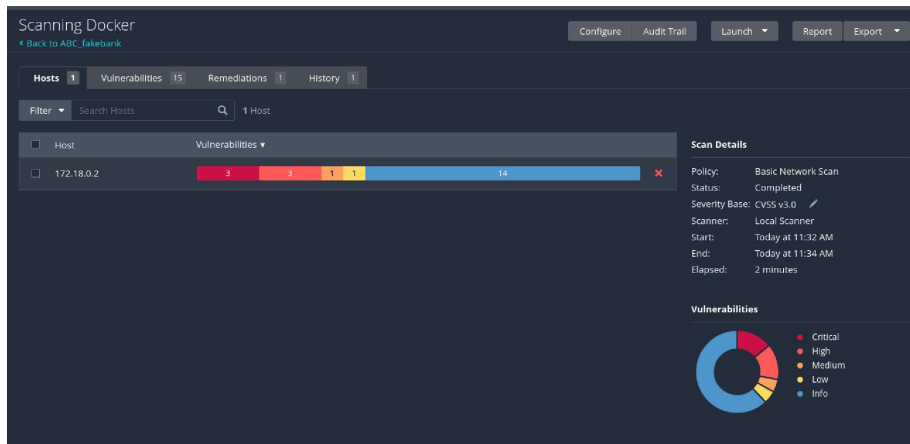
➤ Nessus: Identifying vulnerabilities

- <https://www.tenable.com/downloads/nessus?loginAttempted=true>
- `sudo dpkg -i Nessus.deb`
- `sudo systemctl enable nessusd`
- `sudo systemctl start nessusd`
- In firefox: <https://localhost:8834>
  - With it, open Nessus essential website  
<https://www.tenable.com/products/nessus/nessus-essentials?action=register>
    - Will ask for Business mail ID: For it open other tab and enter Temp mail <https://temp-mail.org/en/view/6794c9af2f960e001865bf26>
  - Now, In Nessus Essential site fill details, An Activation Code Will generate copy that from the temp mail site.
  - Now Relocate to <https://localhost:8834>, continue click on skip and enter Activation Code

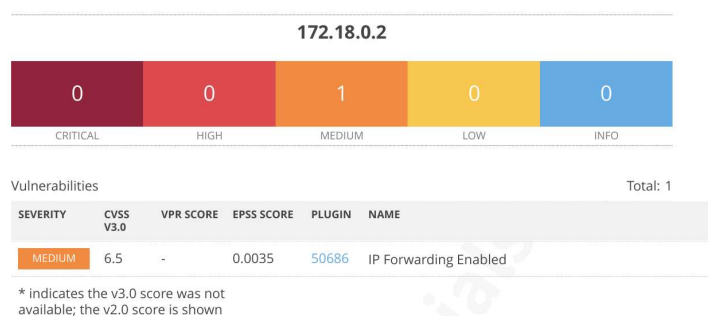




docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' fakebank-web  
An Ip Generated 172.18.0.2 -> Save and Launch



The Nessus scan result



Host: 172.18.0.2

Severity Level: Medium

**Vulnerability:** IP Forwarding Enabled CVSS v3.0

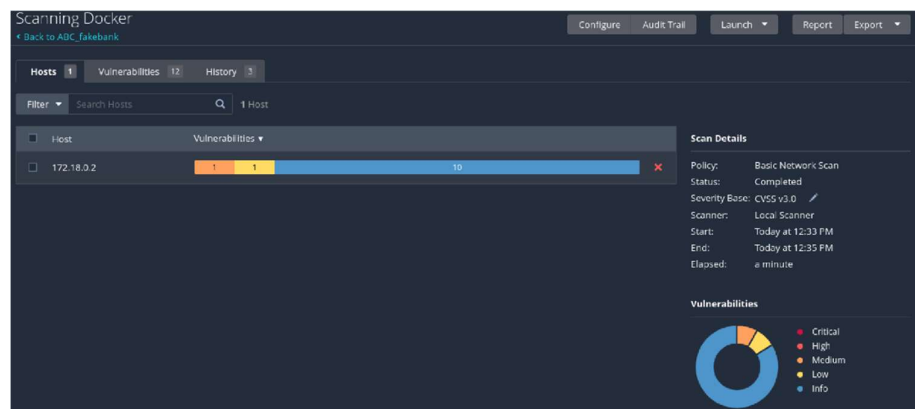
Score: 6.5 EPSS

Score: 0.0035

Plugin Name: 50686 - IP Forwarding Enabled

### Steps To Address Vulnerability:

- docker stop fakebank-web
- docker stop fakebank-mysql
- **Disable IP Forwarding:** docker exec -it fakebank-web /bin/bash
- echo 0 > /proc/sys/net/ipv4/ip\_forward
- Restart docker containers
- Launch the scan again



Vulnerabilities Resolved at some level!

## ➤ TSK (The Sleuth Key):Collects Digital Evidence

- Intalling TSK
  - `sudo apt-get update`
  - `sudo apt-get install sleuthkit`
- Export Container Filesystem:
  - `docker export fakebank-web -o fakebank_web_filesystem.tar`
  - `docker export fakebank-mysql -o fakebank_mysql_filesystem.tar`
- Extract .tar Files to Directories:
  - Create Extract Directories:
    - `mkdir -p /path/to/extracted_web || mkdir -p /home/user/extracted_web`
    - `mkdir -p /path/to/extracted_mysql`
  - Extract .tar Files to Created Directories:
    - `tar -xvf fakebank_web_filesystem.tar -C /path/to/extracted_web`
    - `tar -xvf fakebank_mysql_filesystem.tar -C /path/to/extracted_mysql`
- Convert Extracted Files to Raw Disk Images:
  - `cd /path/to/extracted_web`
  - `sudo dd if=/dev/zero of=fakebank_web_filesystem.img bs=1M count=1024`
  - `sudo mkfs.ext4 fakebank_web_filesystem.img`
  - `sudo mount -o loop fakebank_web_filesystem.img /mnt`
  - `sudo cp -R /path/to/extracted_web/* /mnt`
  - `sudo umount /mnt`
  - `cd /path/to/extracted_mysql`
  - `sudo dd if=/dev/zero of=fakebank_mysql_filesystem.img bs=1M count=1024`
  - `sudo mkfs.ext4 fakebank_mysql_filesystem.img`
  - `sudo mount -o loop fakebank_mysql_filesystem.img /mnt`
  - `sudo cp -R /path/to/extracted_mysql/* /mnt`
  - `sudo umount /mnt`

- List File System Content:

- `fls -r fakebank_web_filesystem.img | grep "access"`

```
(root@kali)-[/home/user/extracted_web]
# fls -r fakebank_web_filesystem.img | grep "access"
+++ r/r 135: other-vhosts-access-log.conf
+++ l/l 142: other-vhosts-access-log.conf
+++ r/r 148: access_compat.load
+++ l/l 292: access_compat.load
++ r/r 493: access.conf
+++ r/r 972: pam_access.so
+++++ r/r 1631: range_access.h
+++++++ r/r 1861: policy_access_fn_imps.hpp
+++++++ r/r 1879: policy_access_fn_imps.hpp
+++++++ r/r 1920: policy_access_fn_imps.hpp
+++++++ r/r 1951: policy_access_fn_imps.hpp
+++++++ r/r 1981: policy_access_fn_imps.hpp
+++++++ r/r 2006: policy_access_fn_imps.hpp
+++++++ r/r 2027: policy_access_fn_imps.hpp
+++++++ r/r 2031: synth_access_traits.hpp
+++++++ r/r 2097: sample_trie_access_traits.hpp
+++++++ r/r 2100: trie_string_access_traits_imp.hpp
++++ r/r 3814: mod_access_compat.so
+++++ r/r 11386: other-vhosts-access-log
+++++ r/r 11396: access_compat
+++ l/l 12290: access.log
+++ l/l 12292: other_vhosts_access.log
```

- `fls -r fakebank_mysql_filesystem.img | grep "error"`

```
(root@kali)-[/home/user/extracted_mysql]
# fls -r fakebank_mysql_filesystem.img | grep "error"
++ r/r 790: 89g_error
+++++ r/r 1901: pyerrors.h
+++++ r/r 2825: pycore_pyerrors.h
+++++ r/r 2889: pyerrors.h
+++++ r/r 2340: errors.cpython-313.pyc
+++++ r/r 2359: errors.py
+++++ d/d 3013: error
+++++ r/r 3076: _cffi_errors.h
+++++ r/r 3086: error.py
+++++++ r/r 3510: work_request_error.py
+++++++ r/r 3528: work_request_error_collection.py
+++++++ r/r 3593: work_request_error.py
+++++++ r/r 3594: work_request_error_collection.py
+++++++ r/r 3663: processing_error.py
+++++++ r/r 3685: work_request_error.py
+++++++ r/r 3686: work_request_error_collection.py
+++++++ r/r 3755: error_details.py
+++++++ r/r 3829: work_request_error.py
+++++++ r/r 3744: document_error.py
+++++++ r/r 3753: entity_label_error_analysis.py
+++++++ r/r 3877: realtime_message_error.py
+++++++ r/r 3973: processing_error.py
+++++++ r/r 3990: work_request_error.py
+++++++ r/r 3991: work_request_error_collection.py
+++++++ r/r 4027: work_request_error.py
+++++++ r/r 4233: work_request_error.py
+++++++ r/r 4234: work_request_error_collection.py
+++++++ r/r 4305: work_request_error.py
+++++++ r/r 4468: work_request_error.py
+++++++ r/r 4469: work_request_error_collection.py
+++++++ r/r 4614: work_request_error.py
+++++++ r/r 4742: work_request_error.py
+++++++ r/r 4608: default_error.py
+++++++ r/r 4785: work_request_error.py
+++++++ r/r 4786: work_request_error_collection.py
+++++++ r/r 5100: work_request_error.py
+++++++ r/r 5191: work_request_error_collection.py
+++++++ r/r 5371: work_request_error.py
+++++++ r/r 5372: work_request_error_collection.py
+++++++ r/r 5466: work_request_error.py
+++++++ r/r 5467: work_request_error_collection.py
+++++++ r/r 5489: work_request_error.py
+++++++ r/r 5490: work_request_error_collection.py
+++++++ r/r 5568: addon_error.py
```

- `istat fakebank_web_filesystem.img [inode]`
- `istat fakebank_mysql_filesystem.img [inode]`

```
(root@kali)-[/home/user/extracted_mysql]
# istat fakebank_mysql_filesystem.img 17107
inode: 17107
Allocated
Group: 2
Generation Id: 1188197487
uid / gid: 0 / 0
mode: rrw-r--r--
Flags: Extents,
size: 2312
num of links: 1

Inode Times:
Accessed: 2025-01-26 16:07:36.643583890 (IST)
File Modified: 2025-01-26 16:07:36.643583890 (IST)
Inode Modified: 2025-01-26 16:07:36.643583890 (IST)
File Created: 2025-01-26 16:07:36.643583890 (IST)

Direct Blocks:
119919
```

By this, We can Study About each activity that could be happened by reading files and can analyze the loophole due to which bank's data got hacked.

➤ Tripwire:

○ Key Steps:

- `sudo apt-get install tripwire`
- `sudo twadmin --generate-keys --site-keyfile /etc/tripwire/site.key --local-keyfile /etc/tripwire/kali-local.key`
  - It will ask you paraphrase: give strong password [combination of Alphanumeric]
- `sudo twadmin --create-cfgfile --cfgfile /etc/tripwire/tw.cfg --site-keyfile /etc/tripwire/site.key /etc/tripwire/twcfg.txt`
- `sudo twadmin --create-polfile --cfgfile /etc/tripwire/tw.cfg --site-keyfile /etc/tripwire/site.key /etc/tripwire/twpol.txt`
- `sudo tripwire --init` [takes time]
- `sudo tripwire --check` [takes time]

```
Open Source Tripwire(R) 2.4.3.7 Integrity Check Report

Report generated by:      root
Report created on:       Sunday 26 January 2025 06:02:46 PM
Database last updated on: Never

=====
Report Summary:
=====

Host name:                kali
Host IP address:          127.0.1.1
Host ID:                  None
Policy file used:         /etc/tripwire/tw.pol
Configuration file used:  /etc/tripwire/tw.cfg
Database file used:       /var/lib/tripwire/kali.twd
Command line used:        tripwire --check

=====
Rule Summary:
=====

Section: Unix File System

=====

```

Rule Name	Severity Level	Added	Removed	Modified
Other binaries	66	0	0	0
Tripwire Binaries	100	0	0	0
Other libraries	66	0	0	0
Root file-system executables	100	0	0	0
Tripwire Data Files	100	0	0	0
System boot changes	100	0	0	0
Root file-system libraries (/lib)	100	0	0	0
Critical system boot files	100	0	0	0
Other configuration files (/etc)	66	0	0	0
Boot Scripts	100	0	0	0
Security Control	66	0	0	0
Root config files	100	0	0	0
* Devices & Kernel information	100	1632	1632	0
Invariant Directories	66	0	0	0

```

Total objects scanned: 302177
Total violations found: 3264

=====
Object Summary:
=====

# Section: Unix File System

=====

Rule Name: Devices & Kernel information (/proc)
Severity Level: 100
=====
```