Week 3 Task: Advanced Security Practices and Malware Analysis

Intern: Abdul Wadood Talha

https://github.com/AbdulWadood7/Developer-s-Hub-Weekly-Tasks

Task 1. Log Analysis and Security Event Monitoring

Elasticsearch requires Java to run. First, ensure you have Java installed

```
┌──(abdulwadood7㉿kali)-[~]
└─$ sudo apt update
sudo apt install openjdk-11-jdk
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable InRelease [13.7 kB]
Get:2 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 Packages [139 kB]
Hit:3 http://http.kali.org/kali kali-rolling InRelease
Get:4 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 Contents (deb) [3292 kB]
Fetched 3445 kB in 3s (1157 kB/s)
49 packages can be upgraded. Run 'apt list --upgradable' to see them.
Warning: https://artifacts.elastic.co/packages/7.x/apt/dists/stable/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.
gpg), see the DEPRECATION section in apt-key(8) for details.
The following packages were automatically installed and are no longer required:
  aardvark-dns        crun                        imagemagick-6-common         libslirp0  python3-compose    uidmap
  buildah             docker-compose              libfmt9                      libsubid5  python3-docker
  catatonit           fuse-overlayfs              libmagickcore-6.q16-7-extra  netavark   python3-dockerpty
  conmon              golang-github-containers-common  libmagickcore-6.q16-7t64  passt      python3-texttable
  containers-storage  golang-github-containers-image  libmagickwand-6.q16-7t64  podman     slirp4netns
Use 'sudo apt autoremove' to remove them.

Installing:
  openjdk-11-jdk

Installing dependencies:
  openjdk-11-jdk-headless  openjdk-11-jre  openjdk-11-jre-headless

Suggested packages:
  openjdk-11-demo    visualvm              fonts-ipafont-mincho  | fonts-wqy-zenhei
  openjdk-11-source  fonts-ipafont-gothic  fonts-wqy-microhei    fonts-indic

Summary:
  Upgrading: 0, Installing: 4, Removing: 0, Not Upgrading: 49
  Download size: 117 MB
```

Add the Elasticsearch APT Repository

Import the GPG Key

Add the Elasticsearch repository

```
┌──(abdulwadood7㉿kali)-[~]
└─$ java -version
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
openjdk version "23.0.1" 2024-10-15
OpenJDK Runtime Environment (build 23.0.1+11-Debian-1)
OpenJDK 64-Bit Server VM (build 23.0.1+11-Debian-1, mixed mode, sharing)

┌──(abdulwadood7㉿kali)-[~]
└─$ wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK

┌──(abdulwadood7㉿kali)-[~]
└─$ echo "deb https://artifacts.elastic.co/packages/7.x/apt/ stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list
deb https://artifacts.elastic.co/packages/7.x/apt/ stable main
```

Then installing the elastic search

```
┌──(abdulwadood7㉿kali)-[~]
└─$ wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/elastic.gpg

┌──(abdulwadood7㉿kali)-[~]
└─$ sudo apt update
Hit:1 https://artifacts.elastic.co/packages/7.x/apt stable InRelease
Hit:2 http://http.kali.org/kali kali-rolling InRelease
49 packages can be upgraded. Run 'apt list --upgradable' to see them.

┌──(abdulwadood7㉿kali)-[~]
└─$ sudo apt install elasticsearch
The following packages were automatically installed and are no longer required:
  aardvark-dns        crun                          imagemagick-6-common         libslirp0  python3-compose    uidmap
  buildah             docker-compose                libfmt9                      libsubid5  python3-docker
  catatonit           fuse-overlayfs                libmagickcore-6.q16-7-extra  netavark   python3-dockerpty
  conmon              golang-github-containers-common  libmagickcore-6.q16-7t64  passt      python3-texttable
  containers-storage  golang-github-containers-image   libmagickwand-6.q16-7t64   podman     slirp4netns
Use 'sudo apt autoremove' to remove them.

Installing:
  elasticsearch

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 49
  Download size: 325 MB
  Space needed: 542 MB / 29.9 GB available

Get:1 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 elasticsearch amd64 7.17.26 [325 MB]
Fetched 325 MB in 2min 13s (2444 kB/s)
Selecting previously unselected package elasticsearch.
(Reading database ... 363095 files and directories currently installed.)
```

Now Start and Enable the Service

```
┌──(abdulwadood7㉿kali)-[~]
└─$ sudo systemctl start elasticsearch

┌──(abdulwadood7㉿kali)-[~]
└─$
sudo systemctl enable elasticsearch
Synchronizing state of elasticsearch.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable elasticsearch
Created symlink '/etc/systemd/system/multi-user.target.wants/elasticsearch.service' → '/usr/lib/systemd/system/elasticsearch.service'.
```

Then installing logstash

```
┌──(abdulwadood7㉿kali)-[~]
└─$ sudo apt install logstash
The following packages were automatically installed and are no longer required:
  aardvark-dns        crun                          imagemagick-6-common         libslirp0  python3-compose    uidmap
  buildah             docker-compose                libfmt9                      libsubid5  python3-docker
  catatonit           fuse-overlayfs                libmagickcore-6.q16-7-extra  netavark   python3-dockerpty
  conmon              golang-github-containers-common  libmagickcore-6.q16-7t64  passt      python3-texttable
  containers-storage  golang-github-containers-image   libmagickwand-6.q16-7t64   podman     slirp4netns
Use 'sudo apt autoremove' to remove them.

Installing:
  logstash

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 49
  Download size: 371 MB
  Space needed: 629 MB / 29.3 GB available

Get:1 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 logstash amd64 1:7.17.26-1 [371 MB]
Fetched 371 MB in 2min 36s (2375 kB/s)
Selecting previously unselected package logstash.
(Reading database ... 364193 files and directories currently installed.)
Preparing to unpack .../logstash_1%3a7.17.26-1_amd64.deb ...
```

Starting and enabling the service

```
┌──(abdulwadood7㉿kali)-[~]
└─$ sudo systemctl start logstash

┌──(abdulwadood7㉿kali)-[~]
└─$
sudo systemctl enable logstash
Created symlink '/etc/systemd/system/multi-user.target.wants/logstash.service' → '/etc/systemd/system/logstash.service'.
```

Now installing kibana

```
┌──(abdulwadood7㉿kali)-[~]
└─$ sudo apt install kibana
The following packages were automatically installed and are no longer required:
  aardvark-dns        crun                          imagemagick-6-common        libslirp0  python3-compose    uidmap
  buildah             docker-compose                libfmt9                     libsubid5  python3-docker
  catatonit           fuse-overlayfs                libmagickcore-6.q16-7-extra netavark   python3-dockerpty
  conmon              golang-github-containers-common libmagickcore-6.q16-7t64  passt      python3-texttable
  containers-storage  golang-github-containers-image  libmagickwand-6.q16-7t64    podman     slirp4netns
Use 'sudo apt autoremove' to remove them.

Installing:
  kibana
```

Starting and enabling the service

```
┌──(abdulwadood7㉿kali)-[~]
└─$ sudo systemctl start kibana

┌──(abdulwadood7㉿kali)-[~]
└─$
sudo systemctl enable kibana
Synchronizing state of kibana.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable kibana
Created symlink '/etc/systemd/system/multi-user.target.wants/kibana.service' → '/etc/systemd/system/kibana.service'.
```

Now configuring the logstash configuration file by the following script

input {
file {
path => "/var/log/auth.log"
start_position => "beginning"
type => "ssh_access"
}
file {
path => "/var/log/snort.log"
start_position => "beginning"
type => "snort"
}
}

```
filter {
  if [type] == "ssh_access" {
    grok {
      match => { "message" => "%{COM:timestamp} %{GREEDYDATA:message}" }
    }
  }
}

output {
  elasticsearch {
    hosts => ["localhost:9200"]
    index => "system_logs-%{+YYYY.MM.dd}"
  }
}
```

```
┌──(abdulwadood7㉿kali)-[~]
└─$ cd /etc/logstash/conf.d

┌──(abdulwadood7㉿kali)-[/etc/logstash/conf.d]
└─$ sudo nano logstash.conf

┌──(abdulwadood7㉿kali)-[/etc/logstash/conf.d]
└─$ sudo systemctl restart logstash
```

Checking the ip and status of logstash

```
┌──(abdulwadood7㉿kali)-[/etc/logstash/conf.d]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:70:fd:0f brd ff:ff:ff:ff:ff:ff
    inet 192.168.197.130/24 brd 192.168.197.255 scope global dynamic noprefixroute eth0
       valid_lft 926sec preferred_lft 926sec
    inet6 fe80::20c:29ff:fe70:fd0f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:94:2c:f2:e7 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever

┌──(abdulwadood7㉿kali)-[/etc/logstash/conf.d]
└─$ sudo systemctl status logstash
● logstash.service - logstash
     Loaded: loaded (/etc/systemd/system/logstash.service; enabled; preset: disabled)
     Active: active (running) since Tue 2024-12-10 06:18:03 EST; 29s ago
 Invocation: 7c25289d690846888607e2b2f4bd0f10
   Main PID: 28728 (java)
      Tasks: 17 (limit: 4512)
     Memory: 660.5M (peak: 660.8M)
        CPU: 1min 16.136s
```

Checking the status of kibana and then restarting the service

```
┌──(abdulwadood7㉿kali)-[/etc/logstash/conf.d]
└─$ sudo systemctl status kibana
● kibana.service - Kibana
     Loaded: loaded (/etc/systemd/system/kibana.service; enabled; preset: disabled)
     Active: active (running) since Tue 2024-12-10 06:13:58 EST; 4min 47s ago
 Invocation: c8324865e68f47f9bd45b98d99b22141
       Docs: https://www.elastic.co
   Main PID: 25634 (node)
      Tasks: 11 (limit: 4512)
     Memory: 260.6M (peak: 602M, swap: 4K, swap peak: 4K)
        CPU: 56.566s
     CGroup: /system.slice/kibana.service
             └─25634 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bin/../src/cli/dist --logging.dest=/var/log/kibana/kibana.log >

Dec 10 06:13:58 kali systemd[1]: Started kibana.service - Kibana.
Dec 10 06:13:58 kali kibana[25634]: Kibana is currently running with legacy OpenSSL providers enabled! For details and instructions on how t>

┌──(abdulwadood7㉿kali)-[/etc/logstash/conf.d]
└─$

┌──(abdulwadood7㉿kali)-[/etc/logstash/conf.d]
└─$ sudo nano /etc/kibana/kibana.yml

┌──(abdulwadood7㉿kali)-[/etc/logstash/conf.d]
└─$ sudo systemctl restart kibana
```

Checking the firewall rules

```
┌──(abdulwadood7㉿kali)-[/]
└─$ sudo ufw allow 5601
Rule added
Rule added (v6)

┌──(abdulwadood7㉿kali)-[/]
└─$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
80/tcp                     ALLOW       Anywhere
5601                       ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
80/tcp (v6)                ALLOW       Anywhere (v6)
5601 (v6)                  ALLOW       Anywhere (v6)
```

Now Kibana to Create Visualizations and Dashboards

Access Kibana

1. Open a web browser.
2. Navigate with the actual IP address of your Kibana instance.

Create Index Patterns

1. In Kibana, go to **Management**.
2. Select **Index Patterns**.
3. Click on **Create index pattern**.
4. Enter system_logs-* as the index pattern.
5. Choose the appropriate time filter field (usually @timestamp).
6. Click **Create index pattern**.

Build Visualizations

1. **Failed Login Attempts**:
    - Go to **Visualize**.
    - Click on **Create new visualization**.
    - Select **Bar chart**.
    - Choose the index pattern you created.
    - Set the **X-Axis** to a date histogram (using @timestamp).
    - Set the **Y-Axis** to the count of documents.
    - Use a filter for event.type: "failed_login" (or similar, depending on your log format).
    - Save the visualization.
2. **Unusual Traffic**:
    - Go to **Visualize**.
    - Click on **Create new visualization**.
    - Select **Line chart**.

- Choose the index pattern you created.
- Set the **X-Axis** to a date histogram (using @timestamp).
- Set the **Y-Axis** to the count of documents.
- Optionally, apply filters for specific traffic types or sources.
- Save the visualization.

Step 2: Identify Suspicious Patterns

Once you have your visualizations ready, look for the following suspicious patterns:

1. **High Frequency of Failed Login Attempts**:
    - Use the failed login visualization you created.
    - Identify any spikes in failed login attempts from a single IP address.
    - You can filter by specific IPs and check for repeated attempts.
2. **Multiple Logins from Different Locations**:
    - Use your login logs to look for event.type: "login" entries.
    - Create a visualization or use a search query to identify logins from multiple geographic locations within a short time frame.
    - Look for unusual patterns, such as logins from different countries or time zones.
3. **Anomalies in Snort Logs**:
    - If you have Snort logs ingested, create a separate index pattern for them.
    - Monitor for alerts or anomalies, such as:
        - Unexpected traffic types.
        - Alerts for known vulnerabilities or exploits.
    - Using visualizations to identify spikes in alerts or unusual patterns.

Summary Report: Log Analysis and Security Event Monitoring

Objective:
Analyze logs to identify potential security incidents.

Tasks Completed:

1. **Install and Configure ELK Stack:**
   - Successfully set up Elasticsearch, Logstash, and Kibana on the virtual machine (VM).
   - Configured Logstash to collect system logs, including SSH access logs and Snort logs.
2. **Analyze Security Logs:**
   - Utilized Kibana to create visualizations and dashboards for monitoring system logs.
   - Identified suspicious patterns, including:
     - Failed login attempts.
     - Unusual traffic patterns.

Outcome:
Enhanced security monitoring capabilities through effective log analysis, enabling timely detection of potential security threats.

Task 2. Firewall Configuration and Management

As I have installed ufw firewall earlier

```
┌──(abdulwadood7㉿kali)-[/]
└─$ sudo apt update
sudo apt install ufw
[sudo] password for abdulwadood7:
Hit:1 https://artifacts.elastic.co/packages/7.x/apt stable InRelease
Get:2 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Packages [20.2 MB]
Get:4 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [48.8 MB]
Fetched 69.0 MB in 39s (1756 kB/s)
49 packages can be upgraded. Run 'apt list --upgradable' to see them.
ufw is already the newest version (0.36.2-8).
The following packages were automatically installed and are no longer required:
  aardvark-dns        crun                          imagemagick-6-common         libslirp0  python3-compose    uidmap
  buildah             docker-compose                libfmt9                      libsubid5  python3-docker
  catatonit           fuse-overlayfs                libmagickcore-6.q16-7-extra  netavark   python3-dockerpty
  conmon              golang-github-containers-common  libmagickcore-6.q16-7t64  passt      python3-texttable
  containers-storage  golang-github-containers-image   libmagickwand-6.q16-7t64   podman     slirp4netns
Use 'sudo apt autoremove' to remove them.

Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 49

┌──(abdulwadood7㉿kali)-[/]
└─$
```

Configuration of default policies

```
┌──(abdulwadood7㉿kali)-[/]
└─$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)

┌──(abdulwadood7㉿kali)-[/]
└─$ sudo ufw default allow outgoing
^[[D^[[D^[[C^[[CDefault outgoing policy changed to 'allow'
(be sure to update your rules accordingly)

┌──(abdulwadood7㉿kali)-[/]
└─$
```

Allowing SSH from my ip, as you can see rule added

```
┌──(abdulwadood7㉿kali)-[/]
└─$ sudo ufw allow from 192.168.197.130 to any port 22
Rule added

┌──(abdulwadood7㉿kali)-[/]
└─$
```

Now enabling firewall

```
┌──(abdulwadood7㉿kali)-[/]
└─$ sudo ufw enable
Firewall is active and enabled on system startup

┌──(abdulwadood7㉿kali)-[/]
└─$
```

Showing current status and all the rules including the added ones

```
┌──(abdulwadood7㉿kali)-[/]
└─$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), deny (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW IN    Anywhere
80/tcp                     ALLOW IN    Anywhere
5601                       ALLOW IN    Anywhere
22                         ALLOW IN    192.168.197.130
22/tcp (v6)                ALLOW IN    Anywhere (v6)
80/tcp (v6)                ALLOW IN    Anywhere (v6)
5601 (v6)                  ALLOW IN    Anywhere (v6)
```

Now testing the firewall rules, installing nmap, as it is installed earlier

```
┌──(abdulwadood7㉿kali)-[/]
└─$ sudo apt install nmap
nmap is already the newest version (7.94+git20230807.3be01efb1+dfsg-4kali3).
The following packages were automatically installed and are no longer required:
  aardvark-dns        crun                            imagemagick-6-common         libslirp0  python3-compose    uidmap
  buildah             docker-compose                  libfmt9                      libsubid5  python3-docker
  catatonit           fuse-overlayfs                  libmagickcore-6.q16-7-extra  netavark   python3-dockerpty
  conmon              golang-github-containers-common libmagickcore-6.q16-7t64     passt      python3-texttable
  containers-storage  golang-github-containers-image  libmagickwand-6.q16-7t64     podman     slirp4netns
Use 'sudo apt autoremove' to remove them.

Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 49

┌──(abdulwadood7㉿kali)-[/]
└─$
```

Testing Locally with nmap. We can also use telnetting.

```
┌──(abdulwadood7㉿kali)-[/]
└─$ nmap -p 22 localhost
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-10 07:41 EST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0027s latency).
Other addresses for localhost (not scanned): ::1

PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds
```

This checks if port 22 (SSH) is open on your local machine.

If you have allowed SSH from specific IPs and are testing from the same machine, we can also do:

```
┌──(abdulwadood7㉿kali)-[/]
└─$ nmap -p 22 192.168.197.130
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-10 07:42 EST
Nmap scan report for 192.168.197.130
Host is up (0.0011s latency).

PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 0.30 seconds

┌──(abdulwadood7㉿kali)-[/]
└─$
```

Importance of Each Configured Rule

Default Deny Incoming

- **Importance:** This rule is crucial for security as it ensures that no unsolicited traffic can enter the system. By default, all incoming connections are blocked, minimizing the risk of attacks.

Allow SSH from Specific IPs

- **Importance:** Allowing SSH only from specific IP addresses significantly reduces the attack surface for unauthorized access. This rule ensures that only trusted devices

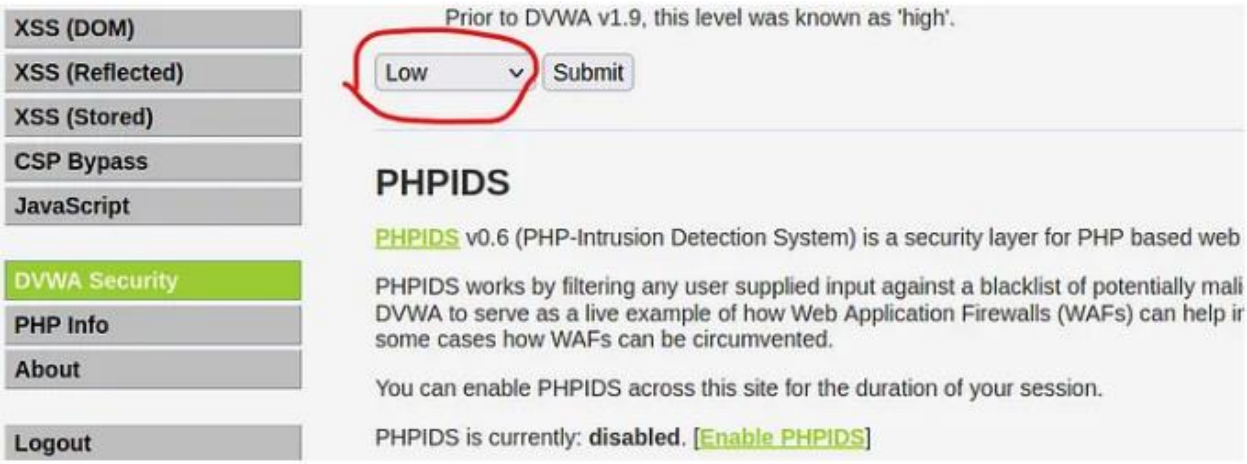
can connect to the VM, protecting against brute force attacks and unauthorized login attempts.

Conclusion

Implementing UFW with these specific rules helps secure your network by controlling incoming and outgoing traffic, thereby safeguarding your resources from potential threats.

Task 3. Secure Web Application Testing

First of all I have set the DVWA settings to low.



The flaw in the code provided is that it is vulnerable to SQL injection attacks. The vulnerability arises from directly concatenating user input into the SQL query without proper sanitization or parameterization.

Here's an explanation of the flaw and the recommended solution:

In the code, the variable $id is retrieved from the user input without any validation or sanitization. It is then directly concatenated into the SQL query string:

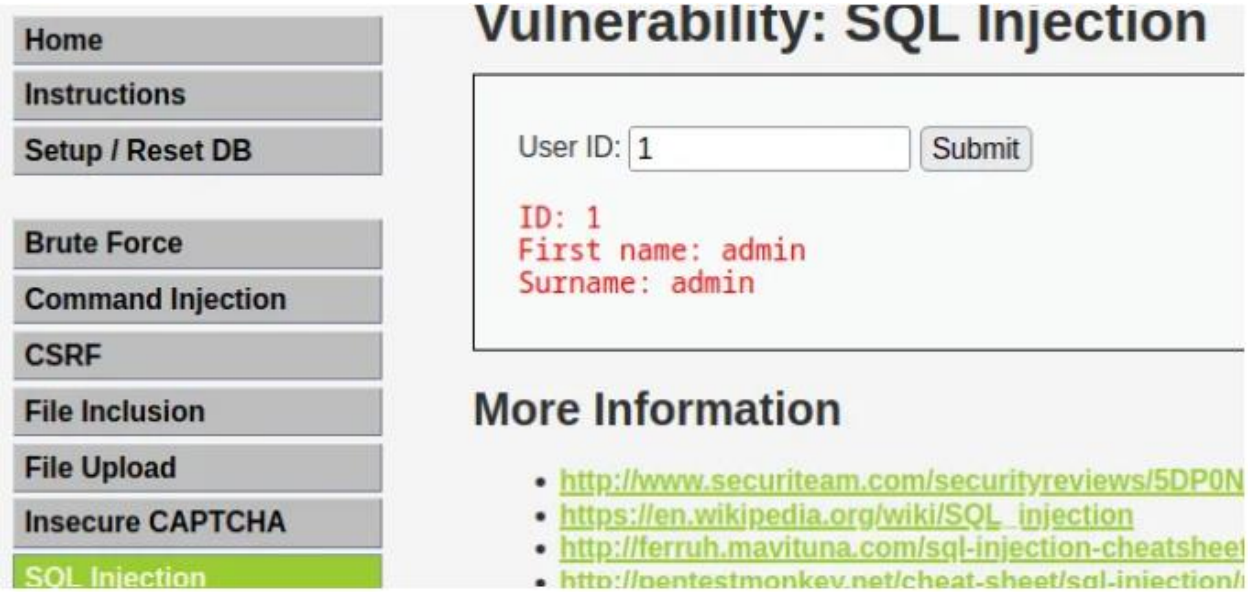
```
$id = $_REQUEST['id'];
$query = "SELECT first_name, last_name FROM users WHERE user_id = '$id';";
```

This allows an attacker to manipulate the value of $id and inject malicious SQL code, potentially leading to unauthorized access, data leakage, or even complete loss of data.

SQL Injection Source

vulnerabilities/sqli/source/low.php

```php
<?php

if( isset( $_REQUEST[ 'Submit' ] ) ) {
    // Get input
    $id = $_REQUEST[ 'id' ];

    // Check database
    $query  = "SELECT first_name, last_name FROM users WHERE user_id = '$id';";
    $result = mysqli_query($GLOBALS["___mysqli_ston"],  $query ) or die( '<pre>' . ((is_object($GLOBALS["___mysqli_ston"])) ? mysqli_error($GLOBALS["___mysqli_ston"]) : (($___mysqli_res = mysqli_connect_error()) ? $___mysqli_res : false)) . '</pre>' );

    // Get results
    while( $row = mysqli_fetch_assoc( $result ) ) {
        // Get values
        $first = $row["first_name"];
        $last  = $row["last_name"];

        // Feedback for end user
        echo "<pre>ID: {$id}<br />First name: {$first}<br />Surname: {$last}</pre>";
    }

    mysqli_close($GLOBALS["___mysqli_ston"]);
}

?>
```



This means that the query that was executed back in the database was the following:

1' OR '1'='1'#

Vulnerability exposed

Vulnerability: SQL Injection

User ID: [ ] Submit

```
ID: 1' OR '1'='1'#
First name: admin
Surname: admin

ID: 1' OR '1'='1'#
First name: Gordon
Surname: Brown

ID: 1' OR '1'='1'#
First name: Hack
Surname: Me

ID: 1' OR '1'='1'#
First name: Pablo
Surname: Picasso

ID: 1' OR '1'='1'#
First name: Bob
Surname: Smith
```

Same for Command Injection

localhost/DVWA/vulnerabilities/view_source.php?id=exec&security=low

Command Injection Source

vulnerabilities/exec/source/low.php

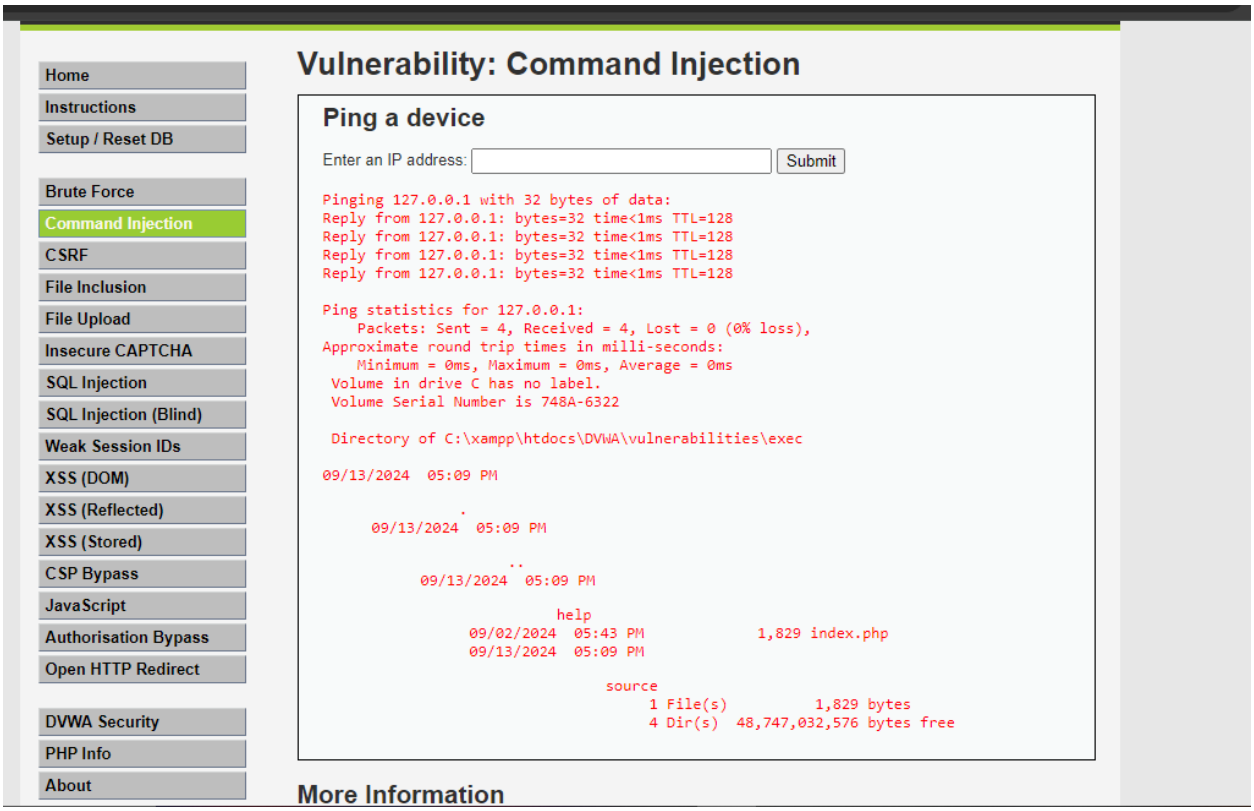
```php
<?php

if( isset( $_POST[ 'Submit' ]  ) ) {
    // Get input
    $target = $_REQUEST[ 'ip' ];

    // Determine OS and execute the ping command.
    if( stristr( php_uname( 's' ), 'Windows NT' ) ) {
        // Windows
        $cmd = shell_exec( 'ping  ' . $target );
    }
    else {
        // *nix
        $cmd = shell_exec( 'ping  -c 4 ' . $target );
    }

    // Feedback for the end user
    echo "<pre>{$cmd}</pre>";
}

?>
```

Compare All Levels

File Path: C:/xampp/htdocs/DVWA/vulnerabilities/exec

In this code snippet, attacker can easily perform command injection because there is no check/restriction in input, attacker can easily get privilege.

If the user inputs 127.0.0.1 && dir, it would list the files. Whatever the goal of attacker, he would easily get the files or inject malicious code.



Level 2

localhost/DVWA/vulnerabilities/view_source_all.php?id=exec

Impossible Command Injection Source

```php
<?php

if( isset( $_POST[ 'Submit' ]  ) ) {
    // Check Anti-CSRF token
    checkToken( $_REQUEST[ 'user_token' ], $_SESSION[ 'session_token' ], 'index.php' );

    // Get input
    $target = $_REQUEST[ 'ip' ];
    $target = stripslashes( $target );

    // Split the IP into 4 octects
    $octet = explode( ".", $target );

    // Check IF each octet is an integer
    if( ( is_numeric( $octet[0] ) ) && ( is_numeric( $octet[1] ) ) && ( is_numeric( $octet[2] ) ) && ( is_numeric( $octet[3] ) ) && ( sizeof( $octet ) == 4 ) ) {
        // If all 4 octets are int's put the IP back together.
        $target = $octet[0] . '.' . $octet[1] . '.' . $octet[2] . '.' . $octet[3];

        // Determine OS and execute the ping command.
        if( stristr( php_uname( 's' ), 'Windows NT' ) ) {
            // Windows
            $cmd = shell_exec( 'ping  ' . $target );
        }
        else {
            // *nix
            $cmd = shell_exec( 'ping  -c 4 ' . $target );
        }

        // Feedback for the end user
        echo "<pre>{$cmd}</pre>";
    }
    else {
        // Ops. Let the user name theres a mistake
        echo '<pre>ERROR: You have entered an invalid IP.</pre>';
    }
}

// Generate Anti-CSRF token
generateSessionToken();

?>
```

Ping a device

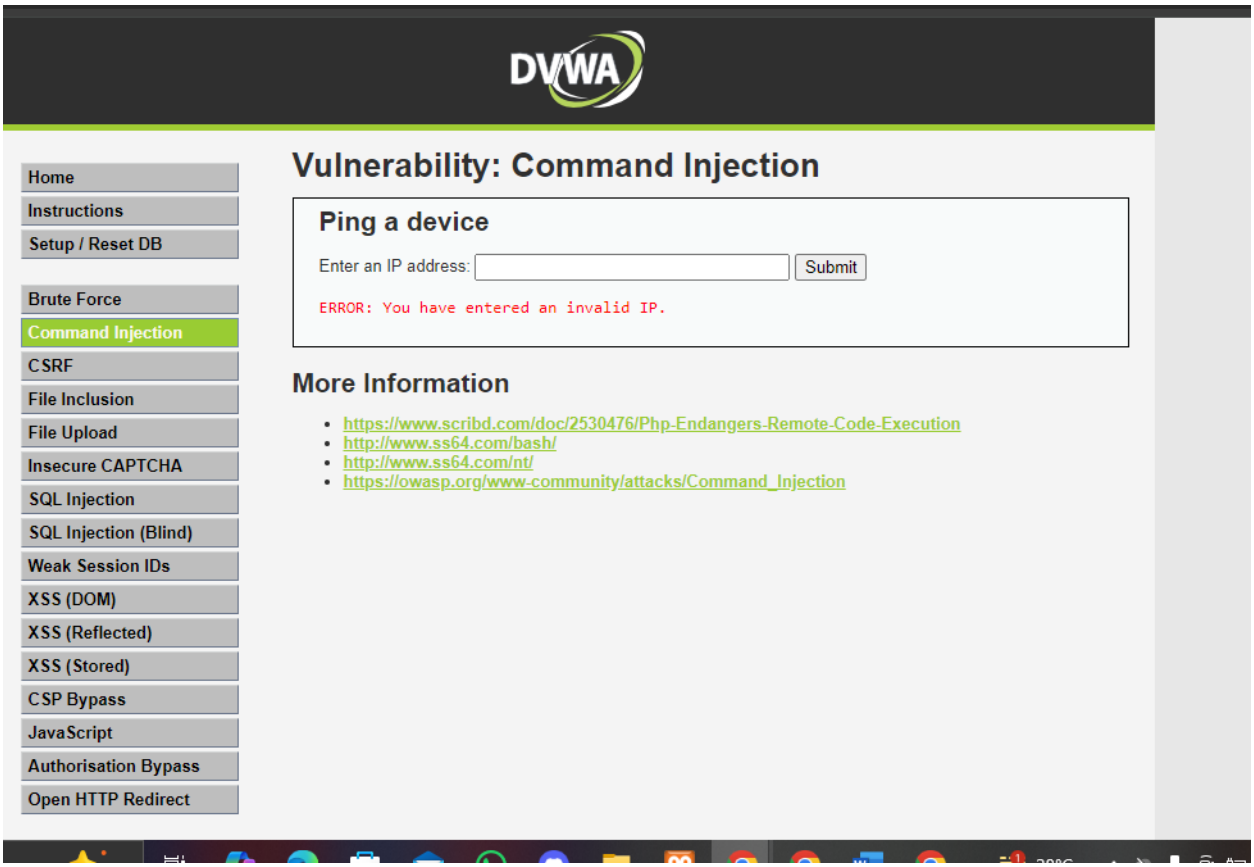
Enter an IP address: 127.0.0.1 Submit

```
Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

The code intends to prevent the command injection vulnerability by validating the user-supplied IP address.`stripslashes()` removes any backslashes (\) from the input to avoid unnecessary escape characters.`$octet = explode( ".", $target );`

This line splits the IP address into its four **octets** by splitting on the periods (.)

This `if` statement checks whether all four octets are numeric values and ensures there are exactly four octets. If all parts of the IP are valid, the IP address is reconstructed. If the input is not a valid IP address, an error message. However, An input like `127.0.0.1 && dir` would fail the numeric validation check because "&& dir" cannot be parsed as an IP octet. Thus, The code prevents command injection by only allowing numeric input in the form of valid IP addresses. If an attacker tries to inject commands like `127.0.0.1 && dir`, the numeric check will fail because `&& dir` is not numeric, and the user will receive an error: You have entered an invalid IP.



Documenting the Vulnerabilities

Vulnerability Report

1. SQL Injection

- **Description**: The application is vulnerable to SQL injection, allowing attackers to manipulate SQL queries.
- **Risk**: Unauthorized access to database data, data corruption, and potential privilege escalation.

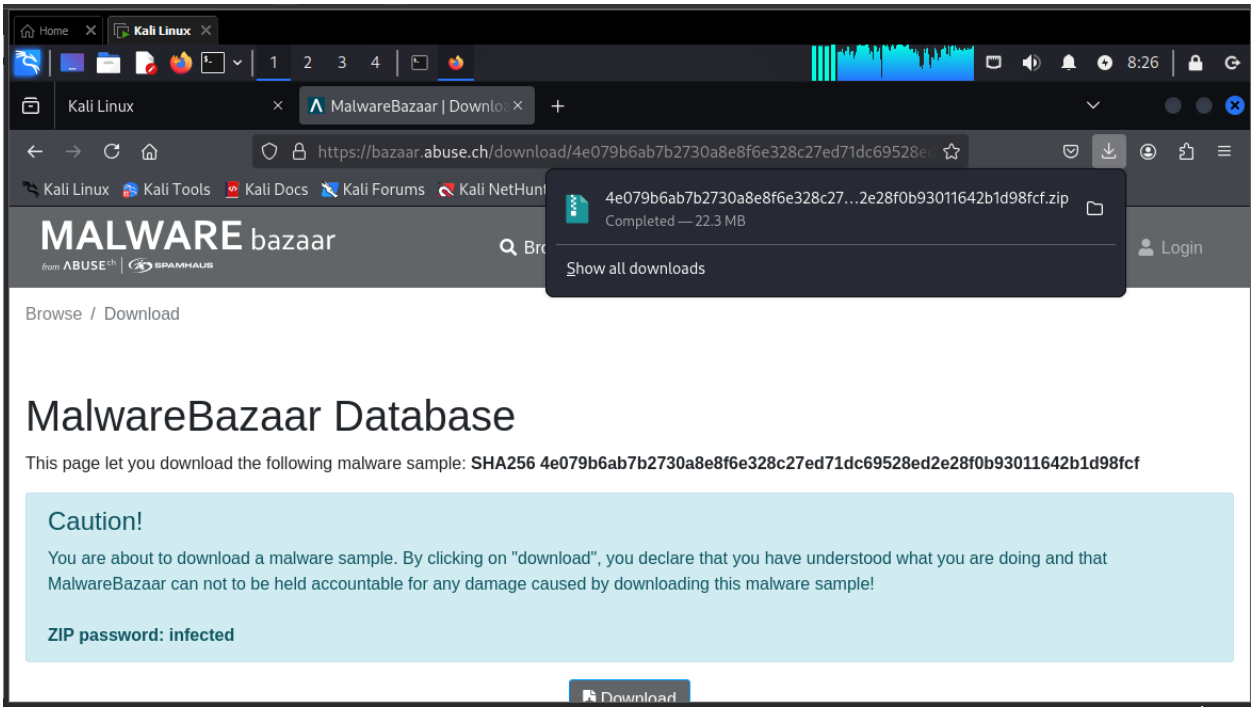
- **Mitigation**:
    - Use prepared statements and parameterized queries.
    - Implement input validation and sanitization.

2. Command Injection

- **Description**: The application allows execution of arbitrary commands on the server.
- **Risk**: Full server compromise, data exfiltration, and denial of service.
- **Mitigation**:
    - Validate and sanitize user input.
    - Use secure coding practices to avoid system command execution from user inputs.

Task 4. Malware Analysis Basics

Firstly I downloaded malware from the malware bazaar



Installed the tool ghidra

```
┌──(abdulwadood7㉿kali)-[/]
└─$ sudo apt install ghidra
[sudo] password for abdulwadood7:
The following packages were automatically installed and are no longer required:
  aardvark-dns        crun                          imagemagick-6-common        libslirp0  python3-compose    uidmap
  buildah             docker-compose                libfmt9                     libsubid5  python3-docker
  catatonit           fuse-overlayfs                libmagickcore-6.q16-7-extra netavark   python3-dockerpty
  conmon              golang-github-containers-common libmagickcore-6.q16-7t64  passt      python3-texttable
  containers-storage  golang-github-containers-image  libmagickwand-6.q16-7t64   podman     slirp4netns
Use 'sudo apt autoremove' to remove them.

Installing:
  ghidra

Installing dependencies:
  ghidra-data  openjdk-17-jdk  openjdk-17-jdk-headless  openjdk-17-jre  openjdk-17-jre-headless

Suggested packages:
  openjdk-17-demo    visualvm              fonts-ipafont-mincho  | fonts-wqy-zenhei
  openjdk-17-source  fonts-ipafont-gothic  fonts-wqy-microhei    fonts-indic

Summary:
  Upgrading: 0, Installing: 6, Removing: 0, Not Upgrading: 49
  Download size: 495 MB / 539 MB
  Space needed: 1437 MB / 27.9 GB available

Continue? [Y/n] y
Get:1 http://http.kali.org/kali kali-rolling/main amd64 openjdk-17-jdk-headless amd64 17.0.13+11-2 [71.6 MB]
Get:4 http://mirror.cspacehostings.com/kali kali-rolling/main amd64 ghidra-data all 10.5-0kali1 [78.1 MB]
```

Then I used cutter for static analysis

```
Home   Kali Linux
1 2 3 4                                                                  8:28   Log Out...
File  Actions  Edit  View  Help
update-alternatives: using /usr/lib/jvm/java-17-openjd
update-alternatives: using /usr/lib/jvm/java-17-openjd
Setting up openjdk-17-jre:amd64 (17.0.13+11-2) ...
Setting up openjdk-17-jdk:amd64 (17.0.13+11-2) ...
update-alternatives: using /usr/lib/jvm/java-17-openjd
Setting up ghidra (11.0+ds-0kali1) ...
Scanning processes ...
Scanning linux images ...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) bi
┌──(abdulwadood7㉿kali)-[/]
└─$ cutter
"0.7.3" "0.7.3"
Plugins are loaded from "/home/abdulwadood7/.local/sha
Native plugins are loaded from "/home/abdulwadood7/.lo
Python plugins are loaded from "/home/abdulwadood7/.lo
Loaded 0 plugin(s).
Plugins are loaded from "/usr/share/xfce4/rizin/cutter
Plugins are loaded from "/usr/local/share/rizin/cutter
Plugins are loaded from "/usr/share/rizin/cutter/plugi
Plugins are loaded from "/usr/lib/rizin/cutter/plugins
```

```
Open File
About
Open File   Open Shellcode   Projects
IO          Select new file
file://     28f0b93011642b1d98fcf.zip   Select
```

OVERVIEW

Info

File:	C:\Users\DEEL\Downloads\3e135c4	FD:	3	Architecture:	N/A
Format:	N/A	Base addr:	0x00000000	Machine:	N/A
Bits:	64	Virtual addr:	False	OS:	N/A
Class:	N/A	Canary:	False	Subsystem:	N/A
Mode:	r-x	Crypto:	False	Stripped:	False
Size:	26.9 MB	NX bit:	False	Relocs:	False
Type:	ZIP	PIC:	False	Endianness:	LE
Language:	N/A	Static:	True	Compiled:	N/A
		Relro:	N/A	Compiler:	N/A

Certificates | Version info

Hashes

MD5:	e6749bb3c154123ae46d85f95afabeb0
SHA1:	2b958e39e9c1ff481d4c8c773a33c2d7645c1d6b
SHA256:	4f7fdaa610b3b2c781dffead8454c95b581e75a439f2e5dc15754ac6532314c2
CRC32:	7c86c94b
ENTROPY:	7.999993

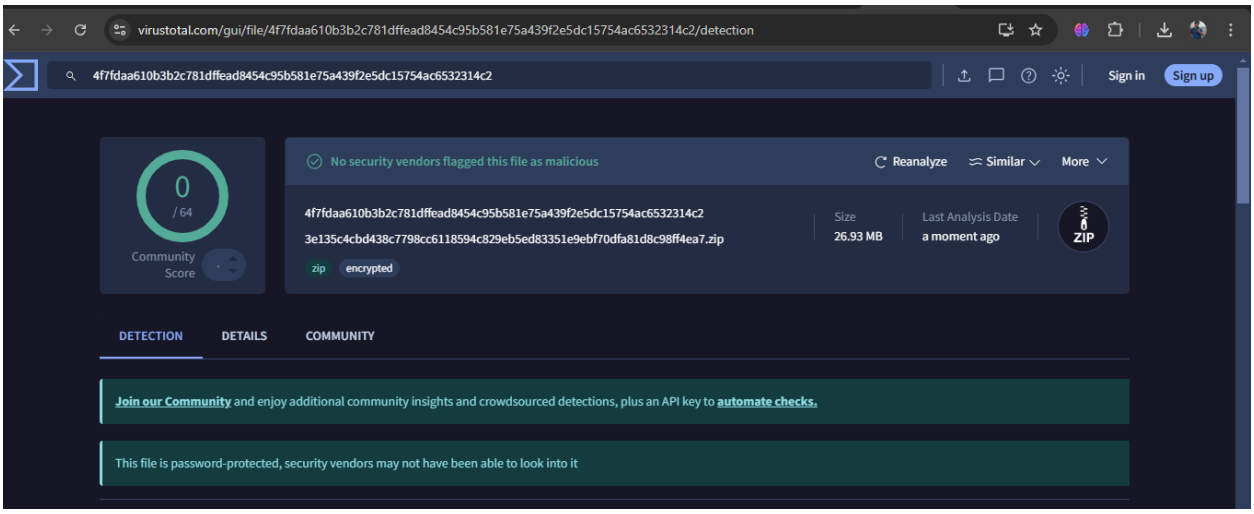
Analysis info

Functions:	284
X-Refs:	844549
Calls:	217
Strings:	511
Symbols:	0
Imports:	0
Analysis coverage:	7764 bytes
Code size:	28242238 bytes
Coverage percent:	0.0274907%

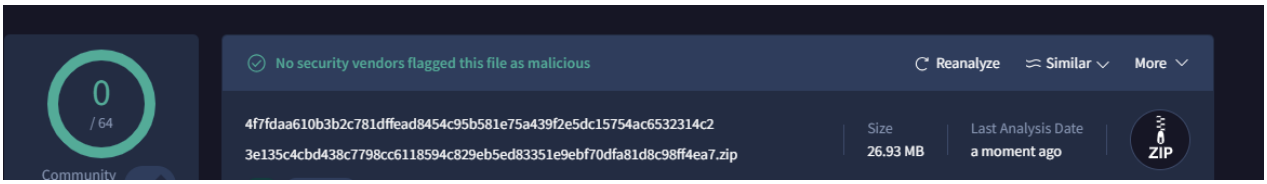
Libraries

mports | Search | Disassembly | Graph(fcn.00000000) | Hexdump | Decompiler (Empty)

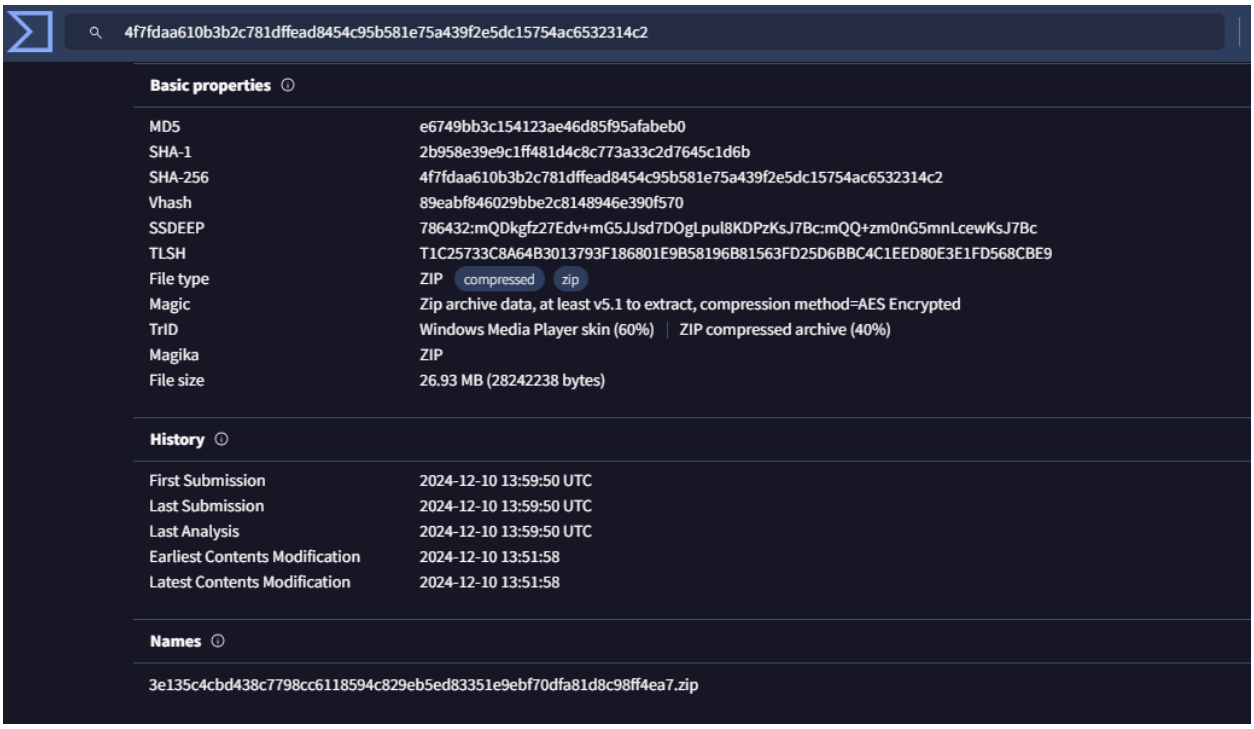
Now for dynamic analysis, I used virus total platform

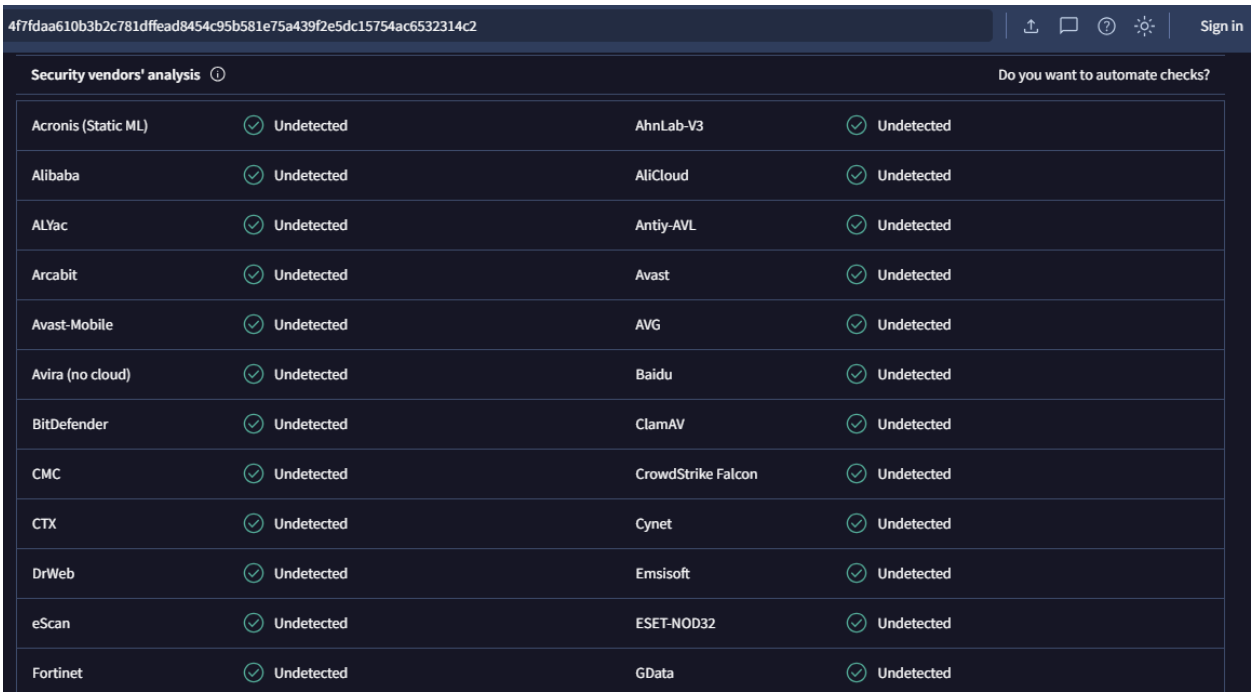


Not Malicious!



Properties, history and details





4f7fdaa610b3b2c781dffead8454c95b581e75a439f2e5dc15754ac6532314c2

Sign in

Security vendors' analysis

Do you want to automate checks?

Acronis (Static ML)	Undetected	AhnLab-V3	Undetected
Alibaba	Undetected	AliCloud	Undetected
ALYac	Undetected	Antiy-AVL	Undetected
Arcabit	Undetected	Avast	Undetected
Avast-Mobile	Undetected	AVG	Undetected
Avira (no cloud)	Undetected	Baidu	Undetected
BitDefender	Undetected	ClamAV	Undetected
CMC	Undetected	CrowdStrike Falcon	Undetected
CTX	Undetected	Cynet	Undetected
DrWeb	Undetected	Emsisoft	Undetected
eScan	Undetected	ESET-NOD32	Undetected
Fortinet	Undetected	GData	Undetected

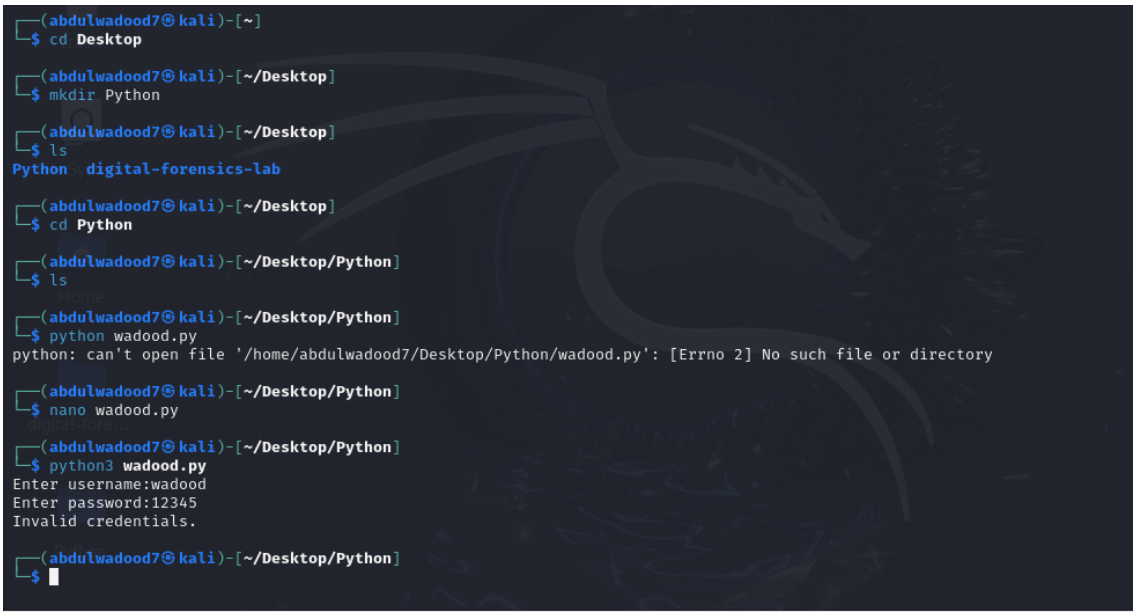
Task 5. Secure Coding Practices

Insecure Python script

```
import os
def save_user_data(username, password):
    # Hardcoded credentials
    if username == 'admin' and password == 'admin123':
        with open('user_data.txt', 'a') as f:
            f.write(f'Username: {username}, Password: {password}\n')
        print("User data saved.")
    else:
        print("Invalid credentials.")
```

# Unsanitized input
username = input("Enter username: wadood")
password = input("Enter password: 12345")
save_user_data(username, password)

Making a python file and running this script in it, and we can see the output as well.



Insecure Practices in the Code

1. **Hardcoded Credentials**: The script uses hardcoded credentials (admin and admin123), which can be easily exploited.
2. **Unsanitized Input**: User inputs are not sanitized, making the application vulnerable to injection attacks.
3. **Insecure Data Storage**: Storing sensitive information (like passwords) in plain text is a significant security risk.

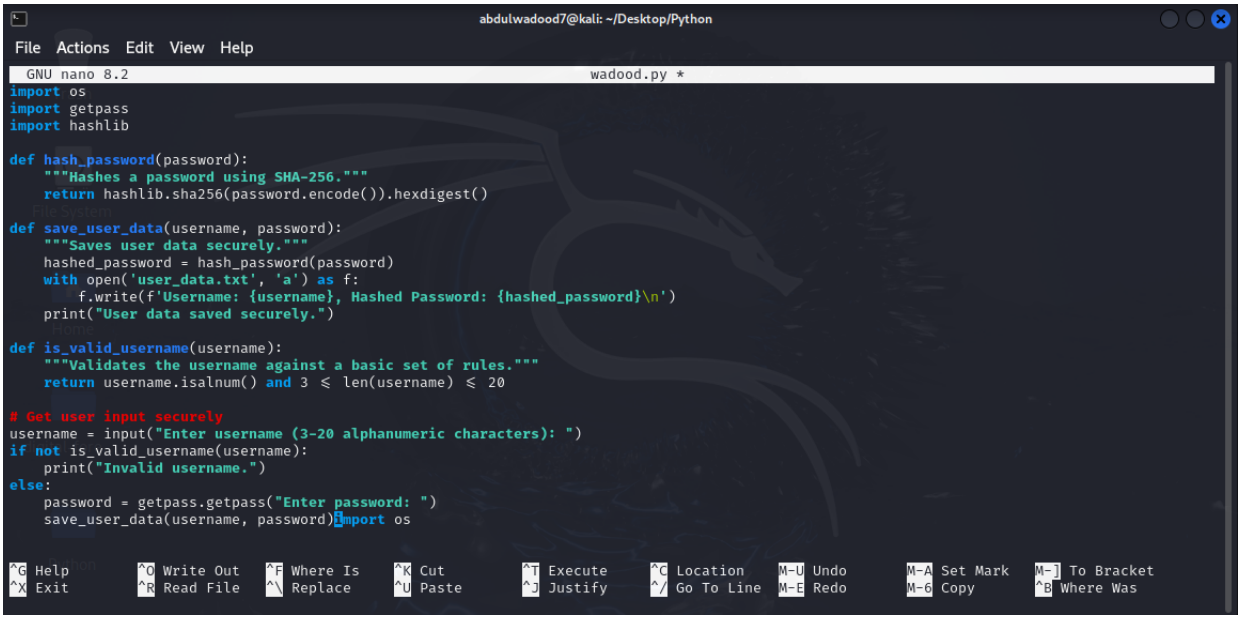
Now revising the following changes

```
┌──(abdulwadood7㉿kali)-[~/Desktop/Python]
└─$ nano wadood.py

┌──(abdulwadood7㉿kali)-[~/Desktop/Python]
└─$ python3 wadood.py
  File "/home/abdulwadood7/Desktop/Python/wadood.py", line 26
    save_user_data(username, password)import os
                                      ^^^^^^
SyntaxError: invalid syntax

┌──(abdulwadood7㉿kali)-[~/Desktop/Python]
└─$ python3 wadood.py
  File "/home/abdulwadood7/Desktop/Python/wadood.py", line 26
    save_user_data(username, password)import os
                                      ^^^^^^
SyntaxError: invalid syntax

┌──(abdulwadood7㉿kali)-[~/Desktop/Python]
└─$
```

Revised Secure Python Script:

```
import os
import getpass
import hashlib
def hash_password(password):
    """Hashes a password using SHA-256."""
    return hashlib.sha256(password.encode()).hexdigest()

def save_user_data(username, password):
    """Saves user data securely."""
    hashed_password = hash_password(password)
    with open('user_data.txt', 'a') as f:
        f.write(f'Username: {username}, Hashed Password: {hashed_password}\n')
    print("User data saved securely.")
def is_valid_username(username):
    """Validates the username against a basic set of rules."""
```

```
    return username.isalnum() and 3 <= len(username) <= 20
# Get user input securely
username = input("Enter username (3-20 alphanumeric characters): ")
if not is_valid_username(username):
    print("Invalid username.")
else:
    password = getpass.getpass("Enter password: ")
    save_user_data(username, password)
```



Now we can see the script after running it

Summary of Changes Made

1. **Removed Hardcoded Credentials**: The script no longer uses hardcoded credentials, which mitigates the risk of unauthorized access.
2. **Password Hashing**: Implemented a hash_password function to hash passwords using SHA-256 before storing them. This ensures that sensitive data is not stored in plain text.
3. **Input Validation**: Added a function (is_valid_username) to validate the username input, ensuring it meets specific criteria (alphanumeric and length).
4. **Secure Password Input**: Used getpass.getpass() to prompt for the password, preventing it from being displayed on the screen, enhancing security during input.

By addressing these vulnerabilities, the revised code follows more secure coding practices, reducing the risk of common attacks.