Penetrasjontest Rapport

For Kunden Boris Lockpicks Webstore

Innholdsfortegnelse

Introduksjon

Boris Lockpicks er vår kunde som har utviklet en nettbutikk som nå er i beta-versjon. Vi har tidligere jobbet på prosjektet og har derfor kjennskap til koden fra tidligere. Målet med penetrasjons testen er å gjennomføre en sikkerhetsvurdering av webapplikasjonen. Formålet er å evaluere sårbarheter i webapplikasjonen. Dette hjelper kunden til å styrke nettbutikkens sikkerhet før lansering.

Metodikk.

Metoden vi har valgt å bruke for denne penetrasjons testen er en hvit-boks-testing. Denne metoden gir oss en oversikt over hvordan applikasjonen er bygd opp. I tillegg har vi muligheten til å gjennomføre en statisk kodeanalyse, som hjelper oss med å identifisere sårbarheter direkte i kildekoden uten å kjøre applikasjonen. Kildekoden for applikasjonen er tilgjengelig i filen:
ETH2100_H24_mappeeksamen_del2_source.zip

Mål IP-adresse

Under hele penetrasjons testen endret IP adressen seg stadig. IP-adresser brukt under testing er følgende:

- 192.168.177.139
- 192.168.177.140
- 192.168.177.141
- 192.168.177.142
- 192.168.177.143

Serverinformasjon

Abyss/2.16.9.1-X1-Linux AbyssLib/2.16.9.1

Verktøy brukt under oppdrag

- Nmap - Scann for å finne åpne porter som kjører
- Nikto - Scann for å teste sårbarheter på port 80 (-h)
- SQLmap - Scann for å teste SQL injeksjon, sensitive data i SQL databaser
- OWASP ZAP – generell sårbarhet scann på IP adressen

Gjennomføring

Vi startet med å planlegge hvordan vi skulle gjennomføre penetrasjons testen. I starten lagde vi en oversikt over de testene vi ønsket å utføre på applikasjonen. Vi planla hvordan vi skulle finne sårbarheter knyttet til nettsiden ved å lage et sjekkpunkt liste over ulike sårbarheter vi ønsket og teste som for eksempel XSS. Før vi kunne begynne med testene, måtte vi finne IP-adressen til systemet. Dette gjorde vi ved å bruke Windows PowerShell på PC-en. Vi skrev inn kommandoen arp -a, som viste en liste over IP-adresser. For å finne den riktige IP-adressen, søkte vi opp adressene og identifiserte den som førte oss til Boris Lockpicks sin nettside.

Deltakere

Testen ble utført av kandidat 50192, fra selskapet Etisk Hackere, på oppdrag for kunden Boris Lockpicks.

Sammendrag funn

Det ble funnet store sårbarheter under testingen. Dette inkluderer flere typer SQL-injeksjoner som gjorde det mulig å hente ut sensitiv data informasjon om brukere. Det ble oppdaget manglende

sikkerhetsfunksjoner som b.la anti-CSRF tokens og Content Security Policy. Dette har økter risikoen for angrep som CSRF og datatyveri. Det er mulighet for å kjøre XXS på applikasjonen. Det har også vært en del feilmeldinger som har avslørt sensitiv databaseinformasjon.

Fargekorrigering av alvorlighetsgrad

Rødt: høy risiko	
Orange: moderat risiko	
Gult: Lav Risiko	
Blått: Informasjon	

Identifisering styrker og svakheter

Styrker	Svakheter
Tydelig struktur	Kritiske SQL-injeksjonsmuligheter
Muligheter for å gjennomføre testing effektivt	Manglende sikkerhetsfunksjoner
Statisk kodeanalyse	Feilmeldinger avslører sensitiv info
Tilgang til kildekoden	Cookies mangler viktige sikkerhetsflagg
	Ingen anti-clickjacking header

Detaljert sårbarhets oversikt

Sårbarhet	Alvorlighetsgrad
Nmap Portskanning	**Høy**
SQL-MAP injeksjon	**kritisk**
Boolean-based blind SQL Injeksjon	**Kritisk**
Error-based SQL Injeksjon	**Kritisk**
Time-Based blind SQL Injeksjon	**Kritisk**
UNION-based SQL Injection	**Kritisk**
Uautorisert Tilgang til Backend-Database	**Kritisk**
Eksponering av Sensitiv Informasjon	**Kritisk**

Tilgangskontroll for Medlemsinnhold	**Kritisk**
Manglende Anti-CSRF Tokens	Høy
Application Error Disclosure	Høy
Content Security Policy (CSP) Header Ikke Satt	Høy
Manglende Anti-Clickjacking Header	Høy
Stor Redirect Detected (Potensiell Sensitiv Informasjonslekkasje)	Lav
Cookie No HTTPOnly Flag	Lav
Cookie Without Secure Flag	Lav
Cookie Without SameSite Attribute	Lav
Server Leaks Version Information via "Server" HTTP Response Header Field	Lav
Strict-Transport-Security Header Not Set	Høy
Informasjon	

Teknisk gjennomgang av skanner

Nmap port skanning

```
└─$ nmap 192.168.177.140
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-07 10:47 EST
Nmap scan report for 192.168.177.140
Host is up (0.40s latency).
Not shown: 989 closed tcp ports (conn-refused)
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

Nmap done: 1 IP address (1 host up) scanned in 7.73 seconds
```

Det første vi gjorde var å undersøke antall åpne porter på IP-adressen. Vi fant flere åpne porter som kan utnyttes.

Undersøkelse av porter

Gjennom testing av forskjellige porter på IP adressen ble det bekreftet at port **80 (HTTP)** og port **443 (HTTPS)** var åpne og tilgjengelige for videre analyse. De andre portene viste ingen sårbarheter eller informasjon som kunne utnyttes. Eksempel på dette er b.la port 9 (discard) bildet viser at angrepet ikke hadde en ingen synlig effekt. Dette kan være grunnet sikkerhetstiltak på målet.

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 -S --flood -V -p 80 192.168.177.143

[sudo] password for kali:
using eth1, addr: 192.168.177.134, MTU: 1500
HPING 192.168.177.143 (eth1 192.168.177.143): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

```
64 bytes from 192.168.177.143: icmp_seq=357 ttl=64 time=1.16 ms
64 bytes from 192.168.177.143: icmp_seq=358 ttl=64 time=1.22 ms
64 bytes from 192.168.177.143: icmp_seq=359 ttl=64 time=0.494 ms
```

SQL-MAP injeksjons

- Vi brukte OWASP ZAP for å skanne nettsiden på følgende IP adresse 192.168.177.142. Testen viste at det mangler en anti-CSRF tokens. Denne sårbarheten er knyttet til en URL https://192.168.177.142/store_viewdetails.php?id=4
- For videre undersøkelser testet vi om ID er sårbar for SQL injeksjon, noe vi oppdaget var positivt. Vi gjennomførte vi en SQLmap-test på samme URL, og resultatene viste at applikasjonen er sårbar for flere typer SQL-injeksjon.

```
[11:45:53] [INFO] GET parameter 'id' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] y
sqlmap identified the following injection point(s) with a total of 52 HTTP(s) requests:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=4 AND 7056=7056

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: id=4 AND (SELECT 9706 FROM(SELECT COUNT(*),CONCAT(0x71787a6271,(SELECT (ELT(9706=9706,1))),0x71716a7171,FLOOR(RAND(0)*2))x FROM INFORM
ATION_SCHEMA.PLUGINS GROUP BY x)a)

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=4 AND (SELECT 1510 FROM (SELECT(SLEEP(5)))WLly)

    Type: UNION query
    Title: Generic UNION query (NULL) - 4 columns
    Payload: id=-3466 UNION ALL SELECT NULL,NULL,NULL,CONCAT(0x71787a6271,0x705244786759796372677678626555 5a4b725352515077494f5251524a4e58745359776
2596f4d56,0x71716a7171)-- -
---
[11:49:08] [INFO] the back-end DBMS is MySQL
web server operating system: Linux
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
[11:49:08] [INFO] fetching database names
[11:49:08] [INFO] retrieved: 'information_schema'
[11:49:08] [INFO] retrieved: 'borislockpicks'
[11:49:08] [INFO] retrieved: 'mysql'
[11:49:08] [INFO] retrieved: 'performance_schema'
available databases [4]:
[*] borislockpicks
[*] information_schema
[*] mysql
[*] performance_schema
```

Boolean-based blind SQL Injeksjon

Beskrivelse	Trekker data ved å sjekke om noe er sant eksempel: Id=6 and 7360=7360
Forbedring	Lett til spørringer for å hindre injeksjon av brukerinput.
System	Abyss/2.16.9.1
Referanser	SQL Injection Attacks – How to Use SQLMap to Find Database Vulnerabilities

Error-based SQL Injeksjon

Beskrivelse	Rask tilgjengelig data via feilmeldinger. Eksempel id=6 AND (SELECT COUNT (*), CONCAT (….))
Forbedring	Skjul detaljerte feilmeldinger fra brukere for å redusere informasjonslekkasje.
System	Abyss/2.16.9.1
Referanser	SQL Injection Attacks – How to Use SQLMap to Find Database Vulnerabilities

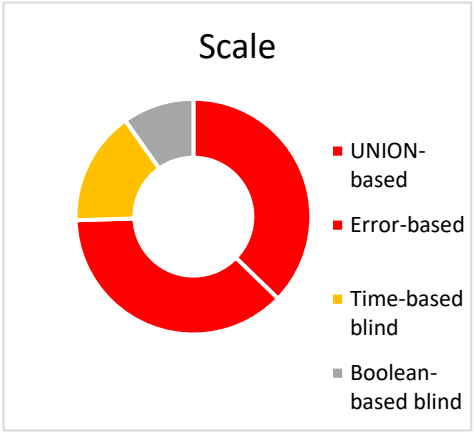
Time-Based blind SQL Injeksjon

Beskrivelse	Bruker forsinkelser for å hente data. Eksempel: id=6 AND (SELECT SLEEP (5))
Forbedring	Bruk spørringer og begrens tidsbaserte funksjoner i spørringer.
System	Abyss/2.16.9.1
Referanser	SQL Injection Attacks – How to Use SQLMap to Find Database Vulnerabilities

UNION-based SQL Injection

Beskrivelse	Henter data fort med UNION spørring. Eksempel id=5560 UNION ALL SELECT NULL, …

Forbedring	Begrens brukertilganger og sjekk all input strengt.
System	Abyss/2.16.9.1
Referanser	SQL Injection Attacks – How to Use SQLMap to Find Database Vulnerabilities



Uautorisert tilgang til backend-Database

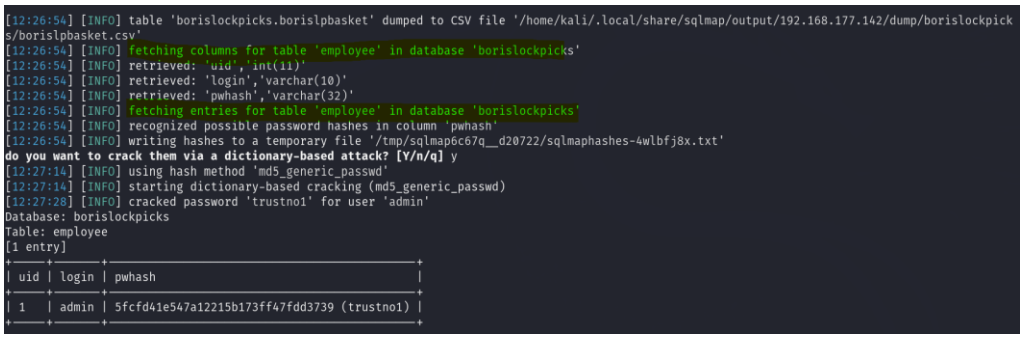
Oppdaget fire tilgjengelige databaser på backend-serveren. En av disse er borislockpicks. Noe som vekket en interesse for å teste hva som befant seg i denne data basen. For å undersøke innholdet i databasen, benyttet vi --dump-all kommandoen i sqlmap under URL: https://192.168.177.142/store_viewdetails.php?id=4, som ga tilgang til informasjon om hele databasen. Denne gjennomgangen avslørte sensitive tabeller som kunde og ansatte. Som videre kan inneholde sensitive data.

Eksponering av sensitiv informasjon

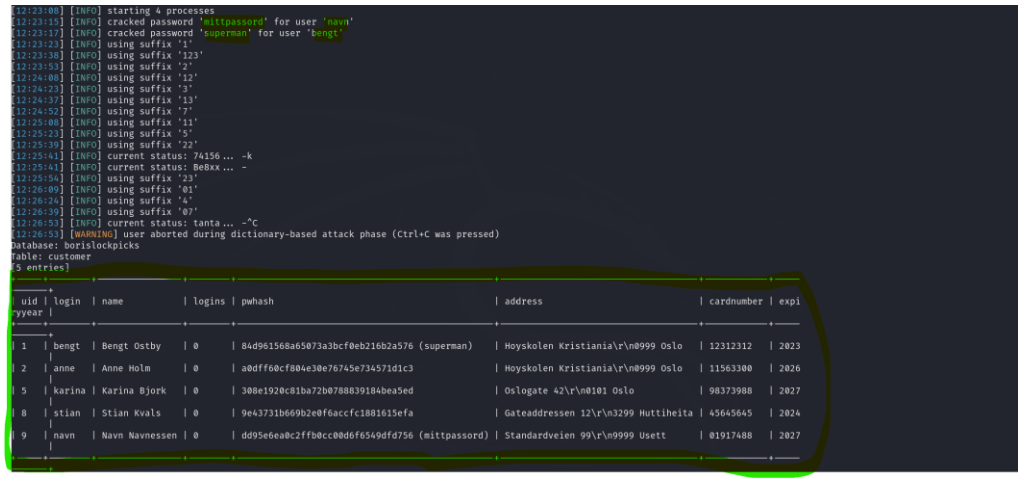
Testen avslørte tabeller som "customer" og "employee" som er sårbare. Vi fikk hentet ut sensitive data som navn, logging og passord hash, adresse og kort informasjon. Dette er en kritisk, her blir eksponering av sensitiv informasjon tilgjengelig.

Henter informasjon fra tabellen customer

Henter ut informasjon fra tabellen employee



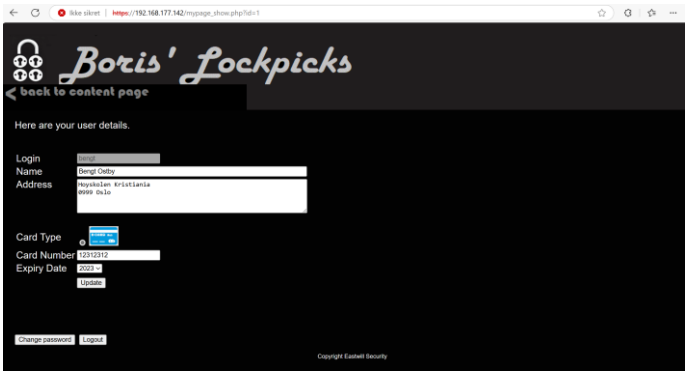
Sensitiv data informasjon



Tilgangskontroll for medlemsinnhold

Vi testen om brukernavn og passord gikk over rens med det skanningen viste. På innloggingssiden til Boris' Lockpick skrev vi inn brukernavn "bengt" og passord "superman". Dette ga oss tilgang til Bengts brukerprofil, hvor vi kunne se eksklusivt innhold om låsåpning, opplæringer og annet kun tilgjengelig for medlemmer. Vi fikk tilgang gjennom en annen bruker som er registrert som medlem,

noe som avslører en alvorlig sikkerhetssvakhet i applikasjonen



Mangled Anti-CSRF Tokens

Beskrivelse	Mulighet til å utføre CSRF-angrep. Nettsiden stoler på at forespørselen kommer fra brukeren, uten noen form for sjekk.
System	Abyss/2.16.9.1-X1-Linux AbyssLib/2.16.9.1
Referanser	ZAP – Anti-CSRF Tokens Check

Utbedring

Hvem	Utviklere
Handling	Oppfordrer bruk anti-CSRF-pakker. Ta hensyn til at applikasjonen er fri for XSS. Sjekk HTTP Referer-headeren for å være sikre for at forespørsler kommer fra riktig sted.

Eksempel

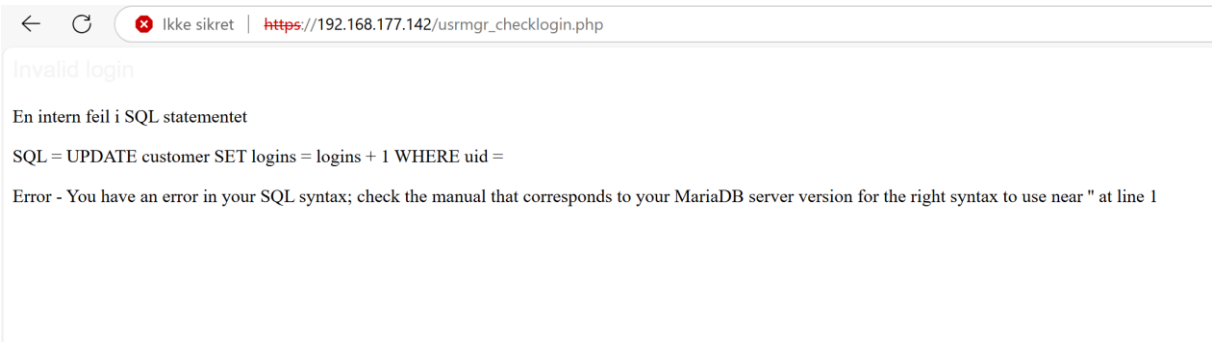
```
   <p align="center"><img border="0" src="store/products/5/image_large.png"></p>
<form name="buyproduct" action="" method="post" onsubmit="return checkqty();"><input type="hidden" name="id" value="5">
<table align=center width=500><tr><td colspan="6" align=left>
<div style="color: #F5F5F5; font-family: Arial;font-size: 20px;">The standard set</div>
</td></tr><tr><td align=left colspan="3">
<p style="color: #F5F5F5; font-family: Arial;font-size: 20px;">set1</p>
</td><td width=100%> </td><td>
<p><img border="0" src="store/products/5/image_small.png">
</td><td><p style="color: #F5F5F5; font-family: Arial;font-size: 20px;">99.50</p>
</td></tr>
      <tr><td> </td></tr>
```

- Manglende Anti-CSRF-token, som OWASP ZAP påpekte. Skjemaet har en id-verdi som er skjult (<input type="hidden" name="id" value="5">)

Application Error Disclosure

Beskrivelse	Siden har feil melding som kan avsløre sensitiv informasjon. Men varslingen kan være en falsk positiv.
Alvorlighetsgrad	Middels
System	Abyss/2.16.9.1
Referanser	ZAP – Application Error Disclosure

Vi ønsket i midlertidig og teste usrmgr_checklogin.php direkte i vår URL. Siden viser en avslører informasjon om databaseforespørsler og SQL-syntaks.



Feilmeldingen avslører SQL-setninger, samt detaljer om syntaksfeilen, som gir oss innsikt i databasen. Her øker risikoen for angrep som b.la SQL-injeksjon.

Utbedring

Hvem	Utviklere
Handling	Detaljene om feilmelding bør holdes på et skjult sted der autoriserte eiere av applikasjonen har tilgang. Vi anbefaler en sikkerhetssjekk av brukerinndata for å beskytte mot SQL-injeksjon.

Content Security Policy (CSP) Header ikke satt

Beskrivelse	Risiko for angrep, som kan brukes til alt fra datatyveri til endring av nettsidens innhold. Det er også risiko for spredning av skadelige programvarer.
System	Abyss/2.16.9.1
Referanser	ZAP – CSP: X-Content-Security-Policy

Utbedring

Hvem	Utviklere
Handling	Sørg for å sette opp web serveren, applikasjon serveren og load balancer riktig slik at dere kan sette opp csp-headeren riktig. dette gir dere en oversikt over hvilke kilder som har tilgang til å laste ned innhold som for eksempel bilder osv... På denne måten har dere mulighet til å kontrollere hvilke ressurser nettleser kan laste. Dermed reduseres risikoen for angrep mot applikasjonen

Manglende Anti-clickjacking Header

Beskrivelse	Risiko for klikkjacking og MIME-type manipulering
System	Abyss/2.16.9.1
Referanser	ZAP – Missing Anti-clickjacking Header Nikto Cheat Sheet - Commands & Examples

Utbedring

Hvem	Utviklere
Handling	Vi oppfordrer dere til å sette opp X-Frame-Options og X-Content-Type-Options. Disse hjelper med å redusere risikoer med nettsiden knyttet til b.la klickjacking og MIME-type manipulering

Stor Redirect Detected (potensiell sensitive informasjonslekkasje)

Beskrivelse	Serveren har sendt en Redirect-respons som ser ut til å gi et stort svar. Den har også lagt ved mye informasjon som kanskje er sensitiv eller hemmelig.
Sytem	Abyss/2.16.9.1
Referanser	ZAP – Big Redirect Detected (Potential Sensitive Information Leak)

Utbedring

Hvem	Utviklere
Handling	Sørg for at ingen sensitiv informasjon lekkes via Redirect-respons. Redirect-respons bør ikke inneholde mye informasjon.

Cookie No HttpOnly Flag

Beskrivelse	JavaScript kan nå Cookies, Skadelig skript kan kjøres på siden. Informasjonen blir da tilgjengelig og sendt til en annen side. Øker risiko for hijacking.
URL	https://192.168.177.142/store_addtobasket.php?id=1
Referanser	ZAP – Cookie No HttpOnly Flag

```
HTTP/1.1 302 Moved Temporarily
Set-Cookie: borislp_basket=107
Location: store.php
Content-type: text/html; charset=UTF-8
Date: Thu, 14 Nov 2024 09:24:41 GMT
Server: Abyss/2.16.9.1-X1-Linux AbyssLib/2.16.9.1
content-length: 435
```

Utbedring

Hvem	Utviklere
Handling	Sørg for at det ikke er sensitive opplysninger som blir lekka gjennom Redirect-respons. Redirect-responser bør også kun inneholde minimalt med innhold

Cookie Without Secure Flag

Beskrivelse	Mulig for angripere å kjøre skadelig kode i brukerens nettleser. Risiko for til tyveri av sensitiv informasjon.
Alvorlighetsgrad	lav
System	Abyss/2.16.9.1

Utbedring

Hvem	Utviklere
Handling	Pass på at HTTPOnly flag er satt for alle informasjonskapsler

Cookie without SameSite Attribute

Beskrivelse	Informasjon kan bli tilgjengelig via usikkre tilkoblinger.
Alvorlighetsgrad	Lav
URL	https://192.168.177.142/store_addtobasket.php?id=1

Utbedring

Hvem	Utviklere
Handling	Sørg for at sikkerhetsfag er aktivert for Cookies som inneholder sensitiv informasjon

Server Leaks Version Information via "Server" HTTP Response Header Field

Beskrivelse	Cookies kan sendes ved forespørsler fra andre nettsteder. Uten same Site attributt kan angrep som CSRF skje.
Alvorlighetsgrad	lav
System	https://192.168.177.142/store_addtobasket.php?id=1

Utbedring

Hvem	Utviklere
Handling	Sørg for at SameSite-attributtet er satt til enten "lax" eller "strict" for alle informasjonskapsler

Strict-Transport-Security Header Not Set

Beskrivelse	Beskytter brukerne mot angrep som man-in-the-middle ved å forhindre at data overføres uten kryptering.
Alvorlighetsgrad	Medium
System	https://192.168.177.142/images/search.png
Referanser	HTTP Strict Transport Security - OWASP Cheat Sheet Series

Utbedring

Hvem	Utviklere
Handling	Sørg for at webserveren og applikasjonsserveren er konfigurert til å håndheve Strict-Transport-Security (HSTS).

Informasjon

IP-adresse i "server"-header	**Falsk sårbarhet**: Rapporterer IP 2.16.9.1, men dette er en serverversjon, ikke en IP-adresse.

Eldre programvareversjoner	**Falsk sårbarhet**: Flere sårbarheter relatert til spesifikke, eldre versjoner av programvare som ikke er installert eller relevant for systemet som ble testet.
Falske positiver (nikto scann)	Det ble avdekket en rekke potensielle sårbarheter på port 80. Noen av disse var falske positiver.
Authentication Request Identified **https://192.168.177.142/mypage_login.php**	Forespørselen er en autentiseringsforespørsel. Hvis autentisering er satt til "Auto-Detect", justeres den automatisk for å passe forespørselen.

Informasjonslekkasje - Mistenkelige kommentarerhttps://192.168.177.142/store_viewdetails.php?id=2	`<script language="javascript">`
	`function checkqty() {`
	`if (isNaN(document.buyproduct.quantity.value)) {`
	`alert ("Only numbers are allowed as quantity`
	
	Avslører informasjon som kan hjelpe angripere med å utføre uønskede handlinger på nettsiden.

Re-examine Cache-control Directives	Cache-control-headeren er ikke satt riktig, noe som gjør at nettleseren og proxy kan lagre innhold.
User Controllable HTML Element Attribute (Potential XSS) **https://192.168.177.142/guestbook.php**	På siden: https://192.168.177.142/guestbook.php ble det funnet brukerinput i en [input]-tag, spesifikt i type-attributtet. **Input funnet:** submit=Submit **Brukerkontrollert verdi:** Submit For å teste om XSS er mulig. Bør undersøkes videre for å vurdere risikoen for et XSS-angrep.