

Rapport

TEKNISK SIKKERHETSREVISJON

Kandidatnr:1057
FLO BLOMSTERBUTIKK AS
03.11.2022

EXECUTIVE SUMMARY

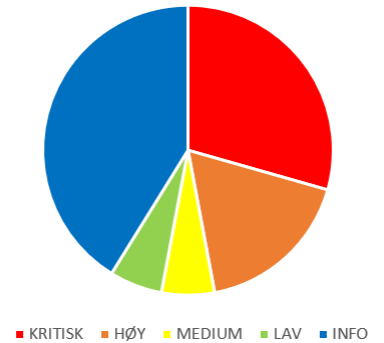
Denne testen er utført av Høyskolen Kristiania på vegne av Flo Blomsterbutikk AS. Hele webapplikasjonen er sikkerhetsrevidert. Det ble utført en White Box-test av webserveren og hensikten med denne testen er å identifisere og utnytte sårbarhetene som ble funnet i webapplikasjonen for å ta en sikkerhetsvurdering som skal bidra med å sikre applikasjonen mot fremtidige hackerangrep. Det ble med hensyn til Owasp Testing Guide, brukt pålitelige verktøy under testingen.

Webapplikasjonen har generelt sett et lavt nivå av sikkerhet og Flo Blomsterbutikk AS benytter seg av sikkerhetskomponenter som autentisering for å oppnå et optimalt nivå av sikkerhet på webserveren. Den underliggende infrastruktur har en usolid oppbygning og serveren består av en del sårbarheter, ukrypterte innhold og eksponerte og utdaterte servere, så det oppfordres sterkt til å oppdatere serveren regelmessig med nødvendige sikkerhetspatcher for å tette sikkerhetshullene i applikasjonen. Men også for å bevare konfidensialiteten, integriteten og tilgjengeligheten i systemet.

CONTENTS

Executive Summary.....	1
Sårbarhetsfunn	3
Gjennomførelse	6
Kontraktsregel.....	6
Deltagere	6
IP adresser brukt under testingen	6
Detaljert beskrivelse av funn	7
KRITISK	7
HØY	15
MEDIUM	19
LAV	20
INFORMASJON	22

SÅRBARHETSFUNN



Sårbarhet 1: Eksponert Nettverk-trafikk over HTTP

Kritisk: Applikasjonen sender data ukryptert over HTTP

Sårbarhet 2: Port 42042 eksponerer sensorveiledningen

Kritisk: Sensorveiledning funnet i klartekst

Sårbarhet 3: XSS i dialogfeltet på applikasjonen

Kritisk: Cross Site Scripting

Sårbarhet 4: Uautorisert databasetilgang

Kritisk: SQL injection

Sårbarhet 5: Uautorisert Admin tilgang

Kritisk: Admin tilgang

Sårbarhet 6: Brute Force

Høy: Passord knekt med sqlmap

Sårbarhet 7: Clickjacking sårbarhet på serveren

Høy: Clickjacking

Sårbarhet 8: Dårlig håndtering av passord

Høy: Ingen passordbeskyttelse

Sårbarhet 9: OpenSSH sårbarhet

Medium: Webserveren kjører en OpenSSH versjon 8.4
p1

Sårbarhet 10: HTTPOnly og secure Flag er ikke definert

Lav: Udefinerte Cookie parametere

Info 1: Informasjon om systemet

Info: Systeminformasjon lekket

Info 2: Server eksponert

Info: Informasjon om Abyss serveren ble eksponert

Info 3: Nmap eksponerer åpen port

Info: Port 80 er åpen.

Info 4: Nmap eksponerer åpen port

Info: Port 22 er åpen.

Info 5: GNOME versjon eksponert

Info: Systeminnstillinger

Info 6: Webserver tilgang

Info: Register siden

Info 7: Hemmelig bufferoverflow eksponert på serveren

Info: Bufferoverflow

GJENNOMFØRELSE

Testen retter seg mot følgende inngangspunkt:

- IP 192.168.10.35

Tester har fått tilgang til følgende brukere/kontoer:

- Tester har ikke fått tilgang til noen brukere

KONTRAKTSREGEL

Kunden har sammen med Høyskolen Kristiania inngått en avtale. Avtalen er signert med hensyn til SECURITY ASSESMENT AGREEMENT. Oppdraget er datert 31.10.22 og dekker perioden 1.11.22 til 19.12.22.

DELTAGERE

Disse testene ble gjennomført av eksamenskandidat:1057, Cyber Security student hos Høyskolen Kristiania.

IP ADRESSER BRUKT UNDER TESTINGEN

Følgende eksterne IP adresser er blitt brukt under testingen:

- 192.168.10.35

DETALJERT BESKRIVELSE AV FUNN

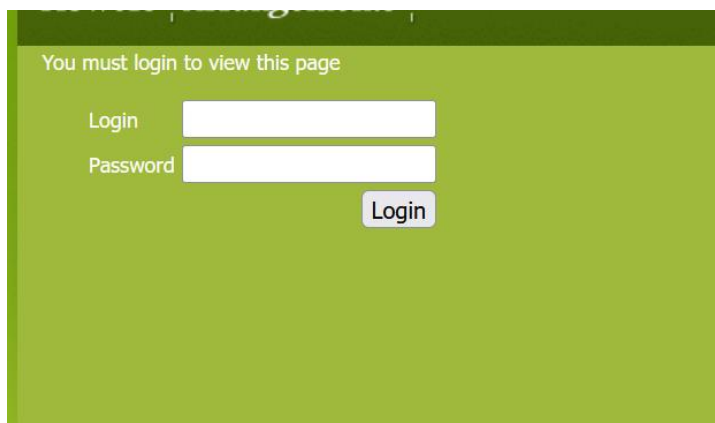
KRITISK

Sårbarhet 1: Eksponert Nettverk-trafikk over HTTP

Kritisk: Applikasjonen sender data ukryptert over HTTP

Observasjon:

Det ble observert at serveren sender login credentials ukryptert over http.



(Vedlegg 1.1: Innloggingsfelt på webserveren)

Påvirket webområde:

<http://localhost/login.php>

Beskrivelse:

Login.php siden sender innloggingsinformasjon over den ukrypterte protokollen HTTP. Dette kan resultere til at en angriper kan få tak i login credentials i klartekst ved å lytte til

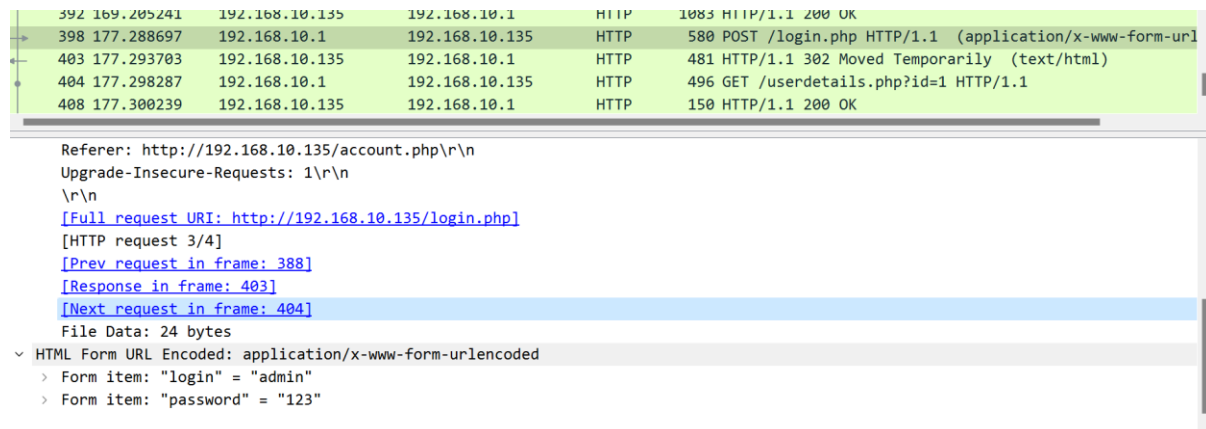
nettverket med et Man -In -The -Middle angrep. Angriper kan dermed hente inn informasjonen direkte etter brukerens inntasting og kan dermed bruke dette til å logge seg på siden med brukerens login credentials.

Anbefalte tiltak:

Det oppfordres til at Flo AS oppgraderer HTTP til å HTTPS på port 443 som bidrar med å sende nettverkstrafikk kryptert ved bruk av TLS/SSL protokollene.

Testing av exploit:

Ved bruk av Wireshark, som er et verktøy for lytting av nettverkstrafikk, kan man lytte på IP adressen til webområde: <http://localhost/login.php>. Man kan se spørringene som går mellom klienten og serveren ved å filtrere inn: «ip.addr==localhost» i Wireshark for å lytte spesifikt til denne IP-adressen. Under vises det både brukernavnet og passordet til brukeren i klartekst som kan misbrukes av angriperne til å få uautorisert tilgang til kontoer.



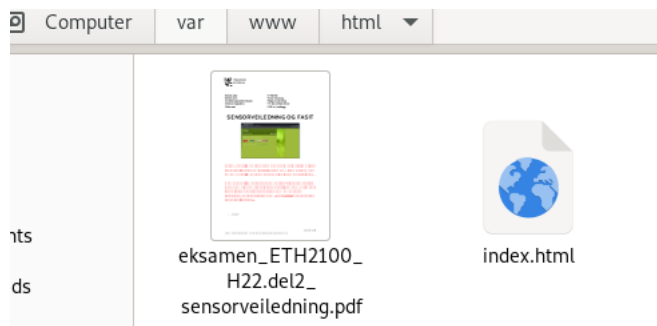
(Vedlegg 1.2: brukernavnet og passordet til brukeren står i klartekst)

Sårbarhet 2: Port 42042 eksponerer sensorveiledninger

Kritisk: Sensorveiledning funnet i klartekst

Observasjon:

Det ble observert at serveren har en port 42042 som eksponerer sensorveiledningen i PDF format (Se vedlegg 2.1).



(Vedlegg 2.1: filplasseringen PDF filen var befant seg i.)

Emnekode: ETH2100
Emnenavn: Etisk Hacking
Vurderingskombinasjon: Mappevurdering
Innleveringsdato: 19. desember 2022
Filformat: PDF m/ vedlegg

SENSORVEILEDNING OG FASIT

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

Men denne filen er lagt her på en ikke-standard port for at de som gjør en grundig jobb skal finne den. Dette skal rapporteres i pentest rapporten som en KRITISK sårbarhet, og rapporteres som «Port 42042 eksponerer sensorveiledning».

(Vedlegg 2.2: PDF filen står i klartekst med en beskjed om at port 42042 eksponerer sensorveiledningen.)

Påvirket filområde:

/var/www/html/

Beskrivelse:

Det er en stor feil av Flo AS å ha sensitive filer slik som denne PDF-filen å stå tilgjengelig åpent for brukere. Denne filen hadde definitivt vært ettertraktet for ETH2100 studentene hvis sensorveiledningen virkelig var en fasit på eksamensbesvarelsen til del 2 av eksamen. Hvis noen av studentene observerer denne filen kan den misbrukes til å få en bedre karakter

og dette hadde medført til at sensor mest sannsynlig ville ha endt opp med å rette veldig mange identiske besvarelser. Dermed kan det medføre til at studentene ville ha blitt tatt ut til fusk og risikere dem for utestenging uavhengig av hvor mange som fusket. Hvis alle studentene observerte denne filen, ville dette ha gitt konsekvenser for sensoren.

I et annet tilfelle kan være at hackeren laster denne filen opp på internett som gjør den offentlig tilgjengelig for alle brukere på internett uten autorisasjon fra firmaet eller brukeren. Hvis en angriper får tak i sensitive filer og data kan vedkommende utnytte disse dataene til noe ondsinnet, det kan være for eksempel å bryte seg inn på brukerens konto eller benytte bruker opplysningene til å eskalere ned på brukernes privilegier og utføre phishing angrep mot andre brukere og spre malware eller linker som manipulerer brukere til å oppgi login credentials som hackere kan utnytte for å få kontroll over andre kontoer.

Anbefalinger:

Det anbefales at kunden benytter seg av krypteringsmetoder for å sikre filene for å forhindre uautorisert tilgang fra angripere. Kjente krypteringsmetoder som Asymmetrisk (RSA kryptering) og Symmetrisk (AES kryptering) kryptering brukes om får å kryptere innholdet i filer og dokumenter. Det finnes mange krypteringsalgoritmer og de mest brukte algoritmene er AES og RSA for mer informasjon henvises det til referanse.

Referanser:

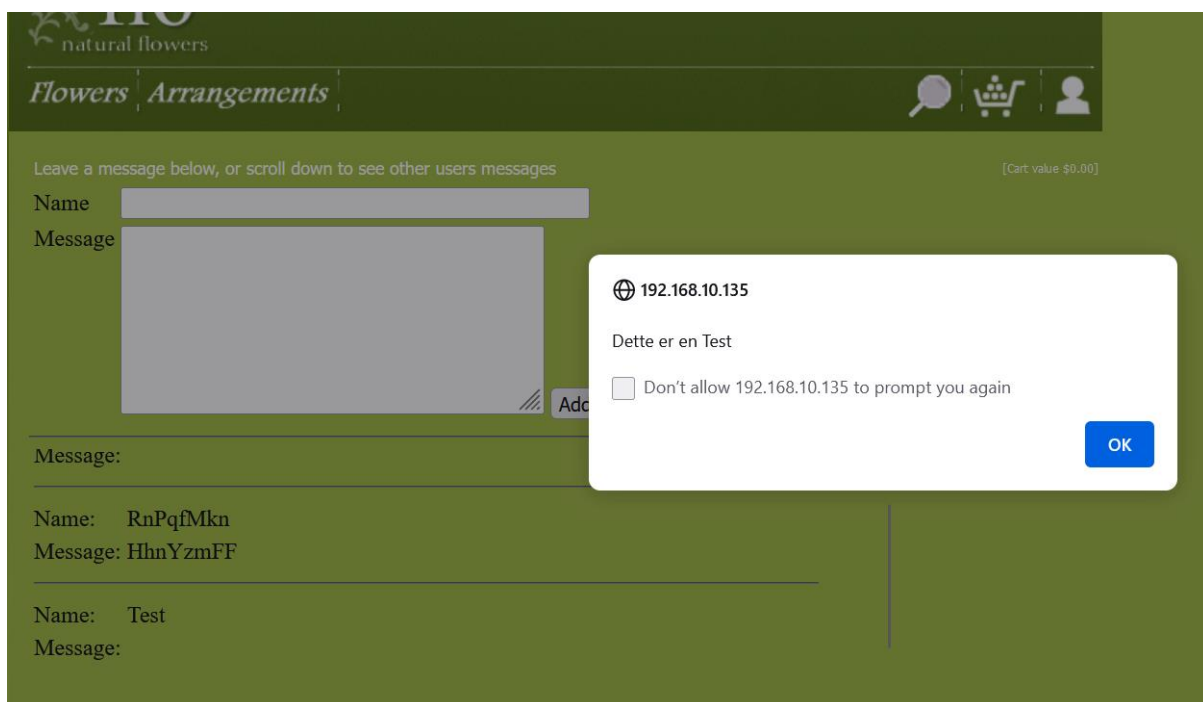
- <https://www.simplilearn.com/data-encryption-methods-article>
- <https://www.hp.com/us-en/shop/tech-takes/what-are-different-types-of-encryption#:~:text=The%20three%20major%20encryption%20types,that%20consumers%20use%20every%20day.>

Sårbarhet 3: XSS i dialogfeltet på applikasjonen

Kritisk: Cross Site Scripting

Observasjon:

Det ble oppdaget en Cross Site Scripting sårbarhet i meldingsboksen.



(Vedlegg 3.1: Med kommandoen: «<script>alert ("Dette er en Test") </script>» kan man kjøre et XSS angrep i meldingsboksen. Dette er et eksempel på at angrepet kan kjøres og utnyttes.)

Påvirket webområde:

<http://localhost/guestbook.php>

Beskrivelse:

Det er mulig å kjøre Cross Site Scripting på meldingstjenesten på webserveren uten å være innlogget. Cross Site Scripting muliggjør det for en angriper å injisere kode som kan skade eller distrahere webapplikasjonen. Dette angrepet kan blant annet brukes til å hente inn sensitiv informasjon, legge inn malware på webserveren, eller stjele sesjonen cookien og utføre en session hijacking som kan gi direkte tilgang til bruker uten innlogging. Hvis angriper får tak i admin brukeren kan vedkommende ta kontroll av hele webapplikasjonen og gjøre hva en de vil på vegne av administrasjonen som for eksempel å slette brukere og utføre ondsinnede handlinger. Det finnes mange former for XSS angrep, se referansene for mer informasjon.

Anbefalinger:

En måte å forhindre XSS angrep på er å sette en CSP-header i webapplikasjonen som bidrar med å detektere og redusere Cross Site Scripting angrep mot serveren. En annen ting kunden kan gjøre er å sette meldingstjenesten inni brukersiden istedenfor at den ligger åpent som bidrar med å minimere risikoen for XSS angrep. Siden koder består av spesielle tegn kan man sette inputfeltene til å kun akseptere bokstaver som ikke består av bokstavene (S, C, I, P, T, R) for å redusere en XSS angrep.

Referanser:

- <https://crashtest-security.com/cross-site-scripting-xss/>
- <https://www.acunetix.com/websitesecurity/xss/>

Sårbarhet 4: Uautorisert databasetilgang

Kritisk: Sql injection

Observasjon:

Verktøyet sqlmap detekterte en database på serveren og det ble oppdaget en brukertabell.

```
Table: users
[2 entries]
```

uid	name	login	address	password	cardnumber	expiryyear	expirymonth
1	Administrator	admin	No address provided	123	4111111111111111	2005	9
8	Mike Andrews	mike	742 Evergreen Terrace\r\nSpringfield, MA 12345	andrews	4111111111111111	2005	1

Vedlegg 4.1: Med kommandoen (sqlmap -u <http://localhost/userdetails.php?id=1> -D flowershop -T users - dump) ble det funnet en users tabell med sensitive opplysninger om brukerne i klartekst.

Påvirket webområde:

<http://localhost/userdetails.php?id=1>

Beskrivelse:

SQL injections er en teknikk som lar angripere å utføre spørringer mot en server som gir dem uautorisert tilgang en database. Dette muliggjør det for angriper å hente inn data fra serveren uten tillatelse. Når en ondsinnet aktør har tilgang til databasen til serveren kan vedkommende aksessere databaseinnholdet som kan gi tilgang til bruker informasjon og annen data som serveren har tilgang til. Angriper kan i tillegg modifisere og slette innhold på databasen, som kan være å endre på brukerens kontoopplysninger eller endre saldoen på brukerne. Slike endringer kan medføre til at dataen blir sabotert og videre til datatap på webapplikasjonen. Slike endringer kan utilsiktet og kan vedvare hvis det ikke oppdages. Et slikt angrep kan være truende for serveren og kan distrahere applikasjonen.

Anbefalinger:

For at et slikt angrep ikke skal påvirke webserveren. Må kunden legge inn input begrensinger som angriper kan utnytte for å direkte tilgang til databasen. For å beskytte seg

mot sql injections kan sette inputene til å ikke akseptere spesialtegnene som (* og =) i feltene. CSP-headeren bør også settes på severen, dette kan bidra med å minimere sql injections hvis de detekteres. En annen ting er at kunden krypterer innholdet på databasen som kan minimere risikoen for at dataen faller hos angriper.

Sårbarhet 5: Uautorisert Admin tilgang

Kritisk: Admin tilgang

Observasjon:

IKKE BRUK DISSE LINKENE DERSOM DU IKKE VET HVA DE ER TIL!" Misbrukes de er Å~vingen over for alle!

[Add flower](#)

[Add arrangement](#)

Housekeeping

[Clear users](#) (should clear sessions and carts after this operation)

[Clear sessions](#)

[Clear carts](#)

[Clear guestbook](#)

Vedlegg 5.1: På <http://localhost/admin/> siden får man tilgang til admin siden.

Påvirket webområde:

<http://localhost/admin/index.php>

Beskrivelse:

Ved å besøke <http://localhost/admin/index.php> siden kan angriperen aksessere admin tilgang på siden og medføre ondsinnede handlinger. Dette muliggjør det for vedkommende å blant annet modifisere innholdet på applikasjonen og slette brukere. Eksempelvis på siden: <http://localhost/admin/clearcarts.php> kan angriperen slette varene i handlekurven til brukeren.

Anbefalinger:

Tiltak for å begrense tilgang til admin siden er å ta i bruk autentisering for å aksessere siden.

HØY

Sårbarhet 6: Brute Force

Høy: Passord knekt med sqlmap

Observasjon:

```
lank> | 0.000000 | N | N | 0 | N |
| localhost | floweradmin | mysql_native_password | N | *619AE30CA1986A1B9BB352A0F92CD0AAA5DDC08D (rosesarered) | <blank>
| Y | Y | Y | Y | Y | Y | N | Y | Y | <blank> | Y | Y | Y |
| Y | Y | Y | <blank> | <blank> | Y | 0 | <blank> | Y | Y | Y | Y |
| 0 | 0.000000 | Y | Y | Y | Y | 0 | Y | N | Y | Y |
19AE30CA1986A1B9BB352A0F92CD0AAA5DDC08D (rosesarered) | Y |
```

(Vedlegg 6.1: Kommandoen (sqlmap -u http://localhost/userdetails.php?id=1 -D mysql -T user --dump) kan brukes får å knekke passordet til serveren. Her ble brukernavnet: floweradmin og passordet: rosesarered eksponert i klartekst.)

Påvirket område:

<http://localhost/userdetails.php?id=1>

<http://localhost/login.php>

Beskrivelse:

Brute Force er en teknikk angripere bruker for å knekke passord. Dette blir utnyttet for å få tilgang til bruker kontoer. Vedkommende kan benytte seg av verktøy slik som sqlmap, som fungerer ved at den prøver ulike passordforsøk på serveren og hvis den lykkes viser den brukernavnet i klartekst og passordet er enten hashet eller som i dette tilfelle er vist i klartekst (se vedlegg 6.1). Hensikten med dette angrep er å stjele data, modifisere, forstyrre

eller infisere applikasjonen med malware. For å gjøre bruteforce mer komplisert er det lurt å sette en ventetid mellom hver spørring for å minimere risikoen for bruteforce forsøk på webserveren. Det er ikke satt noe ventetid i koden til Flo AS applikasjonen, og dette bør prioriteres for å kunne øke sikkerheten i serveren mot bruteforce angrep (Se vedlegg 6.2). Det finnes mange ulike typer for bruteforce angrep, for mer informasjon henvises det til referanse.

```
if (num_rows($result)==0){  
    echo "<p class=\"content\">Invalid login\n";  
}  
else{
```

(Vedlegg 6.2: Det er ikke satt noe ventetid i koden på login.php siden som gir økt fare for at et bruteforce angrep kan skje.)

Anbefalinger:

Det anbefales at kunden legger inn en ventetid i koden for å gjøre det vanskelig for angripere å få uautorisert tilgang til serveren. Kunden bør sette en tid som ikke gjør det vanskelig for en angriper å utføre angrepet. Andre tiltak for å minke risikoen er å kryptere passordet til brukerne. Det er også laget guider for hvordan man sikrer en webserver mot bruteforce angrep. Se referansene for andre tiltak man kan ta i bruk kan sikre serveren mot bruteforce angrep.

Referanser:

- <https://www.kaspersky.com/resource-center/definitions/brute-force-attack>
- <https://sucuri.net/guides/what-is-brute-force-attack/>

Sårbarhet 7: Clickjacking sårbarhet på serveren

Høy: Clickjacking

Observasjon:

```
(Kattor@kali) [~]$ nikto -h 192.168.10.135
- Nikto v2.1.6

+ Target IP:      192.168.10.135
+ Target Hostname: 192.168.10.135
+ Target Port:    80
+ Start Time:     2022-11-06 09:22:37 (GMT-5)

+ Server: Abyss/2.16.4-X1-Linux AbyssLib/2.16.4
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
```

(Vedlegg 7.1: Nikto kommandoen observerer at det ikke er satt noen anti-Clickjacking header.)

Påvirket webområde:

Dette påvirker alle webområdene som aksesseres via hovedsiden: <http://localhost>.

Beskrivelse:

Dette kan resultere til at Futuras webapplikasjon kan mest sannsynlig bli utsatt for Clickjacking angrep, som er en ondsinnet teknikk angriperen benytter seg av for å manipulere brukerne. Det kan manipulere brukerne til å klikke på ting i applikasjonen som kan være skadelig. Årsaken til at dette angrepet kan utføres er fordi «X-Frame-Options-Headeren» ikke er satt i serveren. Denne headeren fungerer som en forsvarsmekanisme og begrenser muligheten for å laste opp upålitelige ting på webserveren som bidrar til økt sikkerheten.

Anbefalinger:

Det henvises til referanse for hvordan man kan sikre webapplikasjonen mot click-jacking angrepet.

Referanser:

<https://owasp.org/www-community/attacks/Clickjacking>

Sårbarhet 8: Dårlig håndtering av passord

Høy: Ingen passordbeskyttelse

Observasjon:

Passordet har ingen forsvarsmekanisme og bruker md5 hashen som standard algoritme og har dermed ikke et sikkert passord håndtering.

Påvirket område:

System login

Beskrivelse:

En angriper kan enkelt få tak i passordet i klartekst så lenge den ikke benytter seg av riktig krypteringsalgoritme for å kryptere den. MD5 muliggjør det for angriperen å regne seg raskt frem til hva passordet er, derfor er det viktig å salte passordet og bruke en sikker hash funksjon som gjør knekking mer komplisert for hackere. Salting av passordet vil også kunne bidra til at passordet får en ekstra variabel. Dette bidrar med å gjøre bruteforce vanskelig for angripere å knekke. Og ved bruk av riktig hash funksjoner vil dette kunne bidra til en bruteforce av passord tar lengere tid. Med disse to teknikkene vil dette kunne bidra til en optimal passordbeskyttelse, og vil minimiserer risikoen for at den faller hos angriper. Det finnes mange ulike hashing funksjoner og andre teknikker for håndtering av passord derfor henvises det til referanse.

Anbefalinger:

Anbefaler kunden å lese igjennom Owasps sine guider på hvordan man håndterer passord.

Referanser:

- [https://cheatsheetseries.owasp.org/cheatsheets/Password Storage Cheat Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html)

MEDIUM

Sårbarhet 9: OpenSSH sårbarhet

Medium: Webserveren kjører en OpenSSH versjon 8.4

p1

Observasjon:

```
floweradmin@osboxes:~$ ssh -v 192.168.10.135
OpenSSH_8.4p1 Debian-5+deb11u1, OpenSSL 1.1.1k  25 Mar 2021
debug1: Reading configuration data /etc/ssh/ssh_config
```

(Vedlegg 9: Kommandoen «ssh- v localhost» i vm-en detekterte at webapplikasjonen kjører OpenSSH versjon 8.4 p1.)

Påvirket webområde:

Berører hele webapplikasjonen

Beskrivelse:

Serveren ble oppdaget for å ha 97 sårbarheter med ulike CVE id-er. Serveren har i tillegg vært patchet en del ganger siden oktober 2022.

En av disse sårbarhetene muliggjør det for angriperen å manipulere brukeren til å koble seg til angriperens server. Dette medfører til at angriperen kjører upålitelig kode på webapplikasjonen som kan distrahere eller sabotere systemet. Se referansene for å få mer informasjon om hvilke sårbarheter som befinner seg i serveren.

Anbefalinger:

Anbefaler å oppdatere OpenSSH til nyeste versjon.

Referanser:

- CVE-2021-28041
- <https://www.cybersecurity-help.cz/vdb/SB2021031404>
- https://www.cvedetails.com/vulnerability-list/vendor_id-97/product_id-585/Openbsd-Openssh.html

LAV

Sårbarhet 10: HTTPOnly og secure Flag er ikke definert

Lav: Udefinerte Cookie parametere

Observasjon:

Webserver har ingen HTTP-flagg som er definert og kan leses eksternt

▼ Cache Storage	Filter Items										+ ↺
http://192.168.10.135											
▼ Cookies											
http://192.168.10.135	Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed	
	flowershop_sessi...	5	192.168.10.135	/	Tue, 15 Nov 2022 11:24:13 ...	19	false	false	None	Tue, 08 Nov 2022 11:25:24 ...	

(Vedlegg 10.1: HTTPOnly flagget er ikke satt i cookien og sesjonsid-en kan leses eksternt.)

Påvirket webområde:

Dette påvirker alle webområdene som aksesseres via hovedsiden: <http://localhost>

Beskrivelse:

HTTPOnly flagget er satt til «false», dette kan resultere til at angriper kan aksessere uautorisert data på applikasjonen ved å lese av cookies fra brukere. HttpOnly er en kode i Cookien som er designet for å forhindre hackeren fra å kjøre XSS-skript på server får å aksessere data, dette bidrar med å gjøre cookien sikrere. HTTPOnly er også designet for å forsvare applikasjonen mot sesjons-hijacking, dette vil kunne bidra med å forhindre en angriper fra å ta over sesjonen.

Det ble observert at secure flagget heller ikke satt til «true», det gjør at cookien sendes via vises i klart tekst via en ukryptert tilkobling til brukeren. Dette kan medføre at en angriper kan overvåke nettverkstrafikken til brukeren. Hvis denne er satt til «true», vil cookien sendes kryptert via HTTPS som forhindrer all avlytting og at bruker cookien ikke falles hos angriper.

Anbefalinger:

Kunden bør sette HTTPOnly og secure flagget til «true» i koden som kan bidra med å sikre brukerne på webapplikasjonen mot nettverksavlytting. Det er også viktig å oppgradere port 80 til port 443 som bidrar med å sende cookien i en kryptert forbindelse med bruker (Se Info 3). Dette bidrar til økt sikkerhet på applikasjonen og forhindrer en angriper fra å få tak i cookien og utføre sesjons hijacking angrep, for mer informasjon henvises det til referanse.


Referanser:

- <https://www.eccouncil.org/cybersecurity-exchange/ethical-hacking/how-to-prevent-session-hijacking-attacks/>
- <https://www.speedguide.net/port.php?port=22#:~:text=An%20unauthenticated%20remote%20attacker%20with,access%20to%20the%20vulnerable%20application.>

INFORMASJON

Info 1: Informasjon om systemet

Info: Systeminformasjon lekket

PHP Version 5.4.45	
	
System	Linux osboxes 5.10.0-8-686-pae #1 SMP Debian 5.10.46-4 (2021-08-03) i686
Build Date	Oct 27 2022 14:30:49
Configure Command	'./configure' '--enable-ftp' '--without-openssl' '--disable-libxml' '--disable-dom' '--disable-simplexml' '--disable-xml' '--disable-xmlreader' '--disable-xmlwriter' '--without-pear' '--with-mysql'
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/lib
Loaded Configuration File	(none)
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API220100525,NTS
PHP Extension Build	API20100525,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled

(Vedlegg 11.1: Informasjon om PHP inklusiv systeminfo ble lekket på phpinfo.php siden «<http://localhost/phpinfo.php>»)

Info 2: Server eksponert

Info: Informasjon om Abyss serveren ble eksponert

```
(kali㉿kali)-[~]
└─$ nikto -h 192.168.10.135
+ Nikto v2.1.6

+ Target IP:          192.168.10.135
+ Target Hostname:    192.168.10.135
+ Target Port:        80
+ Start Time:         2022-11-07 11:11:54 (GMT-5)

+ Server: Abyss/2.16.4-X1-Linux AbyssLib/2.16.4
```

(Vedlegg 12: Nikto kommandoen på kali detekterte at webserveren kjører Abyss /2.16.4 -X1-linux)

Info 3: Nmap eksponerer åpen port

Info: Port 80 er åpen.

Verktøyet Nmap på kali detekterer at oppdaget at webapplikasjonen er åpen på port 80 og bruker en HTTP-protokoll. Port 80 bruker HTTP – protokollen som skaper en usikker forbindelse mellom nettleseren og serveren. Anbefales at kunden bytter til port 443 som benytter seg av HTTPS protokollen som krypterer innholdet på siden, dette vil kunne gi økt sikkerhet på applikasjonen og minimerer risikoen for at sensitiv data ikke falles hos vedkommende i transitt.

Info 4: Nmap eksponerer åpen port

Info: Port 22 er åpen.

Nmap på kali detekterer at serveren har en åpen port 22 som eksponerer SSH protokollen. Hvis en angriper får tak i IP adressen til serveren kan hackeren ved å bruteforce og eller få tak i lekkede SSH nøkler, få uautorisert tilgang til serveren trådløst fra sin egen maskin. Hvis angriper vellyktes med å bryte seg inn på serveren som root bruker kan vedkommende få full kontroll over systemet og utføre ondsinnede handlinger som kan berøre applikasjonen. For å forhindre at en angriper får ssh tilgang anbefales det at kunden går igjennom referansene som ble henvist.

Referanser:

- <https://securitytrails.com/blog/mitigating-ssh-based-attacks-top-15-best-security-practices>

Info 5: GNOME versjon eksponert

Info: Systeminnstillinger

GNOME Version	3.38.5
---------------	--------

(Vedlegg 13: Serveren eksponerer GNOME versjon i klartekst under system settings -> about)

Info 6: Webserver tilgang

Info: Register siden

Anderson

4th Street

☒ VISA ☐ MasterCard ☐ AMERICAN EXPRESS

4111111111111112

1 / 2004

anderson

.....

.....

Create Account

(Vedlegg 14: En angriper kan ved å besøke <http://localhost/register.php>, registrere en bruker.)

Here are your user details.

Name

Address

Card Type ☒ VISA ☐ MasterCard ☐ AMERICAN EXPRESS

Card Number

Expiry Date /

Login

Password

(Vedlegg 15: Etter registreringen står userdetails.php siden tilgjengelig, som er sårbar for sql injection.)

Angriper kan ved å besøke <http://localhost/register.php> lage en falsk bruker som gir vedkommende tilgang til serveren. Når registreringen er konfigurert kan bruker åpne <http://localhost/userdetails.php> siden (Se vedlegg 15) for å se brukerdetaljer. En hacker kan utnytte denne siden ved å utføre et sql injection angrep på severen for å få uautorisert datatilgang til applikasjonens database. Henviser til (Sårbarhet 6) dersom kunden ønsker å få mer informasjon om angrepet.

Info 7: Hemmelig bufferoverflow eksponert på serveren

Info: Bufferoverflow



```
1 <html>
2 <body>
3 <p>Congratulations!!!
4 <p>You've just found the hidden buffer overflow!
5 <p>
6 <p>Of course, buffer overflows in PHP are a lot harder to find than this but you've
  got the idea!
7 </body>
8 </html>
```

(Vedlegg 16: Serveren ble oppdaget for å ha en hemmelig bufferoverflow under filplasseringen:
«/home/floweradmin/apps/abyssws/htdocs»)

Buffer overflow er en feil i programvare koden eller en sårbarhet en hacker utnytter får å få uautorisert tilgang til et system. En buffer er en del av systemets minnet som benyttes til datalagring. Årsakene til at det oppstår en overflow er fordi et program eller en prosess på maskinen leser og skriver inn mer data en det bufferen har kapasitet til. Deretter blir minnet overskrevet og dataen blir lagret i en ny plassering på systemet. Et slikt angrep kan berøre et system eller en server ved å distrahere eller sabotere et program. Overflow kan i tillegg medføre til at et program krasjer. Buffer overflow assosieres ofte med programmeringsspråk som C eller C++.