

# ***Rapport***

## ***Teknisk sikkerhetsrevisjon***

### Executive Summary

Denne testen er utført av Høyskolen Kristiania på vegne av Osboxes.org. Hele webapplikasjonen er sikkerhetsrevidert. Det ble utført en white-box test av webserveren og målet med denne testen er å identifisere og utnytte sårbarhetene i applikasjonen for å ta en sikkerhetsvurdering som kan bidra med å sikre applikasjonen mot fremtidige hacker angrep. Det ble med hensyn til Owasp Testing Guide, brukt pålitelige verktøy under testingen.

Webapplikasjonen har generelt sett et medium nivå av sikkerhet og Osboxes 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, ukryptert innhold 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
Trussel modell (Tabell over sårbarheter) .....	3
Gjennomførelse .....	3
Kontraktsregler .....	3
Deltagere .....	3
IP adresser brukt under testingen .....	4
Detaljert beskrivelse av funn .....	4
Sårbarhet 1 .....	4
Sårbarhet 2 .....	6
Sårbarhet 3 .....	6
Sårbarhet 4 .....	7
Sårbarhet 5 .....	8
Sårbarhet 6 .....	9
Sårbarhet 7 .....	10
Sårbarhet 8 .....	10
Informasjon 1 .....	11
Informasjon 2 .....	12
Bibliografi .....	14

## Trussel modell (Tabell over sårbarheter)

Kategori	Sårbarhet	Sårbarhetsrisiko
<b>Sårbarhet 1</b>	Serveren bruker HTTP som standard protokoll	Kritisk
<b>Sårbarhet 2</b>	Web applikasjonen kjører serveren Apache versjon 2.4.38	Høy
<b>Sårbarhet 3</b>	CSP headeren ikke funnet	Medium
<b>Sårbarhet 4</b>	XSS headeren mangler	Medium
<b>Sårbarhet 5</b>	CSFR tokens lekket i HTML form	Medium
<b>Sårbarhet 6</b>	Mangel på Clickjacking header	Medium
<b>Sårbarhet 7</b>	Serveren kjører OpenSSH 7.9	Medium
<b>Sårbarhet 8</b>	Cookie HttpOnly ikke funnet	Lav
<b>Informasjon 1</b>	Ansattes bruker informasjon funnet	Info
<b>Informasjon 2</b>	Session Id-er lekket	Info

## Gjennomførelse

Testen retter seg mot følgende inngangspunkt:

- IP:192.168.10.134

Tester har fått tilgang til følgende brukere/kontoer:

- VW maskin user: osboxes :password: osboxes.org

## Kontraksregler

Kunden har sammen med Høyskolen Kristiania inngått en avtale. Avtalen er signert med hensyn til SECURITY ASSESSMENT AGREEMENT. Oppdraget er Datert 11.09.22 og dekker perioden 21.09.22 til 07.10.22.

## Deltagere

Disse testene ble gjennomført av Usman Ahmad, Cyber Security student hos Høyskolen Kristiania.

## IP adresser brukt under testingen

Følgende IP-adressene ble brukt under testingen av webapplikasjonen:

- 192.168.10.138 (hentet fra VW maskinen)

## Detaljert beskrivelse av funn

### Sårbarhet 1

```
$ nmap -p 1-65535 -T4 -v 192.168.10.134
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-21 07:18 EDT
Initiating Ping Scan at 07:18
Scanning 192.168.10.134 [2 ports]
Completed Ping Scan at 07:18, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 07:18
Completed Parallel DNS resolution of 1 host. at 07:18, 6.63s elapsed
Initiating Connect Scan at 07:18
Scanning 192.168.10.134 [65535 ports]
Discovered open port 80/tcp on 192.168.10.134
Connect Scan Timing: About 23.87% done; ETC: 07:21 (0:01:39 remaining)
Connect Scan Timing: About 59.67% done; ETC: 07:20 (0:00:41 remaining)
Completed Connect Scan at 07:20, 87.94s elapsed (65535 total ports)
Nmap scan report for 192.168.10.134
Host is up (0.0013s latency).
Not shown: 65534 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 94.67 seconds
```

NMAP scannet webapplikasjonen (se vedlegg 4.1) og oppdaget at webapplikasjonen er åpen på port 80 og bruker en HTTP-protokoll på IP-adressen: 192.168.10.134.

Port 80 bruker HTTP – protokollen som skaper en usikker forbindelse mellom nettleseren og serveren. (purevpn, u.d.) Med tanke på at Futuras webapplikasjon bruker HTTP-protokollen anses dette som en kritisk sårbarhet og kan i verstefall medføre til at angriperen kan kjøre en sårbarhetsanalyse og hente inn uautorisert data fra webapplikasjonen. Da er det snakk om epost adresser, bosted, brukernavn, passord, og andre personopplysninger.

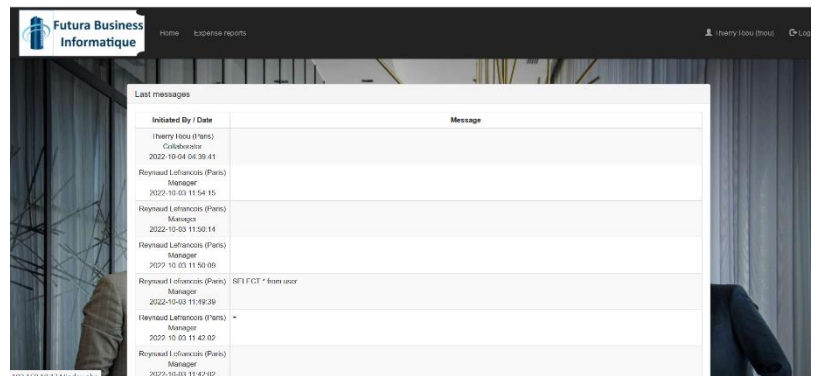


```

return;

if(!($stmt = $GLOBALS['mysqli_ston']->prepare("INSERT INTO user VALUES
(1, 'afoulon', '', md5('wq6hblv3') . '', 'Financial approver', 'Foulon',
'Aristide', 1, 'afoulon@futuraBI.fr', 1, NOW(), 1),
(2, 'pbaudouin', '', md5('Hackle') . '', 'Financial approver', 'Baudouin',
'Paul', 1, 'pbaudouin@futuraBI.fr', 2, NOW(), 1),
(3, 'rlefrancois', '', md5('m4ukhkjq') . '', 'Manager', 'Lefrancois', 'Reynaud',
'rlefrancois@futuraBI.fr', 1, NOW(), 1),
(4, 'mriviere', '', md5('6v7j4tj2') . '', 'Manager', 'Riviere', 'Manon', 2,
'mriviere@futuraBI.fr', 2, NOW(), 1),
(5, 'mnguyen', '', md5('27sfbdij') . '', 'Manager', 'Nguyen', 'Maximilien', 3,
'mnguyen@futuraBI.fr', 2, NOW(), 1),
(6, 'pgervais', '', md5('b98e67ys') . '', 'Collaborator', 'Gervais', 'Placide',
'pgervais@futuraBI.fr', 3, NOW(), 1),
(7, 'placombe', '', md5('qayhned2') . '', 'Collaborator', 'Lacombe',
'Philibert', 1, 'placombe@futuraBI.fr', 3, NOW(), 1),
(8, 'tridou', '', md5('h1rlnifq') . '', 'Collaborator', 'Riou', 'Thierry', 1,
'tridou@futuraBI.fr', 3, NOW(), 1),
(9, 'broy', '', md5('37f52u21') . '', 'Collaborator', 'Roy', 'Baudouin', 1,
'broy@futuraBI.fr', 3, NOW(), 1),
(10, 'brenaud', '', md5('u99hau4d') . '', 'Collaborator', 'Renaud',
'Bernadette', 2, 'brenaud@irtechnologies.fr', 4, NOW(), 1),
(11, 'slamotte', '', md5('fzghn4lw') . '', 'Collaborator', 'Lamotte', 'Samuel',
'slamotte@futuraBI.fr', 4, NOW(), 0),
(12, 'nthomas', '', md5('en3dtdjy') . '', 'Collaborator', 'Thomas', 'Ninette',
'nthomas@futuraBI.fr', 6, NOW(), 3)

```



Slik ser hovedsiden ut etter innlogging

Det ble lekket brukernavn og passord på config.inc.php filen i VM maskinen (se vedlegg 5.1).

Årsaken til det er at innholdet i filen ligger ukryptert, og informasjonen står dermed i klartekst som øker faren for at den kan leses i transitt. Hvis angriper får fysisk tilgang til hardwaren, kan en hacker gå inn i VM-en og finne disse bruker opplysningene. Og deretter utnytte dette for å aksessere webserveren (se vedlegg 5.2) og utføre ondsinnede handlinger på siden. Angriperen kan for eksempel legge inn XSS skript som kjører 80 alerts på siden, som kan være ufattelig distraherende for brukerne som prøver å utføre handlingene sine på applikasjonen. Angriper vil også være i stand til utføre andre angrep som DDOS. Dette angrepet forstyrrer offerets nettverkstrafikk og utelukker brukeren fra serveren.

Angriperen kan i tillegg få full kontroll over systemet som kan medføre til at vedkommende får tilgang til sensitiv data som kan misbrukes til phishing angrep (Se Informasjon 1). Dette gir en konsekvens for både webapplikasjonen og for brukeren som er utsatt for hackerangrepet. Så det anbefales å oppgradere port nummeret i HTTP protokollen til port 443 HTTPS slik at innholdet krypteres på siden og at sensitiv data ikke falles hos vedkommende i transitt.

## Sårbarhet 2

```
$ nikto -h 192.168.10.134
- Nikto v2.1.6

+ Target IP:      192.168.10.134
+ Target Hostname: 192.168.10.134
+ Target Port:    80
+ Start Time:     2022-09-21 06:43:27 (GMT-4)

+ Server: Apache/2.4.38 (Debian)
```

Serveren ble scannet med nikto kommandoen på kali (Se vedlegg 6.1) som oppdaget at denne serveren kjører 2.4.38 versjonen av Apache serveren som består av en lang rekke sårbarheter. Serveren har siden 2014 og frem til nå vært patchet en god del ganger gjennom disse årene. Sårbarhetene som befinner seg i 2.4.38 serveren er CVE-2022-26377, CVE-2021-44790, CVE-2021-44224, CVE2021-34798, CVE-2021-36160, CVE-2021-39275 og mange flere (Cybersecurityhelpz, 2022). De aller fleste sårbarhetene har en skala som er høyere enn 7 og dette tilsvarer en total sårbarhetsrisiko som er høy. (Vulmon.com, 2019)

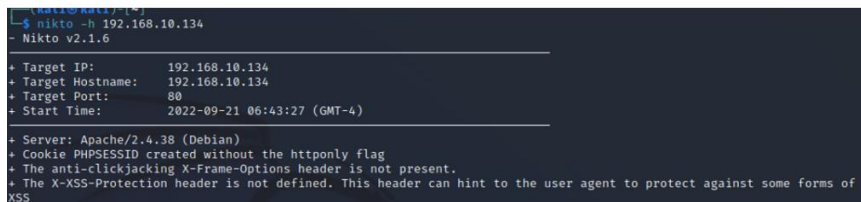
## Sårbarhet 3

Content Security Policy (CSP) Header Not Set	
URL:	http://detectportal.firefox.com/canonical.html
Risk:	🔴 Medium
Confidence:	High
Parameter:	
Attack:	
Evidence:	
CWE ID:	693
WASC ID:	15
Source:	Passive (10038 - Content Security Policy (CSP) Header Not Set)

Owasp Zap fant ut av at serveren mangler CSP (Content Security Policy) headeren (Se vedlegg 6.2). CSP bidrar med å detektere og redusere angrep som angriperen utfører som blant annet XSS angrep og SQL injections angrep. (Crashtest-Security, 2022) Siden serveren mangler CSP headeren, vil dette resultere til at angriperen kan manipulere webapplikasjonen til å kjøre skadelig kode på siden. Serveren tror at koden som kjøres er pålitelig og kjører derfor den skadelige koden med den originalen koden på siden.

For Futura Business Informatique er dette en stor ulempe, om angriperen aksesserer og manipulerer webserveren kan vedkommende kjøre kode som utfører alle mulige angrep mot applikasjonen, det gjelder alt fra XSS angrep til malware angrep på siden. Det kan i tillegg være kode som henter inn sensitiv informasjon om brukerne, kjøre kode som kopierer Cookie fremdriftene til de ulike brukerne som videre overføres til angriperens maskin. Disse opplysningene kan misbrukes til å spre malware til andre brukere ved bruk av phishing metoder eller så kan angriper bruke opplysningene til å lage falske kontoer på sosiale medier. Med tanke på at webserveren ikke klarer å skille mellom ekte og falske koder vil slike skadelige koder kjøre på webserveren. Dette kan medføre blant annet til en sabotasje av systemet.

#### Sårbarhet 4

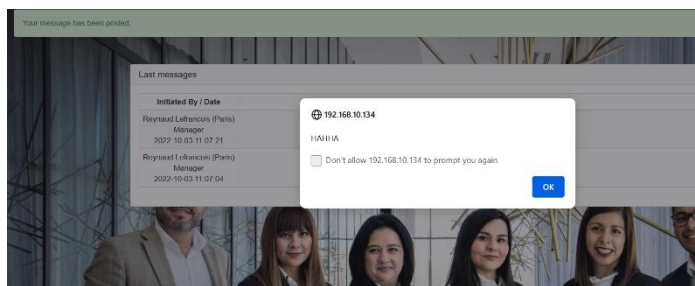


```
(kali@kali) ~$ nikto -h 192.168.10.134
- Nikto v2.1.6

+ Target IP:      192.168.10.134
+ Target Hostname: 192.168.10.134
+ Target Port:    80
+ Start Time:     2022-09-21 06:43:27 (GMT-4)

+ Server: Apache/2.4.38 (Debian)
+ Cookie PHPSESSID created without the httponly flag
+ 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
```

Nikto sikkerhetsanalysen på Linux oppdaget at XSS protection headeren mangler (Se vedlegg 7.1). Denne headeren er viktig mot beskyttelse av XSS (Cross Site Scripting). Denne bidrar med å detektere XSS skript som blir kjørt på webserveren og har som ansvar å blokkere nettsiden og eliminere koden så snart den er oppdaget. Dette forhindrer angriperen fra å utføre et XSS angrep på nettapplikasjonen. (Geeks for Geek, 2022)



Med tanke på at denne headeren ikke befinner seg i Futuras web applikasjon, er det stor sannsynlighet for at den utsettes for XSS angrep. Dette kan medføre til at XSS kode detekteres på nettsiden når den kjører. Dette gjør at angriperen kan kjøre malware kode, stjele sensitiv data til brukerne på nettsiden til Futura. Eller så kan angriperen skrive en kode som kjører pop-up-ads på siden for alle brukere, uansett hvor mange ganger man refresher siden.

Eksempelvis, hvis brukeren går inn på en ukryptert nettside som Futuras webapplikasjon kan brukeren kjøre en XSS kode i inputfeltet på siden som kan medføre til at hackeren kjører for eksempel en alert kode i meldingsfeltet (Se vedlegg 8.1) inne på siden.

Hackeren kan også injisere skadelig kode som legger inn en lenke på siden og deretter kjører når brukeren har trukket på den, dette kan være kode som i verste fall forstyrrer eller krasjer serveren. En annen ting angriperen kan gjøre er å kjøre kode som legger inn en lenke på applikasjonen som gjør at offeret utsettes for å laste ned malware på maskinen ubevist.

## Sårbarhet 5

I sikkerhetsanalysen ble det detektert at webserver ikke hadde Anti-CSRF tokens.

Dette betyr at serveren er sårbar mot CSRF angrep. Cross Site Forgery Request er en angreps metode som bruker social engineering metoder for å manipulere brukeren til å trykke på en link, og deretter sender forfalskede forespørsler mot en server. (Cross Site Forgery Attack, 2022)



Med tanke på at brukeren er autentisert, klarer ikke serveren å skille mellom pålitelige og forfalskede forespørsler. Dette kan manipulere brukeren til å utføre ondsinnet handlinger som for eksempel foreta endringer på epostadressen og passordet sitt utilsiktet eller utføre en pengeoverføre til angriperen. Angriper kan i verstefall få full kontroll over kontoen og sende eposter til de andre ansatte på vegne av brukeren utilsiktet. Eller for eksempel hvis angriperen får kontroll over kontoene til en ansatt på sosiale medier, kan de poste stygge meldinger på kontoen som får andre til å tro at meldingene ble sendt av brukeren selv.

Eksempelvis, hvis du som bruker åpner en epost som utgjør seg å være fra Futura. Og du får beskjed om at du har vunnet en bolig, men for å motta pengene må du klikke på lenken. Videre tar deg den med til Futuras webapplikasjon som krever at du oppgir e-post og passordet ditt for å motta premien. Istedenfor at du mottar premien, modifierer du epostadressen din og passordet utilsiktet på webserveren.

Anti CSFR tokens skal forhindre CSFR angrep mot serveren slik at brukeren ikke blir utsatt for manipulasjon som bidrar med å sikre brukeren mot identitetstyveri eller å utføre handlinger utilsiktet på webapplikasjonen. (Dizdar, 2021)

## Sårbarhet 6

Nikto kommandoen på kali oppdaget at webapplikasjonen mangler en anti- clickjack-header. Dette kan resultere til at Futuras webapplikasjon kan mest sannsynlig bli utsatt for click-jacking angrep, som er en onsinnet teknikk angriperen benytter seg av for å manipulere brukerne. Det kan manipulere brukerne til å utføre handlinger de egentlig ikke skal utføre.

For eksempel at brukeren oppgir informasjon på inputfeltene på signup siden i applikasjonen , kunne ha vært et eksempel på hvor angriperen kunne ha utført et slikt angrep. Så istedenfor at brukeren opprettes , så sendes opplysningene direkte til angriperen.

I slike angrep kan hackeren også få brukeren til å utføre pengeoverføring utilsiktet ved trykke på knapp som <<Vinn>> , men som egentlig opptrer til å være en overføringsknapp. Login siden på Futuras webserver kunne også være et eksempel på dette , der brukernavnet og passordet faller hos angriperen utilsiktet ved å trykke på << Login knappen>> som

egentlig opptres til å være en <<Send bruker navn og passord knapp>>. Som videre medfører til at angriper misbruker dette til å logge seg inn på brukeren og utføre handlinger på vegne av brukeren, som gir konsekvenser for både brukeren og bedriften. Med anti-clickjacking-headeren hadde dette vært uungåelig.

## Sårbarhet 7

```
osboxes@osboxes:~$ ssh -v 192.168.10.134
OpenSSH 7.9p1 Debian-10, OpenSSL 1.1.1c 28 May 2019
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: /etc/ssh/ssh_config line 19: Applying options for *
debug1: Connecting to 192.168.10.134 [192.168.10.134] port 22.
debug1: connect to address 192.168.10.134 port 22: Connection refused
ssh: connect to host 192.168.10.134 port 22: Connection refused
osboxes@osboxes:~$
```

Osboxes serveren kjører en OpenSSH versjon 7.9p1 (Se vedlegg 10.1). Ser at serveren ble oppdaget for å ha 5 sårbarheter med ulike CVE id-er. Serveren har i tillegg vært patchet en del ganger siden oktober 2019. (cybersecurityhelpz, 2019)

En av disse sårbarhetene muliggjør det for angriperen å eskalere privilegere på webapplikasjonen. Med andre ord vil dette si at angripere få uautorisert tilgang til systemet, som å misbruke rettighetene en bruker har til å utføre ondsinnede handlinger på webapplikasjonen som å utføre XSS angrep, CSRF angrep osv.

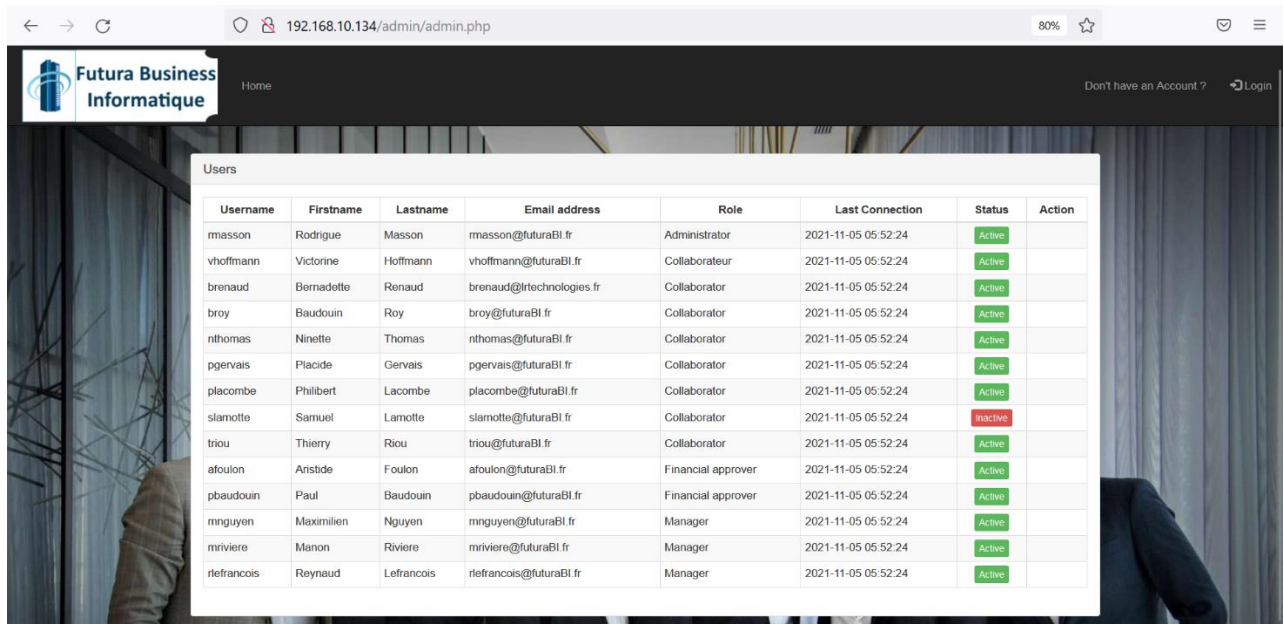
## Sårbarhet 8

```
Set-Cookie: PHPSESSID=r46mc3gu8l8f1cdsv0mftkhhg3; path=/
```

I syntaksen er ikke Cookie satt til HTTPOnly (Se vedlegg 10.2), dette kan medføre til at en angriper får tilgang til uautoriserte data på applikasjonen. HttpOnly er en kode i Cookien som er designet for å forhindre hackeren fra å kjøre skript (I form av et XSS angrep) for å aksessere data, dette bidrar med å gjøre cookien sikrere. Dersom syntaksen bestod av et HttpOnly tag ville ikke angripere være i stand til å kjøre et XSS angrep mot Applikasjonen. Dette forhindrer hackeren til å ikke kjøre et skript som kan for eksempel overfører cookie fremdriftene fra serveren direkte til nettsiden sin. (Nidecki, 2020). Webserveren ser ut til å bestå av en sesjons cookie. Hvis angriperen får tak i denne, kan personen utføre en session hijacking på webapplikasjonen. Dette vil medføre til at vedkommende kan overta sesjonen

til brukeren ved innlogging og gjøre ting på applikasjonen uten å ha autorisasjon til det fra brukeren (Se informasjon 2).

### Informasjon 1



Username	Firstname	Lastname	Email address	Role	Last Connection	Status	Action
masson	Rodrigue	Masson	rmasson@futuraBI.fr	Administrator	2021-11-05 05:52:24	Active	
vhoffmann	Victorine	Hoffmann	vhoffmann@futuraBI.fr	Collaborateur	2021-11-05 05:52:24	Active	
brenaud	Bernadette	Renaud	brenaud@irtechnologies.fr	Collaborator	2021-11-05 05:52:24	Active	
broy	Baudouin	Roy	broy@futuraBI.fr	Collaborator	2021-11-05 05:52:24	Active	
nthomas	Ninette	Thomas	nthomas@futuraBI.fr	Collaborator	2021-11-05 05:52:24	Active	
pgervais	Placide	Gervais	pgervais@futuraBI.fr	Collaborator	2021-11-05 05:52:24	Active	
placombe	Philibert	Lacombe	placombe@futuraBI.fr	Collaborator	2021-11-05 05:52:24	Active	
slamotte	Samuel	Lamotte	slamotte@futuraBI.fr	Collaborator	2021-11-05 05:52:24	Inactive	
triu	Thierry	Riou	triu@futuraBI.fr	Collaborator	2021-11-05 05:52:24	Active	
afoulon	Aristide	Foulon	afoulon@futuraBI.fr	Financial approver	2021-11-05 05:52:24	Active	
pbaudouin	Paul	Baudouin	pbaudouin@futuraBI.fr	Financial approver	2021-11-05 05:52:24	Active	
mnguyen	Maximilien	Nguyen	mnguyen@futuraBI.fr	Manager	2021-11-05 05:52:24	Active	
mriviere	Manon	Riviere	mriviere@futuraBI.fr	Manager	2021-11-05 05:52:24	Active	
rlefrancois	Reynaud	Lefrancois	rlefrancois@futuraBI.fr	Manager	2021-11-05 05:52:24	Active	

Hvis man søker etter <http://192.168.10.134/admin/admin.php> får man opp en side med en tabell som inneholder informasjon om de ulike ansatte i Futura. Her ble det lekket brukernavn, epost adresse, fullt navn, kontoaktivitet og hvilke stillinger de har. Her er konfidensialiteten på det svakest og hvis angripere får tak i disse opplysningene, kan de utføre ondsinnede handlinger som å utføre Social engineering eller phishing angrep. (imperva.com, 2022)

Testeren benyttet seg av OWASP ZAP for å foreta en sikkerhetsanalyse i webapplikasjonen for finne personopplysningene om Futura Business ansatte (Se vedlegg 11.1). Her ble det detektert at alle eposter slutter på (@futuraBI.fr). En annen viktig detalj er at de fleste lekkede epostene består av ansattes brukernavn. De fleste ansatte er linket til LinkedIn, som er en nettside med informasjon om ulike ansatte og hvem de er knyttet til.

En angriper kan bare ved å søke opp disse epostadressene på søkemotorer som google og få opp en haug med informasjon. Det kan være for eksempel informasjon om bosted og telefonnummer, som videre kan medføre at angriperen foretar phishing angrep, ikke bare

mot offeret og de andre ansatte på Futura. Men i verstefall kan disse phishing angrepene spres til de andre ansatte på LinkedIn, sakte men sikker ville dette resultere til å spres til hele internett.

Phishing angrep er en metode angripere eller svindlere bruker for å manipulere brukere til å oppgi sensitiv informasjon om dem selv via SMS, telefonsamtaler eller epost. Det kan være at ansatte for eksempel blir manipulert til å oppgi brukernavnet og passordet til Futuras webapplikasjon eller sosiale medier som Facebook og Twitter. Eller så kan ansatte manipuleres til å oppgi kontoopplysningene sine. På denne måten kan angriperen få uautorisert tilgang til ansattes kontoer i webapplikasjonen eller sosiale medier og utføre ondsinnede handlinger som kan gi store konsekvenser for både bedriften selv og de ansatte. Dette kan blant annet føre til at de ansatte som blir utsatt for phishing angrep, mister rettighetene til å bruke epost igjen.

Angriperen kan i tillegg utnytte kontoopplysningene til å overføre pengene til seg selv. Social engineering metodene kan også brukes til å spre skadevare til alle ansatte på bedriften som for eksempel løsepengevirus.

Siden angriperen har som hovedformål å tjene penger, kan de misbruke ansattes opplysninger for å true bedriften med å betale en viss pengesum. Det kan være for eksempel at de truer Futura Business med å betale 50 millioner kr hvis de ikke ønsker at personopplysningene skal lekkes offentlig på nett.

Derfor oppfordres det sterkt til at alle ansatte i Futura, får opplæring i hvordan man skal unngå slike svindelforsøk. På denne måten vil ansatte trenes opp til å beskytte seg selv mot phishing metoder som kan være skadelig. Dette vil kunne medføre til at informasjonssikkerheten styrkes hos ansatte som bidrar med at ansatte får en forståelse av hvilke konsekvenser social engineerings metoder kan gi, og dermed skjønner hvor viktig det er å unngå slike svindelforsøk. Dette vil bidra med at bedriften klarer å opprettholde konfidensialiteten til bedriften slik at sannsynligheten for at sensitiv data faller hos svindleren minimeres.

## Informasjon 2

```
Cookie: PHPSESSID=s90dfahvgu86558d2Fh1ut1k0e  
Upgrade-Insecure-Requests: 1  
username=admin&password=test&csrf_token=835fc8ecb104500a23d06cb6ffe99174&login=login
```

Sesjons-ID er ble lekket etter å ha kjørt sårbarhetsanalyse verktøyet OWASP ZAP. En sesjon starter fra brukeren er innlogget i en tjeneste som i dette tilfelle blir Futuras webapplikasjon. Hver bruker får utdelt et unikt sesjons-id fra websiden eller webserveren. (Techtarget, 2006) Denne id-en lagres sammen med brukernavnet som har fått koden tildelt. Hvis brukeren lukker webapplikasjonen uten å logge seg ut, vil serveren sende koden til webapplikasjonen neste gang brukeren åpner den. Dette vil kunne bidra til at brukeren kan utføre handlinger på siden uten å være innlogget. Dette kan være en fordel for brukeren uten å ha kjennskap til hva konsekvensene kan være.

For eksempel en bruker er på en nettbutikk og legger inn noen varer i handlekurven og lukker siden, vil nettsiden automatisk generere en sesjons-id til akkurat denne sesjonen til tilhørende bruker som lagres i cookien på webserveren. Når brukeren åpner websiden neste gang, vil varene ligge der slik at bruker slipper å foreta den samme handlingen på nytt

Dessverre er det sånn at sesjons – ider er svært etterlengtet får hackere. Hvis sesjons id-ene til et av de ansatte faller hos angriperen, kan dette medføre til at vedkommende kan få tilgang til sesjonen uten å trenge brukerens brukernavn og passord. Dermed kan angriperen opptre som brukeren i systemet og utføre uønskede handlinger på siden som å sende phishing meldinger og malware til andre brukere osv. I noen tilfeller kan hackeren spore Futuras ansatte hvis han finner siden som lagrer brukeraktiviteten på for eksempel en Facebook side, dette kan misbrukes ved at hackeren overvåker brukeren og finner opplysninger om hvor de bor og hvem de er relatert med osv.

Dermed er det viktig for webserveren å oppgradere webserveren til HTTPS for å kunne sikre applikasjonen mot sesjons id tyveri, slik at angriperen ikke får tak i sesjons id og foretar endringer i ansattes konto utilsiktet. En annen løsning er hvis ansatte benytter seg av VPN, dette bidrar med å holde ansatte anonym på nettet. Dette gjør at angriper ikke tar kontroll over sesjonene til ansatte og personopplysningene forblir trygge som bidrar å øke sikkerheten når de ansatte benytter seg av webserveren.

## Bibliografi

*Cross Site Forgery Attack*. (2022). Hentet fra Imperva.com:

<https://www.imperva.com/learn/application-security/csrf-cross-site-request-forgery/>

*Crashtest-Security*. (2022, 04 03). Hentet fra Content-Security -Policy - Header: <https://crashtest-security.com/content-security-policy-header/>

*cybersecurityhelpz*. (2019, 10 09). Hentet fra openssh 7.9p1: <https://www.cybersecurity-help.cz/vdb/openssh/openssh/7.9p1/>

*Cybersecurityhelpz*. (2022, Juni 08). *www.cybersecurity-help.cz*. Hentet fra Cybersecurityhelpz: <https://www.cybersecurity-help.cz/vdb/SB202109131>

Dizdar, A. (2021, Juni 11). *brightsec*. Hentet fra Csrftokens: <https://brightsec.com/blog/csrf-token/>

*Geeks for Geek*. (2022, Januar 10). Hentet fra HTTP headers X-XSS-Protection: <https://www.geeksforgeeks.org/http-headers-x-xss-protection/>

*imperva.com*. (2022). Hentet fra phishing attack scam: <https://www.imperva.com/learn/application-security/phishing-attack-scam/>

Nidecki, T. A. (2020, 08 24). *www.acunetix.com*. Hentet fra HttpOnly -flag protecting cookies: <https://www.acunetix.com/blog/web-security-zone/httponly-flag-protecting-cookies/>

*purevpn*. (u.d.). Hentet fra http-vulnerability: <https://www.purevpn.com/ddos/http-vulnerability>

*Techtarget*. (2006, 01). Hentet fra Session -ID: <https://www.techtarget.com/searchsoftwarequality/definition/session-ID>

*Vulmon.com*. (2019, 09 26). *Vulmon.com*. Hentet fra Vulmon: <https://vulmon.com/searchpage?q=apache+http+server+2.4.38&sortby=byriskscore>