

Eksamen CYB2100 - november 2024

1.a

Statisk analyse undersøker skadevaren uten å faktisk kjøre den, ved å analysere kode og struktur for å forstå funksjonaliteten og oppbyggingen. Denne metoden anses som trygg ettersom skadevaren ikke kan spre seg. Statisk analyse utføres i et isolert miljø, hvor man kan analysere skadevaren mer detaljert. Samtidig krever denne type analyse høy teknisk kompetanse og erfaring. Statisk analyse fungerer ofte som et første steg i en skadevareanalyse, da den gir en overordnet forståelse som kan legge grunnlaget for videre dynamisk analyse. Ved å kombinere disse metodene oppnås en mer helhetlig forståelse av skadevaren.

Dynamisk analyse innebærer å kjøre skadevaren i et kontrollert miljø, som en sandkasse eller en virtuell maskin, for å observere dens atferd i sanntid (Aqua Security, 2023). Dynamisk analyse medfører en viss risiko, da skadevaren kan bryte ut av det kontrollerte miljøet og potensielt forårsake skade. Sammenlignet med statisk analyse er denne metoden mer tid- og ressurskrevende, men den gir verdifull informasjon som kan avdekke mer sofistikerte trusler. Kombinasjonen av dynamisk og statisk analyse gir en mer helhetlig forståelse av skadevaren og danner et solid grunnlag for vurdering av trussel potensialet og utvikling av mottiltak.

Offline analyse kombinerer dynamisk og statisk analyse, da det ikke krever noe internettilgang. Dette gir en høy grad av sikkerhet, da det ikke er noe risiko for at skadevaren kan kommunisere med eksterne servere eller laste ned skadelige programmer. Skadevaren forblir isolert. Offline analyse er veldig tidskrevende og krever høy teknisk kompetanse, ikke bare for å tolke funn, men også for å gjennomføre manuell analyse. Da det ikke er tilgang til noen databaser, kan det bli utfordrende å identifisere nye trusler eller mer sofistikerte trusler.

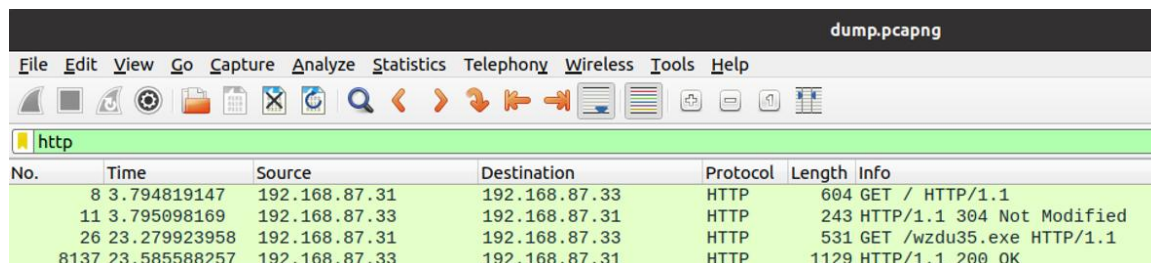
Online analyse er et svært attraktivt alternativ, særlig for personer med begrenset kompetanse/erfaring innen manuell kode analyse. Tjenester som Virus Total og Joe Sandbox gir brukervennlige løsninger med raske og detaljerte resultater. Online analyse tillater innsending av skadevare ved å laste opp filen direkte eller ved å oppgi en URL hvor skadevaren er lokalisert, tjenestene vedlikeholder omfattende databaser med tidligere innsendte prøver, som muliggjør raske og relevante svar basert på eksisterende analyser (Anson, 2020, s 396). Online analyse er et verdifullt verktøy i tidssensitive situasjoner, da det enkelt identifisere skadevarens egenskaper og oppførsel, noe som er avgjørende i situasjoner hvor rask respons er kritisk. En stor ulempe er usikkerheten rundt hvor pålitelige resultatene

faktisk er, samt risikoen for datalekkasje. Når skadevare lastes opp, kan dataen lagres i en database eller deles med en tredjepart. Dette kan eksponere sensitiv informasjon om skadevaren til uautoriserte aktører, noe som er veldig problematisk dersom skadevaren er en del av en pågående etterforskning.

1.b

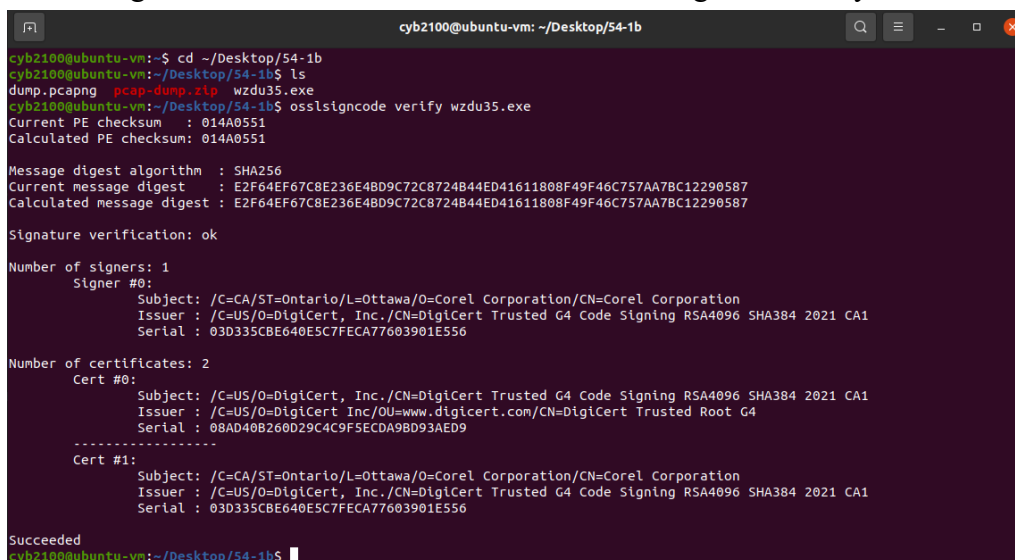
For å identifisere den mistenkelige filen ble verktøyet Wireshark benyttet. Jeg åpnet pakkedumpen dump.pcapng i Wireshark ved hjelp av kommandoen: Wireshark dump.pcapng.

Inni Wireshark brukte jeg filterseksjonen til å skrive inn «http» for å filtrere kun http-trafikk.



No.	Time	Source	Destination	Protocol	Length	Info
8	3.794819147	192.168.87.31	192.168.87.33	HTTP	604	GET / HTTP/1.1
11	3.795098169	192.168.87.33	192.168.87.31	HTTP	243	HTTP/1.1 304 Not Modified
26	23.279923958	192.168.87.31	192.168.87.33	HTTP	531	GET /wzdu35.exe HTTP/1.1
8137	23.585588257	192.168.87.33	192.168.87.31	HTTP	1129	HTTP/1.1 200 OK

Etter å ha analysert HTTP-trafikken identifiserte jeg en GET-forespørsel for filen wzdu35.exe. Dette indikerer at filen ble lastet ned fra serveren med IP-adressen 192.168.87.33. Filen er mistenkelig og krever videre analyse for å vurdere mulige sikkerhetstrusler. For å finne ut om filen er signert ble det tatt i bruk kommandoen: osslsigncode verify wzdu35.exe.



```
cyb2100@ubuntu-vm: ~/Desktop/54-1b
cyb2100@ubuntu-vm:~$ cd ~/Desktop/54-1b
cyb2100@ubuntu-vm:~/Desktop/54-1b$ ls
dump.pcapng  pcap-dump.zip  wzdu35.exe
cyb2100@ubuntu-vm:~/Desktop/54-1b$ osslsigncode verify wzdu35.exe
Current PE checksum : 014A0551
Calculated PE checksum: 014A0551

Message digest algorithm : SHA256
Current message digest : E2F64EF67C8E236E4BD9C72C8724844ED41611808F49F46C757AA7BC12290587
Calculated message digest : E2F64EF67C8E236E4BD9C72C8724844ED41611808F49F46C757AA7BC12290587

Signature verification: ok

Number of signers: 1
  Signer #0:
    Subject: /C=CA/ST=Ontario/L=Ottawa/O=Corel Corporation/CN=Corel Corporation
    Issuer : /C=US/O=DigiCert, Inc./CN=DigiCert Trusted G4 Code Signing RSA4096 SHA384 2021 CA1
    Serial : 03D335CBE640E5C7FECA77603901E556

Number of certificates: 2
  Cert #0:
    Subject: /C=US/O=DigiCert, Inc./CN=DigiCert Trusted G4 Code Signing RSA4096 SHA384 2021 CA1
    Issuer : /C=US/O=DigiCert Inc./OU=www.digicert.com/CN=DigiCert Trusted Root G4
    Serial : 08AD40B260D29C4C9F5ECDA9BD93AED9
  -----
  Cert #1:
    Subject: /C=CA/ST=Ontario/L=Ottawa/O=Corel Corporation/CN=Corel Corporation
    Issuer : /C=US/O=DigiCert, Inc./CN=DigiCert Trusted G4 Code Signing RSA4096 SHA384 2021 CA1
    Serial : 03D335CBE640E5C7FECA77603901E556

Succeeded
cyb2100@ubuntu-vm:~/Desktop/54-1b$
```

Filene er signert, her er flere detaljer rundt signaturen:

- **Signert av:** Corel Corporation fra Ottawa, Canada.
- **Signatur algoritme:** SHA256.
- **Signatur verification:** ok (gyldig signatur)

Kandidatnr: 54

Filen ble først observert på VirusTotal 2024-01-31.

10 av 70 sikkerhetsleverandører flagget filen som skadelig. Her er noen eksempler

- CrowdStrike Falcon: Win/grayware_confidence_100% (D)
- DrWeb: Program.Unwanted.4644
- Fortinet: Adware/DriverReviver

Filen er identifisert som adware av flere aktører. Dette er en type skadelig programvare som kan installere seg selv på brukerens system uten samtykke. Adware er kjent for å vise uønskede reklamer og pop-ups (Malwarebytes, u.å.). Filen blir oppdaget som mistenkelig av noen aktører, men ikke alle. Dette indikere at filen kan være skadelig.

https://www.virustotal.com/gui/file/73176a97801a58e4148e407a2b6336ad8791fd8fc381bffa3cee753ec394d0a

73176a97801a58e4148e407a2b6336ad8791fd8fc381bffa3cee753ec394d0a
wdu35.exe
Size: 20.61 MB
Last Analysis Date: 4 hours ago

10/70 security vendors flagged this file as malicious

Community Score: 10/70

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label: adware:driverreviver Threat categories: adware Family labels: driverreviver

Security vendors' analysis

Vendor	Detection	Signature
CrowdStrike Falcon	Malicious	Win/grayware_confidence_100% (D)
DrWeb	Malicious	Program.Unwanted.4644
Fortinet	Malicious	Adware/DriverReviver
Ikarus	Malicious	PUA.DriverReviver
K7GW	Malicious	Adware (0054dc181)
Acronis (Static ML)	Undetected	
Alibaba	Undetected	
ALYac	Undetected	
Arcabit	Undetected	

Vurdering:

Filen har en gyldig signatur og er signert av Corel Corporation, noe som kan indikere at den kommer fra en troverdig kilde. Dette gir imidlertid ingen garanti for at filen er trygg, da selv signerte filer kan inneholde skadevare. Virus Total gir oss et resultat som sier at 10 av 70 sikkerhetsmotorer vurderer filen som skadelig eller mistenkelig. Filen kan bli brukt til å vise uønsket reklame eller gjennomføre handlinger for en bruker uten samtykke. Denne filen bør behandles med forsiktighet, da den representerer en mulig sikkerhetsrisiko.

1.c

Jeg brukte kommandoen `sigtool --md5` for å generere en MD5-hash for `wzdu35.exe`, som jeg lagret i `signatur.hdb`. Jeg valgte `.hdb`-formatet fordi det er spesielt designet for å lagre hash-baserte signaturer i ClamAV. Deretter brukte jeg kommandoen `cat` `signatur.hdb` for å bekrefte at signaturfilen inneholder den korrekte signaturen basert på MD5-hashen. Dette forsikrer at signaturen er riktig konfigurert. Videre testet jeg signaturen ved å kjøre kommandoen `clamscan -d` `signatur.hdb` `wzdu35.exe`. Resultatet var `UNOFFICIAL FOUND`, noe som bekrefter at signaturen matcher korrekt mot filen `wzdu35.exe`.

Nedenfor finner du screenshots som dokumentere hvordan jeg utførte oppgaven, og verifisering for å sikre at signaturen ikke treffer andre filer:

```
cyb2100@ubuntu-vm:~$ cd ~/Desktop/54-1c
cyb2100@ubuntu-vm:~/Desktop/54-1c$ ls
windows_test  wzdu35.exe
cyb2100@ubuntu-vm:~/Desktop/54-1c$ sigtool --md5 wzdu35.exe > signatur.hdb
cyb2100@ubuntu-vm:~/Desktop/54-1c$ cat signatur.hdb
44eb18e735404426b15014fb1c9b1447:21613504:wzdu35.exe
cyb2100@ubuntu-vm:~/Desktop/54-1c$ clamscan -d signatur.hdb wzdu35.exe
/home/cyb2100/Desktop/54-1c/wzdu35.exe: wzdu35.exe.UNOFFICIAL FOUND

----- SCAN SUMMARY -----
Known viruses: 1
Engine version: 0.103.12
Scanned directories: 0
Scanned files: 1
Infected files: 1
Data scanned: 20.62 MB
Data read: 20.61 MB (ratio 1.00:1)
Time: 0.322 sec (0 m 0 s)
Start Date: 2024-11-26 20:40:33
End Date: 2024-11-26 20:40:33
cyb2100@ubuntu-vm:~/Desktop/54-1c$ clamscan -d signatur.hdb .
/home/cyb2100/Desktop/54-1c/wzdu35.exe: wzdu35.exe.UNOFFICIAL FOUND
/home/cyb2100/Desktop/54-1c/signatur.hdb: OK

cyb2100@ubuntu-vm:~/Desktop/54-1c$ clamscan --database=signatur.hdb ~/Desktop/windows_test
/home/cyb2100/Desktop/windows_test/hemmelig.txt: OK
/home/cyb2100/Desktop/windows_test/Pentest Rapport.pdf: OK

----- SCAN SUMMARY -----
Known viruses: 1
Engine version: 0.103.12
Scanned directories: 1
Scanned files: 2
Infected files: 0
Data scanned: 1.39 MB
```

1.d

PID (Process ID) er en unik ID gitt til en prosess når den starter.

PPID (Parent process ID) er ID-en til prosessen som startet prosessen med den aktuelle PID.

Ved å kjøre kommandoen: `vol.py -f mem.raw windows.pslist`, finner vi ut at `wzdu35.exe` har verdiene: PID: 288, PPID: 2312

***	4184	880	wuauclt.exe	0x8403b0d8	6	124	0	False	2024-11-20 11:00:20.000000	N/A
**	1520	428	vmicsvc.exe	0x856f1360	4	80	0	False	2024-11-20 10:00:15.000000	N/A
**	5996	428	mscorsvw.exe	0x840f2880	6	77	0	False	2024-11-20 11:01:15.000000	N/A
**	1532	428	svchost.exe	0x857aed28	5	94	0	False	2024-11-20 10:00:17.000000	N/A
352	332		csrss.exe	0x852867d0	9	243	1	False	2024-11-20 19:00:11.000000	N/A
*	3960	352	conhost.exe	0x846421f0	2	52	1	False	2024-11-20 11:02:29.000000	N/A
392	332		winlogon.exe	0x853eed28	5	117	1	False	2024-11-20 19:00:12.000000	N/A
2312	2280		explorer.exe	0x8536c660	30	875	1	False	2024-11-20 10:01:07.000000	N/A
*	2408	2312	VBoxTray.exe	0x84c03278	13	157	1	False	2024-11-20 10:01:08.000000	N/A
*	3072	2312	taskmgr.exe	0x84145d28	7	117	1	False	2024-11-20 11:00:07.000000	N/A
*	288	2312	wzdu35.exe	0x8550d888	1	101	1	False	2024-11-20 10:59:33.000000	N/A
**	4124	288	ca92b726-29d9-	0x85548030	1	119	1	False	2024-11-20 10:59:33.000000	N/A
*	3676	2312	DumpIt.exe	0x84653908	1	18	1	False	2024-11-20 11:02:29.000000	N/A
2504	1608		firefox.exe	0x8580bbc0	0	-	1	False	2024-11-20 10:53:56.000000	2024-11-20 10:58:59.000000

cyb2100@ubuntu-vm:~/Desktop/54-1d\$

Har prosessen vår noen nettverksforbindelser åpne, i så fall hvilke?

For å undersøke om prosessen PID 288 har noen aktive nettverksforbindelser, ble tatt i bruk kommandoene: `vol.py -f mem.raw windows.netscan` og `vol.py -f mem.raw windows.netstat`. Ingen nettverksforbindelse ble funnet på denne prosessen.

Bruk Yara funksjonen yarascan i Volatility og lag deteksjon:

Jeg opprettet en Yara-regel, `wzdu35_deteksjon.yar`. Denne regelen ble opprettet for å detektere strengen `wzdu35.exe` i minnedumpen. Deretter brukte jeg yarascan i volatility for å analysere minnedumpen. Da brukte jeg kommandoen:

```
vol.py -f mem.raw yarascan.YaraScan --yara-file=wzdu35_deteksjon.yar > yarascan_resultat.txt
```

- `-f mem.raw` : Dette spesifiserer at `mem.raw` skal analyseres.
- `Yarascan.yarascan` : Aktiverer yarascan i volatility for å bruke yararegelen på `mem.raw`
- `--yara-file=wzdu35_deteksjon.yar`: Dette sier hvilken yara regel som kalles brukes.
- `Yarascan_resultat.txt`: Lagrer resultatet fra skanningen i en egen fil.

```
cyb2100@ubuntu-vm: ~/Desktop/54-1d
cyb2100@ubuntu-vm:~/Desktop/54-1d$ ls
mem.raw  mem.zip
cyb2100@ubuntu-vm:~/Desktop/54-1d$ nano wzdu35_deteksjon.yar
cyb2100@ubuntu-vm:~/Desktop/54-1d$ cat wzdu35_deteksjon.yar
rule detect_Wzdu35 {
  strings:
    $string1 = "wzdu35.exe"
  condition:
    $string1
}
cyb2100@ubuntu-vm:~/Desktop/54-1d$ vol.py -f mem.raw yarascan.YaraScan --yara-file=wzdu35_deteksjon.yar > yarascan_resultat.txt
cyb2100@ubuntu-vm:~/Desktop/54-1d$ scanning finished
cyb2100@ubuntu-vm:~/Desktop/54-1d$ ls
mem.raw  mem.zip  wzdu35_deteksjon.yar  yarascan_resultat.txt
cyb2100@ubuntu-vm:~/Desktop/54-1d$ cat yarascanresultat.txt
cat: yarascanresultat.txt: No such file or directory
cyb2100@ubuntu-vm:~/Desktop/54-1d$ cat yarascan_resultat.txt
Volatility 3 Framework 2.5.0

Offset  Rule      Component      Value
-----
0x8550d9f4  detect_Wzdu35  $string1       77 7a 64 75 33 35 2e 65 78 65
0x86557eca  detect_Wzdu35  $string1       77 7a 64 75 33 35 2e 65 78 65
cyb2100@ubuntu-vm:~/Desktop/54-1d$
```

2.a

Tre områder hvor AI kan bidra positivt innen cyberforsvar er trussel deteksjon, bekjempelse av bots, SOC og responsbehandling.

En av AI sine største styrker er evnen til å prosessere og analysere store datamengder, noe som gjør den svært effektiv til å oppdage mistenkelig aktivitet som kan indikere et cyberangrep (Shutenko, 2024). Ved hjelp av avanserte algoritmer kan AI analysere endringer i atferd, noe som gjør det mulig å detektere både kjente og nye trusler. F.eks. kan AI oppdage

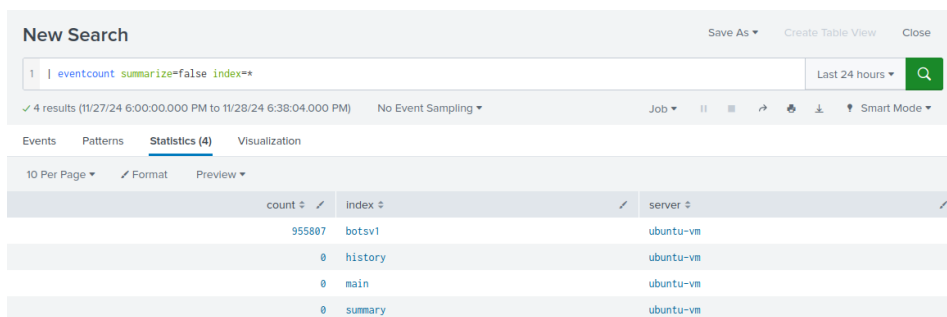
om en ansatt klikker på en phishing e-post ved å analysere nettverksaktiviteten og varsle om mulig sikkerhetsbrudd. Ettersom cyberangrep blir stadig mer avanserte, er AI sin evne til å oppdage mistenkelig aktivitet uten menneskelig interaksjon kritisk. Denne automatiseringen gir organisasjoner muligheten til å reagere tidlig på trusler, noe som vil redusere økonomiske kostnader og minimere skadeomfanget.

Bekjempelse av bots, bots er et stadig økende problem innen cybersikkerhet, de brukes til å spre skadevare, stjele data og utføre angrep som DDoS eller brute-force. Ved hjelp av AI kan vi effektivt bekjempe bots ved å analysere og gjenkjenne deres mønster og atferd (Srivastava, 2024). Bekjempelse av bots er et viktig område innen cyberforsvar fordi bots utgjør en betydelig del av trusselbildet.

SOC og responsbehandling, SOC er et sentralt element innen cyberforsvar og er sammen med Incident Response et område som kan dra stort nytte av AI. Som Telenor påpeker, "mye av det såkalte «førstelinjearbeidet» som utføres av analytikere på SOC-en kan automatiseres eller støttes av AI " (Telenor, 2024). AI kan redusere arbeidsmengden til SOC-teamet ved å håndtere oppgaver som eliminering av falske positive. Dette gjør at SOC-analytikere kan bruke mer ressurser og tid på komplekse og kritiske sikkerhetshendelser.

2.b

Jeg brukte følgende spørring for å liste opp indexene mine i splunk:



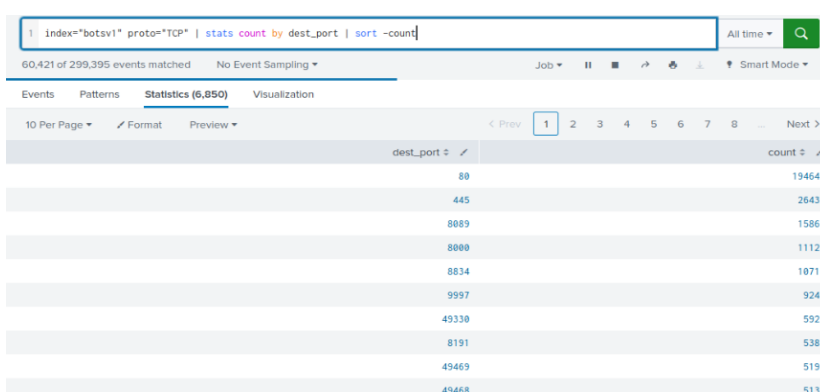
New Search			Save As	Create Table View	Close
1 eventcount summarize=false index=*		Last 24 hours			
✓ 4 results (11/27/24 6:00:00.000 PM to 11/28/24 6:38:04.000 PM) No Event Sampling					
Events	Patterns	Statistics (4)	Visualization		
10 Per Page Format Preview					
count	index	server			
955807	botsv1	ubuntu-vm			
0	history	ubuntu-vm			
0	main	ubuntu-vm			
0	summary	ubuntu-vm			

Eventcount: Teller hendelsene i indexene.

Summarize=false: Dette gir fullstendig data.

Index=* : søker gjennom alle tilgjengelige indexer i splunk.

For å lage en tabell over prokoller som kjører over TCP brukte jeg følgende spørring:



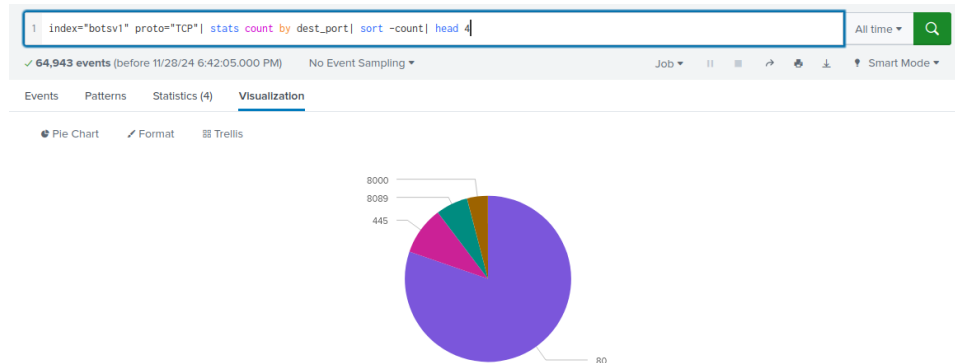
1 index="botsv1" proto="TCP" stats count by dest_port sort -count		All time
60,421 of 299,395 events matched No Event Sampling		
Events	Patterns	Statistics (6,850)
10 Per Page Format Preview		
dest_port	count	
80	19464	
445	2643	
8089	1586	
8000	1112	
8034	1071	
9997	924	
49330	592	
8191	538	
49469	519	
49468	513	

Proto=TCP: Filtrerer slik at kun protokoller som kjører over TCP kommer opp.

Stats count by dest_port: Teller antall forbindelser for hver destinasjonsport og viser hvor mange forbindelser hver port har.

Sort – count: Sorterer resultatet i rekkefølge.

For å vise de fire protokollene med høyest forekomst, brukte jeg følgende spørring:



Her brukte jeg den samme spørring som i forrige oppgave, men jeg la til:

Head 4: Dette gjør at kun de fire med høyest verdier vises. Deretter gikk jeg til visualization og valgte pie chart.

2.c

index="botsv1" "imreallynotbatman.com" | stats count by src_ip | sort - count

src_ip	count
40.80.148.42	34967
192.168.250.70	11493
23.22.63.114	2883

index="botsv1": Søker i botsv1 indexen.

"imreallynotbatman.com": Fokusere data på kun dette domenet.

stats count by src_ip: Teller antall hendelser gruppert etter IP.

sort – count: Sorterer resultatet i rekkefølge.

IP-en 40.80.148.42 ser ut til å skanne imreallynotbarman.com, da den har flest forespørsler.

index="botsv1" imreallynotbatman.com uri=/joomla/Administrator/index.php | stats values(dest_ip) as IP, values(dest_port) as Port, values(server) as Server

IP	Port	Server
192.168.250.70	80	Microsoft-IIS/8.5

stats values(dest_ip) as IP, values(dest_port) as Port, values(server) as Server:

Imreallybatman.com befinner seg på IP 192.168.250.70, TCP port: 80, Server: Microsoft-IIS/8.5.

1

index=botsv1 sourcetype=suricata "Cross site scripting"

2

| stats count by src_ip

All time

🔍

✓ 151 events (before 11/28/24 4:10:36.000 AM)

No Event Sampling ▼

Job ▼

⏸

■

↶

🖨

⬇

💡 Smart Mode ▼

Events

Patterns

Statistics (1)

Visualization

10 Per Page ▼

✎ Format

Preview ▼

src_ip ↕	count ↕
40.80.148.42	151

IP-en 40.80.148.42 ser ut til å trigge Suricata med en Cross Site Scripting advarsel.

Query:

```
index=botsv1 dest_ip="192.168.250.40" dest_port=8191  
| stats values(src_ip) as Attempted_IP, values(response) as Response, values(status) as Status
```

Results: 534 events (before 11/28/24 4:11:46.000 AM) No Event Sampling

Tabs: Events Patterns **Statistics (1)** Visualization

Table Headers: Attempted_IP Response Status

Attempted_IP	Response	Status
192.168.2.50	200	

stats values(src_ip) as Attempted_IP, values(response) as Response, values(status) as Status: Dette henter kilden til forespørgslen, serverens svar på forespørgslen og statuskode.

IP-en som forsøker å logge seg på er 192.168.2.50. Vi får ikke noe respons, men status koden er 200 (ok).

`index=botsv1 sourcetype=stream:http dest_ip="192.168.250.70"`

`uri=/joomla/Administrator/index.php`

`| rex field=form_data "passwd=(?<password>\w+)"`

`| where match(password,"^\d+$") OR match(password,"^s")`

`| eval type=if(match(password,"^\d+$"),"Kun_tall","Starter_med_s")`

`| stats count by type`



The screenshot shows a Splunk search interface. The search bar contains the following query:

```
1 index=botsv1 sourcetype=stream:http dest_ip="192.168.250.70" uri=/joomla/Administrator/index.php
2 | rex field=form_data "passwd=(?<password>\w+)"
3 | where match(password,"^\d+$") OR match(password,"^s")
4 | eval type=if(match(password,"^\d+$"),"Kun_tall","Starter_med_s")
5 | stats count by type
```

Below the search bar, it indicates "72 events (before 11/29/24 3:52:35.000 AM)". The "Statistics (2)" tab is selected, showing a table with the following data:

type	count
Kun_tall	33
Starter_med_s	39

`index=botsv1 sourcetype=stream:http dest_ip="192.168.250.70"`: Søker i botsv1 indexen for HTTP-trafikk til IP 192.168.250.70.

`uri=/joomla/Administrator/index.php`: Setter søkelys på forespørsler mot denne URI-en.

`rex field=form_data "passwd=(?<password>\w+)"`: Henter passord fra form_data og lagrer det som passord.

`where match(password,"^\d+$") OR match(password,"^s")`: Filtrerer passord med kun tall eller som starter med bokstaven s.

`eval type=if(match(password,"^\d+$"),"Kun_tall","Starter_med_s")`: Dette gjør det enklere å lese svaret og ser penere ut.

`stats count by type`: Gir oss antall passord i hver kategori.

Det var 33 passord som inneholdt kun tall og 39 passord som startet med bokstaven s.

3.a

En potensiell problematikk med dette oppsettet er at det er avhengige av skybaserte systemer som iSolarCloud. Dette gjør systemet sårbar ovenfor cyberangrep. Et vellykket angrep kan føre til full kontroll over systemet. Det bør i stedet brukes et segmentert privat nettverk, for styrke sikkerheten. Det å bruke kinesisk teknologi i kritisk infrastruktur kan være problematisk med tanke på overvåkning og spionasje. Kinesiske teknologiselskaper er kjent for å være litt kontroversielle, på grunn av samarbeid med kinesiske myndigheter. Et eksempel på dette er da Australia innførte et forbud mot Huawei og ZTE-utstyr i landets kommende 5G-nett. Ifølge en artikkel på Tek.no: "Australiske sikkerhetsekspertene viste til

Kinas lovendring fra 2017, som pålegger «enhver organisasjon eller medborger å støtte og samarbeide med landets etterforskningsarbeid». Dette har skapt bekymringer for at kinesisk teknologi kan bli utnyttet til spionasje eller sabotasje, særlig i kritisk infrastruktur" (Kvalheim og Plikk, 2018). Siden SunGrow opererer i kritisk infrastruktur, kan bruken av kinesisk teknologi utgjøre en risiko for bakdører i programvaren og uautorisert tilgang. Dette kan medføre sabotasje eller spionasje.

3.b

Basert på NSM grunnprinsipper for IKT sikkerhet ville jeg som CISO tatt i bruk disse tre prinsippene for å stryke sikkerheten:

2.2 – Etabler en sikker IKT-arkitektur. Et IKT system må planlegges på en sikker måte, da dårlig oversikt over byggeprosessen kan føre til sikkerhetshull og inngangsdører for en potensiell angriper. Ifølge NSM bør "IKT-systemet deles opp i forskjellige deler avhengig av tillitsnivå. Slik oppdeling bør etableres for nettverk, samt for logiske deler. Hvis man ikke gjør dette kan konsekvenser i forbindelse med et angrep eller menneskelig driftsfeil omfatte hele virksomheten, i stedet for kun en begrenset del" (NSM, 2024, S. 20).

2.4 – Beskytt varsomhetens nettverk. Målet med dette prinsippet er å beskytte virksomheten mot eksterne og interne trusler. For å beskytte nettverket mot eksterne trusler bør det implementeres en IDS, som kan avdekke og varsle om mistenkelig aktivitet i nettverket. I tillegg bør kommunikasjonen mellom cloud og lokale systemer være kryptert for å unngå datalekkasje eller avlytting. 3.4 – Gjennomfør inntrengingstester, gjennomfør regelmessig testing av systemet for å identifisere sårbarheter før en angriper kan utnytte det.

3.c

Under cyberangrepet i Lviv ble skadevaren FrostyGoop brukt. Som forklart i en artikkel hos Cyberscoop: "The malware, which Dragos has named FrostyGoop, uses Modbus to allow attackers to further attack industrial-controlled systems (ICS)" (Vasquez, 2024). Angrepet ble mest sannsynlig utført av den russiske militære hackergruppen Sandworm, da de er kjent for å ha stor innvirkning på Ukrainas kritiske infrastruktur. De har også tidligere utført flere angrep mot Ukrainas strømmett. Dette kunne vært unngått ved bruk av PR.AA fra NIST CSF 2.0, dette sikrer tilgang til kun autoriserte brukere og enheter. Noe som ville hindret FrostyGoop i å få tilgang til systemet. DE.CM ville oppdaget mistenkelig aktivitet, som Modus og sendt ut et varsel. RC.RP, dersom de hadde hatt en solid recovery plan kunne de ha redusert nedetiden etter angrepet.

Referanseliste:

Anson, S. (2020). *Applied Incident Response*

Aquasec. (2023, 15. Februar). *Malware Analysis: Static vs Dynamic and 4 Critical Best Practices*: <https://www.aquasec.com/cloud-native-academy/cloud-attacks/malware-analysis/>

Malwarebytes. (u.å.). *What is adware?*: <https://www.malwarebytes.com/adware>

Srivastava, S. (2024, 23. september). AI in Cybersecurity - Uses, Threats & Prevention. *Engati*: <https://www.engati.com/blog/ai-in-cybersecurity>

Shutenko, V. (2024, 8. august). AI in Cybersecurity. *Techmagi*: <https://www.techmagic.co/blog/ai-in-cybersecurity>

Telenor. (2024). *AI og sikkerhet – utfordringer og muligheter*: <https://www.telenor.no/om/digital-sikkerhet/2024/ai-og-sikkerhet/>

Kvalheim, F. L og Plikk, N. (2018, 27. august). Huawei stenges ute i flere land - i Norge får de store kontrakter. *TEK.no*: <https://www.tek.no/nyheter/nyhet/i/BRm5QE/huawei-stenges-ute-i-flere-land-i-norge-faar-de-store-kontrakter>

Nasjonal sikkerhetsmyndighet. (2024). *NSMs Grunnprinsipper for IKT-sikkerhet(v2.1)*: <https://nsm.no/getfile.php/1313975-1717589722/NSM/Filer/Dokumenter/Veiledere/NSMs%20Grunnprinsipper%20for%20IKT-sikkerhet%20v2.1.pdf>

Vasquez, C. (2024, 23. Juli). Simple ‘FrostyGoop’ malware responsible for turning off Ukrainians’ heat in January attack, *Cyberscoop*: <https://cyberscoop.com/frostygoop-ics-malware-dragos-ukraine/>

Dragos. (2024, 23. juli). *Protect Against the FrostyGoop ICS Malware Threat with OT Cybersecurity Basics*: <https://www.dragos.com/blog/protect-against-frostygoop-ics-malware-targeting-operational-technology/>