CENTRE FOR CYBER SECURITY BELGIUM

Log4J! What exactly is going on?

What is it? How to Identify, Patch or Mitigate?

How to Monitor for active exploitation attempts?

Kevin HOLVOET

Cyber Threat Intelligence Analyst at CCB/CyTRIS

Cyber Threat Research & Intelligence Sharing

SANS Instructor of FOR578: Cyber Threat Intelligence

@digihash

kevinholvoet

kevin.holvoet@cert.be

TLP WHITE

16 December 2021

Agenda

- What is Log4j?
- What is the Log4Shell vulnerability?
- How to Identify vulnerable software?
- How to patch/mitigate?
- How to monitor/detect intrusion attempts?

What is Log4j?

- 20 years old, initial release on January 8, 2001
- Part of Apache Logging Services
  - Project Apache Software Foundation
- Java-based logging framework
- Abstraction layer for logging in Java programs
- Making it easy for developers to implement fast and reliable application logging.
- Features
  - Default and custom log levels
  - Simple configuration syntax
  - Fast & reliable with asynchronous loggers
  - API
  - Support for multiple formats, databases, log management systems

Log4j Example Code

```
import org.apache.log4j.Logger;

import java.io.*;
import java.sql.SQLException;
import java.util.*;

public class log4jExample{

   /* Get actual class name to be printed on */
   static Logger log = Logger.getLogger(log4jExample.class.getName());

   public static void main(String[] args)throws IOException,SQLException {
      log.debug("Hello this is a debug message");
      log.info("Hello this is an info message");
   }
}
```

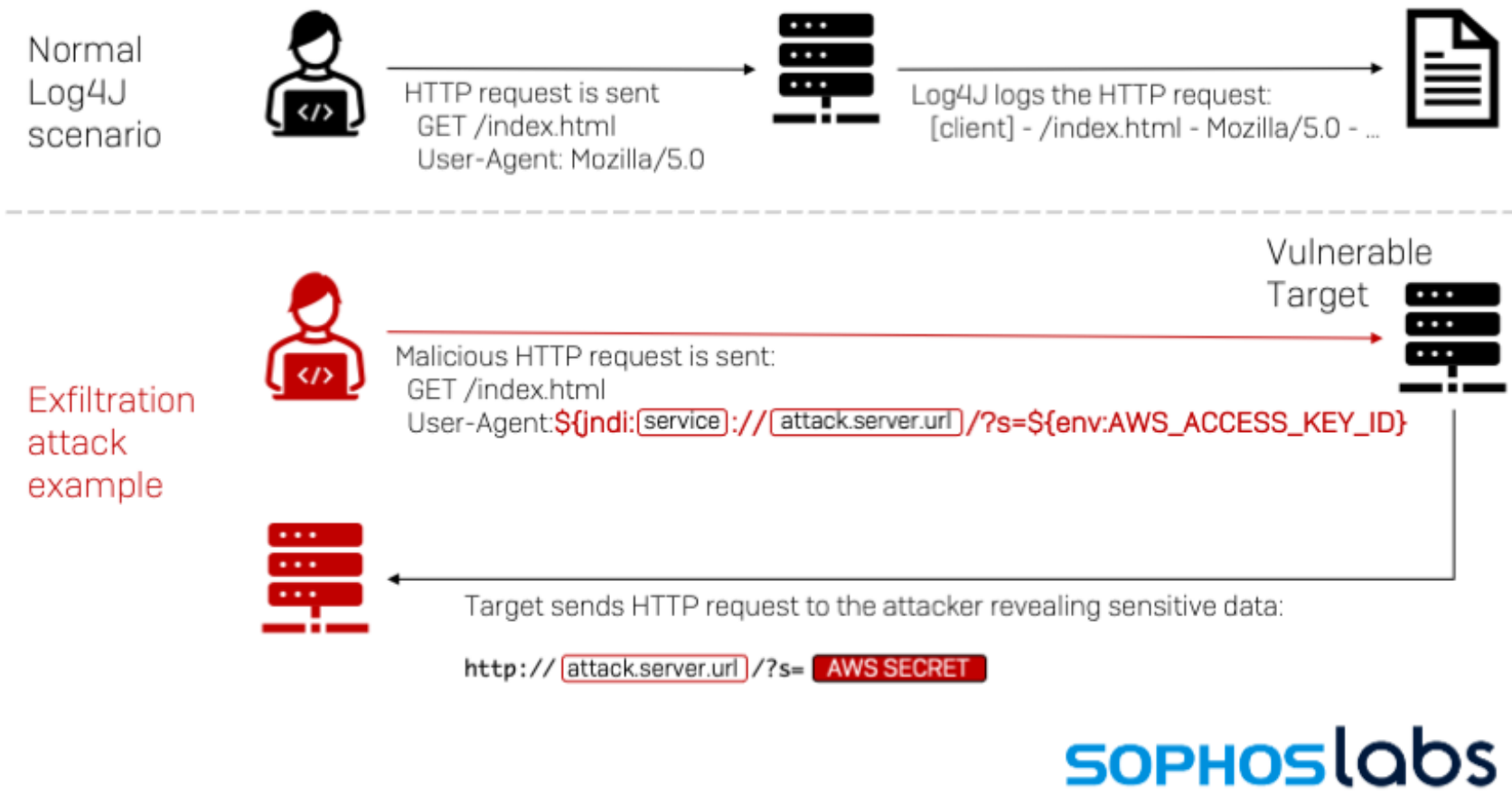
What is the Log4Shell vulnerability?

https://www.lunasec.io/docs/blog/log4j-zero-day/

- CVE-2021-44228
  - CVSSv3 score: 10/10
  - Remote Code Execution (RCE) vulnerability
  - Released with exploit code on Thursday 9 December 2021
- Affected software:
  - Apache Log4j
    - >= v2 2.0-beta9 AND <= v2.12.1
    - >= 2.13.0 AND < 2.15.0
    - Log4J 1.x with specific configs
  - Including ALL software using Log4J as a logging framework
- What is affected?
  - Message lookup substitution feature using JNDI
  - Enabled by default
  - No URL sanitization
  - Any kind of request is fully processed, even obfuscated or encoded (e.g., Base64)
  - Log4J reaches out to external server

- Risks
  - Successful exploitation attempt => unauthenticated attacker can remotely execute arbitrary code.
  - Actors actively scanning & exploiting in the wild
    - Installing coin miners or Cobalt Strike
    - Exfiltrating data from compromised systems
    - HAFNIUM (China state-sponsored)
      - Attack virtualization infrastructure
    - PHOSPHORUS (Iran)
      - Deploy ransomware
    - New Khonsari ransomware family + Orcus RAT
    - Other state actors experimenting: China, Iran, North Korea, Turkey
    - Initial Access Brokers (Windows / Linux)
    - Botnets: Mirai, Tsunami (Muhstik), Kinsing, Elknot (BillGates), m8220, SiteLoader, xmrig miner

What is the Log4Shell vulnerability?



What is the Log4Shell vulnerability?

https://www.lunasec.io/docs/blog/log4j-zero-day/

- CVE-2021-45046
  - CVSSv3 score: 3.7/10
  - Denial Of Service (DOS) attack in non-default configurations
  - Found and released on 13 December 2021
  - When logging configuration uses non-default Pattern layout:
    - With Context Lookup (e.g., $${ctx:loginId})
    - OR Threat Context Map (MDC) pattern (e.g., %X, %mdc, or %MDC)

- Affected software:
  - Java 7: < v2.12.2
  - Java 8+: < v2.16.0

Log4Shell - Timeline

24 November

- Vulnerability reported by Chen Zhaojun of Alibaba
- RCE using JNDI message lookups

26 November

- CVE-2021-44228 first reported

1st December 04:36 UTC

- First exploitations observed
- By Cloudflare & Cisco

9th December 14:25 UTC

- First tweets by @P0rZ9 including PoC exploit on Github
- LunaSec blog post

10th December

- Active scanning and exploitation observed in Belgium, incl. obfuscation
- Log4J 1.x vulnerable through JMSAppender
- CVE-2021-4104

11th December

- Apache releases first patch: v2.15.0
- Minecraft under active exploitation

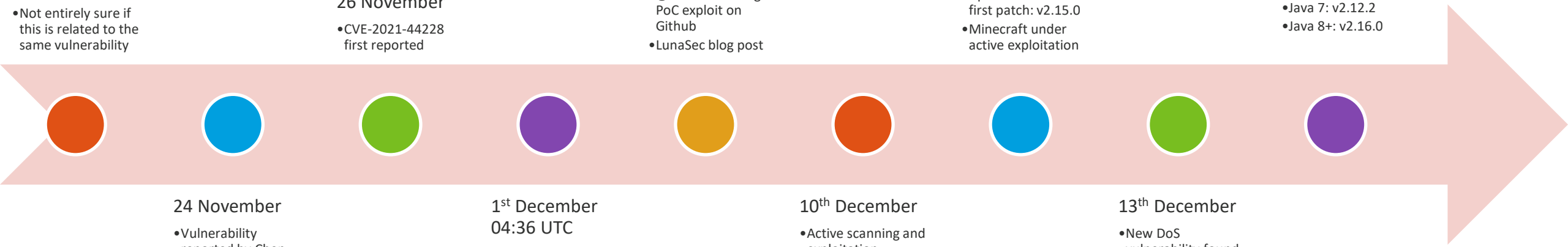
13th December

- New DoS vulnerability found with non-default configurations

14-15th December

- CVE-2021-45046 released
- New Log4J versions
  - Java 7: v2.12.2
  - Java 8+: v2.16.0

Log4Shell - Timeline



13 March

- Possible first POC on Github
- Not entirely sure if this is related to the same vulnerability

24 November

- Vulnerability reported by Chen Zhaojun of Alibaba
- RCE using JNDI message lookups

26 November

- CVE-2021-44228 first reported

1st December 04:36 UTC

- First exploitations observed
- By Cloudflare & Cisco

9th December 14:25 UTC

- First tweets by @P0rZ9 including PoC exploit on Github
- LunaSec blog post

10th December

- Active scanning and exploitation observed in Belgium, incl. obfuscation
- Log4J 1.x vulnerable through JMSAppender
- CVE-2021-4104

11th December

- Apache releases first patch: v2.15.0
- Minecraft under active exploitation

13th December

- New DoS vulnerability found with non-default configurations

14-15th December

- CVE-2021-45046 released
- New Log4J versions
  - Java 7: v2.12.2
  - Java 8+: v2.16.0

How to Identify vulnerable software?

- Software lists
  - https://github.com/cisagov/log4j-affected-db
  - https://github.com/NCSC-NL/log4shell
  - https://www.techsolvency.com/story-so-far/cve-2021-44228-log4j-log4shell/
  - https://gist.github.com/SwitHak/b66db3a06c2955a9cb71a8718970c592
- Scan internally with scripts / commands
  - Use your own DNS server
  - Check which servers are doing a DNS lookup query after executing https://github.com/NorthwaveSecurity/log4jcheck
- Scan your own software project directories with Log4shell from LunaSec
  - https://github.com/lunasec-io/lunasec/releases/, https://www.lunasec.io/docs/blog/log4j-zero-day-mitigation-guide/
  - Scans for specific hashes and packages in your Java projects
  - Other languages like Scala, Groovy, or Clojure can also be impacted
  - Similar tool: https://github.com/logpresso/CVE-2021-44228-Scanner
- Scan your systems for these known vulnerable hashes: https://github.com/mubix/CVE-2021-44228-Log4Shell-Hashes/
- NCSC-UK: A file system search for log4j. Or checking dependency/package manager:

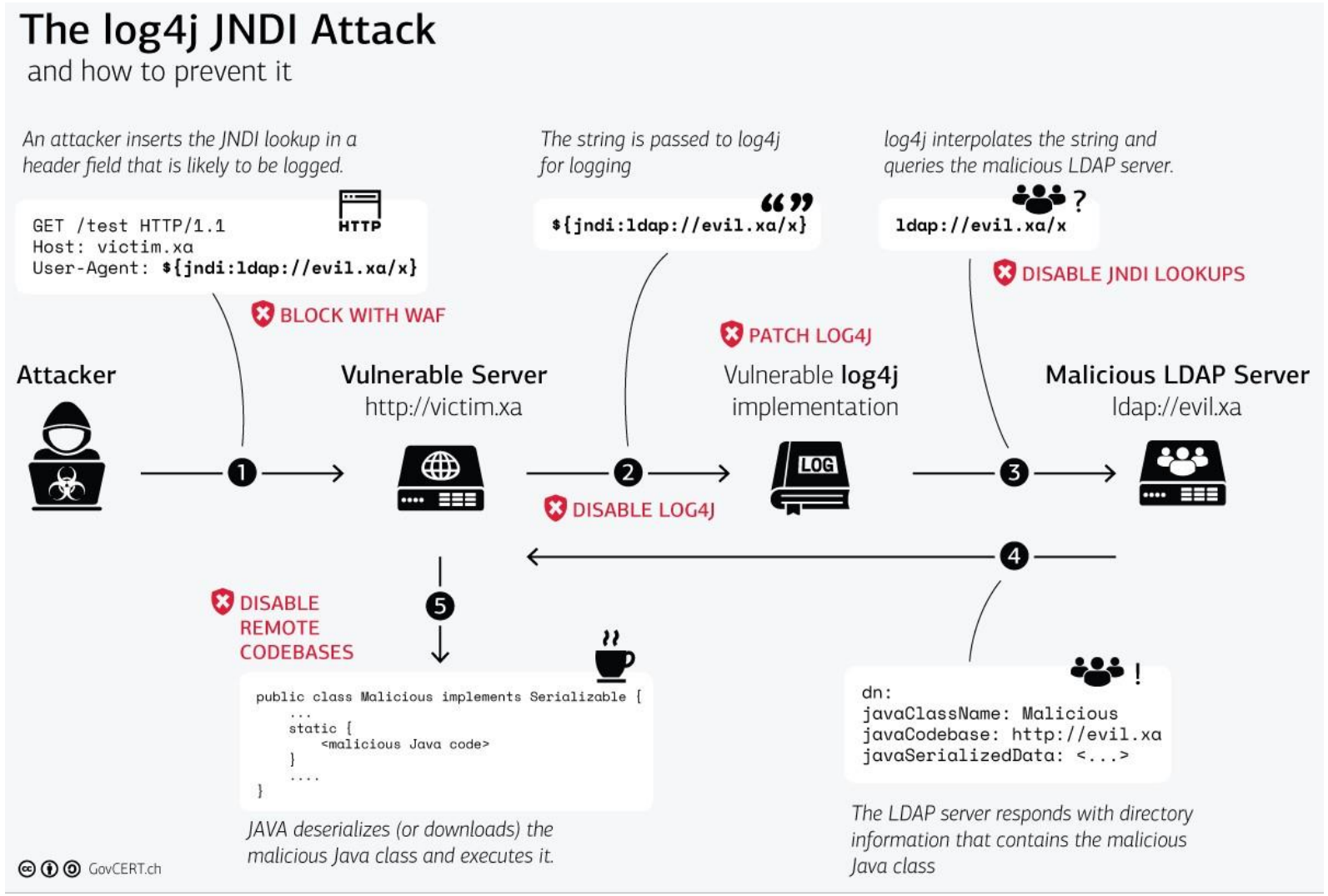
```
find / -type f -print0 |xargs -n1 -0 zipgrep -i log4j2 2>/dev/null
dpkg -l | grep log4j
```

How to patch/mitigate?

- Patch
  - all the software/appliances identified by your vendor with your vendor's recommendations/patches
    - See slide 10 on how to identify vulnerable software from the 4 lists.
  - Log4J libraries you use in your software to:
    - Java 7: v2.12.2
    - Java 8+: v2.16.0
- Mitigate
  - To mitigate vulnerabilities, users should switch **log4j2.formatMsgNoLookupsto to true** by:
    - Passing "-Dlog4j2.formatMsgNoLookups=true" as an argument when invoking Java.
    - Set environment variable
      - LOG4J_FORMAT_MSG_NO_LOOKUPS=true java ...
      - JAVA_OPTS=-Dlog4j2.formatMsgNoLookups=true
  - remove the JndiLookup class from the classpath:

```
zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class
```

  - Check with your security vendors how they can mitigate this attack.
    - E.g.: F5, Microsoft, ESET, etc… have come out with mitigations using their security systems.
  - Nginx LUA script: https://github.com/infiniroot/nginx-mitigate-log4shell



Steps of a log4j JNDI attack and suggested mitigations (Source: Swiss CERT)

Known bad advice

- Updating Java is insufficient
- A Web Application Firewall (WAF) will help mitigate in combination with other mitigation and detection techniques
  - A WAF can not entirely mitigate Log4Shell
  - Cat and mouse game
- Updating the log statement format with %m{nolookupzz} is not advisable

How to monitor/detect intrusion attempts? (1/2)

- Scan local log files with Log4Shell Detector (Python): https://github.com/Neo23x0/log4shell-detector
- Manually search your log files with Grep / Zgrep commands:
  - https://gist.github.com/Neo23x0/e4c8b03ff8cdf1fa63b7d15db6e3860b
  - Has commands for obfuscated variants too
- Find vulnerable software on Windows

```
gci 'C:\' -rec -force -include *.jar -ea 0 | foreach {select-string "JndiLookup.class" $_} | select -exp Path
```

- Use YARA rules: https://github.com/Neo23x0/signature-base/blob/master/yara/expl_log4j_cve_2021_44228.yar
- Implement known IOCs in your detection/protection systems
- General detection regex

```
\${(\${(.*?:.*?:.*?:-)('"`)*(?1)}*[jndi:lapsrm]('"`)*}*){9,11}
```

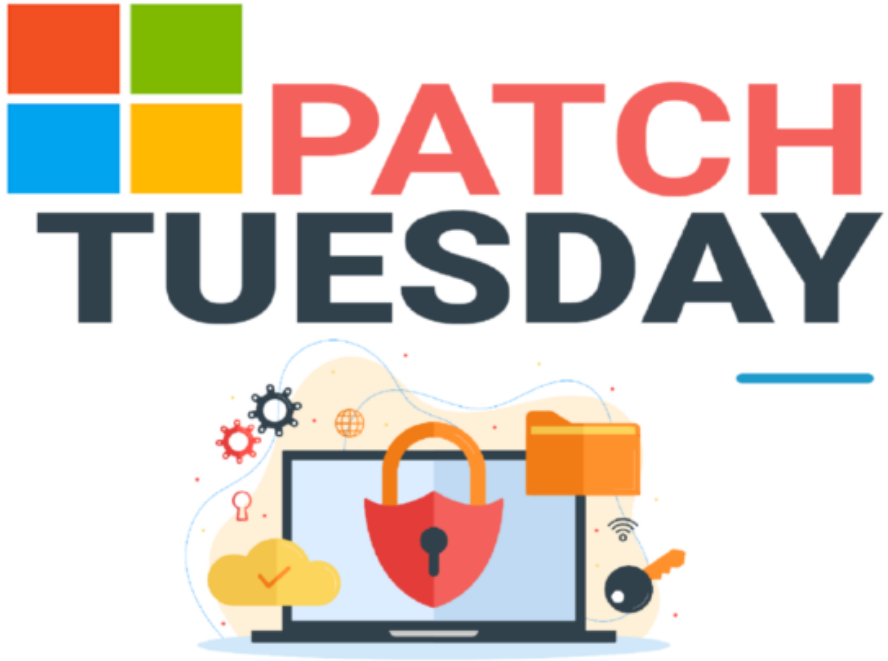
- Network-based detection
  - Diverto Nmap NSE scripts to check against log4shell: https://github.com/Diverto/nse-log4shell
  - NCC Group Log4Shell: Reconnaissance and post exploitation network detection: https://research.nccgroup.com/2021/12/12/log4shell-reconnaissance-and-post-exploitation-network-detection/

How to monitor/detect intrusion attempts? (2/2)

- Snort and Suricata rules
  - These are ET Open free community detections to alert on current exploit activity.
    - SID range 2034647-2034652: https://rules.emergingthreatspro.com/open/
- Host based detection
  - Florian Roth Sigma rule to detect an exploitation attempt against Log4j RCE vulnerability fields: https://github.com/SigmaHQ/sigma/blob/master/rules/web/web_cve_2021_44228_log4j_fields.yml
  - Florian Roth Sigma rule to Detect an exploitation attempt against Log4j RCE vulnerability: https://github.com/SigmaHQ/sigma/blob/master/rules/web/web_cve_2021_44228_log4j.yml
  - Powershell script to detect Log4Shell: https://github.com/sp4ir/incidentresponse/blob/35a2faae8512884bcd753f0de3fa1adc6ec326ed/Get-Log4shellVuln.ps1
  - Powershell script to perform a scan to see if it's vulnerable: https://github.com/crypt0jan/log4j-powershell-checker
  - NCCgroup Version hashes (MD5, SHA1 and SHA256) for Log4j2 versions: https://github.com/nccgroup/Cyber-Defence/tree/master/Intelligence/CVE-2021-44228
  - Huntress Online Log4Shell Vulnerability Tester: https://log4shell.huntress.com

Extras

- Don’t forget Microsoft Patch Tuesday
  - 1 zero-day: Windows AppX Installer Spoofing
  - 7 critical, 60 Important



- Next QCTR event on Thursday 13 January 2021
  - Invitation will be sent to anyone registered for any of the previous events
  - https://app.livestorm.co/ccb/
- Follow us on Twitter/LinkedIn
  - https://twitter.com/certbe
  - https://www.linkedin.com/company/centre-for-cybersecurity-belgium/mycompany/
- Extra source from our partner NCSC-NL:
  - https://github.com/NCSC-NL/log4shell
  - Includes Threat hunting info, identification and mitigation techniques, known software list
  - Includes a collection of IOC lists

Questions?

Updates?

Find extra updates on our advisory on https://cert.be/en/warning-active-exploitation-0-day-rce-log4j

Follow-up questions?

Send a mail to them to ews@cert.be

Intrusion identified?

Please report to us via:

cert@cert.be

https://cert.be/en/report-incident

NOTES

- Slide 5
  - https://www.lunasec.io/docs/blog/log4j-zero-day/
  - https://nvd.nist.gov/vuln/detail/CVE-2021-44228
  - https://www.microsoft.com/security/blog/2021/12/11/guidance-for-preventing-detecting-and-hunting-for-cve-2021-44228-log4j-2-exploitation/#nation-state
  - https://blog.netlab.360.com/ten-families-of-malicious-samples-are-spreading-using-the-log4j2-vulnerability-now/
  - https://businessinsights.bitdefender.com/technical-advisory-zero-day-critical-vulnerability-in-log4j2-exploited-in-the-wild
  - https://thehackernews.com/2021/12/hackers-exploit-log4j-vulnerability-to.html
- Slide 6: https://news.sophos.com/en-us/2021/12/12/log4shell-hell-anatomy-of-an-exploit-outbreak/
- Slide 7: https://nvd.nist.gov/vuln/detail/CVE-2021-45046
- Slide 8 & 9
  - 26/11: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-44228, https://nvd.nist.gov/vuln/detail/CVE-2021-44228
  - 01/12: https://therecord.media/log4shell-attacks-began-two-weeks-ago-cisco-and-cloudflare-say/, https://twitter.com/eastdakota/status/1469800951351427073, https://blog.talosintelligence.com/2021/12/apache-log4j-rce-vulnerability.html
  - 09/12: https://www.lunasec.io/docs/blog/log4j-zero-day/, https://github.com/tangxiaofeng7/apache-log4j-poc
  - 10/12: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-4104
  - 14-15/12: https://github.com/apache/logging-log4j2/tags
- Slide 10
  - https://www.bleepingcomputer.com/news/security/log4j-list-of-vulnerable-products-and-vendor-advisories/
  - https://github.com/cisagov/log4j-affected-db
  - https://github.com/NCSC-NL/log4shell
  - https://www.techsolvency.com/story-so-far/cve-2021-44228-log4j-log4shell/
- Slide 11: https://www.lunasec.io/docs/blog/log4j-zero-day-mitigation-guide/
- Slide 12
  - https://logging.apache.org/log4j/2.x/
  - https://www.govcert.ch/blog/zero-day-exploit-targeting-popular-java-library-log4j/