# BlackTech Updates Elf-Plead Backdoor

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 (.dynsym) 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.

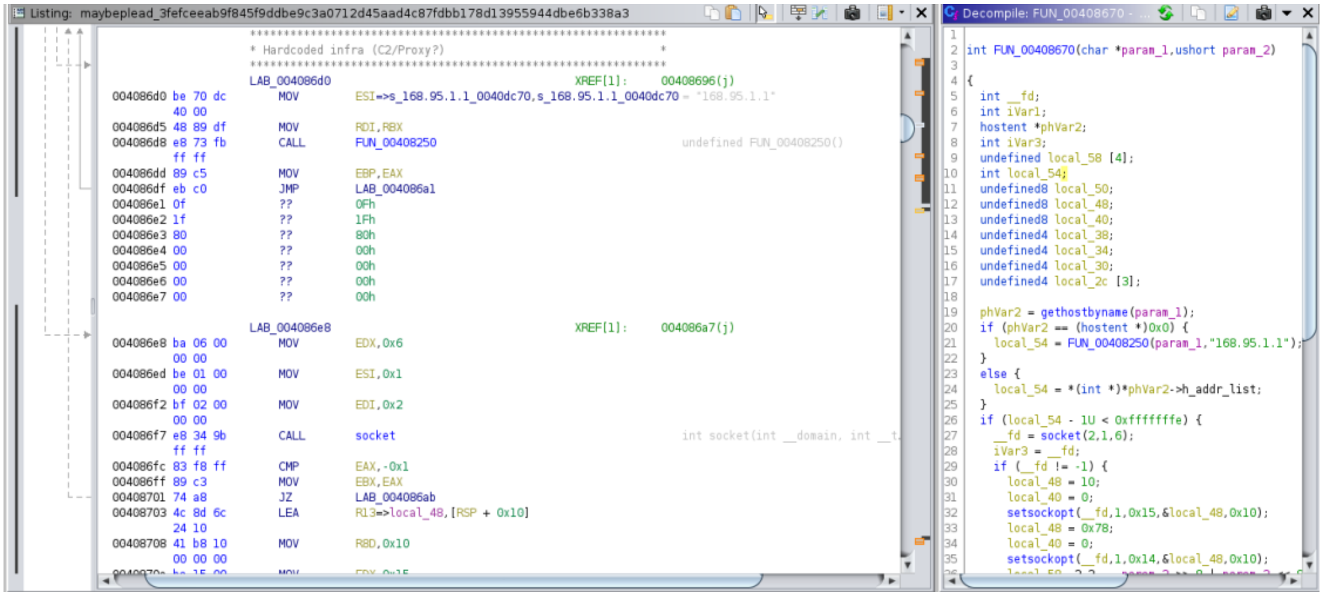

Figure 2 Hardcoded C2 IPv4 address

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.

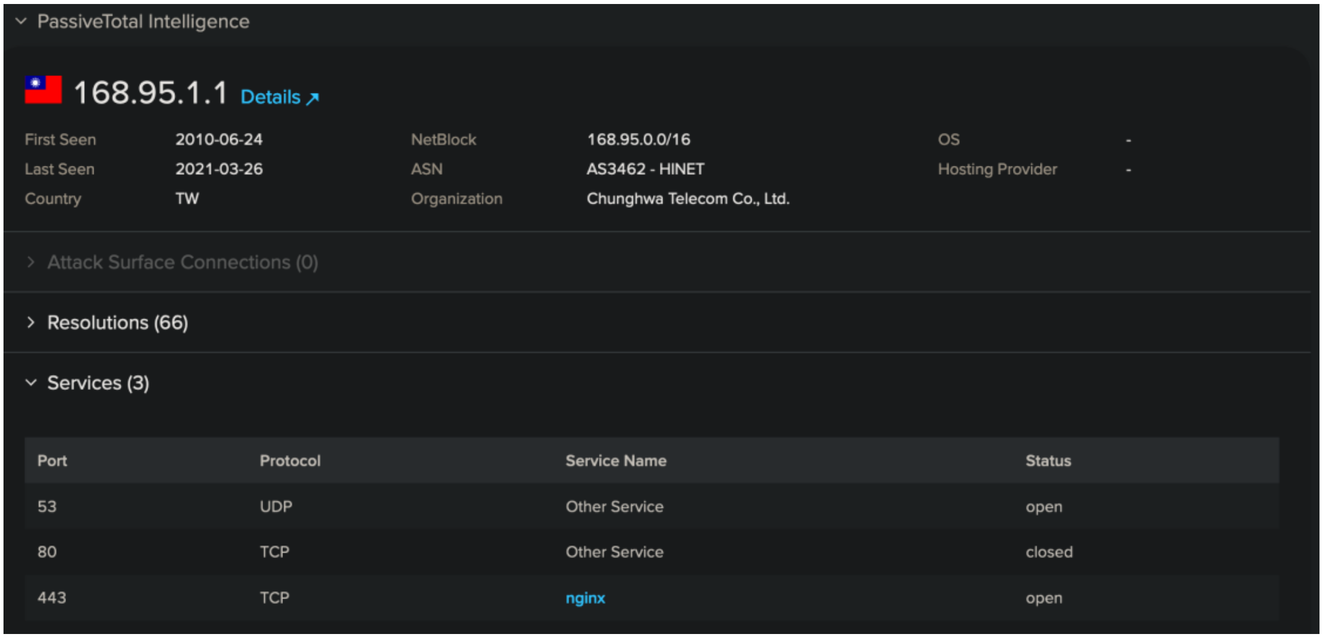

Figure 3

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.

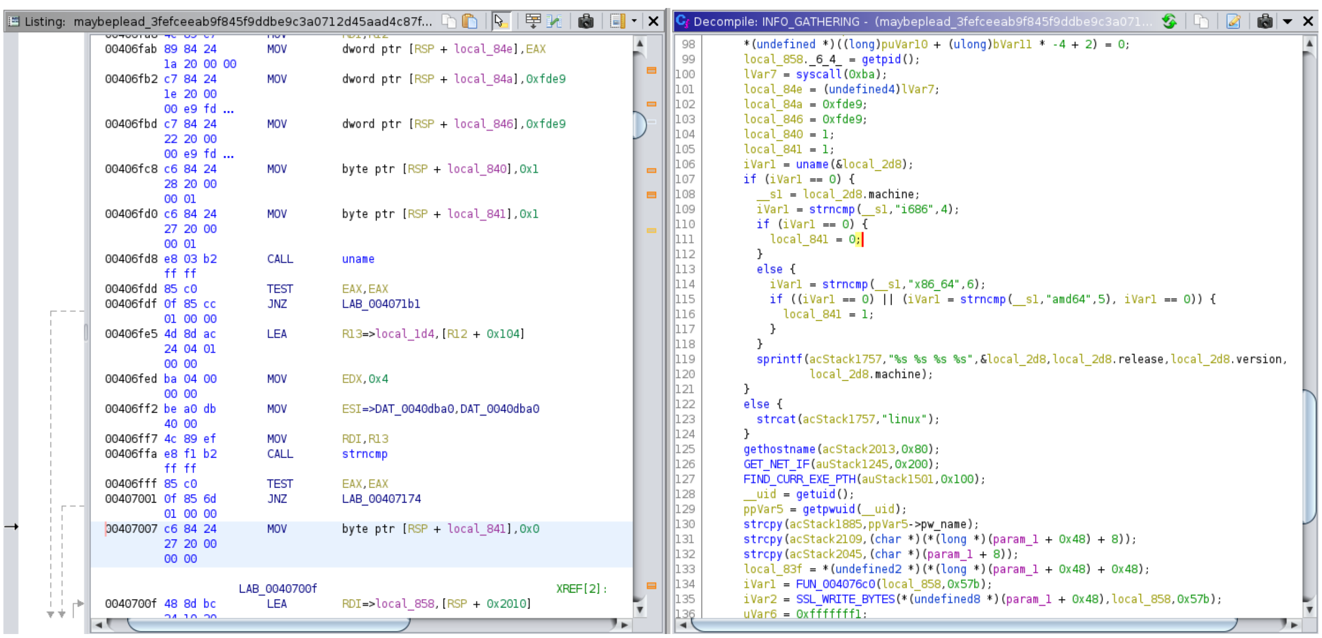
Figure 4

## 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.

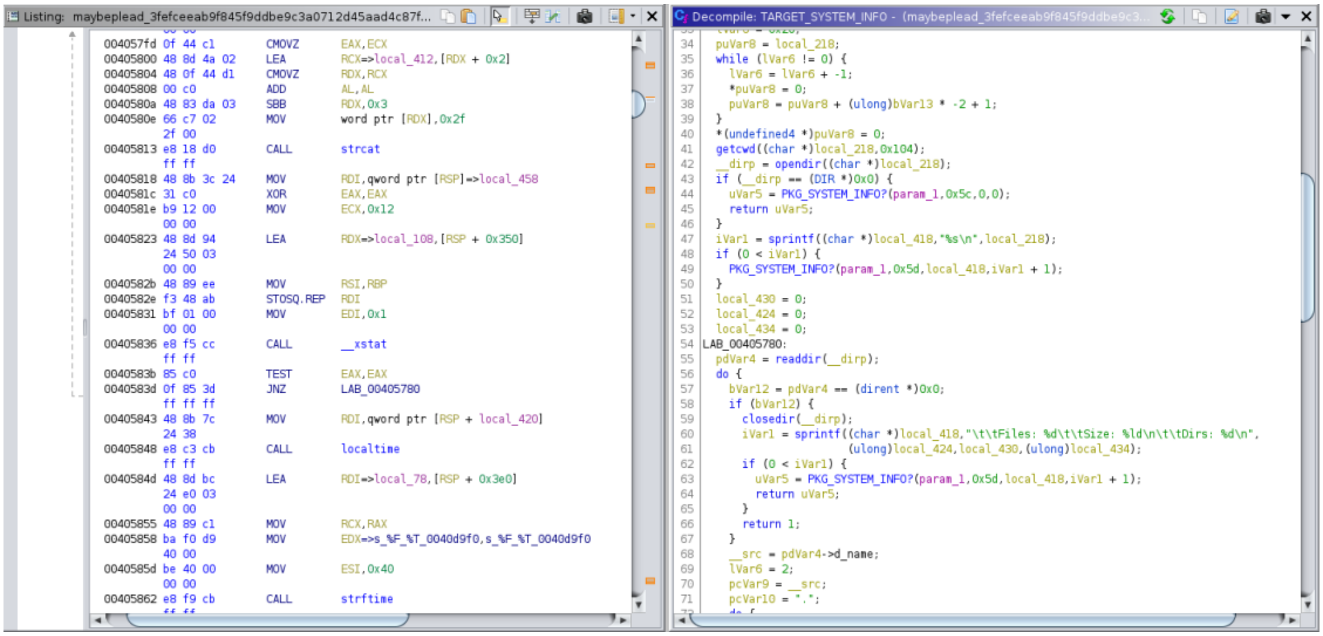

Figure 5

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: 3fefceeab9f845f9ddbe9c3a0712d45aad4c87fdbb178d13955944dbe6b338a3

IP: 168.95.1[.]1

## References

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

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

[3] https://blogs.jpcert.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/