JPCERT Coordination Center official Blog

blogs.jpcert.or.jp/en/2022/03/anti_upx_unpack.html

朝長 秀誠 (Shusei Tomonaga)

March 15, 2022

Anti-UPX Unpacking Technique

IoT

-
- Email

Malware targeting Windows OS (PE format) has a variety of obfuscation and packing techniques in place so that they complicate the code analysis processes. On the other hand, there are only a few types of packing techniques for Linux-targeting malware (ELF format), and it is mainly UPX-based. This blog article explains the details of Anti-UPX Unpacking technique, which is often applied to Linux-targeting malware.

Malware with Anti-UPX Unpacking Technique

The most well-known malware using Anti-UPX Unpacking technique is Mirai and its variants, which target IoT devices. Figure 1 shows the headers of UPX-packed binary and Mirai. The normal UPX packing uses “UPX!” as a magic number, while Mirai assigns a different value to each sample.

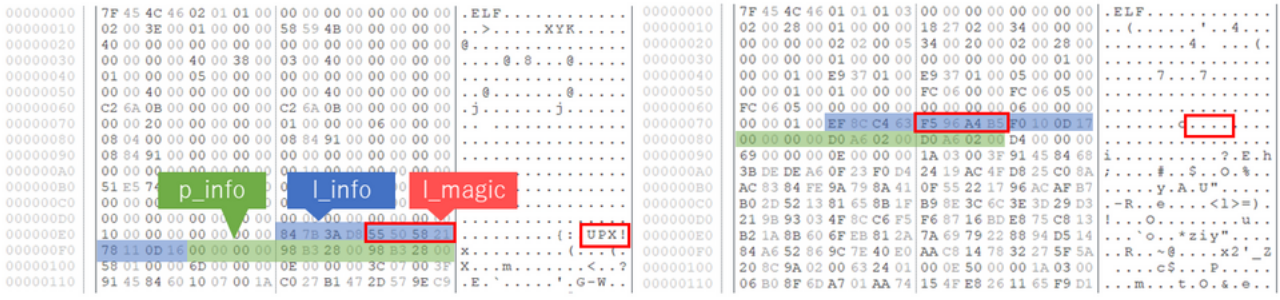


Figure 1: UPX-packed binary (left) and Mirai header (right)

UPX-packed binary contains the following information in the header. Normally, only “l_magic” is altered, but “p_filesize” and “p_blocksize” are also zero-padded in some samples.

```
struct l_info       // 12-byte trailer in header for loader (offset 116)
{
    uint32_t l_checksum;
    uint32_t l_magic;    // magic number = "UPX!"
    uint16_t l_lsize;
    uint8_t  l_version;
    uint8_t  l_format;
};

struct p_info       // 12-byte packed program header follows stub loader
{
    uint32_t p_progid;
    uint32_t p_filesize;
    uint32_t p_blocksize;
};
```

Besides Mirai, there are many other types of malware using this technique, including BoSSaBot (seen around 2014), as well as some coin miners and SBIDIOT malware, more recently. This is also applied to some types of malware which were used by Lazarus group. Figure 2 shows a part of ELF-VSingle's code, which is associated with the group. The magic number is replaced with "MEMS".

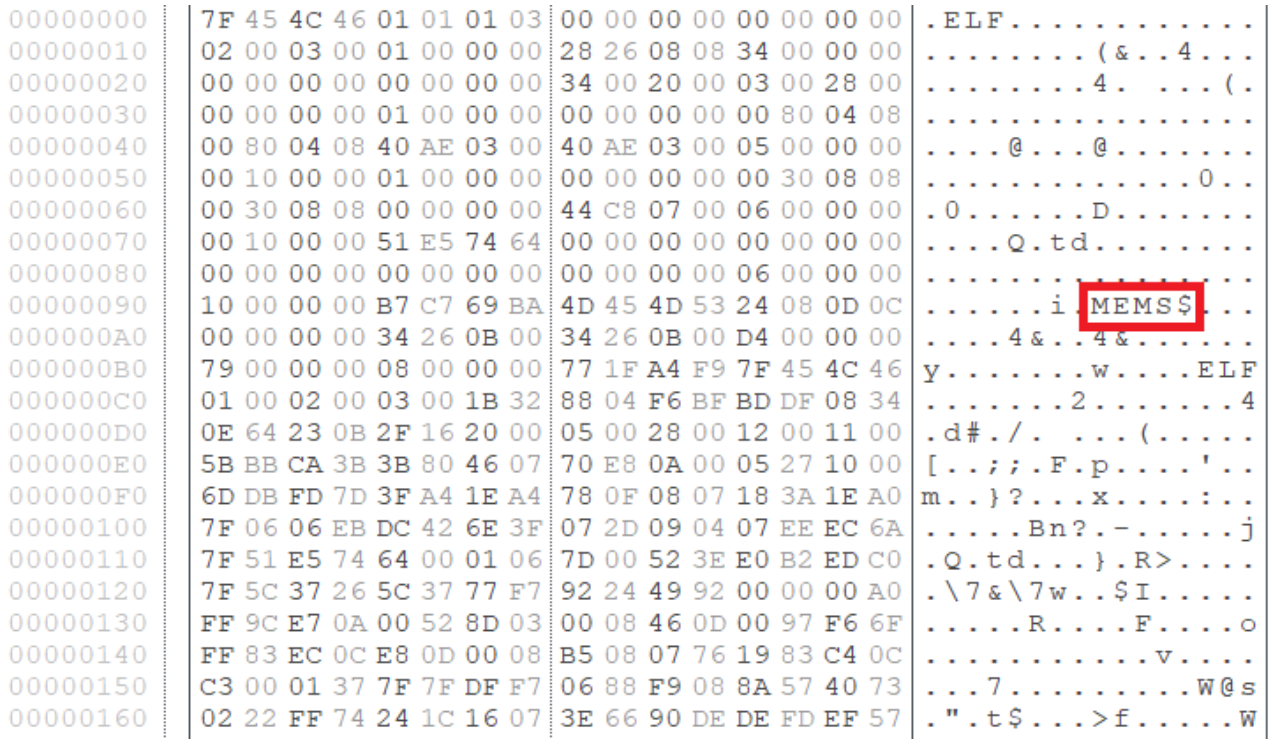


Figure 2: ELF_VSingle header

Unpacking Anti-UPX Unpacking binary

Binary based on Anti-UPX Unpacking technique cannot be unpacked using the normal upx command. However, it is actually easy to unpack it. In most cases, the only change made to such binary is its magic number "UPX!". You can unpack it with upx command by changing

this value back to “UPX!”. Figure 3 shows the process of changing the magic number in order to unpack it using upx command.

```
                /cygdrive/c/data/upx$ xxd sample |head
00000000: 7f45 4c46 0101 0103 0000 0000 0000 0000  .ELF............
00000010: 0200 2800 0100 0000 1827 0200 3400 0000  ..(......'..4...
00000020: 0000 0000 0202 0005 3400 2000 0200 2800  ........4. ...(.
00000030: 0000 0000 0100 0000 0000 0000 0000 0100  ................
00000040: 0000 0100 e937 0100 e937 0100 0500 0000  .....7...7......
00000050: 0000 0100 0100 0000 fc06 0000 fc06 0500  ................
00000060: fc06 0500 0000 0000 0000 0000 0600 0000  ................
00000070: 0000 0100 ef8c c463 f596 a4b5 f010 0d17  .......c........
00000080: 0000 0000 d0a6 0200 d0a6 0200 d400 0000  ................
00000090: 6900 0000 0e00 0000 1a03 003f 9145 8468  i..........?.E.h
                /cygdrive/c/data/upx$ xxd sample |tail -n 4
00013960: abee a5ef 0000 0000 00f5 96a4 b500 0000  ................
00013970: 0000 0000 f596 a4b5 0d17 0e0a 0a81 5265  ..............Re
00013980: 2d24 2d4a 9c03 0000 5101 0000 d0a6 0200  -$-J....Q.......
00013990: 5000 0018 8000 0000                      P.......
                /cygdrive/c/data/upx$ upx -d sample
                      Ultimate Packer for eXecutables
                         Copyright (C) 1996 - 2020
UPX 3.96w       Markus Oberhumer, Laszlo Molnar & John Reiser   Jan 23rd 2020

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
upx: sample: NotPackedException: not packed by UPX

Unpacked 0 files.
                /cygdrive/c/data/upx$ vi -b sample
                /cygdrive/c/data/upx$ xxd sample |head
00000000: 7f45 4c46 0101 0103 0000 0000 0000 0000  .ELF............
00000010: 0200 2800 0100 0000 1827 0200 3400 0000  ..(......'..4...
00000020: 0000 0000 0202 0005 3400 2000 0200 2800  ........4. ...(.
00000030: 0000 0000 0100 0000 0000 0000 0000 0100  ................
00000040: 0000 0100 e937 0100 e937 0100 0500 0000  .....7...7......
00000050: 0000 0100 0100 0000 fc06 0000 fc06 0500  ................
00000060: fc06 0500 0000 0000 0000 0000 0600 0000  ................
00000070: 0000 0100 ef8c c463 5550 5821 f010 0d17  .......cUPX!....
00000080: 0000 0000 d0a6 0200 d0a6 0200 d400 0000  ................
00000090: 6900 0000 0e00 0000 1a03 003f 9145 8468  i..........?.E.h
                /cygdrive/c/data/upx$ xxd sample |tail -n 4
00013960: abee a5ef 0000 0000 0055 5058 2100 0000  .........UPX!...
00013970: 0000 0000 5550 5821 0d17 0e0a 0a81 5265  ....UPX!......Re
00013980: 2d24 2d4a 9c03 0000 5101 0000 d0a6 0200  -$-J....Q.......
00013990: 5000 0018 8000 0000                      P.......
                /cygdrive/c/data/upx$ upx -d sample
                      Ultimate Packer for eXecutables
                         Copyright (C) 1996 - 2020
UPX 3.96w       Markus Oberhumer, Laszlo Molnar & John Reiser   Jan 23rd 2020

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
    173776 <-     80280   46.20%    linux/arm    sample

Unpacked 1 file.
```

Figure 3: Example of unpacking binary

We have created a tool that enables unpacking binary with Anti-UPX Unpacking techniques. This tool is intended for this purpose only, and it may not work otherwise.

JPCERTCC/upx-mod - GitHub https://github.com/JPCERTCC/upx-mod/releases/tag/v4.00-beta

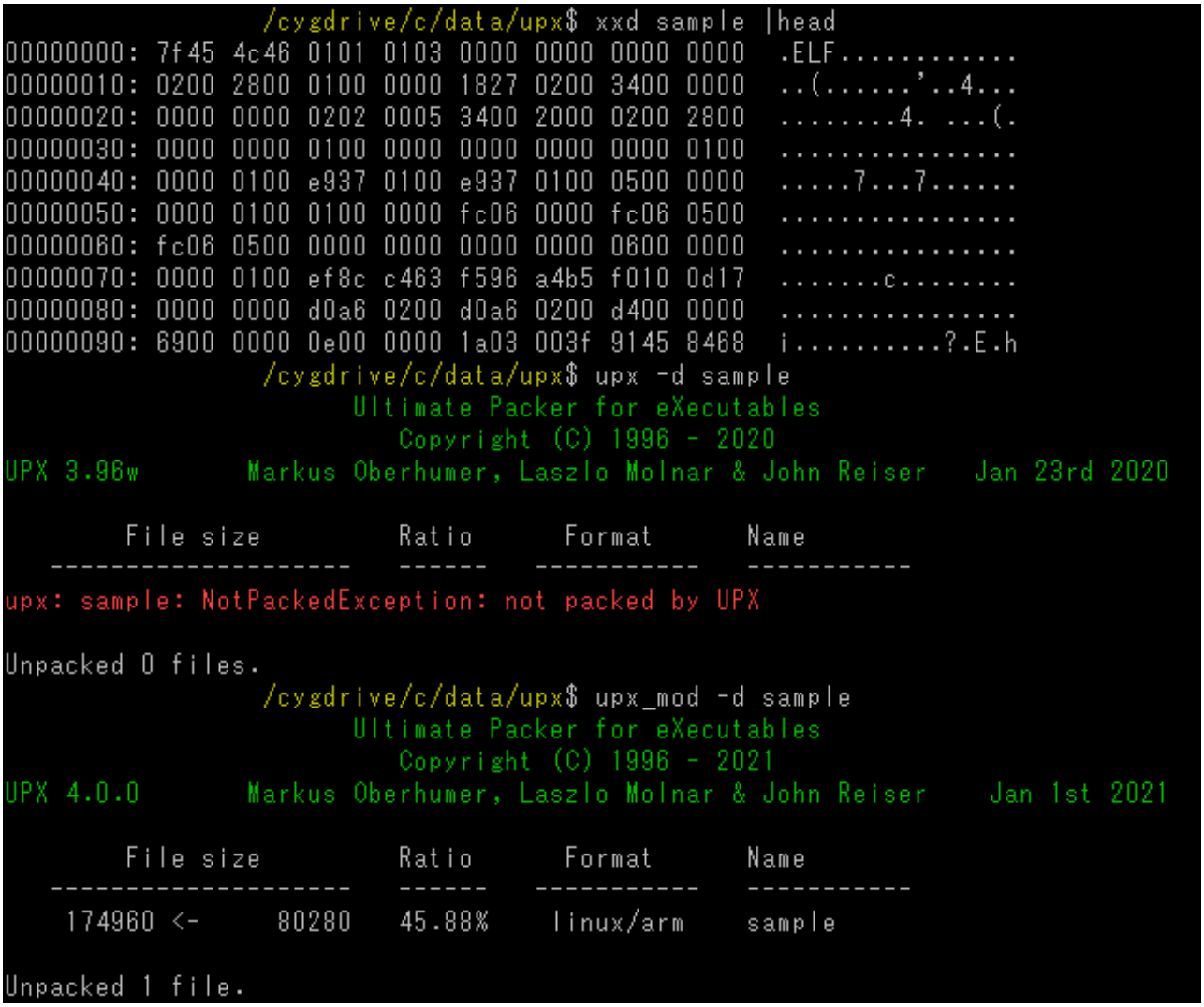


Figure 4: Sample use of upx_mod command

Detect Anti-UPX Unpacking Technique

Binary packed with this technique can be identified manually, just by looking at the code. In order to avoid oversight, we recommend Yara-based automatic detection like below. This rule does not detect binary packed with normal UPX.

```
rule upx_antiunpack_elf32 {
     meta:
        description = "Anti-UPX Unpacking technique to magic renamed for ELF32"
        author = "JPCERT/CC Incident Response Group"

     condition:
       uint32(0) == 0x464C457F and
       uint8(4) == 1 and
       (
         (
           for any magic in (uint32(filesize - 0x24)) : (magic == uint32(uint16(0x2C)
* uint16(0x2A) + uint16(0x28) + 4)) and
           not for any magic in (0x21585055, 0) : (magic == uint32(uint16(0x2C) *
uint16(0x2A) + uint16(0x28) + 4))
         )
         or
         (
           for any magic in (uint32(filesize - 0x24)) : (magic ==
uint32(uint16be(0x2C) * uint16be(0x2A) + uint16be(0x28) + 4)) and
           not for any magic in (0x21585055, 0) : (magic == uint32(uint16be(0x2C) *
uint16be(0x2A) + uint16be(0x28) + 4))
         )
       )
}

rule upx_antiunpack_elf64 {
     meta:
        description = "Anti-UPX Unpacking technique to magic renamed for ELF64"
        author = "JPCERT/CC Incident Response Group"

     condition:
       uint32(0) == 0x464C457F and
       uint8(4) == 2 and
       (
         (
           for any magic in (uint32(filesize - 0x24)) : (magic == uint32(uint16(0x36)
* uint16(0x38) + uint16(0x34) + 4)) and
           not for any magic in (0x21585055, 0) : (magic == uint32(uint16(0x36) *
uint16(0x38) + uint16(0x34) + 4))
         )
         or
         (
           for any magic in (uint32(filesize - 0x24)) : (magic ==
uint32(uint16be(0x36) * uint16be(0x38) + uint16be(0x34) + 4)) and
           not for any magic in (0x21585055, 0) : (magic == uint32(uint16be(0x36) *
uint16be(0x38) + uint16be(0x34) + 4))
         )
       )
}
```

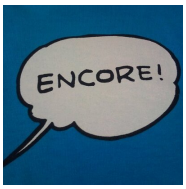
In closing

Many attack groups use malware based on Anti-UPX Unpacking technique. It is easy to unpack such malware, but you may waste your time in unpacking process unless you notice this feature beforehand. When you analyse packed ELF binary, we recommend checking first whether it uses Anti-UPX Unpacking technique.

Shusei Tomonaga
(Translated by Yukako Uchida)

-
- Email

Author



朝長 秀誠 (Shusei Tomonaga)

Since December 2012, he has been engaged in malware analysis and forensics investigation, and is especially involved in analyzing incidents of targeted attacks. Prior to joining JPCERT/CC, he was engaged in security monitoring and analysis operations at a foreign-affiliated IT vendor. He presented at CODE BLUE, BsidesLV, BlackHat USA Arsenal, Botconf, PacSec and FIRST Conference. JSAC organizer.

Was this page helpful?

0 people found this content helpful.

If you wish to make comments or ask questions, please use this form.

This form is for comments and inquiries. For any questions regarding specific commercial products, please contact the vendor.

please change the setting of your browser to set JavaScript valid. Thank you!

Back
Top
Next