New Zero-Detection Variant of Melofee Backdoor from Winnti Strikes RHEL 7.9

Alex.Turing ⁝⁝ 11/12/2024

Background

On July 27, 2024, `XLab's Cyber Threat Insight and Analysis System(CTIA)` detected an ELF file named *pskt* from IP address 45.92.156.166. Currently undetected on VirusTotal, the file triggered two alerts: an Overlay section and a communication domain mimicking Microsoft. Our analysis identified it as a Melofee backdoor variant, specifically targeting Red Hat Enterprise Linux (RHEL) 7.9.

Melofee, a C++ backdoor, enables data collection, process management, file handling, and shell access. Originally exposed by ExaTrack in March 2023 and attributed to the **APT group Winnti**, this latest variant has notable upgrades. Structurally, it embeds an RC4-encrypted kernel driver to mask traces of files, processes, and network connections. Functionally, it adds improvements in persistence, single-instance control, and function ID design.

By examining the sample's Run-Time Type Information (RTTI), we observed source-level modifications. For instance, the network connection class name has changed from `TLSSocket` in earlier samples to `TlsConn` in this variant, suggesting ongoing reconstruction and use of Melofee beyond the security community's radar.

Notably, during our investigation, we encountered an intriguing **misattribution**. The new variant utilizes the C2 address `filemanage.micrsofts-file.com`. According to Passive DNS (PDNS) records, this C2's second-level domain, `micrsofts-file.com` and its associated domain, `www.micrsofts-file.com` resolved to IP address `91.195.240.123` between November 2023 and June 2024. This IP also appeared in BlackBerry's July 2024 report on the **APT group SideWinder** and has been flagged as malicious by several security vendors on VirusTotal. Does this imply that Melofee has circulated among multiple organizations, becoming a cross-group tool rather than being exclusive to a single group?

We believe this is unlikely. The IP address `91.195.240.123` is a parking IP provided by domain registrar NameSilo. **Labeling it as malicious likely constitutes a false positive**. NameSilo automatically resolves new registered second-level domains and "www" subdomains to this IP, leading to potential misattributions, as legitimate domains, unrelated malicious domains, and APT activities may all share this IP.

Due to limited visibility, we currently lack details on the attacker’s entry methods and goals. We invite others to share insights to enrich the technical landscape. Given the low detection rate of this sample and Melofee’s stealth, we’re sharing these findings with the community for broader cybersecurity awareness.

This report covers:

- Overlay structure and decryption method
- Driver module’s functionalities
- Melofee’s capabilities

Technical Details

We have captured a single sample with the following details:

```
MD5: 603e38a59efcf6790f2b4593edb9faf5
Magic: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically
linked, BuildID[sha1]=48bcb3f7c78bc746e25264058a76145b63bbf440, for
GNU/Linux 3.2.0, stripped
```

This variant operates in two modes based on launch parameters: **Infection Mode** and **Management Mode**.

- **Infection Mode** (No Parameters)
  When launched without parameters, Melofee enters Infection Mode, performing the following:
    - Enforces single instance via `/tmp/lock_tmp1`
    - Achieves persistence via `crontab`, disguising the process name as `[md]` or `wwwwww`
    - Decrypts and installs a driver module for stealth across files, processes, network connections, and directories
    - Decrypts and connects to its C2 server, awaiting commands

- **Management Mode** (With Parameters)
  When launched with parameters, Melofee enters Management Mode, controlling driver hiding functionality:
    - **hide**: Activates hiding features
    - **show**: Deactivates hiding
    - **kill**: Terminates the process

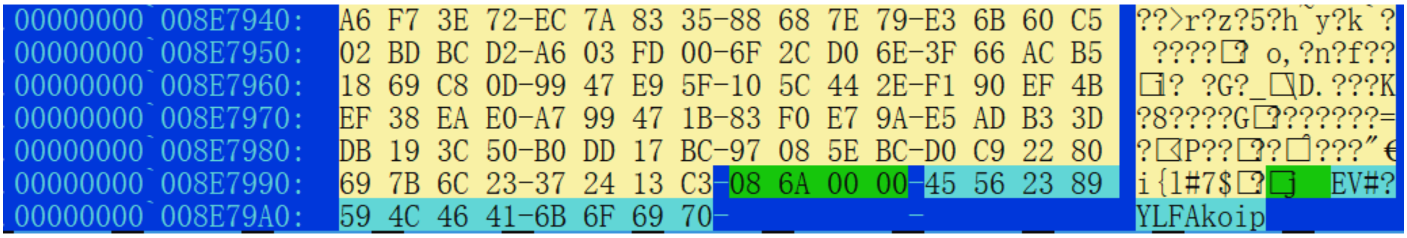
This design enables flexible operation across infection and management needs. The next sections will cover Melofee’s decryption, driver module, and backdoor functions in detail.

Part 1: Decryption

Melofee stores its RC4-encrypted driver module as an overlay appended to the file’s end, using a structure called `drv_overlay`:

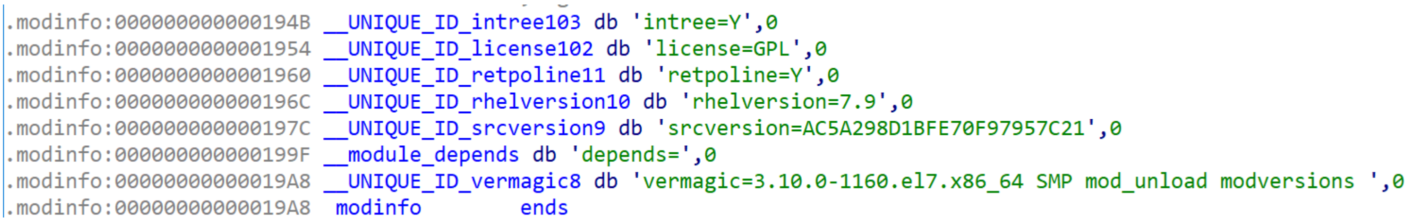
```
struct drv_overlay {
    int encrypted_payload[payload_size];
    int payload_size;
    char flag[12];
}
```

In this sample, the `flag` is set to "EV#?YLFAkoip" and `payload_size` is `0x6a08`. The `encrypted_payload` spans `0x6a08` bytes backward from `payload_size`.



```
00000000`008E7940: A6 F7 3E 72-EC 7A 83 35-88 68 7E 79-E3 6B 60 C5  ??>r?z?5?h`y?k`?
00000000`008E7950: 02 BD BC D2-A6 03 FD 00-6F 2C D0 6E-3F 66 AC B5   ????□? o,?n?f??
00000000`008E7960: 18 69 C8 0D-99 47 E9 5F-10 5C 44 2E-F1 90 EF 4B  □i? ?G?_□\D.???K
00000000`008E7970: EF 38 EA E0-A7 99 47 1B-83 F0 E7 9A-E5 AD B3 3D  ?8????G□???????=
00000000`008E7980: DB 19 3C 50-B0 DD 17 BC-97 08 5E BC-D0 C9 22 80  ?□P??□??□???"€
00000000`008E7990: 69 7B 6C 23-37 24 13 C3-08 6A 00 00-45 56 23 89  i{l#7$□?□j  EV#?
00000000`008E79A0: 59 4C 46 41-6B 6F 69 70-                         YLFAkoip
```

Using the key `87JoENDi`, the `encrypted_payload` is decrypted to reveal the driver module `kworkerx`, designed for RHEL 7.9 with kernel version 3.10.0.



```
.modinfo:000000000000194B __UNIQUE_ID_intree103 db 'intree=Y',0
.modinfo:0000000000001954 __UNIQUE_ID_license102 db 'license=GPL',0
.modinfo:0000000000001960 __UNIQUE_ID_retpoline11 db 'retpoline=Y',0
.modinfo:000000000000196C __UNIQUE_ID_rhelversion10 db 'rhelversion=7.9',0
.modinfo:000000000000197C __UNIQUE_ID_srcversion9 db 'srcversion=AC5A298D1BFE70F97957C21',0
.modinfo:000000000000199F __module_depends db 'depends=',0
.modinfo:00000000000019A8 __UNIQUE_ID_vermagic8 db 'vermagic=3.10.0-1160.el7.x86_64 SMP mod_unload modversions ',0
.modinfo:00000000000019A8  modinfo         ends
```

The C2 configuration is also RC4-encrypted, using the same key `87JoENDi`.

Encrypted C2 Data:

```
00000000  a2 a4 96 0e 27 ee 40 54  a5 3a 52 8e 65 cf b1 e1  |¢¤..'î@T¥:R.eÏ±á|
00000010  29 69 32 86 ae 56 4d 28  a2 b8 da 6e e1 05 5d 65  |)i2.®VM(¢¸Úná.]e|
00000020  fc 86 88 50 43 17                                 |ü..PC.|
```

Decrypted C2 Configuration:

```
0:filemanage.micrsofts-file.com:443:60
```

This configuration includes the following elements:

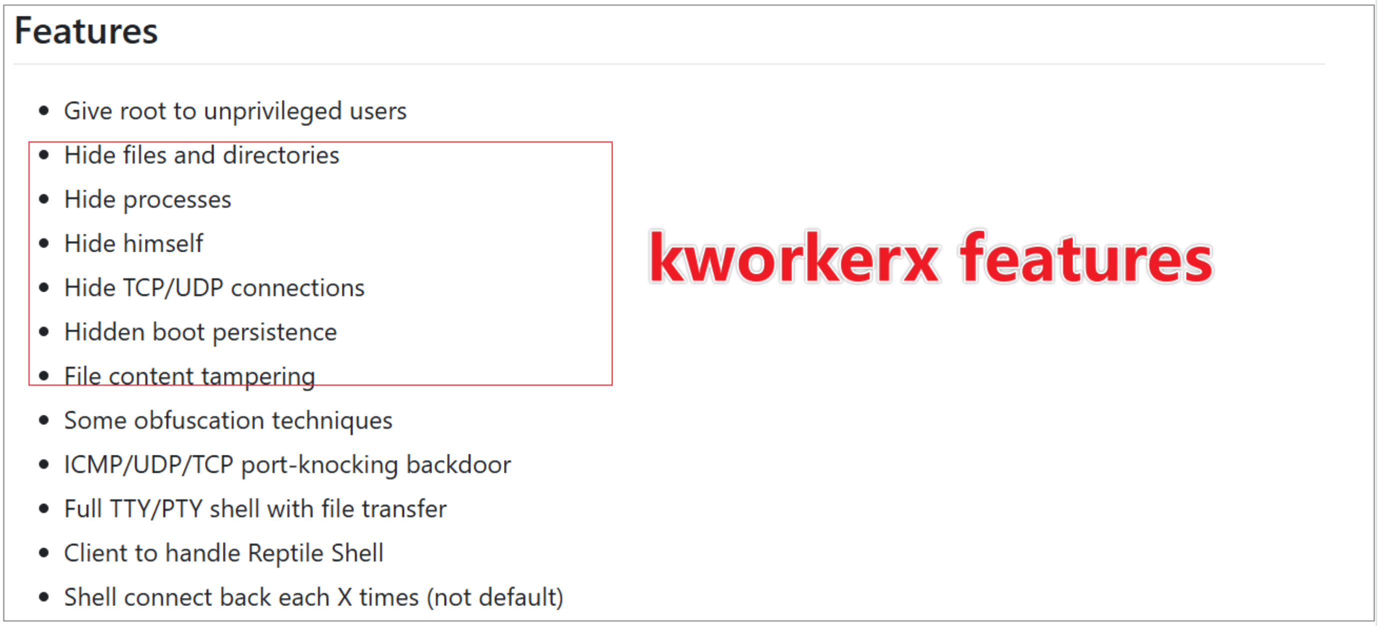
- Connection Type
- C2 Domain
- C2 Port
- Interval

Part 2: Driver Module Analysis

The decrypted driver module, `kworkerx`, has the following basic information:

```
MD5: 839f60efee25f07df7b23ba9d6bef892
Magic: ELF 64-bit LSB relocatable, x86-64, version 1 (SYSV),
BuildID[sha1]=c440028449ebce5c899a51ef0eb4d7fc43493253, not stripped
```

Through analysis, we confirmed that kworkerx is a modified version of the open-source project *Reptile*. The original Reptile project supports 12 functions, categorized into two main types: *hiding* and *backdoor* capabilities. kworkerx primarily utilizes the hiding functions.



Features

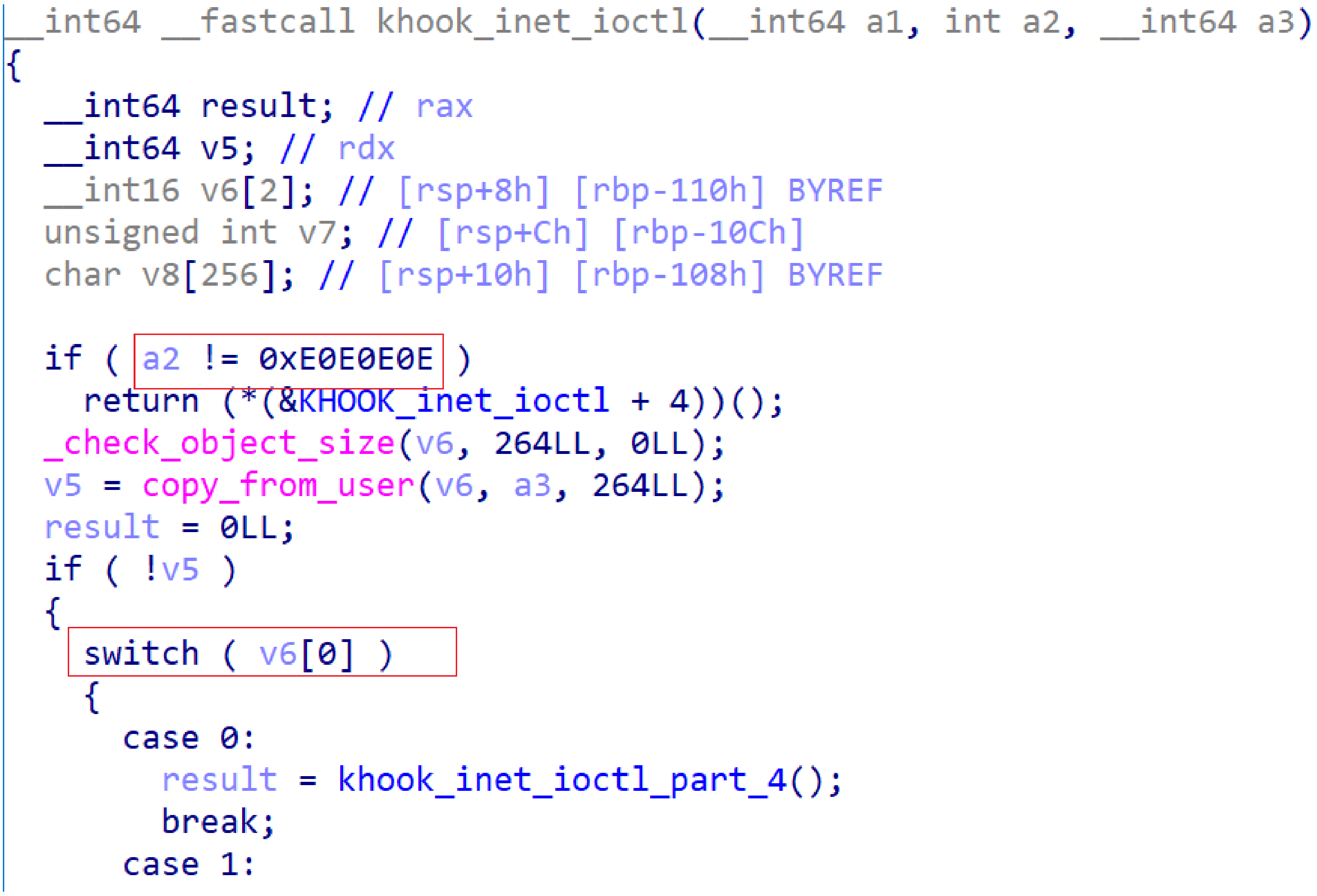
- Give root to unprivileged users
- Hide files and directories
- Hide processes
- Hide himself
- Hide TCP/UDP connections
- Hidden boot persistence
- File content tampering
- Some obfuscation techniques
- ICMP/UDP/TCP port-knocking backdoor
- Full TTY/PTY shell with file transfer
- Client to handle Reptile Shell
- Shell connect back each X times (not default)

Hiding Mechanisms in `kworkerx`

- **Network Communication Hiding**: kworkerx hooks the tcp4_seq_show function within its initialization routine, effectively hiding all network traffic on port 443.
- **File, Process, and Directory Hiding**: To conceal files, processes, and directories, kworkerx hooks several functions, including fillonedir, filldir, filldir64, and vfs_read.

Communication with User Space

kworkerx also hooks the inet_ioctl function to facilitate communication with user-space applications and receive control commands.



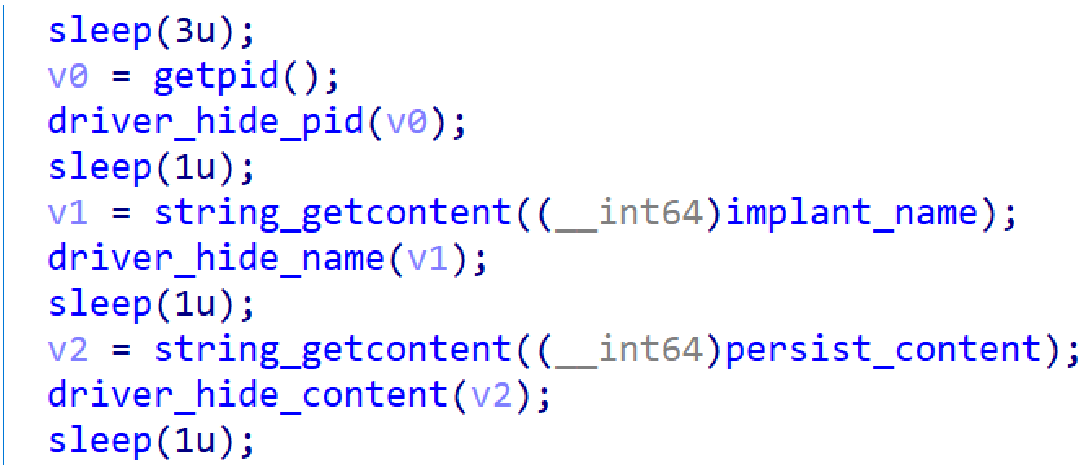
```
__int64 __fastcall khook_inet_ioctl(__int64 a1, int a2, __int64 a3)
{
  __int64 result; // rax
  __int64 v5; // rdx
  __int16 v6[2]; // [rsp+8h] [rbp-110h] BYREF
  unsigned int v7; // [rsp+Ch] [rbp-10Ch]
  char v8[256]; // [rsp+10h] [rbp-108h] BYREF

  if ( a2 != 0xE0E0E0E )
    return (*(&KHOOK_inet_ioctl + 4))();
  _check_object_size(v6, 264LL, 0LL);
  v5 = copy_from_user(v6, a3, 264LL);
  result = 0LL;
  if ( !v5 )
  {
    switch ( v6[0] )
    {
      case 0:
        result = khook_inet_ioctl_part_4();
        break;
      case 1:
```

When a user-space application calls the `ioctl` function with the second parameter set to `0xE0E0E0E`, it triggers the handler function `khook_inet_ioctl` in `kworkerx`. Within this function, `kworkerx` interprets the third parameter to either enable or disable specific hiding functions, providing fine-grained control over its concealment capabilities.

Arg.cmd	Capability
0	show all
1	hide all
2	hide proc
3	show proc
5	file tampering
7	hide file,dir
8	unhide_chdir
9	hide_chdir

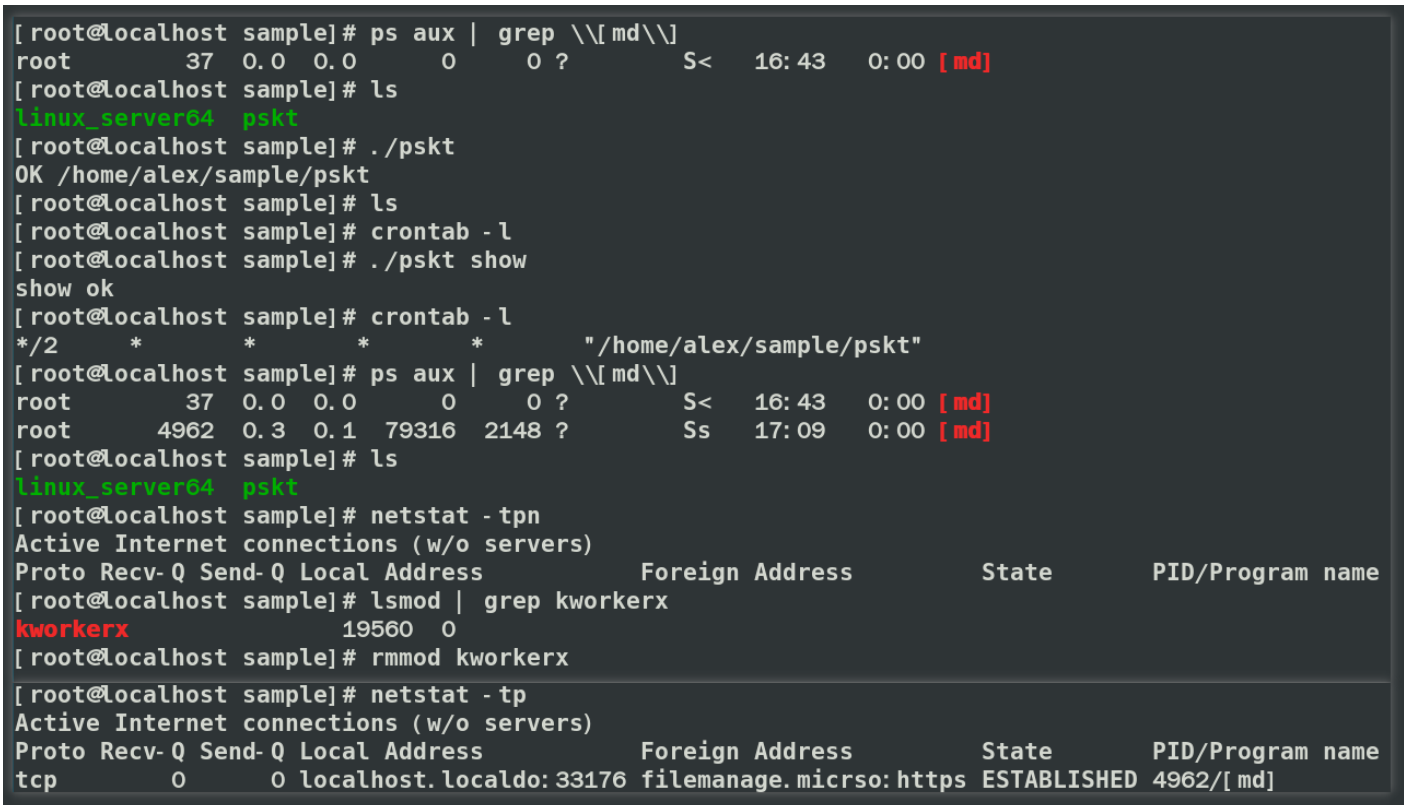
Part 3: Melofee Analysis

After installing the `kworkerx` kernel driver module via the `init_module` function, Melofee enables TCP connection hiding by default. Additional hiding features, such as process, directory, and persistence concealment, are activated through control commands sent via IOCTL.



```
sleep(3u);
v0 = getpid();
driver_hide_pid(v0);
sleep(1u);
v1 = string_getcontent((__int64)implant_name);
driver_hide_name(v1);
sleep(1u);
v2 = string_getcontent((__int64)persist_content);
driver_hide_content(v2);
sleep(1u);
```

When executed without parameters in a virtual machine, Melofee successfully concealed its process, the sample file, the persistence script, and network connections. Running the sample again with the `show` parameter revealed the process, sample file, and persistence script, while the network connection remained hidden. Finally, using the `rmmod` command to unload the `kworkerx` module restored visibility to the previously hidden network connection.



```
[root@localhost sample]# ps aux | grep \\[md\\]
root        37  0.0  0.0      0     0 ?        S<   16:43   0:00 [md]
[root@localhost sample]# ls
linux_server64  pskt
[root@localhost sample]# ./pskt
OK /home/alex/sample/pskt
[root@localhost sample]# ls
[root@localhost sample]# crontab -l
[root@localhost sample]# ./pskt show
show ok
[root@localhost sample]# crontab -l
*/2     *       *       *       *       "/home/alex/sample/pskt"
[root@localhost sample]# ps aux | grep \\[md\\]
root        37  0.0  0.0      0     0 ?        S<   16:43   0:00 [md]
root      4962  0.3  0.1  79316  2148 ?        Ss   17:09   0:00 [md]
[root@localhost sample]# ls
linux_server64  pskt
[root@localhost sample]# netstat -tpn
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
[root@localhost sample]# lsmod | grep kworkerx
kworkerx               19560  0
[root@localhost sample]# rmmod kworkerx
[root@localhost sample]# netstat -tp
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 localhost.localdo:33176 filemanage.micrso:https ESTABLISHED 4962/[md]
```

After installing the driver module, Melofee decrypts the C2 configuration and establishes communication, waiting to receive and execute commands. The functionality of this sample aligns with the description provided in the ExaTrack analysis report, though there are differences in function IDs.

CMD ID	Capability
0x11	uninstall
0x22	collect device info
0x33	launch new command thread
0x34	write file
0x35	read file

CMD ID	Capability
0x36	create new socket
0x37	list directory
0x38	create directory
0x3a	delete directory
0x3b	create process to exec cmd
0x3c	exec command with output (including set new c2 ip)
0x3d	collect process info
0x3e	kill process
0x3f	launch shell
0x7b	ping back

Summary

Melofee offers straightforward functionality with highly effective stealth capabilities. Samples of this malware family are rare, suggesting that attackers may limit its use to high-value targets. Network administrators can check for infection by looking for artifacts like the `/tmp/lock_tmp1` file and the `kworkerx` module. If signs of infection are detected, follow the previous steps to remove associated drivers, processes, files, and persistence mechanisms.

We welcome readers to share additional insights and intelligence. If you're interested in our research, feel free to reach out to us via Platform X.

IOC

MD5

```
603e38a59efcf6790f2b4593edb9faf5 *pskt
839f60efee25f07df7b23ba9d6bef892 *kworkerx
```

C2

```
filemanage.micrsofts-file[.]com:443
```

Downloader

```
http://45.92.156[.]166/klove/pskt
```