Neutralizing Tofsee Spambot – Part 2 | InMemoryConfig store vaccine

spamhaus.com/resource-center/neutralizing-tofsee-spambot-part-2-inmemoryconfig-store-vaccine/

April 6, 2023

Here's the second in our three-part series focused on protecting against Tofsee malware. This spambot is prolific, but various vaccines and kill switches are available to defend against Tofsee. Our malware researchers are sharing two vaccines and a network-based kill switch in this series.

A recap

If you're wondering what malware vaccines are and how they can be utilized, or you'd like to read about the first vaccine our researchers have shared relating to Tofsee and its binary file, read this blog post. Alternatively, keep reading to learn about a second vaccine our team has produced, focused on polluting Tofsee's internal configuration store.

A deeper dive into Tofsee's config stores

During the runtime of Tofsee and the communications with its command and control (C&C) server, Tofsee stores various configuration values pertinent to the proper runtime of the code in a memory-based structure which we call the InMemoryConfig store. This is a circular linked list structure, and Tofsee defines it as follows:

```
struct __ConfigLinkedList_circular
{
    struct __ConfigLinkedList_circular  *next;
    int ConfigTTL; // Config Time to live
    int UN2;
    struct ConfigHeader{
        int ConfigType;
        char ConfigName[0x10];
        int Crc32Config;
        //StartConfig:  0x24
        DWORD Configlen; // Plus mask
        DWORD Unknonw1; // Maybe TIME_??
        DWORD Unknonw_Const3;
        }
    char Configvalue[Configlen];
};
```

InMemoryConfig store structure

Locations of Tofsee's configuration storage

Each **ConfigValue** buffer has its internal structure based on the **ConfigType** value. This chained config is dumped and stored in various locations on the infected system so Tofsee can retrieve it after a reboot.

The various configuration storage locations are:

File Storage
1 %USERPROFILE%\:.repos (ADS)
2 %USERPROFILE%\Local Settings:.repos
3 %USERPROFILE%\Local Settings\Application Data\Microsoft\Windows\UsrClass.dat.repos
4 %USERPROFILE%\wincookie.repos

Registry storage
1: HKEY_CURRENT_USER\\Control Panel\\Buses\\Config0
2: HKEY_CURRENT_USER\\SOFTWARE\\Microsoft\\Buses\\Config0

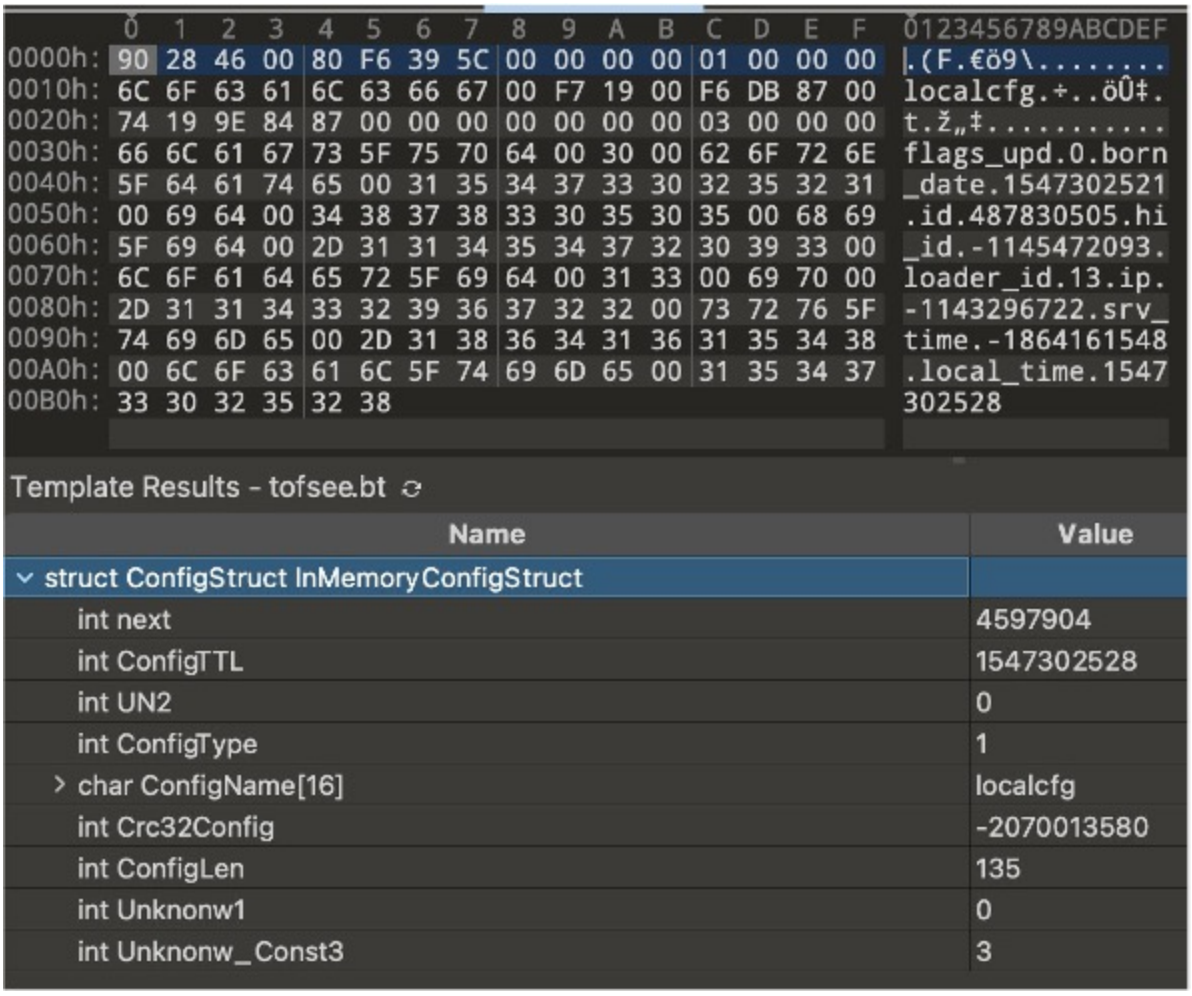
A simple Tofsee xor algorithm encodes the data stored in one of these places:

```
for ( i = 0; i < len; i++)
        {

                dst[i] = (buf[i] ^ key);

                key = x + adder + key;
                x = -x;
        }
```

Once retrieved and decoded, this data looks something like this in its raw parsed form:



The config stores of particular interest to us are the **work_srv** and **start_srv** structures. Both are retrieved during the initial C&C connection of the Tofsee botnet.

Tofsee's botnet C&C environment

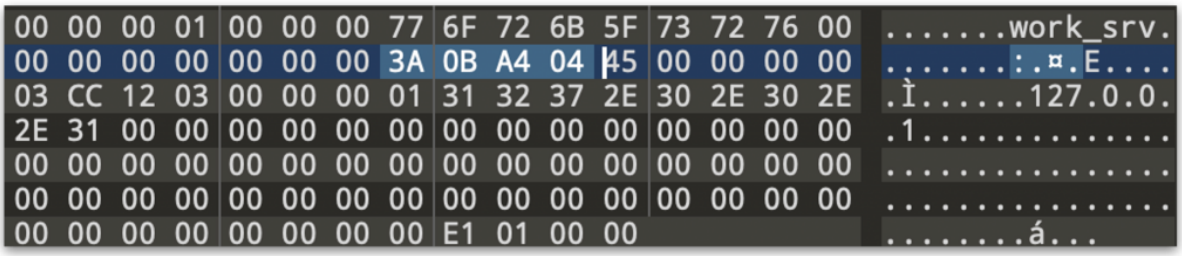
Tofsee has a tier-2 C&C ecosystem. The malware uses the hardcoded C&Cs in the binary only once to retrieve a list of tier-2 peers. These tier-2 piers then act as forwarding C&Cs and are stored in **the work_srv** and **start_srv** config stores.

work_srv and **start_srv** have the following definition in the memory:

```
struct srv
{
char NumElements;
struct __srv
{
char IP_C2[0x41];
DWORD Port;
}Src[NumElements]
};
```

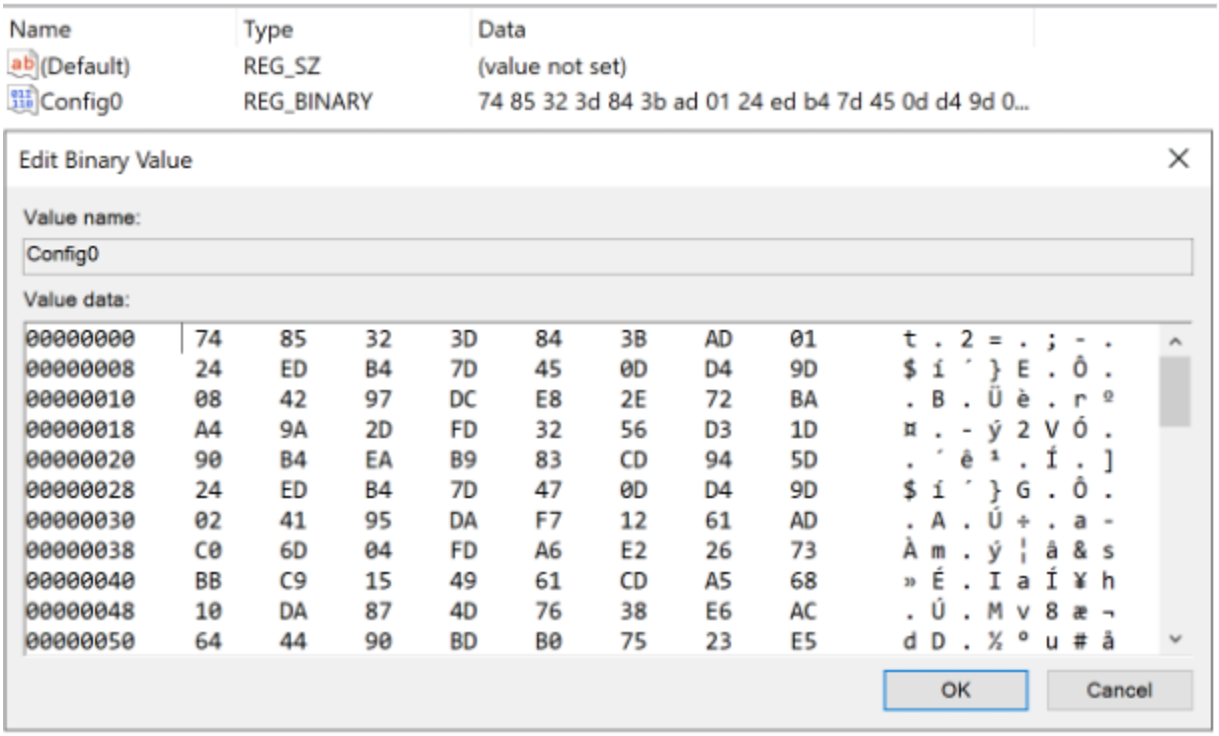
How can you exploit this for a vaccine?

In order to vaccinate Tofsee from connecting to first-tier or second-tier C&Cs, we can pollute these config stores' values before the start of the infection chain.
work_srv will point to a controlled sinkhole IP. In this example, we're going to point it to 127.0.0.1. In addition to this, we will recalculate the crc32 of data buffer so that it passes the integrity check inside the binary:



Modified value for wrk_srv (with proper crc32 hash value)

To create a vaccine, the above binary blob has to be encoded using the same algorithm and written back to one of the config store paths file or registry:



"Config0" modified registry value for vaccine

When Tofsee makes the connection, it only connects to the local sinkhole.

Simple!

The final of our Tofsee series looks at a network-based kill switch to protect against this malware.