So Long, and Thanks for All the Domains

securityblog.switch.ch/2015/06/18/so-long-and-thanks-for-all-the-domains/

June 18, 2015

While Trojans like Dyre and Dridex are dominating malware-related news, we take the time to have a closer look at Tinba (Tiny Banker, Zusy, Illi), yet another Trojan which targets Windows users. In the first part of this post, we give a short historical review, followed by hints about how to detect (and remove) this threat on an infected system. In the second part, we have a look at a portion of the Trojan's code which enhances its communication resilience, and how we can leverage these properties for defensive purposes.

Tinba is a fine piece of work, initially purely written in assembly. CSIS discovered it back in May 2012, and it contained WebInject capability and rootkit functionality in a binary of just 20 KB. The source code of Tinba leaked in July 2014, helping bad guys to create their own, extended versions.

```
.code

;; ------------------------------------------------------------------------------ ;;
NewZwQueryDirectoryFile proc p1:dword, p2:dword, p3:dword, p4:dword, p5:dword, p6:dword,
p7:dword, p8:dword, p9:dword, p10:dword, p11:dword
    local RealZwQueryDirectoryFile : dword

    mov RealZwQueryDirectoryFile, eax

@NextQuery:
    push p11                        ; RestartScan
    push p10                        ; FileName
    push p9                         ; ReturnSingleEntry
    push p8                         ; FileInformationClass
    push p7                         ; FileInformationLength
    push p6                         ; FileInformation
    push p5                         ; IoStatusBlock
    push p4                         ; ApcContext
    push p3                         ; ApcRoutine
    push p2                         ; Event
    push p1                         ; FileHandle
    call RealZwQueryDirectoryFile   ; Real ZwQueryDirectoryFile
    .if eax!=STATUS_SUCCESS
        ret
    .endif

; Only FileBothDirectoryInformation
    .if p8!=3
```

The source code of Tinba leaked in July 2014. Shown are some preparations to hook ZwQueryDirectoryFile.

Tinba on steroids was discovered in September 2014. Two main features are worth noting: First, each binary comes with a public key to check incoming control messages for authenticity and integrity. Second, there is a domain generation algorithm (DGA), which we will discuss later. In October 2014, Tinba entered Switzerland, mainly to phish for credit card information.

Tinba tried to phish credit card information.

Like other commodity Trojans, Tinba checks whether it is running in a virtual machine/sandboxed environment by checking the hard-disk size or looking for user interaction. According to abuse.ch, there was an intense distribution of Tinba in Switzerland early this year. Such spam campaigns can happen again at any time, so it is of use to know how to detect Tinba on an infected system and remove it.

Even though Tinba has the ability to hide directories and files (*rootkit functionality*), cybercriminals were wondering why they should bother using it. Why not simply hide directories and files with the “hidden” flag, which works for most users? Thus, it is relatively

simple for a computer-savvy user to remove this version of Tinba from an infected (see instructions below).

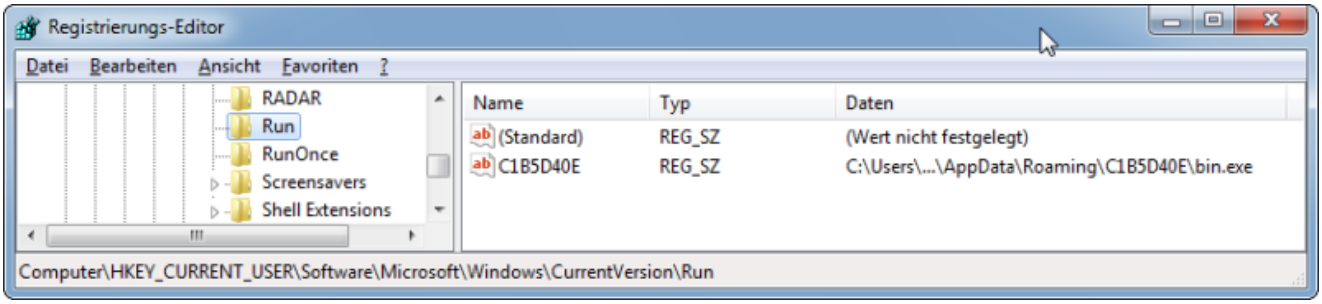
```
push    2                   ; FILE_ATTRIBUTE_HIDDEN
push    eax                 ; %AppData%\3B076ADF (directory)
lea     eax, [ebp-408h]
xchg    eax, [esp+40h+var_40]
-->  API Call: SetFileAttributesA@KERNELBASE
-->  Stack: 00000000,00000002
call    dword ptr [ebx+401165h]
```

A randomly named directory, which contains the Trojan itself, can be hidden by setting its attributes to “hidden”.

Disclaimer: We recommend completely re-installing the infected system. Very often, malware does mean things, even installing other malware, such that the two steps given below might not be sufficient to clean your system. Also, having a state-of-the-art anti-virus product installed helps to prevent infections.

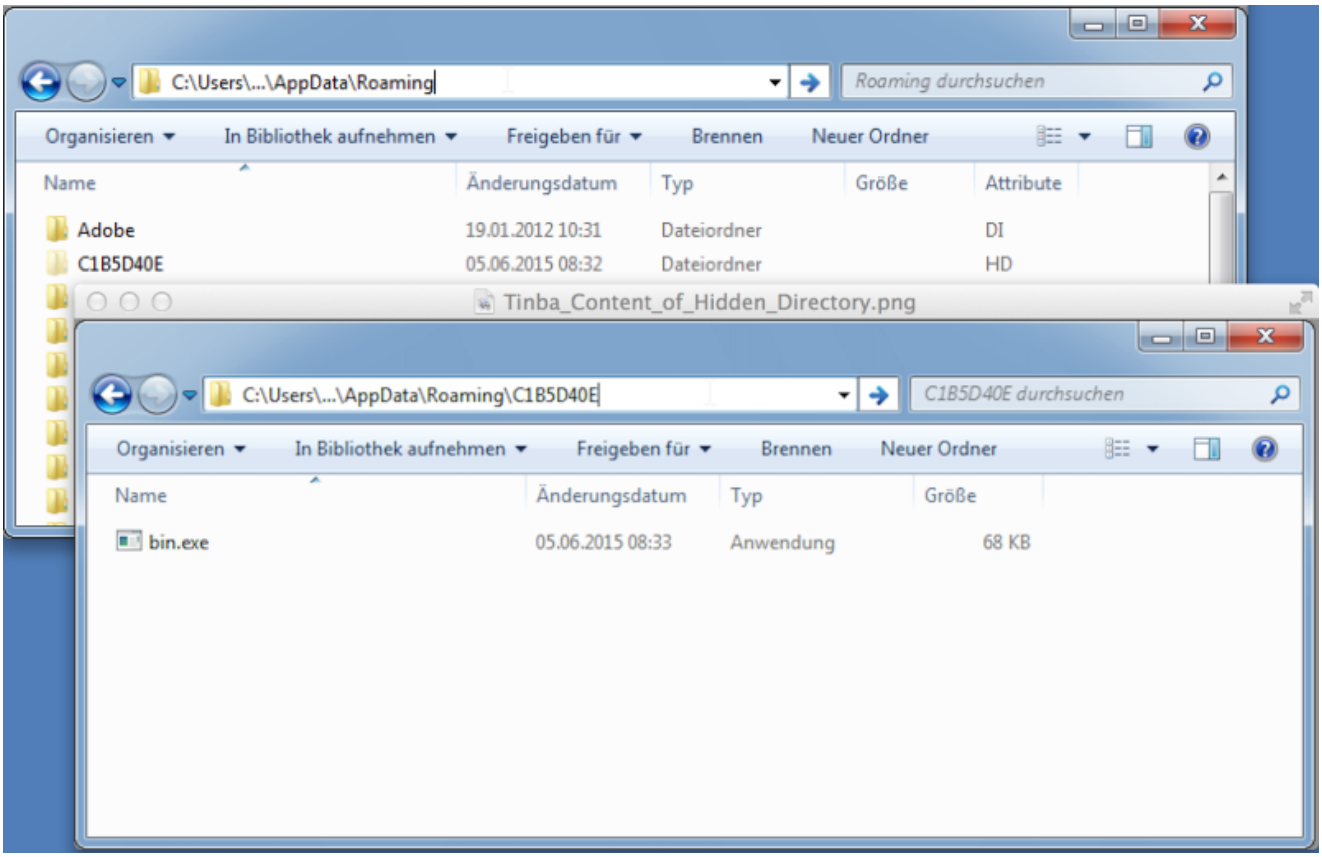
Step 1: Remove the malicious entry in the Windows registry.

- Follow these steps to start the Registry Editor and read the warnings.
- Navigate to the following key: HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
- Delete the weird randomly named entry, here *C1B5D40E*, pointing to a file called *bin.exe* in the directory *C:\Users\USERNAME\AppData\Roaming\C1B5D40E*.



Step 2: Remove the directory containing the malicious file (named bin.exe) or the malicious file itself.

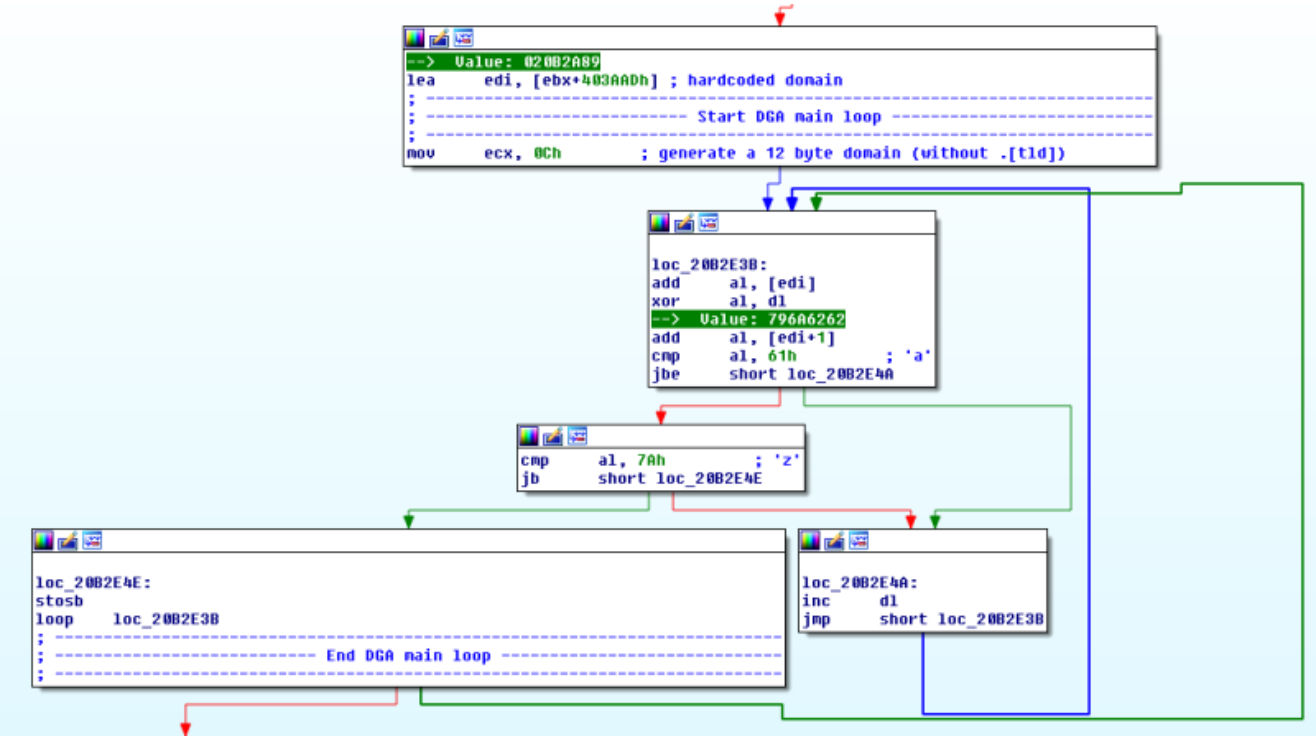
- Follow the steps described for Windows Vista/7 or for Windows 8.1 to show hidden files in Windows Explorer.
- Open Windows Explorer, type *%AppData%*, and hit Enter. Depending on the Windows version, you will end up in a slightly different directory than the one given below in the picture.
- You will see a hidden directory, here *C1B5D40E*. Within this directory, there is a file usually called *bin.exe* – the Trojan itself. The malicious entry in the Windows registry points to this file.
- Delete the randomly named hidden directory (or the file *bin.exe* in that directory) and reboot.



Windows 7: *%AppData%* resolves to *C:\Users\USERNAME\AppData\Roaming*. A randomly named directory will be there, but hidden from the eyes of an ordinary user.

It was mentioned above that a DGA was added to the Trojan's capabilities. The criminals want to enhance its resilience by making it harder for law enforcement agencies and security expoerts to execute a takedown or takover. Based on some well-defined input values, the DGA simply generates a bunch of domain names. Input values can be anything like a date, a foreign exchange reference rate at a certain time or a seed. Tinba uses an initial domain, a seed value and a couple of predefined top-level domains, such as .biz, .com, .in, .net, .pw, .ru, .space and .us.

In the case of Tinba, the DGA has been reverse-engineered by garage4hackers, Johannes Bader and others, and we would like to take this opportunity to say "thank you" for sharing. The DGA may change, of course. Looking at its main routines on a regular basis, or rather checking the generated domains against the predictions given by the reverse-engineered scripts, is thus essential. Currently, the (inner) main loop of the DGA still is consistent with the one provided by garage4hackers in September 2014.



A quick check of the inner main loop of the DGA for modifications: this part still produces the same output as the equivalent one from September 2014.

Would it not be nice to leverage a malicious feature like this for a security control? Yes it would, and we can! A DNS firewall can be built using DNS Response Policy Zones (RPZ). The RPZ gets populated by several means, one of them might be the output from DGA's. By monitoring malware that employs a DGA and extracting necessary input values, malicious domains can be computed and collected, for example in the DGArchive by Daniel Plohmann. These collections can be fed into an RPZ, and by operating a DNS firewall, end users (and Swiss universities in our case) can be protected – not necessarily from being infected, but hopefully from becoming victims of fraud.

Sample SHA-256:
e4627c1ea318a93305bc3b91d3f0a197547093882baff81ad11876a0b9e1e268