Posted on **2017-01-14**

← Previous **Next →**

Beyond good ol' Run key, Part 53

Most of the persistence methods described in this blog series so far focused on the old-school assumption that the system is a typical 'bare metal' Windows host. Over last couple of years many hosts are no longer real, and VDIs and Guest OS technologies they rely on introduce new techniques that attackers and malware can potentially use to stay persistent on the system. The features that are designed to support work of VDI may as well serve a malicious purpose. And this is what this post is about.

I will describe the example of VMWare Workstation as it is easy to test with, but it applies to VM server/cluster technologies as well.

VMWare Workstation is one of the most popular VM products on the market. Among many rich features that it offers, it is also including a well-documented mechanism that leverages a number of batch files that are executed when the power state of the guest OS changes – these states are:

- Power On
- Power Off
- Suspend
- Resume

Anyone who does reverse engineering for living uses these features all the time. VMs are an excellent solution to 'freeze' the session, test some code path, and revert to the previous snapshot if the code 'escapes' or program exits/crashes. It speeds up the analysis a lot and allows us to work in a comfort zone no other reversing tools can typically provide.

The VMware Workstation stores its batch files inside the c:\Program Files\VMware\VMware Tools\ directory – exploring this location we can quickly find a number of interesting files:

- c:\Program Files\VMware\VMware Tools\poweroff-vm-default.bat
- c:\Program Files\VMware\VMware Tools\poweron-vm-default.bat
- c:\Program Files\VMware\VMware Tools\resume-vm-default.bat
- c:\Program Files\VMware\VMware Tools\suspend-vm-default.bat

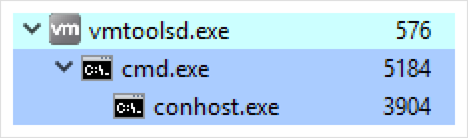
The top of the files typically states that:

```
@REM ##################################################################
@REM # DO NOT modify this file directly as it will be overwritten the r
@REM # time the VMware Tools are installed.
@REM ##################################################################
```

but of course, we are not going to listen.

Modification of these files can provides a decent persistence mechanism. Obviously, re-installing the VMTools and/or reverting to a previous snapshots can mitigate it, but if the VDI host is indeed running for a long time, this could be a possible good place to run something malicious from. In the VDI farm/cluster scenario, anyone who cans modify the source code of a vmware tools baseline could establish such persistent mechanism on (and infect) the whole cluster (deployment to all nodes).

Interestingly, these batch files can serve another purpose: they can cause some DoS annoyance (for a lack of a better name as it's hardly an anti-vm trick). Adding a simple 'pause' command to any of these files will 'prevent' the VMWare from executing the state-changing operation as it will wait for the cmd.exe process that handles the 'pause' command to exit:



```
vmtoolsd.exe      576
  cmd.exe         5184
    conhost.exe   3904
```

Since 'pause' requires pressing the key and the script is running in a different session the vmtoold.exe will just enter a sort of never-ending loop. The side-effect of it is that the user won't be able to f.ex. 'suspend' or 'power off' the guest system.

This entry was posted in **Anti-Forensics**, **Autostart (Persistence)**, **Compromise Detection**, **Forensic Analysis**, **Malware Analysis** by **adam**. Bookmark the **permalink**.