GOPAS
POCITACOVA.SKOLA.CZ

Hiding Things on Windows Platforms

HACKERFest 2015

Martin Dráb

martin.drab@email.cz

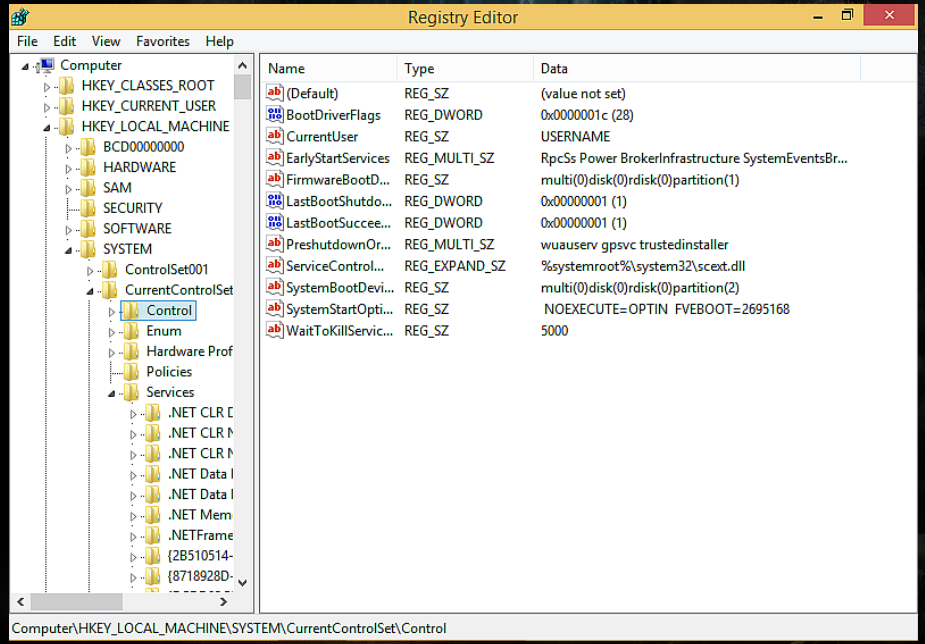
Agenda

- Areas of interest
  - Windows registry
  - Network communication
  - File systems
- Attacker's point of view
- No extra knowledge required


Windows Registry

Windows Registry – basic info

- Configuration storage similar to file systems
  - keys = directories
  - values = files
- Typed, small sized data
- Stored on disk, lives in RAM



Filtering Registry Operations

- Driver-supplied callbacks (CmRegisterCallback(Ex))
  - Invoked before and after a registry operation in context of the thread taht irequested it
  - Multiple callback registrations allowed (layered model)
- Modes of operation
  - Monitor (gather information about each operation)
  - Block (the requestor receives an error)
  - Modify (operation parameters and/or results can be changed)
  - Emulate (the whole operation is performed by the filter driver)

RegHider

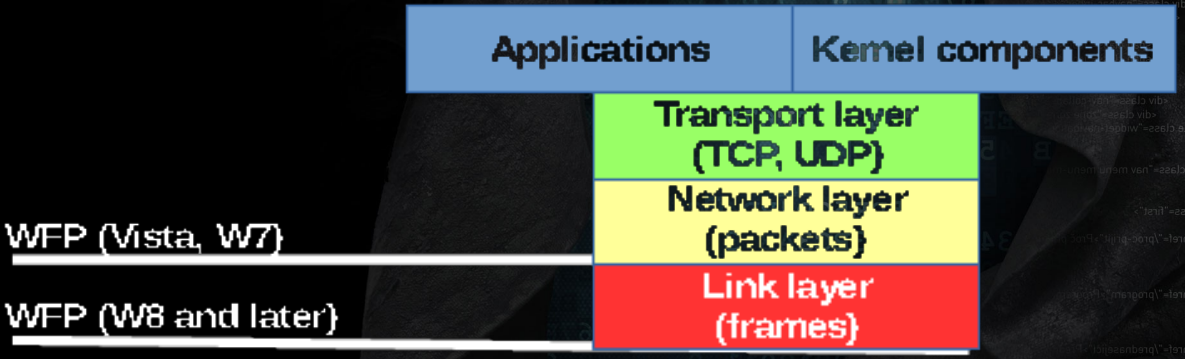
- Hides registry keys
  - QueryKey and EnumerateSubkeys operation modified in order not to report given registry keys
- Reports nonexistent registry values
  - Operations on these values are emulated
  - Each value may be visible either systém-wide, or only to certain processes

Detection

- Special registry operations
- Offline analysis
  - Offline Registry Library
  - Parsing registry files directly (e.g. chntpw)
- Online analysis
  - Registry tree is stored in memory (usually)
  - It can be read directly (without using any system routines)


Network
Communication

Network Basics



Winsock Kernel

- A socket-like interface for the kernel world
  - Usual socket calls (socket(), bind(), connect(), listen())
  - Used in the same way as in usermode
  - An easy way how a driver can comminucate over network
- Implementation details
  - Asynchronous
  - Nonpaged memory

Raw Ethernet Frames

- Kernel dirvers are able to send raw ethernet frames over network, thus bypassing any packet filters placed above
- Difficulty: easy (PoC at least)
  - Just get the ndisprot sample from the WDK package
  - Delete few lines of code and it will work
  - Fill in the IP header correctly

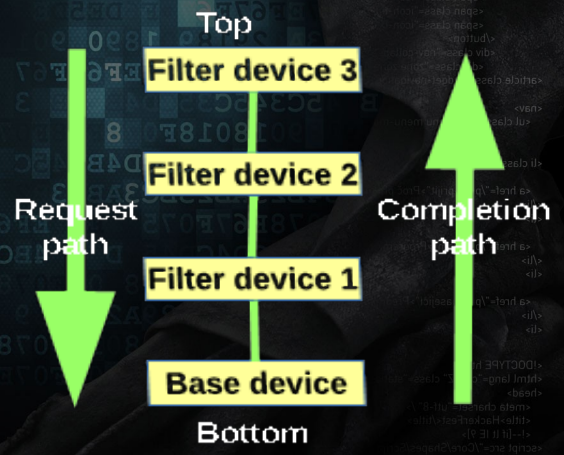
File Systems

Driver and Device Objects

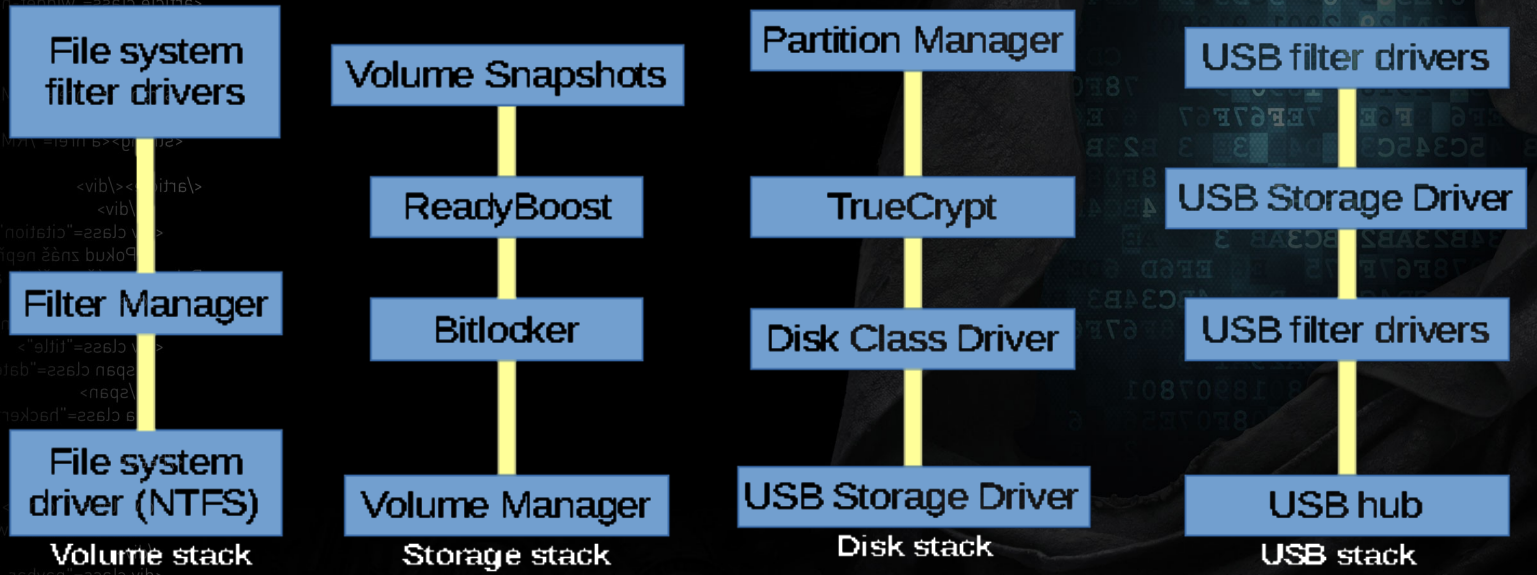
- Driver object
  - Gathers all necessary information about certain device driver, such as addresses of routines servicing various types of requests
- Device object
  - Receives requests from applications and other drivers
  - A driver owning the device is responsible for handling incomming requests

Device Stacks

- When a driver needs to see requests comming to a particular device (of another driver), it can **attach** its own device object **above** that device and form a **device stack**
- Requests usually sent to the top device
- Unknown requests passed down
- All drivers in the stack should keep their manners



Where to Attach?



Implications

- Problems of a lower attacher
  - Hardware dependencies
  - Software encryption
  - No context information
  - Direct (S)ATA/SCSI commands
  - Writting a basic FS parser is quite easy
- Tips & Tricks
  - Build your own file system
  - Take advantage of symbolic links (such as drive letters)


Conclusion

References

- Demo projects (may be updated in the future)
  - https://github.com/MartinDrab/Hackerfest2015
- Registry
  - https://msdn.microsoft.com/en-us/library/windows/hardware/ff545879(v=vs.85).aspx
  - http://dfrws.org/2008/proceedings/p26-dolan-gavitt.pdf
  - http://amnesia.gtisc.gatech.edu/~moyix/suzibandit.ltd.uk/MSc/Registry%20Structure%20-%20Appendices%20V4.pdf
- FAT & NTFS data structures (mirrors)
  - http://www.jadro-windows.cz/download/ntfsdoc-0.5.zip
  - http://www.jadro-windows.cz/download/fat.zip


www.hackerfest.cz
www.gopas.cz
HACKERFest | 2015
GOPAS
POCITACOVA.SKOLA.CZ