**Transcript of Tracking Stuxnet's Footprint Through Memory**

"a normal Windows XP installation has just one instance of Lsass.exe that the Winlogon process creates when the system boots (Wininit creates it on Windows Vista and higher). The process tree reveals that the two new Lsass.exe instances were both created by Services.exe (not visible in the screenshot), the Service Control Manager, which implies that Stuxnet somehow got its code into the Services.exe process." [1] "Mrxnet.sys is the driver that the programmer originally sent me and that implements the rootkit that hides files, and Mrxcls.sys is a second Stuxnet driver file that launches the malware when the system boots." [1] Tracking Stuxnet's Footprint  
Though Memory  
  
with Volatility 2.0 Michael Ligh  
OMFW 2011 modules  
modscan  
moddump getsids "...besides running as children of Services.exe, another suspicious characteristic of the two superfluous processes is the fact that they have very few DLLs loaded..." [1] dlllist "No non-Microsoft DLLs show up in the loaded-module lists for Services.exe, Lsass.exe or Explorer.exe, so they are probably hosting injected executable code. [....] Sure enough, the legitimate Lsass has no executable data regions, but both new Lsass processes have regions with Execute and Write permissions in their address spaces at the same location and same size." [1] malfind ldrmodules "...because we see Lsass.exe drop one of the two Stuxnet drivers, MRxCls.sys, in C:\Windows\System32\Drivers and create its corresponding registry keys" [2] "The final modifications made by the virus include the creation of four additional files in the C:\Windows\Inf directory: Oem7a.pnf, Mdmeric3.pnf, Mdmcpq3.pnf and Oem6c.pnf." [2] "That means Mrxcls.sys called PsSetLoadImageNotifyRoutine so that Windows would call it whenever an executable image, such as a DLL or device driver, is mapped into memory." [3] "The driver scans for the following filesystem driver objects:  
  
\* \FileSystem\ntfs  
\* \FileSystem\fastfat  
\* \FileSytstem\cdfs  
  
A new device object is created by Stuxnet and attached to the device chain for each device object managed by these driver objects. [...] By inserting such objects, Stuxnet is able to intercept IRP requests (example: writes, reads, to devices NTFS, FAT, or CD-ROM devices)." [4] "In the driver file, the project path b:\myrtus\src\objfre\_w2k\_x86\i386 \guava.pdb was not removed." [4] pslist  
psscan  
pstree vadinfo procexedump  
procmemdump apihooks volshell malfind hivescan  
hivelist  
printkey svcscan callbacks driverirp malfind (kernel) filescan   
handles devicetree "The driver also registers to a filesystem registration callback routine in order to hook newly created filesystem objects on the fly." [4] handles connscan/connections  
sockscan/sockets  
netscan "...some Windows system processes (such as the Session Manager, service controller, and local security authentication server) have a base process priority slightly higher than the default for the Normal class (8)." - Windows Internals 5th Edition pg. 395 threads volshell "It registers a window class with the name "AFX64c313" and creates a window corresponding to the class created. The window procedure of the class monitors WM\_DEVICE\_CHANGE messages sent when there is a change to the hardware configuration of a device or the computer. The window procedure of the class handles only requests with wParam set to DBT\_DEVICEARRIVAL." [6] atomscan userhandles  
volshell "Stuxnet communicates between different components via global mutexes." [4] "This function [export #28] is responsible for performing actual data exchange with the C&C server. In the event that there is no iexplore.exe in the system, it calls this function from the address space of the default browser: it starts the default browser as a new process, injects into it the main module, and calls the function performing data exchange.  
  
The malware communicates to the C&C server through http. A list of URLs is included in the Stuxnet configuration data of Stuxnet:  
  
www.mypremierfutbol.com  
www.todaysfutbol.com" [6] "To hook the functions specified above, the malware allocates a memory buffer for code that will dispatch calls to hooked functions, overwrite some data in MZ header of the image with the code that transfers control to the new functions, and hook the original functions by overwriting its bodies..." [6] "Stuxnet calls LoadLibrary with a specially crafted file name that does not exist on disk and normally causes LoadLibrary to fail. However, W32.Stuxnet has hooked Ntdll.dll to monitor for requests to load specially crafted file names. These specially crafted filenames are mapped to another location instead—a location specified by W32.Stuxnet. That location is generally an area in memory where a .dll file has been decrypted and stored by the threat previously. The filenames used have the pattern of KERNEL32.DLL.ASLR.[HEXADECIMAL]..." [4] "The functions hooked for this purpose in Ntdll.dll are:  
  
\* ZwMapViewOfSection  
\* ZwCreateSection  
\* ZwOpenFile  
\* ZwCloseFile  
\* ZwQueryAttributesFile  
\* ZwQuerySection" [4] mutantscan thrdscan malfind + yara "Stuxnet communicates between different components via global mutexes." [4] handles "Mrxnet.sys is the driver that the programmer originally sent me and that implements the rootkit that hides files, and Mrxcls.sys is a second Stuxnet driver file that launches the malware when the system boots." [1] modules  
modscan  
moddump "...because we see Lsass.exe drop one of the two Stuxnet drivers, MRxCls.sys, in C:\Windows\System32\Drivers and create its corresponding registry keys" [2] printkey svcscan "The final modifications made by the virus include the creation of four additional files in the C:\Windows\Inf directory: Oem7a.pnf, Mdmeric3.pnf, Mdmcpq3.pnf and Oem6c.pnf." [2] filescan "That means Mrxcls.sys called PsSetLoadImageNotifyRoutine so that Windows would call it whenever an executable image, such as a DLL or device driver, is mapped into memory." [3] callbacks "The driver scans for the following filesystem driver objects:  
  
\* \FileSystem\ntfs  
\* \FileSystem\fastfat  
\* \FileSytstem\cdfs  
  
A new device object is created by Stuxnet and attached to the device chain for each device object managed by these driver objects. [...] By inserting such objects, Stuxnet is able to intercept IRP requests (example: writes, reads, to devices NTFS, FAT, or CD-ROM devices)." [4] devicetree driverirp "The driver also registers to a filesystem registration callback routine in order to hook newly created filesystem objects on the fly." [4] IoRegisterFsRegistrationChange [Yes]  
IoRegisterDriverReinitialization [No] "In the driver file, the project path b:\myrtus\src\objfre\_w2k\_x86\i386 \guava.pdb was not removed." [4] malfind / kernel "...some Windows system processes (such as the Session Manager, service controller, and local security authentication server) have a base process priority slightly higher than the default for the Normal class (8)." - Windows Internals 5th Edition pg. 395 threads volshell "It registers a window class with the name "AFX64c313" and creates a window corresponding to the class created. The window procedure of the class monitors WM\_DEVICE\_CHANGE messages sent when there is a change to the hardware configuration of a device or the computer. The window procedure of the class handles only requests with wParam set to DBT\_DEVICEARRIVAL." [6] atomscan userhandles volshell connscan / connections  
or  
netscan "This function [export #28] is responsible for performing actual data exchange with the C&C server. In the event that there is no iexplore.exe in the system, it calls this function from the address space of the default browser: it starts the default browser as a new process, injects into it the main module, and calls the function performing data exchange.  
  
The malware communicates to the C&C server through http. A list of URLs is included in the Stuxnet configuration data of Stuxnet:  
  
www.mypremierfutbol.com  
www.todaysfutbol.com" [6] Thank You! EXTRA LSASS PROCESS PRIORITY FEWER DLLS INJECTED CODE HIDDEN DLLS HOOKED APIS PATCHED PE HEADER MUTEXES FILE OBJECTS TCP CONNECTIONS REGISTRY KEYS KERNEL DRIVERS KERNEL CALLBACKS FILESYSTEM HOOKS DEVICES & IRPS PDB REFERENCE WINDOWS & CLASSES RPC server  
Jobs/Tasks  
Vulnerabilities  
UPX packing  
Fake certs [1] Mark Russinovich's Analyzing a Stuxnet Infection with the Sysinternals Tools, Part I  
[2] Mark Russinovich's Analyzing a Stuxnet Infection with the Sysinternals Tools, Part II  
[3] Mark Russinovich's Analyzing a Stuxnet Infection with the Sysinternals Tools, Part III  
[4] Symantec's W32.Stuxnet Dossier  
[5] Amr Thabet's MrxCls - Stuxnet Loader Driver  
[6] ESET's Stuxnet Under The Microscope References: mnin.blogspot.com / iMHLv2 malware author