Updated Report on progress of the project

Mohammad-Reza Memarian

December 17, 2014

1 Report on assigned tasks

Following our previous discussion in skype, we set several steps to follow up to progress the project. Most of the efforts since the last meeting has been focused on development of information gathering module in FVM. As we discussed during last online meeting, the steps set to be done for this session are in the following sub sections.

1.1 New draft in Paper format

The most updated information is placed in ACM paper format. The previous paper draft is reviewd and updated information is added to the new version of the draft. The new version has been sent during last week.

1.2 Information gathering system

The focus of this part is implementation of the system on Dom0 for now. So all the tests take place on Dom0 to read guest memories. For now Dom0 is a linux kernel and guest VM is a 32-bit Microsoft windows 7.

1.2.1 Symptoms identification

In this section, we agreed to look into some malware reports to find out what are the possible matters that our system should look for them. As an example, creating a new registry key is a prevalent symptom among malwares. Existing systems which look for malicious behaviour try to look for existance of certain registry key to detect the malware. So for every malware there should be a signature. But as we look for symptoms, we only care about the matter that in a certain directory, a key has been created and our system does not look into what key is created. Ofcourse a key can be created for legitimate purposes. But adoption of the change over several machines may not be considered normal.

		Rustock (2006)	Virut (2007)	Koobface (2008)	Conficker (2008)	Tidserv (2008)	Akbot (2009)	Pilleuz (2009)	Zbot (2010)
System modification	File/Folder created	✓		✓	✓	✓	✓	✓	✓
	Files/Folder deleted					✓		✓	
	Files/Folder modified		✓		✓	✓	✓	✓	
	Registry entries created	✓	✓	✓	✓	✓	✓	✓	✓
	Registry entries/deleted				✓				
	Registry entries/modified				✓				
	Process Injection	✓	✓				✓	✓	✓

Figure 1: Table showing changes of malware to system

Other matter is that in case of zero-day malware and absense of suitable detectable signature, only looking for one host system changes is not a suitable mechanism. Third point is, existance of polymorphic pieces of malware which may have different behaviour on different system changes across multiple machines. For example name of the files that are created may change. But in our system in this case we only look for symptom of creation of file in certain directories, no matter what name does it have.
Eight recent malware that were intended for formation of botnets are listed in Figure 1. The Malware's functionalities based on system modification criteria has been compared. Goal of creation of this table is to find out what functions are mostly prevalent among malwares that were intended fot botnet activities. Another table thas has been shown is Figure 2, is made so it shows some of the possible instances of events that may occur in result of botnet malware system modifications. We should dig down and choose some of them so our proposed system can detect based on the chosen criteria. In one of the related works, researchers introduced their system by looking only into loaded DLLs. Other researchers in another work presented their system by looking into VM's process and module list. Then their system would compare both lists against a black list. They proposed a detection system that is placed in VMM level so it has a view over all VMs only in one host. In the first work done by university of Birmingham and HP on introduction of FVMs looking for symptoms instead of malware itlsef, also they did introduce limited number of symptoms detection modules. In their paper they have mentioned that It is suitable if more symptoms are looked for while detecting malware. I believe this limitation in looking into symtoms is because of the hard nature of developing VMI tools.

There is an open source project called Volatility which is a tool for digital memory forensic. It can analyze dumps of memories (Snapshot) of Mac, Linux and Windows operating systems. But combining it with LibVMI gives a super powerful feature to users which allows memory forensic (Introspection) on Live VMs. So based On the information which we can obtain using Volatility, it is possible to obtain great information from memory. By having a look into func-

Functionality	Instances of occurance	Entity to look for	Comment
Using Trusted process to astray detection system	Bot loads itself as DLL	Check list of loaded DLLs against a white list	When bot is installed on a machine, it will copy itself into a configurable install directory and change system state to start when the system starts (refer to change registry section). But they may create a directory for residing bot files too.
	Bot loads itself as executable code	check list of running processes against a white list	
	Bot loads itself as a threat	check list of available threats against a white list	
Start of suspicious processes in process list with trivial name-based obfuscation	Processes with encoded name convensions (meaningless names)	Check process list entries against a whitelist	
	processes with names reasembling other legitimate system processes		
Rootkit technique	Downloading tools such as Network sniffers, log-cleaning scripts, SMPT client, HTTP client	Check process list entries against a blacklist	Malware may downlods various tools which also in refered to secondary local infection
Reducing security strength and changing security rules	Disabling firewall or reduction in protection level	Firewall status change detection	
	Disabling antivirous or downloading new one	Antivirous status change detection	
Usage of P2P file sharing protocol	P2P file sharing protocol gets used for malware propogation among the machines	open ports bonding to specidic protocols	Nomally hosts which have 24x7 high speed network connectivity are targets, like university servers
	High network usage		

Figure 2: Table showing result of malware functions applied

Functionality	Instances of occurance	Entity to look for	Comment
Registry modification	Malware may add, delete or modilfy registry key entries	Check for certain registry key changes	For windows, bot may add itself too HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run to start each time system boots.
	Creation of users with names mimicing system users but by admin rights	query user list and their privilages and look for changes compare to their previous status	
	IM protocol gets used to transfer command from C&C to zombies	open ports binded to specidic protocols	
Reduce system capability	Disabling task manager	Identifying critical processes based on a list and detect their status changes	Disabling certain tools on the system to reduce user's capability to control the system
	Disabling command shell		
	Preventing clean-up efforts		
Misuse from Zombies storage spaces	Storing unappropriate contents on servers with large storage capacities	usage of network, disk write and read rate usage	malicious bots make victim machine part of file sharing network and use their storage space as part of a file sharing network specially happens to servers with high speed networks and large storage area (IROFFER) is a file sharing malware.
Address blocking and trafic redirection	host file modification	check status of host file for changes	By modifying host files, legitimate traffic gets redirected resulting in access to unlegitimate hosts instead of legitimate destinations.
	adding of new blocking firewall rule	detection of any change in firewall rules	

Figure 3: Table showing result of malware functions applied

tions that Volatility can offer, we can see some valuable functions. Volatility offers various functions that are related to processes and DLLS, Kernel memory and Objects, process memory, networking, registry, malware and rootkit and miscellaneous. All the funtions do not work on all version of operating systems. For instances, most of the networking functions do not work on Windows 7 operating system. In our case as our focus is just to system modification symptoms, we do not look into network symptoms. But in case of need, it is possible to write the code for them. Instances of functions in Processes and DLLs category are in the following:
- pslist: Lists processes of the system. This function can not find hidden processes.
- psscan: Finds terminated and hidden process which helps to find rootkits.
- dlllist: It can show loaded libraries for the whole processes or just an specific process.
- handles: Displays open handles in a process.
Another category that mentioned earlier is Malware and Rootkits and it comprises of following functions:
- malfind: This function is used to find hidden or injected code/DLL, the one that mentioned in table 2 in user mode memory based on characterestics such as page permissions.
- svcscan: Volatility is the only memory forensic with the ability to list windows services. It is to check what services are registred on memory.
-ldrmodules: dlllist may not find hidden loade libraries but malfind is a function that may help. ldrmodules is another function that can help to find hidden DLLs.

1.3 Machine learning Algorithms

Looking into this phase is not started yet. Most of the focus is currently on rapid development of module module which is located in FVM and transporting them into central processing center.