REGISTRY FORENSICS

The Registry is a various levelled or we can say a hierarchical database that stores low-level settings and other information for the Microsoft Windows Operating System and for applications that pick to utilize the registry. From the point of installation of operating system, registries are used. Kernel, Device Driver settings to the Hardware and User Interface all settings are stored in the windows registry.

When Programs and Applications are installed in the system their configurations and default values are stored in the registry although there are some applications which do not utilize windows registry. For example, .NET framework applications use XML files for configuration, Portable applications usually keep their configuration data within files in the directory/folder where the application executable resides.

Importance of Registry in Windows Forensics

For a Forensic analyst, the Registry is a treasure box of information. It is the database that contains the default settings, user, and system defined settings in windows computer. Registry serves as repository, monitoring, observing and recording the activities performed by the user in the computer. The Data is stored in the main folders in a Tree like structure which is called Hive and its subfolders are called KEYS and SUBKEYS where each component's configuration is stored called VALUES. Some Important aspects of Windows Registry are:

Windows Registry can be considered as a gold mine of forensic evidence.

We can create new registries manually or we can modify the ones that already exist.

Original files that contain registry values are stored in the system directory itself.

Registry files are system protected and can not be accessed by any user unless administration access is provided.

For the investigation purpose, the forensic investigator analyzes registry files via tools such as Registry Viewer, Regshot, Registry Browser etc..

Trojans and Malware information can be found in the registries.

Main Registry Hives

HKEY_CLASSES_ROOT

HKEY_CURRENT_USER

HKEY_LOCAL_MACHINE/SAM

HKEY_LOCAL_MACHINE/SOFTWARE

HKEY_LOCAL_MACHINE/SECURITY

HKEY_LOCAL_MACHINE/SYSTEM

HKEY_USERS

HKEY_CURRENT_CONFIG

While acquiring registry files from the system we need to use an Imaging tool which can obtain system protected files because then only we can access and analyze them with the help of registry viewer. We can not obtain these files directly from the system because they are currently being used by the system to access registry editor.

Virtual Paging

The virtual memory paging process uses page tables, which translate the virtual addresses that the OS and applications use into the physical addresses that the MMU uses. Entries in the page table indicate whether the page is in RAM. If the OS or a program does not find what it needs in RAM, then the MMU responds to the missing memory reference with a page fault exception to get the OS to move the page back to memory when it is needed. Once the page is in RAM, its virtual address appears in the page table.

Segmentation is also used to manage virtual memory. This approach divides virtual memory into segments of different lengths. Segments not in use in memory can be moved to virtual memory space on the hard drive. Segmented information or processes are tracked in a segment table, which shows if a segment is present in memory, whether it has been modified and what its physical address is. In addition, file systems in segmentation are only made up of segments that are mapped into a process's potential address space.

What are the benefits of using virtual memory?

The advantages to using virtual memory include:

1. It can handle twice as many addresses as main memory.
2. It enables more applications to be used at once.
3. It frees applications from managing shared memory and saves users from having to add memory modules when RAM space runs out.
4. It has increased speed when only a segment of a program is needed for execution.
5. It has increased security because of memory isolation.
6. It enables multiple larger applications to run simultaneously.
7. Allocating memory is relatively inexpensive.
8. It does not need external fragmentation.
9. CPU use is effective for managing logical partition workloads.
10. Data can be moved automatically.
11. Pages in the original process can be shared during a fork system call operation that creates a copy of itself.

Malware

Malware is a kind of intrusive software that damages and destroys computer systems, servers, host systems, or networks. It is a catch-all term for all types of malicious software that is specifically intended to cause damage or exploit any programmable device, network, or service. Viruses, worms, adware, spyware, trojan viruses, and ransomware are various types of malware threats.

What is Malware Analysis?

Malware analysis is the process of detecting and reducing potential threats in a website, application, or server. It is a crucial process that ensures computer security as well as the safety and security of an organization with regard to sensitive information. Malware analysis addresses vulnerabilities before they get out of hand.

If you are looking at it more simply, malware analysis can be considered as the process of understanding the behavior and the intended use of a suspicious file or URL. The more you know about the suspicious file, the better it will help to mitigate the threat, if any.

Key Benefits of Malware Analysis

Malware analysis is of immense use to Security Analysts and incident responders. Here are some key benefits of the process:

1. Identifying the source of the attack
2. Determining the damage from a security threat
3. Identifying a malware's exploitation level, vulnerability, and appropriate patching preparations
4. Triaging the incidents according to the level of severity of the threat in a practical manner
5. Uncovering hidden Indicators of Compromise (IOC) that need to be blocked
6. Improving the efficacy of IOC, alerts, and notifications
7. Enriching context when trying to uncover threats

Types of Malware Analysis

There are three types of malware analysis that can be conducted:

1. Static malware analysis
2. Dynamic malware analysis
3. Hybrid malware analysis

Static Malware Analysis

Static malware analysis examines files for signs of malicious intent. A basic static analysis does not require a malware code that is actually running. It is useful for revealing malicious infrastructure, packed files, or libraries.

In this kind of malware analysis, the technical indicators like file names, hashes, strings such as IP addresses, domains, and file header data are identified. Various tools like disassemblers and network analyzers have the ability to observe the malware without running it. These tools can gather information on how the particular malware works.

Since static malware analysis does not run the malware code, there can be malicious runtime behavior in some sophisticated malware, which can go undetected. For example, a file that generates a string and downloads a malicious file depending on the dynamic string. The malware could go undetected if a basic static malware analysis is used. In these cases, dynamic analysis is more helpful in getting a complete understanding of the file behavior.

Dynamic Malware Analysis

In dynamic malware analysis, a suspected malicious code is run in a safe environment called a sandbox. This isolated virtual machine is a closed system that allows security experts to observe the malware closely in action without the risk of system or network infection. This technique provides deeper visibility of the threat and its true nature.

Automated sandboxing, as a secondary benefit, eliminates the time, which otherwise would have been spent for reverse engineering a file to discover a malicious code.

Dynamic analysis can be a challenge, especially against smart adversaries who know sandboxes will be used eventually. So, as a form of deception, adversaries hide their code in a way that it remains dormant until specific conditions are met. The code will run only then.

Hybrid Malware Analysis

We already know now that basic static analysis isn't reliable when the malware has a more sophisticated code, and sophisticated malware are sometimes, able to avoid detection by sandbox technology. Combining both types of malware analysis techniques offers the best of both approaches.

Hybrid analysis can detect hidden malicious code, and extract many more IOCs by statically and previously unseen code.It is capable of detecting unknown threats, even from the most sophisticated malware.

The hybrid analysis applies static analysis to the data that is generated by behavioral analysis. Consider a piece of malicious code that runs and causes some changes in memory. The dynamic analysis will be able to detect that and Analysts will immediately know to perform static analysis on that memory dump. This will result in more IOCs and exposed zero-day exploits.