**Приветствие (Слайд 1)**

Greetings ladies and gentlemen, today we’d like to talk about A Method for Malware Analysis by Virtual Machine Introspection Technique.

**План (Слайд 2)**

**Введение (Слайд 3)**

Malicious code has become one of the biggest threats in the field of computer security, the number of malware has grown in recent years.

Traditional malware monitoring tools are installed in the physical host, however, they are vulnerable to being infected by malware and delivering erroneous results about monitoring.

**Постановка проблемы (Слайд 4)**

Traditional monitoring tools have become too vulnerable and unprotected for malware attacks.

Our goal is to propose and experimentally test a malware monitoring method based on the virtual machine introspection method, which allows you to obtain a virtual machine memory image from the outside

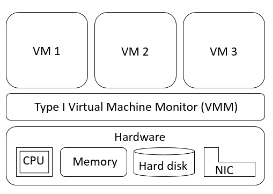
**Virtual Machine Monitor (Слайд 5)**

Virtual Machine Monitor (VMM) or hypervisor is a software that enables communication between Virtual Machines and real host.

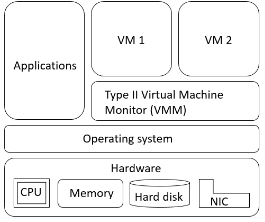
It provides the virtual environment by means of a Virtual Machine where other programs can be executed just as they do in a real environment

**Types of Virtual Machine Monitor (Слайд 6)**

1. Type I VMM is one that runs directly on the hardware, they are used in data centers and in server environment.

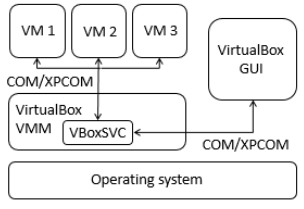


1. The type II VMM is installed on the operating system of the real host as another user program



**VirtualBox Hypervisor (Слайд 7)**

The approach proposed is based in one of the type II VMM that is called VirtualBox. It provides a main API that is implemented using the Component Object Model, an interprocess mechanism for software components.



**Virtual Machine Introspection (Слайд 8)**

Virtual Machine Introspection is a technique to analyze the memory of a given VM to detect its internal activities from outside over the Virtual Machine Monitor layer.

This technique has been used for intrusion detection, malware analysis and memory forensics.

**Design of the Method (Слайд 9)**

1. Access to the asset: This is achieved using an interprocess mechanism called COM/XPCOM
2. Collection: It generates a memory dump of the Virtual Machine volatile memory.
3. Analysis: The program converts low-level bytes into high-level information and extracts objects from the operating system
4. Logging: It generates the log of the malware analysis.
5. Containment: The COM/XPCOM interprocess mechanism sends a killing command to finish the malware execution from outside the Virtual Machine.

**Experimental Results (Слайд 10-11)**

In order to test the operation and functioning of the approach, three experiments were performed, each of them consisted in the execution of a different sample of malware inside the monitored Virtual Machine.

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment** | **Malware** | **MD5** | **Antivirus detection from VirusTotal** |
| 1 | Trojan | 7583a73f73638d23298ddb4900def643 | 56/64 |
| 2 | Trojan | 8915452ee0b8e754ee7b047a849a01a2 | 58/68 |
| 3 | Trojan | c334b788e3da78c413364ef1e163b8ff | 43/68 |

* In the first experiment the introduced malware executed 7 threads, 4 network connections and 24 open files.
* In the second experiment 5 threads, 1 network connection and 21 open files were registered
* The last experiment consisted in the execution of the third sample of malware, which was associated with the creation of 1 thread, 1 network connection and 12 open files

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Experiment** | **Malware** | **#Threads** | Network connections | | **# Open files** | **Suspicious files** |
| Port | Address |
| 1 | Trojan | 7 | 8143 | 199.7.136.88 | 24 | ws2\_32.dll, ws2help.dll, mswsock.dll, dnsapi.dll, 996E.exe. |
| 1743 | 151.80.142.33 |
| 243 | 202.69.40.173 |
| 7447 | 78.47.66.169 |
| 2 | Trojan | 5 | 1029 | 0.0.0.0 | 21 | wsock32.dll, ws2\_32.dll, crypt32.dll |
| 3 | Trojan | 1 | 80 | 211.104.175.45 | 12 | ws2\_32.dll, mswsock.dll, wshtcpip.dll, rpcrt4.dll, |

**Conclusions (Слайд 12)**

By analyzing the virtual machine memory image, the behavior of three malware samples in the virtual machine was obtained, such as their IDs, the IDs of their parent processes, the number of threads, as well as their network connections and open files.

Thus, the experimental results confirm the hypothesis that malware can be identified using the virtual machine introspection method.

**Прощание (Слайд 12)**

Thank you for your attention