BotViz: A Memory Forensic-Based Botnet Detection and Visualization Approach

Iman Sharafaldin, Amirhossein Gharib, Arash Habibi Lashkari and Ali A. Ghorbani

Canadian Institute for Cybersecurity (CIC), University of new Brunswick (UNB)

ABSTRACT

Nowadays, there are many serious cyber security threats such as viruses, worms and trojans but without a doubt botnets are one of the largest threats. Although there are numerous ways to discover botnets and mitigate their effects, most methods have problems effecting detection, due to their evasive characteristics. Also, the majority of previous research uses only one data source (e.g. network traffic), which makes the botnet detection process very difficult over a network. This paper proposes a detection and visualization system, BotViz, to visualize botnets by using memory forensics analysis and a new domain generation algorithm detector. BotViz utilizes machine learning techniques to detect anomalous function hooking behaviors. We established a live Zeus botnet to evaluate the efficiency of the BotViz.

Data Collector Module (DCM)

The DCM is responsible for collecting DNS logs and memory dumps of virtual machines and sending them to other modules for further processing. The DCM also interacts with a hypervisor, and collects memory dumps of virtual machines using two methods, first method uses software development kit (SDK) for that specific hypervisor, second method uses any type of Virtual Machine Introspection (VMI) which is a common technique to examine the memory space of a virtual machine from a secure point.

Memory Forensics Module (MFM)

This module is responsible for processing memory dumps received from the DCM, and for detecting available hooks for each virtual machine. Volatility Framework is used to detect hooks from memory dumps. In order to find user mode and kernel mode hooks, BotViz uses APIhooks plugin. This plugin finds several types of hooks such as IAT (Import Address Table), EAT (Export Address Table), and Inline style hooks.

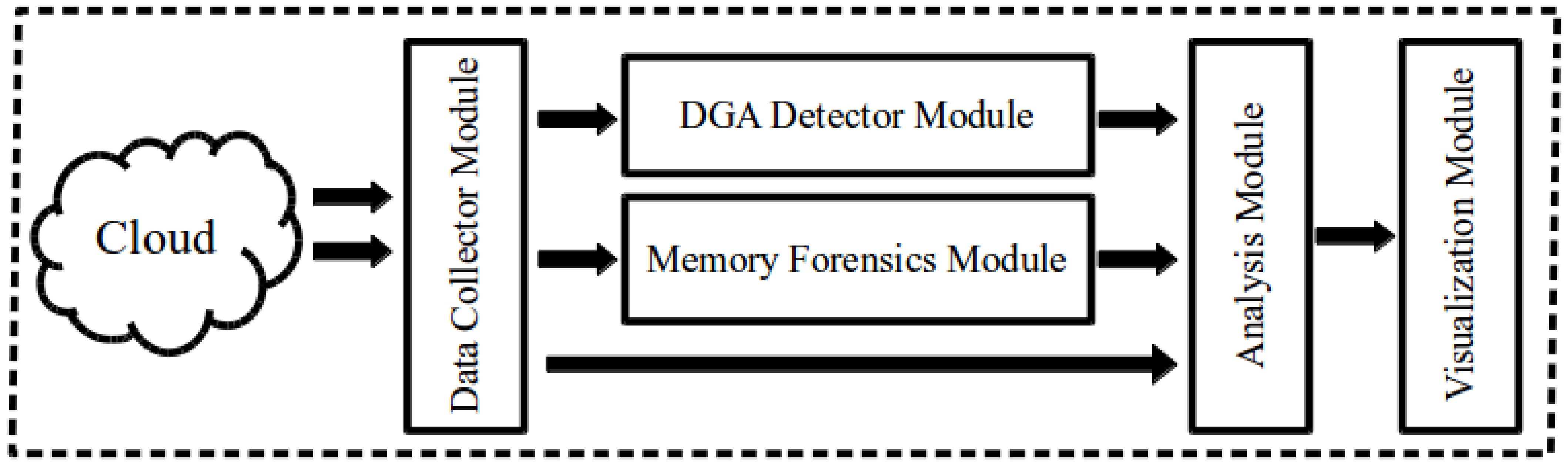
DGA Detector Module (DDM)

Families	Necrus	ZeusBot	Cryptolocker	Torpig	Symmi	Ranbyus	All
Number of samples	2048	1000	1000	20	64	40	4172
True positive	0.847	0.998	0.982	1.00	0.67	1.00	0.923
Some false negatives	lowusoheu.com oxegnusaen.com cuprybmeatskye.org keygtobetheld.es						
Some false positives	aljazeera.tv peyvandha.ir munrvscurlms.com uludagsozluk.com						

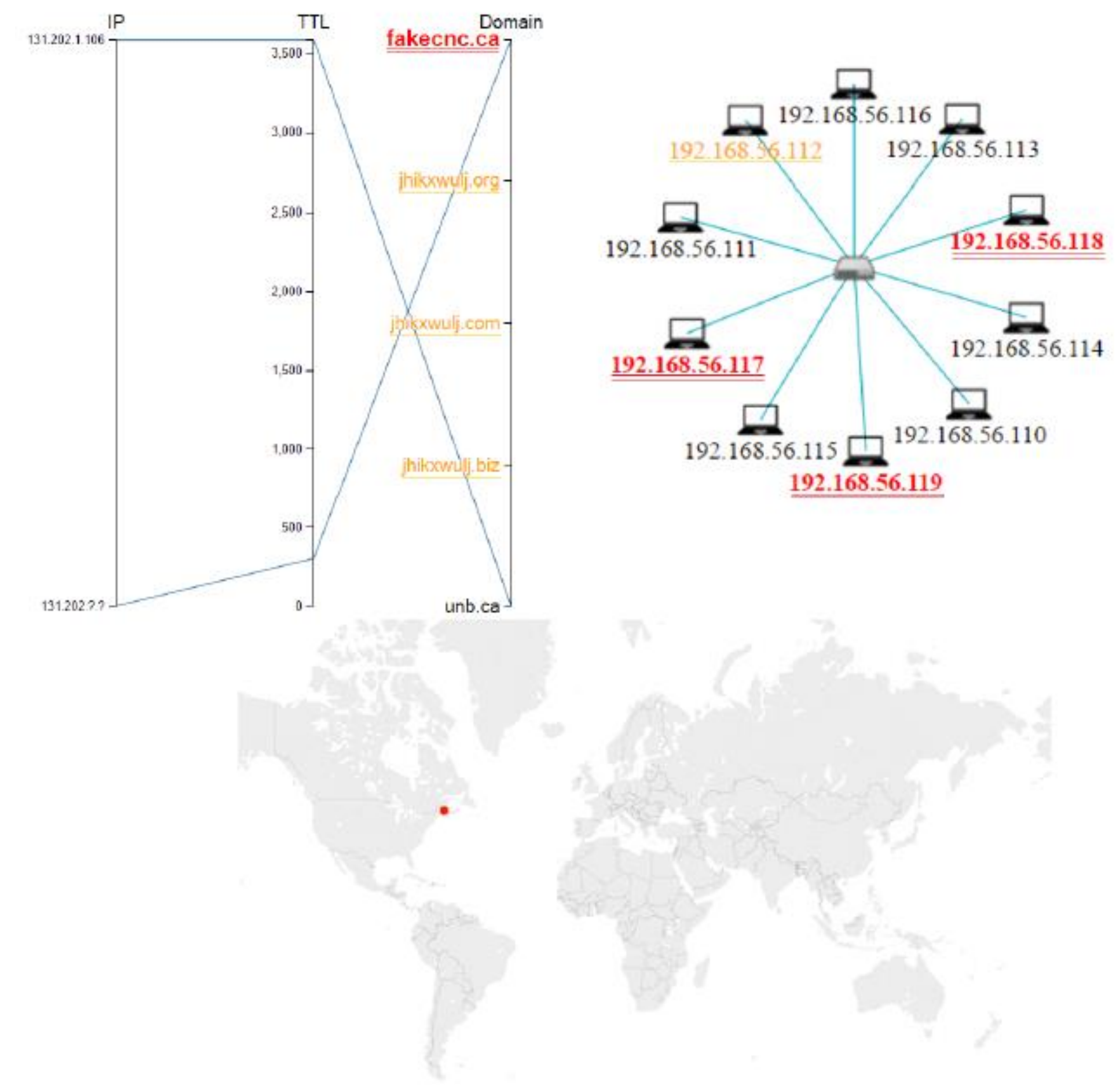
Analyze Module (AM)

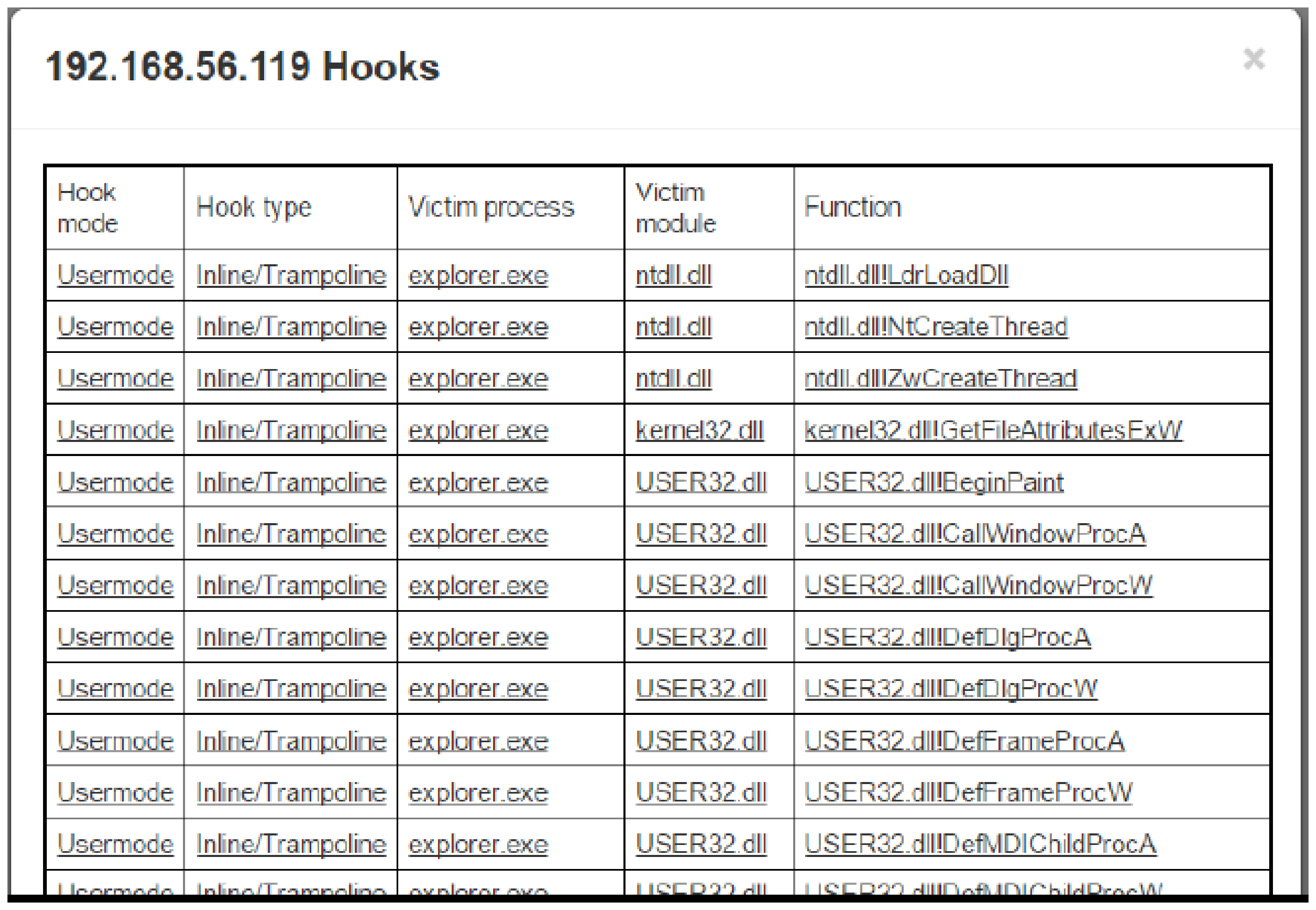
This module is responsible for detecting suspicious hosts and domains by processing the information gathered by the MFM and the DDM. The AM detects suspicious hosts based on abnormal hooks. It uses K-Means clustering algorithm to cluster hosts based on the hooking behaviors.

System Architecture



Visualization Module (VM)





192.168.56.119 Hooks ×

Hook mode	Hook type	Victim process	Victim module	Function
Usermode	Inline/Trampoline	explorer.exe	ntdll.dll	ntdll.dll!LdrLoadDll
Usermode	Inline/Trampoline	explorer.exe	ntdll.dll	ntdll.dll!NtCreateThread
Usermode	Inline/Trampoline	explorer.exe	ntdll.dll	ntdll.dll!ZwCreateThread
Usermode	Inline/Trampoline	explorer.exe	kernel32.dll	kernel32.dll!GetFileAttributesExW
Usermode	Inline/Trampoline	explorer.exe	USER32.dll	USER32.dll!BeginPaint
Usermode	Inline/Trampoline	explorer.exe	USER32.dll	USER32.dll!CallWindowProcA
Usermode	Inline/Trampoline	explorer.exe	USER32.dll	USER32.dll!CallWindowProcW
Usermode	Inline/Trampoline	explorer.exe	USER32.dll	USER32.dll!DefDlgProcA
Usermode	Inline/Trampoline	explorer.exe	USER32.dll	USER32.dll!DefDlgProcW
Usermode	Inline/Trampoline	explorer.exe	USER32.dll	USER32.dll!DefFrameProcA
Usermode	Inline/Trampoline	explorer.exe	USER32.dll	USER32.dll!DefFrameProcW
Usermode	Inline/Trampoline	explorer.exe	USER32.dll	USER32.dll!DefMDIChildProcA
Usermode	Inline/Trampoline	explorer.exe	USER32.dll	USER32.dll!DefMDIChildProcW

Conclusion

- The proposed framework "BotViz", provides a hybrid visual approach for botnet detection in small to medium size networks.
- The first botnet visualization tool which uses suspicious hooks on the hosts to empower its botnet detection algorithm.

Future Work

- Considering more host-based features such as adding the ability to detect process injection in the MFM and the AM.
- Adding more dictionaries to the DDM can increase the rate of detection in the DDM.