The 13th IEEE International Conference on Dependable Systems, Services and Technologies, DESSERT'2023
13-15 October, 2023, Athens, Greece

Threats Detection and Analysis Based on SYSMON Tool

Nataliia Bahniuk
Department of Computer Engineering and Cyber Security
Lutsk National Technical University
Lutsk, Ukraine
bahniuk_nataliia@lutsk-ntu.com.ua

Linchuk Oleksandr
Department of Computer Engineering and Cyber Security
Lutsk National Technical University
Lutsk, Ukraine
alex.stepit@gmail.com

Bortnyk Kateryna
Department of Computer Engineering and Cyber Security
Lutsk National Technical University
Lutsk, Ukraine
katerina.bortnyk@gmail.com

Kondius Inna
Dean of the Faculty of Computer and Information Technologies
Lutsk National Technical University, Lutsk, Ukraine
Innastk@ukr.net

Melnyk Kateryna
Department of Computer Engineering and Cyber Security
Lutsk National Technical University. Lutsk, Ukraine
ekaterinamelnik@gmail.com

Kostiantyn Kondius
Department of Computer Engineering and Cyber Security
Lutsk National Technical University. Lutsk, Ukraine
kondius_kostya@ukr.net

***Abstract* — In this work, an nalysis for the study of threats in a real environment with the possibility of conducting a full-fledged analysis of threats, as well as their simulationhas been developed for research purposes. Designed laboratory was built for the threats research, specification of deploying and configuring Sysmon, imitation of an attack in laboratory conditions and its investigation by implicit signs, the processesing of threat analysis using the Sysmon tool. We present a system based on the analysis of continuous input chan-nels of Sysmon logs. The system is based on the Cyber Threat Analysis Ontology and analyzes SYSMON logs to classify software according to different threat levels and enhance cyber defense capabilities with situational awareness, prediction and auto-mated actions. The developed laboratory improves the effectiveness of threat analysis using the Sysmon tool, makes study of threats, deploying and configuring Sysmon, imitation of an attack in laboratory conditions and its investigation by implicit signs. It can be applied for the study of threats in a real environment with the possibility of conducting a full-fledged analysis of threats, as well as their simulation for research purposes.**

Keywords — SYSMON tool, Threats Detection, Cyberattacks

I. INTRODUCTION

Nowadays, the community is witnessing cyberattacks during which malware in-fects a machine and it quickly diffuses within the network infrastructure of an organi-zation. Its propagation, whether undiscovered or untreated can provoke catastrophic events for the systems as a whole.

For effectively dealing with such incidents in a prompt manner, an effective and targeted methods and approaches the threat is remediated is a necessity. The log evidence and damage related to the attack's also should be collected and analyzed, as the basis for further effective detect and incidents response policy.

The threat actors use a diverse and extensive set of tactics, methods, and proce-dures to achieve their goals. In response, organizations are establishing threat intelligence programs to improve their defenses and reduce risk.

Therefore Microsoft try to improve the software devel-opment process, create software with a high level of security, and meet modern secu-rity requirements. It continuously implements security and privacy guidelines early in the lifecycle of all software development processes therefore the creation new effec-tive attacs detection and treatment instruments are actual especially in the conditions of the Russo-Ukrainian war to cope with malicious tactics and techniques on real-life observations considering the exponential occurrence of events and attack methods.

A large amount of research in the field of examination of attacks, the rule-based policy, successful incidence response in an actual research prpblem. The methodolo-gy regarding event-driven identification of the most impactful techniques through Sysmon is investigated in serie of works, among them [1]. This work attempts to an-swer in a clear way the following key questions regarding the optimal initialization of the Sysmon tool, towards the identification of Lateral Movement in the MS Windows ecosystem. One of the first reseach conducting a full investigation concerning catego-ries of logs shown in [2].

Nowadays the known techniques based on Sysmon present-ed in [3] were executed against targeted clients and compromised servers, related to each attack method. This work gives the proposition of a log-oriented MS Windows regulatory policy, allowing the optimal information related to potential Lateral Movement (LM) methods.

The work [4, 5] was developed the Cyber Threat Intelli-gence Ontology, based on Cyber Threat Intelligence, authors proposed a data-driven threat classification methodology for four distinct threat categories such as high, medium, low and unknown. The proposed threat assessment methodology has a disadvantage is prone to long delays if big logging data take place on real-life scenarios. In [6] a large ammount of DLL files were analyzed from different tools namely, Mimikatz, PowerShell Empire, China Chopper, and HUC Packet Transmitter, to investigate the existence of differences of the specific files among various versions of the OS MS Windows.

They proposed tools based on a Dynamic Link Library (DLL)-oriented method for malicious files detection towards logs collected by Sysmon. The work in [7] based on advance persistent threat detection with aid of Mimikatz password stealing tool. The authors implemented Mutex memory objects together with the identification of Mimikatz related DLL files although Mimikatz tool can be deliberately

obfuscated by the adversary, thus evading identification. The researchers in [8,9] also use a DLL-oriented malware detection methodology with aid of analysis of malicious files, for example Cuckoo.

More improved method, was proposed in [6] using a list with the most commonly appeared DDL files related with malicious intrusion tools. Note, that the enhancement of the DLL detection policy with aid of using Syminish the false positive results. The ELK Stack was also used in [10,11] for analysis of smon event logs and massive log records, should improve the detection accuracy and identification of attacs.

Based on this overview, and making our proposed research we also take into ac-count that MS Windows' strong security foundation leverages Microsoft's Security Development Lifecycle (SDL) fix pack, support for product security standards and certifications, and Azure code signing. Existing approaches to detecting threats using Sysmon are proposed mainly using solutions focused on search engines.

So, this article presents a system based on the analysis of continuous input chan-nels of Sysmon logs. Therefore, in our paper, a laboratory for the study of threats in a real environment with the possibility of conducting a full-fledged analysis of threats, as well as their simulationhas been developed for research purposes.

Designed laboratory uses specification of deploying and configuring Sysmon, imitation of an attack in laboratory conditions and its investigation by implicit signs, the processesing of threat analysis using the Sysmon tool. It is worth noting that the example considered concrete techniques that often occur in real incidents, including targeted attacks - for their detection, it must to have properly configured standard capabilities of the operating system or free tools.

Active threat intelligence must be integrated into the security and event information management system, forming a threat analysis platform. A threat intelligence platform aggregates log data from mul-tiple disparate sources by deploying multiple collection agents and provides central-ized analysis and reporting of an organization's security events to detect malicious activity.

The system is based on the Cyber Threat Analysis Ontology and analyzes Sysmon logs to classify software according to different threat levels and enhance cyber defense capabilities with situational awareness, prediction and automated actions. So, the purpose of this work is to improve the effectiveness of threat analysis using the sysmon tool, we propose a laboratory was built for the study of threats; deploying and configuring sysmon, imitation of an attack in laboratory conditions and its investigation by implicit signs.

The proposed method has great perspectives to apply, it can be extended with aid of more advanced telemetry and special tools, since the development of EDR (End-point Detection and Response) class solutions, attackers' techniques are becoming more and more sophisticated and often allow them to bypass detection rules that use standard operating system audit capabilities and free tools. This work can be extended to incorporate using cloud and satellite communication facilities that can be resulting if nessesary in a targeted interruption of the satellite broadband services across the Ukraine territory in the war contions and other European countries.

This paper is organized as follows. In Section II the threats investigation using SYSMON are described, in Section III zero-day threats and IOC-based approach has been done. SYSMON and tool-based approach is developed in Section IV, SYSMON and TTPS-based approach are presented in Section V. Conclusion and recommendation for a future work are finally presented in the last Section.

II. Threats Investigation Using Sysmon

A. *Sysmon Windows system service*

System Monitor (Sysmon) is a device driver and Windows system service that, if installed on a system, remains resident across system reboots. It provides detailed information about possible network connections, creation processes that took place, and files creation time changes. It is assigned for monitor and log system activity in the Windows event log.

Therefore by collecting the events it generates using Windows Event Collection or SIEM agents and then analyzing them, it can be possible identify anomalous and/or malicious activity and conclude how hackers, computers users that try to gain unauthorized access to data, attackers and malware operate on a given network. In this paper, we design the architecture of the threat research labora-tory, (see Fig.1).

For this goal the ELK stack (Elastic, Logstash, Kibana) is used for collection, processing and analysis. To collect windows logs, the winlogbeat software is installed as a collector. In addition, the sysmon module is installed on the target host.

The target host is deployed in the proxmox hypervisor.

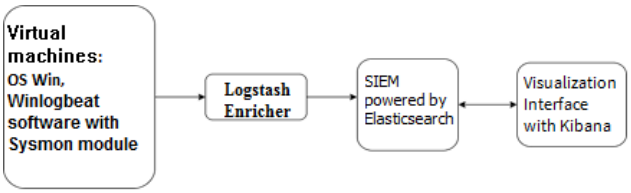

Fig. 1. The threat research laboratory architecture

The Sysmon module handles event log entries from the Sysinternals (Sysmon) sys-tem monitor, which is a Windows service and device driver that records system ac-tivity in the event log. Sysmon is not included with Windows or Winlogbeat and must be installed separately.

The default configuration file contains the configuration for Sysmon. If Sysmon is not installed, Winlogbeat will log a warning that it failed to read from the Microsoft-Windows-Sysmon/Operational channel. It will continue to read from other configured channels.

If you install Sysmon later, then you need to restart Winlogbeat to start reading from the channel. In this declared conditions, the module works on the basis of Sysmon v10 event manifests (according to the recom-mendations for winlogbeat version 7.17).

It contains a transformation for each of the defined event IDs.

Sysmon log collection

The following action steps were taken to prepare the sysmon log collection from the target host:

1. The winlogbeat-7.17 package is downloaded and installed;
2. The winlogbeat.yml file is configured;
3. The Sysmon is installed;
4. The elasticsearch is configured for receiving winlogbeat logs and their central-ized storage, configured template in kibana for visualization of the sysmon log event.

B. Using of Sysmon Methodology for Implicit Threats Investigation

Implicit threats are considered the threats that are not detected by the security software installed at the user's workplace. These include zero-day threats, malicious actions performed by the user knowingly or unknowingly, and attacks stretched over time. Under current investigation, we use the following Sysmon event codes:

- Event code 1: Process creation.
- Event code 3: Connecting to the network.
- Event code 8: Remote connection established.
- Event code 11: File creation.
- Event code 13: Registry event.
- Event code 22: DNS queries.

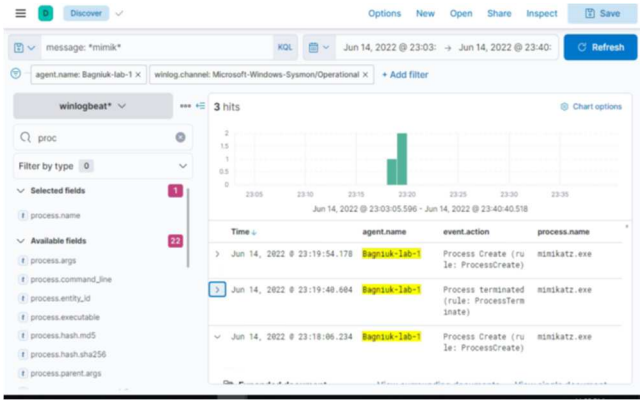

Fig. 2 – Mimikatz detection

Example 1. We consider an Advanced Persistent Threat (APT) style attack in a Windows environment and make the investigation under the following stages.

*Stage 1.*Consider an Advanced Persistent Threat (APT) style attack in a Windows environment.

When Mimikatz (see Fig.2) is detected in the first stage, it usually indicates the following three things:

- the attacker has gained a foothold in the network;
- the attacker managed to escalate privileges to run Mimikatz (meaning they reached the privilege escalation stage of the attack lifecycle);
- an attacker is most likely using Mimikatz to obtain credentials to attempt horizontal distribution across the infrastructure. At this stage of detection, Mimikatz can provide insight into where the attacker is in the lifecycle and what they might do next.

Stage 2. Further analysis revealed a malicious document that the user had opened through his mail. After the user accepted the security warning, the document executed a malicious macro that made a Windows call to WMI (WmiPrvSE.exe) to create an instance of Powershell (Powershell.exe). This instance of Powershell executes a script wrapped in a malicious macro. The Powershell program executes and returns to the attacker's server to download additional software. Now that the initial compromise vector is understood, assumptions can be made about the attacker's next actions.

Stage 3. For further investigation, the following query was made in our ELK in the search string "process.parent.name: wmiprvse.exe and process.name: powershell.exe".

Stage 4. Based on the results of the given query, several key-value pairs were manually selected that are important to create a basis for further search for threats. Next, it is always important to understand the software that generated this event log, which in this case is Sysmon.

Sysmon records all new processes created on the machine and flags these events.

Event Code: 1, which is an important piece of information for use in analysis.

The hostname for this machine is important because it can be used to further lookup this infected machine. Username for the account that was compromised to discuss with the user and possibly see if their account was used elsewhere in the environment.

For example, is the user accessing their Office 365 account from a new IP address, does the user account have access to internal resources, etc.?

The process ID can be used to trace all the activities that this malicious process is doing.

*Stage 5.*Next, we analyze the relationship between parent and child processes in Windows. This combination is not inherently harmful if WMI (WmiPrvSE.exe) creates a Powershell instance, but it is a technique used by attackers. Finally, the command line process confirms that this Powershell instance created by WMI is malicious. The command line string (powershell -noP -sta -w 1 -enc *) follows the common format used by frameworks used by attackers, such as Powershell Empire.

So, as a result conclusion, we can use the following search queries in the future work:

- *"powershell -noP -sta -w 1 -enc *"* – detect other instances of malicious powershell frameworks running on other machines in the environment;
- *"event.code: 1 and parent.process: wmiprvse.exe and process.name: powershell.exe"* – this query will not only find when Microsoft Office Word uses WMI to create Powershell instances, but also when other applications use the same technique;
- *"event.code: 11 and file.name: *"* – Sysmon event code 11 tracks file creation. If you know the name of the document, you can search in our environment for other users who have this document;
- *"event.code: 11 and "docm"* – Sysmon event code 11 tracks file creation. Documents with the file extension ".docm" support macros. In this way, we will find all documents with macro support in our environment.

We also conclude the following. In order for the attacker to maintain access to the network, he "implants" permanent mechanisms on infected machines. Some of the more common persistent mechanisms can take the form of scheduled tasks, adding a local user to the system, or adding autorun to the user registry. If the infected machine is rebooted, the attacker loses access because the contents of the memory are erased during the reboot. Persistent mechanisms allow attackers to maintain access through a reboot, user logout, loss of network connectivity, unexpected process termination, or security team interaction.

In our example, at earlier phase of the attack lifecycle detection a Powershell agent running on the machine was detected. Sysmon records all activities that occur on a machine, such as all activities performed by a specific process.

Besides, all malicious actions performed by the malicious Powershell process are also detected (chronological order of these events - from bottom to top).

First, a Powershell script packaged in a malicious Word document is executed, the program sends a DNS query for malwarelove.xyz, creates a Windows registry value, and creates a Windows scheduled task.

Unfortunately, Sysmon does not have an event code for newly created scheduled jobs, but it does monitor process execution, such as when a scheduled job's binary has been accessed via the command line, or when new files are created.

The first query looks for when Powershell (process.parent.name: powershell.exe) calls the Scheduled Task binary via the command line and tries to create a scheduled task. Specifically, the scheduled task example will create an instance of Powershell, read the content from the following registry key

“HKCU:\Software\Microsoft\Windows\CurrentVersion\debug,base64”,

decode the content from the registry key, and execute it.

The second query below looks for new files to be created in the path used to store scheduled tasks:

“C:\\Windows\\System32\\Tasks”.

Both queries lead to the same result:

- *“event.code: 1 and process.parent.name: powershell.exe and process.name: schtasks.exe and process.args: “/Create””;*

- *“event.code: 11 and “C:\\Windows\\System32\\Tasks””.*

Windows logon is another common technique used by attackers to maintain persistence.

The Windows logon element will be executed every time the user logs on.

When attempting to use this technique, Powershell creates a registry key in the following path: HKCU:\Software\Microsoft\Windows\CurrentVersion\Run*, which is the registry path for Windows logon items. The malicious Powershell code executes the contents of the same registry key used in the scheduled task above.

To confirm or deny this, you can run the following query: “event.code: 13 and “Software\\Microsoft\\Windows\\CurrentVersion\\Run””.

Double \\ symbols are required to escape the query, otherwise an error will occur.

C. New User Account

Creating a new user account is a common technique used by attackers to establish resilience. Unfortunately, Sysmon does not have an event code to track the creation of new user accounts. However, Sysmon tracks registry changes and when new accounts are created, they are added to the following registry path:

"HKLM\\SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\ProfileList".

No new account requests were detected, but such events should be looked for in order to confirm or deny the use of a particular technique.

More specifically, based on the data currently available to me in the Sysmon index and the time period indicated, there is no evidence of any new accounts being created.

So, from a SIEM perspective the account was not created, but in reality that may not be true. It is also important to understand that threat hunting is data-driven, so if there is poor or missing data, it is likely to produce incomplete or misinformed results that are used to make decisions.

Request:

“event.code: 12 AND registry.path: “HKLM\\SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\ProfileList””.

D. Additional searches

We execute the following additional searches.

“event.code: 1 and process.parent.name: powershell.exe and process.name: schtasks.exe and (process.command_line: “-W hidden” or process.command_line: “-W 1”) – find all scheduled tasks that Powershell executes in the background;

event.code: 12 and registry.key: "Software\\Microsoft\\Windows\\CurrentVersion\\Debug" - existence of registry key used by scheduled task and Winlogon event.

In the privilege escalation phase, the attacker tries to elevate his privileges from the normal user context to the administrator.

This is usually achieved through an elevation of privilege vulnerability in the local operating system on the local machine.

In our example, our logging did not record anything that could be used to detect privilege escalation at this point.

E. Network connections

In this phase, the attacker examines the network environment to gain an understanding of how users interact with the network. Ideally, attackers want to expose network services such as shared document storage or an internal server used to store documentation about the network (an internal wiki).

Accessing internal knowledge base tools (Sharepoint, Confluence, wiki, etc.) can be a goldmine of information.

For example, let's say an attacker tried to reset a database that stores credit cards.

An internal wiki contains documentation about these databases, such as the network subnet (location), how to access them (VPN, pass-through window, VLAN, configuration management, etc.), database type (Postgres, MySQL, Mongo, etc.), administrators (future purposes) that have access to servers or database backups that may not be as secure.

An attacker can also eavesdrop on network traffic to observe DNS queries, to determine where a DNS server is requesting a zone transfer, to learn how machines are being remotely managed (PsExec, Chef, Ansible) and which internal services users are interacting with.

The main goals are to understand the structure of the network, find interesting information, and see how users typically interact with the network and services. Each query returns results about hosts discovered, services used, environment information, services that can be used for horizontal migration, SMB shares, and user locations.

The first is an ARP scan on the local network to discover all hosts on the same subnet. Second, collecting information about the domain, such as the version of the Windows server, the forest domain, the IP address of the domain controller, the FQDN of the domain controller, and the LDAP schema for that domain.

Third, recursive DNS queries were performed on the domain controller to obtain a list of the hostnames of the machines in the domain. In fact, each hostname returned has a corresponding IP address, so the attacker knows the network location of each machine.

An attacker can also start a port scan to discover open ports on the network and then perform a service list scan. These port scans can provide a lot of information about the environment.

The list of services will allow us to identify web services, file shares, and potential services that an attacker can use for further migration. Sysmon only monitors OSI layer 3 traffic and ARP requests at layer 2.

Also searched for any commands using the ARP command and found none. Thus, using Sysmon, an analysis was made of a threat that was not determined by other means. Since this was done on a test lab machine in an isolated environment, results may vary in other environments.

III. Zero-Day Threats and IoC-Based Approach

When a zero-day threat appears, as a rule, compromise marks appear after a certain time, and based on them, a rule for sysmon can be created. The rule is created as a configuration file in xml format and then sysmon is started with the new configuration file. An example of a configuration file for the known attack (which was once in the zero-day stage) PrintNightmare CVE-2021-1675 (Figure 3.14) is shown on resource: https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-1675.

Hypothesis generation when using the IoC-based approach is based on the search for indicators of compromise in the protected infrastructure. In the case of this incident, such an indicator can be the IP address of the attacker's control server obtained from the Threat Intelligence database. Suppose we chose the indicator shown in Fig. 3 to work out the hypothesis

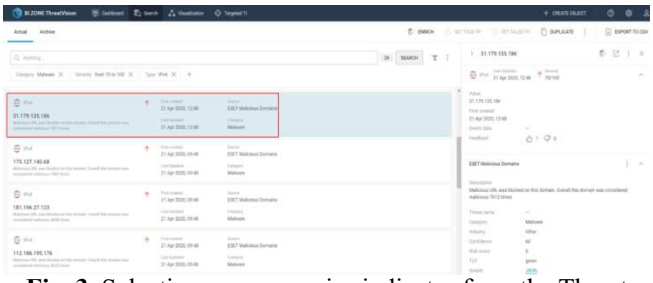

Fig. 3 Selecting a compromise indicator from the Threat Intelligence database

According to Threat Intelligence, the IP address 31.179.135.186 is used by malware. Now let's generate an initial hypothesis. It might declared like this: We have a compromised host or group of hosts in our infrastructure that made or continue to make connections to the malicious management server with IP address 31.179.135.186. Using the ELK platform and available telemetry, we will try to confirm or refute this hypothesis. The query results show that the host was making network connections to the malicious IP address on ports 443 and 8443. The hypothesis was confirmed - we can come to the disappointing conclusion that this host is compromised. Sysmon Event ID 3 and Windows Event ID 5156 events contain a field with the name of the process that created the network connection. Let's check which processes performed network connections to the malicious host. The first connection was made by the process C: Program Files (x86) Microsoft Office16 excel.exe, which is logical, since the incident started with the user opening a malicious Microsoft Excel attachment. A number of connections to the rundll32.exe process are also recorded.

As a result of testing the hypothesis, the incident was discovered. In addition to the IP address, other indicators such as sysprov32.dll files or userprep registry values can be used to detect this incident.

However, an attacker can easily change all the listed indicators, which will allow him to avoid detection by the IoC-based approach.

IV. SYSMON AND TOOL-BASED APPROACH

Now let's see how you can detect malicious activity within this incident using a Tool-based approach. This approach is based on highlighting the characteristic fea-tures of hacking tools, such as command lines, channels, PowerShell cmdlets, or net-work signatures.

Fig.4. Identifying specific cmdlets of the Invoke-Mimikatz utility

To steal user credentials, the attacker used the well-known Mimikatz utility and its PowerShell version Invoke-Mimikatz, which uses the reflective PE injection technique to download Mimikatz to the target host. As a Tool-based hypothesis, let's assume that Mimikatz or Invoke-Mimikatz utilities could be used in our infrastructure to dump user credentials. The search for specific command lines was performed by process start events (Windows Security Event ID 4688 and Sysmon Event ID 1), and PowerShell cmdlets - by Windows PowerShell log events (Event IDs 400, 800) and Microsoft-Windows-PowerShell/Operational (Event ID 4104). An example of a que-ry to test our hypothesis on the Threat Hunting platform is presented below on Fig.4.

The query detected the execution of cmdlets specific to the Invoke-Mimikatz utili-ty. Let's execute one more request that will allow us to check the presence of specific command lines of the Mimikatz utility in the process start events. Hypothesis testing revealed the use of the Invoke-Mimikatz and Mimikatz utilities on the target host. Judging from the command lines, the attacker performed a dump of local user credentials from the SAM database, as well as a dump of credentials from the LSASS process memory. As you can see, the Tool-based approach is quite reliable. To bypass such detection, an attacker would need to customize their favorite tools or abandon their use altogether.

V. SYSMON AND TTPS-BASED APPROACH

The TTPs-based approach is aimed at detecting the attacker's tactics, techniques and procedures or, in other words, his behavioral patterns. Therefore, when hunters use this approach, it will be most difficult for an attacker to avoid detection.

To work out the hypothesis, we integrate with the Threat Intelligence source. If events contain certain indicators of compromise (such as hash sums, IP addresses, domain names, email addresses, etc.), they can be enriched with a special tag such as malicious.

Our platform implements IP address enrichment based on Threat Intelligence for all Net-workConnection events, which allows you to test your hypothesis with the following simple query.

Fig. 5. Access to malicious hosts from Microsoft Office programs

Hypothesis testing revealed that the C:\Program Files (x86)\Microsoft Of-fice16\excel.exe process had network connections to the malicious host with IP ad-dress 31.179.135.18. Despite the similarity to the IP address detector described above, this approach has several advantages over it.

First, we do not use one specific indicator, but work with all indicators from the Threat Intelligence database. Second, even if the attacker's IP address was not in the TI platform database, we would have noticed that the office application was connecting to an external IP address. And this fact in itself deserves attention and requires careful analysis.

VI. CONCLUSIONS

A threat research lab architecture is developed and an Advanced Persistent Threat (APT) style attack in a Windows environment is considered. This laboratory can be used to study threats in a real environment with the possibility of conducting a full analysis of threats, as well as their simulation for research purposes.

It is worth noting that the example considered concrete techniques that often occur in real incidents, including targeted attacks - for their detection, it must to have properly configured standard capabilities of the operating system or free tools.

The proposed method has great perspectives to apply, it can be extended with aid of more advanced telemetry and special tools, since the development of EDR (Endpoint Detection and Response) class solutions, attackers' techniques are becoming more and more sophisticated and often allow them to bypass detection rules that use standard operating system audit capabilities and free tools. This work can be extended to incorporate using cloud and satellite communication facilities that can be resulting if nessesary in a targeted inter-ruption of the satellite broadband services across the Ukraine territory in the war con-tions and other European countries.

REFERENCES

[1] Smiliotopoulos, C.; Barmpatsalou , K.; Kambourakis, G. Revisiting the Detection of Lateral Movement through Sysmon. Appl. Sci. 2022, 12, 7746.

[2] Coordination, J. Detecting lateral movement through tracking event logs, June 2017.

[3] Russinovich, M.; Garnier, T. Sysmon v13. 22. Retrieved June 2021, 28, 2021.

[4] Mavroeidis, V.; Jøsang, A. Data-driven threat hunting using sysmon. Proceedings of the 2nd International Conference on Cryptography, Security and Privacy, 2018, pp. 82–88.

[5] Mavroeidis, V.; Bromander, S. Cyber Threat Intelligence Model: An Evaluation of Taxonomies, Sharing Standards, and Ontologies within Cyber Threat Intelligence. 2017 European Intelligence and Security Informatics Conference (EISIC), 2017, pp. 91–98. doi:10.1109/EISIC.2017.20.

[6] Matsuda, W.; Fujimoto, M.; Mitsunaga, T. Real-Time Detection System Against Malicious Tools by Monitoring DLL on Client Computers. 2019 IEEE Conference on Application, Information and Network Security (AINS), 2019, pp. 36–41. doi:10.1109/AINS47559.2019.8968697.

[7] El-Hadidi, M.G.; Azer, M.A. Detecting Mimikatz in Lateral Movements Using Mutex. 2020 15th International Conference on Computer Engineering and Systems (ICCES), 2020, pp. 1–6. doi:10.1109/ICCES51560.2020.9334643.

[8] Juwono, J.T.; Lim, C.; Erwin, A. A comparative study of behavior analysis sandboxes in malware detection. International Conference on New Media (CONMEDIA), 2015, p. 73.

[9] Narouei, M.; Ahmadi, M.; Giacinto, G.; Takabi, H.; Sami, A. DLLMiner: structural mining for malware detection. Security and Communication Networks 2015, 8, 3311–3322.

[10] Rajesh, P.; Ismail. Ismail. B, M.; Alam, M.; Tahernezhadi, M. Network Forensics Investigation in Virtual Data Centers Using ELK. 2021 International Symposium on Electrical, Electronics and Information Engineering, 2021, pp. 175–179.

[11] Jain, U.; others. Lateral movement detection using ELK stack. PhD thesis, University of Houston, 2018.