Honeypot-based Ransomware Detection as a Component of Security Posture Monitoring in Zero Trust Architecture*

Danyil Zhuravchak[1,*,†], Pavlo Hlushchenko[1,†] and Valerii Dudykevych[1,†]

[1] Lviv Polytechnic National University, Information Security Department, 12 Stepan Bandera str., 79000 Lviv, Ukraine

Abstract

This thesis explores the use of file-based honeypots for ransomware detection as a component of security posture monitoring in zero trust architecture (ZTA). Generally, this approach involves luring attackers away from critical systems using decoy systems, gathering valuable insights about attackers' behavior, tactics, techniques, and methods, and using this information to identify and prevent potential threats and enable early detection and response. Research papers and blog posts from cybersecurity experts support the effectiveness of this approach. The authors of these sources emphasize the early warning system provided by honeypots and an indirect way of ransomware detection using the extended Berkeley Packet Filter (eBPF) technology. Leveraging eBPF, the system achieves real-time monitoring of filesystem activity by intercepting malicious operations at the syscall level before execution thus reducing the mean time to detect (MTTD) and respond to ransomware incidents. This method not only detects potential threats but also contributes to understanding evolving ransomware strategies, enhancing security posture significantly. Overall, using honeypots for ransomware detection is seen as an effective and valuable approach to enhancing the security posture of organizations in a zero-trust architecture. The thesis includes a case study that demonstrates the effectiveness of honeypot-based ransomware detection in a zero-trust architecture. A detailed case study evaluates the deployment within a hypothetical organization, demonstrating the practical benefits of file-based honeypots in detecting and mitigating ransomware attacks. Testing with various ransomware families highlights the flexibility, accuracy, and covert nature of this approach, which resists detection by attackers. The results confirm that eBPF-based honeypots can operate with minimal system impact while providing robust defenses against real-world threats.

Keywords

cybersecurity, honeypot, zero trust architecture, ransomware detection, intrusion detection system, eBPF

1. Introduction

As cyber threats become more advanced and their number continues to grow, the need to update information security practices has never been more significant [1, 2]. Standards are not updated in time to respond to modern threats [3, 4].

Ransomware is a financially motivated cybercrime, where the individuals and organizations responsible are well-resourced and increasingly sophisticated. Ransomware attacks continue to grow in both frequency and sophistication, posing a significant threat to organizations of all sizes. Recent high-profile incidents resulted in payments of multi-million dollar ransoms [5] to recover encrypted data. Today, ransomware is among the major cybersecurity threats affecting individuals, businesses, and organizations daily. We have seen a huge rise in ransomware attacks: an 85% increase since 2022. A ransomware attack occurs every 2 seconds (4000+ a day) [6]. The average cost of a ransomware attack in 2022 was $812.000 with the average cost to recover from a ransomware attack being $1.85 million. The highest demand in 2021 was $70 million while the highest ransom paid was $3.2 million. This increase in sheer numbers and the continuous evolution of such kinds of attacks highlights the urgent need for organizations to adopt proactive and adaptable security measures. Beyond the financial losses, ransomware incidents often lead to

*CPITS 2025: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, February 28, 2025, Kyiv, Ukraine

* Corresponding author.

† These authors contributed equally.

danyil.y.zhuravchak@lpnu.ua (D. Zhuravchak); pavlo.k.hlushchenko@lpnu.ua (P. Hlushchenko); valerii.b.dudykevych@lpnu.ua (V. Dudykevych)

0000-0003-4989-0203 (D. Zhuravchak); 0000-0002-1262-5484 (P. Hlushchenko); 0000-0001-8827-9920 (V. Dudykevych)

severe operational disruptions, reputational damage, and cascading effects on supply chains and smaller businesses reliant on the affected organizations.

Another source [7] reports that 70% of companies paid to recover their data after being compromised in 2022. There were 236.1 million attacks in H1 2022 alone. The year 2022 saw a rise in the number of businesses that fell prey to ransomware attacks, with 71 percent of them reporting such incidents. This percentage was found to be the highest recorded figure so far and represented an increase from the previous five years. In each of the previous years, over half of the survey respondents reported that their employers had been victimized by ransomware, indicating a widespread threat across various industries.

The introduction of honeypots marks a pivotal development in cybersecurity, tracing its origins from basic decoy systems to the sophisticated, eBPF-based implementations of today. Honeypots have proven their value in real-world ransomware incidents by acting as early warning systems and providing invaluable insights into attacker behaviors. Modern implementations, particularly those leveraging eBPF technology, take this concept further by integrating directly into kernel-level operations. The eBPF framework allows for fine-grained monitoring of filesystem events, enabling the detection of suspicious activities such as unauthorized file access or modification. This capability is particularly critical in combating ransomware, which relies heavily on filesystem interactions to encrypt data. Additionally, eBPF's ability to operate with minimal system impact and its versatility in integrating with existing tools make it a powerful asset in honeypot design. By intercepting and analyzing syscall-level events, eBPF-based honeypots can proactively detect ransomware behaviors and initiate appropriate responses before significant damage occurs. These advancements underscore the critical role honeypots play in modern defense strategies, especially within the framework of Zero Trust Architecture (ZTA). Honeypot-based ransomware detection can be a highly effective component of security posture monitoring in zero trust architecture, as evidenced by several sources in the field of cybersecurity [8, 9].

A research paper by M. R. Amal and P. Venkadesh [10] examines the use of honeypot-based ransomware detection in conjunction with other security measures, such as intrusion detection and prevention systems. A honeypot is a security resource that is purposely designed to be explored, exploited, or hacked. Its value lies in its ability to be examined, attacked, and potentially exploited, which implies that we expect and intend for the system to be compromised. Honeypots are primarily used as a detection and reaction tool, and they have limited utility in prevention. They gather data and detect attack trends. This data is then used by defenders to construct stronger defenses and countermeasures against future security threats.

A blog post by cybersecurity firm Symantec [11] highlights the benefits of using honeypots for ransomware detection in a zero-trust architecture. According to the post, honeypots can provide an early warning system for potential threats and can help security teams identify and respond to attacks more quickly. The post notes that by creating a honeypot environment, organizations can test and refine their incident response plans, helping to ensure a quick and effective response in the event of a real attack. The post also provides practical guidance on how to set up honeypots for ransomware detection.

Kerman et al. [12] describes the advantage of zero trust architecture like this: "By protecting each resource individually and employing extensive identity, authentication, and authorization measures to verify a subject's requirement to access each resource, zero trust can ensure that authorized users, applications, and systems have access to only those resources that they need to access to perform their duties, not to a broad set of resources that all happen to be within the network perimeter." This way, the attacker has to compromise each resource and circumvent each security measure individually because if an attacker manages to gain access to one resource, it will not allow him to move laterally to the other resources.

On the other hand, there are some disadvantages [13] of ZTA such as the difficulty of upgrading or migration from legacy infrastructure, and the difficulty of deployment and management of such architecture.

In summary, using honeypots for ransomware detection can be a valuable component of security posture monitoring in zero-trust architecture. By creating decoy systems to lure attackers away from valuable systems, organizations can proactively identify and respond to potential threats, reducing the risk of a successful ransomware attack. Additionally, by creating a honeypot environment, organizations can test and refine their incident response plans, helping to ensure a quick and effective response in the event of a real attack. These benefits are supported by research and practical guidance from cybersecurity experts.

The **goal** of this thesis is to develop and evaluate a file-based eBPF honeypot network for detecting and preventing ransomware attacks within a zero-trust architecture and to provide recommendations for integrating honeypot-based ransomware detection into a comprehensive security posture monitoring strategy. This research aims to contribute to the development of effective, proactive defense mechanisms against ransomware attacks, particularly in the context of a zero-trust architecture, and to provide insights into the benefits and limitations of honeypot-based ransomware detection as a component of overall security posture monitoring.

The **objectives** of this research are:

1. To explore the current state of ransomware attacks and the threat they pose to organizations and the impact they can have on businesses.
2. To examine the concept of honeypots and their role in detecting ransomware.
3. To investigate the use of file-based eBPF honeypots as a means of detecting ransomware attacks. This would involve an analysis of eBPF, its features and capabilities, and how it can be used to detect suspicious activities within a file system.
4. To design and implement a file-based eBPF honeypot for detecting ransomware attacks. This would involve implementing the necessary eBPF-based monitoring and alerting features.
5. To evaluate the effectiveness of the proposed solution in detecting ransomware attacks within a zero-trust architecture, including its ability to detect ransomware and provide real-time security alerts.
6. To provide recommendations for integrating honeypot-based ransomware detection into a zero trust architecture security posture monitoring strategy, as well as for future research and development in the field.

2. Zero trust architecture as ransomware detection framework

Zero trust architecture plays a key role in providing a framework for implementing the honeypot-based ransomware detection approach. Zero trust architecture is a security model that assumes that no user, device, or application should be trusted by default, regardless of whether it is inside or outside the network perimeter. It operates on the foundational principle of “never trust, always verify”, ensuring that no user, device, or application is granted access without rigorous authentication and authorization [14]. Because the network location is no longer the main component in the security posture of the resource, ZTA focuses on protecting resources and not network segments. This approach addresses the limitations of traditional perimeter-based security models, which often fail to mitigate threats arising from compromised internal actors or lateral movement within a network. By enforcing identity verification and least privilege access controls, ZTA minimizes the attack surface and restricts potential damage from ransomware. A comparison between ZTA and traditional models reveals the stark contrast in their effectiveness; where legacy systems focus on securing boundaries, ZTA prioritizes individual resource protection, making it an ideal framework for deploying honeypot-based detection systems. Furthermore, ZTA’s granular access policies and robust segmentation capabilities align seamlessly with the operational needs of modern, dynamic organizations.

The honeypot-based approach leverages zero trust principles by assuming that all network traffic to the honeypots should be treated as potentially hostile because attackers will always find a

way to breach some parts of the organization. The approach uses decoy systems to lure attackers away from critical systems, thus distracting the attackers and giving the organization more time and opportunity for detection and prevention of unauthorized access, incident response, and reducing the potential impact of a ransomware attack.

Furthermore, the monitoring of file access with eBPF is a crucial part of this proposed component of zero-trust architecture. By monitoring file system events, organizations can quickly detect and respond to potential threats in real-time (after the syscall is called, but before it is executed), regardless of where they originate from.

Moreover, ZTA itself is a powerful mechanism that restricts the lateral movement and reconnaissance capabilities of the attacker compared to the traditional perimeter-based approach.

In summary, zero trust architecture provides the foundational principles and framework for a honeypot-based ransomware detection approach, allowing organizations to implement a proactive, multi-layered defense against ransomware attacks and reduce mean time to detect (MTTD) and mean time to respond (MTTR) to minimal levels. The logical components of a ZTA architecture [15] are described in Fig. 1.

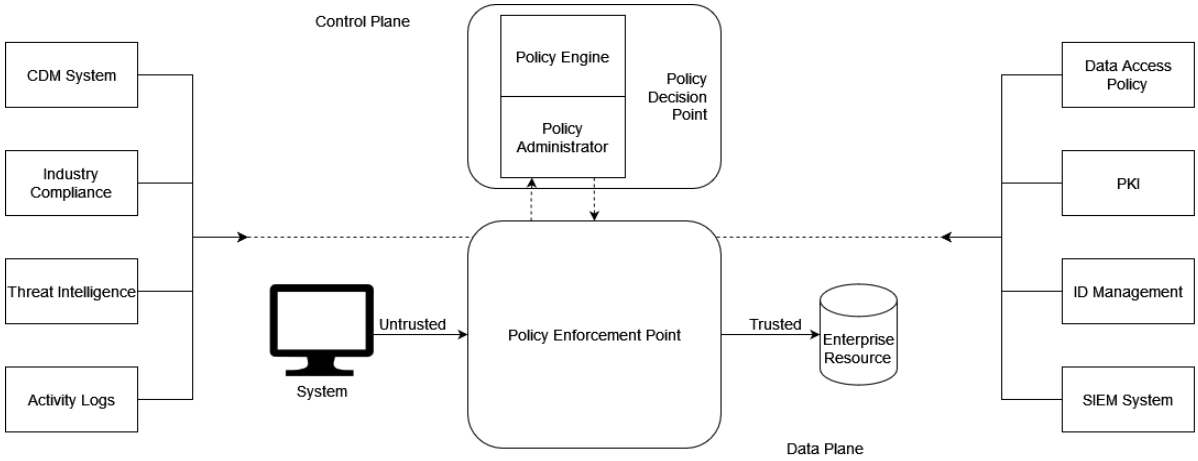

Figure 1: ZTA component diagram

This is a complex architecture that incorporates logically separate control and data planes, meaning control and data flow on separate networks. It also includes a multitude of sources from where additional information and context can be obtained to augment the decision-making specifically and security posture monitoring, access policies, PKI, and logging in general. The central component is the policy engine ultimately decides whether to grant access to a resource for a given subject based on the predefined policies.

For this paper, we greatly simplified the architecture and focused on the honeypot itself and the interaction between the honeypot and policy engine/enforcement point in the scope of security posture monitoring. The honeypot is based on eBPF technology [16], which allows the execution of user-defined programs in kernel space and allows attachment programs to syscall invocations. This particular feature is used in the honeypot to monitor the **open()** syscall which is used to open a file on a filesystem. This syscall is of particular interest to file-based honeypots because this syscall is the most frequently used by ransomware during its execution. After all, it needs to open the file to get its contents, encrypt it, and write it back to the disk.

Our proposed solution catches such ransomware at the moment of syscall invocation but before the syscall is executed. The high-level algorithm can be described in Fig. 3. depicting the operational workflow of the honeypot system. It begins with honeypot initialization and proceeds to monitor access attempts to honeypot files. If the accessing process is not in the allowlist, a security event is generated, and information is submitted to the policy enforcement point. The architecture of the proposed honeypot solution in conjunction with the policy engine is described in Figure 4. The

first version of the honeypot heavily relies on the popular frameworks and tools built around eBPF, such as BCC [17], bpftrace [18], and bpftool [19].

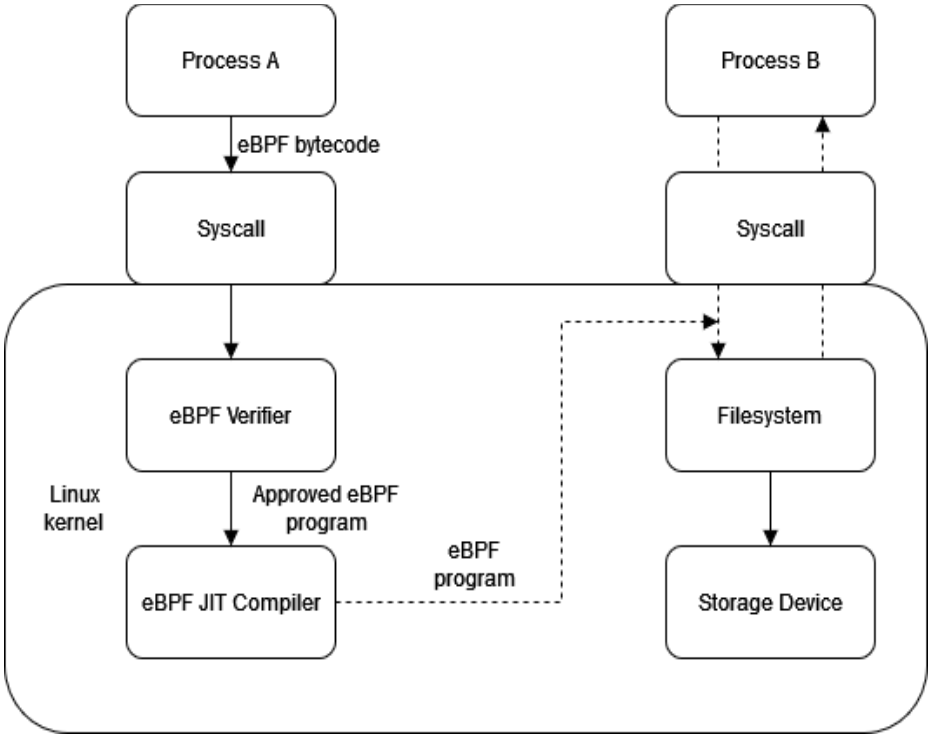

Figure 2: Conceptual eBPF diagram.

A high-level illustration of the role of eBPF within the Linux kernel. It shows how eBPF bytecode from processes is verified, compiled, and executed within the kernel, enabling syscall monitoring and interaction with the filesystem and storage devices. This process forms the backbone of the eBPF-based honeypot's functionality.

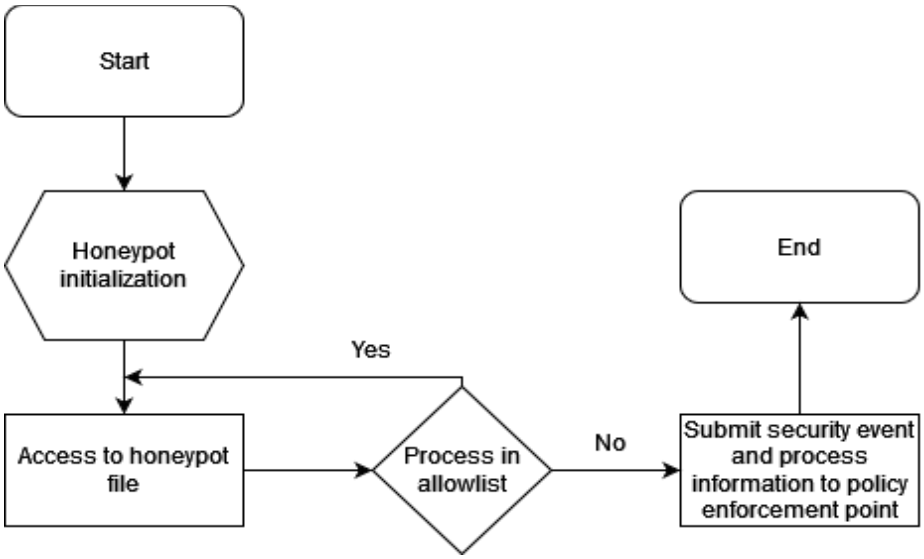

Figure 3: eBPF honeypot flowchart

The method that ransomware uses to get access to the endpoint is not discussed because our focus is the effectiveness of the detection of ransomware assuming it already compromised the endpoint.

The selection of appropriate ransomware samples for testing honeypot-based ransomware detection systems is critical to ensuring the effectiveness and reliability of the system. To this end,

the research identified some well-known and widely used ransomware families that had been observed in the wild and selected samples from each family for testing.

However, running these ransomware samples in a controlled environment for testing purposes was not always straightforward. Many modern ransomware strains have built-in "sandbox detection" features that allow them to detect when they are running in a virtualized or emulated environment, and to alter their behavior accordingly. This makes it difficult to accurately simulate a real-world attack scenario in a lab environment, and to evaluate the effectiveness of honeypots in detecting and preventing these attacks.

To overcome this challenge, we used a combination of techniques to evade the sandbox detection features of the ransomware samples, including modifying the virtualization environment to appear more like a real system and running the samples on real hardware rather than in a virtualized environment. In addition, the research also used different ransomware samples to ensure that the honeypot was able to detect a wide range of attack types and that the results were representative of real-world ransomware attacks.

Despite these challenges, the research was able to successfully test the honeypot network against a range of ransomware samples and demonstrate its effectiveness in detecting and preventing these attacks. This highlights the importance of rigorous testing and evaluation of honeypot-based ransomware detection systems and the need for ongoing research and development to stay ahead of the constantly evolving threat landscape.

For the testing, we selected a few recent ransomware samples: CryptoLock, AIRad, and DIMAQS.

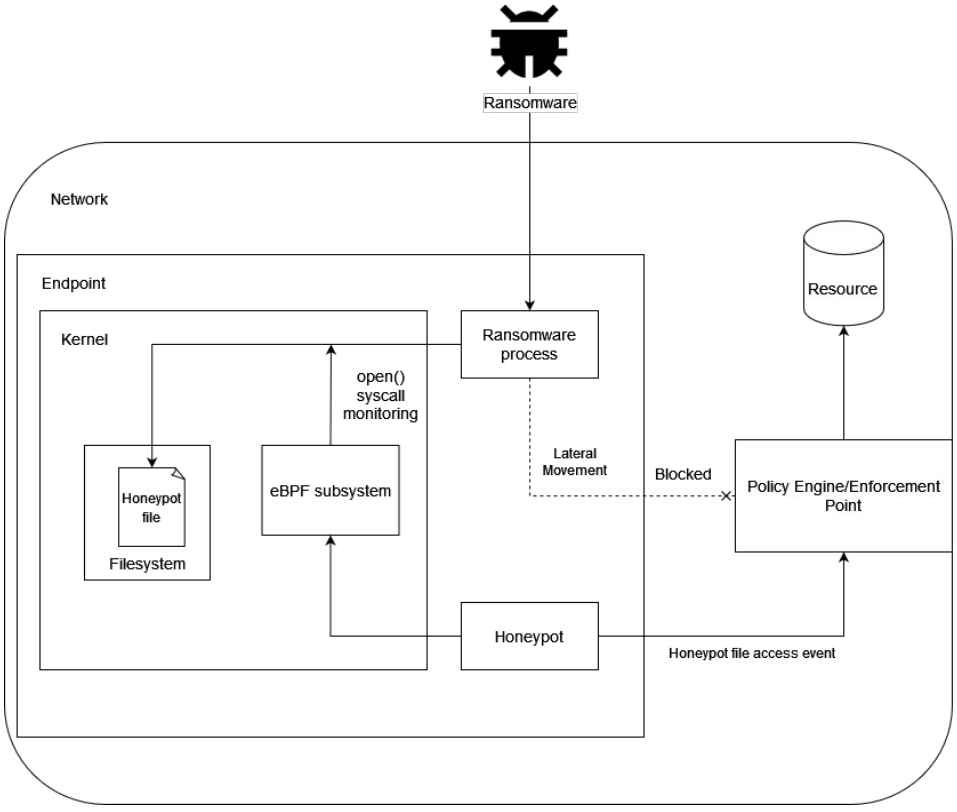

Figure 4: eBPF honeypot architecture in the scope of ZTA. Describes the flow of ransomware detection after the malicious process starts to run in the target system

Table 1
Ransomware detection testing results

Ransomware	Detection rate	False positive rate

CryptoLock	100%	0.05%
AIRad	100%	0.067%
DIMAQS	100%	0.5%

Note that while the results in Table 1 look very promising, they come with a warning. The honeypot does not employ sophisticated behavioral or IOC-based detection algorithms. It is rather simple in its operation. It monitors the *open()* syscall invocations, checks if the file name of the file that is being opened matches the name of the honeypot file and if yes—sends an event to the policy engine. This means that the detection rate is not 0%, but can be higher if some other legitimate processes (e.g. file system indexing process for search capability) or actual users access this file. That said, there is an allowlist that can be defined to exclude certain processes from triggering the security event thus bringing the false positive rate back down. This configuration step needs a prior inventorization performed on the host to identify and exclude such legitimate processes.

Detecting the presence of a honeypot on an endpoint can be a formidable task for ransomware. Honeypots in general are designed to simulate the behavior of legitimate systems, rendering them difficult to differentiate from actual processes or services that typically run on production systems. eBPF-based honeypots also are hard to detect with high confidence, because the same toolset employed for our honeypot can be used for application/system monitoring, observability, and general troubleshooting by a system administrator. Although it is possible to list all the programs loaded into the kernel, it is highly unlikely that a program will be able to tell with high confidence that a particular eBPF program is a honeypot because it looks like hundreds of other legitimate programs and tools that listen to syscall invocations run by either the system, the application or the administrators. For example, here is the output of *bpftool* command that can be used to list currently loaded eBPF programs:

Table 2
Comparison of bpftool output for legitimate system program and the honeypot

Probe attached by the system	Probe attached by the honeypot
{	{
"id": 33,	"id": 53,
"type": "cgroup_device",	"type": "tracepoint",
"tag": "03b4eaae2f14641a",	"name": "sys_enter_opena",
"gpl_compatible": true,	"tag": "9081693d56ded011",
"loaded_at": 1676542285,	"gpl_compatible": true,
"uid": 1000,	"loaded_at": 1676558452,
"bytes_xlated": 296,	"uid": 0,
"jited": true,	"bytes_xlated": 528,
"bytes_jited": 166,	"jited": true,
"bytes_memlock": 4096,	"bytes_jited": 315,
"map_ids": [1]	"bytes_memlock": 4096,
}	"map_ids": [6]
	}

There is a mention of *sys_enter_openat* [20] syscall (which is a hook into *openat()* syscall entry tracepoint which is in turn called upon an invocation of *open()* function by a user-space program), but it looks like any other program that monitors files (for example, for logging purposes). Therefore, having this information about a program is insufficient to decide that this is a honeypot.

Another advantage of eBPF-based security applications, in general, is their non-intrusive and lightweight operation that does not require code changes (which is good for integration) and does not have a noticeable effect on performance [21].

There are a couple of disadvantages in this first version of our solution as well. While it is good enough to demonstrate the capabilities of the technology it is based on, it does not fully leverage all of the capabilities of the eBPF subsystem yet. Integrating the data about multiple aspects of the malicious process and correlating them with statistical or behavioral analysis would be an interesting topic to explore and it is certainly something we keep in mind for further development. It also does not do prevention yet due to the limitations imposed by the kernel on the eBPF subsystem. But this can be done with other means (such as seccomp-bpf filters or a kernel module) that can be utilized based on the process information provided by the detection module.

Ransomware attacks continue to pose a significant threat to organizations of all sizes, making it essential to have effective defenses in place. The use of honeypots, which are decoy systems designed to lure in and detect attackers, can be a valuable tool in detecting and analyzing ransomware attacks. The effectiveness of honeypots in detecting ransomware depends on the sophistication and techniques employed by the honeypot. Advanced honeypots that use passive and indirect detection techniques and operate covertly can be particularly challenging for ransomware to detect, making them an effective tool for detecting and analyzing ransomware attacks. When combined with file system monitoring using eBPF, honeypot-based ransomware detection can provide a powerful defense against and early notification of ransomware attacks. The eBPF program can capture low-level file system events and trigger alerts or responses when suspicious activities are detected. By combining honeypots with file system monitoring, organizations can create a multi-layered defense that can detect and respond to ransomware attacks in real-time.

However, it is important to note that the proposed approach is not a silver bullet and should be used in combination with other security measures. Honeypots and eBPF-based monitoring are only a single component among the various systems needed for ensuring operational and security qualities in zero-trust architecture. Other techniques such as backup and recovery, antivirus software, firewalls, SIEMs, IDS/IPS, incident response, and threat hunting are also critical to a comprehensive defense against ransomware attacks.

One of the key benefits of honeypots coupled with modern technologies such as eBPF is the inherent difficulty of detecting them. Passive, indirect, and non-interactive detection techniques allow honeypots to avoid any network traffic or system calls that could typically make the ransomware believe it is being analyzed or detected. This characteristic makes it highly unlikely for ransomware to identify the presence of the honeypot with high confidence and without leaving a trace itself.

Furthermore, the proposed approach can be further developed, optimized, and extended for specific environments and use cases. For example, the Python wrapper for eBPF can be modified to support additional features such as machine learning, and statistical or behavioral algorithms for more advanced threat detection and correlation with other security information that can also be obtained by leveraging the eBPF subsystem. By leveraging anomaly detection models, honeypots could identify previously unseen ransomware behaviors, enabling more proactive threat mitigation. Additionally, real-time data analysis through distributed computing frameworks could significantly improve the responsiveness of the system, ensuring rapid containment of potential threats.

Optimization of eBPF programs is another critical area of focus. By fine-tuning the performance of eBPF bytecode and minimizing resource overhead, organizations can ensure seamless deployment in resource-constrained environments, such as edge computing or IoT devices. Furthermore, incorporating adaptive honeypot mechanisms that dynamically adjust decoy configurations based on attacker behaviors could increase the effectiveness of these systems in luring sophisticated threats. This comes with an additional challenge due to the hard constraints

imposed on the eBPF programs and the increasing complexity of the proposed measures to implement.

The scalability of honeypot deployments also remains a key consideration. Future work could explore the development of centralized management platforms for deploying and monitoring large-scale honeypot networks. These platforms could provide comprehensive dashboards, automated policy enforcement, and integration with incident response tools, streamlining the overall security posture of organizations.

In future work, we will conduct a comprehensive review of already existing file-based honeypots and compare their effectiveness and differences in detection approaches with our proposed solution as this was not an objective for this research project.

Beyond ransomware detection, eBPF-based honeypots have the potential to address a wide range of cybersecurity challenges. For example, they could be adapted for detecting insider threats by monitoring unusual access patterns to sensitive files or databases. Similarly, in the context of cloud environments, these honeypots could integrate with cloud-native security frameworks to provide additional layers of defense against lateral movement and privilege escalation.

Another area of interest would be the security of decentralized databases in a zero-trust environment based on the findings and comparative analysis by Petriv et al. [22].

Drawing from our findings, experience, and the reviewed related works, we recommend integrating various kinds of honeypots into the ZTA of an organization to serve as early warning sensors that can help filter out the noise and, in some, cases, even reconstruct the actions of a malicious actor inside the network and systems of an organization. This applies to honeypots in general and not only to the file-based honeypots reviewed in our research. Also, the more tuning and customization for the behavior of processes, systems, services, and networks the better.

Conclusions

The continuous evolution of ransomware necessitates equally adaptive and innovative defense mechanisms. Honeypots, particularly those utilizing eBPF technology, represent a critical component of modern cybersecurity strategies. Their ability to detect and analyze ransomware activities covertly, coupled with the foundational principles of Zero Trust Architecture, offers organizations a robust and proactive approach to threat mitigation. However, ongoing research and development are imperative to stay ahead of attackers who continuously refine their methods. Future integrations with advanced machine learning models hold promise for enhancing the predictive capabilities of honeypot systems, enabling organizations to anticipate and neutralize threats before they materialize. With the introduction of AI/ML models and LLMs into the organization's technical landscape, it is important to also maintain compliance according to the standard the organizations often must adhere to [23].

This thesis has explored the practical honeypot-based ransomware detection as a component of security posture monitoring within a zero-trust architecture. The research reviewed the current state of ransomware attacks and their impact on organizations, highlighting the limitations of traditional security measures in detecting and preventing these attacks.

Through an investigation of honeypots and their capabilities in detecting and preventing ransomware attacks, the research demonstrated the potential of file-based eBPF honeypots as an effective tool in the detection of these attacks, particularly within the context of a zero trust architecture.

The thesis presented a file-based eBPF honeypot, which was implemented and evaluated in a lab environment. The results showed that the honeypot network was able to detect various forms of ransomware attacks and provide real-time security alerts.

While honeypots are not a silver bullet and should be used in conjunction with other security measures, the research suggests that they can be an important and effective tool for defending against ransomware attacks, particularly when deployed as part of a comprehensive security posture monitoring strategy within a zero trust architecture.

Based on the research findings, the thesis provides recommendations for integrating honeypot-based ransomware detection into a zero-trust architecture security posture monitoring strategy, as well as suggestions for future research and development in the field.
Overall, the thesis makes a valuable contribution to the field of cybersecurity by demonstrating the potential of honeypot-based ransomware detection as part of a comprehensive security posture monitoring strategy, and by providing insights into the benefits and limitations of this approach within the context of a zero-trust architecture. Honeypot-based ransomware detection not only strengthens an organization's security posture but also provides a flexible and scalable solution for adapting to the ever-changing threat landscape. By integrating these systems into a comprehensive Zero Trust framework, organizations can achieve a resilient and forward-looking defense against one of the most pressing cybersecurity challenges of our time.

Acknowledgments

The authors would like to thank the Armed Forces of Ukraine for providing security to perform this work. This world has become possible only because of the resilience and courage of the Ukrainian Army. Additionally, we recognize the initial work by R. Ward and B. Beyer [24] for the huge amount of effort put into making practical use of the theoretical concepts of zero trust and their following works [25, 26].

Declaration on Generative AI

While preparing this work, the authors used the AI programs Grammarly Pro to correct text grammar and Strike Plagiarism to search for possible plagiarism. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the publication's content.

References

[1] S. Vasylyshyn, I. Opirskyy, V. Susukailo, Analysis of the use of software baits as a means of ensuring information security, in: 2020 IEEE 15th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2020, 2, 2020, 242–245.
[2] S. Vasylyshyn, I. Opirskyy, S. Shevchenko, Honeypot security efficiency versus deception solution, in: Cybersecurity Providing in Information and Telecommunication Systems, vol. 3188, 2021, 229–236.
[3] V. Susukailo, I. Opirsky, O. Yaremko, Methodology of ISMS establishment against modern cybersecurity threats, in: Future Intent-Based Networking. Lecture Notes in Electrical Engineering, vol. 831, 2022. doi:10.1007/978-3-030-92435-5_15
[4] T. Fedynyshyn, O. Mykhaylova, I. Opirskyy, Security implications of mobile development frameworks: Findings from static analysis of Android apps, in: IEEE 17th Int. Conf. Adv. Trends Radioelectron. Telecommun. Comput. Eng., 2024, 444–448.
[5] Information is beautiful: ransomware-attack. URL: https://informationisbeautiful.net/visualizations/ransomware-attacks/
[6] Norton Company Blog, Ransomware statistics: 102 facts and trends you need to know in 2023. URL: https://us.norton.com/blog/emerging-threats/ransomware-statistics
[7] Statista, Annual number of ransomware attacks worldwide from 2016 to first half 2022. URL: https://www.statista.com/statistics/494947/ransomware-attacks-per-year-worldwide/
[8] P. Skladannyi, et al., Improving the Security policy of the distance learning system based on the zero trust concept, in: Cybersecurity Providing in Information and Telecommunication Systems, vol. 3421 (2023) 97–106.
[9] R. Syrotynskyi, et al., Methodology of network infrastructure analysis as part of migration to zero-trust architecture, in: Cyber Security and Data Protection, vol. 3800 (2024) 97–105.

[10] M. R. Amal, P. Venkadesh, Review of cyber attack detection: Honeypot system, Webology, 19(1) (2022) 5497–5514. doi:10.14704/WEB/V19I1/WEB19370
[11] Broadcom documentation: Symantec zero trust framework. URL: https://docs.broadcom.com/doc/symantec-zero-trust-framework
[12] A. Kerman, et al., Implementing a zero trust architecture, National Institute of Standards and Technology (NIST), 2020.
[13] E. Bertino, Zero trust architecture: does it help? IEEE Security & Privacy, 19(05) (2021) 95–96. doi:10.1109/MSEC.2021.3091195
[14] D. Shevchuk, Designing secured services for authentication, authorization, and accounting of users, in: Cybersecurity Providing in Information and Telecommunication Systems II, vol. 3550, 2023, 217–225.
[15] S. Rose, et al., Zero trust architecture, National Institute of Standards and Technology, 2020. doi:10.6028/NIST.SP.800-207
[16] L. Deri, et al., Combining System Visibility and Security Using eBPF, in: Italian Conference on Cyber Security, vol. 2315, 2019.
[17] Github, IO Visor Project: BPF Compiler Collection (BCC). URL: https://github.com/iovisor/bcc
[18] Github, IO Visor Project: bpftrace. URL: https://github.com/iovisor/bpftrace
[19] Github, libbpf: bpftool. URL: https://github.com/libbpf/bpftool
[20] L. Torvalds: Linux BPF examples. URL: https://github.com/torvalds/linux/blob/master/tools/perf/examples/bpf/sys_enter_openat.c
[21] C. Cassagnes, et al., The rise of eBPF for non-intrusive performance monitoring, in: NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium, 2020, 1–7. doi:10.1109/NOMS47738.2020.9110434
[22] P. Petriv, I. Opirskyy, N. Mazur, Modern technologies of decentralized databases, authentication, and authorization methods, in: Cybersecurity Providing in Information and Telecommunication Systems II, vol. 3826, 2024, 60–71.
[23] O. Deineka, et al., Information classification framework according to SOC 2 Type II, in: Cybersecurity Providing in Information and Telecommunication Systems II, vol. 3826, 2024, 182–189.
[24] R. Ward, B. Beyer, BeyondCorp: A new approach to enterprise security, login, vol. 39(6), 2014.
[25] B. Spear, et al., Beyond Corp: The Access Proxy, login, vol. 41(4), 2016.
[26] B. Osborn, et al., Beyondcorp: Design to deployment at Google, 2016.