# Cryptography as a dual-faceted instrument of security and vulnerability

Maksim Iavich[1,†], Sergei Simonovi[1,†] and Tetiana Okhrimenko[2,*,†]

[1] *Caucasus University, 1 Paata Saakadze str., 0102 Tbilisi, Georgia*
[2] *National Aviation University, 1 Lubomyra Guzara ave., 03058 Kyiv, Ukraine*

## Abstract

Cryptography is an important field dedicated to securing data transmission through advanced algorithms and techniques. Everyday applications of cryptographic algorithms such as TLS for encrypted web traffic and Diffie-Hellman or RSA for secure remote server management showcase their critical role in protecting information. However, the same cryptographic techniques that protect data can also be misused by malicious actors for malicious purposes. This research focuses on analyzing the innovative applications of cryptographic methods in both safeguarding data and facilitating cyberattacks, emphasizing the dual-edged nature of these technologies in the evolving landscape of cybersecurity. The research underscores the critical need to recognize encrypted traffic as a significant threat and provides targeted recommendations for improving defensive and offensive strategies.

## Keywords

cryptography, IDS, DLP, penetration testing.

## 1. Introduction

In our everyday life, we utilize cryptographic methods and algorithms to secure the data and prevent eavesdropping. This can be achieved with various algorithms, for example: TLS, RSA, AES, and many more. With the help of cryptography, we can be almost sure that even if our data is stolen, it cannot be read by the threat actor. Though, this also works in the opposite direction: if the malicious user sends malicious data over the encrypted channel, the so-called blue team will struggle with identifying such traffic, as without decrypting the traffic, it is rather problematic to conclude, whether it is malicious indeed or not. The development of machine learning can help with the problem: the artificial intelligence is trained on the datasets and learns to identify the malicious encrypted traffic, though, if the attacker utilizes self-written encryption or obfuscation algorithm, the artificial intelligence will fail to spot it, as the data signature will be unmatched. This situation causes the dilemma: cryptography, a savoir of confidentiality, can be used as a double-edged sword to hide malicious traffic and data transfers. Cryptography can be used to hide the following attacks: web-based attacks, reverse shells and remote code execution, and data exfiltration. This paper discusses the effectiveness of network Intrusion Detection Systems (IDS) and Data Loss Prevention (DLP) tools against encrypted malicious traffic. The paper also discusses the nested encryption, and obfuscation techniques and their usage in penetration tests [1–3].

The objective of this research is also to conduct experimental evaluations to assess the effectiveness of defensive software in detecting and mitigating encrypted malicious traffic. Additionally, the study aims to explore the dual role of cryptography as both a defensive mechanism and a tool exploited by penetration testers and cybersecurity criminals. By examining how cryptographic techniques are employed in both safeguarding and attacking digital systems, this research seeks to provide insights into the strengths and limitations of current cybersecurity defenses in the face of sophisticated encryption-based threats [4]. The research also aims to show the critical need to recognize encrypted traffic as a significant threat and to provide targeted recommendations for improving defensive and offensive strategies.

## 2. Review of the literature

Cryptographic methods are pivotal for ensuring data security, employing algorithms such as Transport Layer Security (TLS), RSA, and Advanced Encryption Standard (AES) to protect information from unauthorized access [5, 6]. These techniques are essential in maintaining confidentiality and integrity in data transmission. Despite their effectiveness, encrypted communication poses significant challenges to network security, particularly in detecting malicious activities. Traditional IDS and DLP tools often struggle with encrypted traffic, as these systems require decryption to analyze the content, making detection of malicious activities complex [7, 8].

*CPITS-II 2024: Workshop on Cybersecurity Providing in Information and Telecommunication Systems II, October 26, 2024, Kyiv, Ukraine*
* Corresponding author.
† These authors contributed equally.
miavich@cu.edu.ge (M. Iavich);
s_simonovi@cu.edu.ge (S. Simonovi);
t.okhrimenko@nau.edu.ua (T. Okhrimenko)

0000-0002-3109-7971 (M. Iavich);
0009-0000-0124-2931 (S. Simonovi);
0000-0001-9036-6556 (T.Okhrimenko)

Machine learning and artificial intelligence have emerged as potential solutions to enhance detection capabilities. These technologies can identify patterns and anomalies in network traffic, even when encrypted [9]. However, their effectiveness is compromised when attackers employ custom encryption or obfuscation techniques, which can render traffic patterns unrecognizable and evade detection [10]. This illustrates a critical challenge: while cryptography ensures data confidentiality, it can also be leveraged to obscure malicious activities.

The use of cryptography in concealing cyberattacks is increasingly prevalent. Attackers can exploit encryption to hide various types of malicious activities, including web-based attacks, reverse shells, remote code execution, and data exfiltration [11]. Techniques such as nested encryption and obfuscation further complicate the detection and analysis of malicious traffic, presenting substantial challenges for defensive strategies [12]. To address these issues, research suggests integrating SSL certificates into IDS solutions, employing anomaly-based and Indicator of Compromise (IOC) detection, and deploying Endpoint Detection and Response (EDR) systems to enhance overall security [13, 14].

Recent studies have also explored obfuscation techniques that challenge malware detection and analysis. Techniques such as code obfuscation and metamorphism are used to hide malware from detection systems, complicating the analysis and remediation process [15–17]. These methods demonstrate how attackers can leverage encryption and obfuscation to enhance the stealth of their activities, further emphasizing the need for advanced defensive strategies.

The authors of the papers also emphasize the necessity for adaptive defensive and offensive strategies to keep pace with evolving cryptographic threats [18–21]. Recognizing encrypted traffic as a significant threat and developing targeted recommendations for improving detection and response capabilities are essential for strengthening cybersecurity posture in the face of sophisticated encryption-based threats [21–25].

## 3. Problem statement

The increasing number and complexity of cyber threats and the prevalence of cryptographic techniques in securing communications have created a significant challenge for cybersecurity protection measures. While encryption technologies such as TLS, RSA, and AES are essential for securing data from unauthorized access, they also can pose serious difficulties in the detection and analysis of malicious activities conducted over encrypted channels.

The main issue appears from the dual role of cryptography: while it protects legitimate data, it also allows attackers to obfuscate malicious payloads, rendering traditional IDS and DLP tools less effective. These systems often struggle to analyze encrypted traffic comprehensively, as they are unable to inspect the content without decrypting it. This limitation is exacerbated when attackers employ advanced techniques such as nested encryption or custom obfuscation algorithms, which further obscure the nature of the malicious traffic and complicate the reconstruction of attack sequences.

As a result, current cybersecurity measures face significant challenges:

1. Inadequate Detection: Traditional IDS and DLP systems frequently fail to identify malicious activities within encrypted traffic, leading to potential blind spots in network security.
2. Complex Analysis: Even when encrypted threats are detected, the difficulty in decrypting and analyzing the traffic impedes the ability to understand and mitigate attacks effectively.
3. Advanced Obfuscation: Attackers' use of nested encryption and proprietary obfuscation techniques introduces additional layers of complexity, making it difficult for security professionals to reconstruct attack timelines and assess the full scope of threats.

This problem needs a critical evaluation of how current defensive technologies must be improved to address these challenges. There is a serious need to create and implement advanced detection techniques that can efficiently handle encrypted malicious traffic and ensure that cybersecurity defenses can keep pace with evolving threats.

## 4. Laboratory

To validate the hypothesis regarding the effectiveness of encrypted malicious traffic, we constructed a virtual laboratory comprising several key components. The setup included:

- VirtualBox is the hypervisor, providing the virtualization environment necessary for the lab.
- Kali Linux serves as the attacker machine, equipped with tools for executing and managing attacks.
- pfSense 2.7.0 is configured as the router and firewall, facilitating network traffic management and security.
- Ubuntu 22.04.3 runs Suricata 6.0.4 as the network IDS and Damn Vulnerable Web Application (DVWA) as the target vulnerable software.

The network topology of this virtual laboratory is depicted in Fig. 1.

The goal is to emulate the global network environment, where the victim machine is located behind the NAT router, and the attacker machine is outside of the victim's local network. Kali Linux is a part of 192.168.1.0/24 network and has a network interface in a "bridged" mode.

The victim host is a part of 172.16.0.0/24 network and has a network interface in an "internal network" mode.

PfSense plays the role of a NAT router and is a part of both networks, having two network interfaces: WAN (192.168.1.150, works in a "bridged network mode") and LAN (172.16.0.1, works in an "internal network" mode).

To make a victim server reachable from 192.168.1.0/24 network, the port forwarding rules for ports 80 and 443 are added on PfSense. Also, "block private networks" checkbox is unchecked. All egress traffic is permitted, all ingress traffic, except port forwarding, is prohibited.

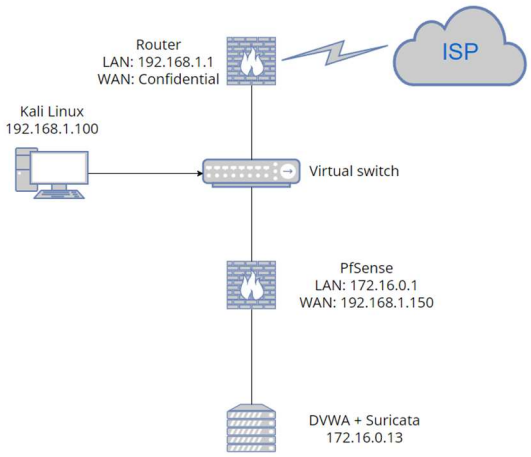


**Figure 1:** Virtual laboratory network topology.

The victim machine uses apache2 2.4.52 and Suricata 6.0.4. Suricata uses custom rules and the rules are taken from the following GitHub repository [26].

## 5. Experiments

In the virtual laboratory, the following experiments were executed:

- Running SQL injection over HTTP.
- Running SQL injection over HTTPS.
- Running reverse shell over the unencrypted socket.
- Running reverse shell over the encrypted socket.

The first experiment is running the SQL injection over HTTP. Being straightforward, the attack signature is well known, and as it is unencrypted, is easily detected by the IDS. The results of the attack can be observed in Figs. 2 and 3.

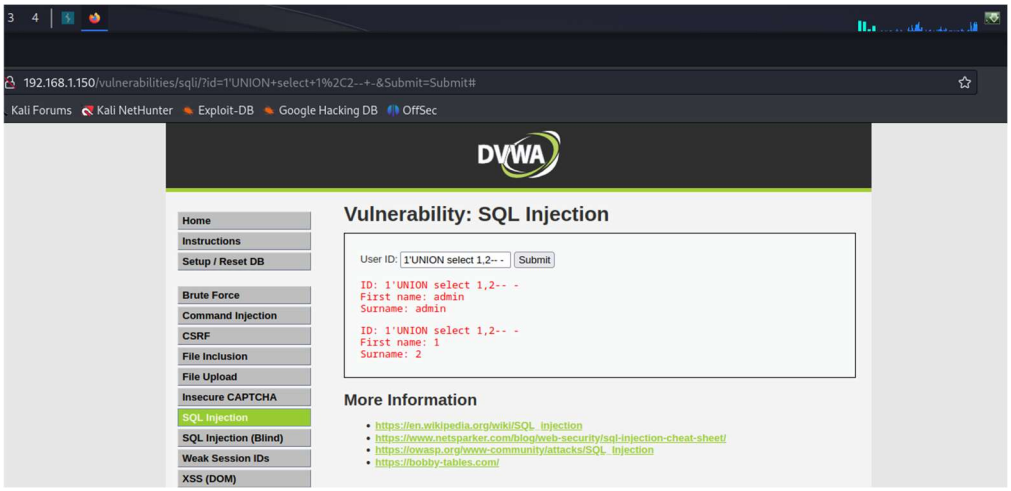


**Figure 2:** Running the SQL injection over HTTP.


**Figure 3:** Suricata detects the attack.

The second experiment is running the SQL injection over HTTPS. To perform this, a self-signed SSL certificate will be generated. The Apache will be configured to use a domain name "dvwa.local". The according line (192.168.1.150 dvwa.local) will be added to the "/etc/hosts" on Kali Linux. This will solve the problem with IP hostnames over the NAT. The virtual host configuration file can be seen in Fig. 4.

When running the attack, Suricata can detect the attack no more, as the payload is encrypted with SSL. This can be seen in Figs 5 and 6. Timestamps are included.

```
root@victim-VirtualBox:/etc/apache2/sites-available# cat dvwa.conf
<VirtualHost *:80>
        ServerName dvwa.local
        Redirect / https://dvwa.local/
</VirtualHost>

<VirtualHost *:443>
   ServerName dvwa.local
   DocumentRoot /var/www/dvwa/

   SSLEngine on
   SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
   SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
</VirtualHost>
root@victim-VirtualBox:/etc/apache2/sites-available#
```

**Figure 4:** DVWA virtual host.

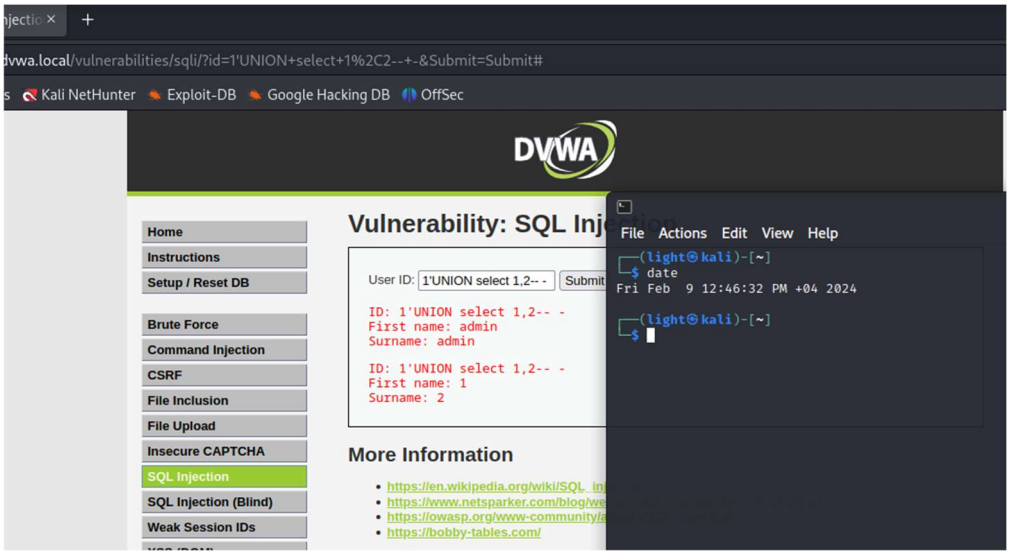


**Figure 5:** Running the SQL injection over HTTPS.

```
root@victim-VirtualBox:/var/log/suricata# date
2024 წლის 09 თებერვალი, 12:47:26 +04
root@victim-VirtualBox:/var/log/suricata# tail -f fast.log

^C
root@victim-VirtualBox:/var/log/suricata# date
2024 წლის 09 თებერვალი, 12:47:55 +04
root@victim-VirtualBox:/var/log/suricata#
```

**Figure 6:** Suricata fails to detect the attack performed over HTTPS.

The third experiment involves running the reverse shell over the unencrypted socket. To perform this, "command injection" tab in DVWA will be used. The payload running the reverse shell will be

bash -c "bash -i >& /dev/tcp/192.168.1.100/443 0>&1"

where 192.168.1.100 is an address of the Kali Linux and 443 is a port on which the attacker will "catch" the shell. Even if the payload is run over HTTPS, the new unencrypted connection opens between a victim and the attacker.

The Suricata rule to detect the malicious traffic is:

**alert tcp any any -> any any (msg:"WHOAMI issued"; flow:not_established,to_server; content:"whoami"; nocase; sid:4000006; rev:1;)**

The results of the attack and the detection are depicted in Figs. 7 and 8. Timestamps are included.

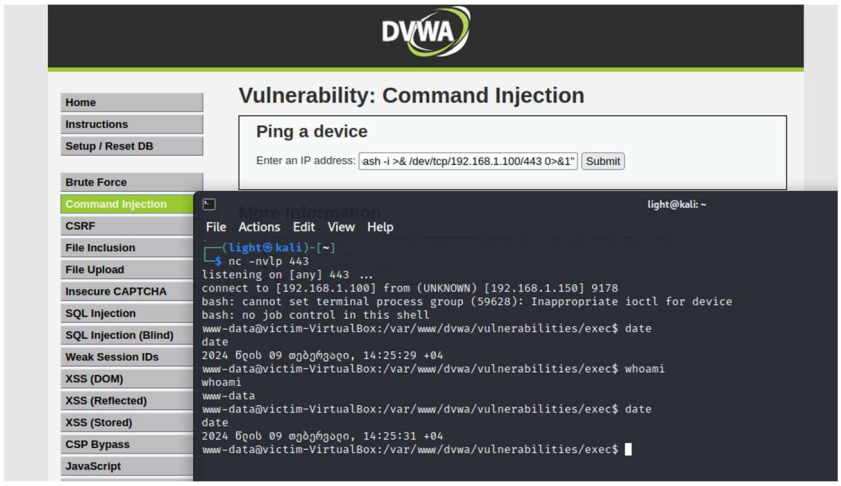


**Figure 7:** Received reverse shell and issued "whoami" command.

```
victim@victim-VirtualBox: /var/...   root@victim-VirtualBox: /var/lo...   root@victim-VirtualBox: /etc/su...   victim@victim-VirtualBox: /etc/...
root@victim-VirtualBox:/var/log/suricata# tail -f fast.log

02/09/2024-14:25:30.679271  [**] [1:4000006:1] WHOAMI issued [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.1.100:443 ->
172.16.0.13:40394
```

**Figure 8:** Suricata detects the attack.

The fourth experiment involves running the reverse shell over the encrypted socket. To perform this, "socat" tool will be used. The encryption is achieved using a self-signed certificate generated by "openssl". Results of the attack can be observed in Figs. 9, 10, and 11.

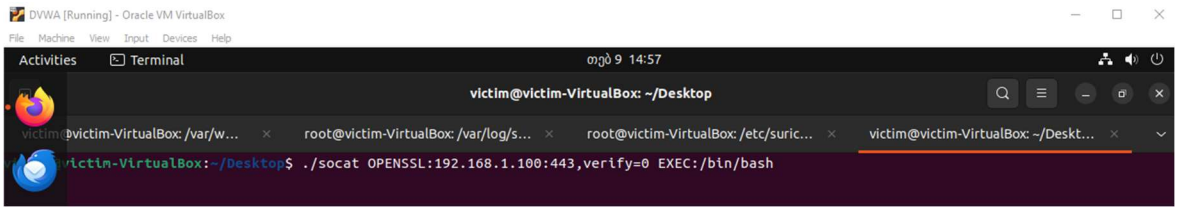


**Figure 9:** Running the encrypted reverse shell.

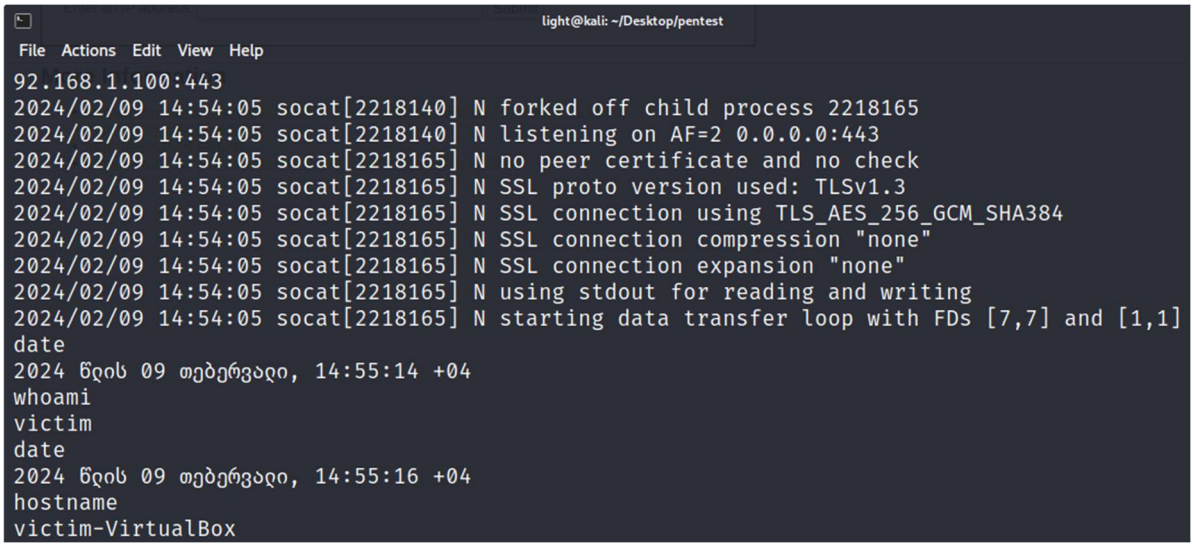


**Figure 10**: Receiving the reverse shell and running the "whoami" command.

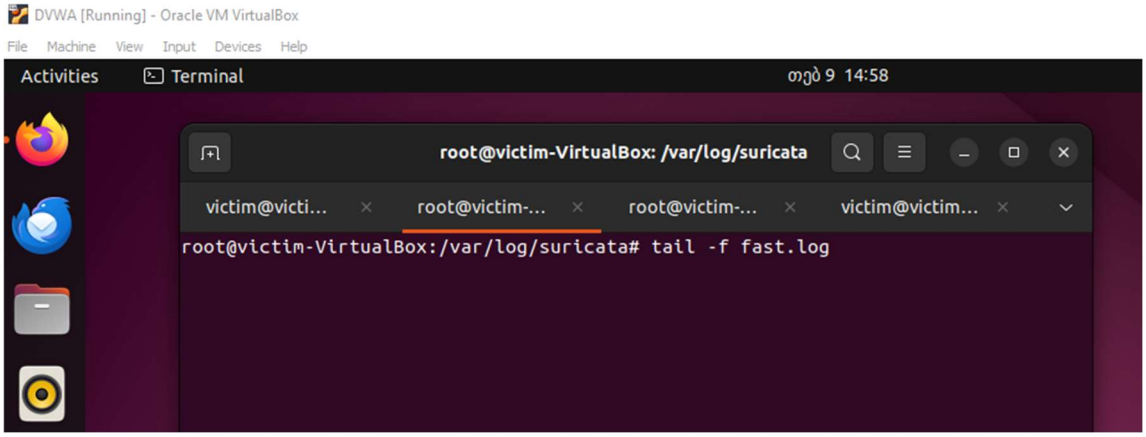


**Figure 11:** Suricata fails to detect the encrypted malicious traffic.

## 6. Results discussions

The research presented in this paper explores the effectiveness of cryptography as a tool for executing attacks and evaluates the efficacy of IDS in defending against such attacks when they occur over encrypted channels. The study demonstrates that encrypted traffic can effectively evade detection by widely used IDS software, such as "Suricata." While these IDS systems may be capable of detecting the presence of payloads, they face significant challenges in analyzing the attack timeline and reconstructing the sequence of actions performed by the attacker.

The paper highlights that even if an IDS identifies the payload, the complexity of analyzing the attack increases considerably when dealing with encrypted traffic. The challenge is further exacerbated by techniques like nested encryption. For instance, the research describes a scenario where a blue team successfully obtained the certificate used for encrypting the data. However, upon decrypting the malicious traffic, they discovered that the data had been further scrambled using an unknown algorithm, complicating the process of understanding and mitigating the attack.

The findings underscore the potential threats posed by any encrypted traffic, which can be exceedingly

difficult to detect and analyze using traditional security tools. This reveals a critical gap in current cybersecurity measures and highlights the need for more advanced techniques to address the evolving challenges posed by encrypted attack vectors.

## 7. Conclusions

In conclusion, the research presented in this whitepaper underscores the dual role of cryptography as both a protective measure and a potential attack tool, highlighting significant limitations in the efficacy of traditional IDS when faced with encrypted traffic. The study provides substantial evidence that well-established IDS solutions, such as Suricata, can be bypassed by encrypted communications. Furthermore, even if such encrypted payloads are detected, the process of reconstructing the attack timeline and understanding the sequence of actions undertaken by the attacker remains profoundly challenging, particularly when nested encryption techniques are employed.

The findings of this research emphasize the critical need to acknowledge encrypted traffic as a sophisticated threat that traditional security tools may inadequately address. To mitigate these challenges, the whitepaper offers specific recommendations for both blue and red team practitioners:

For Blue Team Members:

- Import SSL Certificates: Integrate SSL certificates used by web servers into IDS solutions to enhance visibility and detection capabilities.
- Enable Advanced Detection Techniques: Implement anomaly-based and Indicator of Compromise-based detection methods to improve the identification of malicious activities.
- Deploy Endpoint Detection and Response (EDR): Utilize EDR solutions on both servers and client systems to strengthen endpoint protection and response mechanisms.

For Red Team Members:

- Leverage Encrypted Channels: Use encrypted channels to execute payloads and attacks, making it more challenging for IDS systems to detect and analyze the traffic.
- Apply Nested Encryption: Employ nested encryption strategies to further complicate detection efforts and hinder the blue team's ability to reconstruct attack sequences.

By applying these recommendations, organizations can greatly enhance their preparedness and responsiveness to the evolving threats posed by encrypted traffic, therefore improving their overall cybersecurity level in an increasingly complex threat landscape.

## Acknowledgments

This work was supported by the Shota Rustaveli National Foundation of Georgia (SRNSFG) (NFR-22-14060).

## References

[1] R. Marusenko, V. Sokolov, P. Skladannyi, Social Engineering Penetration Testing in Higher Education Institutions, Advances in Computer Science for Engineering and Education VI, vol. 181 (2023) 1132–1147.

[2] R. Marusenko, V. Sokolov, V. Buriachok, Experimental Evaluation of Phishing Attack on High School Students, Advances in Computer Science for Engineering and Education III, vol. 1247 (2020) 668–680. doi:10.1007/978-3030-55506-1_59.

[3] R. Marusenko, V. Sokolov, I. Bogachuk, Method of Obtaining Data from Open Scientific Sources and Social Engineering Attack Simulation, Advances in Artificial Systems for Logistics Engineering, vol. 135 (2022) 583–594. doi: 10.1007/978-3-031-04809-8_53.

[4] R. Chernenko, et al., Encryption Method for Systems with Limited Computing Resources, in: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, CPITS, vol. 3288 (2022) 142–148.

[5] I. Ristic, Bulletproof SSL and TLS: Understanding and deploying SSL/TLS and PKI to secure servers and web applications. Feisty Duck (2014).

[6] M. Iavich, et al., Comparison and Hybrid Implementation of Blowfish, Twofish and RSA Cryptosystems, in: IEEE 2nd Ukraine Conf. on Electrical and Computer Engineering (UKRCON) (2019) 970–974, doi: 10.1109/UKRCON.2019.8880005.

[7] R. Oppliger, SSL and TLS: Theory and Practice. Artech House (2023).

[8] A. S. Ashoor, S. Gore, Importance of Intrusion Detection System (IDS), International Journal of Scientific and Engineering Research, vol. 2, no. 1 (2011) 1–4.

[9] O. Depren, et al., An Intelligent Intrusion Detection System (IDS) for Anomaly and Misuse Detection in Computer Networks, Expert systems with Applications, vol. 29, no. 4 (2005) 713–722.

[10] H. J. Liao, et al., Intrusion Detection System: A Comprehensive Review, Journal of Network and Computer Applications, vol. 36, no. 1 (2013) 16–24.

[11] D. Day, B. Burns, A Performance Analysis of Snort and Suricata Network Intrusion Detection and Prevention Engines, in: 5th International Conference on Digital Society, Gosier, Guadeloupe (2011) 187–192.

[12] J. Donadio, G. Guerard, S. B. Amor, Collection of the Main Anti-Virus Detection and Bypass Techniques, in: Network and System Security: 15th International Conference (NSS) (2021) 222–237.

[13] E. Albin, N. C. Rowe, A Realistic Experimental Comparison of the Suricata and Snort Intrusion-Detection Systems, in: 26th International Conference on Advanced Information Networking and Applications Workshops (2012) 122–127.
[14] K. Wong, et al., Enhancing Suricata Intrusion Detection System for Cyber Security in SCADA Networks, in: IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE) (2017) 1–5.
[15] S. Liu, R. Kuhn, Data Loss Prevention. IT Professional, vol. 12, no. 2 (2010) 10-13.
[16] S. Alneyadi, E. Sithirasenan, V. Muthukkumarasamy, A Survey on Data Leakage Prevention Systems, Journal of Network and Computer Applications, vol. 62 (2016) 137–152.
[17] J. Singh, Challenge of Malware Analysis: Malware Obfuscation Techniques, International Journal of Information Security Science, vol. 7, no. 3 (2018) 100–110.
[18] I. You, K. Yim, Malware Obfuscation Techniques: A Brief Survey, in: International Conference on Broadband, Wireless Computing, Communication and Applications (2010) 297–300.
[19] B. B. Rad, M. Masrom, S. Ibrahim, Camouflage in Malware: From Encryption to Metamorphism, International Journal of Computer Science and Network Security, vol. 12, no. 8 (2012) 74–83.
[20] D. Maiorca, et al., Stealth Attacks: An Extended Insight into the Obfuscation Effects on Android Malware, Computers & Security, vol. 51 (2015) 16–31.
[21] D. Park, H. Khan, B. Yener, Generation & Evaluation of Adversarial Examples for Malware Obfuscation, in: 18th IEEE International Conference on Machine Learning and Applications (ICMLA) (2019) 1283–1290.
[22] M. Christodorescu, S. Jha, Testing Malware Detectors. ACM SIGSOFT Software Engineering Notes, vol. 29, no. 4 (2004) 34–44.
[23] M. I. Sharif, et al., Impeding Malware Analysis Using Conditional Code Obfuscation, in: NDSS (2008).
[24] G. Canfora, et al., Obfuscation Techniques against Signature-based Detection: A Case Study, in: Mobile Systems Technologies Workshop (MST) (2015) 21–26.
[25] E. Jintcharadze, M. Iavich, Hybrid Implementation of Twofish, AES, ElGamal and RSA Cryptosystems, in: IEEE East-West Design & Test Symposium (EWDTS) (2020). doi: 10.1109/ewdts50664.
[26] M. Daffa, Suricata Rules (2023). URL: https://github.com/daffainfo/suricata-rules