DOI:10.22616/RRD.29.2023.039

IMPROVING INTRUSION DETECTION INTELLIGENCE BY OPEN DATA USAGE

***Nauris Paulins**
Latvia University of Life Sciences and Technologies, Latvia
*Corresponding author's e-mail: nauris.paulins@lbtu.lv

Abstract
Cyberattacks have become a regular part of network activity. To mitigate the risks from possible threats, organisations have implemented firewalls and intrusion detection systems, which can help stop network attacks. The problem is that often the accuracy of these systems is not effective enough. Another part of network security is security information and management platforms. These systems are more advanced versions of Threat Intelligence Platforms, because it is possible to make in-depth analyses of real-time events in a network. This research paper proposes improving intrusion detection system functionality using Open-Source Intelligence. Anomaly-based intrusion detection systems often generate alerts, but these alerts require deeper analysis to understand whether it is a real attack or just a false alarm. By making Open-Source Intelligence requests and evaluating extra information, it is possible to make more precise rules to stop attacks against network infrastructure. Open-Source Intelligence requests are generated directly from the intrusion detection system or with Python scripts based on the organisation's infrastructure profile. The proposed architecture was experimentally tested by automating Open-Source Intelligence requests and intrusion detection rule generation by Python scripts.
Key words: Threat intelligence, intrusion detection, cybersecurity.

Introduction

Nowadays, information system faces a wide range of threats and cyberattacks. With COVID-19 and the Russian invasion of Ukraine, hackers and other malicious actors have become even more active than before. Latvian Information Technology Security Incident Response Institution has stated in its report that malicious activity has grown multiple times (CERT.LV, 2022).

At the same time, network bandwidth has grown exponentially and devices are becoming more and more portable. It brings great opportunities to business, but it also makes threat actors even more dangerous. IBM Threat Intelligence Index (IBM, 2017) shows that hacktivism and destructive malware have grown dramatically – a 100% increase in hijacking attempts, a 62% increase in phishing attacks using spear phishing attachments, a 21% increase in backdoors deployed, etc.

Simply analysing packet per packet or file by file is not enough and cybersecurity professionals must use more sophisticated tactics and implement more and more intelligence in their tools. The problem nowadays is that information collection simply for knowledge base also is not effective, because then information can become noise. It is important to collect information with context and when it is needed.

The most effective instrument for attack mitigation is intrusion detection systems (IDS) which can identify attack behaviour. These systems generally are classified into two groups: Misuse-based detection systems and Anomaly-based detection systems. Misuse-based detection is based on patterns or strings that correspond to a known attack or threat, but Anomaly-based detection monitors deviation from the normal behaviour of network connections or other activities in systems (Liao *et al.*, 2013). These systems have been developed already several years and a lot of research has been done to improve their effectiveness and detection algorithms as well (Lata & Singh, 2022; Nuaimi *et al.*, 2023). All this research concentrates on the implementation of algorithms and detection accuracy. The situation with Misuse-based systems is quite simple. If packets match certain rules or defined patterns, then an alarm is announced. In this case, research focuses on system performance and search algorithms which could check rules as fast as possible. Another situation is with Anomaly-based systems. These systems do not search for an exact match but for a deviation from normal. That is why alarms often require for deeper research. Alarms like Hight Total Traffic, Suspect Data Flow or Hight Concern Index can be a hacker attack activity, but at the same time, it can be a normal user who does something different from everyday activities. The number of alarms also depends on threshold boundaries in the system. There would be a possibility to rise system intelligence from different knowledge sources which are publicly available on the internet, like MITRE Cybersecurity vulnerabilities catalogue (MITRE.org, 2023) or Google Hacking Database (OffSec, 2023). But in current research, IDS systems are not focusing on this potential but more concentrating on machine learning and algorithm improvement.

That does not mean that cybersecurity does not use the possibilities of intelligence analysis. In fact, there are separate movements of cybercrime intelligence called Threat intelligence platforms (TIP) which concentrate on proactive measures of computer and network security and incident response (Cascavilla *et al.*, 2021). These systems are based on intelligence traditional steps – planning, collection, processing, analysis, dissemination and evaluation. Such a process is often used in Security Operation Centers (SOC) where

analysts try to analyse incidents by collecting various information from the internet. Basically, it works with Source Intelligence (OSINT), for improving the analytical quality of TIP data and trying to find indicators of compromise (IOC), to show possible indicators that the system can be compromised (Costa-Gazcon, 2021). There are possible benefits that can bring data enrichment via OSINT sources, like blacklisted IP recognition, exploit activity detection, attack planning recognition via Natural language processing, create detection rules based on open data.

In both cases there is the possibility of false positive information – in IDS it can be false alerts about attacks and in TIP gathered OSINT information can be nothing more than disinformation. That is why the important part is not just an extraction of information, but also the evaluation of data source trust and relevance with events in real infrastructure.

This paper tries to evaluate possibilities to improve IDS detection quality by the usage of TIP intelligence possibilities and OSINT data integration for attack detection. The research paper describes various data sources for information security improvement and how information from these sources can be extracted and integrated into intrusion detection systems in automated way. There is other research done for evaluation of OSINT in intrusion detection (Shannon, Pournouri, & Ibbotson, 2021), but previous research more concentrates on algorithmic improvement of systems, but not so much on system architecture and contextual extraction of OSINT feeds.

By proposing intellectual IDS architecture, the author tries to provide possibilities to improve data source quality assessment, and data exchange between systems and shows deployment possibilities of system over real attack scenarios.

Materials and Methods

Testing lab environment intelligent intrusion detection inspired by the HELK platform (Rodriguez, 2020) this is an open source threat hunting platform which works more like a Security information and event management (SIEM) tool. But in this case, to collect information from various OSINT sources, SpiderFoot project was installed, which can be combined with the so-called ELK stack (Elastic, 2022) consisting of Elasticsearch, Logstash and Kibana. Elasticsearch is a database engine for data collection, Logstach is for log integration in Elasticsearch, and Kibana is for data visualization.

Each OSINT source needs to be parsed to harmonize, categorize incoming data and prepare data so that it can be defined as IOC of some specific event. OSINT sources are categorized by the following categories:

- abused IP – IP addresses which are blacklisted by security organizations and reported as malicious;
- early warnings – OSINT feeds about possible attacks. It can be text feeds from hacker forums or social networks about possible attacks or zero-day vulnerabilities;
- threat patterns – specific properties or features of some exploit or attack pattern, it can be used to describe this threat in IDS rules;
- attack patterns – these can be information from vulnerability and exploit databases about known misconfigurations, and possible requests against the network;
- vulnerability warnings – based on existing infrastructure, it can be alarms about Common vulnerabilities of relevant application versions, and possible exploits of certain software versions;
- abusive content – it can be some content which should not be used in an organization's network, like spam, disinformation feeds, pornography, etc.

As shown in Figure 1, during the generation of IOC, it is necessary to avoid duplicates, because threats can be recognized already before or multiple sources can generate the same IOC. If duplicates are not found, they can be stored in a database.

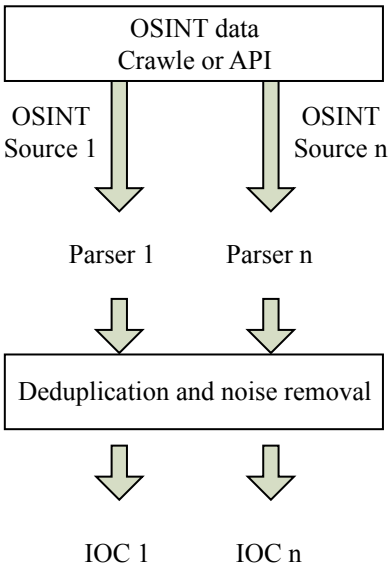

Figure 1. OSINT data collection process.

An open-source intrusion detection system Suricata IDS (OISF, 2020) was chosen for this research. Suricata outperforms other open-source IDS like Suricata and Zeek due to better exploitation of underlaying hardware. Suricata IDS architecture can be seen in Figure 2, most important part is detection engine because it provides possibility to split detection process in multiple threads, so it can bring advantage in multicore environments.

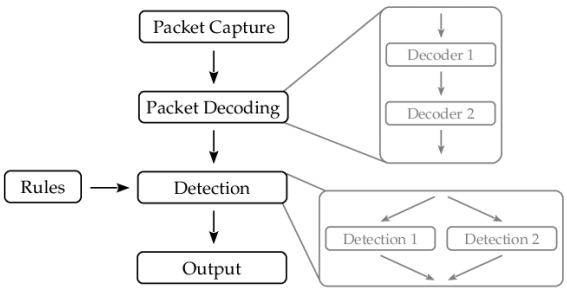

Figure 2. Suricata IDS architecture (Prenosil & Hammoudeh, 2017).

Suricata IDS provide possibility to make scripts for complex matching and even gain efficiency by combining multiple rules into one script (Waleed, Jamali & Masood, 2022). This IDS has its own rules dataset for possible attack detection for application-layer requests and better hardware acceleration can be very useful if proposed architecture will be implemented in production environment where network speed is much faster.

The base idea behind cooperation between OSINT and IDS is that Suricata alerts about suspicious activity can be triggers for OSINT requests to collect extra information about IP addresses, activity topics, suspicious files and other relevant information which has been noticed in network flow. It is necessary to convert IOC data to the Suricata rule, so that it can be used for intrusion detection. But simply converting all OSINT indicators would not be effective, it would generate too many rules and also would require too much time to check these rules that is why it would be necessary to understand whether these generated IOCs correlate with infrastructure attributes or data flows. There are several possibilities to check the similarity between values in IOCs (Leskovec *et al.*, 2014), but in this case, Locality-Sensitive Hashing will be used, which can compress large values into small signatures and preserve the expected similarity between two pairs. The Apache Spark is used to calculate event similarity and related event grouping.

By such computation it can be calculated if vulnerability X can be used in the infrastructure of company Y. This can help analysts to avoid analysis of unrelated events, for example, IDS should not check against vulnerabilities of MikroTik Routers if there are no such communication devices in the organization's network. There could be some sources in which IOC could be used straight without checking its relevance, for example, if we check the Abused IP database and some IP has been reported as malicious IP with 100% confidence, then it must be blocked without specific analytics behind it. But then such IP must be blocked in the firewall, not in the IDS system.

An important part of OSINT data relevance checking and relevance evaluation is the company profile which collects the main features of an organization and its infrastructure and indicators of possible threat vectors. The company profile consists of multiple parts:

- domain indicators – industry of business – financial or healthcare, private or governmental agency, clients of organizations, partners, or other high-level description indicators;
- infrastructure – server OS, switches, firewalls, IDS, proxy servers, VPN gateways and other nodes for information storage, accessibility, and intercommunication;
- applications – web servers, SMTP services, Mail servers, web applications, FTP services and other systems which are used in daily work and processing organization data;
- endpoints – these could be workstations, laptops, printers, mobile phones, some IoT devices, or VoIP devices;
- cloud services – cloud storage services, software as service applications, and online applications which are used by organization staff.

There were included multiple well know OSINT sources parts, some of which are shown in Table 1, but information about the rest of the sources can be found in IntelMQ documentation.

Table 1

OSINT Source examples

OSINT source	URL
Exploit Database	https://www.exploit-db.com/
Abused IP abuse.ch	https://abuse.ch/
AlienVault OTX Reputation list	https://otx.alienvault.com/
Malware Panels Tracker	http://benkow.cc/export.php
Darkreading.com	https://www.darkreading.com/
Cybersecurity vulnerabilities	https://www.cve.org/
Threat sharing platform	https://www.misp-project.org/
BitcoinAbuse	https://www.bitcoinabuse.com/
AdGuard DNS	https://adguard.com/en/welcome.html
HackerTarget	https://hackertarget.com/

Based on the organization profile and its assets, possible attack vectors can be generated. Asser features can help evaluate IOC relevance and its possibility to generate an IDS rule for infrastructure protection. Attack vectors may depend on infrastructure, but they can be – phishing, recognizance, malware, ransomware, compromised credentials, misconfiguration, and potential vulnerabilities.

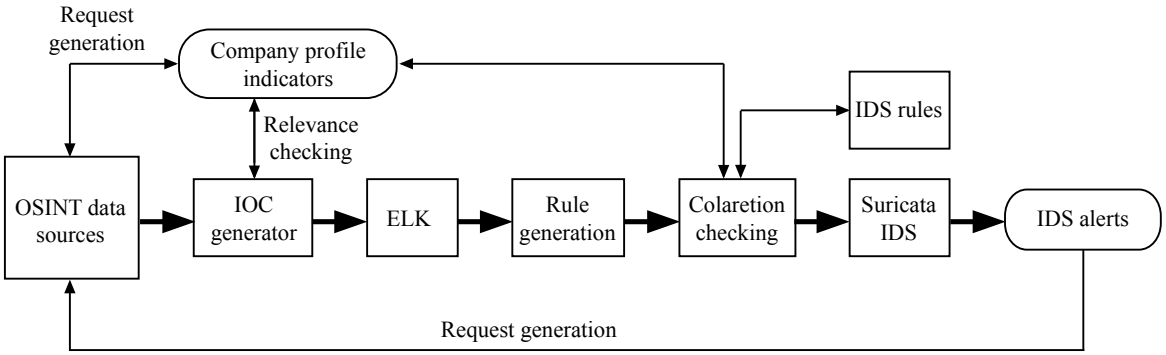

Figure 3. Intelligent IDS with OSINT capabilities.

Potential threat vector enumeration is an essential part of intelligence teams' work to provide better attack mitigation steps and recommendations.

The architecture of the system can be seen in Figure 3. IntelMQ threat intelligence platform has been installed on Kali Linux (OffSec, 2021) which is an open-source distribution for Penetration testing. The system was made on a VirtualBox machine with 8 GB RAM and 2 processor cores and 40 GB of HDD. For OSINT feed classification, it was decided to use Security Incident Taxonomy which was created by ENISA and TF-CSIRT and approved on 26 September 2018 (TF-CSIRT, 2018).

Open source database Reidis Output bot (IntelMQ community, 2023) was used to feed IntelMQ messages in the ELK. MinHash LSH also supports Redis (Zhu, 2021) which goes together with chosen similarity algorithm for this research. The similarity between events can be computed by Python:

```
from pyspark.ml.feature import MinHashLSH
from pyspark.mi.linalg import Vectors
from pyspark.sql.function import col

mh=MinHashLSH(inputCol='features', outpuCol='hashes', numHashTables=10)
model = mh.fit(result)

r=model.aproxSimilarityJoin (result, result, 0.3, distCol='JaccardDistance').cache()
```

The threshold of similarity can be set from 0 to 1, where 0 would be an exact match, but 1 would be different values. After finding similarities between events, it is possible to make a grouping of these events and put them in joint tables having records about similarity which can be quickly extracted during event checking.

For testing reason, .pcap files from the real network environment have been used. Those files have been provided by Latvian governmental institutions with an agreement that the organization name and network sensitive information cannot be exposed. Network flow files have been taken from Cisco Network Analytics software and are stored because of their suspicious activity, so there are a lot of suspicious IPs with Hight concern index (Cisco, 2017).

Results and Discussion

Table 2 shows a summary of experimental results. During a 5-day period of network flow .pcap files 1048 IP addresses were approved as abused and malicious. Also, 23 domains were recognized as malicious.

In the IDS system, these addresses performed activities like port scanning, recognizance, and ping scanning which in normal situations would not be a reason for address blocking. In this case, abused IPs were written in a separate .csv file which was configured as a blacklisted source for the organization's Palo Alto firewall. It is hard to say that communications were related to certain countries, because traffic was communing not just from countries like China, Russia, and Bulgaria, but also from European countries like Germany and Netherlands.

Table 2

OSINT feed results in IDS system

OSINT activity	Count
Blacklisted IP addresses	1048
Blacklisted domain names	23
Whitelisted IP addresses	128
Vulnerabilities alerts	36
Attack alerts	0

In this case, also 128 addresses were related to services or organizations which approve that they do not provide a threat to the organization. Mostly they were cloud services of several updates or services which extracted data from organizations' API services. Due to a large amount of data or large count of requests, these addresses were marked as suspicious, but their activity objective was not malicious.

It was also possible to extract 36 OSINT feeds about potential vulnerabilities which are announced in open databases and connected with organization-used services. It was discovered that some OSINT sources already provide API interfaces for Suricata rule generation like AlienVault OTX Suricata rule generator. In the case of certain exploit announcements and their connection with organization recognition, a specific rule was generated to be sure that IDS will recognize malicious content if it reaches the network. Python script was used to convert OSINT feeds from JSON to Suricata rules.

Trying to detect DoS attacks or hacking announcement feeds, the research failed to get positive match for certain organisation.

This research is still ongoing to improve the rule generation process and OSINT request targeting. Result numbers are quite small because research has not been done on real networks, but with selected .pcap flows. Threshold should be set for which category events OSINT requests must be generated, otherwise it would generate too much traffic and could cause more noise than benefit in an attack with threat actors.

Conclusions

This research demonstrates the possibility to improve IDS system rules with OSINT data. OSINT requests are based on contextual information of organization infrastructure and IDS alerts. Such an approach can be very useful for Anomaly-based IDS which analyses unusual activities in the network and could be useful to extract extra information about communication parties. Especially it can be used for suspicious IP address recognition and blocking by putting these addresses in a firewall blocking list. Experimental results show that OSINT data can be used for extra rule generation in IDS systems and help to prevent suspicious threat actors before analysts make the decision whether alarm is suspicious or not.

Acknowledgements

Academic study was financed by the project 'Support for doctoral studies in LUA', 2009/0180/1DP/1.1.2.1.2/09/IPIA/VIAA/017/ agreement No. 04.4-08/EF2.D2.28.

References

IBM. (2017). IBM X-Force Threat Intelligence Index 2017. Retrieved October 22, 2022, from https://securityintelligence.com/ibm-x-force-threat-intelligence-index-2017/.

Cascavilla, G., Tamburri, D.A., & Van Den Heuvel, W.J. (2021). Cybercrime threat intelligence: A systematic multi-vocal literature review. Computers and Security, 105, 102258. DOI: 10.1016/j.cose.2021.102258.

CERT.LV. (2022). Publiskais pārskats par CERT.LV uzdevumu izpildi 2022. (CERT.LV Public Performance Report 2022). Retrieved March 12, 2023, from https://cert.lv/uploads/parskati/cert-ceturksna-C4-atskaite-2022-LV.pdf. (in Latvian).

Cisco. (2017). Alarm Category : High Concern and High Target Index. 1–15. Retrieved June 12, 2022, from https://cisco.bravais.com/s/orHXCs9vFa5QxuoyPto9.

Costa-Gazcon, V. (2021). Practical Threat Intelligence and Data-Driven Threat Hunting: A hands-on guide to threat hunting with the ATT&CK™ Framework and open source tools. Packt Publishing, 398.

Elastic. (2022). Elastic Stack. Retrieved April 24, 2022, from https://www.elastic.co/what-is/elk-stack.

IntelMQ community. (2023). Configuring IntelMQ for Logstash. Retrieved March 18, 2022, from https://intelmq.readthedocs.io/en/develop/user/ELK-Stack.html.

Lata, S., & Singh, D. (2022). Intrusion detection system in cloud environment: Literature survey & future research directions. *International Journal of Information Management Data Insights, 2*(2), 100134. DOI: 10.1016/j.jjimei.2022.100134.

Leskovec, J., Rajaraman, A., & Ullman, J.D. (2014). Finding Similar Items. Mining of Massive Datasets, 68–122. DOI: 10.1017/cbo9781139924801.004.

Liao, H.J., Richard Lin, C.H., Lin, Y.C., & Tung, K.Y. (2013). Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications, 36(1), 16–24. DOI: 10.1016/j.jnca.2012.09.004.

The MITRE Corporation. (2023). About the CVE Program. Retrieved March 12, 2022, from https://www.cve.org/About/Overview.

OffSec. (2022). Kali Linux. Retrieved February 12, 2021, from https://www.kali.org/.

OISF. (2020). Suricata. Retrieved June 12, 2020, from https://suricata.io/.

Prenosil, V., & Hammoudeh, M. (2017). A Survey on Network Security Monitoring Systems. 28th Modern Artificial Intelligence and Cognitive Science Conference, MAICS 2017, February 2017, 189–190. DOI: 10.1145/1235.

Rodriguez, R. (2020). HELK – Hunting ELK. Retrieved March 3, 2022, from https://thehelk.com/intro.html. OffSec. (2023). Google Hacking Database. Retrieved August 28, 2022, from https://www.exploit-db.com/google-hacking-database.

Nuaimi, M., Fourati, L., & Hamed, B. (2023). Intelligent Approaches toward Intrusion Detection Systems for Industrial Internet of Things: A Systematic Comprehensive Review. *Journal of Network and Computer Applications*, 090(1), 1–10. DOI: 10.1016/j.jnca.2023.103637.

TF-CSIRT. (2018). References Security Incident Taxonomy Task Force. Retrieved April 12, 2020, from https://github.com/enisaeu/Reference-Security-Incident-Taxonomy-Task-Force/.

Waleed, A., Jamali, A.F., & Masood, A. (2022). Which open-source IDS? Snort, Suricata or Zeek. Computer Networks, 213, 1389–1286. DOI: 10.1016/j.comnet.2022.109116.

Wass, S., Pournouri, S., & Ibbotson, G. (2021). Prediction of Cyber Attacks during Coronavirus Pandemic by Classification Techniques and Open Source Intelligence. Advanced Sciences and Technologies for Security Applications. DOI: 10.1007/978-3-030-68534-8_6.

Zhu, E. (2021). Datasketch: Big Data Looks Small. Retrieved October 12, 2022, from https://ekzhu.com/datasketch/.