MATRIX: A Comprehensive Graph-Based Framework for Malware Analysis and Threat Research

Marco Simoni[1,3] and Andrea Saracino[2]
[1]*Sapienza Università di Roma, Rome, Italy*
[2]*Department of Excellence in AI and Robotics (DiPE), TeCIP Institute, Scuola Superiore Universitaria Sant'Anna, Pisa, Italy*
[3] *Istituto di Informatica e Telematica, Consiglio Nazionale delle Ricerche, Pisa, Italy*

Keywords: Malware Analysis, Cyber Threat Intelligence, Knowledge Graph, Structured Threat Information Expression.

Abstract: This paper presents *MATRIX (Malware Analysis and Threat Research with STIX)*, a graph database for the comprehensive analysis and research of malware and threats. To provide a unified view of the threat landscape, *MATRIX* integrates data from major cybersecurity frameworks, including *MITRE ATT&CK*, *DEF3ND*, *CAPEC*, *Malware Behavior Catalog* (MBC), *Metasploit*, *Common Vulnerabilities and Exposures* (CVE) and *Common Weakness Enumeration* (CWE). Developed in Neo4j using the *Structured Threat Information Expression* (STIX™) standard, *MATRIX* includes more than 22,910 nodes and combines 14 *STIX Domain Objects (SDOs)* and 6 *STIX Relationship Objects (SROs)* to provide a detailed analysis of malware behavior, detection rules and defense strategies, making it a valuable tool for cybersecurity research. The system also integrates real-world malware reports and is automatically updated with data from sources such as *VirusTotal*, *MalwareBazaar* and *VirusShare*, supporting continuous and up-to-date threat analysis. We demonstrate its versatility through case studies comparing malware objectives and analyzing the impact of detection and mitigation.

1 INTRODUCTION

Cybersecurity research demands efficient methods to represent and analyze diverse data. Graph databases are increasingly adopted for their ability to model complex relationships (Reading, 2021), integrating alerts and logs from multiple tools (Neo4j, 2021) and revealing hidden patterns (Sheikhalishahi et al., 2022). Knowledge graphs, a form of graph database, map real-world entities and relationships, supporting CTI (Sikos, 2023) (Bolton et al., 2023) and enabling real-time retrieval, especially in Retrieval Augmented Generation (RAG) systems (Lewis et al., 2020).

However, cybersecurity research still lacks unified models that combine disparate data and provide real-time updates, essential for dealing with evolving threats. Effective analysis of malware and threats requires an understanding of both the individual components and their interactions. For this reason, systems that are able to logically and semantically combine different cybersecurity elements into a cohesive structure are essential to improve threat and malware analysis. Graphs can help researchers achieve this goal by enabling the connection of different components, such as vulnerabilities, exploits, malware and attack patterns, into a single, interconnected model. It is also important to constantly update this system to reflect the ever-changing threat landscape.

Contribution. This paper presents *MATRIX (Malware Analysis and Threat Research with STIX)*, a graph-based framework specifically designed for the comprehensive analysis of malware and threats. It integrates and links data from *MITRE ATT&CK* (Corporation, 2025b), *DEF3ND* (Corporation, 2025d), *CAPEC* (Corporation, 2025a), *Malware Behavior Catalog* (MBC), *Metasploit Framework* (Project, 2025a) (Rapid7, 2025), *Common Vulnerabilities and Exposures* (CVE), and *Common Weakness Enumeration* (CWE) to provide a comprehensive overview of the threat landscape. All data within MATRIX has been obtained through an extensive crawling process of the aforementioned sources. The structured information is collected and organized using the *Structured Threat Information Expression* (STIX™) (OASIS, 2020) standard, leveraging datasets from `mitre/cti` (Corporation, 2025c) and `MBCProject` (Project, 2025b) to ensure a detailed and consistent representation of malware and threats. The latest data crawling operation was conducted in January 2025, ensuring that MATRIX maintains an up-to-date and reliable knowledge base for cybersecurity analysis. MATRIX contains 14 differ-

Simoni, M. and Saracino, A.
MATRIX: A Comprehensive Graph-Based Framework for Malware Analysis and Threat Research.
DOI: 10.5220/0013629300003979
In *Proceedings of the 22nd International Conference on Security and Cryptography (SECRYPT 2025)*, pages 495-502
ISBN: 978-989-758-760-3; ISSN: 2184-7711

ent node types, called *STIX Domain Objects (SDOs)*: **Malware**, **Malware Behavior**, **Malware Objective**, **Malware Method**, **Indicator**, **Course of Action**, **Data Component**, **Data Sources**, **Tool**, **Intrusion Set**, **Campaign**, **Weaknesses**, **Vulnerabilities** and **Exploit**. There are 6 different edge types, called *STIX Relationship Objects (SROs)*: *related-to*, *mitigates*, *uses*, *indicates*, *detects*, *exploits*. By linking different cybersecurity elements, MATRIX helps to better understand the behavior of different malware and improve detection, analysis and defense against complex threats, making it a valuable tool for cybersecurity research and intelligence. The main contributions of *MATRIX*:

- We introduce a Neo4j-based graph that integrates `mitre/cti`, `MBCProject`, 14 SDOs, 6 SROs and rules from CAPA (Mandiant, 2025a) and SIGMA (SigmaHQ, 2025). Malware reports from VirusTotal (VirusTotal, 2025) are linked via ElasticSearch (Elastic, 2025). All data and containers are publicly available.[1] [2]
- The graph includes over 22,910 nodes (excluding vulnerabilities and exploits), making it 5x larger than `mitre/cti` and 25x larger than `MBCProject`, and contains more than 10,000 real malware hashes.
- *MATRIX* is continuously updated with data from VirusTotal, MalwareBazaar (abuse.ch, 2025), and VirusShare (VirusShare.com, 2025), ensuring ongoing relevance and completeness.
- The graph enables detailed analyses of malware behavior, objectives, and defensive impact, with case studies such as rule-based comparison of objectives, impact evaluation of mitigations, behavior linking across malware families, and tactic-specific API analysis.

Paper Organisation. The paper is organized as follows: Section 2 covers background on graph databases, CTI, and STIX. Section 3 describes the MATRIX architecture. Section 4 showcases example analyses. Section 5 reviews related work in cybersecurity knowledge graphs. Section 6 concludes with future directions.

2 BACKGROUND

Graph Databases and Knowledge Graphs. A *graph database* is a NoSQL model optimized for managing complex, often directed, relationships via graph structures. A *knowledge graph* extends this by defining a labeled, directed graph $G = (V, E, L)$, where entities V are linked by labeled edges $E \subseteq V \times V \times L$, representing typed relationships.

Cyber Threat Intelligence (CTI). CTI provides actionable insights on threats, enabling organizations to enhance defenses. It operates at tactical (immediate threats), operational (actors/campaigns), and strategic (long-term planning) levels. Effective CTI must be complete, accurate, relevant, and timely.

Structured Threat Information Expression (STIX). STIX is a standardized format for sharing machine-readable CTI. STIX 2.1 models threat data as a graph, using *STIX Domain Objects (SDOs)* as nodes and *STIX Relationship Objects (SROs)* as edges. It supports objects like *Malware*, *Indicator*, and *Threat Actor*, connected via predefined or custom relationships (e.g., `indicates`).

3 MATRIX ARCHITECTURE

The MATRIX architecture, shown in Fig. 1, was built using the Neo4j framework to organize and connect the key elements of malware and threat analysis. Most of the components are based on the MITRE ATT&CK framework, but to provide a more complete view of the threat landscape, we have also integrated data from the *Malware Behavior Catalog* (MBC), which focuses specifically on malware objectives and behaviors. The `mitre/cti` and `MBCProject` are two of the most important STIX standards and collections used in cybersecurity. In MATRIX, all nodes and relationships are based on the STIX objects from the `mitre/cti` dataset, with the exception of Malware Behavior, Objective and Method (in blue in Fig. 1), which follow the format of the `MBCProject`. This ensures that our graph conforms to the MBC STIX standard and provides a more detailed and consistent approach to analyzing malware. The nodes *Weaknesses, Vulnerabilities, and Exploit* are not included in any of the two standard collections; the node *Weaknesses* is a new SDO that we have specifically defined. In addition, the graph is kept up to date through continuous and automatic updates and becomes more comprehensive over time so that it always reflects the latest threat data. Below is a breakdown of the *SDOs* of the graph.

Malware includes definitions from MITRE ATT&CK and MBC, providing details such as aliases, descriptions, and external references. **Malware Behavior** captures the actions of malware using techniques from MITRE ATT&CK, MBC (with prefixes

[1] https://github.com/MATRIX-Malware-Analysis/MATRIX/

[2] https://hub.docker.com/r/matrixmalware/matrix

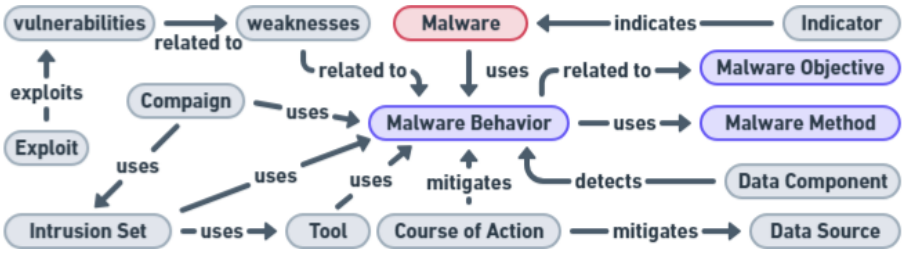


Figure 1: MATRIX Architecture Overview.

T, **B**, **E**, **C**, **F**), and CAPEC. These behaviors are associated with *CAPA* and *Sigma* rules through the `detection_rules` field and are modeled as *Attack Pattern* objects in STIX. **Malware Objective** represents the high-level intent behind malware behaviors, derived from ATT&CK tactics and expanded with objectives from MBC. **Malware Method** refers to how behaviors are executed, often represented as sub-techniques or specific implementations. These methods are always associated with behaviors and cannot exist independently. **Indicators** include over 10,000 malware hashes and *YARA* rules from 269 families, automatically collected from sources like *MalwareBazaar* and *VirusShare*. Corresponding reports from *VirusTotal* are stored in an Elasticsearch database for analysis, with regular updates ensuring the dataset remains current. **Course of Action** nodes represent defensive strategies derived from MITRE ATT&CK Mitigations and the *DEFEND* framework, providing guidance on how to prevent or reduce the impact of threats. **Intrusion Sets** describe groups of threat actors—referred to as *Groups* in ATT&CK—that operate over time to conduct campaigns or coordinated attacks. **Campaigns** are coordinated sets of malicious activities carried out by an intrusion set, usually targeting specific sectors or organizations. **Data Sources** represent broader categories of information such as logs or telemetry that are relevant to identifying ATT&CK techniques. **Data Components**, on the other hand, are the specific elements or system events—like API calls or process creations—that allow the detection of malicious behaviors. Finally, **Tools** are legitimate software applications that can be leveraged by attackers. Analyzing their usage helps in profiling threat actor tactics and understanding how campaigns are executed. **Weaknesses** refer to software or hardware flaws identified in the CWE catalog, which may expose systems to potential risks. **Vulnerabilities** correspond to publicly disclosed issues listed in the CVE database, each describing a specific flaw that can be exploited to compromise a system. **Exploits** are modules from the Metasploit Framework designed to target known CVEs, used to simulate or conduct real-world attacks.

Table 1: MATRIX Nodes and Relationships Summary.

Node	Size	Number of Nodes
Campaign	120K	28
Course of Action	25M	6029
Data Component	452K	109
Data Source	156K	38
Exploit	38M	4531
Indicator	54M	13407
Intrusion Set	860K	163
Malware	3.4M	829
Malware Behavior	20M	1705
Malware Method	2.0M	482
Malware Objective	144K	35
Tool	352K	85
Vulnerabilities	968M	231315
Weaknesses	5.1M	964

Relationship Type	Size	Count
MATRIX Relationships	351M	87,642

Table 2: Dataset Comparison.

Dataset	Nodes	Relat.	Malware	Indic.
`mitre/cti`	4237	22259	734	-
`MBCProject`	892	1015	50	183

Table 1 outlines the MATRIX dataset structure, including over 230K **Vulnerabilities** (968MB), 13K **Indicators** (54MB), and 6K **Courses of Action** (25MB), along with **Malware**, **Tools**, and **Weaknesses**. Behavioral data is captured through nodes like **Malware Behavior** (1.7K nodes, 20MB) and **Malware Method**. The graph comprises 87,642 relationships (351MB). As shown in Table 2, MATRIX is 541% larger than *mitre/cti* and 2568% larger than *MBCProject*, with over 22,910 nodes (excluding vulnerabilities and exploits), and includes over 13,000 real-world indicators and malware reports.

4 APPLICATION OF MATRIX TO THREAT AND MALWARE ANALYSIS

We present 7 examples of analytical insights that can be derived from the graph, along with the time re-

quired to compute them. The analyzes that can be performed are not limited to those we have shown. For example, while we often perform studies on Malware Objectives, similar analyzes can also be performed on Malware Behaviors that are more complex to visualize. More examples can be found here[3]. The first five examples were determined using the graph based on information from MITRE, which would otherwise be difficult to recognise without putting this information into a graph. However, the last two examples depend on the number of real hashes collected; the more malware samples collected, the more accurate the analysis will be.

Comparison of Malware Objectives Based on API and String Correlation. Fig. 2 compares the objectives of 60 Ransomware families, 38 Spyware families, and 112 Trojan families based on APIs and strings extracted from detection rules in *Malware Behavior* nodes. The graphs were created by linking the behaviors of each malware to its objectives and measuring the correlation between APIs, strings and objectives. The distribution shows how often APIs or strings are correlated with an objective, while the entropy (Morato et al., 2018) indicates the variety of APIs or strings used to reach each objective. Fig. 2a shows that all malware types correlate strongly with the objectives *Discovery* and *Defense Evasion* based on APIs. Ransomware and Trojan have a low correlation with *Impact* and *Persistence*, while spyware has none. Trojan are the only ones that show a low correlation with *Credential Access*. Fig. 2b shows higher API entropy for *Discovery* and *Defense Evasion* for ransomware and Trojan, while spyware uses more predictable APIs. Fig. 2c shows that all malware types correlate strongly with *Credential Access* based on strings, but Trojan show no correlation with *Impact* and *Persistence*. Finally, Fig. 2d shows that ransomware and Trojan use more distinct strings for *Discovery* and *Defense Evasion*, while spyware is more predictable across all objectives. The time required to obtain these results is approximately 0.06s.

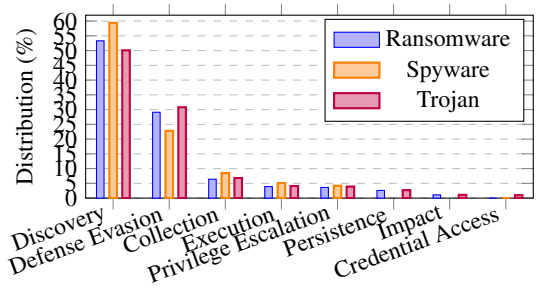
Analyzing Data Component Impact on Malware Categories. Figure 3 shows the impact of various common *Data Components* on 60 Ransomware families, 38 Spyware families, 112 Trojan families and 15 Worm families. The impact is calculated from the frequency with which each Data Component contributes to the detection of a malware type in relation to the total data components that affect this malware. *Process Creation* and *Command Execution* are very influential for all malware types, especially spyware, suggesting that system-level behaviors are important detection indicators. Spyware also shows a greater reliance on *OS API Execution*, *Script Execution* and *File Access*, reflecting the use of commands and file operations for malicious purposes. Trojan are also characterized by their dependency on *OS API Execution*, but also on *Network Traffic Flow* and *Connection Metadata*, so network monitoring is crucial for their detection. Both the worm and the ransomware affect *Service Metadata* and *Windows Registry Key Modification*, probably to persist in the system and maintain control by changing critical settings. On the other hand, other data components such as *Process Modification* and *File Creation* have minimal impact on all malware types, making these behaviors less important for detecting these specific threats. The time required to obtain these results is approximately 0.3 ms.

Prioritizing Mitigation Techniques Based on Malware Type. Figure 4 shows the impact of different mitigation strategies (Course of Action) on 60 Ransomware families, 38 Spyware families, 112 Trojan families and 15 Worm families. The impact is based on how often each strategy effectively defends against a malware type in relation to the total *Courses of Action* that affect that malware. The graph shows that spyware relies heavily on *Data Loss Prevention* and *User Training*, underlining the importance of educating users and preventing data exfiltration. Trojan particularly benefit from *Execution Prevention* and *Endpoint Behavior Prevention on Endpoint*, highlighting the need to block unauthorized execution and monitor malicious behavior. Ransomware is primarily influenced by *User Account Management* and *Restrict File and Directory Permissions*, which emphasizes the importance of managing access rights. Worms, on the other hand, are strongly influenced by *User Account Management* and *Restrict File and Directory Permissions*, showing that restrictions and user permission management are important strategies. Less effective strategies such as *Limit Access to Resources Over Network* and *Account Use Policies* play a lesser role in containing all malware types. The time required to obtain these results is approximately 0.04s.

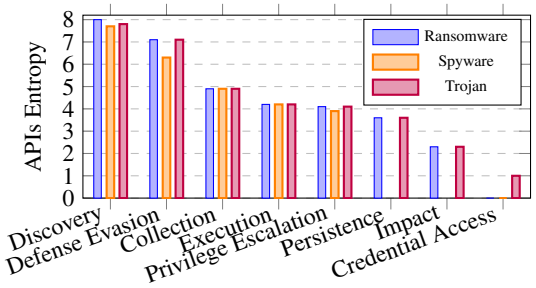
How Key Techniques Connect and Support Diverse Malware Types. Measuring *Betweenness Centrality* (Pontecorvi and Ramachandran, 2015) allows us to identify which *Techniques* play a crucial role in connecting different malware categories. The objective is to pinpoint key actions that multiple malware families depend on for propagation, persistence, or execution. By targeting techniques with high betweenness centrality, defense strategies can effectively disrupt multiple malware types simultaneously.

As illustrated in Figure 6, *T1082 (System Infor-*

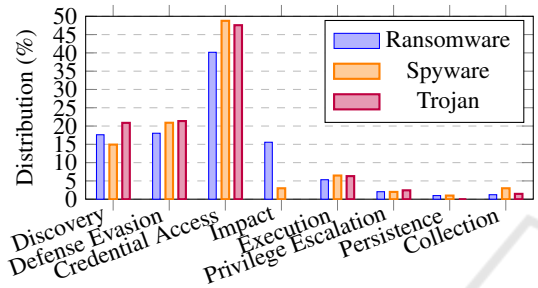
[3]https://github.com/MATRIX-Malware-Analysis/MATRIX/tree/main/EXAMPLES



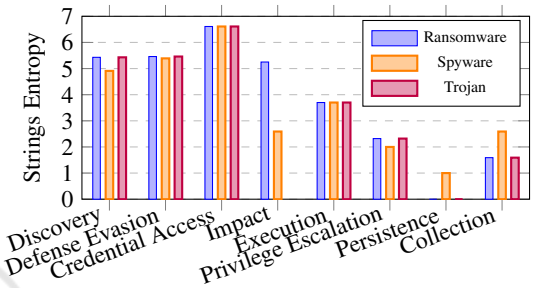
(a) Percentage distribution of Objectives in Ransomware, Spyware, and Trojan based on unique APIs



(b) Entropy of Objectives in Ransomware, Spyware, and Trojan based on unique APIs based on unique Strings



(c) Percentage distribution of Objectives in Ransomware, Spyware, and Trojan



(d) Entropy of Objectives in Ransomware, Spyware, and Trojan based on unique Strings

Figure 2: Comparison of percentage distribution and entropy of objectives across Ransomware, Spyware, and Trojan based on unique APIs, (2a) and (2b), and Strings, (2c) and (2d).

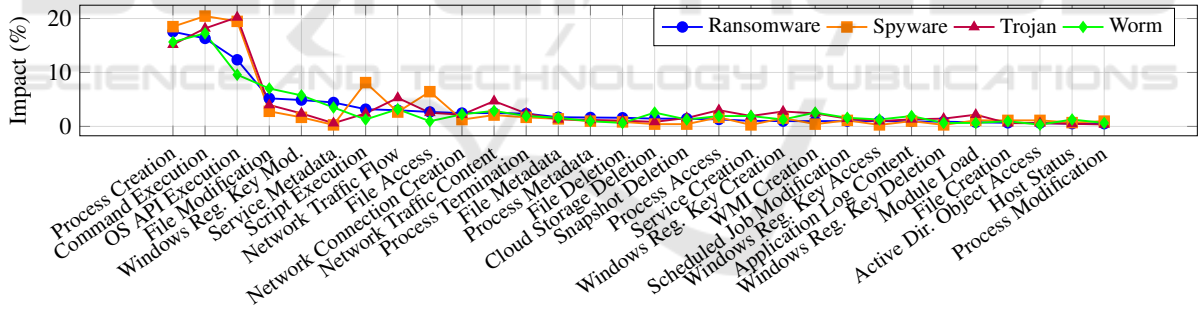


Figure 3: Impact of various common *Data Components* on Ransomware, Spyware, Trojan and worms.

mation Discovery) exhibits the highest centrality for Ransomware-RAT and RAT-Spyware connections, a finding consistent with reports from (Security, 2025) (Mandiant, 2025b). Similarly, *T1105 (Ingress Tool Transfer)* is critical for Backdoor-Ransomware and Backdoor-Worm relationships, as confirmed by (Canary, 2025) (for Threat-Informed Defense, 2025) and (MITRE, 2025). Additionally, *T1140 (Deobfuscation/Decoding)* serves as a central technique linking Backdoor-Spyware and RAT-Spyware, corroborated by findings in (Mandiant, 2025b). *T1210 (Command and Scripting Interpreter)*, cited as first techniques in the top-10 by (MITRE, 2025) exhibits highest centrality for Ransomware-RAT and Backdoor-Ransomware.

Overall, techniques characterized by high interdependence, such as deobfuscation and system discovery, act as essential links between different malware categories. This makes them prime targets for disrupting malware operations and strengthening cybersecurity defenses. The computational time required to obtain these results is 5.9 seconds.

Evaluating the Importance of Malware Techniques Using PageRank Analysis. Figure 5 shows the PageRank (Gleich, 2015) values for the techniques used by 60 Ransomware families, 38 Spyware families, 112 Trojan families, 15 Worm families, 15 RAT families and 208 Backdoor families.

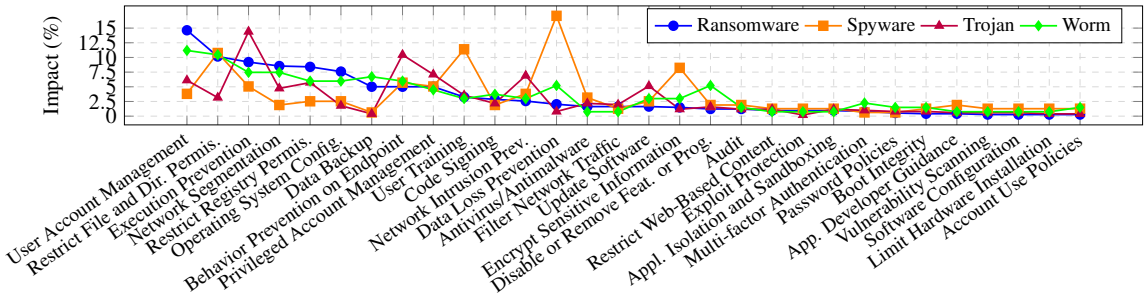


Figure 4: Impact of different mitigation strategies *(Course of Action)* on Ransomware, Spyware, Trojan and Worm.

A higher PageRank indicates that a particular technique plays a greater role in the malware's operations or is used more frequently. Worms have consistently high PageRank values for several key techniques, such as *File and Directory Discovery*, as confirmed by (MITRE, 2025) and *Native API*, which are essential for their distribution and operation. This means that the worms rely on a few key actions, making these techniques prime targets for disrupting their spread. Backdoor relies on techniques such as *Ingress Tool Transfer*, *Web Protocols* (also very important for spyware) and *Windows Command Shell*, as confirmed by (Canary, 2025), to gain unauthorized access and assert itself in a system. In particular, ransomware relies on techniques such as *Inhibit System Recovery*, also this confirmed by (MITRE, 2025) and *Native API* to disable restore options and manipulate system-level functions, making it more difficult for users to restore their system. In contrast, RAT and spyware have lower PageRank scores, suggesting that they use a wider range of techniques without relying heavily on a single action. Even though these types of malware spread their operations over several techniques, focusing on a wide range of defense strategies can still help mitigate their impact. The time required to obtain these results is approximately 0.15s.

Behavioral and Technical Similarities Between Two Emotet Samples. Table 3 summarises the common API calls, registry keys, loaded modules and MITRE ATT&CK techniques observed in two Emotet malware samples: 2fd433c3ff68507ddbf0ec3e90a6320b35b44c8089504403c457bc9819190a0a and 214946b987ad69fa46f1d27ab35026b856a4fcd2abd46b0b5ba86dc71be58d89. The data was extracted from CAPE sandbox reports with real malware samples from VirusShare. The Jaccard similarity (Fender et al., 2017) score of *0.97* indicates a very high similarity between the two malware samples based on their common characteristics. The listed API calls, such as `VirtualProtect`, `GetCPInfo` and `CloseHandle`, show that both malware samples are involved in similar activities, including process control, memory management and system information retrieval. These are typical actions used by Emotet to achieve persistence and execute its malicious operations. MITRE techniques used by both malware samples include *Credential Dumping*, *Virtualization/Sandbox Evasion* and *Impair Defenses*, suggesting that they focus heavily on credential evasion and theft. This reflects the typical behaviour of Emotet, which is known for its ability to bypass security measures and collect sensitive information from infected systems. he loaded modules, including `BCRYPT.DLL` and `WININET.DLL`, also confirmed by (Shaddy43, 2025), indicate that both examples use the same Windows libraries for cryptographic functions, Internet communication and shell operations. The time required to obtain these results is 0.2 s.

5 RELATED WORK

Knowledge graphs (KGs) have become crucial in cybersecurity for representing and analyzing complex, multi-source data. Sikos (Sikos, 2023) emphasizes their role in enhancing cyber situational awareness and supporting machine learning. Li et al. (Li et al., 2024) focus on KG construction and quality evaluation to improve cybersecurity analysis, while Li et al. (Li et al., 2023a) and (Li et al., 2023b) propose methods integrating KGs and pre-trained models for cyber threat intelligence extraction and automation. Bolton et al. (Bolton et al., 2023) explore ATT&CK-based threat mapping, and Wang et al. (Wang et al., 2021) demonstrate how graph databases capture attack behaviors to improve 6G network security. Ren et al. (Ren et al., 2022) present CSKG4APT, combining KGs and deep learning for APT tracking and proactive defense. Liu et al. (Liu et al., 2020) design an ontology for network security based on STIX, enhancing attack representation and CTI sharing. Chen et al. (Chen et al., 2024) improve IoC management on OpenCTI, achieving a 25.18% increase in confidence

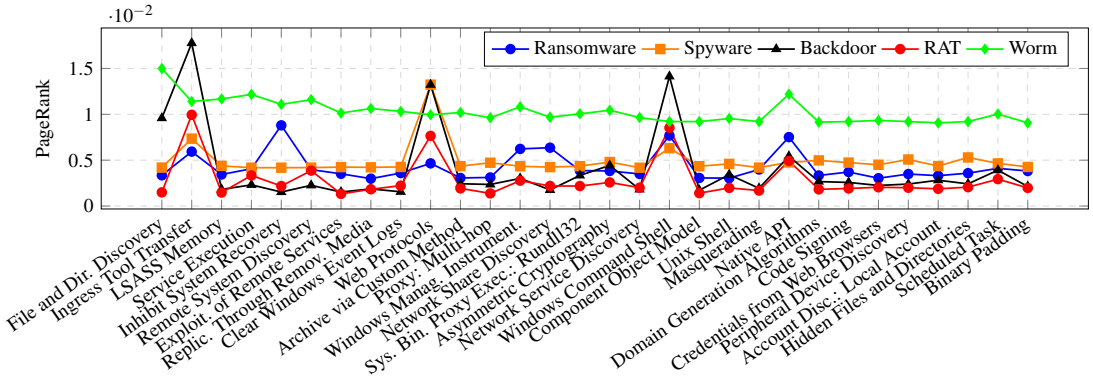


Figure 5: PageRank values for Techniques used by Ransomware, Spyware, Backdoor, RAT and Worm.

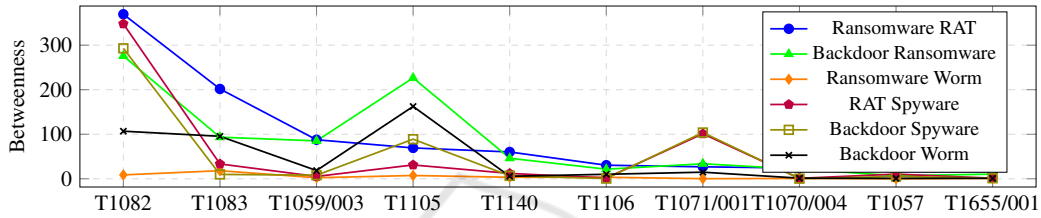


Figure 6: Betweenness Centrality of Behaviors across Malware Categories.

Table 3: Summary of Common Calls, MITRE Techniques between two Emotet hashes.

Category	Values
Calls Highlighted	VirtualProtect, CryptStringToBinaryA, GetCurrentHwProfileA, CloseHandle, TlsGetValue, EnterCriticalSection, GetLastError, srand, IsValidCodePage, GlobalLock, VirtualAlloc, CreateToolhelp32Snapshot, GetCurrentThreadId, CoCreateInstance, GetCPInfo, HeapAlloc, LeaveCriticalSection, InterlockedDecrement, GetProcessHeap, GetVersionExA, Process32Next, RegOpenKeyExA, GetModuleHandleW, ...
MITRE Techniques	T1503, T1497.001, T1003, T1552.001, T1005, T1562, T1081, T1071, T1032, T1555, T1106, T1497, T1562.001, T1071.001, T1012, T1552, T1082, T1089, T1057, T1555.003.
Modules Loaded	BCRYPT.DLL, NTMARTA.DLL, WINHTTP.DLL, SHELL32.DLL, GDIPLUS.DLL, CRYPT32.DLL, NTDLL.DLL, SECHOST.DLL, WININET.DLL, WS2_32.DLL, CRYPTBASE.DLL, CFGMGR32.DLL, OLE32.DLL, ADVAPI32.DLL, GDI32.DLL, RPCRT4.DLL, SHLWAPI.DLL, RSTRTMGR.DLL, USER32.DLL, MSVCR100.DLL, NSI.DLL, KERNEL32.DLL ...

scoring. Bhalekar et al. (Bhalekar and Saini, 2024) and Habaybeh and Marshall (Habaybeh and Marshall,) highlight the use of graph databases for cybersecurity data analysis and legal assessments. Unlike these approaches, MATRIX aggregates malware, threat, and vulnerability data from multiple sources into a unified and extensible framework, improving research and advanced analysis capabilities.

6 CONCLUSION AND FUTURE WORKS

We presented *MATRIX*, a unified graph-based framework for malware and threat analysis. Built on STIX 2.1 and integrating data from seven cybersecurity frameworks (`MITRE ATT&CK`, `MBCProject`, `CAPEC`, `DEF3ND`, etc.), MATRIX provides a semantically consistent view of the threat landscape. MATRIX is over 5x larger than `mitre/cti` and 25x larger than `MBCProject`, and includes 10,000+ real malware samples from `VirusTotal`, `MalwareBazaar`, and `VirusShare`, linked to detection rules, behaviors, and objectives for in-depth analysis. We showcased MATRIX's capabilities via case studies on malware behavior correlations, mitigation impacts, and technique influence across families. The system is designed for continuous updates and future expansion. Upcoming work will focus on integrating MATRIX into RAG systems to support real-time analysis.

ACKNOWLEDGMENTS

This work was partly supported by the HORIZON Europe Framework Programme by the MUR PRIN-2022-PNRR ASSISTANTS (P2022WEAH7) project funded under the EU RESTART program.

REFERENCES

abuse.ch (2025). Malwarebazaar - malware samples and feeds.

Bhalekar, P. M. and Saini, J. R. (2024). Comprehensive exploration of the role of graph databases like neo4j in cyber security. In *2024 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pages 1–4. IEEE.

Bolton, J., Elluri, L., and Joshi, K. P. (2023). An overview of cybersecurity knowledge graphs mapped to the mitre att&ck framework domains. In *2023 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 01–06. IEEE.

Canary, R. (2025). Threat detection report - top att&ck techniques.

Chen, S., Hwang, R., Ali, A., Lin, Y., Wei, Y., and Pai, T. (2024). Improving quality of indicators of compromise using STIX graphs. *Comput. Secur.*, 144:103972.

Corporation, M. (2025a). Common Attack Pattern Enumeration and Classification (CAPEC).

Corporation, M. (2025b). MITRE ATT&CK.

Corporation, M. (2025c). Mitre cti github repository.

Corporation, M. (2025d). MITRE DEF3ND.

Elastic (2025). Elasticsearch - distributed, restful search and analytics engine.

Fender, A., Emad, N., Petiton, S. G., Eaton, J., and Naumov, M. (2017). Parallel jaccard and related graph clustering techniques. In Alexandrov, V., Geist, A., and Dongarra, J. J., editors, *Proceedings of the 8th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems, ScalA@SC 2017, Denver, CO, USA, November 13, 2017*, pages 4:1–4:8. ACM.

for Threat-Informed Defense, C. (2025). Top 15 techniques sightings ecosystem.

Gleich, D. F. (2015). Pagerank beyond the web. *SIAM Rev.*, 57(3):321–363.

Habaybeh, N. and Marshall, A. M. Towards a historic malware frequency database. *Available at SSRN 4392182*.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Li, H., Shi, Z., Pan, C., Zhao, D., and Sun, N. (2024). Cybersecurity knowledge graphs construction and quality assessment. *Complex & Intelligent Systems*, 10(1):1201–1217.

Li, J., Li, J., Xie, C., Liang, Y., Qu, K., Cheng, L., and Zhao, Z. (2023a). Pipckg-bs: A method to build cybersecurity knowledge graph for blockchain systems via the pipeline approach. *Journal of Circuits, Systems and Computers*, 32(16):2350274.

Li, Z.-X., Li, Y.-J., Liu, Y.-W., Liu, C., and Zhou, N.-X. (2023b). K-ctiaa: automatic analysis of cyber threat intelligence based on a knowledge graph. *Symmetry*, 15(2):337.

Liu, Z., Sun, Z., Chen, J., Zhou, Y., Yang, T., Yang, H., and Liu, J. (2020). Stix-based network security knowledge graph ontology modeling method. In *ICGDA 2020: 3rd International Conference on Geoinformatics and Data Analysis, Marseille, France, April 15-17, 2020*, pages 152–157. ACM.

Mandiant (2025a). Capa rules - mandiant.

Mandiant (2025b). M-trends report.

MITRE (2025). Top 10 lists.

Morato, D., Berrueta, E., Magaña, E., and Izal, M. (2018). Ransomware early detection by the analysis of file sharing traffic. *Journal of Network and Computer Applications*, 124:14–32.

Neo4j (2021). Graphs for cybersecurity: Defending against sophisticated attacks.

OASIS (2020). Stix version 2.1.

Pontecorvi, M. and Ramachandran, V. (2015). A faster algorithm for fully dynamic betweenness centrality. *CoRR*, abs/1506.05783.

Project, M. (2025a). Exploit mapping to maec.

Project, M. (2025b). Malware behavior catalog stix repository.

Rapid7 (2025). Metasploit framework.

Reading, D. (2021). Picking the right database tech for cybersecurity defense.

Ren, Y., Xiao, Y., Zhou, Y., Zhang, Z., and Tian, Z. (2022). Cskg4apt: A cybersecurity knowledge graph for advanced persistent threat organization attribution. *IEEE Transactions on Knowledge and Data Engineering*.

Security, P. (2025). The top ten mitre att&ck techniques.

Shaddy43 (2025). Emotet malware analysis.

Sheikhalishahi, M., Saracino, A., Martinelli, F., and Marra, A. L. (2022). Privacy preserving data sharing and analysis for edge-based architectures. *Int. J. Inf. Sec.*, 21(1):79–101.

SigmaHQ (2025). Sigma rules - generic signature format for siem systems.

Sikos, L. F. (2023). Cybersecurity knowledge graphs. *Knowledge and Information Systems*, 65(9):3511–3531.

VirusShare.com (2025). Virusshare - collection of malware samples.

VirusTotal (2025). Virustotal - free online virus, malware and url scanner.

Wang, W., Zhou, H., Li, K., Tu, Z., and Liu, F. (2021). Cyber-attack behavior knowledge graph based on capec and cwe towards 6g. In *International Symposium on Mobile Internet Security*, pages 352–364. Springer.