Strategic selection of data sources for cyber attack detection in enterprise networks: A survey and approach

Manuel Kern, Florian Skopik, Max Landauer
Austrian Institute of Technology
Vienna, Austria
firstname.lastname@ait.ac.at

Edgar Weippl
University of Vienna
Vienna, Austria
edgar.weippl@univie.ac.at

ABSTRACT

Cyber attacks leave traces in data sources, such as in log files, memory or data-streams. Detection systems utilize these data sources to detect the application of specific attack techniques. Attack techniques vary considerably in terms of their effectiveness, potential impact and application by threat actors. Data sources, on the other side, may contain traces of one or several attack techniques, and the effort to process their output may differ heavily. Therefore, it is obvious that not all data sources are of equal value for detection and organizations must carefully survey which sources shall be analyzed and what attack techniques need to be found. This paper introduces D3TECT, a process model that describes a procedure for dynamically ranking and selecting data sources suitable for detection. The novelty is that this model accounts for constraints in the selection process. For instance if a certain data source cannot be utilized in a specific setting, e.g., due to data privacy constraints, the discovery of the most important attack techniques are still ensured by the remaining data sources. Eventually, the D3TECT approach solves the challenge of strategically selecting data sources while accounting for their varying usefulness for attack detection. The model is tested with real data, utilizing the MITRE ATT&CK framework and numerous public cyber threat intelligence databases. The paper shows the ranking results and discusses their plausibility to validate D3TECT.

CCS CONCEPTS

• **Security and privacy** → **Intrusion detection systems**; **Systems security**; **Network security**;

KEYWORDS

cyber attacks, intrusion detection, data source selection, ranking attack techniques, survey

ACM Reference Format:
Manuel Kern, Florian Skopik, Max Landauer and Edgar Weippl. 2022. Strategic selection of data sources for cyber attack detection in enterprise networks: A survey and approach. In *The 37th ACM/SIGAPP Symposium on Applied Computing (SAC '22), April 25–29, 2022, Virtual Event, .* ACM, New York, NY, USA, Article 4, 10 pages. https://doi.org/10.1145/3477314.3507022

SAC '22, April 25–29, 2022, Virtual Event,

ACM ISBN 978-1-4503-8713-2/22/04.
https://doi.org/10.1145/3477314.3507022

1 INTRODUCTION

For several decades, a multi-layered approach has been pursued in security and detection systems have become a key element of modern information security architectures. Despite these advances, organizations still detect breaches at an average of 287 days as stated in a study [11] of 537 real breaches across 17 countries and regions. Economic damage of late detection is quantified[1] with 250% of the costs for detection after one week, in comparison to instant detection of threats in enterprise networks. There has always been an arms race between attackers and defenders, and despite great effort of research, there is still a lack of widespread application of effective detection systems.

Detection systems utilize data sources, such as log files, memory or data-streams to detect the application of specific attack techniques. Depending on size, exposure and complexity of an enterprise network, there are hundreds of data sources that can be analyzed to detect adversarial activities. Since resources are limited in real world, covering all data sources is not only costly, but also hardly feasible. Attack techniques vary considerably in terms of their effectiveness, potential impact and application by threat actors. Data sources, on the other side, may contain traces of one or even several attack techniques, and the effort to process their output may differ heavily. Therefore, it is obvious that not all data sources are of equal value for detection. Furthermore, if an organization rates data sources independently from each other in terms of their usefulness for detection, they neglect the fact that most of them detect multiple attack techniques and are therefore redundant to some degree, depending on the order in which sources are picked and integrated.

This paper addresses the question of how to strategically select data sources for the detection of cyber attacks in enterprise networks. The paper follows a risk-based approach to holistically mitigate risks to an acceptable level, taking economic and compliance aspects into account. In particular, the paper introduces the D3TECT process model that describes a procedure for dynamically ranking and selecting data sources suitable for detection, and which aims to achieve a high detectability of relevant attack techniques. In contrast to already existing approaches, D3TECT accounts for constraints in the selection process of data sources used for detection. For example, if a particular data source potentially reveals the application of a certain attack technique, but cannot be utilized due to technical, economic or compliance issues, D3TECT ensures that an organization is still able to detect this attack technique by utilizing appropriate alternative data sources and thus still achieves a high degree of detectability.

[1] https://www.kaspersky.com/blog/security_risks_report_financial_impact/

The remainder of this paper is organized as follows. Sect. 2 introduces D3TECT, a process model and a methodology, as well as strategic questions regarding detection characteristics of the model's key elements. The model, along with some metrics based on a survey of data sources, is implemented to utilize extracted data from the MITRE ATT&CK Framework in Sect. 3. The metrics form a basis for the selection process of data sources presented in Sect. 4 and are verified with the aid of various other public cyber threat intelligence data in Sect. 5. Ranking results are critically reviewed and their plausibility to validate D3TECT is discussed. Section 6 outlines related work and Sect. 7 concludes the paper.

2 D3TECT APPROACH

2.1 Overview

The design, implementation and deployment of detection systems are usually steered by detailed knowledge about attack techniques and their impact on actual systems. Not all attack techniques are relevant for all organizations as they heavily depend on the organization's assets, their vulnerabilities and their exposure. This is subject to an institution's risk assessment and evaluated in advance. IT risk management answers the following questions: What are the most important IT assets to protect? What are the business, compliance and security priorities? What threats are the organization facing? Which vulnerabilities are exploited within an asset? What are the impact and likelihood of an exploit related to such a vulnerability? Which risks have to be mitigated in accordance with the risk appetite? D3TECT (marked in blue) and its risk management integration is outlined in Fig. 1.

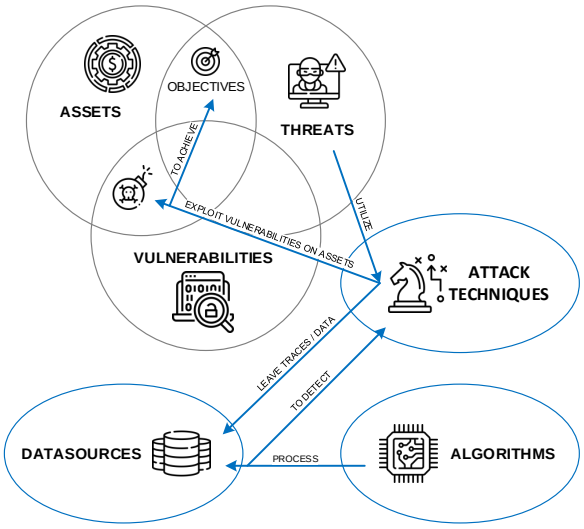

Figure 1: D3TECT risk management integration.

Detection is achieved with algorithms that process data sources to detect attacks realized by threats. Therefore detection algorithms are usually selected after attack techniques and data sources are fully known and understood. However, in practice not all attacks are known in advance, and it is valuable to consider the complete set of detection algorithms and data sources separately. D3TECT extends risk management and deals with the following questions:

Attack techniques:

(1) Which attacks affect the assets in scope?
(2) Which attack techniques are most often (successfully) used?
(3) Which attack techniques require low skill by the threat actor? For example techniques that are easily discovered and are exploitable with commercially available equipment.
(4) Which attack techniques are universally applicable? For example attacks that target multiple platforms and most infrastructures.

Data sources:

(1) Which data sources are easy wins, for example, enable detection of most attack techniques?
(2) More specific: Which data sources enable detection of the most often (successfully) used attack techniques?
(3) Which data sources detect only a few, but with other data sources not detectable, attack techniques.
(4) Which data sources enable detection early in the attack chain?
(5) Which data sources have to be analysed to detect all attacks in enterprise IT networks?
(6) In what order need data sources be implemented to detect the most common and most used attacks first?

Detection algorithms:

(1) Which detection algorithms are particularly well suited for a given data source or a specific attack technique?
(2) Are there detection algorithms that apply universally to all data sources?

Answers are elaborated utilizing the D3TECT methodology described in the following section.

2.2 D3TECT methodology

The D3TECT methodology shown in Fig. 2 is a structured risk-based approach to implement detection capabilities. Following the phases of D3TECT enables a prioritized step-by-step implementation of detection. Section 3 shows an example and describes the methodology.

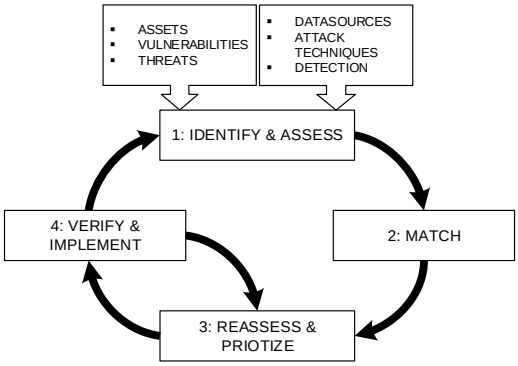

Figure 2: D3TECT methodology.

First, identification and assessment of the key elements is performed during the **IDENTIFY & ASSESS** phase. Depending on the organization's risk management maturity level, threats, assets and vulnerabilities are already identified and assessed. In an organization-specific context this data serves as a basis for D3TECT. One way to identify attack techniques is to analyze threats that target an institution. Identification of data sources is achieved by

analyzing attack techniques that exploit vulnerabilities on assets, or by utilizing advanced algorithms on data sources to detect zero day attacks. The methodology does not define a strict series of steps. To obtain a complete picture attack techniques, data sources and detection algorithms are identified without organizational context and independent of each other. This ensures an unbiased view, as otherwise attack techniques, data sources or detection algorithms might not be taken into consideration.

In the **MATCH** phase, D3TECT's key elements (Fig. 1 in blue) are matched. Their relations are shown in Fig. 1 and serve as a base for the matching process. In the most simple form a mapping between the entities that match is created. For example an attack leaves traces in a data source, or detection of an attack is achieved through algorithms that process one or many data sources. In a more complex implementation of the model, connections between entities are bound to a weight determined by the connection properties. These are for example the density of attack traces in a data source or accuracy of detection algorithms processing a specific data source.

The **REASSESS & EVALUATE** phase combines assessment results of phase one based on the correlations defined in phase two. For example, considering an organizational context, attack techniques used by threat actors targeting the institution and applicable to vulnerable assets are prioritized. Depending on the assessment results, detection for these attack techniques shall be implemented earlier, later or not at all.

Corresponding algorithms on data sources are implemented and verified in the **IMPLEMENT & VERIFY** phase, beginning with the most important pair. The results are fed back to phase three, where corresponding attack risks are recalculated. Detection properties, such as accuracy, have an impact on the assessment and attack risks are treated up to an acceptable level. This process is repeated until there are no more attacks above a threshold. In D3TECT elements have to be reevaluated on a regular basis by continuously cycling through the model. This guarantees continuous improvement.

3 MODEL IMPLEMENTATION

The first step of the methodology is to **IDENTIFY & ASSESS** the model components outlined in Fig. 1. The implementation is agnostic of any organizational context, but focuses on enterprise IT systems. Hence common threats and attack techniques applicable to a broad range of organizations, their assets and vulnerabilities have to be identified. Security researchers publish reports on threats, their techniques and strategies. This information is made public in an unstructured manner utilizing blog posts, news articles but also various documents. Commercial providers, but also state founded institutions, share knowledge with their customers via various information channels, including "Structured Threat Information Expression" (STIX) [2]. The implementation in Sect. 3 utilizes open source information, the process is described in the next section.

3.1 Survey of APT cases to identify relevant attack techniques

A collected structured view on tactics, techniques and procedures (TTPs) that are made public by security researchers is provided by the MITRE ATT&CK Framework [15]. The Framework maps kill chain phases already known from other models [1, 8] in a matrix of tactics and attack techniques. MITRE [15] defines the ATT&CK Framework as "globally-accessible knowledge base of adversary tactics and techniques based on real-world observations" open community project. The organization focuses on the most complete listing of attack techniques used by attackers in real world, broken down into the tactics of the ATT&CK matrix. It structures tactics into fine granular sections and, in contrast to Lockheed Martin Cyber Kill Chain [8], does not provide a model which maps attacks into phases building on each other, but remains attack-centric and technique focused.

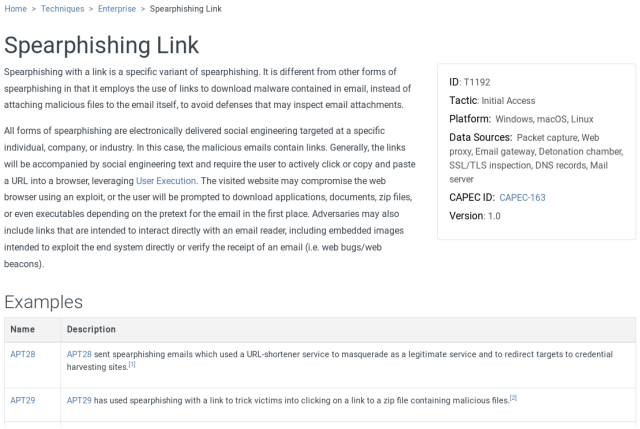

Figure 3: Excerpt of ATT&CK detail page T1192[2].

A detail page (Fig. 3) for each of the currently 185 main- and 367 sub-techniques ("ATT&CK Enterprise v9.0") lists information [15] about the attack, mitigation and detection. Most relevant for the models implementation are the following details:

- **Data sources** that enable detection. MITRE defines data sources as "Source of information collected by a sensor or logging system that may be used to collect information relevant to identifying the action being performed, sequence of actions, or the results of those actions by an adversary." [15].
- **Procedure examples** of software (malware) and threats implementing that technique.
- **Platforms** (Linux, Windows, ..) on which the techniques are applicable.
- **Tactics** in which the technique is used.

The ATT&CK Framework lists threats, attacks and data sources, but does not answer all questions raised in Sect. 2.1. In contrast to techniques, adversarial groups (threats) and software which go into great detail, data sources in the framework are only listed within the detail page of the attack technique and are not considered as entities. To date, there is no overview nor a complete picture of all data sources mentioned within the framework. Moreover, there is no structured information on the underlying detection algorithms provided. To close this gap, attack techniques and the connection to data sources are examined in detail in this paper.

The implementation of the model uses data from the MITRE ATT&CK Framework and is thus attack-centric. As basis for the

[2]https://attack.mitre.org/techniques/T1192/

Table 1: D3TECT Top 20 Techniques (E_W)

ID	Technique	Rank: E_W	Rank: E_{TOT}	Rank: E_{GRP}	Rank: E_{SW}
T1059	Command and Scripting Interpreter	1: 605.43	1: 339 (0, 0.0)	1: 83 (0, 0.0)	1: 256 (0, 0.0)
T1027	Obfuscated Files or Information	2: 478.86	3: 267 (1, -0.2)	2: 66 (0, 0.0)	5: 201 (3, -0.43)
T1105	Ingress Tool Transfer	3: 470.76	2: 291 (-1, 0.2)	7: 56 (4, -0.4)	2: 235 (-1, 0.2)
T1071	Application Layer Protocol	4: 404.45	4: 260 (0, 0.0)	10: 45 (6, -0.43)	3: 215 (-1, 0.14)
T1059.003	Command and Scripting Interpreter: Windows Command Shell	5: 402.92	6: 236 (1, -0.09)	9: 52 (4, -0.29)	7: 184 (2, -0.17)
T1071.001	Application Layer Protocol: Web Protocols	6: 366.61	7: 235 (1, -0.08)	15: 41 (9, -0.43)	6: 194 (0, 0.0)
T1082	System Information Discovery	7: 362.35	5: 250 (-2, 0.17)	21: 35 (14, -0.5)	4: 215 (-3, 0.27)
T1547	Boot or Logon Autostart Execution	8: 337.45	10: 193 (2, -0.11)	11: 45 (3, -0.16)	11: 148 (3, -0.16)
T1070	Indicator Removal on Host	9: 328.56	8: 213 (-1, 0.06)	18: 36 (9, -0.33)	8: 177 (-1, 0.06)
T1547.001	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	10: 322.45	13: 178 (3, -0.13)	12: 45 (2, -0.09)	14: 133 (4, -0.17)
T1083	File and Directory Discovery	11: 313.14	9: 204 (-2, 0.1)	22: 34 (11, -0.33)	9: 170 (-2, 0.1)
T1566	Phishing	12: 300.86	34: 89 (22, -0.48)	3: 66 (-9, 0.6)	93: 23 (81, -0.77)
T1204	User Execution	13: 300.02	28: 101 (15, -0.37)	4: 62 (-9, 0.53)	61: 39 (48, -0.65)
T1070.004	Indicator Removal on Host: File Deletion	14: 298.56	12: 183 (-2, 0.08)	19: 36 (5, -0.15)	12: 147 (-2, 0.08)
T1036	Masquerading	15: 290.45	15: 146 (0, 0.0)	13: 45 (-2, 0.07)	19: 101 (4, -0.12)
T1059.001	Command and Scripting Interpreter: PowerShell	16: 286.34	22: 113 (6, -0.16)	8: 54 (-8, 0.33)	39: 59 (23, -0.42)
T1204.002	User Execution: Malicious File	17: 279.18	33: 93 (16, -0.32)	5: 58 (-12, 0.55)	66: 35 (49, -0.59)
T1057	Process Discovery	18: 277.09	11: 184 (-7, 0.24)	31: 29 (13, -0.27)	10: 155 (-8, 0.29)
T1016	System Network Configuration Discovery	19: 270.3	14: 174 (-5, 0.15)	29: 30 (10, -0.21)	13: 144 (-6, 0.19)
T1566.001	Phishing: Spearphishing Attachment	20: 257.97	42: 75 (22, -0.35)	6: 57 (-14, 0.54)	104: 18 (84, -0.68)
Rank-biased overlap (RBO) Total			0.93	0.92	0.85
Rank-biased overlap (RBO) Top 20			0.85	0.64	0.8
Kendall's τ Total			0.89	0.86	0.7
Kendall's τ Top 20			0.72	0.27	0.66

evaluation ATT&CK in version 9.0 available in the STIX-format[3] is used. This paper's implementation and application of D3TECT is open source and available on Github[4].

3.2 D3TECT metrics and ranking

In the following section two ranking methods are presented:

(1) D3TECT Top Techniques (Tab. 1)
(2) D3TECT Top Data Sources (Tab. 2)

The results provide answers to the questions raised in Sect. 2.1 and are evaluated with data sets of well-known cyber security solution providers in Sect. 5.3.

3.2.1 D3TECT Top Techniques. If every attack on all systems is known and categorized according to uniform attack techniques (T), the question "Which attack techniques are most often (successfully) used?" can be answered. Since this will never be the case, the industry uses approximations. In this example the MITRE dataset is utilized. For each attack technique, there are examples that have been extracted from publicly available resources. A metric (E) is given by the sum of all examples where the technique was (successfully) used. The datasets distinguishes between:

(1) groups (GRP) that have used this technique.
(2) software, tools, scripts and malware (SW) that implement this technique.

The sum of examples per technique is referenced as $E_{GRP/SW}(T)$. To achieve even distribution of the Top Techniques, weighting is applied. Is is assumed that attack techniques used by groups are discovered less frequently compared to software examples. To support their goals, groups use publicly available software. In contrast to the rather static process of categorizing attack techniques through reverse engineering, attribution of groups is a more complex process[15]. This is also reflected in the dataset as group examples are underrepresented in their frequency. Therefore group examples receive a higher weight. A group weight (W_{GRP}) of the total count of group examples (E_{GRP}) relative to the total count of software examples (E_{SW}) is assigned. Group examples weight is calculated as follows: $W_{GRP} = \frac{E_{SW}}{E_{APT}}$.

The metrics listed are always linked to a technique (T) and can be expressed as follows:

- E_{GRP}, the count of group examples.
- E_{SW}, the count of software examples.
- $E_{TOT} = E_{GRP} + E_{SW}$, the count of all examples.
- $E_W = E_{GRP} * W_{GRP} + E_{SW}$, the sum of all software and group examples, relative to the total count of software and group examples of the given dataset.

Starting with version 7 of the ATT&CK framework, techniques are divided into main- and sub-techniques. The latter are more specific versions of main-techniques. For example "Command and Scripting Interpreter" (T1059) links to eight sub-techniques: PowerShell, AppleScript, Windows Command Shell, Unix Shell, Visual Basic, Python, JavaScript/JScript, Network Device CLI. Ranking can be done by taking the sub-techniques into consideration, or by merging sub- into main-techniques.

D3TECT's Top 20 Techniques include sub-techniques to achieve a more fine-grained overview. To better compare rankings to other threat intelligence (TI) data sets in Sect. 5, procedure examples of sub-techniques are summed up to their corresponding main-technique. The techniques "Command and Scripting Interpreter", "Application Layer Protocol", "Boot or Logon Autostart Execution", "Indicator Removal on Host", "Phishing", "User Execution" and "Masquerading" are part of the Top 20 when merged. While the largest relative change of "Command and Scripting Interpreter" moving from 62nd place to the first, and the largest absolut change "Boot or Logon Autostart Execution" moving from 401 to place eight.

[3] https://github.com/mitre-attack/attack-stix-data
[4] https://github.com/d3tect/d3tect

Table 2: D3TECT Top 20 Datasources (E_W)

Name	Rank $\sum E_W$	Rank $\sum T_{TOT}$	Rank $\sum E_{TOT}$	Rank $\sum E_{GRP}$	Rank $\sum E_{SW}$
Command: Command Execution (10/14)	1: 13834.52	1: 242 (0, 0.0)	1: 7376 (0, 0.0)	1: 2012 (0, 0.0)	1: 5364 (0, 0.0)
Process: Process Creation (12/14)	2: 12333.09	2: 196 (0, 0.0)	2: 6462 (0, 0.0)	2: 1829 (0, 0.0)	2: 4633 (0, 0.0)
Process: OS API Execution (8/14)	3: 6238.7	7: 77 (4, -0.4)	3: 3767 (0, 0.0)	4: 770 (1, -0.14)	3: 2997 (0, 0.0)
Network Traffic: Network Traffic Content (11/14)	4: 5476.34	4: 90 (0, 0.0)	4: 2735 (0, 0.0)	3: 854 (-1, 0.14)	4: 1881 (0, 0.0)
File: File Creation (10/14)	5: 4690.71	5: 82 (0, 0.0)	5: 2280 (0, 0.0)	5: 751 (0, 0.0)	6: 1529 (1, -0.09)
Network Traffic: Network Traffic Flow (11/14)	6: 4459.93	6: 82 (0, 0.0)	7: 2107 (1, -0.08)	6: 733 (0, 0.0)	8: 1374 (2, -0.14)
File: File Modification (8/14)	7: 3999.86	3: 95 (-4, 0.4)	6: 2183 (-1, 0.08)	7: 566 (0, 0.0)	5: 1617 (-2, 0.17)
Network Traffic: Network Connection Creation (10/14)	8: 3331.81	8: 58 (0, 0.0)	10: 1531 (2, -0.11)	8: 561 (0, 0.0)	11: 970 (3, -0.16)
Windows Registry: Windows Registry Key Modification (7/14)	9: 3327.08	9: 56 (0, 0.0)	8: 1889 (-1, 0.06)	10: 448 (1, -0.05)	7: 1441 (-2, 0.12)
Module: Module Load (6/14)	10: 3160.32	11: 49 (1, -0.05)	9: 1581 (-1, 0.05)	9: 492 (-1, 0.05)	9: 1089 (-1, 0.05)
Script: Script Execution (5/14)	11: 2469.07	15: 21 (4, -0.15)	12: 1291 (1, -0.04)	12: 367 (1, -0.04)	12: 924 (1, -0.04)
File: File Metadata (5/14)	12: 2441.45	13: 32 (1, -0.04)	11: 1334 (-1, 0.04)	14: 345 (2, -0.08)	10: 989 (-2, 0.09)
File: File Access (6/14)	13: 2210.81	12: 45 (-1, 0.04)	13: 1052 (0, 0.0)	13: 361 (0, 0.0)	13: 691 (0, 0.0)
Application Log: Application Log Content (10/14)	14: 1997.41	10: 50 (-4, 0.17)	15: 646 (1, -0.03)	11: 421 (-3, 0.12)	24: 225 (10, -0.26)
Windows Registry: Windows Registry Key Creation (3/14)	15: 1329.59	19: 15 (4, -0.12)	14: 755 (-1, 0.03)	16: 179 (1, -0.03)	14: 576 (-1, 0.03)
Service: Service Creation (6/14)	16: 1031.56	20: 14 (4, -0.11)	16: 595 (0, 0.0)	19: 136 (3, -0.09)	15: 459 (-1, 0.03)
Logon Session: Logon Session Creation (8/14)	17: 999.21	14: 31 (-3, 0.1)	25: 354 (8, -0.19)	15: 201 (-2, 0.06)	29: 153 (12, -0.26)
Process: Process Access (5/14)	18: 988.34	17: 18 (-1, 0.03)	17: 494 (-1, 0.03)	17: 154 (-1, 0.03)	20: 340 (2, -0.05)
User Account: User Account Authentication (7/14)	19: 919.13	16: 20 (-3, 0.09)	23: 428 (4, -0.1)	18: 153 (-1, 0.03)	23: 275 (4, -0.1)
Process: Process Metadata (4/14)	20: 884.83	25: 10 (5, -0.11)	18: 490 (-2, 0.05)	20: 123 (0, 0.0)	16: 367 (-4, 0.11)
Rank-biased overlap (RBO) Total		0.88	0.97	0.97	0.94
Rank-biased overlap (RBO) Top 20		0.9	0.96	0.95	0.91
Kendall's τ Total		0.72	0.96	0.92	0.91
Kendall's τ Top 20		0.77	0.91	0.89	0.82

D3TECT Top Techniques based on the MITRE ATT&CK dataset serve as an indicator and as a possible answer to the questions raised in Sect. 2.1. It is assumed that attack techniques implemented in ready to use software require low skill by the threat actor. Therefore $E_{SW}(T)$ is used to address the question "Which attack techniques require low skill by the threat actor?". Due to the nature of MITREs enterprise ATT&CK matrix and its intrinsic goal to list all attack techniques that target typical enterprise IT systems, the dataset also provides answers to the question "Which attack techniques are universally applicable?".

Table 1 shows a ranked version of the metrics sorted by group examples. Next to the rank, absolute and relative distance to rank E_W is noted in brackets. The lists are compared utilizing rank-biased overlap (RBO) [17] and Kendall's τ [3]. Since there is no weighting applied in the upper areas using Kendall's τ, the RBO provides more accurate results. Results are discussed in Sect. 5.1 and 5.3. Application of metrics and ranks is context- and data set related. For example, if the aim is the detection of hacking groups then E_{GRP} is in focus.

3.2.2 D3TECT Top Data Sources. Depending on the system in scope, the number of data sources typically range between one to several hundred to be tapped for detection. A complete picture of all data sources available is either gathered by analyzing the assets or attack techniques in scope. This is a demanding task that can be prioritized if risk of the latter is already assessed. ATT&CK already maps techniques and data sources, therefore this information is used in the **MATCH** phase and to **REASSESS & PRIORITIZE** data sources.

Due to different characteristics (e.g. resource consumption, detection capabilities) metrics are required to support prioritization. Characteristics of data sources and detailed, structured information about detection algorithms are missing in the data set. Nevertheless, combined with the results gathered in Sect. 3.2.1, the data is detailed and comprehensive enough to derive the following metrics:

- $\sum T_{TOT}$, the total number of attack techniques leaving traces in a given data source.
- $\sum E_{TOT}, \sum E_{GRP}, \sum E_{SW}, \sum E_W$, the sum of technique-metrics leaving traces in a given data source. The attack technique metrics are described in Sect. 3.2.1.

The Top 20 Data Sources are listed and ranked according to $\sum E_W$ in Tab. 2. Next to the data source name the tactic coverage is noted in brackets. "Command: Command Execution" is ranked highest in all variations of the metric. As with the Top Techniques, the absolute and relative ranking change are noted in brackets next to the metrics. The metrics are compared utilizing Kendall's τ as well as rank-biased overlap and are discussed in Sect. 5.1 and 5.3.

4 MODEL APPLICATION

Phase one, two and three of D3TECT are presented in Sect. 3.2. Threats, assets, attack techniques and data sources have been identified and assessed in the **IDENTIFY & ASSESS** phase. Matching was done in the **MATCH** phase then reassessed and prioritized in the **REASSESS & PRIORITIZE** phase. The data was extracted from the MITRE ATT&CK Framwork and metrics based on the information provided were defined in Sect. 3.2. In the following section, these results are used to **IMPLEMENT & VERIFY** detection capabilities. Results of this phase are fed back to **REASSESS & PRIORITIZE** several times, until detection is implemented to an acceptable level. This process is repeated until all the techniques are detected or no data sources are available to detect the remaining techniques.

4.1 Naive selection process

After **REASSESS & PRIORITIZE**, the most critical data source / algorithm combination for detection is implemented in the **IMPLEMENT & VERIFY** phase. Once implemented it is verified if and to what extent attack techniques are detected. In case of MITREs data set, data sources are in focus. The connection of data sources

to attack techniques is simplified and for the application in this section, it is assumed that a technique is always detected when the data source is processed. First, a suitable ranking is selected, e.g., in the following a ranking accounting for $\sum E_W$ is applied. If two or more data sources are ranked equally, $\sum T_{TOT}$ and then the DS name is utilized. The most critical data source is implemented and detection is verified. The detected attack techniques are excluded from further consideration. The results are fed back to **REASSESS & PRIORITIZE** and it is again evaluated, which data source is the most critical, but taking only remaining attack techniques into account. A simplified algorithm for this process is illustrated in Alg. 1. Beginning with the most critical data source, according to for example metric $\sum E_W$, mapped techniques are evaluated and marked as detected. If a technique is detected, it is no longer considered in the next iteration of subsequent data sources. Since metrics of data sources rely on techniques in detection, metrics of data sources are recalculated for the next iteration so that ranking changes are reflected in the selection. Data source excludes are considered and examined in detail in the following section.

Algorithm 1: Iterative selection of a minimal set of data sources, beginning with the most important according to *metric*

```
datasources ← all data sources for detection
Function getOptimalDatasourceSelection(metric, datasourceExcludes):
    for datasource in datasources sorted by metric decending do
        if datasource in datasourceExcludes then
            continue
        for technique in datasource.techniques do
            if not technique is detected then
                set technique is detected
                set datasource detects techniques
                datasource.techniqueDetections ← append technique
        if datasource detects techniques then
            optimalDatasourceSelection ← append datasource
        for datasource in datasources do
            ajdust dataSource.metric to datasource.techniqueDetections
            ensure that next datasource has highest dataSource.metric
    selectionMissedTechniques ← all techniques not detected by selection
    sort optimalDatasourceSelection by metric
    return optimalDatasourceSelection, selectionMissedTechniques
```

When applied on the data set given, a total count of 31 data sources have to be processed to detect all relevant attack techniques. In the following all data sources beginning with the most important according to metric $\sum E_W$ are listed:

(1) Command: Command Execution, $\sum E_W$ 13834.52 (63.74% / 63.74%), $\sum T_{TOT}$ 242 (51.27% / 51.27%)
(2) Network Traffic: Network Traffic Content, $\sum E_W$ 4246.34 (19.56% / 83.31%), $\sum T_{TOT}$ 71 (15.04% / 66.31%)
(3) Process: Process Creation, $\sum E_W$ 1205.74 (5.56% / 88.86%), $\sum T_{TOT}$ 26 (5.51% / 71.82%)
(4) File: File Metadata, $\sum E_W$ 739.68 (3.41% / 92.27%), $\sum T_{TOT}$ 13 (2.75% / 74.58%)
(5) User Account: User Account Authentication, $\sum E_W$ 507.84 (2.34% / 94.61%), $\sum T_{TOT}$ 15 (3.18% / 77.75%)
(6) Process: OS API Execution, $\sum E_W$ 477.92 (2.2% / 96.81%), $\sum T_{TOT}$ 20 (4.24% / 81.99%)
(7) Network Traffic: Network Traffic Flow, $\sum E_W$ 220.93 (1.02% / 97.83%), $\sum T_{TOT}$ 14 (2.97% / 84.96%)
(8) File: File Creation, $\sum E_W$ 119.36 (0.55% / 98.38%), $\sum T_{TOT}$ 9 (1.91% / 86.86%)
(9) Application Log: Application Log Content, $\sum E_W$ 85.99 (0.4% / 98.77%), $\sum T_{TOT}$ 11 (2.33% / 89.19%)
(10) Driver: Driver Load, $\sum E_W$ 56.31 (0.26% / 99.03%), $\sum T_{TOT}$ 1 (0.21% / 89.41%)
(11) Drive: Drive Modification, $\sum E_W$ 52.47 (0.24% / 99.28%), $\sum T_{TOT}$ 2 (0.42% / 89.83%)
(12) Active Directory: Active Directory Credential Request, $\sum E_W$ 42.68 (0.2% / 99.47%), $\sum T_{TOT}$ 4 (0.85% / 90.68%)
(13) File: File Content, $\sum E_W$ 33.05 (0.15% / 99.62%), $\sum T_{TOT}$ 2 (0.42% / 91.1%)
(14) Logon Session: Logon Session Creation, $\sum E_W$ 23.63 (0.11% / 99.73%), $\sum T_{TOT}$ 5 (1.06% / 92.16%)
(15) User Account: User Account Modification, $\sum E_W$ 18.84 (0.09% / 99.82%), $\sum T_{TOT}$ 5 (1.06% / 93.22%)
(16) Firmware: Firmware Modification, $\sum E_W$ 15.42 (0.07% / 99.89%), $\sum T_{TOT}$ 4 (0.85% / 94.07%)
(17) File: File Access, $\sum E_W$ 8.21 (0.04% / 99.93%), $\sum T_{TOT}$ 1 (0.21% / 94.28%)
(18) Drive: Drive Access, $\sum E_W$ 6.21 (0.03% / 99.96%), $\sum T_{TOT}$ 1 (0.21% / 94.49%)
(19) Cloud Storage: Cloud Storage Access, $\sum E_W$ 4.21 (0.02% / 99.98%), $\sum T_{TOT}$ 1 (0.21% / 94.7%)
(20) File: File Modification, $\sum E_W$ 3.0 (0.01% / 99.99%), $\sum T_{TOT}$ 11 (2.33% / 97.03%)
(21) Logon Session: Logon Session Metadata, $\sum E_W$ 2.0 (0.01% / 100.0%), $\sum T_{TOT}$ 1 (0.21% / 97.25%)
(22) Instance: Instance Creation, $\sum E_W$ 0.0 (0.0% / 100.0%), $\sum T_{TOT}$ 3 (0.64% / 97.88%)
(23) Snapshot: Snapshot Creation, $\sum E_W$ 0.0 (0.0% / 100.0%), $\sum T_{TOT}$ 2 (0.42% / 98.31%)
(24) Cloud Service: Cloud Service Disable, $\sum E_W$ 0.0 (0.0% / 100.0%), $\sum T_{TOT}$ 1 (0.21% / 98.52%)
(25) Cloud Service: Cloud Service Enumeration, $\sum E_W$ 0.0 (0.0% / 100.0%), $\sum T_{TOT}$ 1 (0.21% / 98.73%)
(26) Cloud Storage: Cloud Storage Enumeration, $\sum E_W$ 0.0 (0.0% / 100.0%), $\sum T_{TOT}$ 1 (0.21% / 98.94%)
(27) Firewall: Firewall Disable, $\sum E_W$ 0.0 (0.0% / 100.0%), $\sum T_{TOT}$ 1 (0.21% / 99.15%)
(28) Image: Image Creation, $\sum E_W$ 0.0 (0.0% / 100.0%), $\sum T_{TOT}$ 1 (0.21% / 99.36%)
(29) Instance: Instance Deletion, $\sum E_W$ 0.0 (0.0% / 100.0%), $\sum T_{TOT}$ 1 (0.21% / 99.58%)
(30) Instance: Instance Modification, $\sum E_W$ 0.0 (0.0% / 100.0%), $\sum T_{TOT}$ 1 (0.21% / 99.79%)
(31) User Account: User Account Creation, $\sum E_W$ 0.0 (0.0% / 100.0%), $\sum T_{TOT}$ 1 (0.21% / 100.0%)

In contrast to Tab. 2 ranking depends on previous decisions and only 31 of the 99 data sources of the data set have to be implemented to detect all techniques that are linked to at least one data source. Attacks without data sources are discussed in Sect. 5.2.1.

With perfect detection applied on the top four data sources, already 92.27% of weighted procedure examples ($\sum E_W$) and 74.58% of techniques ($\sum T_{TOT}$) are detected.

4.2 Selection process under constrains

In real world scenarios, some data sources (and algorithms on top) cannot be implemented because of budgetary or technical limitations, contractual issues, or tapping of data is simply not permitted by the asset owner. In some cases legal regulations limit granularity

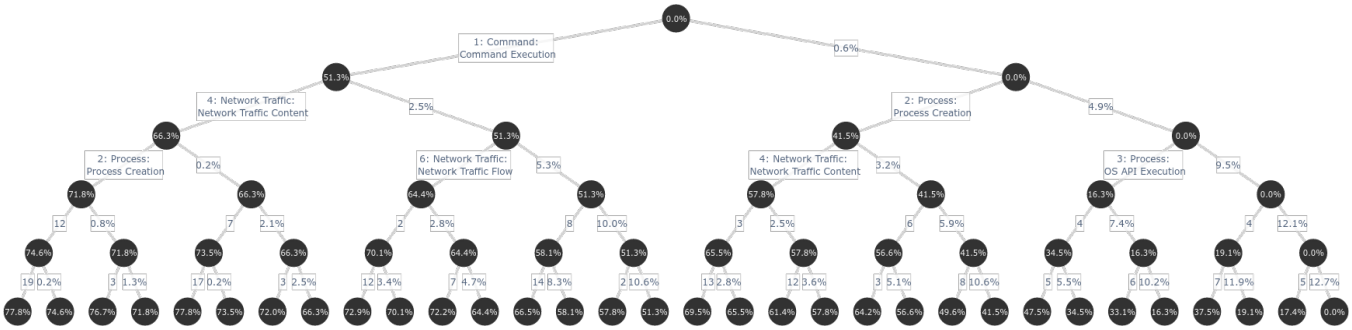

Figure 4: Detection selection decision tree.

and algorithms to be used, as it is the case with personal data in the EU [6]. Figure 4 presents a binary decision tree that illustrates different selection paths, described by Alg. 2.

Algorithm 2: Create binary decision tree

```
detectableTechniques ← all techniques detectable (at least one data source mapping)
Node (datasourceIncludes, datasourceExcludes, lastNode):
    Function __init__:
        optimalDatasourceSelection, selectionMissedTechniques ←
          getOptimalDatasourceSelection(metric, datasourceExcludes)
        set NodeText coverage by datasource.techniqueDetections along
          datasourceIncludes of detectableTechniques in percent
        if not lastNode then
            if optimalDatasourceSelection length is zero then set lastNode
              True
            createLeftNode()
            createRightNode()
    Function createLeftNode:
        datasourceIncludes ← append top datasource of
          optimalDatasourceSelection
        Node(datasourceIncludes, datasourceExcludes, lastNode)
        set LeftEdgeText top datasource name of
          optimalDatasourceSelection
    Function createRightNode:
        datasourceExcludes ← append top datasource of
          optimalDatasourceSelection
        Node(datasourceIncludes, datasourceExcludes, lastNode)
        set RightEdgeText coverage by selectionMissedTechniques of
          detectableTechniques in percent
```

Nodes of the decision tree represent (NodeText) the coverage rate of attack techniques detected through the data sources within the path. Edges to the left (LeftEdgeText) denote data sources included in the selection path, edges to the right (RightEdgeText) represent percentage of techniques undetectable when excluding the data source on the left hand side; meaning the percentage of techniques that leave no traces in the remaining data sources in the path. For example, if "Command: Command Execution" is excluded, techniques 'T1609'[4], 'T1003.005'[4] and 'T1059.008'[4] are undetectable[4], representing 0.6% of all techniques but just 0.1%[4] of the total weight. Because of layout limitations the data source rank is noted instead of the corresponding name beginning at a depth of four. In the case of Fig. 4 metric $\sum E_W$ is used. The binary tree illustrated in Alg. 2 first evaluates optimal data source selection (see Alg. 1) and spawns new left and right nodes. Under perfect conditions and without any excludes, the left path leads to early detection of critical techniques according to metric $\sum E_W$. In real world scenarios, data sources are excluded, this is illustrated following the path to the right. If the path length is zero, which means there are no detecting data sources remaining in the path, the recursion finishes.

[4]https://github.com/d3tect/d3tect/tree/main/work

5 DISCUSSION

5.1 Critical techniques

The strict adherence to the order of D3TECT Top Data Sources (Sect. 3.2.2) hinders detection of critical techniques. Figure 5 highlights attacks that are of high risk according to the ranking, but are considered at a late stage of detection implementation following the Top Data Sources. The most critical techniques can be found on the bottom left side. This plot is particularly interesting taking E_{GRP} into consideration, as it shows relevant APT techniques that might evade detection. Each dot represents a technique beginning with the highest rank (E_{GRP}) on the x-axis and the maximum data source value ($\sum E_W$) in which the technique leaves traces on the y-axis. Critical techniques are located in the lower left corner. Techniques on the bottom left, for example "T1203 Exploitation for Client Execution", are not mapped to any data source and therefore are "undetectable", but are from high risk according to metric E_{GRP}.

"T1078 Valid Accounts" is a critical technique with rank 16, and is not considered within the Top 10, but in the Top 20 data sources: "Logon Session: Logon Session Creation" ($\sum E_W$ rank 17) and "User Account: User Account Authentication" ($\sum E_W$ rank 19). These two data sources are not part of the Top 20 if ranked by $\sum E_{T_{TOT}}$ and $\sum E_{SW}$. The importance of "User Account: User Account Authentication" is also visible in Sect. 4.1 as it is the fifth data source and based on the data set processing of this data source is sufficient for the detection of T1078.

Detection of "T1068 Exploitation for Privilege Escalation" is specific to the vulnerable component and differs depending on the use case. The data set only maps "Driver: Driver Load" as detection data source with rank 21 according to $\sum E_W$. Again, the selection

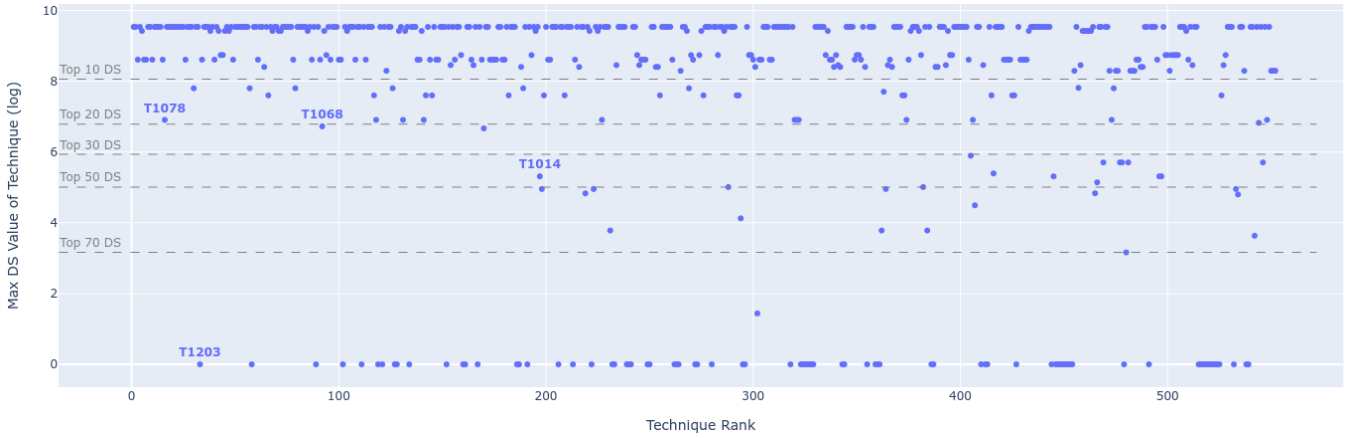

Figure 5: Critical techniques (E_{GRP}) to be missed following the Top Data Sources ($\sum E_W$).

process prioritizes "Driver: Driver Load" even more and as the tenth data source to be implemented. This is because T1068 is not detected by any other of the first nine data sources mentioned in Sect. 4.1. Furthermore, it is the only detectable technique of "Driver: Driver Load" that was not already detected by its predecessors. Another example is "T1014 Rootkit" that is detected only by "Firmware: Firmware Modification" (rank 47) and "Drive: Drive Modification" (rank 55). These data sources are part of the selection process with rank 16 and 11. The results highlight the importance of the selection process presented in Sect. 4.1 and Sect. 4.2.

5.2 Limits of the data set and metrics

The attack techniques in the data set are rather generic, and procedures of threats that implement these techniques differ in many details. Furthermore, procedures change and evolve over time. This is especially true for APTs, as these groups undertake great efforts to remain undetected. The analysis on data sources of the quite generic attack techniques is not yet in focus of the framework and therefore results of Sect. 3 and Sect. 4 should be taken with caution, as further discussed in Sect. 5.3.

The underlying algorithms required for detection are not part of the data set. Before version 9.0 of the framework, data sources where mostly sensor focused, for example "Anti-Virus" or "SSL/TLS inspection". Furthermore confidence of detection is usually increases when operating on multiple data sources. Attack details and real world examples that use a technique are shown in the ATT&CK framework's technique detail pages (Fig. 3). These examples are taken from public incidents reports. In order to allow a qualitative evaluation, it can therefore be assumed that the number of examples corresponds approximately to the spread of real APT attacks and their techniques used. However, the quality also depends on the following characteristics:

(1) The number of articles discussing publicly known cases in which attackers use these techniques.
(2) The quality (correctness, completeness, traceability, consistency) of the
   (a) processing of the editors in the analysis and publication of the results.
   (b) review by MITRE.
(3) The systematic classification of techniques and data sources according to the same scheme.
(4) The timely publication and integration of new techniques.
(5) The quality of the approaches of the APTs, taking into account that high-quality APT operations only leave traces of intent.
(6) The skill levels of defenders and analysts for general detection and specific detection of zero-day techniques.
(7) The political and commercial influence under which the materials are published and processed.
(8) The free availability of material online.
(9) The availability of the material in English.

5.2.1 Limits of data sources linked. A total of 80 techniques in ATT&CK version 9.0 do not have any data source attached at all. Most of these techniques are located in the tactic resource-development and reconnaissance. A resource-development technique for example is "T1585.001 Establish Accounts: Social Media Accounts", which is most often not explicitly investigated by institutions but might come from a TI data provider or via an information sharing platform. Some information is gained passively by attacks utilizing other sources not in control by an institution and therefore not in scope of detection systems, for example "T1589.003 Gather Victim Identity Information: Employee Names".

In contrast to the rather passive tactics mentioned, some techniques are not fully elaborated. "T1203 Exploitation for Client Execution" has no data source linked, but in ATT&CK v8 the data sources "Anti-virus, Process monitoring, System calls" were associated. Sensor-centric data sources, like "Anti-Virus" are not mentioned in newer versions of ATT&CK. This is also the case for "T1200 Hardware Additions" which once had "Asset management, Data loss prevention" noted, or "T1211 Exploitation for Defense Evasion" with former data sources "File monitoring, Process monitoring, Windows Error Reporting".

5.3 Benchmarking results with other datasets

From security vendors, to TI data vendors and security enthusiasts – mapping of attack techniques to the ATT&CK Framework is becoming an industry standard. There are various reports that give details about the current state of threat procedures mapped to attack techniques that are openly accessible. For example, the security company FireEye lists 211 techniques in their M-Trends Report 2021 [7]. The use of 63% of MITRE's attack techniques were observed by FireEye's security experts in 2020. In this report FireEye presents their findings and maps 211 techniques to a percent value which represents the use of these technique in all their investigated intrusions. Several other companies categorize their findings according to the ATT&CK Framworks techniques, for example:

- Sophos with their "Active Adversary Playbook 2021"[5] which lists about 54 techniques from attacks during 2020/2021. These techniques are without ranking and based on telemetry data of their security products, as well as manual findings of the threat-intelligence team.
- RedCanary with their "Threat Detection Report 2021" [9] lists about 19 ranked techniques from of their endpoint detection telemetry data analytic's, not manually reviewed by detection engineers.
- Cisco Talos Incident Response publishes quarterly blog posts[6] on techniques most frequently observed without ranking.
- PricewaterhouseCoopers (PwC) also lists[7] their most observed techniques in 2020.
- This is followed by others, like Rapid7[8] with about 23 techniques and McAfee[9] with 36 techniques (ranked 5 per tactic).

Out of all reports listed above, FireEye's report is most detailed in terms of attributed techniques and ranking. Therefore, results of Sect. 3.2 and 4.1 are compared with the structured data extracted from [7]. The Top 20 Techniques are compared against D3TECT's Top Techniques in Tab. 3. Sect. 3.2 ranking method delivers almost identical results in the Top 2, 50% of the Top 10 Techniques can be found in both FireEye and in D3TECT's Top Techniques.

[5] https://news.sophos.com/en-us/2021/05/18/the-active-adversary-playbook-2021/
[6] https://blog.talosintelligence.com/2021/03/ctir-trends-winter-2020-21.html
[7] https://www.pwc.co.uk/cyber-security/pdf/pwc-cyber-threats-2020-a-year-in-retrospect.pdf
[8] https://www.rapid7.com/research/report/2020Q2-threat-report/
[9] https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-threats-apr-2021.pdf

Table 3: Benchmark - FireEye Top 20 Techniques

Name	Rank	RedCanary	Rank $\sum E_W$
Obfuscated Files or Information	1	13 (12, -0.86)	2 (1, -0.33)
Command and Scripting Interpreter (CSI)	2	1 (-1, 0.33)	1 (-1, 0.33)
CSI: PowerShell	3	8 (5, -0.45)	16 (13, -0.68)
System Services (SS)	4	15 (11, -0.58)	83 (79, -0.91)
SS: Service Execution	5	17 (12, -0.55)	84 (79, -0.89)
Remote Services (RS)	6		36 (30, -0.71)
RS: Remote Desktop Protocol	7		66 (59, -0.81)
Indicator Removal on Host (IRH)	8		9 (1, -0.06)
Ingress Tool Transfer	9	14 (5, -0.22)	3 (-6, 0.5)
System Information Discovery	10		7 (-3, 0.18)
File and Directory Discovery	11		11 (0, 0.0)
Exploit Public-Facing Application	12		104 (92, -0.79)
Obtain Capabilities (OC)	13		155 (142, -0.85)
OC: Code Signing Certificates	14		336 (322, -0.92)
Subvert Trust Controls (STC)	15		61 (46, -0.61)
STC: Code Signing	16		69 (53, -0.62)
IRH: File Deletion	17		14 (-3, 0.1)
Process Injection	18	12 (-6, 0.2)	32 (14, -0.28)
Encrypted Channel (EC)	19		28 (9, -0.19)
EC: Asymmetric Cryptography	20		98 (78, -0.66)
Rank-biased overlap (RBO) Total		0.26	0.62
Rank-biased overlap (RBO) Top 20		0.26	0.44

To not only compare the results to one data set, RedCanary's report [9] is taken into consideration. An empty value means, that the technique is not represented in RedCanary's comparatively small data set. Kendall's τ requires the same elements in both lists and can not be calculated for smaller data sets. "T1569 System Services" and its sub-techniques procedure examples are underrepresented in MITREs data set. RedCanary also attributes more relevance to these techniques with rank 15 and 17. When extending the analysis with extracted TI data from the DeTTECT[10] project in version 27de915[11], T1569 is only listed in three (FireEye, RedCanary, Sophos) of ten TI data sets. DeTTECT extracted the TI vendors ATT&CK attribution data from the reports listed above in machine readable form.

In addition FireEye puts greater importance to the "Remote Services" techniques, which are most often used by groups and are accordingly slightly higher considering E_{GRP}. "T1021.001 Remote Services: Remote Desktop Protocol" is listed in six out of ten TI data sets. "T1588.003 Obtain Capabilities: Code Signing Certificates" also shows significant differences. While there is just one procedural example within the MITRE data set, FireEye attributes 21% usage to this technique. Besides FireEye no other TI data set notes this techniques as critical. "T1190 Exploit Public-Facing Application" is the last technique within the Top 20 that has a large relative ranking change (> +/- 0,7). T1190 is noted in 40% of the reports including FireEye, Sophos, McAfee and Cisco Thalos.

Summarizing, the most comprehensive publications in this area by now, are those by FireEye and the evaluation in this paper utilizing the MITRE ATT&CK data set itself. The results of the two publications overlap significantly, and in comparison with the other sources by a wide margin. The largest and most detailed release to date came from FireEye, which is superseded with the release of D3TECT's Top Techniques. When comparing the Top 20 to FireEye's Top 20 vice-versa, the same RBO is achieved both overall and within the first 20 results. Still, the data sets show differences in some cases. For example: "T1036 Masquerading", rank 15 in D3TECT and rank 16 in RedCanary, is ranked 108 in FireEye's data set. T1036 is found in six out of ten TI data sets and therefore a very common technique. In general it can be said that attribution is not uniform across all datasets. This is partly due to the different ways in which this data is collected. Vendors have different expertise (malware analysis, intrusion detection system telemetry) and different attribution approaches of more than 500 techniques.

5.3.1 Data sources. D3TECT's Top Data Sources results are compared with data from FireEye's M-TRENDS 2021 report [7], when extracted, the largest, publicly available data set ranking attributed MITRE ATT&CK attack techniques. MATCHING and REASSESS & PRIORITIZE is done as described in Sect. 3.2.2 utilizing attack technique / data source links from the ATT&CK Framework. Because of the size of RedCanary's data set, ranking of data sources is only applied to FireEye's and D3TECT's Top Techniques shown in Fig. 4. The two data sets provide almost the same results when applied with the ATT&CK Framework's links with only slight changes in data source prioritization.

Table 4: Benchmark - FireEye Top 20 Data Sources

Name	Rank	Rank $\sum E_W$
Command: Command Execution	1	1 (0, 0.0)
Process: Process Creation	2	2 (0, 0.0)
Process: OS API Execution	3	3 (0, 0.0)
Network Traffic: Network Traffic Content	4	4 (0, 0.0)
File: File Modification	5	7 (2, -0.17)
Network Traffic: Network Traffic Flow	6	6 (0, 0.0)
Windows Registry: Windows Registry Key Modification	7	9 (2, -0.12)
File: File Creation	8	5 (-3, 0.23)
Module: Module Load	9	10 (1, -0.05)
File: File Metadata	10	12 (2, -0.09)
Script: Script Execution	11	11 (0, 0.0)
Network Traffic: Network Connection Creation	12	8 (-4, 0.2)
Application Log: Application Log Content	13	14 (1, -0.04)
Logon Session: Logon Session Creation	14	17 (3, -0.1)
Service: Service Creation	15	16 (1, -0.03)
File: File Content	16	23 (7, -0.18)
Windows Registry: Windows Registry Key Creation	17	15 (-2, 0.06)
Windows Registry: Windows Registry Key Deletion	18	25 (7, -0.16)
File: File Deletion	19	24 (5, -0.12)
User Account: User Account Authentication	20	19 (-1, 0.03)
Rank-biased overlap (RBO) Total		0.91
Rank-biased overlap (RBO) Top 20		0.91

A solid selection (Sect. 4.1) requires a extensive, detailed, elaborated data basis. Apart from the attack framework, this is most likely to be the case with FirEye's data set. Others, like RedCanary, do not provide enough techniques and are not sufficient for solid results. Without data source exclusions RedCanary's selection includes three data sources. When applied on FireEye's data, 20 data sources are elaborated. The RBO of the data source selection indicates medium to high similarity with a value of 0.73. Some data sources in D3TECT's selection seem to be missing compared to FireEye, but when analyzed more closely, this is due to ranking changes and the selection of more critical data sources that achieve the same, or greater coverage. Compared to D3TECT's selection, ten data sources are missing in FireEye's selection, for example "Firmware: Firmware Modification", "User Account: User Account Modification", "File: File Access" as well as Cloud and Virtualization related data sources that are from importance when dealing with sophisticated threats. The results highlight the importance of a large and most complete data set when applying D3TECT's selection process of Sect. 4.1.

[10] https://github.com/rabobank-cdc/DeTTECT/tree/master/threat-actor-data
[11] Commit: 27de9154282a499463a3a681365b4e76a1267cda

6 RELATED WORK

Since the last century, research is concerned with intrusion analysis and detection. Stoll [14] and Cheswick [5] observed malicious activities at a time when the Internet was first commercialized. Cheswick [5] documented his efforts baiting and trapping a malicious actor in a chroot jail to observe his behavior. Cheswick came to the conclusion, that if a hacker obtains a login, he would become root sooner or later. Decades later, research is still concerned with cyber threats. Multiple [1, 8] frameworks exist to describe adversarial behavior. With the ATT&CK Framework [15], MITRE creates the most complete public available list of attack techniques on enterprise networks. There are numerous publications on adversarial behavior and most widely used techniques by modern threats. Examples were given in Sect. 5.3 and utilized [7, 9] to benchmark D3TECT's implementation. D3TECT's Top Techniques is now the largest publicly available data set on the most often used techniques replacing [7] taking the ATT&CK Framework as a base.

Attribution of attacks is still a major challenge [4, 12, 13]. Without information sharing, the model presented could not be evaluated. "Malware Information Sharing Platform" (MISP) [16] is a malware and threat sharing platform to collect and share important Indicators of compromise. The "Standardizing Cyber Threat Intelligence with the Structure Threat Information eXpression (STIX)" [2] projects made efforts to standardize cyber threat intelligence information that can be shared.

Zimmermann [18] addresses the issue of detection sensor placement and the selection of data sources in a practical approach. He provides best practice recommendations from many years of Security Operation Centre experience. However, Zimmermann remains incident-agnostic and without a documented process for strategic data source selection. Sanders and Smith [10] present the Applied Collection Framework (ACF). ACF is designed to support organizations in selecting data sources based on risk. Data sources are identified beginning with the most critical assets and selected according to management's economic decision. In contrast to Sanders and Smith's ACF, D3TECT accounts for constraints in the selection process so that even if a certain data source cannot be utilized in a specific setting, the most important attack techniques are still covered by the remaining data sources.

7 CONCLUSION

Recent research has provided a more complete understanding of attacker behavior. In addition to modern analysis tools and models that support the understanding of complex, multi-stage attacks, information sharing is key to advanced cyber attack detection systems. This paper introduced D3TECT, a novel process model and procedure for dynamically ranking and selecting data sources suitable for detection. The novelty is that this model accounts for constraints in the selection process so that even if a certain data source cannot be utilized in a specific setting, e.g., due to data privacy constraints, the early detection of the most important attack techniques is still ensured by the remaining data sources. The model was tested with real data, utilizing the MITRE ATT&CK Framework and numerous public cyber threat intelligence databases. The results underline the importance of an extensive, detailed and well elaborated data set to be fed into D3TECT to achieve best results. Further, the paper demonstrated that the metrics derived from the ATT&CK Framework using D3TECT's approach deliver similar, but by far more extensive results than those publicly available. Since the extracted features are not the main focus of MITRE, inaccuracies in the elaboration of data sources, as well as the procedural examples of the framework were identified. Besides, attack techniques vary considerably in terms of detectability, as different and sometimes multiple data sources have to be analyzed. In addition capabilities of detection systems differ with respect to accuracy of detection.

Future work deals with a more sophisticated algorithm that, for example, takes capabilities of detection algorithms to discover an attack technique into account. Our data set and metrics will be expanded accordingly.

ACKNOWLEDGMENTS

This work was partly funded by the FFG projects SPOTTED (grant no. FO999887725) and CyberMonoLog (grant no. FO999886330).

REFERENCES

[1] Jim Aldridge. 2012. *Remediating Targeted-threat Intrusions.* Technical Report. MANDIANT, Alexandria, VA. 10 pages. https://www.fireeye.com/content/dam/fireeye-www/global/en/products/pdfs/wp-bh2012-aldridge-remediation.pdf
[2] Sean Barnum. 2014. Standardizing cyber threat intelligence information with the Structured Threat Information eXpression (STIX™). *MITRE Corporation, July* (2014).
[3] J. M. Bevan and M. G. Kendall. 1971. *Rank Correlation Methods.* Vol. 20. 74 pages. https://doi.org/10.2307/2986801
[4] Sergio Caltagirone, Andrew Pendergast, and Christopher Betz. 2013. The Diamond Model of Intrusion Analysis. *Threat Connect* 298 (2013). Issue 0704.
[5] Bill Cheswick. 1992. An Evening with Berferd in Which a Cracker is Lured, Endured, and Studied. *In Proceedings of the Winter USENIX Conference* (1992).
[6] European Commission. 2016. REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC. *Official Journal of the European Union* (2016), 88. https://eur-lex.europa.eu/eli/reg/2016/679/oj
[7] Inc. FireEye. 2021. *M-TRENDS 2021.* Technical Report. FireEye, Milpitas, CA. 82 pages. https://www.fireeye.com/current-threats/annual-threat-report.html
[8] Eric M Hutchins, Michael J Cloppert, and Rohan M Amin. 2011. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Leading Iss. in Inf. Warfare & Sec. Research* 1 (2011), 80.
[9] RedCanary. 2020. *2021 Threat Detection Report.* Technical Report. RedCanary, Denver, CO. 122 pages. https://redcanary.com/threat-detection-report/
[10] Chris Sanders and Jason Smith. 2014. *Applied Network Security Monitoring: Collection, Detection, and Analysis.*
[11] IBM Security. 2021. *IBM: Cost of a Data Breach Report.* Technical Report. IBM Corporation. Issue 8. https://doi.org/10.1016/s1361-3723(21)00082-8
[12] Florian Skopik. 2017. *Collaborative cyber threat intelligence: detecting and responding to advanced cyber attacks at the national level.* CRC Press.
[13] Florian Skopik and Timea Pahi. 2020. Under false flag: using technical artifacts for cyber attack attribution. *Cybersecurity* 3 (2020). Issue 1. https://doi.org/10.1186/s42400-020-00048-4
[14] Clifford Stoll. 1988. Stalking the wily hacker. *Commun. ACM* 31 (1988). Issue 5. https://doi.org/10.1145/42411.42412
[15] Blake E Strom, Andy Applebaum, Doug P Miller, Kathryn C Nickels, Adam G Pennington, and Cody B Thomas. 2020. *MITRE ATT&CK - Design and Philosophy.* Technical Report March. McLean, VA. 46 pages. https://attack.mitre.org/docs/ATTACK_Design_and_Philosophy_March_2020.pdf
[16] Cynthia Wagner, Alexandre Dulaunoy, Gérard Wagener, and Andras Iklody. 2016. MISP - The design and implementation of a collaborative threat intelligence sharing platform. *WISCS 2016 - Proceedings of the 2016 ACM Workshop on Information Sharing and Collaborative Security, co-located with CCS 2016.*
[17] William Webber, Alistair Moffat, and Justin Zobel. 2010. A Similarity Measure for Indefinite Rankings. *ACM Trans. Inf. Syst.* 28, 4 (2010), 38. https://doi.org/10.1145/1852102.1852106
[18] Carson Zimmerman. 2014. Ten Strategies of a World-Class Cybersecurity Operations Center.