Actionable Cyber Threat Intelligence for Automated Incident Response

Cristoffer Leite[1,2], Jerry den Hartog[1], Daniel Ricardo dos Santos[2], and Elisa Costante[2]

[1] Eindhoven University of Technology, 5612 AZ Eindhoven, Netherlands
[2] Forescout Technologies, 5612 AB Eindhoven, Netherlands

Abstract. Applying Cyber Threat Intelligence for active cyber defence, while potentially very beneficial, is currently limited to predominantly manual use. In this paper, we propose an automated approach for using Cyber Threat Intelligence during incident response by gathering Tactics, Techniques and Procedures available on intelligence reports, mapping them to network incidents, and then utilising this map to create attack patterns for specific threats. We consider our method actionable because it provides the operator with contextualised Cyber Threat Intelligence related to observed network incidents in the form of a ranked list of potential related threats, all based on patterns matched with the incidents.
We evaluate our approach with publicly available samples of different malware families. Our analysis of the results shows that our method can reliably match network incidents with intelligence reports and relate them to these threats. The approach allows increasing the automation of its use, thus addressing one of the major limiting factors of effective use of suitable Cyber Threat Intelligence.

1 Introduction

In our ever more digital and online society it is essential for organizations to properly protect themselves against cyber threats. Related information, so called Cyber Threat Intelligence (CTI), includes analysed knowledge about capabilities, infrastructure, methods, and victims of cyber threat actors. As such, this intelligence has the potential to help organizations to better perform threat detection, incident response, threat hunting, and risk management as well as to make strategic decisions to protect themselves. And the standardisation of CTI information allows organisations to gather this intelligence from different CTI sources.

Threat Intelligence can be divided in different groups based on their level of detail and long-term use, including Technical and Tactical [1]. Examples of Technical CTI include Indicators of Compromise (IoC) such as hashes of infected files, known malicious IP addresses and domain names. Some of this Technical CTI is easy to use: IoCs can be matched with network traffic or endpoint information in real-time to generate alerts that indicate a network intrusion is taking

place. However, this relies on aspects that are easy for attackers to change, for example by simply acquiring new infrastructure or recompiling a malware with slightly different code. This limits how helpful such CTI use is for the detection of more sophisticated attacks.

Tactical intelligence describes not just isolated IoCs but also the Tactics, Techniques and Procedures (TTPs) used by adversaries. TTPs are useful for incident response because they are harder for an attacker to change than IoCs. The problem with Tactical intelligence is that, although there are plenty initiatives for standardisation and usage of Cyber Threat Intelligence (CTI), there currently is no easy, automatic way to ingest it into threat detection systems. This results in a lack of automated Tactical CTI use on incident response [2] [3] [4] and by consequence most incident response teams verify CTI manually if at all.

In this paper we present a solution that allows the use of higher level CTI in an easier and (semi)-automated way. We do that by automating the gathering of relevant CTI reports, associating it with known threats described by these reports, then mapping them to network patterns of observable events. Our core contributions include allowing the (automated) use of CTI during detection and incident response, providing context to alerts in form of ranked related CTI, and the automation of an important step in the response by providing the useful alerts.

With this method, work that is currently done manually, or not at all, can be largely automated and done as soon as a threat becomes known. This process can be highly beneficial for current scenarios were the lack of a similar automation makes it hard to properly respond to incidents. Our method can also be done as a preparation step for any company that wants to improve its usage of threat intelligence in advance rather than only when unknown incidents are observed on the organization's network.

We evaluate the approach by taking 27 samples of 4 different families of malware/ransomware, automatically generating patterns for them and cross-validating the usage of patterns between these families. We find that the approach is able to build patterns based on intelligence reports that capture the families with high accuracy and thus provide context to network incidents. We also find that availability of suitable high-level CTI is currently limited. However, with an easier way to consume such CTI by linking it to network events now in place, the value of such CTI increases. To further support the availability of usable CTI we plan to look at improving the generation of (higher level) CTI in future work.

The remainder of the paper is organised as follows. Section 2 introduces the general scenario on how Cyber Threat Intelligence is currently used, with the motivation for a more automated approach, and then summarizes the state-of-the-art in actionability and automation for CTI. Section 3 presents our methodology and its core components in detail. Section 4 describes our implementation as well as its chosen platforms and formats for our tests, and then discusses the experimental results. An finally Section 5 highlights our conclusions and future work directions.

2 Background and Related Work

In this section we sketch how Cyber Threat Intelligence (CTI) is currently used in network intrusion detection also defining some related terminology, and then review related work.

2.1 Current Situation

Figure 1 shows an example setup of a Network Intrusion Detection System (NIDS) generating alerts about incidents for analysis while using some form of CTI. Blacklists are usually implemented by sub modules of the NIDS, while the event correlation could also be implemented externally by a Security Information and Event Management (SIEM) or a Security Orchestration, Automation and Response (SOAR).

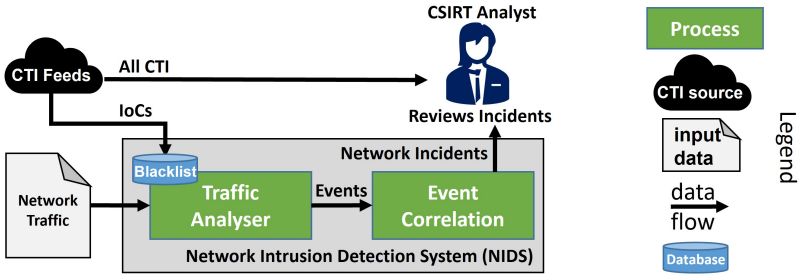

Fig. 1. A Diagram of a Simple NIDS Solution

To identify attacks based on (correlated) NIDS events, a Cyber Security Incident Response Team (CSIRT) usually resorts to CTI available on a Threat Intelligence Platform (TIP) or in specific intelligence feeds. CTI is usually aggregated in the form of reports related to a threat or a campaign, although a report can also contain a compilation of many other reports. According to the authors on [4], while doing a Threat Intelligence and Attack Path Analysis for example, it is necessary to acquire information about observables from a TIP to see if they match known Indicator of Compromise (IoC)s.

Features from events monitored in a network are called *observables*. When an atomic observable, like an IP address or payload hash, is potentially linked to security breaches, it is called an Indicator of Compromise (IoC). IoCs are non-contextual CTI that allow incident response to be executed in an automated (or semi-automated) manner, e.g. through blacklisting, which also holds as a valid approach even during surges of data to analyse. But for higher-level CTI, such as Tactical and Operational, analysts manually review events in network incidents and compare them with reports from CTI feeds. This manual approach is typically for active defence using higher-level CTI [2].

Yet the more contextual information provided by these higher-level CTI is needed in the majority of the cases to properly analyse incidents. The manual process to acquire required intelligence becomes a problem especially when attack campaigns flood the NIDS with information, creating surges of data to be analysed.

As the authors on [2] point out, active defence with the help of CTI is mostly done manually for these cases. In this scenario, some automation is required to improve the capability of responding to threats, which would allow the CSIRT to better act on the intelligence received. Thus, automation of its application on incident response would require making the use of CTI more actionable [3].

2.2 Related Work

Most of the recent works on Cyber Threat Intelligence focus on managing Indicators of Compromise [5], gathering unstructured Open Source Cyber Threat Intelligence to extract Indicators, Tactics, Techniques and Procedures from them, with a broad usage of Natural Language Processing (NLP) for these cases [6, 7], and assessing the quality of Open Source Cyber Threat Intelligence [8–11] or the formats used by them [12]. Some also focus on generating CTI from network events for specialised use case scenarios [13], and improving the visualisation of CTI.

There is a broad acceptance that there is a need for more semi-automated or actionable forms of consuming Cyber Threat Intelligence during incident response [2]. Actionability in this context is the capability of reacting during network incidents while using the knowledge provided by CTI. As mentioned, there are some recent works aiming to make the use or creation of CTI more actionable by including semi-automated mechanisms. A notable work on the use of NLP is done by the authors on [14], where they propose a trigger mechanism to create an actionable CTI discovery system. It focus on portraying the relationship between IoCs and campaign stages to generate actionable CTI from intelligence reports by using NLP. They try to explain the attack stages by using keywords that would represent the specific stage.

Some other works try to focus on standardisation and on how platforms can improve the overall presentation of their reports and their usability. The authors on [15] present a list of software functions that should be implemented by CTI sharing platforms in order to support the intelligence cycle to generate actionable threat intelligence. They focus on specific functions that could be added to these platforms in order to improve the actionability of generated CTI when shared between multiple entities.

The work on [12] compares how different data formats try to standardise the integration of response mechanisms such as firewalls with the feedback received from existing CTI artifacts and precise identification of workflows. They provide an analyses on how the integration of standardised formats for CTI representation and decision making on incident response can lead to more precise defensive actions.

As for implementation of actionable CTI on defense, the authors on [16] propose the integration of the information from intelligence platforms into Security-Policy-Controlled Systems (SPCS) in a more automated manner. By assuming that the information extracted from this platforms support detection of threats and that SPCS focus on responding to threats, they suggest two approaches to integrate these detection and response scenarios: A direct integration with intelligence obtained is received and processed directly by security-critical systems, and an indirect one where it is integrated to the security tools used by the organisation.

The work [17] combines a preparation step with application of Cyber Threat Intelligence to improve the usage of Indicators of Compromise on Incident Response. It does not focus on more behavioural aspects linked to Tactics, Techniques and Procedures, but rather on revealing patterns of malicious activities by correlating IoCs from multiple malware instances to CTI reports available in different sources.

Using self-gathered intelligence, the work on [18] applies a deep learning model to link exploits from the Dark Web to vulnerabilities in a bidirectional manner. They include an attention mechanisms to automatically link exploits to their post date and vulnerability severity. They focus on using this self-gathered intelligence to assist cybersecurity professionals to prioritisation and risk management efforts.

In another similar approach, the authors on [13] create their own CTI specifically for energy systems by analysing metering infrastructures and disseminating the gathered intelligence through a energy cloud platform. Then, they combine their internal and external CTI to generate security policies that can control the behavior of their internal devices, or create rules about device isolation to block the operation of malicious processes.

The closest method to ours is the service provided by Hybrid Analysis [3] where after executing a malware sample in a sandboxed environment, they generate a list of processes executions and correlate them with host-based Tactics, Techniques and Procedures (TTPs). Although they don't use these to integrate CTI into detection and response, it follows a similar approach by linking these two sources.

A considerable amount of the presented works prioritise generating better formats, analysing existing ones or suggesting new capabilities to sharing platforms in a way to allow the use of CTI together with response mechanisms. A small number actually tries to use available CTI in a more automated way or focus on applying it to incident response.

From the discussion above and to the best of our knowledge, there is not any work that focuses on suggesting a methodology that increases the automation and the actionability of available CTI on incident response by linking known attacker behaviour from reports to network incidents.

[3] https://www.hybrid-analysis.com/

3 Methodology

In this section, we describe our solution for adding automation and actionability to the use of higher level CTI in network intrusion detection and incident response. As depicted in Figure 2, we introduce a Pattern Module that will match network incidents to related higher-level CTI. This directly provides the analyst with contextual information for matching incidents. Related incidents matching the same pattern can also be grouped to help further reduce the workload.

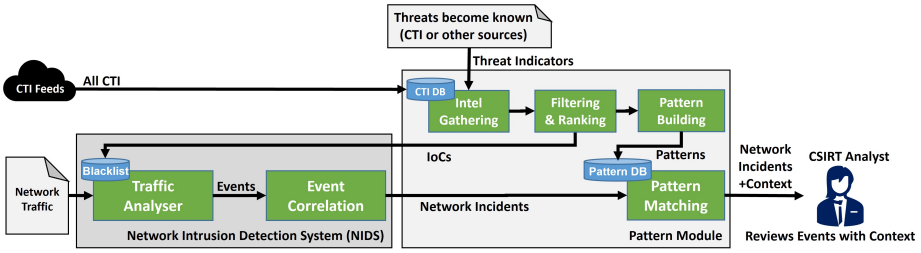

Fig. 2. Proposed Solution with a Pattern Creation Engine for Actionable CTI on Network Events

During attack campaigns, network events can have single or multiple occurrences of a single type. Thus, combining events and merging recurrent or redundant ones during visualisation can help to reduce the effort necessary on analysing similarities between known attacks and situations emerging from observables.

The information flow of the pattern module detailed in Figure 3 is divided into four main steps that cover pattern creation and their use. Creation of patterns is triggered by threats and reliable related indicators becoming known. These indicators can, for example, come directly from sandboxed samples of a threat or from a TIP where initial related CTI is found. The first step is then **Intelligence Gathering**, in which the indicators are matched with available CTI. Next in the **CTI Filtering and Ranking** phase low level CTI is filtered out and just provided to the NIDS for blacklisting as in the currect situation (Figure 1). High level CTI are ranked based on their usefulness for network intrusion detection. All sufficiently high scoring CTI report can be used to build patterns, but optionally the ranking can be presented to an analyst for manual adjustments and exclusion of certain CTI report. In the following **Pattern Building** step, the CTI is matched with TTPs from MITRE ATT&CK and mapped into network detectable events. We combine these events into a pattern which is stored, including related information, for matching against future network incidents. Finally in **Pattern Matching** previously created patterns are compared with network incidents, and matching ones are enriched with the information from the pattern.

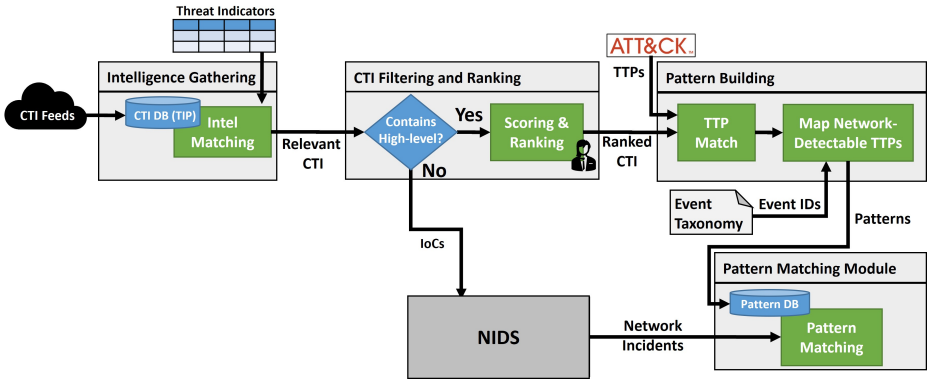

Fig. 3. The Information Flow of our Solution

3.1 Intelligence Gathering

To trigger the Intelligence Gathering phase, a number of indicators related to a known threat are gathered. These indicators are the starting point to define what CTI identify or characterise the threat, and they will be used to match with reports that point to it. The relevance of a type of indicator depends on the threat, e.g. file hashes of payloads or exploit downloader files might be more useful for identifying reports about an instance of a Ransomware than the IPs used by it in a specific campaign.

At the time the process is triggered, our data base will have been filled with reports from CTI feeds, some related to the threat we are currently considering. With an initial list of indicators related to a threat, we select all CTI reports in the our database that include at least one of the indicators. If these reports include additional indicators, we add these to our initial list and iterate this process until no new reports are added. The outcome of this process is a set of reports with relevant CTI.

3.2 CTI Filtering and Ranking

We need to find, amongst the Relevant CTI, those reports that are most useful to then build patterns with in the pattern module. To that end we first filter reports that only contain low-level CTI. The IoCs included in these reports are provided to the NIDS (for addition to blacklists as in existing solutions) as shown in Figure 3. But these reports are not used when building patterns as they do not include behavioural information. Next, we rank the remaining ones on their usefulness for attack patterns that are compatible with data coming from a NIDS. i.e. on the quality of their information regarding attack execution plans and methods that can help identify an ongoing attack by observing the network.

To define the the level of intelligence of a CTI, we refer to the expanded Detection Maturity Level (DML) Model [19,20] as shown in Figure 4. We define low-level CTI as those of DML 1 or 2, equivalent to the group of Technical intelligence, as described by [1]. (CTI with DML 0 is ignored as it is does not contain relevant information by definition.) Higher-level CTI is defined as those with DML 3 or higher, which includes Tactical (TTPs), Operational (Goals and Strategy) and Strategic (Identity) CTI. This separation also matches with the different perspectives of CTI described by the authors on [12], where IoCs are artifacts, TTPs describe the attacker behaviour, and the higher levels indicate the response. For reports, we define the level based on the highest level of CTI included in the report, so a report is low-level if it only contains low-level CTI and high-level if it contains at least one high-level CTI.

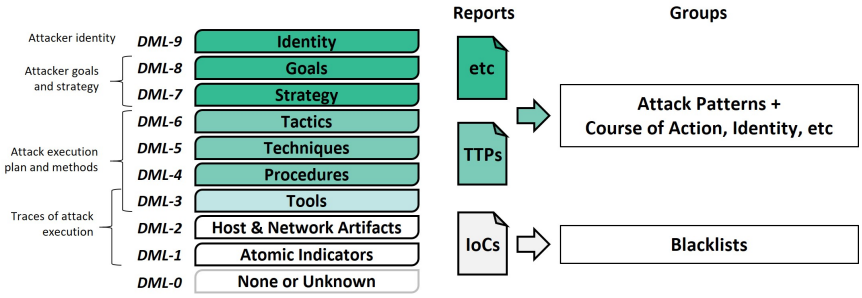

Fig. 4. Filtering CTI with the Detection Maturity Level Model [20]

Not all high-level information is useful for network based detection. If related to network detectable events, Tactical (TTPs) CTI might be useful to capture the attacker's behaviour detectable by a NIDS. Operational (Goals and Strategy) and Strategic (Identity) CTI on the other hand can be useful for the analist in planning a response. We thus needs to find which TTPs are *network-mappable*, i.e. related to network detectable events.

As TTPs are typically expressed in term of the MITRE ATT&CK framework [**?**], we use that framework to find which TTPs are network-mappable, as described in more detail in Subsection 4.1, assuming that the robustness of that framework allows an adequate coverage of up-to-date TTPs.

The initial ranking score assigned to reports is how many network-mappable TTPs they contain. In principle any CTI with a sufficiently high score can be used to build patterns. However, after the automated ranking, security analysts can manually check the list of reports related to an attack and adjust the rank accordingly, thus adapting which reports will be used in the pattern building. This (optional) review step is included because, as mentioned before, one of the problems with Open-Source CTI (OSCTI) is the quality of its reports in regards to coverage and inter-report conciseness [8–11].

By the end of this step, the output is a ranked list of reports that include network-mappable CTI which will be used for the creation of the attack patterns. The next step will be the translation of the CTI into the pattern itself. Note that the actual scores are only relevant for presentation to and evaluation by the analyst in the review step. For the Pattern Building, it only matters whether reports are included or not.

3.3 Pattern Building

With a list of CTI reports ranked, it is possible to start extracting information from them. Each object type related to attack execution plan and methods in a report needs a specific mapping to a network event to allow an explainable result. In order to achieve that, events in the network were ordered in a taxonomy based on their types. Then, event types were mapped to related TTPs. Patterns are formed from event types related to the TTPs mentioned in the ranked CTI reports.

An *event type* describes the behaviour of an observable possibly related to a threat. The list of event types used by the NIDS is the result of an aggregation of many threat data resources, including Industrial Control Systems Cyber Emergency Response Team (ICS-CERT) and NIST National Vulnerability Database (NVD). Event types are arranged in a taxonomy tree that indicates first where the event comes from (alerts or logs), then its variations per additional level, with a short representation by an event type ID. For example, the event type ID *alert_ops_net_unscon* is an Unstable Connection network issue that falls in the operational category of alert events, while *alert_ops_net_netmis* would be a Network Misconfiguration in the same category.

Mapping some event types from the NIDS to ATT&CK was done by using the four lower stages of the framework for adversarial threat hunting described by the authors on [21]. We aim to link event types to TTPs, which we do by looking if the type is related to observable which belong to a TTP. Gunter's framework gives the notion of observables as being the result of a step in the attack, and being related to TTPs.

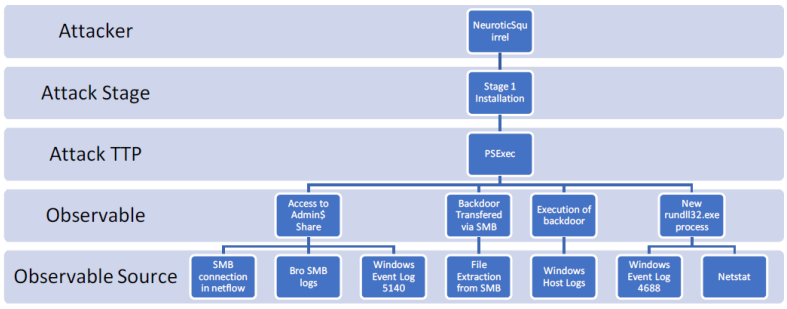

Fig. 5. Gunter's Framework for Adversarial Threat Hunting with PSExec [21]

A NIDS could monitor possible sources of observables to detect them as network events (of a certain type). Figure 5 taken from [21] shows an example where threat hunting for PSExec (a tool used to run processes remotely using any user's credentials) gives related observables and sources that can be linked to TTPs in the higher stages. As such it provides the information needed to create the required mapping.

In our scenario, the NIDS itself is the Observable Source, and the types of events detected by it are the Observables. The TTPs are directly linkable to ATT&CK. From our analysis, each event type can be mapped to one or two Techniques on ATT&CK, e.g. a security alert related to FTP CMD buffer overflow attempt is mappable to both Network Denial Of Service (T1498) and Exploitation of Remote Services (T1210)[4]. With the map between events and techniques, we create a pattern by using the TTPs in a report.

```
{
    "name": "Chain of Malicious Activities",
    "description": "Alerts generated by Malicious Act",
    "events": [
         alert_sec_event_type1
         alert_sec_event_type2
    ],
    "regex":[
       "[ab]"
    ],
    "base_severity": "HIGH",
    "time_window_size": "day",
    "category": "SECURITY"
}
```

Fig. 6. An Example of a Pattern in JSON Format with Additional Information

Regular expressions stand out as enablers of feature-rich, translatable and actionable format for representing CTI on incident response. They can describe patterns of events that can be processed by a Finite State Machine (FSM), creating automation and actionability.

The part of a *pattern* used for detection consists of a group of event types. Additionally we include some meta data with a pattern. As such, a pattern is defined by its name, an array of its events' IDs, a regular expression to represent these events in a computer-readable format and the time window size, and optional fields such as its severity, a short description, and its category. These optional fields can be gathered from any relevant CTI, with DML 7 to 9, irrespective it the report that contains is is ranked for inclusion in the (detection part of the) pattern or not. The pattern meta data also includes (links to) the reports with related CTI. Figure 6 shows an example of a pattern in JSON format. The event IDs are represented in the regex by a letter according to the order that they appear in the array of event types, i.e the letter *a* in the example refers to *alert_sec_event_type1*.

[4] https://attack.mitre.org/techniques/enterprise/

Any pattern built is added to a database of patterns for use by the Pattern Matching.

3.4 Pattern Matching

With the database of patterns, we then used them to match sets of events occurring in the network within the specific time window defined in each the pattern. As shown in Figure 2 for our proposed solution, the NIDS constantly monitors the network traffic, tries to match events in the network with known threats and then feeds the incidents to the Pattern Matching Module. The module then aggregates these events according to their related machine in the network and tries to match them with its known patterns.

We analyse events as static unordered information over the entire time window and check for candidate patterns. A pattern can be considered as a candidate pattern in two situations: When there a multiple event types observed in the network in that pre-defined time window and they have at least two events types from the pattern, or when there are multiple events observed in the network but all of them are from the same single type of an event in the pattern.

After getting a list of candidate patterns, we rank them based on their confidence level. Candidates patterns have their confidence level analysed by calculating the *Pattern Predominance* (P) in the events observed in the network, which is the percentage of all events observed (E) that match any event types (ε) from the pattern, i.e. $P = \varepsilon/E$. A pattern is considered to be matched if the pattern predominance is higher than a threshold t ($P \geq t$) which can be set according to the needs of the analysts. Any matched pattern is then added to a list of possible good sources of CTI that can help on responding to that incident, ranked by their Pattern Predominance.

With this methodology, an analyst or a CSIRT can use the CTI reports related to the matched patterns in a semi-automated manner for incident response. Next section evaluates this approach by implementing it and testing the creation of patterns on sandboxed scenarios.

4 Implementation and Evaluation

This Section details the implementation of the methodology described in Section 3, with explanations about the experimental setup and results obtained.

4.1 Implementation

We detail below the formats and TIPs chosen as source for information, as well as the thresholds set for minimum compatibility of patterns.

Intelligence Gathering

TIP. Several public CTI feeds are available, like AlienVault, VirusTotal, Malware Traffic Analysis and Hybrid Analysis. There are also open-source platforms, such as MISP and OpenCTI for implementing a TIP and optionally offering its own combined info as a new feed. We run an OpenCTI instance to collect CTI reports from the four feeds mentioned above. We select OpenCTI as it has a slight focus on more contextualised information for indicators and is capable of linking them to related threats and also to their to primary source (a report, a MISP event, etc). OpenCTI is also able to consume MISP generated feeds.

Internal CTI format. In our implementation, we use reports in the Structured Threat Information Expression (STIX) 2.1 format for its versatility in exchanging CTI and also because it is a widely adopted standard. In STIX we represent CTI as structured objects called STIX Domain Object (SDO)s and reports as containers called STIX Bundles.

Intel matching. We match which reports (expressed as STIX bundles) include given threat indicators (also expressed in STIX) to get all the SDOs related to those samples.

Pattern Building

TTP Match. In this step, information about the TTPs is requested using MITRE's API and then added to the report. This is done for all network-mappable TTPs.

Mapping Events to TTPs. The NIDS can only operate network-based events, which makes it compatible with only a subset of MITRE ATT&CK. We use MITRE's diagram of techniques and their linked data sources [22] to filter the interesting ones for a NIDS. A total of 1.267 event types from the NIDS we use were mapped to network mappable techniques from both MITRE ICS and Enterprise. For each event type, only a single TTP or the two most relevant ones were assigned. As a result, only some of the mappable TTPs have related event types. Figure 7 shows a list of network mappable MITRE techniques from their Enterprise framework and a highlight for the ones mapped.

Pattern Matching Module

Pattern Matching. The threshold for pattern predominance, used to determine pattern compatibility in pattern matching (see Subsection 3.4), is set to 0.5 based our empirical experimentation. Thus patterns with at least 50% of predominance ($P \geq 0.5$) are considered as being matched during the pattern analysis. Note that low-confidence approximates, i.e. candidate patterns with low predominance ($P < 0.5$), can be suggested for analysts if needed (clearly marked with as being low-confidence).

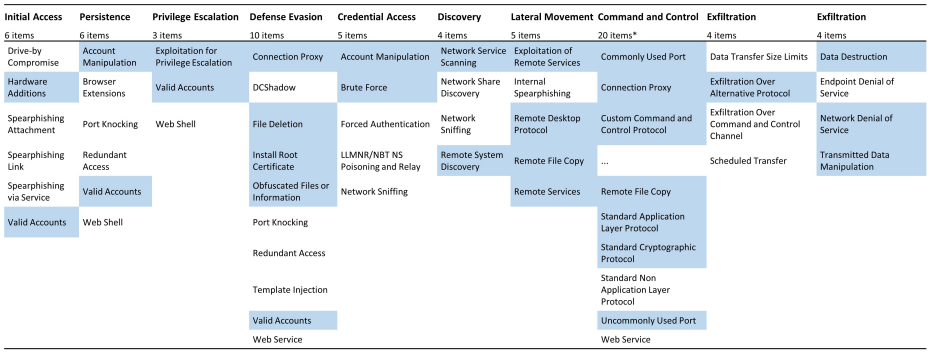

Fig. 7. Network Detectable Techniques from MITRE with the Ones Used in Blue

4.2 Experimental Setup

Dataset. For our experiments, we use a dataset consisting of 27 sand-boxed samples of ransomware from 4 different families: Cerber, Crysis, REvil/Sodinokibi and WannaCry [23], with a total of 78.5 GB in PCAPs. Each sample refers to an instance of a ransomware from one of the families. These PCAPS represent the network traffic and the I/O operations executed by these malwares while encrypting a network shared directory.

The use of these samples was decided upon based on some factors: Malwares and more specifically ransonwares are a growing threat industry-wide, an open dataset with good availability and adoption helps the reproductibility of the experiments, malware families with a considerable number of samples will allow a proper validation, availability of OSCTI data related to these samples allows the creation of the patterns.

Evaluation Metrics In the tests, we want to verify if a pattern made out of a CTI report is strong enough to define and match a sample from a malware family rather than a single instance, and if it is unique enough to differentiate malware families. To evaluate the suitability of our methodology, we then want to check if the patterns: (1) Match same-family samples with a high score. (2) Do not match different families or match with a low enough score.

Mapping Malwares to Patterns. We divide our malware families in two groups: One is the group of sample with indicators that point to reports with higher level of CTI available, which will be then used to generate the initial patterns. Based on observations about the available CTI related to them, we selected Cerber and Sodinokibi/REvil as the sources for the attack patterns as part of the first group. And the second group consists of all samples that will be used for the validation, including the ones used to create the patterns themselves.

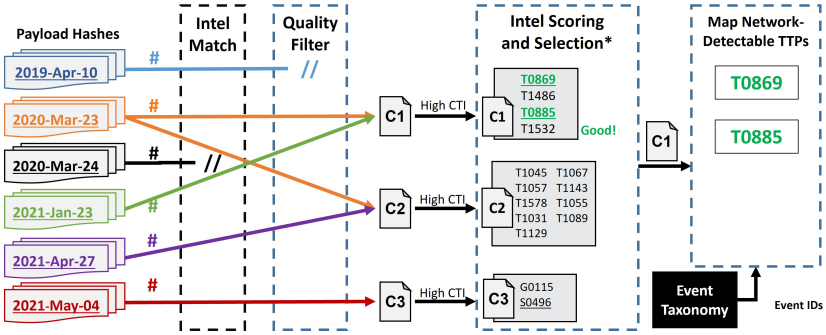

Fig. 8. Mapping Cerber Samples to Related High Level Network-Detectable CTI

Figure 8 shows the process of mapping the samples from Cerber to related CTI and extracting network mappable TTPs from them. Sodinokibi/REvil followed a similar flow. To create actionable information out of reports related to these malwares, we use as an starting point a list of hashes related to the malware payloads from each sample. It is possible to match which reports as STIX bundles include these payloads to get the SDOs related to those samples. These reports are then filtered based on the level of CTI they have, and then ranked based on two scores: The percentage of related payload hashes they include from that initial list, and the amount of contextual CTI on these reports. In the case of similar reports, they can be grouped based on their related SDOs

In our case, we decided to use the hashes from the samples themselves as the starting point to search for related CTI in a way of validating if the patterns can detect related samples, but as stated before, there is also the possibility of extracting this information from other sources, such as the reports themselves.

Fig. 9. Final Patterns Generated for Cerber and REvil

Using the first group, we generated patterns for the malware families and then added them to our database. Figure 9 shows the resulting patterns. We replay the samples from that malware family in a network monitored by the NIDS, which sends the events to the pattern module for scoring and validation.

After that, we run the same experiment again using the samples from all other families as a validation step. Next section shows the results of our experiments.

4.3 Results

Out of the eleven samples from Cerber instances, all of them matched with the pattern, we define eight of them as high confidence matches, because they have at least two events types matching and $t \geq 0.5$, and the remaining two as medium confidence because there is only one event type match, but $t = 1$. At the same time, there has been no match with other malware samples. One sample, *REvil-2021-May-04*, did not have any anomalous events detected by our NIDS, and by consequence it did not appear as anomalous on our observations. We ran our tests using the patterns presented and analysed the results from the main experiment described on Subsection 4.2. Figure 10 shows the results using the pattern generated for Cerber.

Dataset	Candidate Pattern	Predominance (ε/#)	Confidence	Final Result
REvil-2019-Apr-10	No	0	-	No
REvil-2020-Jan-23	No	0	-	No
REvil-2020-Mar-23	No	0.012	-	No
REvil-2020-Mar-24	No	0	-	No
REvil-2021-Apr-27	No	0	-	No
REvil-2021-May-04		0		
Cerber-2016-Oct-03	Yes	0.988	High	MATCH
Cerber-2016-Oct-04	Yes	0.995	High	MATCH
Cerber-2016-Oct-12	Yes	0.993	High	MATCH
Cerber-2016-Oct-31	Yes	0.994	High	MATCH
Cerber-2016-Nov-28	Yes	0.995	High	MATCH
Cerber-2016-Dec-11	Yes	1	Medium	MATCH
Cerber-2016-Dec-22	Yes	0.995	High	MATCH
Cerber-2017-Jan-04	Yes	1	Medium	MATCH
Cerber-2017-Jan-20	Yes	1	Medium	MATCH
Cerber-2017-Jan-24	Yes	0.995	High	MATCH
Cerber-2017-Feb-06	Yes	0.995	High	MATCH
Crysis-2016-Dec-19	No	0	-	No
Crysis-2017-Jan-01	No	0	-	No
Crysis-2018-Nov-19	No	0	-	No
Crysis-2018-Dec-27	No	0	-	No
Crysis-2020-Jun-18	No	0	-	No
Crysis-2020-Aug-20	No	0	-	No
Crysis-2020-Aug-23	No	0	-	No
WannaCry-2017-May-16	No	0	-	No
WannaCry-2021-Aug-09	No	0	-	No
WannaCry-2021-May-04	No	0	-	No

Fig. 10. Final Results for Detection with Cerber

Figure 11 shows the results of the experiment now using the pattern generated for REvil. With the exception of the same sample as mentioned above, all the others matched the pattern created with high confidence. As mentioned before, the sample that did not match was not detected as anomalous. This may be due to the map presented on Figure 7 not being broad enough to include events

related to its incidents. In this test, there has been also one erroneous match with a sample from the WannaCry ransomware family. All the other sample did not match with the REvil pattern.

Dataset	Candidate Pattern	Predominance (ε/#)	Confidence	Final Result
REvil-2019-Apr-10	Yes	0.625	High	MATCH
REvil-2020-Jan-23	Yes	0.972	High	MATCH
REvil-2020-Mar-23	Yes	0.662	High	MATCH
REvil-2020-Mar-24	Yes	0.699	High	MATCH
REvil-2021-Apr-27	Yes	0.585	High	MATCH
REvil-2021-May-04		0		
Cerber-2016-Oct-03	No	0.006	-	No
Cerber-2016-Oct-04	No	0	-	No
Cerber-2016-Oct-12	No	0	-	No
Cerber-2016-Oct-31	No	0	-	No
Cerber-2016-Nov-28	No	0	-	No
Cerber-2016-Dec-11	No	0	-	No
Cerber-2016-Dec-22	No	0	-	No
Cerber-2017-Jan-04	No	0	-	No
Cerber-2017-Jan-20	No	0	-	No
Cerber-2017-Jan-24	No	0	-	No
Cerber-2017-Feb-06	No	0	-	No
Crysis-2016-Dec-19	No	0	-	No
Crysis-2017-Jan-01	No	0	-	No
Crysis-2018-Nov-19	Yes	0.074	Low	No
Crysis-2018-Dec-27	No	0	-	No
Crysis-2020-Jun-18	No	0.500	-	No
Crysis-2020-Aug-20	No	0.035	-	No
Crysis-2020-Aug-23	No	0.250	-	No
WannaCry-2017-May-16	Yes	0.162	Low	No
WannaCry-2021-Aug-09	Yes	0.705	High	MATCH
WannaCry-2021-May-04	No	0.333	-	No

Fig. 11. Final Results for detection on REvil

The detection achieved a False Negative Rate of 5.88% when considering the REvil sample with no events detected as incidents by the NIDS. At the same time, the detection achieved a False Positive Rate of 2.77%. The results give a Detection Rate of 94.11% with an accuracy of 96.22%.

5 Conclusion

The work proposed in this paper aims to increase the actionability of using CTI on incident response. Our methodology helps to define a structured way of consuming available CTI by linking them to known threats and their expected behaviour. It enables the use of the gathered intelligence by matching its attack patterns with network events related to incidents.

The test scenarios showed that by using this approach, it is possible to correlate intelligence about known threats with a good precision by using the map between the behavioural CTI and the incidents by using their event type. For future work, we plan to use our methodology to generate CTI reports out of the network incidents and to create an advanced version of our patterns that include event chains.

References

1. D. Chismon and M. Ruks, "Threat Intelligence: Collecting, Analysing, Evaluating," *MWR InfoSecurity*, p. 36, 2015.
2. D. Schlette, "Cyber Threat Intelligence," in *Encyclopedia of Cryptography, Security and Privacy*, S. Jajodia, P. Samarati, and M. Yung, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2021, pp. 1–3. [Online]. Available: http://link.springer.com/10.1007/978-3-642-27739-9_1716-1
3. P. Nespoli, D. Papamartzivanos, F. Gómez Mármol, and G. Kambourakis, "Optimal Countermeasures Selection Against Cyber Attacks: A Comprehensive Survey on Reaction Frameworks," *IEEE Communications Surveys Tutorials*, vol. 20, no. 2, pp. 1361–1396, 2018, conference Name: IEEE Communications Surveys Tutorials.
4. A. Groenewegen and J. Janssen, *TheHive Project: The maturity of an open-source Security Incident Response platform*, Jun. 2021.
5. Y. Gao, X. LI, H. PENG, B. Fang, and P. Yu, "HinCTI: A Cyber Threat Intelligence Modeling and Identification System Based on Heterogeneous Information Network," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2020, conference Name: IEEE Transactions on Knowledge and Data Engineering.
6. P. Gao, F. Shao, X. Liu, X. Xiao, Z. Qin, F. Xu, P. Mittal, S. R. Kulkarni, and D. Song, "Enabling Efficient Cyber Threat Hunting With Cyber Threat Intelligence," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, Apr. 2021, pp. 193–204, iSSN: 2375-026X.
7. N. Afzaliseresht, Y. Miao, S. Michalska, Q. Liu, and H. Wang, "From logs to Stories: Human-Centred Data Mining for Cyber Threat Intelligence," *IEEE Access*, vol. 8, pp. 19 089–19 099, 2020, conference Name: IEEE Access.
8. A. Tundis, S. Ruppert, and M. Mühlhäuser, "On the Automated Assessment of Open-Source Cyber Threat Intelligence Sources," in *Computational Science – ICCS 2020*, V. V. Krzhizhanovskaya, G. Závodszky, M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos, and J. Teixeira, Eds. Cham: Springer International Publishing, 2020, vol. 12138, pp. 453–467, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/978-3-030-50417-5_34
9. U. Noor, Z. Anwar, J. Altmann, and Z. Rashid, "Customer-oriented ranking of cyber threat intelligence service providers," *Electronic Commerce Research and Applications*, vol. 41, p. 100976, May 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1567422320300533
10. R. Brown and R. M. Lee, "2021 SANS Cyber Threat Intelligence (CTI) Survey," p. 19, 2021.
11. A. Berndt and J. Ophoff, "Exploring the Value of a Cyber Threat Intelligence Function in an Organization," in *Information Security Education. Information Security in Action*, ser. IFIP Advances in Information and Communication Technology, L. Drevin, S. Von Solms, and M. Theocharidou, Eds. Cham: Springer International Publishing, 2020, pp. 96–109.
12. D. Schlette, M. Caselli, and G. Pernul, "A Comparative Study on Cyber Threat Intelligence: The Security Incident Response Perspective," *IEEE Communications Surveys Tutorials*, vol. 23, no. 4, pp. 2525–2556, 2021, conference Name: IEEE Communications Surveys Tutorials.
13. S. Gong and C. Lee, "Cyber Threat Intelligence Framework for Incident Response in an Energy Cloud Platform," *Electronics*, vol. 10, no. 3, p. 239, Jan. 2021, number: 3 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2079-9292/10/3/239

14. J. Liu, J. Yan, J. Jiang, Y. He, X. Wang, Z. Jiang, P. Yang, and N. Li, "TriCTI: an actionable cyber threat intelligence discovery system via trigger-enhanced neural network," *Cybersecurity*, vol. 5, no. 1, p. 8, Dec. 2022. [Online]. Available: https://cybersecurity.springeropen.com/articles/10.1186/s42400-022-00110-3
15. C. Sauerwein, D. Fischer, M. Rubsamen, G. Rosenberger, D. Stelzer, and R. Breu, "From Threat Data to Actionable Intelligence: An Exploratory Analysis of the Intelligence Cycle Implementation in Cyber Threat Intelligence Sharing Platforms," in *The 16th International Conference on Availability, Reliability and Security*, ser. ARES 2021. New York, NY, USA: Association for Computing Machinery, Aug. 2021, pp. 1–9. [Online]. Available: https://doi.org/10.1145/3465481.3470048
16. P. Amthor, D. Fischer, W. E. Kühnhauser, and D. Stelzer, "Automated Cyber Threat Sensing and Responding: Integrating Threat Intelligence into Security-Policy-Controlled Systems," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, ser. ARES '19. New York, NY, USA: Association for Computing Machinery, Aug. 2019, pp. 1–10. [Online]. Available: https://doi.org/10.1145/3339252.3340509
17. N. Serketzis, V. Katos, C. Ilioudis, D. Baltatzis, and G. Pangalos, "Improving Forensic Triage Efficiency through Cyber Threat Intelligence," *Future Internet*, vol. 11, no. 7, p. 162, Jul. 2019, number: 7 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1999-5903/11/7/162
18. S. Samtani and Y. Chai, "Linking Exploits from the Dark Web to Known Vulnerabilities for Proactive Cyber Threat Intelligence: An Attention-based Deep Learning Deep Structured Semantic Model," *MIS Quarterly*, vol. 46, pp. 911–946, Jun. 2022.
19. R. Stillions, "The DML model," 2014. [Online]. Available: http://ryanstillions.blogspot.com/2014/04/the-dml-model_21.html
20. S. Bromander, A. Jøsang, and M. Eian, "Semantic Cyberthreat Modelling," p. 5, 2016.
21. D. Gunter, "Hunting with Rigor: Quantifying the Breadth, Depth and Threat Intelligence Coverage of a Threat Hunt in Industrial Control System Environments," p. 21, 2018.
22. MITRE, "MITRE ATT&CK Techniques Mapped to Data Sources," Tech. Rep., 2019. [Online]. Available: https://attack.mitre.org/docs/attack_roadmap_2019.pdf
23. E. Berrueta, "Open Repository for the Evaluation of Ransomware Detection Tools," Feb. 2020, publisher: IEEE Type: dataset. [Online]. Available: https://ieee-dataport.org/open-access/open-repository-evaluation-ransomware-detection-tools