IDENTIFYING ATTACKERS BY USING MACHINE LEARNING ON UNSTRUCTURED CYBER THREAT INTELLIGENCE

APRIL 2020

Cyrill Gössi

me@goescy.ch

ABSTRACT

Using cyber threat intelligence in automated processing systems requires structured data as input. Cyber threat intelligence on the Internet, however, often appears as unstructured cyber threat reports (CTRs) written by cyber security researchers. Such CTRs often, without knowing the actual adversary behind an attack, contain information regarding techniques and tools applied by the adversary during the execution of the attack. Due to their lack of structure, CTRs are difficult to work with and information must be manually extracted. In this work, we developed a machine learning based tool that helps identifying the adversary having executed the attack described by a CTR. The tool takes a CTR as input and extracts, among others, the techniques and tools described by the report to have been used during the attack. Based on this information, the tool then calculates the similarity to a set of previously learned adversary group profiles and outputs the calculated similarities in a sorted ranking. The higher the rank of an adversary group in the ranking, the more the information contained in the CTR resembles the profile previously learned about the adversary group. Evaluating the tool over 57 adversary groups with a total of 227 CTRs finds that the tool can output the correct adversary group with a probability of 63.54% on rank 1. Furthermore, as the tool allows to assign weights to different parts of a profile, we could confirm that assigning weights in accordance with the hierarchy proposed by the Pyramid of Pain outperforms assigning equal weights. This is seen as an interesting confirmation of the concept proposed by the Pyramid of Pain.

1. INTRODUCTION

Cyber threat intelligence (CTI) is commonly seen as evidence-based knowledge about cyber threats[1]. For an organization, it is vital to continuously obtain and process CTI in order to understand the current cyber threat landscape and to timely initiate appropriate measures of defense. CTI includes various types of digital forensic artifacts such as hashes of malware or IP addresses and domain names of botnet command and control servers. Such artifacts are often referred to as Indicators of Compromise (IoC)[2]. In recent years, standards like STIX[3] and TAXII[4] have emerged which enable the automated exchange of IoCs via special purpose feeds. These structured feeds are complemented by cyber threat reports (CTRs) which are a popular means sometimes chosen by cyber security researchers to provide more technical details of findings from investigating cyber security incidents. For example, in 2013, Mandiant published a CTR[5] with in-depth descriptions of findings from investigating the activities of a Chinese nation state adversary. Nation state adversaries are often referred to as Advanced Persistent Threats (APTs) and relating various cyber incidents to a single APT takes place by finding patterns among IoCs discovered. The previously mentioned IoCs, however, are easy to obscure by an APT. The hash of a certain malware, for example, can be changed by adding just a single no-op to the executable. According to the Pyramid of Pain[6], more difficult to change by an APT are the techniques and tools used during an attack and identifying these will thus give most insight into which APT might have executed the attack. Contrary to structured CTI feeds based on STIX, CTRs are often textual descriptions of research findings and, as such, constitute unstructured CTI. The intelligence contained in such CTRs must thus be extracted by a tedious and manual study of the reports. In this work, we developed a machine learning based tool which takes a CTR as input and tries to predict the APT possibly having executed the attack described by the input CTR. At the core, the tool works by building up profiles of APTs based on extracting techniques and tools used during attacks as well as countries and industry sectors targeted during the attacks. The tool then calculates the similarities between the profile contained in a new CTR and the previously learned APT profiles and outputs the similarities in a sorted ranking. Learning the tool on 57 APTs with a total of 227 CTRs, we found that the tool, upon input of a new CTR, outputs the correct APT with a probability of 63.54% on similarity-rank 1. Furthermore, we found that by assigning weights to the techniques, tools, targeted countries and targeted in-

dustry sectors in accordance with the hierarchy proposed by the Pyramid of Pain, the tool performs better than assigning equal weights. This is an interesting confirmation of the concept and methodology proposed by the Pyramid of Pain. The remainder of this paper is now structured as follows: Chapter 2 describes the methodology developed to profile APTs as well as the methodology used to calculate the similarity between APT profiles. Chapter 3 then describes the implementation of these methodologies in a command line tool. Chapter 4 presents the evaluation of the precision of the implementation, and chapter 5 concludes this work and outlines future research possibilities.

2. METHODOLOGY

In this section, we describe the methodology developed and the data used to implement an automated profiling of APTs as well as the methodology developed to calculate the similarity between a pair of APT profiles.

2.1. Profiling APTs

As formalized by the Pyramid of Pain[6], techniques and tools used during a cyber attack are two of the most telling indicators as to which APT might have executed an attack. As such, techniques and tools will be used to profile an APT. Furthermore, as APTs are nation state adversaries and as such are directed by the interest of a state, we also include the targeted countries as a third sub-profile to an APT's profile. After all, a Chinese APT, for example, likely doesn't target China. We also add the industry sector of the victim companies as a fourth sub-profile to an APT's profile, as some APTs[7] are know to almost exclusively target companies belonging to a single industry sector. The following sections now describe the approaches chosen on how to extract, from a CTR, each of the four APT sub-profiles mentioned above.

2.1.1. Techniques

Extracting techniques from a CTR was approached as a multiclass classification problem. For this work, we classified techniques according to the MITRE ATT&CK[8] framework, which currently defines a total of 266 techniques. The multiclass classification problem was reduced to a binary classification problem by training a dedicated logistic-regression classifier for each technique. Logistic-regression based classifiers were chosen for this work as they are simple in nature and are often used as baselines for classification problems. Investigating different types of classifiers was out of scope for this work. The data to train the logistic-regression classifiers was again taken from MITRE, which contains a list of example descriptions for each of the 266 techniques it defines. This training data, however, only contained enough training samples for a subset of 87 techniques, which reduced the number of classes down to 87. After training 87 classifiers on the training data, only 39 of the classifiers were found to have an average precision-recall score of more than 0.80. With this, the resulting classification took place across 39 techniques only. In order to find the set of techniques described by a CTR we now give each sentence of the CTR to all the 39 previously trained classifiers and include a technique in the result only if the classifier responsible for that technique predicts the technique from the sentence with a confidence more than 0.98.

2.1.2. Tools

Extracting tools from a CTR was approached as a simple search for keywords in the sentences making up the CTR. For this, we again used data provided by the MITRE framework and which currently lists names and alternative names of 307 different malwares. The set of tools mentioned by a CTR is then the set of malwares who either have their name or have one of their alternative names mentioned in the CTR.

2.1.3. Countries Targeted

Extracting countries targeted was again approached as a simple search for keywords in the sentences making up the CTR. Based on the assumption that the input CTR is a textual description of the findings resulting from investigating a cyber security incident without knowledge of the adversary that executed the attack, the set of countries targeted was simply the set of countries mentioned in the CTR. For this, we used a list of names and official alternative names of 140 countries as provided by Wikipedia[9]. The set of countries targeted mentioned by a CTR is then the set of countries who either have their name or have one of their official alternative names mentioned in the CTR.

2.1.4. Industry Sectors Targeted

Likewise, the extraction of targeted industry sectors was also approached as a simple search for keywords in the sentences making up the CTR. For this, we used the Global Industry Classification Standard (GICS)[10] which defines 11 industry sectors via 158 sub-industries. Based on these definitions, we compiled a list of keywords to be used for the search of industry sectors targeted. The set of targeted industry sectors as mentioned by a CTR is then the set of industry sectors for which a keyword can be found in the CTR.

2.2. Similarity of APT Profiles

As described above, we are ultimately looking for an automated way to learn adversary profiles from reading CTRs and to then leverage this knowledge to predict the APT that might have executed an attack as described by a new CTR. For this and given such a new CTR, we will first extract the profile hidden in the new CTR and then compare this profile to all APT profiles previously learned in a learning stage. We then output a sorted ranking of how similar the profile extracted from the new CTR is to all the known and previously learned APT profiles. The notion of similarity of two profiles, as described and defined below, is based on 2 ideas: 1) a pair of profiles is similar if the overlap of the two profiles is large 2) for two pairs of profiles with equal similarity, the one pair is more similar whose overlap relative to the size of their joint profile is larger.

Definition 1. *The set of techniques belonging to the profile of adversary i is denoted as π_i^T. Similarly, for adversary i, the set of tools, the set of targeted countries and the set of targeted industry sectors are denoted as π_i^O, π_i^C and π_i^S, respectively.*

Definition 2. *The profile of adversary i is denoted as π_i and is defined as follows: $\pi_i = (\pi_i^T, \pi_i^O, \pi_i^C, \pi_i^S)$.*

Definition 3. *The similarity between the profiles π_i and π_j of adversaries i and j, respectively, is denoted as $\sigma_{i,j}$ and is defined as follows:* $\sigma_{i,j} = \frac{\omega^T \cdot |\pi_i^T \cap \pi_j^T| + \omega^O \cdot |\pi_i^O \cap \pi_j^O| + \omega^C \cdot |\pi_i^C \cap \pi_j^C| + \omega^S \cdot |\pi_i^S \cap \pi_j^S|}{1 + |\pi_i^T \cup \pi_j^T| + |\pi_i^O \cup \pi_j^O| + |\pi_i^C \cup \pi_j^C| + |\pi_i^S \cup \pi_j^S|}$, *for ω^T, ω^O, ω^C and ω^S such that $\omega^T + \omega^O + \omega^C + \omega^S = 1$.*

With this notion of similarity, we can evaluate the power of different weightings of sub-profiles. For example, with $\omega^T = 1$ and with $\omega^O = \omega^C = \omega^S = 0$, we can evaluate the power of identifying adversaries based on solely looking at the techniques used during the attacks. With $\omega^T = \omega^O = \omega^C = \omega^S = 0.25$ we can evaluate the power of identifying adversaries by equally weighting the techniques and tools used during an attack as well as the targeted countries and the targeted industry sectors. And with a setting of $\omega^T = 0.4$, $\omega^O = 0.3$, $\omega^C = 0.2$ and $\omega^S = 0.1$, we can evaluate the power of identifying adversaries based on the Pyramid of Pain.

3. IMPLEMENTATION

The methodology outlined in section 2 was implemented as a Python based command line tool. The tool has a SQLite database as backend and features the four functionalities *learn*, *predict*, *evaluate* and *visualize*. The following sections will outline the purpose and working of each of these four functionalities.

3.1. Learn

This functionality learns APT profiles by reading CTRs that are known to describe incidents related to certain APTs. Each such CTR can either be a PDF or an HTML website. For such a CTR associated with APT i, the tool extracts from the CTR, according to the approach outlined in chapter 2.1, the profile π_i and saves π_i as the profile of APT i. In case the database already contains a profile for APT i, then this existing profile is simply extended with the new information contained in π_i.

3.2. Predict

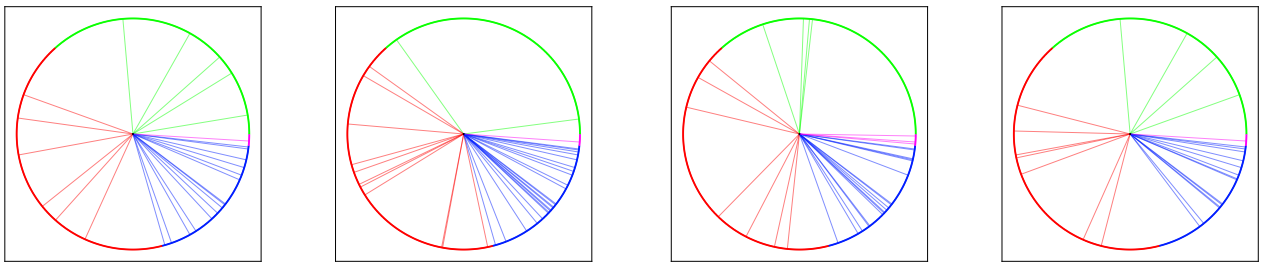
This functionality predicts an APT from a given CTR, which again can either be a PDF or an HTML website. According to the approach outlined in 2.1, the tool first extracts a profile π_{ctr} from such a CTR. Then the tool calculates the similarity $\sigma_{i,ctr}$ of π_{ctr} to all the previously learned APT profiles π_i. This calculation of similarity takes place according to the approach outlined in 2.2. The tool then finally outputs a sorted ranking of all the APTs according to how similar their previously learned profiles are to the profile extracted from the CTR. In this ranking, APT i appears before APT j if $\sigma_{i,ctr} \geq \sigma_{j,ctr}$.

3.3. Evaluate

This functionality evaluates the precision of the tool. This works by only considering APTs having at least two CTRs associated with them. Among such APTs, assume n to be the largest number of CTRs associated with any one of the APTs. The tool then executes n rounds that work as follow: In every round, the tool builds a completely new database and learns the profile of all the eligible APTs but leaves out exactly one CTR for each APT. Once the tool learned all the profiles, it predicts the APT from all the left out CTRs and checks how well the tool predicted the APT. The final precision of the tool is then the average of the precision of these predictions across all the n rounds. This is also the methodology according to which the actual evaluation of the tool outlined in section 4 took place.

3.4. Visualize

This is a purely illustrative functionality and creates visualizations of all the APT profiles currently existing in the database. As such, it provides a means for visual inspection of APT profiles. Examples of such a visualization are shown in figure 1 where the green edges form the technique sub-profile of an APT, the red edges form the tool sub-profile of an APT, the blue edges form the targeted countries sub-profile of an APT and the purple edges form the targeted industry sectors sub-profile of an APT.



(a) APT32 (b) BlackOasis (c) Deep Panda (d) FIN7

Fig. 1: Sample profiles after learning 96 APTs from a total of 263 CTRs provided and pre-classified by the MITRE framework. Green edges form the technique sub-profile, red edges form the tool sub-profile, blue edges form the targeted countries sub-profile, purple edges form the targeted industry sectors sub-profile.

4. EVALUATION

The evaluation took place according to the methodology outlined in chapter 3.3. By using data provided by the MITRE framework, the evaluation took place over 57 APTs and 227 CTRs where for each APT at least 2 CTRs were available and the maximum number of CTRs available for an APT was 17. Thus, the following evaluation is an evaluation of the precision averaged over 17 runs of learning all 57 APTs over all CTRs where in each round one CTR was ignored for each APT. Once in every round the learning process was finished, all the CTRs ignored during the learning phase were used to test the precision of the tool when predicting the APT from the CTR. We evaluated the power of each sub-profile to by itself predict an APT from a CTR (for example, to predict an APT solely from the technique sub-profile we set $\omega^T = 1$ and $\omega^O = \omega^C = \omega^S = 0$), evaluated the power of equally-weighted sub-profiles to predict an APT from a CTR (that is $\omega^T = \omega^O = \omega^C = \omega^S = 0.25$) and evaluated the power of sub-profiles weighted such that the weighting reflects the methodology introduced by the Pyramid of Pain (that is $\omega^T = 0.4$, $\omega^O = 0.3$, $\omega^C = 0.2$ and $\omega^S = 0.1$). In total, this amounts to an evaluation of 6 different ways of weighting sub-profiles, and the results of these evaluations are shown in table 1. Table 1 contains one row for each of the 6 ways of weighting the sub-profiles with the results of evaluating the precision of the tool with this particular sub-profile weighting scheme. For each sub-profile weighting scheme, table 1 then shows the probabilities that, for a given new CTR, the APT calculated as the one APT with a profile most similar to the profile contained in the CTR is indeed the correct APT (column *Rank 1*), that the APT calculated as the one APT with a profile that's second-most similar to the profile contained in the CTR is indeed the correct APT (column *Rank 2*), and that the APT calculated as the one APT with a profile that's third-similar to the profile contained in the CTR is indeed the correct APT (column *Rank 3*). Each cell of the first 3 columns then lists a triple of values indicating the following: the first value indicates the probability that the tool returns the correct APT on this similarity-rank, the second value indicates the cumulative probability that the tool returns the correct APT on a rank up to this similarity-rank, and the third value indicates how much this probability outperforms a 57-sided dice. For example, looking at the technique-only weighting scheme, the correct APT, given a new CTR, is returned on similarity-rank 1 with a probability of 54.38%, which is a probability that outperforms a 57-sided dice 30.99 times. The cumulative probability up to similarity-rank 3 is 57.60%, which outperforms a 57-sided dice 10.94 times. The last column (column R_{P50}) in table 1 lists the similarity-rank for which the probability is at least 50% that the correct APT, given a new CTR, is returned on a rank up to this similarity-rank. For example, for the technique-only weightings scheme this is already the case as of rank 1, but for the targeted-countries only weighting scheme, this is only the case as of rank 13, with a cumulative probability of 51.98% up to this rank. This outperforms a 57-sided dice by a mere 2.28 times. The best performing sub-profile is the targeted industry sectors sub-profile. This sub-profile by itself can identify the correct APT with a probability of 63.54%, which outperforms a 57-sided dice 36.22 times. In the long run, this sub-profile is however outperformed by the tools sub-profile, for which the probability that the correct APT is on one of the first three ranks is 73.44%. The worst performing sub-profile is the sub-profile for the targeted countries. With this sub-profile, the correct APT can only be identified with a mere 16.67% and for which only as of rank 13 it holds that the correct APT is among them with a probability at least 50%. For the weighting schemes mixing the sub-profiles together, we see that giving equal weights to all the sub-profiles leads to the correct APT being identified with a probability of

	Rank 1	Rank 2	Rank 3	R_{P50}
Techniques Only	54.38% / 54.38% / 30.99	1.77% / 56.15% / 16.00	1.46% / 57.60% / 10.94	1 / 54.38% / 30.99
Tools Only	54.38% / 54.38% / 30.99	12.81% / 67.19% / 19.15	6.25% / 73.44% / 13.95	1 / 54.38% / 30.99
Countries Only	16.67% / 16.67% / 9.50	7.50% / 24.17% / 6.89	4.79% / 28.96% / 5.50	13 / 51.98% / 2.28
Industry Sectors Only	63.54% / 63.54% / 36.22	2.40% / 65.94% / 18.79	0.62% / 66.56% / 12.65	1 / 63.54% / 36.22
Equal Weighting	24.69% / 24.69% / 14.07	5.83% / 30.52% / 8.70	5.42% / 35.94% / 6.83	9 / 52.19% / 3.31
Pyramid of Pain	27.08% / 27.08% / 15.44	5.83% / 32.92% / 9.38	4.58% / 37.50% / 7.12	6 / 50.73% / 4.82

Table 1: Results of evaluating different sub-profile weightings over 57 APTs and over 227 CTRs

24.69% and leads the correct APT to be among the first 9 ranks with a probability of at least 50%. For the weighting scheme reflecting the Pyramid of Pain, we see that this weighting scheme can identify the correct APT with a probability of 27.08% and that the correct APT is among the first 6 ranks with a probability of at least 50%. This outperforms the equal-weights scheme and confirms the concept and methodology established by the Pyramid of Pain.

5. CONCLUSION AND OUTLOOK

From the evaluation and table 1 we could see that the current implementation can, among a set of 57 APTs and by only considering the targeted industry sectors sub-profile of APTs, identify the correct APT, given a new CTR, with a large probability of 63.54%. The techniques-only and the tools-only sub-profiles closely follow with probabilities of 54.38%. The countries-only sub-profile has a bad performance of 16.67%. This may be explained by the implementation currently simply searching for the occurrence of names of countries in the CTR. These names come from a fixed list. Whatever country it finds is stored as part of the targeted countries sub-profile of the APT. However, some CTRs are in-depth analysis of an APT and as such likely also mention the home country of an APT. As long as the implementation doesn't filter out known home-countries from the targeted countries stored in the profile, the implementation won't really live up to the purpose of the targeted countries sub-profile. To solve this, a knowledge base needs to be created where some prior knowledge including the home countries of APTs would need to be incorporated. Moving on from this, the evaluation showed that giving weights to the sub-profiles according to the Pyramid of Pain leads to a better performance than giving equal weights to the sub-profiles. This is seen as a nice confirmation of the concept and methodology established by the Pyramid of Pain. All in all, the results shown in section 4 are seen as supporting the methodology outlined in this work. The problem of automating the extraction of threat intelligence from unstructured cyber threat intelligence and, for example, to give cyber security researchers an automated means for identifying adversaries based on textually summarized findings of investigations into cyber security incidents (for example in the shape of a CTR), is of sufficient importance to pursue further research in this area and several possible research directions are now outlined in the following sections.

5.1. Extracting Techniques from CTRs

According to the Pyramid of Pain, techniques applied during an attack are some of the most telling indicators as to which adversary might have executed the attack. Extracting techniques was approached by machine learning instead of a simple search for keywords, which seems to be a necessity when unstructured CTRs should be classified into techniques, at least when the classification should take place according to the 266 techniques defined by the MITRE framework. This is a large number of classes with often only subtle differences. A classification problem of this size requires extra care with respect to compiling training-data for learning classifiers. Such training-data preferably contains hundreds of labeled examples for each class and needs to be well balanced across all classes. Obtaining such a large and well-balance set of training-data was not yet feasible and the only training-data used was the data provided by the MITRE framework. Moreover, in order to have as little parameters to explain as possible, the classifiers chosen were out-of-the box logistic-regression classifiers and an in-depth measurement of the performance of the resulting classifiers was out of scope so far. As further work in the area of extracting techniques from CTRs, we see it as a necessity to thoroughly evaluate the performance of the classifiers for the techniques. Based on this, we could then go on to investigate whether logistic-regression classifiers are appropriate for this problem domain or whether different types of classifiers are more suited. Lastly, the tool feeds the classifiers sentence-by-sentence with each sentence previously being freed from stop-words and then stemmed. It remains to be investigated if this approach is sensible in the context of CTI or if more suited approaches exist.

5.2. Additional Sub-Profiles

As outlined in section 2.1, the current methodology is based on extracting four sub-profiles. Looking at the Pyramid of Pain, what is currently missing is a sub-profile for the tactics

applied by an adversary during the execution of an attack. The MITRE framework currently defines 12 tactics, but extracting these was out of scope for this work. Once tactics would be extracted, the techniques extracted could be associated with the tactics under which the technique was applied. This would likely allow for a more clear distinction among adversary groups.

5.3. Different Notions of Profile-Similarity

No research has yet been conducted into whether, for example, graph-theory has established notions of graph-similarity. The notion of profile-similarity introduced in 2.2 seems to be a good starting point, but it would be interesting to investigate the predictive power of alternative notions.

6. REFERENCES

[1] Rob McMillan. Open Threat Intelligence. https://www.gartner.com/en/documents/2487216/definition-threat-intelligence, 2013

[2] L. Obrst, P. Chase, R.Markeloff. Developing an Ontology of the Cyber Security Domain. In STIDS, p. 49-56, 2012

[3] S. Barnum. Standardizing Cyber Threat Intelligence Information with the Structured Threat Information eXpression (STIX). MITRE Corporation, 2012

[4] J. Connolly, M. Davidson, M. Richard, C. Skorupka. The Trusted Automated eXchange of Indicator Information (TAXII). MITRE Corporation, 2012

[5] Mandiant. APT 1: Exposing One of China's Cyber Espionage Units, 2013

[6] D. Bianco. The Pyramid of Pain. Enterprise Detection & Response, 2013

[7] FireEye. Follow the Money: Dissecting the Operations of the Cyber Crime Group FIN6, 2016

[8] MITRE ATT&CK, MITRE Corporation. Online: https://attack.mitre.org/, Accessed 14.02.2020

[9] List of alternative country names - Wikipedia, The Free Encyclopedia. Online: https://en.wikipedia.org/wiki/List_of_alternative_country_names, Accessed: 14.02.2020

[10] Global Industry Classification Standard, Online: https://www.spglobal.com/marketintelligence/en/documents/112727-gics-mapbook_2018_v3_letter_digitalspreads.pdf, Accessed 14.02.2020