HuMa: A Multi-layer Framework for Threat Analysis in a Heterogeneous Log Environment

Julio Navarro, Véronique Legrand, Sofiane Lagraa, Jérôme François, Abdelkader Lahmadi, Giulia de Santis, Olivier Festor, Nadira Lammari, Fayçal Hamdi, Aline Deruyver, et al.

HuMa: A multi-layer framework for threat analysis in a heterogeneous log environment

Julio Navarro[1 2], Véronique Legrand[4 5], Sofiane Lagraa[6], Jérôme François[6], Abdelkader Lahmadi[6], Giulia De Santis[6], Olivier Festor[6], Nadira Lammari[4], Fayçal Hamdi[4], Aline Deruyver[1], Quentin Goux[5], Morgan Allard[5], Pierre Parrend[1 2 3]

[1] ICube, Université de Strasbourg, France
{navarrolara,pierre.parrend,aline.deruyver}@unistra.fr
[2] Complex System Digital Campus (UNESCO Unitwin)
http://unitwin-cs.org/
[3] ECAM Strasbourg-Europe, Schiltigheim, France
[4] CEDRIC, Conservation National des Arts en Métiers (CNAM), Paris, France
{veronique.legrand,ilham-nadira.lammari,faycal.hamdi}@cnam.fr
[5] Intrinsec Sécurité, Nanterre, France
{veronique.legrand,quentin.goux,morgan.allard}@intrinsec.com
[6] LORIA, INRIA Nancy Grand-Est, France
{sofiane.lagraa,jerome.francois,abdelkader.lahmadi,giulia.de-santis,
olivier.festor}@inria.fr

Keywords: Security knowledge, cognitive computing, cybersecurity, log analysis

Abstract

The advent of massive and highly heterogeneous information systems poses major challenges to professionals responsible for IT security. The huge amount of monitoring data currently being generated means that no human being or group of human beings can cope with their analysis. Furthermore, fully automated tools still lack the ability to track the associated events in a fine-grained and reliable way. Here, we propose the HuMa framework for detailed and reliable analysis of large amounts of data for security purposes. HuMa uses a multi-analysis approach to study complex security events in a large set of logs. It is organized around three layers: the event layer, the context and attack pattern layer, and the assessment layer. We describe the framework components and the set of complementary algorithms for security assessment. We also provide an evaluation of the contribution of the context and attack pattern layer to security investigation.

This work was partially supported by the French Banque Publique d'Investissement (BPI) under program FUI-AAP-19 in the frame of the HuMa project.

1 Introduction

Security analysis is a rigorous process encompassing several phases described in ISO 27043. It is conducted by security analysts having expertise in the identification and understanding of indicators of potential threats in logs (computer system traces). They achieve this using business rules mainly based on their past experience and the technical documentation of devices in the network. The continuous growth of the volume of logs to be analyzed, as well as their heterogeneity, make the task of the analyst increasingly difficult and error-prone even with current support tools. Since no automated method exists that integrates human level complex reasoning, the support tools generate confusing results and a multitude of false positives and false negatives. The situation is further complicated by the advent of more complex and hard to find attacks, such as Advanced Persistent Threats (APTs). As a result, new tools capable of addressing the challenge of identifying threats in massive log repositories are needed. A broad distinction can be made between simple attacks, which can be analyzed from individual events, and the more complex or targeted ones, including the APTs, which affect more than one asset and require an in-depth investigation. The more complex attacks can be thought of as being composed of different steps that are spatially and temporally spanned.

Taken individually, the multiple steps composing an APT are not necessarily illegal. Furthermore, since they are spatially and temporally spanned, they may seem to be unrelated. Nevertheless, as a whole, they constitute a single powerful attack. Therefore, in order to detect and predict such threats, it is necessary to collect, analyze and correlate various sources of data and to create summarized views that are exploitable by security analysts. Most systems save the actions related to them in lines of text called *logs*. Then, a security investigation is usually based on manual analysis of these logs [4]. Applications that collect logs are known as SIEM (Security of Information and Event Management) in industry. They include correlation methods for the automatic search of attack evidence. However, in the current context of *big data* and the huge amount of logs generated in a network makes the analysis difficult, if not impossible.

To address this challenge, we propose HuMa, a multi-layer framework for the analysis of complex security threats. It brings together the individual contributions made by the authors under the label of the HuMa project. The framework is composed of three layers: the event layer, responsible for the representation of individual traces of malicious activities; the context and attack pattern layer, responsible for gathering information about technical requirements of the attacks; and the assessment layer, responsible for extracting attack information from massive logs. The event layer is typically based on monolithic rules. The attack pattern layer, is constructed from databases, such as the CVE and CAPEC repositories. The context layer includes information about the system that we aim to defend. The assessment layer represents complex attacks as attack graphs, and searches for matches between the graphs and the actual traffic in the Information System under supervision. Apart from the architecture of the framework, another contribution of this work is a set of complementary approaches for the

assessment layer. Two kinds of dependencies between events are considered: temporal and spatial dependencies. The analysis is performed either through a root-cause approach, or through graph matching using dynamic weighted graphs. This latter approach is implemented in Morwilog, which is an application of the Ant-Colony algorithm to security investigation in logs. An evaluation of the framework is performed by assessing the contribution of the context and attack pattern layer for such investigation.

This work is organized as follows. Section 2 presents the state of the art on Advanced Persistent Threat analysis. Section 3 defines the multi-layer investigation framework. Section 4 focuses on the assessment layer and introduces key algorithms to address the investigation challenge. Section 5 discusses the evaluation of the framework and provides further insight into its application scope. Section 6 concludes this work.

2 State of the art

In this section we present a brief summary of the state of the art in the detection and analysis of Advanced Persistent Threats.

2.1 Modelling and analysis of APTs

APTs [7, 30] are one of the most serious information threats that enterprises and government agencies are faced with today. Examples include Stuxnet [8] and Carbanak [7]. Although individual APTs vary considerably, they are customized to the target system, and they all share the same 6 phases: reconnaissance, delivery, exploitation, operation, data collection and exfiltration [7].

Some work has been done to model APTs [15, 5, 10]. For instance, APTs can be described using low-level details, such as monitoring events, vulnerability descriptions and exploit information. Others use a top-down approach with high-level abstractions, based on attack trees or graphs. The most widely used model for APTs is the attack tree [29], where leaves or branches are linked by AND or OR gates. An improved attack tree is described in [5], where an O-AND gate and some extra attributes are added. Similar work is described in [3], where a SEQ gate and probability distributions are added to the leaves of the tree. However, attack trees are not the most suitable models for such threats [12], because they lack technical details, and providing them would make the trees too complex and difficult to read. A new conceptual attack model, the *attack pyramid*, is proposed in [14], which shows that an attack path may go across different environments of the organization. Cui et al. [10] focus on identification of attacks at early stages and prediction of their evolution using Hidden Markov Models (HMMs). Abraham and Nair [1] predict changes over time by capturing interrelations of vulnerabilities using attack graphs. They propose a three-layer architecture: layer 1 contains the attack graph model, whose vulnerabilities are

[7] https://securelist.com/files/2015/02/Carbanak_APT_eng.pdf

quantified in layer 2. Layer 3 describes attacks by applying stochastic processes over the attack graph.

2.2 Automatic analysis of APT scenarios

Although HuMa still considers the human analyst as a key player in the identification of attack scenarios, automatic methods for finding links between events are used to facilitate the expert analysis. The methods for identifying APT scenarios found in the literature work to a large extent on alerts generated by an Intrusion Detection System (IDS). An example method that uses this approach is AEC, or Active Event Correlation [6], applied on top of a Bro IDS. This allows it to interact with network traffic. Marchetti et al. [22] use alert graphs in a pseudo-Bayesian algorithm with the previous alert history as a reference. In other systems, such as RIAC [34], attacks are deduced from manually described prerequisites and consequences of individual alerts. However, in [32] the knowledge base of alerts is already contained in the system but the casual relationships are automatically extracted. Unsupervised methods have also been developed, such as the one in [31], based on the aggregation of similar alerts from the same time period in possible attack scenarios. Other approaches exist that do not work only with IDS alerts. Mathew et al. [23], for example, propose an anomaly detection method on a heterogeneous dataset using Principal Component Analysis (PCA). Another example is described in [13], where hypothesis and rules are deduced from a dataset of logs without attacks.

2.3 Correlation between vulnerabilities and attack patterns

Discovered vulnerabilities in information systems are in general publicly disclosed by means of the CVE (Common Vulnerabilities and Exposures) format, an open industrial standard widely adopted by many organizations. Each CVE document has an identifier and mainly provides a textual description of a security vulnerability or exposure. The documents are publicly available through multiple databases[8]. In addition to CVE, CAPEC (Common Attack Pattern Enumeration and Classification)[9] patterns are distributed by MITRE in XML format. They also provide a textual description of an attack, its prerequisites, its steps, severity and the attack methods used.

Several papers have studied the known vulnerabilities in order to predict their exploitation. In [33], the authors apply several machine learning algorithms on the National Vulnerability Database (NVD), with the goal of predicting undiscovered vulnerabilities. An interesting work regarding how vulnerabilities are exploited by attackers is described in [2]. The authors found that only a small subset of vulnerabilities in the NVD and Exploit-DB are found in exploit kits in the wild. However, little work has addressed the correlation between security

[8] https://nvd.nist.gov, http://cve.mitre.org, http://www.cvedetails.com
[9] http://capec.mitre.org

alerts, CVE and CAPEC documents. In [28], the authors use data mining techniques to map CAPEC patterns to security logs. The obtained representations are then matched using a K-nearest-neighbour algorithm to obtain the closest events to an attack pattern. In [16], CAPEC and CVE patterns are used to generate attack graphs and then match security events against them to identify running attack scenarios.

3 The security multi-layer framework

In this section, we discuss how HuMa uses the idea of classical SIEMs to create a cognitive platform where the human being is a key player. Thanks to novel representations of logs and attacks, HuMa combines the power of several analysis algorithms, whose results are continuously improved by feedback from the human expert.

3.1 Handling security knowledge

One of the key challenges of HuMa is to capitalize on the complex reasoning of security analysts. It aims to help experts in their decision making by providing them with a guiding tool. The tool should allow them to react on the fly to threats even in an environment where the logs are massive and heterogeneous and where malicious tactics are continuously evolving. The analysis is made more efficient as the focus is placed on the most noteworthy events.

To incorporate guidance, the proposed tool must integrate all the useful knowledge on the security analysis business. This knowledge can be represented using a domain ontology. Given the heterogeneity of the logs, in order to facilitate the correlation between them, we need to define a unique vocabulary of concepts to represent them. The domain ontology must contain this vocabulary and all other knowledge related to the security analysis.

3.2 Representation of logs

We have developed a new representation of logs suitable for both machines and humans. The HuMa analysis process is based on concepts for representing logs, that are able to extract the relevant information about a security incident. Logs are big data streams, so they inherit the 4 properties of big data, called 'the 4 V's': volume, velocity, veracity and variety. The concepts aim to reduce one dimension of the heterogeneity present in logs, i.e. the variability in expressing similar pieces of information. For instance, a 'login failure' event can be represented in a totally different way by a SQL server or by a router, although they have similar meanings and thus share some concepts. An extractor module automatically extracts concepts from the original logs (raw logs), guided by a knowledge representation model and without human intervention.

In addition, HuMa provides vulnerability analysis based on the concepts that have been developed by the MITRE in the CVE documents and CVSS scores.

Each CVE identifier includes a unique identifier number, references about its vulnerability and a description of the vulnerability or exposure. This description contains information about the weakness of the affected asset and the outcome and consequences of exploiting it. The CVSS associated with each CVE takes into account the information from the description. The CVE description and the assigned CVSS include knowledge provided by the security community that maintains the database, so each CVE identifier benefits from the knowledge of security experts on each of the vulnerabilities.

Thanks to machine learning tools, expert knowledge can contribute to reduce error in the process of concept extraction. The concept extraction module automatically generates pertinent questions that are then presented to the analyst in order to improve its internal knowledge database.

Concepts also allow to unify the way logs express information, beyond preprocessing relevant data. This guarantees that the analysis processes have a direct access to the information, regardless of the device or service implementation that generates it. Moreover, our use of concepts leads to a direct reduction in the complexity of the logs, compared to the information extracted by Splunk[10] from raw logs. For instance, in a test involving a log dataset of 600000 entries from 6 different devices, the number of attributes is reduced from 62 to 16 and the number of values is reduced from 518 to 72. The concepts enrich the raw log such that the data added is compatible with existent technologies, while preserving the original log. The dataset composed of the raw log and its enriched data is processed by the rest of the HuMa platform.

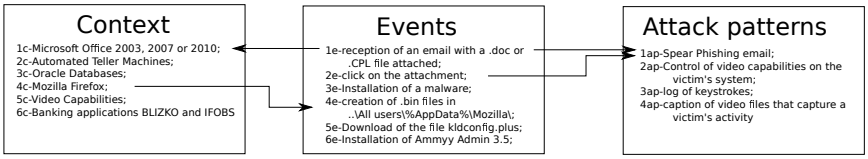


Fig. 1: Modelling elements and their relations for the Carbanak attack

3.3 Representation of attacks

An important question in HuMa is to find the best way to represent event scenarios that may pose a threat to the system. The representation should be common to all the methods integrated in the framework. To introduce our approach, we use the Carbanak cyberattack as an example. It is possible to describe this attack through the context in which it takes place, the collected events and known attack patterns (Figure 1). For Carbanak to operate, the presence of Microsoft Office 2003, 2007 or 2010 (context) is required since the .doc file received via e-mail (event) has to be opened (event). The malware installed by this action creates a .bin file (event) in a folder created by Mozilla Firefox (context), which

[10] https://www.splunk.com

therefore needs to be installed as well. The reception of the .doc file via e-mail and the following double click on it are also related to the attack pattern of spear phishing (attack pattern). HuMa incorporates a multi-layer approach that identifies the links between the elements characterizing an APT like Carbanak: known attack models, detected events and knowledge of the system. Most of the work mentioned in Section 2 focuses on modelling already known attacks [5, 15, 30], whereas we focus on predictive modelling of an APT. Our work relies on a multi-layer modelling technique whose schema is shown in Figure 2. The first layer consists of normal actions and alarms generated by security systems. These events are correlated and matched to the context (i.e. the configuration of the system), and to already known attack patterns. Context and well-known attack patterns form the second layer. The assessment layer contains the model for a possible attack scenario, created from the link between elements of the other layers. The linking process considers elements as the time-to-live of each step, the time in which each step takes place, the probability of success, shared context or users in common.

As shown in Figure 2, each level is connected to the one above and below it. The collected events contribute to the selection of known attack patterns, and are then matched to them. Vice versa, the selected set of attack patterns helps to direct the search for events in the system. For example, in the Carbanak attack, if the spear phishing attack pattern is included in the selection, a search for the "reception of e-mails with attachments" event is performed. Similarly, the events help to define the context, which in turn guides the search for events. For instance, if the received attachment is a .doc file, HuMa checks whether Microsoft Office is installed in the system. Similar connections also exist between the selected attack patterns and the system context, and the assessment model. These layers work together to select attack models and to identify which part of the context needs to be investigated. For this task, we rely on the CAPEC and CVE databases and the help of security analysts (see 4.2).

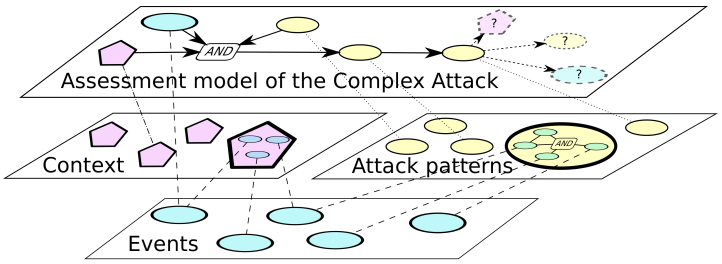


Fig. 2: Proposed approach for predictive APT modelling

4 Analysis engines in HuMa

The analysis of APTs in HuMa involves several bricks that work together to return a combined result in the same dashboard.

4.1 Dependencies

Temporal dependencies An attack usually involves periodic changes in behaviour over time. We can find clear examples in denial of service attacks or port scans. Although the time lapse between two requests may vary, there is a certain periodicity that can be identified by studying the shift in the process state of the communication with a target machine. The difficulty of this task is related to the fact that changes are not necessarily explicit and can be mixed with other types of events. HuMa includes a method to find temporal dependencies between logs ordered in time. It is based on data mining techniques and the representation of logs by the high-level semantic concepts introduced in 3.2. The goal is to discover temporal dependencies via frequent and periodic patterns of logs ordered in time. The method automatically returns unexpected temporal changes, as well as the context in which these changes take place. Figure 3 presents the results of mining with fixed slice-window periods. Patterns are represented on the Y-axis and time windows on the X-axis. The attributes in the patterns refer to those in Snort[11]. Each cell represents a pattern frequency. Darker cells represent frequent patterns and lighter cells, infrequent ones. This approach reduces the number of patterns to be analyzed and guarantees the temporal consistency of conceptualized logs. More details can be found in [17].

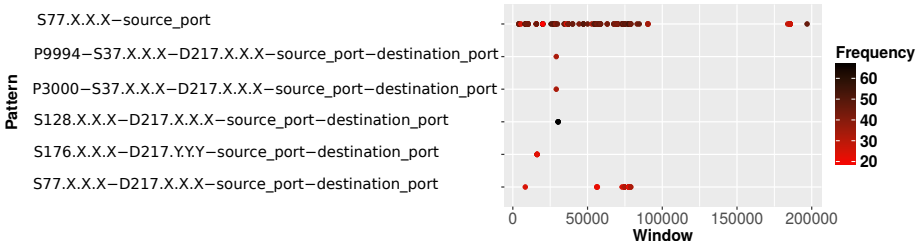


Fig. 3: Overview of frequent and periodic patterns.

Spatial dependencies In HuMa, we also exploit spatial dependence discovery techniques to find correlations or similarities among events in order to build clusters. Indeed, rather than analyzing a single event, a human expert can benefit from clusters in order to understand a group behaviour and speed up the analysis. Relevant methods can be found in the area of data mining with clustering approaches. There are numerous options available, but a major bottleneck of many of them is their computational complexity as they require pair-wise comparisons between initial data points, *i.e.* events or concepts in our case. We thus propose to use TDA (Topological Data Analysis) [25]. which reduces the high dimensionality of the data through a simpler representation that can be searched for invariants. Such invariants can then be considered as significant patterns of

[11] https://www.snort.org/

the underlying data. To achieve our goal, *i.e.*, the Mapper [26] algorithm from TDA is integrated in HuMA. Rapidly, the algorithm works as follows:

1. The original highly dimensional space is decomposed as overlapped hypercubes.
2. In each hypercube, a clustering algorithm is applied.
3. A graph over all data points is created, where a vertex represents a cluster in a hypercube. An edge between two vertices exists if and only if the two underlying clusters share at least one original data point, which is possible due to the overlapping of hypercubes.

There are therefore three major parameters: (1) the resolution, representing the number of hypercubes (the smaller, the greater the amount of hypercubes); (2) the overlap between hypercubes, and (3) the clustering algorithm. In our case, DBSCAN is used. It is a density-based clustering algorithm that does not require an a priori estimate of the number of clusters. However, it induces two other parameters to be set, namely the minimum number of neighbours at a given maximal distance for each clustered point. This technique has been successfully applied to Darknet analysis in our prior work [9] and it has been fine-tuned with respect to the HuMa cognitive framework for the analysis of concepts extracted from logs as a first step.

HMMs applied to logs of scanning activities During the reconnaissance phase of APTs [7], powerful scanning tools are used by attackers. The availability of models describing various aspects of these scanning activities can help security experts to predict whether an attack is underway. In HuMa, we model intensity, spatial and temporal movements of scanning techniques using mixture distribution models and HMMs, based on logs extracted from a /20 darknet. A combination of mixture distribution models and HMMs are used since logs may be divided into unobserved clusters. First, mixture distribution models provide the probability of the clusters. Second, the corresponding HMM, whose states are the distributions of the mixture, provides the transition probabilities between clusters. The obtained models are presented in the dashboard of HuMa, so that the security analyst can determine whether there is a scanning activity and prepare for a security attack. So far, this engine has been applied to logs in the reconnaissance phase of an APT, but we are currently working on its application in all phases. A more detailed description of the method has been published in [11].

Dependency analysis The conceptualization of the heterogeneous logs in HuMa implies that abnormal behaviour mining techniques can be used efficiently. These approaches aim to find the dependencies of abnormal behaviour. Given a set of conceptualized log sequences, the problem is both to identify the abnormal behaviours and the set of dependencies able to guide the analyst. Figure 4 shows an abnormal behaviour graph extracted from a set of logs. The graph represents the activity of an IP address 81.89.X.X targeting a web server. The

red and blue boxes represent the start and end of the graphs, respectively. This graph helps the analyst to understand an abnormal behaviour which may be a potential threat and localize it within a time window. Indeed, this graph was confirmed as a representation of a real attack performed from the IP address.

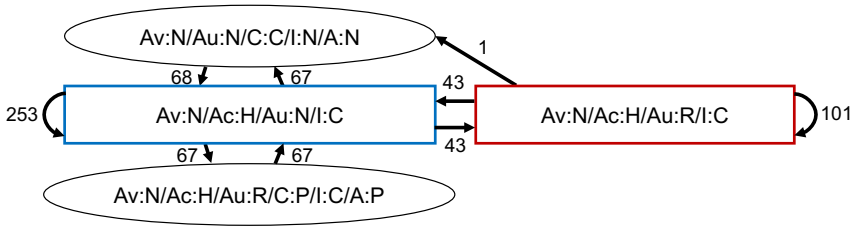
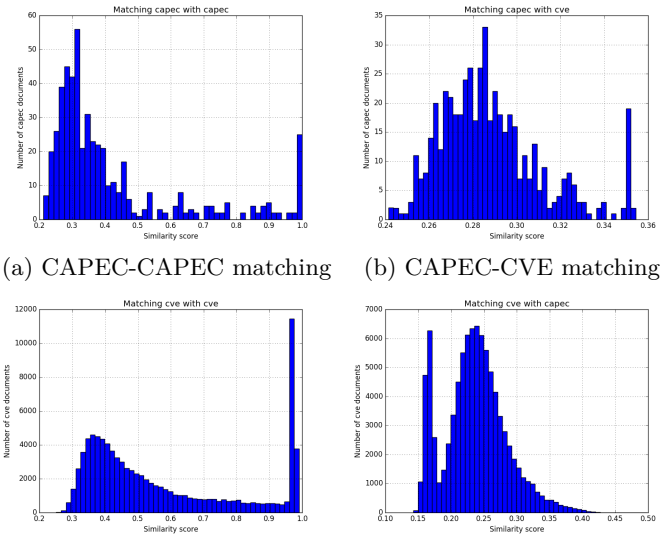


Fig. 4: Abnormal behaviour graph.

4.2 Root cause analysis

A root cause analysis (RCA) is a specific description of an attacker's procedure that identifies all the requirements and causes that led to an incident [19, 20]. A RCA is particularly useful for producing an analysis and incident report. The description of a RCA is different from that of dependencies as it provides an explanation about the incident and details about what happened. The description includes the conditions required for executing the described actions as states of the system. They may also include a description of vulnerabilities. The MITRE database is an extremely rich public database, containing more than 80000 CVE identifiers. Each CVE identifier contains specific details of the affected system. The exploitation of a vulnerability is generally crucial in the execution of an APT. For instance, one of the steps of Carbanak is the exploitation of CVE-2015-5262, related to an incorrect configuration of the 'keepalive'.

Matching vulnerabilities and attack patterns Our approach for matching security events to CVE and CAPEC documents is close to that of [28], since we share the same goal. However, we apply a recent machine learning technique, *doc2vec* [18], in order to learn from the textual descriptions of CVE and CAPEC documents. We used the cosine similarity metric to mutually match the embedding vectors obtained from the text, and evaluate their correlation. We applied this technique to the available set of 510 CAPEC patterns and a set of 91405 CVE documents available from the MITRE web site. In both cases, we compute the respective embedding vectors using the *gensim* [27] *doc2vec* python library. Then, for each CVE and CAPEC document, we compute the 10 most similar documents using the learned vectors, that results on 10 similarity values ranging between 0 and 1.



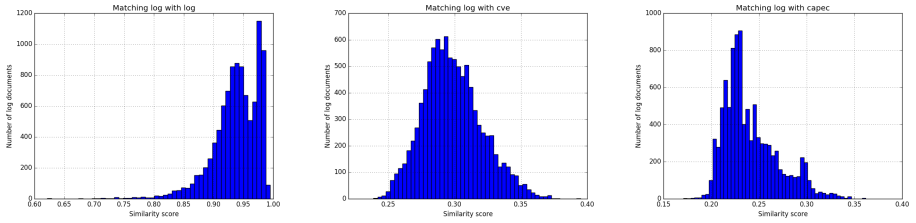
(a) CAPEC-CAPEC matching (b) CAPEC-CVE matching

(c) CVE-CVE matching (d) CVE-CAPEC matching

Fig. 5: Distribution of best matching scores using *doc2vec* algorithm.

Evaluation In this section, we present the experiments performed on the module matching vulnerabilities and attack patterns, which were not published previously. In a first step, we analyzed the distributions of the similarity scores obtained by matching CAPEC and CVE documents. Figure 5 shows the histograms of these scores for the most similar document only. As shown in Figure 5a, when matching CAPEC to CAPEC documents, we observe that for a similarity value about 0.3 we obtain the most matched documents, around 56, or 10% of the documents. However, when matching CVE to CVE documents, as shown in Figure 5c, we found that 11458 documents are similar with a score up to 1, representing 12.5% of the analyzed documents. When matching CAPEC with CVE documents, we found that 33 CAPEC documents match CVE documents with a score around 0.28, representing 6.4% of the CAPEC documents, as shown in Figure 5b. When matching CVE with CAPEC documents, as shown in Figure 5d, we found 6895 CVE documents match CAPECs with a score around 0.25, representing 7% of the analysed CVE documents. We thus observe better matches between CVE documents with a score up to 1 for 12.5% of the analyzed documents. For the other matching scenarios, the results are close with similarity scores between 0.25 and 0.3.

In a second step, we calculated the similarity scores for a sample of 10000 conceptualized logs coming from the test environment of a security company and compared them with CAPEC and CVE documents. The results are shown in Figure 6. As shown in 6a, we observe that log-log matches obtain high similarity scores. The scores are mostly between 0.8 and 1, which means that multiple logs



(a) log-log matching (b) log-CVE matching (c) log-CAPEC matching

Fig. 6: Distribution of similarity scores using *doc2vec* algorithm.

are similar and could be easily aggregated before being presented to a human analyst. Matching the logs with CVE documents, as shown in Figure 6b, produces similarity scores normally distributed around 0.3, with 613 logs achieving this score. When matching logs with CAPEC documents, as shown in Figure 6c, we observe a maximum number of 907 log documents that match CAPECs with a similarity score around 0.22. Using the *doc2vec* technique, we are able to match logs with their respective most similar CAPEC and CVE documents, which means that we can associate them to vulnerabilities and attack patterns. These associations help security analysts to better understand what is happening in the system.

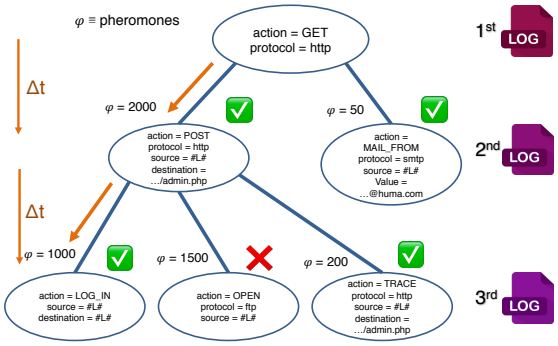


Fig. 7: Tree of log sequences

4.3 Searching for paths in event graphs with Morwilog

Once a database of event graphs is generated, we need to know which paths may be of special interest to the analyst. Paths may also exist that are erroneous or that no longer pose a threat. To address this in the context of HuMa, we developed an algorithm called Morwilog based on Ant Colony Optimization (ACO), a metaheuristic to solve discrete optimization problems. ACO is inspired by the

behaviour of a colony of foraging ants when they leave the anthill in the search of food. In this process, ants depose pheromones so that other ants can follow the traces to the food source. This results in the formation of well-ordered trails from the anthill to the food source. After some time, almost every ant follows the shortest path, where pheromones are deposited at a higher rate. In ACO, a set of artificial ants is generated to find the shortest path in a graph.

Artificial ants in Morwilog are called *morwis*, and their generation is associated with the arrival of a log. When a suspicious log arrives at Morwilog, a *morwi* is generated and it proceeds through the event graph whose root node corresponds to this log. Event graphs are deployed here as trees, with deeper levels corresponding to logs arriving later in time. The *morwi* chooses a path to follow according to the level of pheromones on it. A path with a higher pheromone level has a higher probability of being chosen. At each node, the *morwi* waits a certain time for the arrival of logs in the following level. Figure 7 shows an event tree with the path followed by a *morwi*. The sequence of logs found is returned as an alert. After human validation, the level of pheromones in that path is incremented if the sequence corresponds to a threat, and decremented otherwise. More details about the algorithm can be found in [24].

5 Discussion

5.1 Innovation in log analysis

In classical SIEMs, correlation is based on rules composed of text strings, which are searched as patterns in the logs in a linear way. If a log matches a rule, it becomes a possible candidate to be part of an attack scenario. Each rule matches one of the steps in the attack. An alert is generated when the whole scenario or part of it is detected. One of the disadvantages of rule-based analysis engines is the description of the rule sets. They are manually written by analysts, so it is easy to find errors. In addition, the volume of rules that must be created is too high to be managed by a human analyst, as the number of technologies used in an organization continually increases. In the cognitive framework developed for HuMa, there are no predefined and static rules. In contrast to classic correlations where the human analyst is situated at the end of the linear process chain, the objective in HuMa is to place the analyst at the core of the analysis process. This means that the analyst can intervene at any point of the process. The interface with the human operator is a key component in the conception of HuMa. The learning loop, which allows the system to automatically learn new links between the logs, turns the system into an extension of the analyst's way of thinking. Besides, HuMa is not intended as a substitute for classical SIEMs, but to complement them. Rule-based systems are necessary for detecting well-known threats. We can obtain signatures directly from security vendors, whose research teams identify and analyze attacks from all over the world.

5.2 Innovation in the representation of logs and attacks

HuMa also proposes an innovative approach to processing and representing information. SIEMs are generally based on a broad classification of logs, and not much work has been done on the development of well defined ontologies. In HuMa, we incorporated the work on log concepts developed by Legrand [21], who applies an ontology based on security indicators. The transformation of raw logs into sets of concepts allows the preservation of the original information, which is enriched with underlying meaning provided by security analysts. The automatic semantic analysis is crucial in HuMa, resulting in more enriched logs that allow security analysis methods to work in a more efficient way. Moreover, these concepts are better understood by humans than raw log text, so they are also useful to the security analyst during an investigation. The multi-step nature of APTs necessitate an innovative way of representing attacks. In the context of HuMa, we propose a novel approach to model APTs that integrates low-level events with attack patterns to identify relations between them. The model relies on three layers: one for events, one for context and known attack patterns, and the assessment layer where the model of the advanced persistent threat is stored. Existing approaches focus on handling of events [14], or rely on existing attack patterns to be matched with detected events [10]. Our approach combines both of these. This representation of attacks is at the core of the set of security analysis algorithms developed for HuMa. Having a common format eases the exchange of information between algorithms. The analyst can thus obtain a single result, which is the combined outcome of the set of methods.

6 Conclusions and perspectives

In this work, we introduce, implement and evaluate a complete multi-layer investigation framework to address the challenge of Advanced Persistent Threats. This framework is organized into three layers: the assessment layer, the context and attack pattern layer, and the event layer. We propose and evaluate a set of algorithms for the assessment layer, including temporal and spatial dependencies, root cause analysis, and ant-colony based analysis. A qualitative application of the framework to the Carbanak attack is presented. The investigation process for the assessment layer algorithm is defined. A quantitative evaluation of the contribution of the context and attack pattern layer to the investigation performance is given. This highlights how the integration of insights from CVE and CAPEC resources improves the ability to identify complex attacks such as APTs in massive logs. This work represents a first step in the definition of a comprehensive framework for the investigation of APTs. HuMa still needs to be complemented with more features for the integration of the human expert, who, beyond being a simple observer, also has the knowledge required to enrich the preliminary analyses proposed by the framework. Assisted learning is likely to become a major topic of interest for security investigation in the near future.

References

1. Abraham, S., Nair, S.: A predictive framework for cyber security analytics using attack graphs. International Journal of Computer Networks & Communications (January 2015), http://arxiv.org/abs/1502.01240
2. Allodi, L., Massacci, F.: A preliminary analysis of vulnerability scores for attacks in wild: The ekits and sym datasets. In: Proceedings of the 2012 ACM Workshop on Building Analysis Datasets and Gathering Experience Returns for Security. pp. 17–24. BADGERS '12, ACM, New York, NY, USA (2012), http://doi.acm.org/10.1145/2382416.2382427
3. Arnold, F., Hermanns, H., Pulungan, R., Stoelinga, M.: Time-dependent analysis of attacks. In: Principles of Security and Trust, Lecture Notes in Computer Science, vol. 8414, pp. 285–305. Springer Berlin Heidelberg (2014), http://dx.doi.org/10.1007/978-3-642-54792-8_16
4. Benali, F., Ubéda, S., Legrand, V.: Collaborative approach to automatic classification of heterogeneous information security. In: Emerging Security Information, Systems and Technologies, 2008. SECURWARE'08. Second International Conference on. pp. 294–299. IEEE (2008)
5. Camtepe, S., Yener, B.: Modeling and detection of complex attacks. In: SecureComm Third International Conference on Security and Privacy in Communications Networks and the Workshops, 2007. pp. 234–243 (Sept 2007)
6. Chen, B., Lee, J., Wu, A.S.: Active event correlation in bro ids to detect multistage attacks. In: Fourth IEEE International Workshop on Information Assurance (IWIA'06). pp. 16 pp.–50. IEEE, London, United Kingdom (2006)
7. Chen, P., Desmet, L., Huygens, C.: A study on advanced persistent threats. In: Communications and Multimedia Security, Lecture Notes in Computer Science, vol. 8735, pp. 63–72. Springer Berlin Heidelberg (2014)
8. Chen, T.M., Abu-Nimeh, S.: Lessons from stuxnet. Computer 44(4), 91–93 (2011)
9. Coudriau, M., Lahmadi, A., Francois, J.: Topological Analysis and Visualisation of Network Monitoring Data: Darknet case study. In: International Workshop on Information Forensics and Security (WIFS). IEEE, Abu Dhabi, United Arab Emirates (2016)
10. Cui, Z., Herwono, I., Kearney, P.: Multi-stage attack modelling. In: Proceedings of Cyberpatterns 2013, pp. 78–89 (2013)
11. De Santis, G., Lahmadi, A., Francois, J., Festor, O.: Modeling of ip scanning activities with hidden markov models: Darknet case study. In: 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS), 2016. pp. 1–5. IEEE (2016)
12. Flåten, O., Lund, M.S.: How good are attack trees for modelling advanced cyber threats? Norwegian Information Security Conference (NISK) 7(1) (2014)
13. Friedberg, I., Skopik, F., Settanni, G., Fiedler, R.: Combating advanced persistent threats: From network event correlation to incident detection. Comput. Secur. 48, 35–57 (2015)
14. Giura, P., Wang, W.: Using large scale distributed computing to unveil advanced persistent threats. SCIENCE 1(3), pp–93 (2013)
15. Kordy, B., Pièvre-Cambacèdés, L., Schweitzer, P.: Dag-based attack and defense modeling: Don't miss the forest for the attack trees. Computer science review 13-14, 1–38 (2014)
16. Kotenko, I., Chechulin, A.: A cyber attack modeling and impact assessment framework. In: 2013 5th International Conference on Cyber Conflict (CYCON 2013). pp. 1–24 (June 2013)

17. Lagraa, S., Legrand, V., Minier, M.: Behavioral change-based anomaly detection in computer networks using data mining. Int. J. Network Manage. (Submitted)
18. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Jebara, T., Xing, E.P. (eds.) Proceedings of the 31st International Conference on Machine Learning (ICML-14). pp. 1188–1196. JMLR Workshop and Conference Proceedings (2014)
19. Legrand, V., State, R., Paffumi, L.: A dangerousness-based investigation model for security event management. In: Internet Monitoring and Protection, 2008. ICIMP'08. The Third International Conference on. pp. 109–118. IEEE (2008)
20. Legrand, V., Ubeda, S.: Enriched diagnosis and investigation models for security event correlation. In: Internet Monitoring and Protection, 2007. ICIMP 2007. Second International Conference on. pp. 1–1. IEEE (2007)
21. Legrand, V.: Confiance et risque pour engager un ?change en milieu hostile. Ph.D. thesis, INSA-Lyon (2013)
22. Marchetti, M., Colajanni, M., Manganiello, F.: Identification of correlated network intrusion alerts. In: Third International Workshop on Cyberspace Safety and Security (CSS). pp. 15–20. IEEE, Milan, Italy (2011)
23. Mathew, S., Upadhyaya, S.: Attack scenario recognition through heterogeneous event stream analysis. In: IEEE Military Communications Conference (MILCOM). pp. 1–7. IEEE, Boston, MA, USA (2009)
24. Navarro-Lara, J., Deruyver, A., Parrend, P.: Morwilog: an ACO-based system for outlining multi-step attacks. In: IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, Athens, Greece (2016)
25. Offroy, M., Duponchel, L.: Topological data analysis: A promising big data exploration tool in biology, analytical chemistry and physical chemistry. Analytica Chimica Acta 910, 1 – 11 (2016)
26. Pearson, P., Muellner, D., Singh, G.: TDAmapper: Analyze High-Dimensional Data Using Discrete Morse Theory (2015), https://github.com/paultpearson/TDAmapper/, r package version 1.0
27. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010)
28. Scarabeo, N., Fung, B.C., Khokhar, R.H.: Mining known attack patterns from security-related events. PeerJ Computer Science 1, e25 (Oct 2015)
29. Schneider, B.: Attack trees. Dr. Dobb's Journal (December 1999)
30. Sood, A.K., Enbody, R.J.: Targeted cyberattacks: A superset of advanced persistent threats. IEEE Security & Privacy 11(1), 54–61 (2013)
31. Wang, L., Ghorbani, A., Li, Y.: Automatic multi-step attack pattern discovering. International Journal of Network Security (IJNS) 10(2), 142–152 (2010)
32. Zali, Z., Hashemi, M.R., Saidi, H.: Real-time attack scenario detection via intrusion detection alert correlation. In: 9th International ISC Conference on Information Security and Cryptology (ISCISC). pp. 95–102. IEEE, Tabriz, Iran (2012)
33. Zhang, S., Caragea, D., Ou, X.: An empirical study on using the national vulnerability database to predict software vulnerabilities. In: Proceedings of the 22Nd International Conference on Database and Expert Systems Applications - Volume Part I. pp. 217–231. DEXA'11, Springer-Verlag, Berlin, Heidelberg (2011)
34. Zhaowen, L., Shan, L., Yan, M.: Real-time intrusion alert correlation system based on prerequisites and consequence. In: 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM). pp. 1–5. IEEE, Chengdu City, China (2010)