

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter

DMAPT: Study of Data Mining and Machine Learning Techniques in Advanced Persistent Threat Attribution and Detection

P.V. Sai Charan, P. Mohan Anand and Sandeep K. Shukla

Abstract

Modern-day malware is intelligent enough to hide its presence and perform stealthy operations in the background. Advance Persistent Threat (APT) is one such kind of malware attack on sensitive corporate and banking networks to stay there for a long time undetected. In real-time corporate networks, identifying the presence of intruders is a big challenging task for security experts. Recent APT attacks like Carbanak, The Big Bang, and Red Echo attack (targeting the Indian power sector) are ringing alarms globally. New data exfiltration methods and advancements in malware techniques are the two main reasons for rapid and robust APT evolution. Although many traditional and hybrid methods are available to detect this stealthy malware, the number of target-specific attacks are increasing rapidly at global level. Attackers have been crafting payloads resistant to malware sandbox environments so that traditional sandboxing techniques may not work with these APT malware detection. In this paper, we shed light on various Data Mining, Machine Learning techniques and frameworks used in both Attribution and Detection of APT malware. Added to this, our work highlight GAP analysis and need for paradigm shift in existing techniques to deal with evolving modern APT malware.

Keywords: APT, Targeted Malware, Data Exfiltration, APT Attribution

1. Introduction

Recent advances in the design of sophisticated malware tools are posing a significant challenge not only to the global IT industry but also to the banking and security organisations. Advanced Persistent Threat (APT) is a key player in highly targeted and sophisticated state-sponsored attacks [1]. These APT groups design and deploy malware in a unique way depending on the target. After selecting the targeted organisation, they come with different Tools, Techniques and Procedures (TTP) to bypass the traditional line of defences (intrusion detection systems or firewall). Once they get access, these APT groups stay inside targeted networks for a long time to observe the workflows. These APT groups use intelligent multi-stage malware

deployment techniques to stay low under the radar for a long time [2]. Finally, gathered sensitive information is pushed in small chunks to its external control and command servers (C2C) using some clever exfiltration techniques.

The whole process of the APT life cycle is broadly divided into seven different phases as shown in **Figure 1** [3]. In the Reconnaissance phase, the attacker chooses the target network and studies the internal network structure and comes up with the necessary strategy, TTP, to bypass the initial layer of defence. Reconnaissance is followed by the Initial compromise phase, where attackers exploit open vulnerabilities to get an initial foothold into the targeted network. After that, the attackers try to replicate and propagate into another machine and establishes backdoors to pull more sophisticated payloads in Establishing foothold phase. Later in the Lateral movement phase, attackers escalate various privileges to perform more sophisticated tasks to hide its traces. In this particular phase, attackers traverse from one network to another network in search of sensitive information. After collecting the necessary data, the attackers strategically centralises this collected data to staging servers. In the data exfiltration phase, attackers use different custom encoding and encryption mechanisms to push these collected data to external control and command servers. Finally, to preserve the anonymity of the process, attacker leaves no traces by clearing the tracks and creates a backdoor to revisit that particular organisation in the future.

APT has grown to become a global tool for cyber warfare between countries. Carbanak APT campaign infected thousands of people worldwide and caused nearly \$1 billion damage across the globe [4]. APT actors carried out a variety of actions in this operation, including opening fraudulent accounts and employing bogus services to obtain funds, as well as sending money to cybercriminals via the SWIFT (Society for Worldwide Interbank Financial Telecommunication) network. Similarly, in 2018, Big Bang APT developed a much more robust and sophisticated multi-stage malware targeting the Palestinian Authority [5]. This APT malware includes several modules

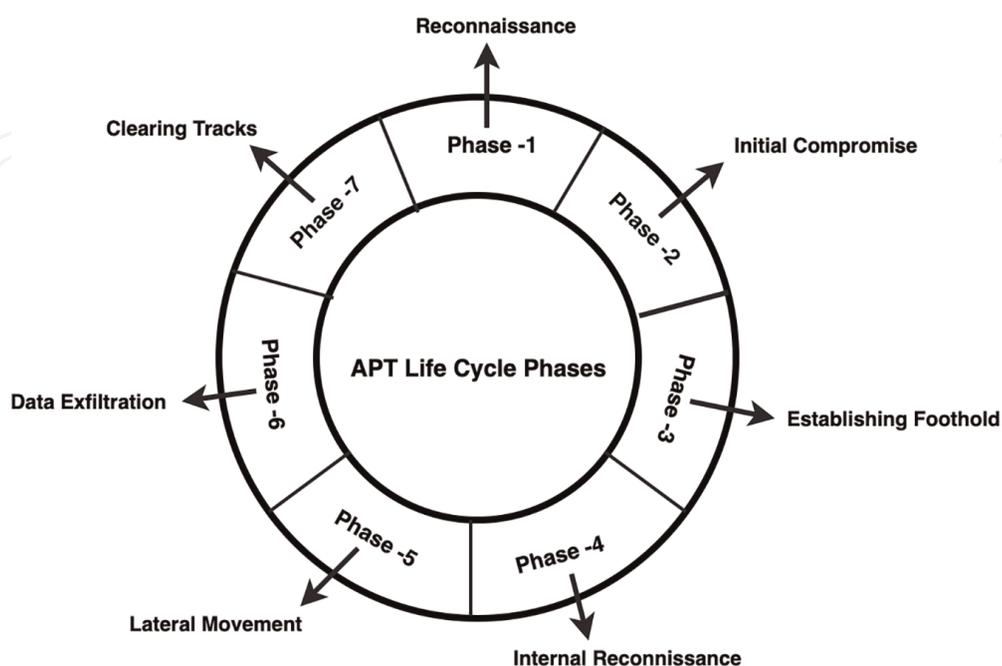


Figure 1.
APT life cycle phases.

that perform tasks ranging from obtaining a file list, capturing screenshots, rebooting the machine, retrieving system information, and self-deletion. More recently, a supply chain attack on solar winds by the Russian APT group was considered one of the sophisticated attacks. RefreshInternals() method in solar winds attack depict the maturity of these state-sponsored APT groups in terms of malware design and payload delivery [6].

In order to deal with these kinds of state-sponsored targeted attacks, security experts consider APT attribution and detection as two key pillars. Attribution is an analysis process that explains about “who” is behind particular cyber espionage and “why” they have done it [7]. This process gives insights about particular APT threat actors and their targeted areas as well. Based on this preliminary information, the security community try to detect these attacks by fixing issues at different levels of an organisation. Since APT attribution and detection became crucial for many security firms/govt agencies, both these processes require massive data pre-processing and analysis. To address these issues, researchers propose different data mining and machine learning techniques in both attribution and detection as well. In this paper, we discuss various data mining and machine learning techniques in both detection and attribution of APT malware. In addition to this, we compare different detection techniques, and we highlight research gaps among those techniques which need to be addressed by the security community to combat this sophisticated APT malware.

This paper is organised as follows. Section 1. details APT overview and phases of APT, followed by the need for data mining and ML techniques in both attribution and detection of APT malware. Section 2. talks about the process of attribution and different techniques proposed to perform APT attribution. Section 3. discuss about various state of the art data mining and ML techniques proposed by the research community in APT detection. Section 4. details research gap analysis followed by conclusion and future scope.

2. Data mining and ML techniques in APT attribution

APT attribution is an analysis process that reveals the identity of the threat actors and their motto through a series of steps [8]. First, security firms collect data from different victim organisations by performing forensic analysis on the respective networks and collect different Indicators of Compromise (IOC). In general, attackers repeat this pattern in several other organisations as well. Security firms observe and analyse these repeated patterns in IOC and TTP’s together, and cluster these combinations as intrusion sets. Performing data analytics on these intrusion sets over a period will eventually reveal the threat actor and motivation behind the attack as depicted in **Figure 2**, respectively.



Figure 2.
Overview of APT attribution process.

2.1 DeepAPT: APT attribution using deep neural network and transfer learning

APT attribution is quite a challenging task to the security community because of various reasons. Majorly, State-sponsored APTs are developed in the supervision of different units and equipped with default Anti-VM and Anti-Debugging techniques to obfuscate the payloads. This technique makes feature extraction extremely challenging to most security firms. In addition to this, APT malware samples are highly targeted so that very few samples will be available for analysis purposes. In order to address this issue, Rosenberg et al. proposed a technique for APT attribution by using a Deep Neural Network (DNN) classifier [9]. In this research work, the authors used 3200 malware samples for training DNN classifiers, 400 samples for validation and 1000 samples for testing the model. All the APT malware samples are executed in a cuckoo sandbox environment, and generated reports are used as raw input in training the classifier. DNN is effective in learning high-level features on its own from raw inputs. In order to train DNN models more effectively, in this work, the authors removed top 50,000 frequent words from input features of all cuckoo reports. So, DNN models take very uncommon words from all cuckoo reports and build a much more effective model to perform APT attribution. This DNN architecture is a 10-layer, fully connected network (50,000 neurons at the input layer and 2,000 in the first hidden layer) with a ReLU activation function. The final trained APT attribution model did decent work on test data with 98.6% accuracy. Added to this, the authors also applied transfer learning on trained DNN models by removing and retraining top layer neurons. After applying transfer learning, the model still performs exceptionally well with 97.8% accuracy. From the t-distributed stochastic neighbour embedding algorithm (used to reduce from 500 dimension space to 2 dimension space), we can see that the trained model could separate different APT malware groups as shown in **Figure 3**, respectively.

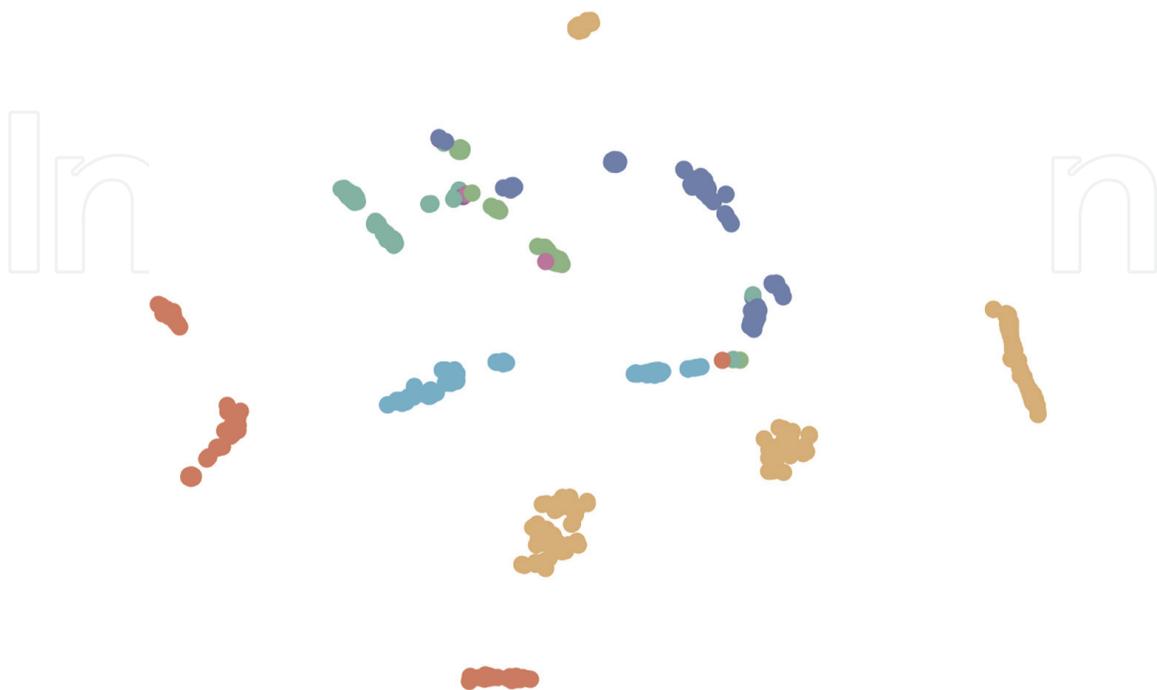


Figure 3.
2-dimensional visualisation of APT families using t-SNE algorithm [9].

2.2 APT attribution based on threat intelligence reports

Most APT attribution techniques heavily rely on performing analysis for malware samples used in that particular campaign. The key disadvantage of this strategy is that the same malware samples can be used in several operations. In some situations, APT groups specifically buy malware from the dark web based on their requirements. So, the ML models constructed by only considering malware samples may not give efficient results in terms of APT attribution. In order to address this issue, Lior Perry et al. proposed a method named NO-DOUBT, i.e. Novel Document Based Attribution, by constructing models on threat intelligence reports with the help of Natural Language Processing (NLP) techniques [10]. In this research, the authors collected 249 threat intelligence reports of 12 different APT actors and considered APT attack attribution as a multi-text classification problem. The proposed model consists of mainly two phases, as shown in **Figure 4**. In the training phase, labelled reports and word embeddings transform the input data to a vector representation. For generating this vector representation, authors propose SMOBI (Smoothed Binary Vector) algorithm, which will find cosine similarities between input words in labelled data sets and word embeddings to form a huge $n \times m$ matrix. This vector representation and labels are given to the ensemble xGBoost classifier to construct a known actor model. In the deployment phase, new test reports (unlabelled) are also converted to vector representation and given to the known actor model to determine the probability predictions to the known classes. These probability predictions are given to a New Actor Model (a binary classifier that outputs whether it is a known APT actor or a new unknown actor) to make final predictions. Although this model struggles to detect Deep Panda and APT29 actors, SMOBI based APT attribution outperforms previous text-based APT attribution models (unigrams + bigrams and tf-idf) in terms of Accuracy, Precision and Recall.

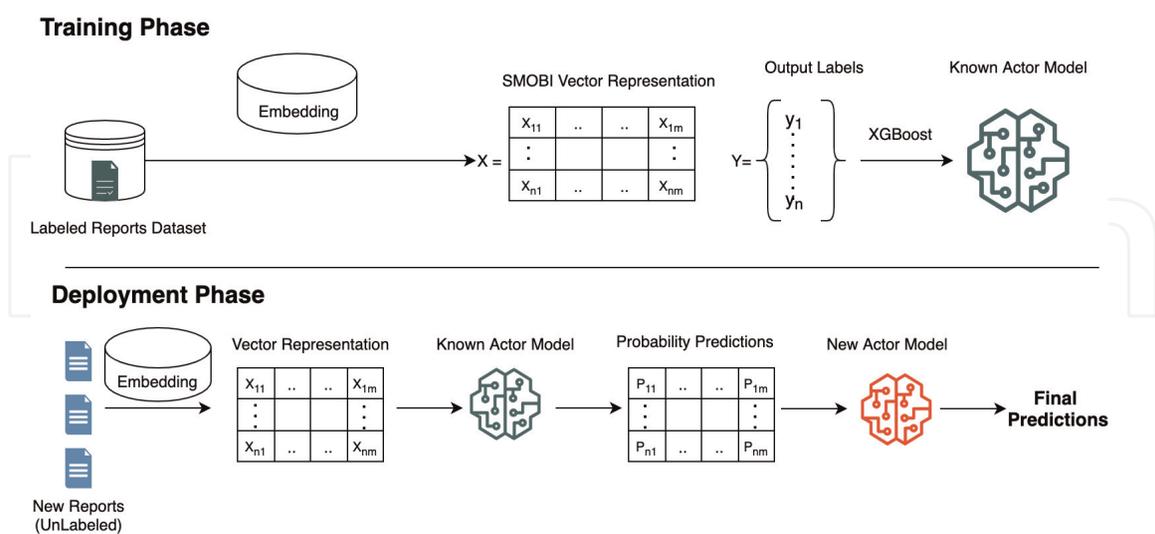


Figure 4. NO-DOUBT method for APT attribution [10].

2.3 ML based attribution framework using high level IOC

Most of the APT attribution processes depend upon the manual analysis in victim networks and collecting low-level indicators of compromise (forensic analysis at

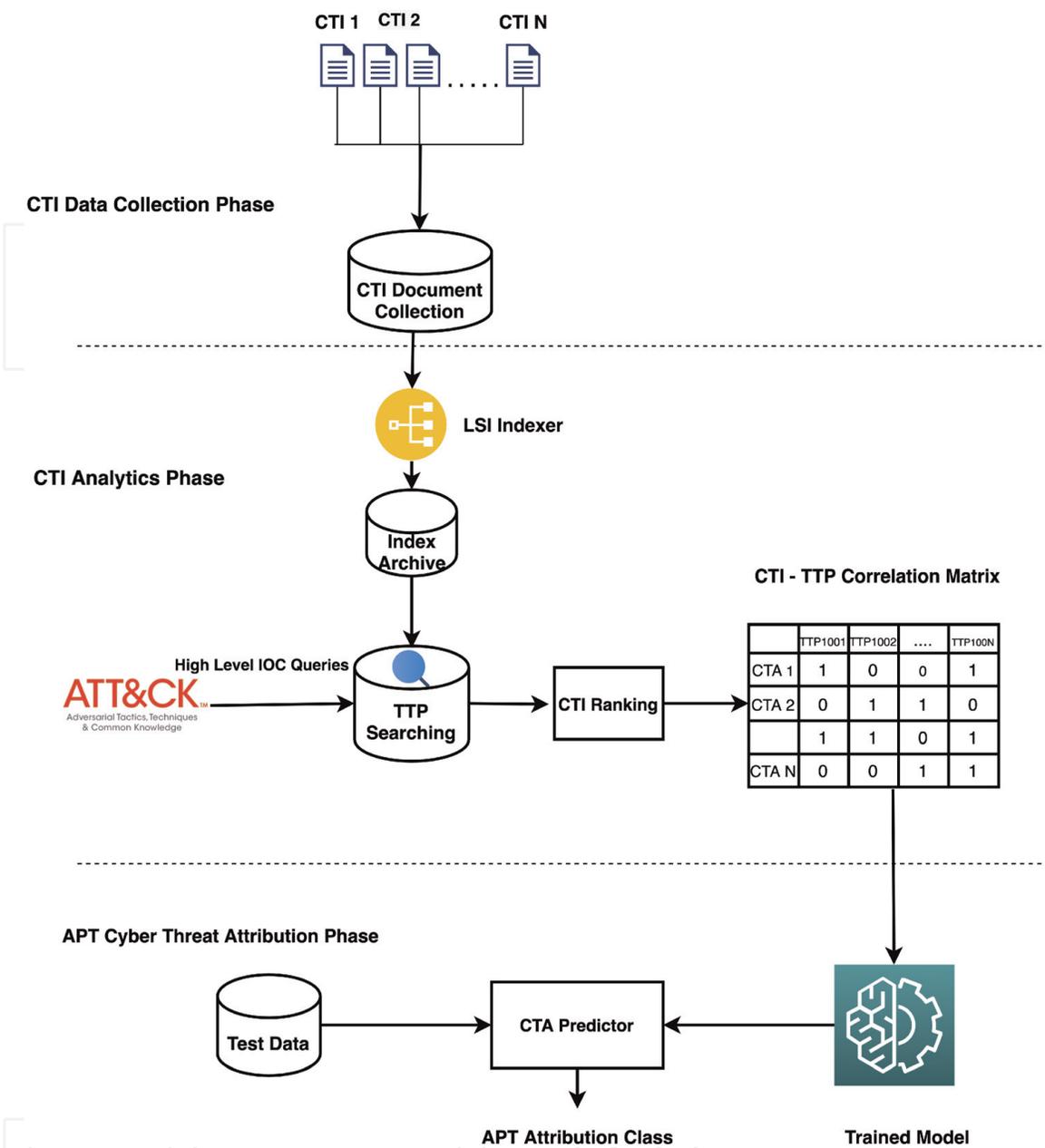


Figure 5. Cyber threat attribution framework [11].

firewalls, tracebacks, IDS and Honeypots). However, APT actors change this low-level IOC from one organisation to another organisation. ML models built based on this low-level IOC, results in inadequate cyber intelligence systems. On the other hand, collecting high-level IOC's for each organisation is time-consuming. Such high-level IOC's are published in the form of Cyber Threat Intelligence (CTI) reports across the organisations as a common practice. In 2019, Umara Noor et al. proposed a distributional semantic technique of NLP to build a cyber threat attribution framework by extracting patterns from CTI reports [11]. The proposed attribution framework is broadly divided into three phases, as depicted in **Figure 5**. In this experiment, authors used a customised search engine to collect 327 unstructured CTI documents corresponding to 36 APT actors as a part of data collection phase. The CTI documents do not contain the exact keyword described in the standard taxonomy due to varying textual definitions and choices for communicating a concept. Rather than using a simple

keyword-based search, the authors developed a semantic search method based on the statistical distributional semantic relevance technique (LSA), to retrieve relevant documents. The input CTI records are indexed using LSA. The statistically derived conceptual indices (from LSA indexer) are searched for semantically relevant topics using the high-level IOC labels specified in MITRE ATT&CK [11]. Based on cosine similarity, the CTA-TTP correlation matrix is constructed in the CTI analytics phase. ML models are built on top of the CTA-TTP correlation matrix in the Cyber Threat Attribution phase. Among various classifiers, the Deep Neural Network turned out to be the best performer with 94% attribution accuracy on test data with high precision and recall values.

2.4 APTMalInsight: recognising APT malware based on system call information and ontology framework

Behavioural analysis of APT malware gives better insights on both APT attribution and detection. Based on this motivation, Weijie Han et al. proposed that, dynamic system call information reveals behavioural characteristics of APT malware [12]. Furthermore, the authors built an ontology model to understand in-depth relation between the maliciousness of APT malware to its families, as depicted in **Figure 6**, respectively. APTMalInsight framework mainly consists of two modules i.e. APT malware family classification module and detection module. The basic concept behind the APTMalInsight framework is to profile the behavioural characteristics of APT malware. It obtains dynamic system call information from the programs to reliably detect and attribute APT malware to their respective families. Primarily, APT malware samples are executed to extract dynamic API calls. After extracting API calls, authors calculated the feature importance of each API call and built a feature vector by selecting top N-API calls from the API call sequence. ML models built on top of that feature vector will output the APT attribution class for test data, as shown in **Figure 7**. For the experiment, authors considered a total of 864 APT malware samples belonging to five different families. As per the experimentation results, Random Forest turned out to be the best model in terms of Accuracy(98%), Precision and Recall for APT malware family attribution.

2.5 ATOMIC: FireEye's framework for large scale clustering and associating APT threat actors

Security firms like FireEye investigate many victim networks and collect IOC and group them together as uncategorised (“UNC”) intrusion sets. Over time, this type of UNC sets are increasing rapidly, and security firms need to either merge these other APT groups or assign a new group name based on manual analysis. FireEye security

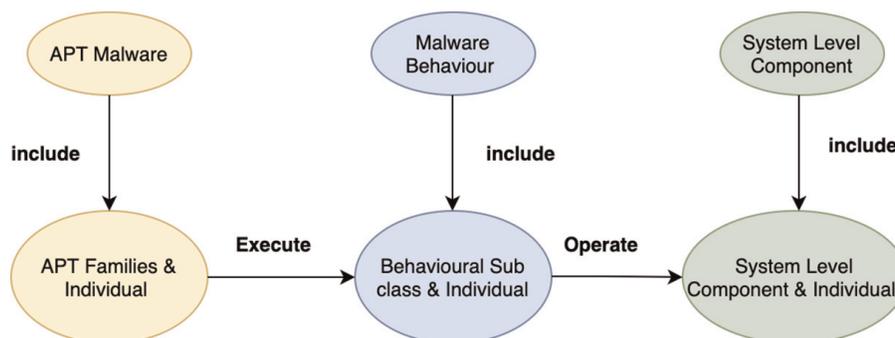


Figure 6.
APT malware ontology model [12].

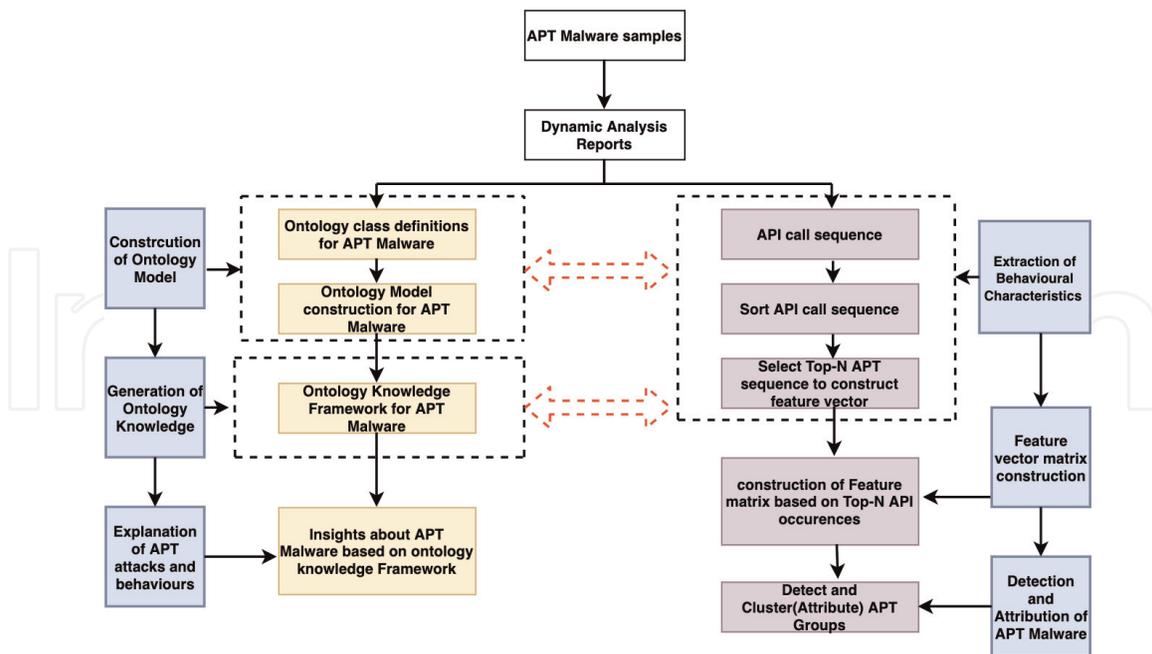


Figure 7. High-level overview of APTMalInsight framework [12].

researchers proposed an automated framework with the help of ML models to perform investigation, analysis, and rationale for the whole APT attribution process [13]. In this framework, the researchers suggest a document clustering approach using term frequency and - inverse document frequency method (TF-IDF). The TF-IDF algorithm assigns more importance to a term if the word often appears in the document. Similarly, if the term appears common across all the documents, the algorithm decreases its importance. This method favours unique terms like custom malware families, which may appear in just a few classes, and downplays popular terms like ‘phishing’, which appear more often. After calculating scores using the TF-IDF algorithm, each UNC group is converted into a vector representation, and researchers calculate cosine similarity between these APT groups as shown in **Figure 8**, respectively. As angle between the two vectors decreases, they tend to become parallel. The decrease in the angle helps the researchers to determine the extent of similarity

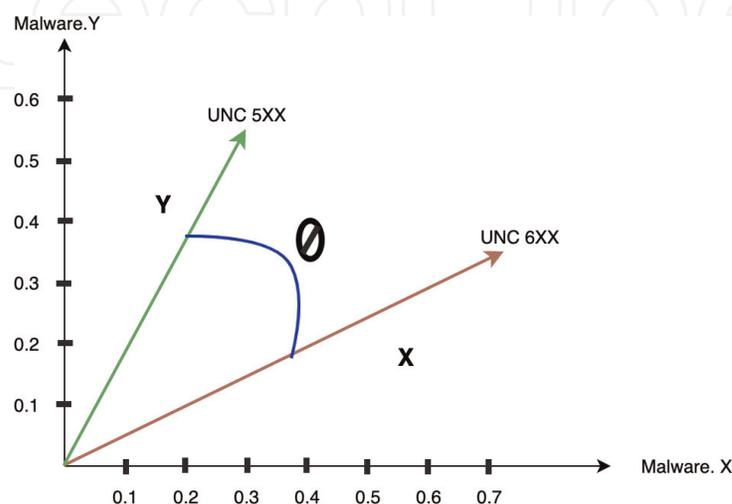


Figure 8. Cosine similarity between different un-attributed APT groups [13].

between two different APT groups. Based on this idea, FireEye automated the whole process of APT attribution and merging different uncategorised groups.

3. Data mining and ML techniques in APT detection

Most of the APT families stay undetected for a long period and use intelligent ways to damage the vulnerable hosts. When a traditional malware executes, most of the events occur sequentially and leave some traces behind. These traces help modern-day intelligent systems like SIEM, IDS, IPS to prevent these attacks. But, when it comes to the case of APTs, they clean the attack traces and also prevent sequence execution of events. Also, APT employs Anti-VM and Anti-debugging techniques for making things harder for the detection systems. The hardness in detecting the APT has made the cyber security enthusiasts draw their attention towards this domain. Some of the important contributions in the research area are mentioned below. A detailed comparison among different detection techniques are illustrated in **Table 1**, respectively.

| Research item | DM/ML technique employed in APT detection | Input data | Novelty |
|---------------|--|--|---|
| [14] | RNN-LSTM and GHSOM | Network Traffic Flow | Deep learning stack with sequential neural networks to detect APT. |
| [15] | Provenance Graph Mining | Host Audit Data (Linux audit or Windows ETW) | Suspicious information flows are identified using MITRE ATT&CK framework. |
| [16] | Directed Graph Mining and One Class SVM | SIEM Event Logs | Extracting attack vectors from SIEM logs |
| [17] | Continuous Association Rule Mining Algorithm (CRAMA) | IDS Logs | Identify correlation rules between various system events to develop an APT attack graph |
| [18] | RNN-LSTM | SIEM Event Logs | Identify possible event codes and their sequence to detect an APT attack in realtime |
| [19] | Ensemble Classifier | Network Traffic Flow | Separate threat detection sub-module for APT life cycle phases. |
| [20] | Multi fractal based error minimization | Network Traffic Flow | Multi fractal analysis to extract the hidden information of TCP connections. |
| [21] | Correlation Analysis | Multiple data sources | Construction of Attack Pyramid using multiple planes to detect APT |
| [22] | J48 Classifier | API log data | API calls to track process injection and privilege escalation activities. |
| [23] | Ensemble Classifier | Domain Names (Alexa and data.netlab.360) | Identify malicious C2C communication using lexical features of domain names |
| [24] | Ensemble Classifier | Domain Names (Alexa and DGArchive) | Identify malicious C2C communication using lexical, network features of domain names |

Table 1.
 Comparison of different APT detection methods.

3.1 A novel deep learning stack for APT detection

Tero et al. [14] proposed a theoretical approach for detecting APT by developing a stack of Deep Learning methods where each layer has a particular task in handling APT events. The authors consider network payload and packet header information as features, and they streamlined the input to the detection stack without any data filtering mechanism. The detection stack is designed sequentially. The initial layers, i.e. layer-1 and layer-2, are used to detect the known attacks and legitimate network traffic from the data flow respectively. Layer-3 of the detection stack employs in identifying the outliers having historical presence. It uses Recurrent Neural Network-Long Short Term Memory (RNN-LSTM) units to confirm whether an outlier has historical occurrence. Layer-4 helps to classify the outliers into four categories, i.e. regular traffic, known attack, predicted attack and unknown outlier using an anomaly detection method named Growing Hierarchical Self-Organising Map (GHSOM). The stack's final layer helps to map the anomalies (i.e. interconnections between the outlier events) using a Graph Database (GDB). The proposed stack model is highly modular and was designed to perform dynamic detection of APT events with a decent detection accuracy. However, this detection system is complex in design and result in higher time complexity when dealing with massive data inputs.

3.2 Real-time APT detection through correlation of suspicious information flows (HOLMES)

HOLMES model of APT detection is strongly based on the principles of the APT kill chain model. The cyber kill chain model gives a higher-level overview of the sequence of events in successful APT espionage, i.e. reconnaissance, command and control communication, privilege escalation, lateral movement, data exfiltration, and trace removal. Audit data from various operating systems are converted to a common data representation format and passed as input to the proposed model in the initial step. Lower-level information flows are extracted from the audit data such as process, files, memory objects and network information etc. The core part of the proposed model is to map the lower-level information data flows to the phases of the APT-kill chain by constructing an intermediate layer. The intermediate layer is responsible for identifying various TTP's (Tools, Techniques, Procedures) from the low-level information data flow that correlates with respective phase of the APT life cycle. The authors considered around 200 TTP patterns based on MITRE ATT&CK framework [15]. The TTP patterns and noise filtering mechanism are employed in constructing a High-Level Scenario Graph (HSG) from which we can detect the APT attack with decent accuracy.

3.3 Anomaly detection in log data using graph databases and machine learning to defend advanced persistent threats

Schindler et al. proposed an APT detection engine based on the principles of APT kill chain phases [16]. In this work, SIEM logs were considered as data source. The correlation is identified between the event logs and the phases of APT kill chain. An adapted kill chain model is constructed to identify the possible attack vectors from the SIEM event logs. This model is implemented at two different levels.

Level-1 deals with graph-based forensic analysis where logs from different programs are aggregated based on timestamp to identify events within the network. A directed graph is constructed from the multiple layers of event sequences. Each event sequence reveals whether the event flow matches with the partial/full phases of the APT kill chain.

Level-2 helps in identifying various anomalous activities using the Machine Learning approach. An ML classifier is constructed to make the model robust in detecting APT events along with the graph model. Authors considered “one-class SVM” as the classifier model and used windows logs, firewall logs, file audit logs of benign system programs as its data source. This model is expected to identify all the events that differ from the benign programs.

The proposed model achieved a decent accuracy score of 95.33% in detecting APT events. However, considering the case of smart malware where malicious programs mimic normal user behaviour, the proposed model tends to produce a relatively high false-positives.

3.4 A study on cyber threat prediction based on intrusion event for APT attack detection

Yong-Ho Kim et al. [17] proposed a theoretical model for APT detection that consider intrusion detection system logs as data source. From the IDS logs, correlation rules between various system events are identified to build an attack graph. Identifying the correlation between the intrusion detection logs helps in predicting the future attacks. In the initial phase, intrusion detection logs are collected and corresponding intrusion events were extracted. The extracted events are passed to different function blocks, each corresponding to a particular detection activity. One of the functional block identifies the single-directional i.e. (host to C2C interaction) and bi-directional (host to C2C, C2C to host) communication activities. Another block identifies the repetitive intrusion events and combines them as a single event to optimise the time and resource constraints. A correlation analysis block identifies the context of intrusion detection events and creates sequential rules based on the principles of 5 W and 1H (When, Where, Why, Who, What and How). Finally, the prediction engine consider the attack scenario and tries to predict one or more events that can occur after a single intrusion event. This module consider data mining principles such as support and confidence to produce the best possible result. The time constraint is one of the practical problems with this model, as some of the functional blocks take a longer time to process events. Another important aspect is that, rules of the intrusion detection systems will directly affect the outcome of this model.

3.5 APT detection using long short term memory neural networks

Charan et al. [18] proposed an APT detection engine that takes SIEM event logs as input and use LSTM neural networks to detect the successful APT espionage. The author consider Splunk SIEM logs as a data source and streamline data to the Hadoop framework to process and obtain the event codes for every activity. Based on the APT life cycle phases, the author listed out the possible event codes and their sequence, leading to successful APT espionage. The core part of this work is to identify the event codes occurring in a sequence, and this process requires memorising the previous state event codes. So, in the proposed model, LSTM (a variant of RNN) is considered a

classifier because it overcomes vanishing gradient problem by remembering the previous state event codes to confirm APT attack presence. However, this model may suffer from a high false-positive rate when smart malware techniques are employed in crafting the APT attack.

3.6 MLAPT: detection of APT attacks using machine learning and correlation analysis

APT detection research mainly rely on the analysis of malware payload used in different phases of APT attack. This kind of approach result in high false positives in case of multi-stage malware deployment. In order to address this issue, Ghafir et al. proposed a model to detect multi-stage APT malware by using machine learning and correlation analysis (MLAPT) [19]. The MLAPT system is broadly divided into three modules, i.e. 1) Threat detection module, 2) Alert correlation module and 3) Prediction module. Initially, network traffic is passed to the Threat detection module in which authors built several submodules to detect multi-stage attacks. The Output alerts from the Threat detection module are passed to the Alert correlation module. Alert correlation module filters redundant alerts and clusters these alerts based on correlation time interval. The correlation indexing sub-module determines a given scenario is either a full APT scenario or sub-APT scenario based on alert correlation score. The prediction module consider sub APT scenarios and predict its probability of becoming a full APT scenario. Based on that prediction module, alerts are escalated to the network security team to stop this APT kill chain. The novelty of this research lies in the detection of APT across all life cycle phases. Added to this, the MLAPT system monitors and detects real-time APT attacks with a decent 81% Accuracy.

3.7 Detection of APT attacks using fractal dimensions

Detecting APT network patterns is a complex task as it tries to mimic the behaviour of regular TCP traffic. APT malware opens and closes TCP connections to its C2C servers like any other regular legitimate connection with a minimal data transfer to stay low under the radar. Single scale analysis does not extract the complexities of this kind of APT traffic and lowers the detection accuracy. Researchers found that current supervised ML models use euclidean based error minimization, which results in high false positives while detecting complex APT traffic. To address these issues, Sana Siddiqui et al. proposed an APT detection model using multi-fractal based analysis to extract the hidden information of TCP connections [20]. Initially, the authors considered 30% of labelled datasets and computed prior correlation fractal dimension values for normal and APT data points. Both these computed values are loaded into the memory before processing the remaining 70% unlabelled dataset. Each point in the remaining 70% dataset is added to both normal and APT labelled dataset, and posterior fractal dimension values are calculated in the next step. The absolute difference between prior and posterior values for both regular and APT samples are calculated to determine the closest cluster to the data point. If fd_anom (absolute difference between prior and posterior for APT sample) \leq fd_norm (absolute difference between prior and posterior for normal sample), then that data point is classified as an APT sample and vice versa. As per the experimental observations, fractal dimension based ML models performs better in terms of accuracy (94.42%) than the euclidean based ML models.

3.8 APT detection using context-based detection framework

Paul Guira et al. proposed a conceptual framework known as the Attack Pyramid for APT detection [21]. In this approach, the goal of the attack (data exfiltration in most of the cases) should be identified and placed on top of the pyramid. Further more, the model identifies various planes such as user plane, application plane, network plane and physical plane where the possibility of attacks are maximised. From the proposed approach, one can identify the correlation between various events across different planes. In general, an APT attack span multiple planes as the attack life cycle progresses. So, it is possible to identify the attack contexts that span through multiple attack planes. Events from different sources, i.e. VPN logs, firewall logs, IDS logs, authentication logs, system event logs are passed as data source to the detection engine. From these logs, the context of attack is identified using correlation rules. In the next step, the suspicious activities are identified by matching the attack contexts using a signature database. This model requires updating signatures at regular intervals to identify new attack contexts in real-time scenarios.

3.9 APT detection system based on API log data mining

Chun-I Fan et al. [22] proposed a generalised way for APT detection using system calls log data. The model was built based on the principles of dynamic malware analysis where API call (system call) events were passed through a detection engine. The novelty of this work lies in the approach of handling the API calls. Modern APT malware is often used to create child processes or inject code into a new process to evade detection. Authors have created a program named "TraceHook" that monitors all the code injection activities. Tracehook outputs the API count for the executable samples (benign/malware), and a machine learning classifier model is constructed on top of the obtained API count values. The proposed model considers only six important DLLs to monitor and can be combined with other APT detection models to build a robust APT detection engine.

3.10 Ensemble models for C2C communication detection

Identifying and stopping a particular life cycle event can break the full APT cycle and minimise damage to a considerable proportion. Based on this idea, researchers proposed various methods to stop malicious C2C communication. Modern-day malware employed a new way to communicate with their C2C server with the help of Domain Generation Algorithms (DGA). DGA creates a dynamic list of domain names in which a few domain names are active for a limited amount of time. So, the malware communicates to a different C2C domain name for every successful communication. This practice helps the smart malware to avoid detection from the traditional antivirus, firewalls, and other network scanning software. Anand et al. [23] proposed a classification technique to detect character-based DGA, i.e. domain names are constructed by concatenating characters in a pseudo-random manner, for example, wqzdsqtuxsbht.com. In this method, author extracted various lexical-based features such as n-grams, character frequencies, and statistical features to build an ensemble classifier. The proposed model can detect character-based DGA domain names with a decent accuracy score of 97%. Charan et al. [24] proposed a similar technique to detect word-based DGA domain names where domain names are constructed by concatenating two or three words from dictionaries, for example crossmentioncare.com. In their model, the author consider lexical, statistical, network-based features to build an

ensemble classifier. A combination of the above two models can detect the C2C communication activity with a decent accuracy.

4. Conclusion and future scope

Although the security community propose different techniques to detect APT malware, there is a clear gap between current detection mechanisms and APT groups evolution. APT attack detection is extremely difficult due to an unavailability of benchmark datasets for training and evaluation. Added to this, constant change in TTP usage by APT groups result in high false positives in terms of detection. Due to the persistent nature of APT campaigns, it is cumbersome to capture the data over a long period of time. This raises the issue of storing and processing such large amounts of data so that real time detection is still a challenging task to the security community. Many state of the art APT detection models can be bypassed using modern Load Off Land Binaries (LolBins) and process injection through fileless malware. Lately, targeted APT malware evolved into a new variant named smart malware which is highly modular, robust and intelligent enough to evade detection from state of the art ML techniques. Along with these issues, adversarial machine learning is a potential threat to the existing detection mechanisms. Some of the APT groups also started using GAN to modify the payloads in such a way to evade detection and attribution as well. In order to address these serious security concerns in APT detection and attribution, there is a need for benchmark datasets and robust ML models working at different levels of the APT kill chain.

Conflict of interest

The authors declare no conflict of interest.

Abbreviations

| | |
|--------|---|
| APT | Advanced Persistent Threat |
| C2C | Control and Command Server |
| LoLBin | Living Off Land Binaries |
| TTP | Tools, Techniques and Procedures |
| GAN | Generative Adversarial Networks |
| DGA | Domain Generation Algorithms |
| CTI | Cyber Threat Intelligence |
| IOC | Indicators of Compromise |
| SMOBI | Smoothed Binary vector |
| GHSOM | Growing Hierarchical Self Organising Map |
| RNN | Recurrent Neural Network |
| LSTM | Long Short Term Memory Neural Networks |
| SIEM | Security Information and Event Management |
| IDS | Intrusion Detection System |
| IPS | Intrusion Prevention System |
| ETW | Event tracing for windows |
| CTA | Cyber Threat Attribution |
| DLL | Dynamic Link Library |

IntechOpen

Author details

P.V. Sai Charan^{*†}, P. Mohan Anand[†] and Sandeep K. Shukla[†]
Department of Computer Science and Engineering, Indian Institute of Technology,
Kanpur, India

*Address all correspondence to: pvcharan@cse.iitk.ac.in

† These authors contributed equally.

IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Stojanović, Branka, Katharina Hofer-Schmitz, and Ulrike Kleb. "APT datasets and attack modeling for automated detection methods: A review." *Computers & Security* 92 (2020): 101734. DOI: <https://doi.org/10.1016/j.cose.2020.101734>
- [2] Zhou, Peng, et al. "Detecting multi-stage attacks using sequence-to-sequence model." *Computers & Security* 105 (2021): 102203. DOI: <https://doi.org/10.1016/j.cose.2021.102203>
- [3] APT Security: What Are Advanced Persistent Threats?. [Internet]. 2020. Available from : <https://securitytrails.com/blog/advanced-persistent-threats-apt> [Accessed: 25 May 2021]
- [4] Kaspersky Lab: The Great Bank Robbery: The Carbanak APT (Detailed Investigation Report). [Internet]. 2015. Available from : <https://securelist.com/the-great-bank-robbery-the-carbanak-apt/6873/> [Accessed: 25 May 2021]
- [5] The Big Bang APT Report. [Internet]. 2018. Available from: <https://research.checkpoint.com/apt-attack-middle-east-big-bang/> [Accessed: 25 May 2021]
- [6] Microsoft Internal Solorigate Investigation Update. [Internet]. 2020. Available from : <https://blogs.microsoft.com/on-the-issues/2020/12/17/cyberattacks-cybersecurity-solarwinds-fireeye/> [Accessed: 25 May 2021]
- [7] Steffens, Timo. Attribution of Advanced Persistent Threats. Springer Berlin Heidelberg, 2020. DOI : <https://doi.org/10.1007/978-3-662-61313-9>
- [8] The power of APT attribution. [Internet]. 2016. Available from : <https://media.kaspersky.com/en/business-security/enterprise/threat-attribution-engine-whitepaper.pdf>. [Accessed: 25 May 2021].
- [9] Rosenberg, Ishai, Guillaume Sicard, and Eli Omid David. "DeepAPT: nation-state APT attribution using end-to-end deep neural networks." *International Conference on Artificial Neural Networks*. Springer, Cham, 2017. DOI: https://doi.org/10.1007/978-3-319-68612-7_11
- [10] Perry, Lior, Bracha Shapira, and Rami Puzis. "NO-DOUBT: Attack attribution based on threat intelligence reports." 2019 IEEE International Conference on Intelligence and Security Informatics (ISI). IEEE, 2019. DOI: 10.1109/ISI.2019.8823152 [Accessed: 25 May 2021]
- [11] Noor, Umara, et al. "A machine learning-based FinTech cyber threat attribution framework using high-level indicators of compromise." *Future Generation Computer Systems* 96 (2019): 227-242. DOI : <https://doi.org/10.1016/j.future.2019.02.01> [Accessed: 25 May 2021]
- [12] Han, Weijie, et al. "APTMalInsight: Identify and cognize APT malware based on system call information and ontology knowledge framework." *Information Sciences* 546 (2021): 633-664. DOI: <https://doi.org/10.1016/j.ins.2020.08.095> [Accessed: 25 May 2021]
- [13] Going ATOMIC: Clustering and Associating Attacker Activity at Scale. [Internet]. 2019. Available from : <https://www.fireeye.com/blog/threat-research/2019/03/clustering-and-associating-attacker-activity-at-scale.html> [Accessed: 25 May 2021]
- [14] Bodström, Tero, and Timo Hämäläinen. "A novel deep learning

stack for APT detection.” *Applied Sciences* 9.6 (2019): 1055. DOI : <https://doi.org/10.3390/app9061055> [Accessed: 25 May 2021]

[15] Milajerdi, Sadegh M., et al.” Holmes: real-time apt detection through correlation of suspicious information flows.” 2019 IEEE Symposium on Security and Privacy (SP). IEEE, 2019. DOI: 10.1109/SP.2019.00026 [Accessed: 25 May 2021]

[16] Schindler, Timo.” Anomaly detection in log data using graph databases and machine learning to defend advanced persistent threats.” arXiv preprint arXiv: 1802.00259 (2018). DOI: 10.18420/in2017_241 [Accessed: 25 May 2021]

[17] Kim, Yong-Ho, and Won Hyung Park.” A study on cyber threat prediction based on intrusion detection event for APT attack detection.” *Multimedia tools and applications* 71.2 (2014): 685-698. DOI : <https://doi.org/10.1007/s11042-012-1275-x> [Accessed: 25 May 2021]

[18] Charan, PV Sai, T. Gireesh Kumar, and P. Mohan Anand.” Advance persistent threat detection using long short term memory (LSTM) neural networks.” *International Conference on Emerging Technologies in Computer Engineering*. Springer, Singapore, 2019. DOI : https://doi.org/10.1007/978-981-13-8300-7_5 [Accessed: 25 May 2021]

[19] Ghafir, Ibrahim, et al.” Detection of advanced persistent threat using machine-learning correlation analysis.” *Future Generation Computer Systems* 89 (2018): 349-359. <https://doi.org/10.1016/j.future.2018.06.055> [Accessed: 25 May 2021]

[20] Siddiqui, Sana, et al.” Detecting advanced persistent threats using fractal

dimension based machine learning classification.” *Proceedings of the 2016 ACM on international workshop on security and privacy analytics*. 2016. DOI: <https://doi.org/10.1145/2875475.2875484> [Accessed: 25 May 2021]

[21] Giura, Paul, and Wei Wang.” A context-based detection framework for advanced persistent threats.” 2012 International Conference on Cyber Security. IEEE, 2012. DOI : 10.1109/CyberSecurity.2012.16 [Accessed: 25 May 2021]

[22] Fan, Chun-I., et al.” Malware detection systems based on API log data mining.” 2015 IEEE 39th annual computer software and applications conference. Vol. 3. IEEE, 2015. DOI : 10.1109/COMPSAC.2015.241 [Accessed: 25 May 2021]

[23] Anand, P. Mohan, T. Gireesh Kumar, and PV Sai Charan.” An Ensemble approach for algorithmically generated domain name detection using statistical and lexical analysis.” *Procedia Computer Science* 171 (2020): 1129-1136. DOI : <https://doi.org/10.1016/j.procs.2020.04.121> [Accessed: 25 May 2021]

[24] Charan, PV Sai, Sandeep K. Shukla, and P. Mohan Anand.” Detecting Word Based DGA Domains Using Ensemble Models.” *International Conference on Cryptology and Network Security*. Springer, Cham, 2020. DOI : https://doi.org/10.1007/978-3-030-65411-5_7 [Accessed: 25 May 2021]