International Journal of Digital Information and Wireless Communications (IJDIWC) 11(1): 12-22
The Society of Digital Information and Wireless Communications, 2021 ISSN: 2225-658X (Online); ISSN 2412-6551 (Print)

Innovative Architectural Framework Design for an Effective Machine Learning Based APT Detection

Mourad M.H Henchiri and Sharyar Wani
KICT, IIUM, Malaysia
mourad@unizwa.edu.om
sharyarwani@iium.edu.my

ABSTRACTS

Generating regular rules to be passed to different security appliances set within the work environment, is a stressing job to be carefully set; since even the data bouncing and the data collision impact over a network trunk might be considered as an anomaly by a variety of filters [45, 47, 48, 49, 52]. Network data flows load balancing, also, leads to a pattern based anomaly when not configured as per the network potential [46, 48, 49]. Thus, securing a platform against the APT attacks, whether a prevention scenario or a detection process, research demonstrates that security intelligence and big data analytics would enormously prevent and detect abnormalities, this is all by keeping an eye on the difficulty of data classification [44, 50, 51, 52, 53, 56]. In this research we would be generating an APT detection framework diagram via which we enhance the weaknesses seen in regular and commercialized filter based IDS, IPS and Firewalls. Which would give a remarkably enhanced live data flow clustering and classification algorithm.

Key Words: APT, IDS, IPS, FW, ML, Malware, Framework.

I- INTRODUCTION

Our dependence on the services offered by the digital applications, digital environment and remote services is growing. We use systems and applications to shop online, to travel, to communicate, for entertainment, and we use professional dedicated software to accomplish duties and work, for businesses, etc. This digital craze has created a wide digital economy that is gathering momentum from year to year. As a result, attacks, with various and varied motivations, have developed and become more and more sophisticated. They mainly target data related to economic activities.

Thus, they cause significant damage to the overall functioning of digital solutions, and work platforms.

However, several efforts are being made by a growing community around the security of digital assets and applications. This community has become aware of the risks associated with the exposure of information systems on the public or to the Internet. Consequently, it

actively contributes through productions on an informational level; classification and inventory of attacks, publication of vulnerabilities, good practices, etc., but also through operational solutions such the Intrusion detection systems, firewalls, anti-malware, ... to mitigate the impact of attacks against remote services and applications. These solutions are based on well proven approaches, in particular, in systems of intrusion detection. However, problems persist when it comes to detecting attacks on the digital environment, and especially within public environments. Indeed, the diversity of attacks, which is in large part linked to the richness of the still emerging application semantics, has considerably increased the number of obstacles to overcome in order to solve once for all these problems.

In this study, we define an intelligent perimeter, based on wise learning, to validate our approach. This approach consists in focusing on a single type of attack: APT injections and APT attacks. This kind and category of attacks pose a significant danger and have been the source of major high profile attacks recently, and since the early 2006 with the Stuxnet attack, and others targeted different digital services providers [1]. The choice of perimeter does not limit us to this single type of attack, but encourages us to open up perspectives to other types of attacks specific and targeting the computer networks as a platform of action.

II- APT – ADVANCED PERSISTENT THREATS ATTACK MODELS

a. Layered Security Architecture

The layered model must meet the following criteria in order to be effective:

- Only the processes and applications of the immediately outermost layer can gain access.
- To get around the controls and gain access to a layer, the attacker must start with the outermost layer and execute a kill chain.
- The chances of discovering common weaknesses in the filters used to protect the various layers must be extremely

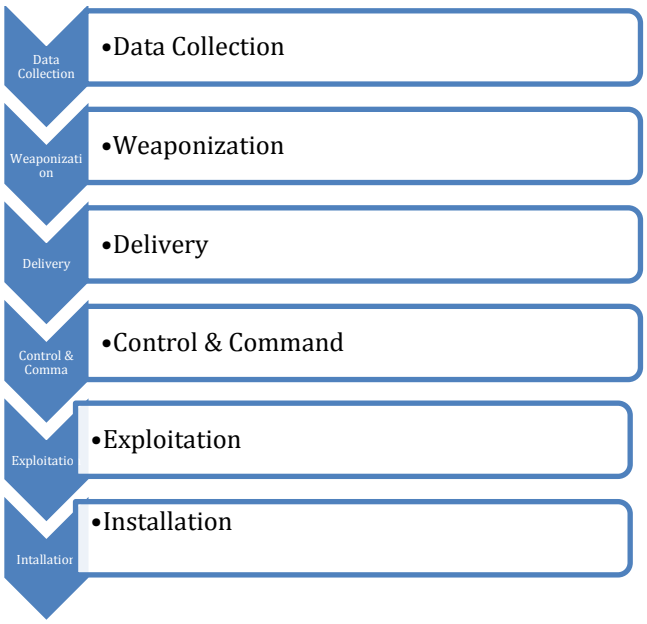
slim. The goal is to reduce the amount of information about a layer's vulnerabilities that is reused to target another layer. The protection will stymie the threat, forcing the attackers to gather more data and build new weapons to get around each layer.

b. Multi-stage Attack Model

The malware kill stages are summarized by the Intrusion Kill Chains (IKC) phases are as follows [3]:

- Data collection – selecting targets, data collection on the target, tools used by the target, possible vulnerabilities, and so on.
- Weaponization – creating malicious code to test found vulnerabilities and combining it with unknown deliverable payloads such as pdfs, docs, and ppts.
- Command and control (CC) - In order to control the malware and continue their behavior, the adversary needs a communication channel. As a result, it must be linked to a CC server.
- Activities – this is the final step of the kill chain, in which the enemy accomplishes its goals by actions such as data exfiltration. Defenders should be assured that the adversary can reach this step after completing the previous phases.

Figure 1: Intrusion Kill Chain (IKC)



c. A Security Event Collection and Analysis System

The defense of a layer must be able to detect an IKC identifying one of its seven stages. For each layer, it is elaborated a defense plan, as illustrated in Table I. A defense plan identifies for each phase of an IKC the attack mechanisms that can be used at each stage, and the controls to prevent and detect the attacks. Each row of the table can be understood as a defense line which can prevent or detect a phase of an IKC. The attack mechanisms were extracted from CAPEC list maintained by Mitre [15].

Table 1: Defense Plan

Phase Defense	CAPEC	Prevention	Detection
Line	**Mechanisms**		
Data Gathering	Data Leakage	IPS, Firewall, Proxy	NIDS, NABD
	Footprinting	IPS, Firewall	HIDS, NIDS
	Fingerprinting	IPS, Firewall, Obfuscation	NIDS, NABD
	Social Engineering	Awareness	User Monitoring
	Network Recognition	IPS, Firewall	NIDS, NABD
Delivery	Injection	IPS, Input filtering	NIDS, HIDS
	Action spoofing	Content filtering	Proxy
	Hacking Hardware devices	Configuration control	HIDS
	Supply chain attack	Security life cycle	NIDS, HIDS
	Spear phishing	Identity verification, blacklisting	Source correlation
Weaponization		Patching, Auditing, Vulnerability Scanning	
Actions	Resource Depletion	IPS	HIDS, NIDS
	Exploitation of Privilege/Trust	Password Control, Firewall	HIDS
	Resource Manipulation	Firewall, Proxy, Encryption Use Control	HIDS, NIDS
C2		Firewall, Proxy, Encryption Use Control, blacklisting	HIDS, NIDS, Content Analysis
Execution	Privilege Escalation	Password Control, Firewall	HIDS
Exploit	Data Structure Attack	Patching	HIDS

IPS – Intrusion Prevention System
NIDS – Network Intrusion Detection System
NABD- Network Anomaly Behavior Detection
HIDS-Host Intrusion Detection System

III- STATISTICAL FACTORS

The detection of anomalies is assessed by analyzing the anomalies that have been correctly or wrongly detected as anomalies or as normal behavior.

The approaches that interest us for the detection

of anomalies are classification as well as clustering. The classification in particular is evaluated by checking in advance whether the classes assigned to the observations are the correct ones. Clustering, for its part, is not developed to assign semantic meaning to the identified clusters (anomaly cluster or not). Two scenarios therefore arise for the performance evaluation of clustering. The first case is that for which the observations which are not assigned to clusters are considered as anomalies. The second case is to use a classification procedure after processing the clusters to identify whether or not a cluster is an anomaly. In the field of machine learning, a classification procedure learns and predicts so called positive or negative labels. Behaviors that are qualified in our work as Negatives are the normal behaviors and Positive for cases of anomaly. The expected results and the results obtained by the detection (supervised or unsupervised learning) are then organized in a confusion matrix, which is illustrated in table 1, composed of the following main metrics:

- True Negative (TN): normal behavior predicted as such.
- True Positive (TP): anomaly predicted as such.
- False Positive (FP): normal behavior predicted as an anomaly.
- False Negative (FN): anomaly predicted as normal behavior.
- Positives: total number of anomalies (P = TP + FN)
- Negative: total number of normal behaviors (N = TN + FP)

Several metrics can be deduced from these main metrics. Among those that are used in our field we find accuracy, precision, recall also called true positive rate (TPR) and false positive rate (FPR).

- Accuracy is the proportion of correctly predicted observations.
- Precision represents the probability that an observation classified as Positive is indeed Positive.

It therefore corresponds to the total number of True Positives returned, out of the total number of observations that were returned as Positives,

wrongly or rightly. The recall meanwhile represents the rate of True Positives. It is therefore the total number of True Positives returned, divided by the total number of Positives that the approach should have recognized. The definition of these measures is given in table 2.

Precision	$\frac{TP}{TP + FP}$
Reminder (or TPR)	$\frac{TP}{TP + FN}$
FPR	$\frac{FP}{FP + TN}$

These measurements are used in particular in various detection work [8, 9, 10, 11, 12].

The Receiver Operating Characteristic (ROC) [3] and precision-recall (PR) curves are often used to summarize these metrics for several prediction thresholds (as we can see it demonstrated and proven in [4, 5, 6]). In our context, we consider that the study of ROC curves is necessary but not sufficient to conclude detection performance. We want to maximize the rate of true positives and minimize the rate of false positives while observing whether the precision and associated recall remain high.

Table 2: Confusion matrix.

	Real Positives	**Real Negatives**
Positives Predicted	TP	FP
Negative Predicted	FN	TN

Table 3: Definition of performance measures.

Measure	**Formula**
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$

IV- DATA TYPES FOR AN APT DETECTION

Anomaly detection is usually defined by the action of discriminating in a set of data made up of observations (ie the rows) and attributes (ie the columns) not known in advance characterizing a target system, the observations which do not correspond to the global trend represented by the majority of observations [7]. The difficulty of the task is to identify precisely this global trend.

Such methods differ from each other in particular by the data processed to detect anomalies as well as the detection algorithms used.

Different types of data

The data to be processed for the detection of anomalies in a computer system can be of three main types.

First, we find historical logs of events (or logs), systems or applications as used for the detection of work anomalies [8, 9, 10, 11]. System logs provide a global view of a machine. Their analysis depends on the machine's operating system.

Second, the detection can also be done from audit trails which group together high level events corresponding to exchanges on a network or to actions performed by users, as used for the detection of work anomalies [2, 3, 4, 5].

Third, some detection methods deal with application usage statistics or system performance observations. Application usage statistics are, for example, system performance indicators used by this application or counters (resource counters, errors, etc.).

Thus, data sets to be the feed for the learning process are generated from:

- journal logs
- traceability audits
- systems monitoring statistics

V- LEARNING-BASED TECHNIQUES AS A DETECTION METHODOLOGIES

The detection of anomalies must therefore distinguish an observation corresponding to an anomaly (i.e. an observation that was made while the system was undergoing an anomaly) or an observation of normal behavior. When anomaly detection is implemented by machine

learning techniques, algorithms are used to perform this discrimination, based on previous observations. It is called prediction [6].

Where a machine-learning algorithm predicts the category of a new observation (i.e. it has never been processed before), based on previous observations that have already been processed.

The category of an observation, for example the normal behavior category and the anomaly category, is assigned to an observation called a *label* (often a whole number). The prediction made by the algorithm can be confirmed or invalidated in advance by an operator.

In the case where a model can be built from labeled observations in order to discriminate the observations of different labels, we speak of supervised classification of data [7]. And this classification would lead us to the anomalies detection phase.

VI- EXPERIMENTAL DETECTION PERFORMANCE EVALUATION

The evaluation of detection techniques leads to the use of two main classes of data: data resulting from operational exploitation including errors reported by users, and data resulting from controlled experiments in which errors are deliberately caused in an observed target system. The second case is the one that interests us in our work; we are therefore interested in how to reproduce these on a controlled environment test platform. Such a task is called fault injection. This method makes it possible to inject faults into a system while observing it during the injection (the dates of the start of injection being prepared in an injection protocol). Here, the detection of anomalies is assessed by analyzing the anomalies that have been correctly or wrongly detected as anomalies or as normal behavior. Therefore, the approaches that interest us for the detection of anomalies are classification as well as clustering.

VII- IMPLEMENTATION & STRATEGY:

We consider the analysis of files and network services from the perspective of a provider offering various services to customers. Our goal is to enable such a provider to detect anomalies in their work files and network services. Discovery should be automatic, be

done online, also adapt to changes in workload, and not be dependent on the provider's function or type of service implementation.

The strategy here considers an anomaly detection target system. The system is potentially distributed over several machines; *computers and network equipment*.

Regarding the database, considering each service one by one, as well as the files by extension, is necessary. Since a set of services contributing to the same function or to different functions is too heterogeneous to be able to aggregate the data into intelligible information [7, 11, 13].

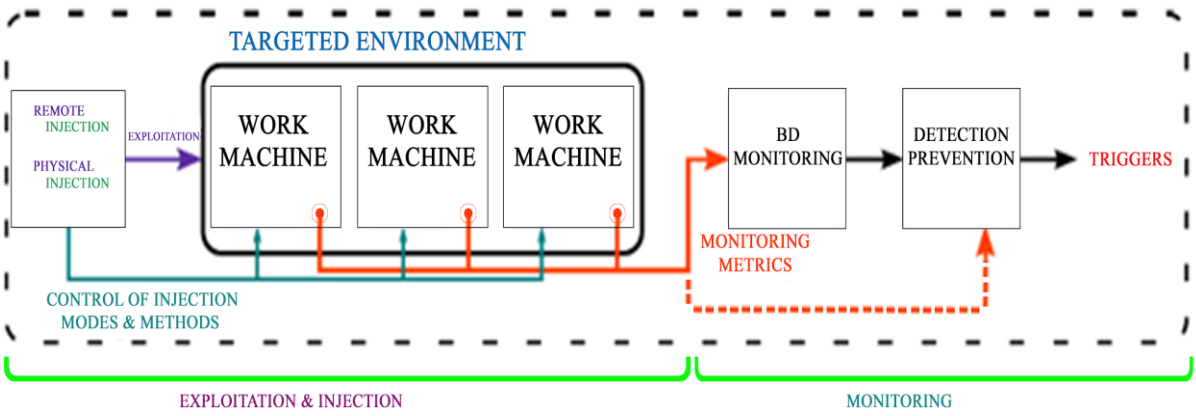


Figure2: APT Detection set up Diagram

Due to the fast pace changing capabilities of the APTs, the benign decision making engine would re-scan and re-classify the already set benign processes in a second round after doing a round-trip. This way the complexity of the APTs would be revealed and if an APTs has succeeded to trespass the initial Macro-Level classifier, it would be detected within the second scan trip and be classified as infected.

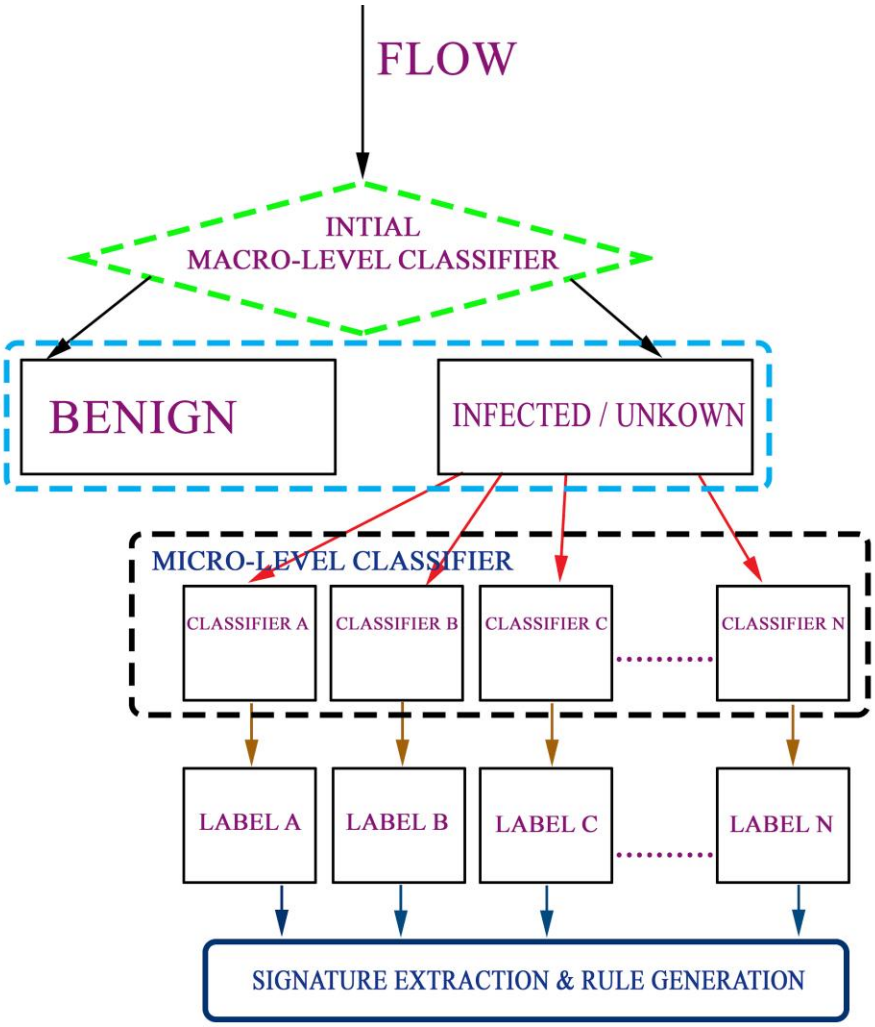


Figure3: Flow Chart

VIII- CONCLUSION

In this scientific study, we conclude that in order to achieve a high quality of anomaly detection, we determine the classification as well as the clustering as the methodology main objectives. Thus, the classification in particular is evaluated by checking in later times whether the classes assigned to the observations are the correct ones. Clustering, on the other hand, is not developed to

assign a semantic meaning to the identified clusters (anomaly cluster or not). Yet, two scenarios therefore arise and they are, also, utilized for the performance evaluation of clustering. The first case is that for which the observations which are not assigned to clusters are considered as anomalies. The second case is to use a classification procedure after processing the clusters to identify whether or not a cluster is an anomaly.

REFERENCES:

1. Bailey, M., Cooke, E., Jahanian, F., Xu, Y., & Karir, M. (2009). A survey of botnet and botnet detection. *Conference For Homeland Security, CATCH'09, Cybersecurity Applications \& Technology* (pp. 268-273). IEEE.

2. Bhatt, P., Toshiro Yano, E., & Gustavsson, P. M. (2014). Towards a Framework to Detect Multi-stage Advanced Persistent Threats Attacks. *IEEE International Symposium on Service Oriented System Engineering (SOSE)* (pp. 390-395). IEEE.

3. Hutchins Eric M., Cloppert Michael J., Amin Rohan M,"IntelligenceDriven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains" ICIW2011

4. Bilge, L., Balzarotti, D., Robertson, W., Kirda, E., & Kruegel, C. (2012). Disclosure: detecting botnet command and control servers through large-scale netflow analysis. *Proceedings of the 28th ACM Annual Computer Security Applications Conference* (pp. 129-138). ACM.

5. Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

6. Brewer, R. (2014). Advanced persistent threats: minimising the damage. *Network Security*, (pp. 5-9).

7. Brockwell, P. J., & Davis, R. A. (2013). *Time series: theory and methods*. Springer Science & Business Media.

8. Canali, C., Casolari, S., & Lancellotti, R. (2010). A quantitative methodology to identify relevant users in social networks. *IEEE International Workshop on Business Applications of Social Network Analysis (BASNA)*, (pp. 1-8).

9. Casolari, S., Tosi, S., & Lo Presti, F. (2012). An adaptive model for online detection of relevant state changes in Internet-based systems. *Performance Evaluation*, (pp. 206-226).

10. Chari, S., Habeck, T., Molloy, I., Park, Y., & Teiken, W. (2013). A bigData platform for analytics on access control policies and logs. *Proceedings of the 18th ACM symposium on Access control models and technologies (SACMAT '13)*.

11. Data Breaches. (2020, July). Retrieved from World most popular data breaches, Information is beautiful: http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks.

12. De Vries, J., Hoogstraaten, H., van den Berg, J., & Daskapan, S. (2012). Systems for Detecting Advanced Persistent Threats: A Development Roadmap Using Intelligent Data Analysis. *IEEE International Conference on Cyber Security (CyberSecurity)*, (pp. 54-61).

13. Denning, D. E. (1987). An intrusion-detection model. *Software Engineering, IEEE Transactions on*, (pp. 222-232).

14. Duffield, N. G., & Lo Presti, F. (2009). Multicast inference of packet delay variance at interior network links. *IEEE Computer and Communications Societies*., (pp. 280-285).

15. CAPEC – Common Attack Pattern Enumeration and Classification online Mechanism of Attack at http://capec.mitre.org/data/definitions/1000.html [accessed 1 Jan 2014]

16. Friedberg, I., Skopik, F., Settanni, G., & Fiedler, R. (2015). Combating advanced persistent threats: from network event correlation to incident detection. *Computers & Security*, (pp. 35-57).