International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V70I4P208
Volume 70 Issue 4, 95-106, April 2022

Original Article

A Significant Detection of APT using MD5 Hash Signature and Machine Learning Approach

R C. Veena[1], S H. Brahmananda[2]

[1,2]*Department of Computer Science and Engineering, GITAM University, Bengaluru, Karnataka, India.*

[1]vchalapa@gitam.edu

Received: 15 February 2022 Revised: 28 March 2022 Accepted: 30 March 2022 Published: 25 April 2022

Abstract *- The overwhelming penetration of the internet has created day-to-day life easy. Associated with the rich benefits of the internet come new threats and challenges. An Advanced Persistent Threat (APT) is one such threat caused by suspicious agents accessing data or surveillance servers over a prolonged period. APT attacks have been using a variety of specialized tools and techniques. APT hackers and malware are more common and improvised than ever. Attackers have previously aimed at a system for financial and personal benefit. The type of attack includes several other political motives supported by governments or nations. Nations like the United States, India, Russia, and the U.K. are sufferers. APT involves several stages and a definite approach to operational strategy. Besides, techniques and technologies used in APT attacks vary to camouflage the surveillance applications and penetrate unsuspecting networks. This work presents a Machine Learning (ML) Algorithm-based APT Attacks detection framework. MD5 is even more hazardous than previously thought in cryptography techniques. Attackers can impersonate clients to servers that support MD5 hashing for handshake transcripts. The proposed detection framework resulted in highly effective detection of APT attacks at the initial stage based on the MD5 signature using the ML approach. More than 50% of antivirus software has validated the identified MD5 signature as malicious. This detection framework prevents APTs from fast-spreading from compromising a single computer to taking over several systems or the complete infrastructure. The developed system got trained with 76 types of APT signatures. The total number of threats variant used for training is 645. The proposed ML framework has an accuracy of 99% compared to the published accuracy of 96.1% [23] for early detection of APT from an unknown domain.*

Keywords *- APT, MD5 Hashing, Network Security, Hackers, Machine Learning, Threat Hunting.*

1. Introduction

Increased cloud adoption poses a greater security risk to your I.T. and business users. Cybersecurity is one of the critical areas in any enterprise I.T. infrastructure. The APT threat is one of the important attack types where a cybersecurity consultant has primary responsibility for preventing the attack. While implementing a security solution, the key elements that meet the infrastructure needs must be analyzed. The use of ML-based implementation of solutions to maintain cybersecurity is growing. There is a need to evaluate existing and new technologies to determine their applicability and value for inclusion in security solutions. Some of the key aspects of the process are host and network security solutions, network performance monitoring, overseeing any required modification or reconfiguration of network elements, and ongoing technical research to meet security solution requirements.

Most cloud breaches are due to compromised credentials. The adoption requires ensuring that the users have reliable but secure access to cloud services and applications. The users include remote employees, third parties, and contractors. The primary challenge in cloud adoption is to solve access security challenges unique to the cloud. Ensure and secure remote worker and third-party access to cloud resources. APTs create a threat to information technology infrastructure. During the attack, the procedure that cybercriminals use is rapidly changing. Many enterprises depend on old cybersecurity measures and an averse to changing approaches to prevent cyberattacks.

APT hackers and malware are more prevalent and sophisticated than ever. APTs fast grow from attacking a single user to the whole network in just a few hours. APT hackers push backdoor Trojan malware on the attacked system within the compromised environment or phishing email. Since an internet-based connected worldwide network, cybercriminals leverage both known and unknown attacks. They are encouraged to use their learning to commit malicious attacks such as defacing an enterprise website, unauthorized access to institutions of national importance and stealing terabytes of classified data.

Broadly following are the classification of attackers behind an APT:

1.1 Cybercriminals

They are usually a software developer with cyber tools or use tools obtained on the dark web.

1.2 Business Competitor

They access business information from enterprise networks using unauthorized ways for creating sabotage.

1.3 Cyber-Mercenaries

They develop tools and offer them to the highest bidder.

1.4 Hacktivists

They use complex toolsets and causes serious security threat to an organization.

1.5 Government Agencies

Using sophisticated, expensive, and hard to detect for spy activities.

A framework for analyzing cyber threats is developed based on proper labelling and understanding of attack vectors, discerning an attacker's motivation. Sound principles of cyber security are required to close vectors or determine who is behind the incidences. Assist cybersecurity management in delivering information security services, awareness & training, and other associated tasks aligned with information security strategy, policies, and technology requirements. The key driver is to conduct information security & cybersecurity risk assessment to support, enhance and refine information security policies and controls to enable business without compromising information security.

The framework requires countermeasures and other secure channels to address data leakage and develop technical solutions to enable secure data sharing and collaboration requirements. The cloud security aspect of the framework and SaaS (Solution as a Service) controls review to enable I.T. infrastructure with the right security controls. Further challenges in this regard are assessing data protection exception requests, recommending appropriate actions in creating and maintaining internal incident communication plans, execute, and escalating breach responses whenever required. The framework should have the ability to conduct information security & cybersecurity risk assessment to support, enhance and refine information security policies and controls to enable an enterprise to do a transaction without compromising information security.

Malicious File Hash Detection (MFHD) is a recent method of APT detection. The proposed approach is to use MD5 hashing to detect the suspicious pattern. The research found that authentication and impersonation attacks are prevalent for protocols that still use MD5 in communication. The MD5 hashing function continues in some parts of encrypted communications protocols, including TLS causing a potential threat to security.

To start with the proposed work, a detailed literature survey is conducted to assess the state-of-the-art technologies available and the gap and issues.

2. Literature Survey

The literature survey refers to the book Code E. to understand the danger and how to protect from APT. The APT is often performed by an organized team, including a foreign country or criminal group, with the ability and desire to repeatedly and effectively target a certain organization and inflict damage. This is the first thorough method for understanding when hackers get into organizations and what could be done to protect and defend from attacks. [1]. Hyunjoo et al. presented behaviour-based anomaly detection on big data. Using Massive Storage and computational technologies, the proposed approach analyses various log data and tracking information more quickly and accurately. They found that utilizing MapReduce to examine large-scale behaviours by monitoring and logging data from multiple sources is successful in detecting fraudulent behaviour [2].

Luh et al. developed a method for describing suspicious behaviour inside a user session by considering the effect abnormalities discovered by comparing them to a collection of baseline process graphs. This entails a clever anomaly explanation utilizing a decision tree algorithm to generate and assess various competence challenges [3]. Mees W. presented an approach for detecting command & control channels between malware. They used a server and a multi-agent aggregation of evidence method. Their work combines inputs from the anomaly and signature-based techniques. Two anomaly-based agents, one for detecting suspicious HTTP transactions and one for detecting suspicious DNS requests [4]

APT is usually deployed in a series of phases and steps. The complete APT assault will fail if one of the phases or stages fails. The strategy presented in this study for identifying APT assaults is dependent on monitoring access to unexpected domains. This identification system is quite successful in the early stages of an APT assault [5]. As part of an APT assault, spear-phishing is currently being utilized to breach devices and provide an entry point into the system. With the latest reports indicating 91 per cent of APT assaults start with spear-phishing emails and that computers and hackers are frequently targeted smartphones utilizing sensitive information collected from social networking sites, it's obvious that the threat landscape has evolved [6]. Xiaohua et al. worked on SID as an effective early intrusion detection system based on defined modelling of cyberattack activity to identify underlying high-level behavioural patterns in internet traffic that are considered initial signals of cyber-attacks. The created system can identify attacks in their

initial stages, enabling defensive steps before genuine intrusions occur. The results indicate that utilizing high-level behaviour patterns to forecast attacks performs significantly better than limited internet traffic analyses [7].

Zimba et al. Proposed The APT exfiltration stages are linked to a cyber kill chain in a multi-phase transferrable Markov process. To create the simplest APT attack vectors, they use Bayesian networks. To define numerous threat tracks from the source to the target node, they employed quantitative inductive to collectively illustrate graded threat routes and linked margin and conditioned probability. The chance of an APT occurring in a specific track was calculated as part of the methodology. Moreover, an efficient approach for calculating the smallest treat route from multiple sources based on important nodes and key edges was developed. Limitation: Detection of APT attack is not included [8]. In their study, Lajevardi et al. utilized a low-level intercepting to link os activities to network operations using the conceptual connections specified between the elements in systems ontologies. The ontology's wholeness and accuracy are essential to the presented approach's effectiveness. This approach recognizes harmful events, particularly those that indirectly break security regulations, on incident relations and established security procedures. In addition, utilizing a memory transition/manipulation model to rebuild dispersed attack vectors [9].

Yan et al. presented a paradigm for describing an information-based APT attack on a corporate network. The early entry paradigm for identifying access places and the targeted attack concept for examining information collection, strategic decision-making, weaponization, and lateral movement are part of their mathematical model. The researchers used simulations to find the best potential nodes in the first entrance model, examine the targeted attack framework altered dynamically, and verify the APT assault's properties [10]. In another study, the authors discussed ML techniques. They find models commonly used to identify an APT threat are support vector machine (SVM), k-NN, and D.T. The life cycles of a threat were analyzed using various phases from these cycles. The framework effectiveness was measured using an APT attack simulation in a controlled space. The validation of the proposed methods with real traffic is significant to the current research paper [11].

Chu et al. discussed several network intrusion detection systems using the SVM algorithms. However, they are very greedy in terms of I.T. infrastructure performance. A dimension reduction using a given data set to minimize the problem and feature extraction is used for improving processing speed. Enhancement in the feature information of the data set availed to analyze the APT threat and the dimensionality reduction method [12].

A review of APT approaches, methods, limitations, and research prospects was published by Alshamrani A. et al. The researchers of this research study try to bring together all of the strategies and processes that can be used to identify various phases of APT attacks as well as the learning techniques that should be used and how to make the threat detection framework smart and undecipherable for those adapting APT attackers [13]. Li Z., Chen Q. A. proposed a hierarchical strategy for APT identification using innovative attention-based Graph Neural Networks to address the limitations of traditional systems (GNNs). Authors developed a meta route aggregated GNN for provenance graph embedding and an edge improved GNN for host interaction graph embedding, allowing for capturing APT behaviours at both the system and network levels. In addition, a unique improvement strategy for dynamically updating the detection algorithm in the hierarchical detection method is presented. The results suggest that the proposed technique exceeds existing thresholds for detecting APT [14].

Commonly used causal graph generation approaches are primarily offline. Generally, they take a long duration to reply to investigator queries, leading to a room for malicious threats to camouflage the threat signature, gain persistence and spread to other systems. To get rid of the issue of slow reply, Wajih et al. presented Swift, a threat analysis platform with real-time causal graph creation and high-throughput causality tracking capabilities. Swift is extensible, adaptable, and responds to forensics inquiries on a real-time basis, also when reviewing audit records with millions and millions of incidents, as per the study results [15].

Q. Wang et al. presented a PROVDETECTOR, Detecting stealthy malware using a provenance-based method. According to the concept-driven development, although stealthy malware seeks to blend in with innocuous processes, its harmful activities unavoidably interact with the underlying operating system (O.S.). This can be detected by provenance monitoring. Results based on a large provenance dataset demonstrated that it achieved a high detection performance of stealthy malware with an average F1 score of 0.974 [16].

M. N. Hossain, S. Sheikhi, R. Sekar, and C. concentrated on two unique approaches to mitigate dependency explosion: tag attenuation and tag decay. The new technique intends to take advantage of general tendencies of benign activities while also broadening cautious handling of processes and systems of suspect origins. By filtering through billions of event logs in seconds, the new MORSE system can create a compact scenario graph that describes attacker behaviour [17].

The concept of universal provenance was suggested by W. U. Hassan et al., which incorporates every forensically important causal relationship regardless of the level of provenance. OmegaLog, the proposed system, is a

provenance tracker that connects the semantics of the system and application log context. OmegaLog provides succinct provenance graphs with rich semantic data with an estimated runtime overhead of 4% compared to the state-of-the-art in real-world attack situations [18].

Using AST characteristics and paragraph vectors, S. Ndichu, S. Kim, S. Ozawa, T. Misu, and K. Makishima proposed an ML technique to detect JavaScript-based assaults. The study used an ML approach called Doc2vec to execute feature learning and an Abstract Syntax Tree (AST) for code structure representation. Compared to frequently available methodologies, the experimental findings imply that the new AST features and Doc2Vec give improved performance and rapid classification in harmful J.S. code detection and can flag dangerous J.S. codes previously recognized as difficult. [19Z. Li et al. developed the first semantic-aware PowerShell threat detection mechanism using the novel deobfuscation technology. Using the conventional objective-oriented association mining approach, the authors discovered 31 new semantic signatures for PowerShell attacks. On average, 92.3 per cent of true positives and 0 per cent of false positives were attained [20]. H. Wang et al. presented an evolutionary study on malware attacking devices. They used the information from two sources that complement each other such as online articles and real malware samples. A final lineage graph for 72 malware families was constructed by correlation [21] based on the data.

DAMBA, a revolutionary prototype system based on a C/S architecture, was developed by W. Zhang, H. Wang, H. He, and P. Liu. Applications' dynamic and static characteristics are extracted using DAMBA. The TANMAD algorithm, a two-step Android malware detection technique, is also employed. The modelling of object reference information by generating directed graphs, referred to as object reference graph birthmarks, is a novel notion in this article (ORGB). DAMBA surpasses McAfee, a well-known detector based on signature recognition, according to test findings. DAMBA has also been shown to effectively resist known malicious assaults and variations and malware that use obfuscation [22].

In their article, H. S. Ham et al. proposed a linear SVM to identify malware targeting Android O.S. and performance compared to that of other ML classifiers. The author studied the current domestic pattern of malware attacking Androids for this, and 14 malware packages were chosen to be applied to the built SVM. The practical findings suggest that the suggested SVM outperforms other machine learning classifiers [23]. A. Calleja et al. published a fascinating article in which the researchers looked into the rise of malware from 1975 to the present. This entails analyzing 456 samples from 428 families to determine their size, code quality, and development expenses. In terms of size and anticipated effort, the results demonstrate an exponential increase of roughly one order of magnitude every decade, with code quality criteria equal to benign software. Finally, the findings back with allegations that malware is growing more complicated and that its development is increasingly becoming a business [24].

3. Md5 Threat

Scientists from the INRIA institute in France have engineered many attacks to establish that the ongoing support for MD5 in cryptographic protocols is a high threat than earlier belief. Hijacking and redirecting user signatures through protocols that use a TLS channel binding mechanism is possible [25]. MD5 signatures have been risky and susceptible to threat and practical collisions since at least 2005. Till TLS 1.1, the transcript hash was generated using a combination of MD5 and SHA1, but in TLS 1.2, this is negotiable. Although TLS 1.2 allows a stronger hash function like SHA-256 and SHA-512 still supports MD5. The study says 30% of HTTPS servers still use the MD5 server signature. This is a high risk for intrusion and potential APT [26-27]. Figure 1 shows the malicious MD5 Hash signature sample.

47870ff98164155f088062c95c448783
2c1e73da56f4da619c4c53b521404874
6acf316fed472300fa50db54fa6f3cbc
9573f452004b16eabd20fa65a6c2c1c4
3772a34d1b731697e2879bef54967332
d967d96ea5d0962e08844d140c2874e0
a80bbd753c07512b31ab04bd5e3324c2
37dc2eb8ee56aeba4dbd4cf46f87ae9a

Fig. 1 MD5 Hash Signature

Identifying cyber-attacks is a continuous research topic. Companies and organizations might suffer enormous expenditures as a result of APIs. Before being terminated, each cyberattack passes through numerous stages. This research looked at several feature sets collected from logs and compared machine learning algorithms for accurately categorizing threatening events.ML uses computer algorithms to improve automated decisions through learning and data. In the present work, an ML implementation motivated by Ahuja, R. et al. [29] is proposed, as shown in Fig. 2, to detect malicious APT using the MD5 signature.

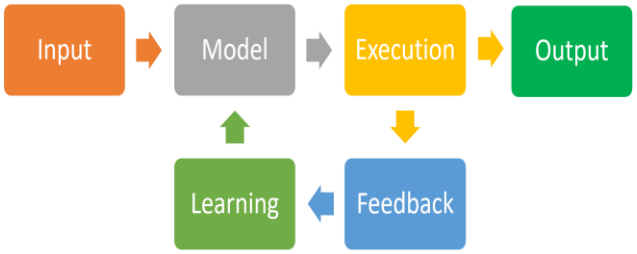


Fig. 2 Machine Learning for APT Detection

The steps in the execution of the proposed ML-based techniques involve:

- **Model** – Develop an ML model based on available APT signatures.
- **Execution** – Train the developed model based on training data
- **Feedback** – Run the model and seek feedback based on test data
- **Learning** – Capture the feedback to refine the model.

The proposed approach is to build a knowledge base using MD5 signatures from the cloud using a Python-based application. The proposed MD5 model then uses the updated database to validate a new authentication request from an unknown requester. The algorithm is shown in Algorithm 1.

Algorithm 1 APT detection using MD5 Sign

For Each Authentication Request
Check if *negotiation* involves MD5 Sign
If *negotiation* is Yes, Then
Then invoke ML
Else
Pass on to the next level of validation for other negotiation protocols.
End
For Each MD5 Signature
Check *existence* in the database.
If *existence* is Yes, Then
Send Alarm with details & history of attacks.
Else
Search in the cloud for *existence*
If *Search* is True, then
Send Alarm with details found
Else
Send Alarm stating "Unknown."
Decline Request
End
End
Update History and Status information in the database.
End

4. Proposed Design

Table 1 shows the Malicious threats used to train the ML algorithm. Each of the attack types has further variants in terms of MD5 signatures.

Table 1. APT data used for ML training

Apt_id	Apt Name	First known	Apt_id	Apt Name	First Known	Apt_id	Apt Name	First Known
Apt_01	Topinambour	2019	Apt_28	Animal farm	2007	Apt_55	Duqu 2.0	2014
Apt_02	Tajmahal	2013	Apt_29	Kimsuky	2011	Apt_56	Hellsing	2012
Apt_03	Sneakypastes	2018	Apt_30	Crouching Yeti	2010	Apt_57	Lazarus	2009
Apt_04	Octopus	1990	Apt_31	Cosmicduke	2012	Apt_58	Project Sauron	2011
Apt_05	Fruityarmor	2018	Apt_32	Black energy	2010	Apt_59	Carbanak 2.0	2015
Apt_06	Muddy water	2017	Apt_33	Desert falcons	2011	Apt_60	Dropping elephant	2016
Apt_07	Olympic destroyer	2017	Apt_34	Hacking team rcs	2008	Apt_61	Saguaro	2009
Apt_08	Zoopark	2015	Apt_35	Nettraveler	2004	Apt_62	Strongpity	2016
Apt_09	Whitebear	2016	Apt_36	Miniduke	2008	Apt_63	Stonedrill	2016
Apt_10	Skygofree	2014	Apt_37	Equation	2002	Apt_64	Shamoon 2.0	2016
Apt_11	Shadowpad	2017	Apt_38	Naikon's aria	2009	Apt_65	Bluenoroff	2016

Apt_12	Satellite Turla	2007	Apt_39	Turla	2007	Apt_66	Spring dragon	2012
Apt_13	Penquin Turla	2010	Apt_40	Blue termite	2013	Apt_67	Wannacry	2017
Apt_14	Lamberts	2008	Apt_41	Sofacy	2008	Apt_68	Atmitch	2016
Apt_15	Expetr	2017	Apt_42	Adwind	2012	Apt_69	Blackoasis	2015
Apt_16	Blackoasis	2015	Apt_43	Poseidon	2005	Apt_70	Expetr	2017
Apt_17	Atmitch	2016	Apt_44	Cloud atlas	2014	Apt_71	Lamberts	2008
Apt_18	Wannacry	2017	Apt_45	Carbanak	2013	Apt_72	Penquin Turla	2010
Apt_19	Spring dragon	2012	Apt_46	Regin	2003	Apt_73	Satellite Turla	2007
Apt_20	Bluenoroff	2016	Apt_47	Dark hotel	2007	Apt_74	Shadowpad	2017
Apt_21	Shamoon 2.0	2016	Apt_48	Epic Turla	2012	Apt_75	Skygofree	2014
Apt_22	Stonedrill	2016	Apt_49	Finspy	2007	Apt_76	Whitebear	2016
Apt_23	Strongpity	2016	Apt_50	Miniflame	2010	Apt_77	Zoopark	2015
Apt_24	Saguaro	2009	Apt_51	Winnti	2009			
Apt_25	Dropping elephant	2016	Apt_52	Sabpub	2012			
Apt_26	Carbanak 2.0	2015	Apt_53	Wild neutron	2011			
Apt_27	Project Sauron	2011	Apt_54	Cozyduke	2014			

Fig. 3 the methodology of the implementation of APT detection [30]. ML model is used to learn from the MD5 signatures, and for a new request based on the learning, the ML algorithm generates an alarm for necessary actions [31].

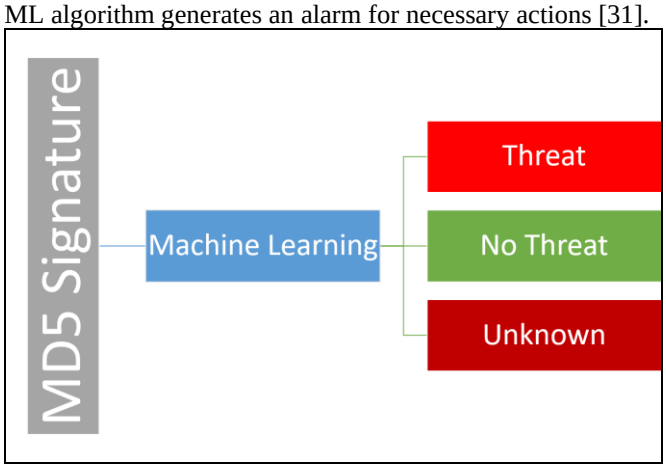


Fig. 3 Methodology of the ML

The most important aspect of using an ML-based approach for detecting APT is the signature of the APT attack. In the present design, a malicious MD5 hash signature is considered the basis for detection. Each type of attack may have several variations in signature. The greater the number of signatures available, the better the detection capability will be. Like APT_01 type has more than 70 hash signatures. So for accurate detection of APT_01, all the 70 types of hash patterns are to be used to train the proposed algorithm.

Fig. 4 shows the APT-wise number of variants. This means that even if the attack may have the same approach to breaching the system, the signatures may vary for the attack [32].

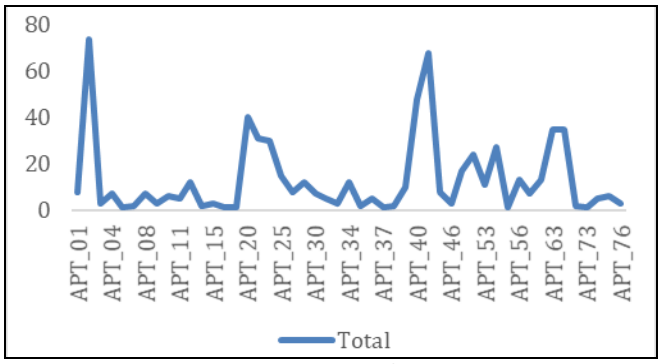


Fig. 4 APT wise variants of training data

5. APT Detection Application

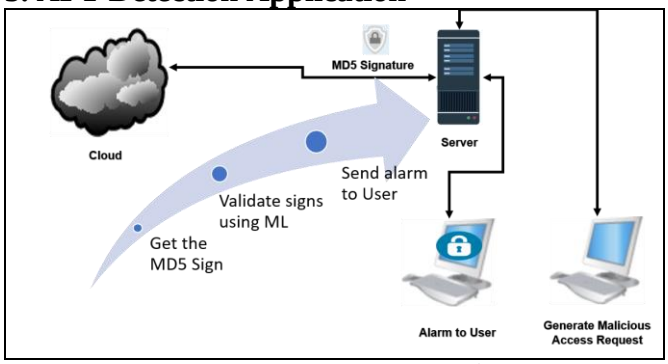


Fig. 5 APT Application Workflow

Fig. 5 shows the designed application workflow. When the server receives a malicious request, ML validates the signature, and as detailed in Algorithm 1, necessary action is initiated. Fig. 6 shows the installation of the Malicious Content Detector (MCD) application developed using python. The installation needs to be before the firewall of the desired network.

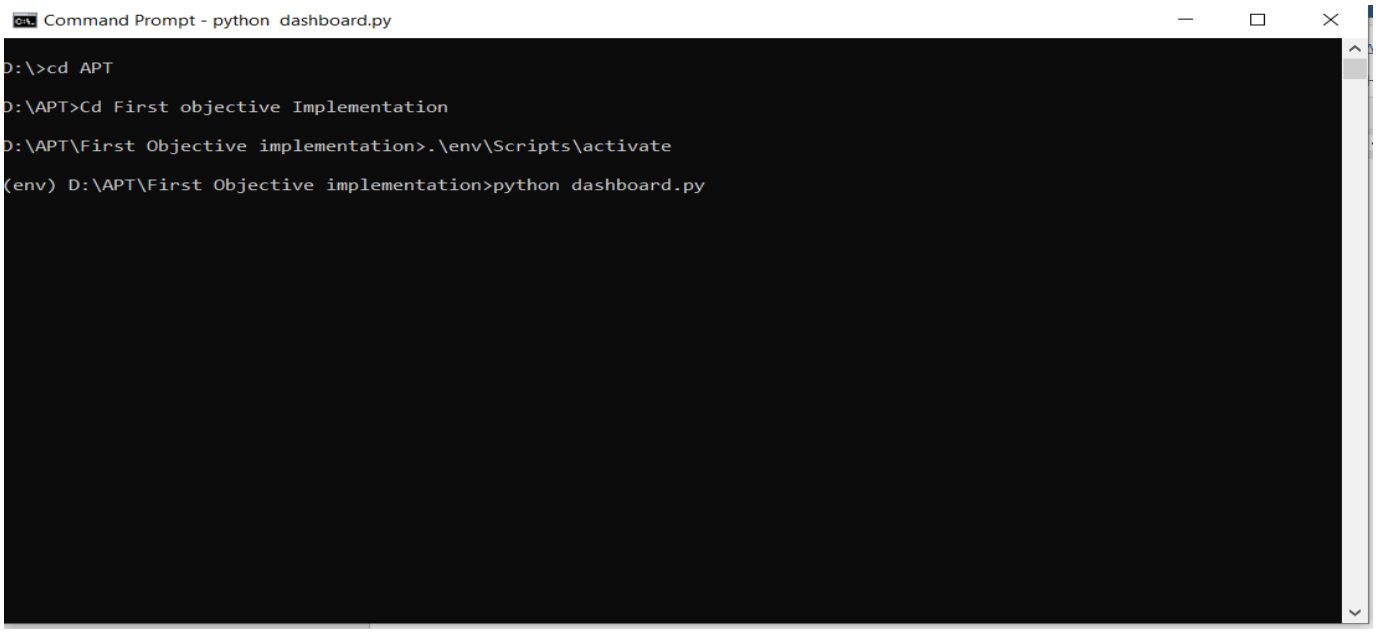


Fig. 6 Installation of MCD

This requires setting up the path and updating the shell instance's local path, the directory within the file system where the shell is presently working. The next step is to create a new environment and activate the shell script. Based on the operating system, the virtual environment gets activated. On executing the python script, the dashboard will be opened in an internet browser. Fig. 7 shows the user interface of the MCD application.

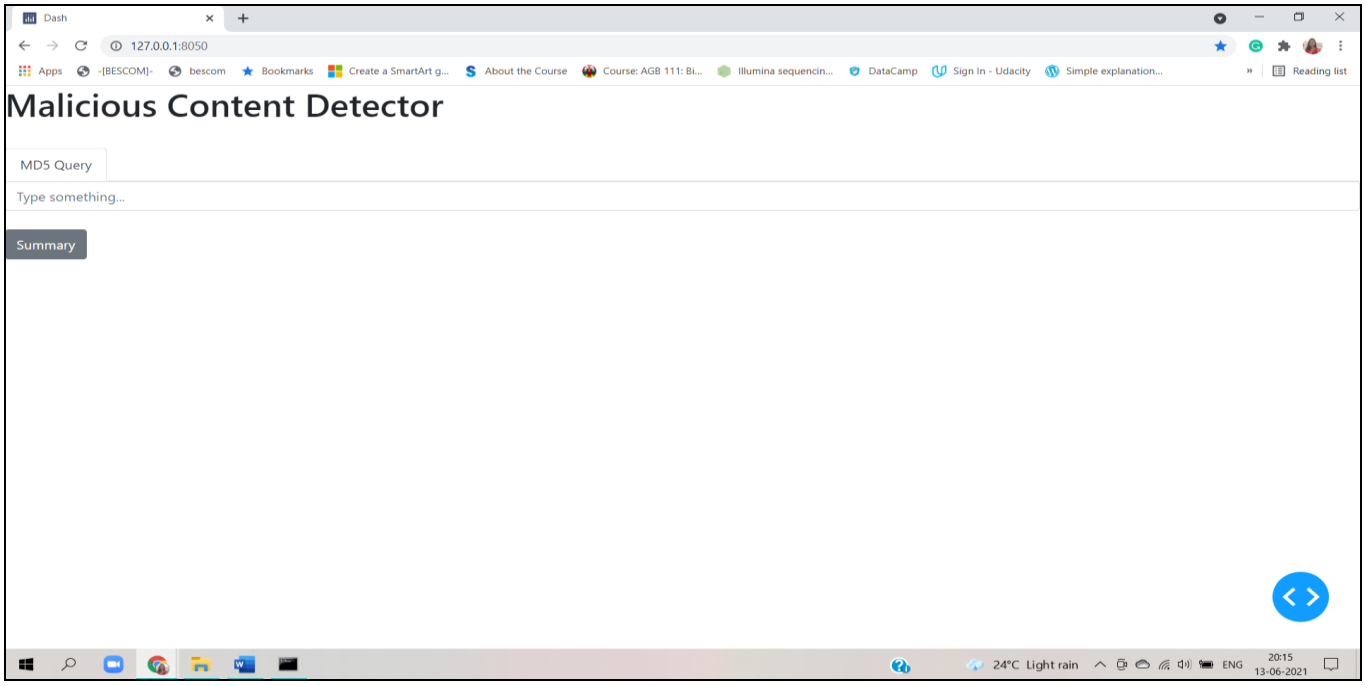


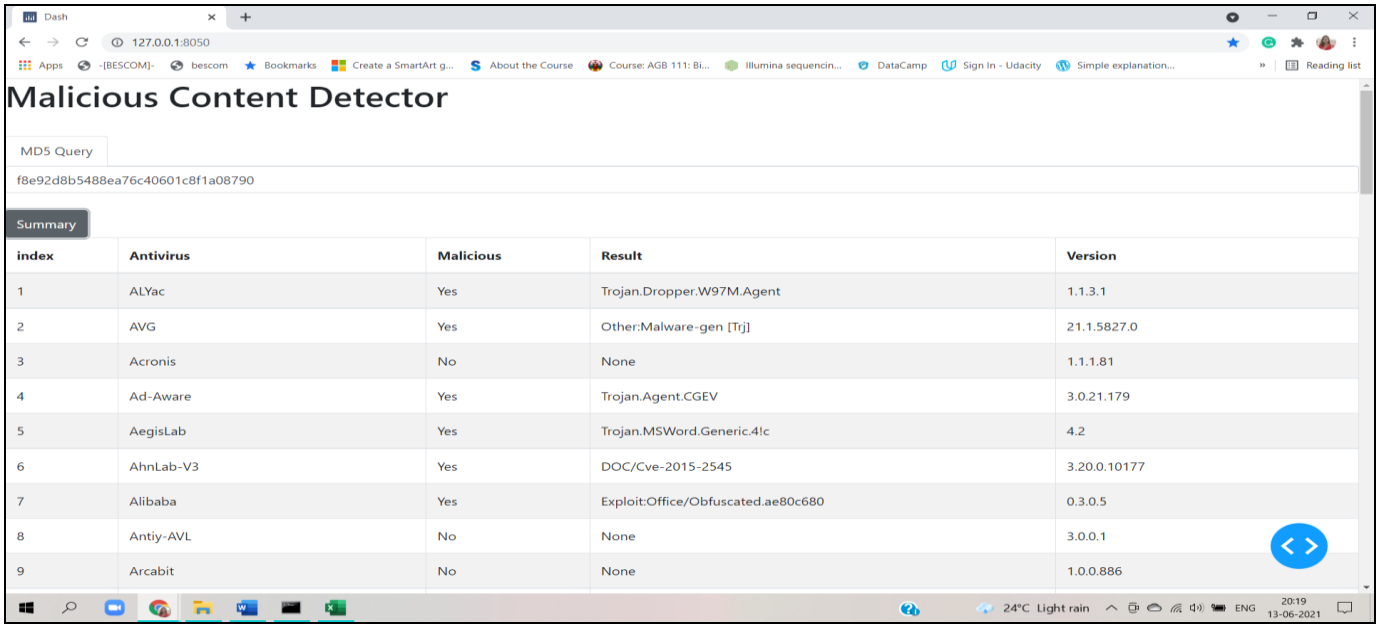
Fig. 7 User Interface for the developed application

To test a pattern, one can enter the MD5 pattern manually. However, the developed application can read from the DNS log [33] from the installed environment. MCD help to find APT threats using values that various APT threats have generated.

6. Results

Suppose the MD5 value entered is an APT threat. In that case, the MCD will provide the details of that threat, including the information provided by different antiviruses along with the type of threat and the version of the threat detected by the antivirus.

As shown in Fig. 8, MCD will provide the details of that threat, including the information provided by different antiviruses along with the type of threat and the version of the threat detected by the antivirus.



index	Antivirus	Malicious	Result	Version
1	ALYac	Yes	Trojan.Dropper.W97M.Agent	1.1.3.1
2	AVG	Yes	Other:Malware-gen [Trj]	21.1.5827.0
3	Acronis	No	None	1.1.1.81
4	Ad-Aware	Yes	Trojan.Agent.CGEV	3.0.21.179
5	AegisLab	Yes	Trojan.MSWord.Generic.4!c	4.2
6	AhnLab-V3	Yes	DOC/Cve-2015-2545	3.20.0.10177
7	Alibaba	Yes	Exploit:Office/Obfuscated.ae80c680	0.3.0.5
8	Antiy-AVL	No	None	3.0.0.1
9	Arcabit	No	None	1.0.0.886

Fig. 8 Threat results based on a pattern

Antivirus software uses a virus signature to find a virus in a computer system, allowing detection, quarantine, and removal of the virus. In antivirus software, the virus signature is a definition file or DAT file. The developed MCD can collect publicly available information from the internet regarding the latest malware, APT signatures. The summary in Fig. 9 shows how many antiviruses detected an MD5 as an APT threat, and also it shows how many total antivirus has detected it as malicious. The developed MCD then uses the collected signatures to detect a new threat in the network.

Fig. 9 Threat history summary

The further result shows how many total antiviruses have identified it as malicious. If more than 50% of antivirus indicates it as malicious, then that particular MD5 value is a malicious threat.

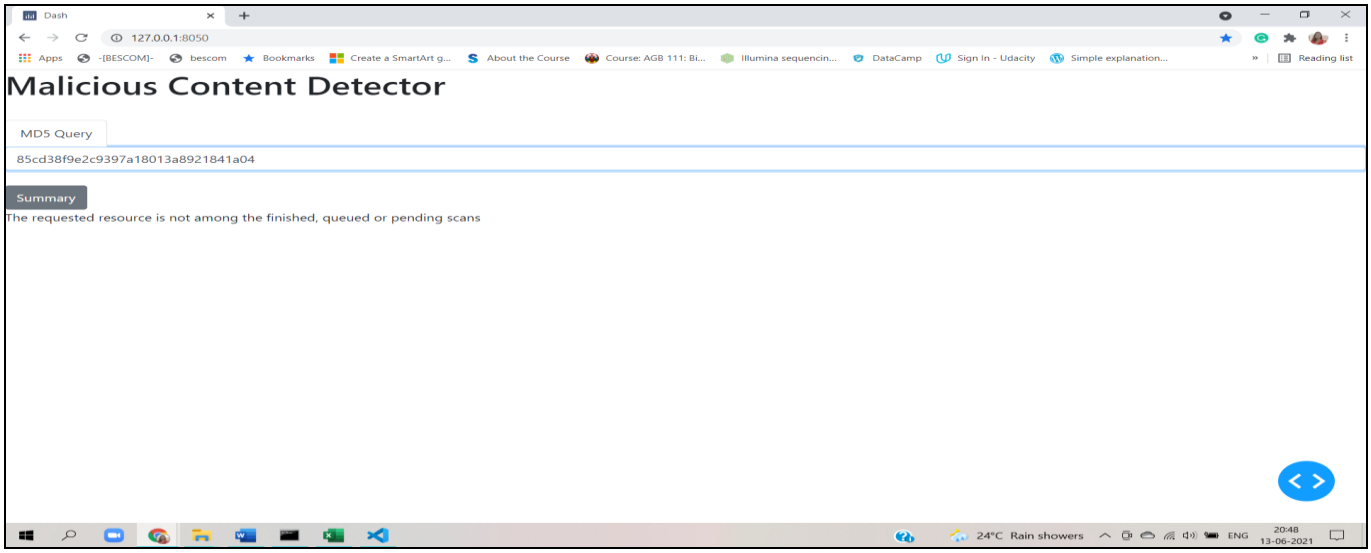


Fig. 10 Nonmalicious request

As shown in Fig. 10, if the signature is not a threat, the alarm message states, "The requested resource is not among the finished, queued or pending scans". The corresponding MD5 signature has not been detected as malicious by any of the antiviruses available over the cloud.

7. Analysis of Result

The author has taken 645 test signatures from public data [34] to test the MCD application with the breakup of the result shown in Table 2. The threat category defined in MCD as “unknown" means not matching with available signatures. Category "Yes" means the threat signature matches the database available with MCD. In the case of “No,” the incident signature is known as a non-APT signature.

Table 2. APT detection result

Threat	Count	%
Unknown	6	1%
No	267	41%
Yes	371	58%
Grand Total	**645**	**100%**

As shown in Table 2, 58% of the MD5 signature tested were malicious, 41% were clean, and 1% could not be identified.

Linear regression analysis is helpful to assess the probability of future threat incidence. Linear regression analysis is used to predict the value of a variable based on the value of another variable. The Linear regression analysis was done based on Equation 1.

$$Y = a + bX \quad (1)$$

Where x is the explanatory variable and y is the dependent variable. The objective is to predict future attacks.

The R-squared goodness-of-fit measure was used for the linear regressions in Equation 2. R-squared evaluates the scatter of the data points around the fitted regression line.

$$R^2 = \frac{\text{Variance explained by the model}}{\text{Total variance}} \quad (2)$$

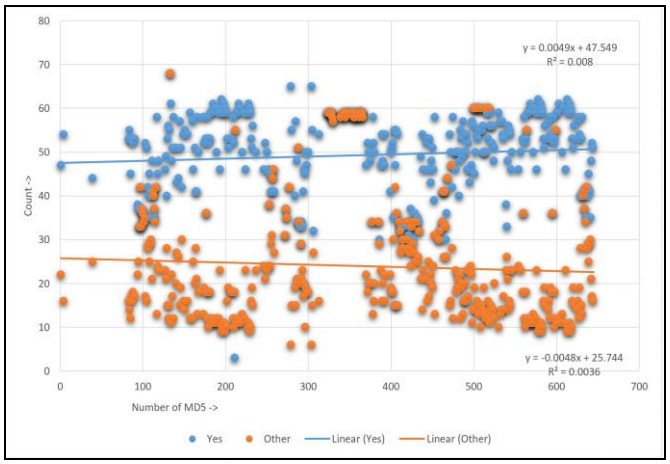


Fig. 11 Test result cluster analysis

Fig. 11 shows the cluster analysis of the result. From the cluster analysis, the following are inferences can be drawn:

- The chances of incidences of known APT are high compared to the others (NO+ Unknown)
- The detection chances are quite low if the number of MD5 signatures is less.
- There is a chance of a window between 300-400 MD5 where only unknown APT were hitting the network. In a way, the training data between this range do not have any know signatures.
- The highest number of incidences is for unknowns (>65)
- As the number of MD5 signature samples grows count of known signatures outgrows the count of unknown signatures.

8. Comparative Analysis

Based on Equations 1 and 2, identified malicious trends can be represented as in Equations 3 and 4

$$y = 0.0049x + 47.549 \quad (3)$$

$$R^2 = 0.008 \quad (4)$$

and non-malicious trend can be represented as in Equation 5 and 6.

$$y = -0.0048x + 25.744 \quad (5)$$

$$R^2 = 0.0036 \quad (6)$$

As the learning continues, several tests are more the identification trend is higher, and others are lower. The Trend line in Fig. 12 shows the trend. APT generators use several techniques to improvise, and the patterns are continually changing in nature of the task to train the system is challenging. Still, the author could achieve validation from antivirus software about the finding of the ML algorithm, and 99% of cases matched with only 1% remained as unknown. The proposed framework has a higher accuracy of 99% than the published accuracy of 96.1% [35] for early detection of APT from an unknown domain. The most important advantage of the proposed MCD algorithm is the use of ML. The more the system will get trained, the better its accuracy will achieve.

9. Conclusion

More companies are transitioning to cloud-based resources. Several of them are early adopters of artificial intelligence, quantum computing, and blockchain. But most companies face a challenge in technology innovation to make the world free from cyber threats. And as a result, cybercriminals have re-tooled to target attacks on cloud resources. Preventing cyberattacks during migration requires strong technology leadership in infrastructure migration methodologies and techniques with mass application movements into the cloud within larger regulated enterprise environments.

Privileged credentials are one of the biggest targets for cybercriminals. Tools explicitly built for on-premise resources and applications are insufficient for applications, databases, and software development platforms spread across multiple clouds. The concern is that 30% of HTTPS servers still use the MD5 server signature. The paper attempts to use an ML algorithm for the early detection of potential APT. The objective is to use the MD5 signature as the pattern to trace to root of the request and identify if the source has malicious intent or history.

To develop the ML algorithm, used known APTs and their variants caused havoc over the last decade. The present work is on an application (MCD) to detect an APT and send an alarm to the intended unsuspecting user. Since the variants are ever-changing, the author could conclude that the novel idea that the author used can detect a potential APT at the early stage of attacks with an accuracy of 99%, where there is a chance of compromise due to MD5 hashing for authentication approach.

10. Future Scope

An issue with TLS 1.2 is mostly not configuring accurately, making websites susceptible to a threat. TLS 1.3 has removed legacy and risky elements from TLS 1.2, such as MD5. Still, it will be there for some time as upgrading all systems will take time and may happen in phases. There is ample scope to tap this vulnerability using early detection and alert system [36]. The future lies in real-time tracking of the DNS logs, authentication requests, and broadcasting the malicious incidences to prevent further security breaches. The developed MCD can provide software as a service over the cloud. MCD will provide a benefit for onsite to cloud migration platforms. The design of cloud environments with a tool such as MCD can play a vital role in the production, staging, Q.A., and development of Cloud Infrastructures running in 24×7 environments. As shown in Fig. 12, there is a potential window with only unknown. A 24×7 sampling of DNS and continuous validation of signatures will reduce the risk of this vulnerable window. The success lies in transforming vital information from the internet to train MCD. This approach will power globally, collaborating and integrating into enterprise systems in the future.

References

[1] Code E, Advanced Persistent Threat, Understanding the Danger and How to Protect Your Organization. 1st Edition. Amsterdam: Elsevier. (2012).

[2] Hyunjoo Kim, Jonghyun Kim, Ikkyun Kim, Tai-myung Chung, Behaviour-Based Anomaly Detection on Big Data. Australian Information Security Management Conference. 13 (2015) 73-80.

[3] Luh R, Schrittwieser S, Marschalek S, Janicke H. Design of an Anomaly-Based Threat Detection & Explication System, International Conference on Information Systems Security and Privacy. 3 (2017) 397-402.

[4] Mees W, Multi-Agent Anomaly-Based APT Detection, Information Assurance and Cyber Defence. 111 (2012) 03.

[5] Vatamanu C, Gavrilut D, Benches R, A Practical Approach on Clustering Malicious Pdf Documents, Journal of Computer Virology and Hacking Techniques. 8 (2012) 151-163.

[6] Caldwell T, Spear-Phishing, How to Spot and Mitigate the Menace, Computer Fraud and Security. 1 (2013) 11-16.

[7] Xiaohua Yan, Joy Ying Zhang. Early Detection of Cyber Security Threats using Structured Behavior Modeling, ACM Transactions on Information and System Security. V(N) (2013) A.

[8] Zimba A, Chen H, & Wang Z, Bayesian Network-Based Weighted APT Attack Paths are Modelling in Cloud Computing, Future Gener. Comput. Syst. Elsevier. 96 (2019) 525-537.

[9] Lajevardi, Amir & Amini, Morteza. A Semantic-Based Correlation Approach for Detecting Hybrid and Low-Level Apis. Future Generation Computer Systems. 96 (2019).

[10] D. Yan, F. Liu and K. Jia, Modelling an Information-Based Advanced Persistent Threat Attack on the Internal Network. ICC 2019 - 2019 IEEE International Conference on Communications (ICC). (2019) 1-7. doi: 10.1109/ICC.2019.8761077.

[11] Bai, Tim & Bian, Haibo & Abou Daya, Abbas & Salahuddin, Mohammad & Limam, Noura & Boutaba, Raouf. A Machine Learning Approach for RDP-based Lateral Movement Detection. (2019).

[12] Chu, Wen-Lin & Lin, Chih-Jer & Chang, Ke-Neng. Detection and Classification of Advanced Persistent Threats and Attacks Using the Support Vector Machine. Applied Sciences. 9 (2019) 4579.

[13] Alshamrani A, Myneni S, Chowdhary A, and Huang D, A Survey on Advanced Persistent Threats: Techniques, Solutions, Challenges, and Research Opportunities, IEEE Communications Surveys & Tutorials. 21(2) (2019) 1851–1877.

[14] Li Z, Chen Q. A, Yang R, and Chen Y, Threat Detection and Investigation with System-Level Provenance Graphs: A Survey. (2020).

[15] W. Ul Hassan, D. Li, K. Jee, X. Yu, K. Zou, D. Wang,Z. Chen, Z. Li, J. Gui, A. Bates, J.-i. Gui, This is Why We Can't Cache Nice Things: Lightning-Fast Threat Hunting using Suspicion-Based Hierarchical Storage, in ACSAC. 14 (2020). doi:10.1145/3427228.3427255.

[16] Q. Wang, W. U. Hassan, D. Li, K. Jee, X. Yu, K. Zou, J. Rhee, Z. Chen, W. Cheng, C. A. Gunter, H. Chen, You Are What You Do: Hunting Stealthy Malware via Data Provenance Analysis. (2019) 17.

[17] M. N. Hossain, S. Sheikhi, R. Sekar, Combating Dependence Explosion in Forensic Analysis Using Alternative Tag Propagation Semantics, In IEEE S&P. (2020).

[18] W. U. Hassan, M. A. Noureddine, P. Datta, A. Bates, OmegaLog: High-Fidelity Attack Investigation via Transparent Multilayer Log Analysis. (2020) 16.

[19] S. Ndichu, S. Kim, S. Ozawa, T. Misu, K. Makishima, A Machine Learning Approach to Detecting Javascript-Based Attacks Using AST Features and Paragraph Vectors, Applied Soft Computing Journal. 84 (2019) 105721. doi: 10.1016/j.asoc.2019.105721.

[20] Z. Li, Y. Chen, Q. Chen, T. Zhu, C. Xiong, H. Yang, Effective and Light-Weight Deobfuscation and Semantic-Aware Attack Detection For Powershell Scripts, In Proceedings of the ACM Conference on Computer and Communications Security. (2019). doi:10.1145/3319535.3363187.

[21] H. Wang et al., An Evolutionary Study of IoT Malware, in IEEE Internet of Things Journal. 8(20) (2021) 15422-15440. doi: 10.1109/JIOT.2021.3063840.

[22] W. Zhang, H. Wang, H. He and P. Liu, DAMBA: Detecting Android Malware by ORGB Analysis, IEEE Trans. Rel. 69(1) (2020) 55-69.

[23] H.-S. Ham, H.-H. Kim, M.-S. Kim and M.-J. Choi, Linear SVM-Based Android Malware Detection, Proc. Front. Innov. Future Comput. Commun. (2014) 575-585.

[24] A. Calleja, J. Tapiador and J. Caballero, The Malsource Dataset: Quantifying Complexity and Code Reuse in Malware Development, IEEE Trans. Inf. Forensics Security. 14(12) (2019) 3175-3190.

[25] Lucian C. Ongoing MD5 support endangers cryptographic protocols. [Online]. Available: https://www.computerworld.com/article/3020066/ongoing-md5-support-endangers-cryptographic-protocols.html

[26] X. Y. Han, T. Pasquier, A. Bates, J. Mickens, and M. Seltzer, UNICORN: Runtime Provenance-Based Detector for Advanced Persistent Threats, In Proceedings of the Network and Distributed System Security Symposium, San Diego, CA, USA. (2020).

[27] (2019). Paganini, P. Phishers Continue to Abuse Adobe and Google Open Redirects. [Online]. Availables: https://securityaffairs.co/wordpress/91877/cyber-crime/adobe-google-open-redirects.html.

[28] Geluvaraj B, Satwik P.M, Ashok Kumar T.A, The Future of Cybersecurity: Major Role of Artificial Intelligence, Machine Learning, and Deep Learning in Cyberspace. In Lecture Notes on Data Engineering and Communications Technologies; Springer Singapore: Singapore. 15 (2019) 739–747.

[29] Ahuja R, Chug A, Gupta S, Ahuja P, Kohli S, Classification and Clustering Algorithms of Machine Learning with their Applications. In Nature-Inspired Computation in Data Mining and Machine Learning; Yang, X.S., He, X.S., Eds.; Springer International Publishing: Cham, Switzerland.11 (2020) 225–248.

[30] Cho, Do & Nam, Ha. A Method of Monitoring and Detecting APT Attacks Based on Unknown Domains. Procedia Computer Science. 150 (2019) 316-323.

[31] Weina Niu, Xiaosong Zhang, GuoWu Yang, Jianan Zhu, Zhongwei Ren. Identifying APT Malware Domain Based on Mobile DNS Logging. Mathematical Problems in Engineering. 2 (2017) 1-9.
[32] Marchetti M, Pierazzi F, Colajanni M, Guido A. Analysis of High Volumes of Network Traffic for Advanced Persistent Threat Detection. Computer Networks. 109 (2016) 127–141.
[33] A. Zimba, H. Chen, Z. Wang, and M. Chishimba, Modeling and Detection of the Multi-Stages of Advanced Persistent Threats Attacks Based on Semi-Supervised Learning and Complex Networks Characteristics, Future Generation Computer Systems. 106 (2020) 501–517.
[34] [Online]. Available: https://apt.securelist.com/
[35] Do Xuan Choa, Ha Hai Nam. A Method of Monitoring and Detecting APT Attacks Based on Unknown Domains, Procedia Computer Science. 150 (2019) 316–323
[36] Zitong Li, Xiang Cheng, Lixiao Sun, Ji Zhang, Bing Chen, A Hierarchical Approach for Advanced Persistent Threat Detection with Attention-Based Graph Neural Networks, Security and Communication Networks. (2021) 14. https://doi.org/10.1155/2021/9961342
[37] (2020). Zou, Q. An Approach for Detection of Advanced Persistent Threat Attacks. Computer, IEEE Computer Society. [Online]. Availabe: https://www.researchgate.net/publication/34726137
[38] An_Approach_for_Detection_of_Advanced_Persistent_Threat_Attacks
[39] Yan D, Liu F, & Jia K, Modelling an Information-Based Advanced Persistent Threat Attack on the Internal Network. In ICC 2019-2019 IEEE International Conference on Communications (ICC) IEEE. (2019) 1-7. IEEE. https://ieeexplore.ieee.org/abstract/document/8761077
[40] Lv K, Chen Y, & Hu C, Dynamic Defence Strategy Against Advanced Persistent Threat Under Heterogeneous Networks, Information Fusion. 49 (2019) 216-226. https://doi.org/10.1016/j.inffus.2019.01.001
[41] Joloudari J. H, Haderbadi M, Mashmool A, GhasemiGol M, Band S. S, & Mosavi A, Early Detection of the Advanced Persistent Threat Attack Using Performance Analysis of Deep Learning. IEEE Access. 8 (2020) 186125-186137. https://ieeexplore.ieee.org/abstract/document/9214817
[42] Chen W, Helu X, Jin C, Zhang M, Lu H, Sun Y, & Tian Z, Advanced Persistent Threat Organization Identification Based on Software Gene of Malware. Transactions on Emerging Telecommunications Technologies. 31(12) (2020) e3884.
[43] Cheng X, Zhang J, Tu Y, & Chen B, Cyber Situation Perception for Internet of Things Systems Based on Zero Day Attack Activities Recognition within the Advanced Persistent Threat, Concurrency and Computation: Practice and Experience. (2020) e6001.
[44] Fraser N, Plan F, O Leary J, Cannon V, Leong R, Perez D, & Shen C, APT41—A dual espionage and cybercrime operation. FireEye Blog. (2019).
[45] Vencelin Gino V, Amit KR Ghosh, Enhancing Cyber Security Measures for Online Learning Platforms, SSRG International Journal of Computer Science and Engineering. 8(11) (2021) 1-5. https://doi.org/10.14445/23488387/IJCSE-V8I11P101
[46] Tara Kissoon, Optimum Spending on Cybersecurity Measures, Transforming Government: People, Process and Policy. (2020).
[47] Vencelin Gino V & Amit KR Ghosh. IJCS. 8(11) (2021) 1-5.
[48] Donald Somiari Ene, Isobo Nelson Davies, Godwin Fred Lenu, Ibiere Boma Cookey, Implementing ECC on Data Link Layer of the OSI Reference Model. SSRG International Journal of Computer Science and Engineering. 8(9) (2021) 12-16. https://doi.org/10.14445/23488387/IJCSE-V8I9P103
[49] Evans Mwasiaji, Kenneth Iloka, Cyber Security Concerns and Competitiveness for Selected Medium Scale Manufacturing Enterprises in the Context of Covid-19 Pandemic in Kenya. SSRG International Journal of Computer Science and Engineering. 8(8) (2021) 1-7. https://doi.org/10.14445/23488387/IJCSE-V8I8P101