A spatial-temporal correlation based method for advanced persistent threat detection

Luoli Wang

1 Pre-Science, University Of Washington, Seattle, Washington, 98105, United States
*Corresponding author’s e-mail: luoliwang0612@outlook.com

Abstract: Advanced Persistent Threats (APT) have caused severe damage to the core information infrastructure of many governments and organizations. APT attacks usually remain low and slow which makes them difficult to be detected. In this case, the way of correlative analyzing massive logs generated by various security devices for effectively detecting the new type of cyber threat turns out to be more and more significant. In this paper, on the basis of analyzing the principles and characteristics of APT, we propose an intelligent threat detection method based on the expanded Cyber Attack Chain (CAC) model and the long short-term memory network (LSTM) autoencoder to extensively correlate malicious behaviors from spatial and temporal dimensions, which provides a brain new idea for the application and practice of complex network attack detection.

1. Introduction

Recently, Advanced Persistent Threats, one that is launched by professional and resourceful organization and that targets on specific and valuable objects, have gradually caught people’s attention. To be more specific, compared to traditional models of attacks, APTs possess more threats to us with the following characteristics: its targets are specific and clear; its ways of attacking are complex and diverse; its action of attacking stays low and slow; its purposes are political and financial; and it remains destructive and contagious[1].

Speaking of conventional detection of APT attacks, signature-based detection (uses security devices, such as Intrusion Detection System (IDS) in which behaviors that match the rules are detected as attacks, to analyze events in network)\(^2\)[2-4] and anomaly-based detection (applies statistical machine learning methods on training models by collecting normal data, in which the one that deviates from threshold of the model will be considered as an attack)\(^5\)[5-7] are less effective on dealing with APTs attacks. The reasons are followings. First of all, it’s difficult for signature-based detection to detect an attack that is unknow or has no record in the database. To be more specific, IDS pays more attention on one single stage of the attack instead of focusing on the combination of network and system information from different attack stages or links to establish an attack model. In this case, some behaviors that seem to be normal might be a bridge step in the APTs attacks, which might also cause us to neglect some harmful but seemingly low risk warning messages\(^8\). On top of that, speaking of anomaly-based detection, since the action of APT attacks possess the feature of low and slow, in which it’s hard for the model of traditional machine learning to effectively correlate this type of covert, time-lapse attack\(^9\). Moreover, most of the data sets for anomaly-based detection are private and proprietary. Given that many malicious sample sources are commercial in nature and the issue of intellectual property set a high threshold for this field, researchers can only train their model with limited amount of data\(^10\).
To deal with this complex attack, we have dug into the principles of attacks of APTs and corresponding ways of defending and we have proposed a design for APTs detection approach based on expanded attack chain and LSTM-autoencoder, which improves our ability to recognize APTs attacks by correlating network behaviors both in time and space. The advantages of this model are followings. To begin with, by inputting the network event sequence to the LSTM and mapping the event to the expanded attack chain model, we can amplify the attack behaviors that remains low. On top of that, by using LSTM-autoencoder, even if the behavior of the frequency of APTs attacks remains slow, we can detect attacks that are abnormal, and even in an unsupervised way. Last but not least, by training offline and detecting online, we can even capture APTs attacks in the production environment, which shows this detection approach is pragmatic.

The rest of this paper is organized as follows. Preliminaries are given in section 2. Details of the proposed method is given in section 3. A case study is presented in section 4. Section 6 concludes the paper and gives the future work.

2. Preliminaries
In advanced persistent threat attacks, the attack behavior is not only a direct destructive behavior but also a continuous penetrating process with multiple stages, which allows attacks that stay secretly and persistently in the target system. On top of that, unlike regular attacks APTs attacks remain low and slow. Owing to these features, characterizing the behavior of each APT attack individually is difficult. Thus, it’s necessary to introduce two key models, cyber attack chain and LSTM, which allows us to correlate them in time and space respectively to capture attacks.

2.1 Cyber attack chain
To begin with, attack chain, one that contains a collection of paths and means of attacks taken by the attacker and that is carried out based on different steps of the attack in target system to divide and associate each stage to form a complete attack[11], makes the detection of APT possible. For instance, a classic chain model of cyber-attack, is proposed by Lockheed Martin in 2011 from the perspective of the attacker, assisting security in identifying suspicious activities at each stage of a potential attack. The advantage of CAC model is that if security systems fail to find signs of an attack in a certain step, they can still find clues in the subsequent stages, which maximizes the chance of defending and allows security system to detect attacks in advance. Consequently, analyzing attack chains to depict and capture the logic behind malicious attacks provides conditions for network security event tracking and tracing, relationship extraction and attack prediction.

As shown in figure 1, in reconnaissance stage, intruder screens the target and looks for vulnerabilities; in weaponization stage, intruder develops malware designed to exploit the vulnerability; in delivery stage, intruder transmits the malware via a phishing email or another medium; in exploit stage, the malware begins executing on the target system; in the install stage, the malware installs a backdoor or other ingress accessible to the attacker; in the call back stage, the intruder gains persistent access to the victim’s network; in the persist stage, intruder initiates end goal actions, such as data theft, data corruption, or data destruction.

Fig 1. Classic cyber attack chain
2.2 Long short-term memory network

Long Short-Term Memory (LSTM) network, a refined version of the Recurrent Neural Network (RNN), has been explored, successfully handling the trouble that potential exploding gradient and vanishing gradient problem\cite{12}. Speaking of LSTM, even though all recurrent neural networks have the form of a chain of neural network repeating modules, the repeating module of LSTM has a different structure. In other words, there are three gates, namely input gate, forget gate and output gate, as well as memory cells with identical shapes in LSTM, which allows it to record additional information. The unit of LSTM is shown in figure 2.

![Fig 2. Long short-term memory network unit](image)

Specifically, let the number of hidden units be \( h \), the given time step is \( t \), the input is \( X_t \), the hidden state of the previous time step is \( h_{t-1} \), and the input gate \( I_t \), forgetting gate \( F_t \) and output gate \( O_t \) of time step \( t \) are calculated separately as formula 1:

\[
I_t = \sigma (X_t W_{xi} + H_{t-1} W_{hi} + b_i)
\]
\[
F_t = \sigma (X_t W_{xf} + H_{t-1} W_{hf} + b_f)
\]
\[
O_t = \sigma (X_t W_{xo} + H_{t-1} W_{ho} + b_o)
\]
\[
C_t = \tanh (X_t W_{xc} + H_{t-1} W_{hc} + b_c)
\]
\[
H_t = O_t \cdot \tanh (C_t)
\]

Among these, \( W_{xi}, W_{xf}, W_{xo} \), \( W_{xc} \) and \( W_{hi}, W_{hf}, W_{ho} \), \( W_{hc} \) are the corresponding weight parameters, and \( b_i, b_f, b_o \) and \( b_c \) are the corresponding bias parameters.

3. Our method

3.1 Expanded attack chain

The classic attack chain model described above is only suitable for modeling external attacks, but not for internal threats\cite{13}. In APT attacks, a large number of attack behaviors are carried out after obtaining internal system privileges. In this case, the classic attack chain model needs to be expanded. The expanded attack chain model is shown in Figure 3.
Compared with the classic attack chain, the expanded attack chain can be subdivided into the internal behavior and added the internal attack chain during the process of getting to the goal. In other words, the internal attack chain is to gain access to the target system, including internal scouting, internal exploit, escalate privilege, lateral movement and internal target manipulation.

### 3.2 LSTM autoencoder

The encoder-decoder model is often applied to the sequence modeling problem. It is an unsupervised learning model. The encoder converts the input sequence into a fixed-length vector representation, inversely the decoder, convert the previously generated representation into an output sequence. The performance of the model is evaluated based on the model's ability to recreate the input sequence. The essence of the autoencoder model is to realize the mapping between intuitive representations (such as network event sequences) and vector representations.

As illustrated in figure 4, there are two recurrent neural networks in the model, the encoder LSTM and the decoder LSTM. The input of the model is the vector sequence (network events). When the last input is read in, the internal state and output state of the encoder will be directly fed into the decoder. The decoder reconstructs a sequence the same as the target sequence, except that the order is reversed. Reversing the order can make the optimization easier because the output of LSTM is also the reverse.
3.3 APT detection based on expanded CAC and LSTM-autoencoder

In our design, we propose a framework for APT detection using the expanded attack chain model and the LSTM autoencoder model. Figure 5 shows the overall framework of the method, which can be divided into the following four steps.

**Step 1 Expanded attack chain mapping:**
Input the network event sequence $e_{i,j}, ..., e_{i,k}, e_i$ in a window with length $j$ to LSTM-autoencoder, and at the same time, map the event detected by a rule-based IDS to expanded attack chain model;

**Step 2 Event weight update:**
Obtain the weight $w_i$ of each event $e_i$ according to the result of attack chain mapping, and update it to the event sequence that has been input to the LSTM-Autoencoder model in step 1;

**Step 3 Offline latent representation learning:**
Train the LSTM autoencoder model with updated weight sequence generated by the step 2 to combine the information of events at different times to extract the changes of characteristics of the event, and perform the learning of the representation vector, and represent the entire sequence of events in the window as a fix-length vector (Entire training process conducted offline);

**Step 4 Online anomaly event detection:**
Use the mature anomaly detection algorithm RRCF$^{[14]}$ on the data stream to give anomaly scores. In order to determine the sequence of anomalies, a threshold can be set and it is considered that the sequence whose score exceeds the threshold is anomaly. (Entire detecting process conducted online).

**4. Case study**

We deployed the above-mentioned prototype system on the server of the information technology center of Campus Network. The center provides fundamental network services with the purpose of scientific researching, is responsible for maintaining and managing more than 10,000 pieces of information, 500 network equipment, and 40,000 campus online accounts. Programs running on the host server on Campus Network carry the main function of many service systems and development systems. Once a cyber invasion occurs, it’s inevitable that the operation of the Campus Network system will be affect in a large extent. Consequently, the protection of the host server and ability to detect cyber-attacks are particularly significant. Following is the study we’ve carried out based on the scenario mentioned above.

The monitoring objects of the detection application cover Linux hosts in a laboratory of the campus network. The subject of the detection is the operation of Command line (CMD request) after logging in.
Shell remotely and the interfaces of the remote terminal to the host Call (URL request). For example, a CMD operation sequence is \([\text{cd}, \text{netstat}, \text{ls}, \text{ll}, \text{chage}, \text{ll}, \text{ifconfig}, \text{ll}, \text{chpasswd}, \text{chage}, \text{chage}, \text{more}, \text{exit}, \text{ll}, \text{chpasswd}, \text{exit}, \text{ll}, \text{ll}, \text{more}, \text{chage}, \text{grep}, \text{grep}]\); and an interface call request (url) is “ip:port/login/login_frame.php?site=default”.

Speaking of the case of operation of CMD, the detection found a large number of SCP commands input behavior. The abnormal behavior occurred on a host from 16:00 to 17:00 on April 8, 2021. Afterwards, the time of occurrence of the abnormal behavior of the host and the sequence of operations in the past six months were counted separately. The distribution showed that within the six months, there were only 28 SCP operations, however, all of which happened within this one hour. This case is very likely to be abnormal in human opinion, but unfortunately, the existing detection system did not notice this anomaly.

![Fig 6. Top7 distribution of shell command (One-hour and Six-month)](image)

Speaking of the interface call request, the anomaly detection engine found that a certain host interface was accessed abnormally 13 times with the abnormal URL that was "ip:port/_vti_bin/shtml.exe/_vti_rpc" on May 20, 24, and 27, 2021. It has been verified that the call is a buffer overflow vulnerability in Microsoft FrontPage, which can cause the host-side server to crash and have the possibility of executing arbitrary code.

Consequently, both cases of the CMD operation and URL interface did not generate corresponding alarm in existing security system, which causes the internal threat behavior successfully to evade the monitoring system. Under this circumstance, further attacks on system will occur.

5. Conclusion and future work
Given that existing defensive approaches of network attack mainly focus on the detection of regular network attack behaviors and the detection of APT attacks is ineffective, we have proposed a detection model combined with expanded attack chain and LSTM-autoencoder which makes broadly correlating network events possible.

By correlating network events in APTs attacks with respect to time and space in our approach, there is no doubt that the ability to recognize APTs attacks has been improved. In fact, the fine granularity of the expanded attack chain can still be refined. On the other hand, while the length of the network events keeps increasing, the accuracy of the detection will drop. Thus, we will keep digging into the detection of APTs attacks and dealing with it as effective as possible.

Reference
[1] M. Li, W. Huang, Y. Wang, W. Fan, and J. Li, “The study of APT attack stage model,” in 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Jun. 2016, pp. 1–5. doi: 10.1109/ICIS.2016.7550947.
[2] I. Karim and Q.-T. Vien, Snort-based intrusion detection system for practical computer networks: implementation and comparative study. LAP LAMBERT Academic Publishing, 2017. Accessed: Sep. 26, 2021. [Online]. Available:
[3] S. More, M. Matthews, A. Joshi, and T. Finin, “A Knowledge-Based Approach to Intrusion Detection Modeling,” in 2012 IEEE Symposium on Security and Privacy Workshops, May 2012, pp. 75–81. doi: 10.1109/SPW.2012.26.

[4] R. Rajendran, S. V. N. Santhosh Kumar, Y. Palanichamy, and K. Arputharaj, “Detection of DoS attacks in cloud networks using intelligent rule based classification system,” Cluster Comput, vol. 22, no. 1, pp. 423–434, Jan. 2019, doi: 10.1007/s10586-018-2181-4.

[5] J. Hong, C.-C. Liu, and M. Govindarasu, “Integrated Anomaly Detection for Cyber Security of the Substations,” IEEE Transactions on Smart Grid, vol. 5, no. 4, pp. 1643–1653, Jul. 2014, doi: 10.1109/TSG.2013.2294473.

[6] D. S. Terzi, R. Terzi, and S. Sagiroglu, “Big data analytics for network anomaly detection from netflow data,” in 2017 International Conference on Computer Science and Engineering (UBMK), Oct. 2017, pp. 592–597. doi: 10.1109/UBMK.2017.8093473.

[7] A. L. Buczak and E. Guven, “A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection,” IEEE Communications Surveys Tutorials, vol. 18, no. 2, pp. 1153–1176, 2016, doi: 10.1109/COMST.2015.2494502.

[8] P. Gao et al., “An Intelligent Threat-Detection Method for Power Monitoring System Based on Attack Chain Knowledge,” in 2021 IEEE 4th International Electrical and Energy Conference (CIEEC), May 2021, pp. 1–6. doi: 10.1109/CIEEC50170.2021.9510852.

[9] “Advance Persistent Threat Detection Using Long Short Term Memory (LSTM) Neural Networks | SpringerLink.” https://link.springer.com/chapter/10.1007/978-981-13-8300-7_5 (accessed Sep. 26, 2021).

[10] R. Harang and E. M. Rudd, “SOREL-20M: A Large Scale Benchmark Dataset for Malicious PE Detection,” arXiv:2012.07634 [csf], Dec. 2020, Accessed: Sep. 26, 2021. [Online]. Available: http://arxiv.org/abs/2012.07634

[11] “Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains: National CyberWatch Center.” https://www.nationalcyberwatch.org/resource/intelligence-driven-computer-network-defense-informed-by-analysis-of-adversary-campaigns-and-intrusion-kill-chains-2/ (accessed Sep. 26, 2021).

[12] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[13] T. Dargahi, A. Dehghantanha, P. N. Bahrami, M. Conti, G. Bianchi, and L. Benedetto, “A Cyber-Kill-Chain based taxonomy of crypto-ransomware features,” J Comput Virol Hack Tech, vol. 15, no. 4, pp. 277–305, Dec. 2019, doi: 10.1007/s11416-019-00338-7.

[14] S. Guha, N. Mishra, G. Roy, and O. Schrijvers, “Robust Random Cut Forest Based Anomaly Detection on Streams,” in Proceedings of The 33rd International Conference on Machine Learning, Jun. 2016, pp. 2712–2721. Accessed: Sep. 26, 2021. [Online]. Available: https://proceedings.mlr.press/v48/guha16.html