APTGen: An Approach towards Generating Practical Dataset Labelled with Targeted Attack Sequences

Yusuke Takahashi
NEC Corporation

Shigeyoshi Shima
NEC Corporation

Rui Tanabe
Institute of Advanced Sciences, Yokohama National University

Katsunari Yoshioka
Graduate School of Environment and Information Sciences, Yokohama National University

Abstract

In incident response for targeted cyber attacks, the responders investigate the sequence of attacks (attack sequence) that intruders have followed by analyzing the remaining logs. Their goal is to grasp and understand the whole picture of the incident. For accelerating incident response, it is important to develop technologies to automate the investigation of the attack sequences. However, we see lack of open dataset that contains logs and corresponding attack sequence information in order to evaluate these technologies.

In this paper, we propose APTGen, an approach for generating targeted attack dataset. APTGen is top-down, that is, it first generates artificial attack sequence from existing security reports based on the attack model defined in MITRE's ATT&CK. Then, in order to obtain logs from execution environments, it executes corresponding attack tools to realize the attack sequences. Thanks to the top-down approach, we can obtain the attack sequence information corresponding to the attack trace left in the logs. We generate 800 different attack sequences and logs based on reports of eight actual security incidents. We publish generated sequences and logs as a dataset for R&D of incident responses.

1 Introduction

The constant threats of targeted cyber attacks are one of the major security challenges in nowadays. Companies and organizations face a great number of cyber attacks on a daily basis and security vendors have made effort in developing security solutions. Especially, countermeasures against targeted cyber attacks seems to continue evolving. However, attackers are trying to bypass these security measurements and it is hard to prevent all security breaches from happening. Therefore, it is important to respond promptly after the breaches to minimize the damage. In many cases, the attackers' motivation is to steal confidential business information. They achieve their goal by interactively and remotely operating the compromised host and by spreading the infection among the corporate network. Security products and indicators of compromise (IoC) are used to prevent, detect and mitigate such attacks. When the organization realizes the security breach within its network, the computer security incident response team (CSIRT) responds to the incident. The mission of the CSIRT is to reveal the whole picture of the attack through an incident response cycle, which consists of detection, analysis, containment, eradication and recovery [19, 35]. During the process, the CSIRT not only investigates the attack methods that an attacker executed in the corporate network, but also investigates the sequence of these attack methods (attack sequence) and the attacker's purpose. The faster the whole picture of the attack gets revealed, the period between detection and containment or eradication becomes shorter. As a result, security researchers have shed light in revealing the purpose of the attack and developed methods to automate or support investigating attack sequences.

Targeted cyber attacks are executed remotely and evidences of the attack are often left in network logs and endpoint logs. For this reason, a CSIRT investigates the attack sequence by analyzing attack traces left in devices that were operated in the corporate network. Various technologies are developed to help the CSIRT conduct the task. In order to evaluate these technologies for incident handling, it is important to prepare a dataset that corresponds with attack campaigns in the wild. The dataset is required to consist not only the attack sequence data, but also the network logs and endpoint logs that contain attack traces related to the sequence. Moreover, to evaluate whether a proposed method is practical, it is desired that there are various kinds of attack sequences and logs in the dataset. However, to the best of our knowledge, datasets that contain both the attack sequences and corresponding logs are limited. For this reason, we have decided to build the dataset for R&D for incident handling by ourselves.

We first considered obtaining logs of actual targeted cyber attacks and use them as a dataset. There are two ways to obtain this kind of data. The first way is to get the log from organizations victimized by targeted cyber attacks. Several researches that use data from the victims of actual targeted

cyber attacks have been conducted [17, 26]. Although there are series of researches and tools [14] that could be used for anonymizing the logs from real incidents, we find that victim organizations are often reluctant to publish the logs as open dataset. The other way is to observe cyber attacks purposely initiated in an observation environment. Farinholt et al. have observed human-operated attacks in their paper [23]. Still, it is difficult to keep deceiving attackers into thinking that they were in the target network to observe their long-term behavior to reveal the whole attack campaign and learn how an attacker achieves his goal. Along with above, even if we successfully observe the conducted attacks, it is difficult to infer corresponding attack sequences without knowing the true intention of the remote attackers.

With above observation, in this paper, we propose APTGen, an approach for generating attack sequences and executing them for building a dataset. APTGen first generates artificial attack sequences and executes corresponding attacks in an experimental environment. Thanks to this approach, we can obtain the attack sequence corresponding to an attack trace left in the logs. We generate 800 different attack sequences based on eight actual security incidents and obtain logs by executing generated sequences in an experimental environment. We show the relationship between 800 generated sequences by visualization. We publish generated sequences and logs as a dataset on our webpage [3].

The contributions of this paper are as follows:

- We propose an approach for generating artificial attack sequences of targeted cyber attacks and executing corresponding attacks to generate dataset of targeted attacks.
- We develop tools that artificially generate and execute attack sequences that attackers may execute in the targeted organization's network.
- We analyze the relationship between the generated attack sequences by visualization.
- We release a dataset consisting of generated attack sequences and logs that were obtained by executing these sequences.

2 Related Work

The closest works to our approach are the researches of automatically generating attack trees, automation of penetration and red team [15,22,27,32,36]. Falco et al. proposed a method of automatically generating attack trees for smart cities [22]. Automation techniques of red team were proposed by [15,32]. The works in [27,36] surveyed planning techniques and frameworks in these researches of automated attacks and discussed planning problems. Specially, the importance of providing multiple attack plans was mentioned in [27]. These works used planning techniques and focused on a single attack sequence that an attacker may execute in a targeted environment. However, the purpose of this paper is to generate multiple attack sequences that an attacker may execute in a targeted environment. In addition, obtaining logs was out of scope in these previous works, but we also focus on it. To the best of our knowledge, we are the first to cover generation of attack sequences through collection and use of logs for the purpose of building a dataset.

3 Problems on Generating and Executing Attack Sequences

Figure 1 shows our proposed method of generating attack sequences artificially and obtaining logs by executing them. The method is designed for security researchers and analysts (hereafter called users) to provide them with dataset of targeted cyber attacks with corresponding logs. The users start with fragmentary information about the attack from existing incident reports and security articles. Next, they decide attack methods that suit fragmentary information and generate an attack sequence by arranging the methods to match the detail of the attack. Finally, they execute a generated attack sequence in their experimental environment to obtain logs containing traces of the attack.

There are problems from two perspectives in a user carrying out this method.

Reproducibility: There is a problem that the names of attack methods and their scopes are not unified nor consistent due to the lack of their details in referred incident reports and security articles. For example, whether to describe the method as Exploit or Buffer Overflow depends on the user. We cannot effectively develop and evaluate without unified attack sequence information because it is treated as the ground truth in the dataset. Thus, it is necessary for the users to be able to generate attack sequences with unified expression.

Reality: Although we are generating the attack sequences and corresponding logs from the security reports of real incidents, we need to evaluate how realistic they are compared to the logs obtained from the real cyber attacks. Our future plan is to discuss with the experts in CSIRT and have them evaluate the reality of the generated dataset.

We would like to generate various attack sequences and logs artificially to enrich the dataset that could be used in various studies and exercises.

Diversity: There are several possible attack sequences we can generate from fragmentary information about the attack in incident reports and security articles. This is because there are multiple sequences to achieve an attacker's goal. However, a simple reordering of attack methods obtained from fragmentary information may result in a sequence that does not work as an attack. Therefore, it is necessary to generate various attack sequences that function as targeted cyber attacks.

We first develop a method to solve the problem concerning reproducibility and then develop one to solve the problem concerning diversity in this paper.

4 Method of Attack Sequence Generation

In this section, we describe attack sequence generation in APTGen. In order to solve the reproducibility problem, it is necessary to systematize attack methods used in targeted cyber attacks and unify their names and scopes. We use ATT&CK [7,34], which was developed for attacker emulation. ATT&CK is a framework and knowledge base of attacker tactics and techniques, which is built based on actual incidents. ATT&CK abstracts an attacker's actions in terms of *Tactic*, *Technique*, and *Software*. We show ATT&CK model and ATT&CK model example based [34] in Figure 2. Software is an implementation of a Technique, and a Technique is a method of accomplishing a Tactic. Attack method names in generating attack sequences are unified using ATT&CK. At the same time, the scopes of attack methods are also unified due to the design that a Technique is used to accomplish the corresponding Tactic.

We structure an attack sequence to generate it by using ATT&CK. In this study, we treat attack sequences structured with ATT&CK as the ground truth in the dataset for investigating attack sequences. We show the structure of an attack sequence in Figure 3. In this attack sequence, attack methods are arranged in order of execution and software called Mimikatz is used in the k-th step.

4.1 Manual Attack Sequence Generation

We confirm whether structuring attack sequences with ATT&CK meets the requirement of reproducibility by generating an attack sequence manually. The attack sequence is generated in the following procedure by referring to [8].

1. Identify the incident reports or articles.
2. Extract information about attack methods from identified reports or articles and map them to Techniques in ATT&CK.
3. Generate an attack sequence to execute the use of Techniques corresponding to extracted attack methods. Compensate for the lack of a Technique if an attack sequence is not chained logically with only extracted Techniques.
4. Apply Software to Techniques and set up executable codes in the experimental environment.

We targeted incident reports about the targeted cyber attack on Japan Pension Service [18, 29, 31] because there have been reports issued by several organizations and we thought that there was little missing information about the attack. We generated an attack sequence based on the targeted cyber attack. The names of attack methods did not vary because they were selected from Techniques defined on ATT&CK. Reports described incident response in detail, but all attack methods attacker had executed were not described in reports. We compensated for the lack of a Technique because an attack sequence was not chained logically with only extracted Techniques. It was easy to identify the Techniques to compensate for, because the scope of a Technique was specific depending on the Tactic.

4.2 Attack Sequence Generation Tool

After addressing the reproducibility problem using ATT&CK, we now consider diversity. In order to solve the diversity problem, it is necessary to generate multiple attack sequences from fragmentary information related to an incident. However, it is difficult to generate them manually. Therefore, we developed attack sequence generation tool (hereafter, generation tool), which generates attack sequences automatically. We show a summary of the input and output of this tool in Figure 4. Generation tool generates attack sequences, which are then used as an input for the attack sequence execution tool (hereafter, execution tool). We explain the execution tool in Section 6. The install phase in Intrusion Kill Chain [28] is out of the scope of our execution tool because this tool can execute only an attack sequence after the install phase. Therefore, we exclude Initial Access on ATT&CK from selectable Tactics in generating a sequence. We now explain other excluded Tactics. We exclude the Execution Tactic from selectable Tactics because the Technique in the Execution Tactic is regarded as Software that Techniques in other Tactics run through. We exclude the Command and Control Tactic in ATT&CK in a similar manner because it consists of Techniques that show not attacks executed on the host by the client of our execution tool but the attributions of the command and control channel. The Impact Tactic is excluded from selectable Tactics, so we generate attack sequences, the goal of which is to spread infection or steal information in an organization's network.

Our generation tool generates one step in an attack sequence in chronological order. If it finishes generating a step in an attack sequence and the sequence fulfills the input generation condition, it will output the sequence and terminate. The generation condition can specify the following: the length of an attack sequence, the Tactic and the Technique in the last step, and the Tactic and the Technique contained in an attack sequence. In case of an attack sequence to steal information, the Exfiltration Tactic or Technique in it should be specified. Elements in a step are selected randomly to generate various attack sequences, and the Tactic in a step is selected from selectable Tactics. The Technique in a step is selected from the input Technique list obtained by mapping information about attack methods in incident reports and security articles

Figure 1: The overview of steps in APTGen. Cells with rounded corners and light blue background indicate the process to generate data and those with orange background indicate data.

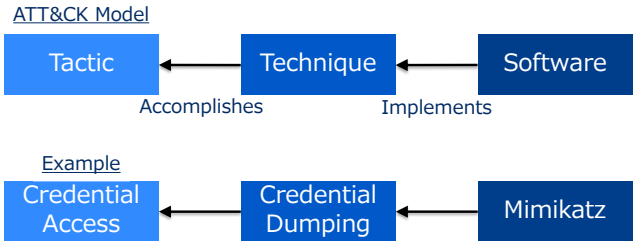


Figure 2: ATT&CK model and ATT&CK model example.

Step	Tactic	Technique	Software
1	Ta_1	Te_1	S_1
⋮	⋮	⋮	⋮
k	Credential Access	Credential Dumping	Mimikatz
⋮	⋮	⋮	⋮
n	Ta_n	Te_n	S_n

Figure 3: Structure of attack sequence

to ATT&CK. The Software in a step is selected from an executable code database that contains executable codes, which are executed on the host by the client of our execution tool, and their definition names. The definition names are used in Software in an attack sequence. We explain an executable code database in Section 6. If elements in a sequence are simply selected randomly, the attack sequence will break down based on the following three concerns. Therefore, we set constraints from each concern. The first concern is that attack sequences may contradict ATT&CK design by selecting elements in a step randomly. In ATT&CK, Tactic, Technique, and Software correspond to each other, as shown in Figure 2. We set a constraint that selects elements in a step following the ATT&CK design because the correspondence in the design is broken by naive random selection. For example, a Technique called Remote Desktop Protocol is selected in only the step in which a Tactic called Lateral Movement has already been selected because Remote Desktop Protocol belongs to Lateral Movement. The second concern is that Techniques are not chained into a logical sequence of actions in an attack sequence by naive random selection. This is because there are Techniques that require information obtained from other Techniques. A concrete example is Techniques to spread infection cannot be executed without information about other hosts. We set the

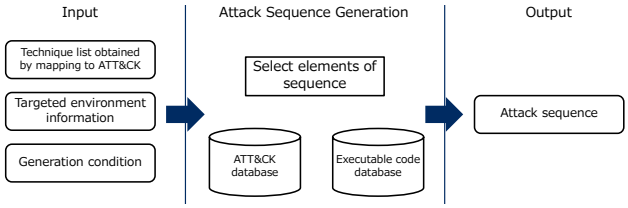


Figure 4: Overview of our attack sequence generation tool

following constraints so that an attack sequence is logically completed.

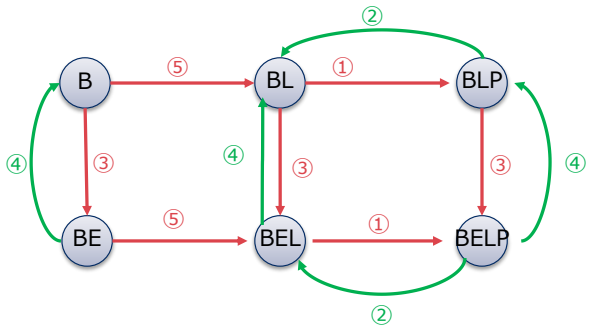
- It is necessary to execute any Techniques in Discovery before executing any Techniques in Lateral Movement.
- It is necessary to execute any Techniques in Collection before executing any Techniques in Exfiltration.
- Techniques in Exfiltration take all information outside of an organization's network.
- It is not necessary for any Techniques in Persistence to be repeated on the same host.

Selectable Tactics in generating an attack sequence change according to the Technique selected in a step, such as in Figure 5, due to these constraints. The last concern is about attack sequence execution based on generated attack sequences. Attack sequences generated by naive random selection may not be suitable for the experimental environment because there are Techniques and Software that depend on a specific OS. Our generation tool manages host information where the generated step will run based on input information about the experimental environment. The host where the generated step will run is changed to another host when selecting any Techniques in Lateral Movement.

5 Experiment

5.1 Experiment Design

We evaluated our generation tool with incident reports and security articles of actual targeted cyber attacks. The purpose of evaluation is to confirm diversity of outputted attack sequences. It is necessary to target incidents with multiple attack methods the attacker executed in incident reports and security articles. We found the eight incidents shown in Table 1 by



ID	Selected Technique	Operation on Selectable Tactics
①	Any Techniques in Lateral Movement	add Persistence
②	Any Techniques except Port Knocking in Persistence	remove Persistence
③	Data Staged	add Exfiltration
④	Exfiltration Over Command and Control, Exfiltration Over Alternative Protocol, Exfiltration Over Other Network Medium, Exfiltration Over Physical Medium	remove Exfiltration
⑤	Network Service Scanning, Remote System Discovery	add Lateral Movement

Figure 5: Transition of selectable Tactics set in generating sequence. ID in table is same as that in transition figure. First, selectable Tactics is in B state, which means Basic selectable Tactics. B state contains Credential Access, Discovery, Persistence, Collection, Defense Evasion, and Privilege Escalation. When Remote System Discovery is selected in B state, next state is BL. BL is state in which Lateral Movement is added to Basic selectable Tactics.

searching for security incident reports from security vendors and articles on security news sites. We created Technique list correspond to each incidents from documents in Table 1. Table 2 shows the Technique list corresponding to the incident of Japan Pension Service. Technique lists share some common Techniques such as Exfilteration Over Command and Control Channel because these Techniques are often used by attackers. Please see our published dataset for the details of the other Technique lists. We generated subspecies of attack sequences that are executed by an attacker in each incident by using the created Technique list. We considered the difference in attack sequences from the perspective of the series of the 3-tuple in a step. Specifically, we made tuples that consist of the Tactic, Technique, and Software in a step and arranged the tuples according to the attack sequence. If a series corresponds with other series in terms of elements in each tuple and order of tuples, we consider that the two series are the same. If not, they are different series. We hypothesize that if we can generate various short attack sequences, we will generate more attack

Table 1: 8 different incidents targeted in experiment. Name in Incident column means targeted organization, event, or attacker.

Incident	Reports or Articles
APT29	[2, 16, 21, 24]
Bronze Butler	[4, 13, 33]
Clinton campaign	[25]
Japan Pension Service	[18, 29, 31]
National Institute of Advanced Industrial Science and Technology	[11]
SingHealth	[9]
South Korean banks and broadcasting organizations	[12]
Ukrainian electricity distribution companies	[20]

sequences by changing the sequence length or defining the Software in the executable code database and generate 100 sequences per incident.

We prepared an experimental environment imitating an enterprise network, which consisted of an intranet, a DMZ, and a pseudo internet, for executing attack sequences (see Section 5.2). We input information about the hosts on the intranet as targeted environment information into our generation tool. We set five generation conditions per Technique list from the following perspectives: sequence length, Tactics/Techniques that the experts in CSIRT, characteristic Tactics/Techniques in a incident, and Techniques that attackers commonly use such as File Deletion. We generated 20 attack sequences targeting the Windows hosts per generation condition. Table 3 lists examples of the generation conditions for generating sequences based on the incident of Japan Pension Service.

5.2 Experimental Environment

The experimental Environment consists of an intranet, a DMZ, and a pseudo internet. There are four segments with five Windows hosts and a segment with file server and Active Directory (AD) server on the intranet in the environment. There are a proxy server and a DNS server on the DMZ. The intranet and the DMZ are connected to the pseudo internet, not the real Internet.

5.3 Analysis

We visualized the components of attack sequences to confirm diversity of outputted attack sequences. To visualize attack sequences, we focused on the frequency of a 3-tuple, which consists of Tactic, Technique, and Software in a step. It was reported that some attackers have common attack vectors, but some have characteristic attack vectors [1, 5]. Therefore, we

Table 2: Technique list in Japan Pension Service Incident. Tactic column shows Tactic to which Technique in Technique name column belongs.

Technique name	Tactic
Email Collection	Collection
Data Staged	Collection
Account Manipulation	Credential Access
Credential Dumping	Credential Access
Credentials in Registry	Credential Access
File Deletion	Defense Evasion
System Information Discovery	Discovery
System Network Configuration Discovery	Discovery
File and Directory Discovery	Discovery
Account Discovery	Discovery
Permission Groups Discovery	Discovery
Network Share Discovery	Discovery
Remote System Discovery	Discovery
Exfiltration Over Command and Control Channel	Exfiltration
Pass the Hash	Lateral Movement
Remote File Copy	Lateral Movement
Scheduled Task	Persistence

reduced the degree of importance of a 3-tuple, which is in many sequences, and increased it, which is in few sequences, by TF-IDF. Figure 6 shows the visualization of TF-IDF vectors for attack sequences by using t-SNE [30]. The points, which signify the attack sequences in Figure 6, are labelled incident IDs from 1 to 8, which indicate the incidents in Table 1. We found that attack sequences based on the same incidents build a cluster. In other words, similar attack sequences are distributed in a cluster. This means that various attack sequences can be generated from a Technique list.

6 Method of Attack Sequence Execution

In this section, we describe attack sequence execution in APT-Gen. We consider generating logs that contain traces of attack sequences because we solved the problems of reproducibility and diversity. We implemented our execution tool, which executes an attack sequence in an experimental environment. The tool executes the attack sequence and gets the logs from a host that it runs after execution. The tool consists of a server that controls execution and a client that executes attacks under the server control. The server has the function of editing the parameters of executable codes, the function of controlling attack sequence execution, and the function of ordering the client to obtain logs from the host. The server sends executable codes corresponding to a step of an attack sequence to the client when receiving a request packet from the client.

Table 3: Conditions for generating sequences based on Japan Pension Service Incident.

ID	Generation Conditions
1	Sequence length is 8 or more.
2	Sequence length is 1 or more and last Tactic in sequence is Lateral Movement.
3	Sequence length is 1 or more and last Technique in sequence is Exfiltration Over Command and Control Channel.
4	Sequence length is 1 or more, sequence contains Lateral Movement, and last Technique in sequence is File Deletion.
5	Sequence length is 1 or more, last Technique in sequence is File Deletion, and sequence contains Lateral Movement and Exfiltration Over Command and Control Channel.

Then, it refers to the same executable code database as our generation tool and obtains executable codes corresponding to Software in an attack sequence. The client executes the codes received from the server on a host then saves the executed codes and execution times in execution logs.

We used the software definitions[1] of Atomic Red Team [10] as the executable code database because the software definitions grouped by each Technique in ATT&CK are suitable for generating and executing attack sequences. However, since there were definitions that did not suit our methodology, we made the following changes to them. Several Techniques in ATT&CK are not defined in Atomic Red Team. Exfiltration Over Command and Control Channel in the Exfiltration Tactic is one of them. We defined the executable code corresponding to this technique in the executable code database because this technique was important for the execution of attack sequences whose purpose is to steal information. We modified the software definitions that need to connect to the actual Internet so that they could be executed in our experimental environment because no host in the environment connected to the actual Internet. There were some Techniques that belong to multiple Tactics and we could not judge which Tactic the executable code of software defined in these Techniques signified because there were no Tactic information in software definitions in Atomic Red Team. Therefore, we added the corresponding Tactics to software definitions in these Techniques. At present, note that our execution handles only exploit codes/tools in Atomic Red Team. Atomic Red Team has few definitions about the exploitation. We will combine some exploitation frameworks/databases with the definitions of Atomic Red Team in the future to strengthen the exploitation in attack sequence execution.

We executed the generated 800 attack sequences mentioned

[1] We used the definitions described in YAML in the *atomics* directory in the repository as the database.

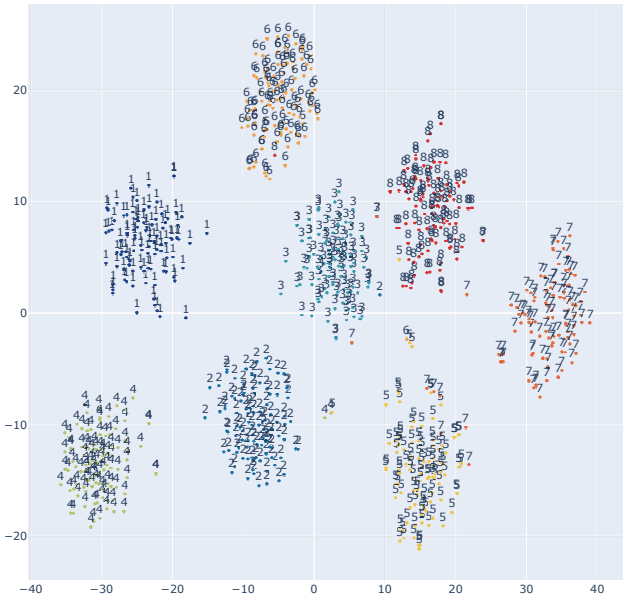


Figure 6: t-SNE visualization of attack sequences

in Section 5 by using our execution tool and obtained logs from the experimental environment. We executed them in an environment on which targeted environment information input into our generation tool was based. We targeted Windows hosts and started execution after the client of our execution tool ran in the environment. This means that execution occurs after the install phase in Intrusion Kill Chain. After the client ran, we input an attack sequence to execute into the server and set suitable values in the parameters of executable codes. After executing attack sequences, we obtained logs on the host on which the client ran by the server function and obtained ones on proxy and AD servers manually.

Therefore, we showed that we could generate correct attack sequences, network logs, and endpoint logs while meeting the requirements of reproducibility and diversity.

7 Discussion

7.1 Preparation for publishing dataset

We target logs, such as Windows event logs, that can be obtained from general enterprise environments in building a dataset so that many organizations can use our dataset. Our dataset do not contain logs related to benign background activity such as user interaction because attack sequence execution is carried out in an experimental environment. There are two ways to include not only attack logs but also benign activity logs. The first way is to execute attack sequences in actual environment. The other way is to generate benign activity logs in an experimental environment. However, the former requires a mitigating of the impact on the actual environment. Moreover, it would be more difficult to publish the dataset as it may contain sensitive information transmitted in the actual environment. The latter would require a method different from APTGen. It is our future work to prepare a dataset with both malicious and benign activities.

We share the results of our study with other researchers in two forms:

- Attack sequences and logs
  We provide attack sequences, network logs, and endpoint logs, mentioned in Section 5 [3] as a dataset. It also contains execution logs and input data for our generation tool.

- Attack sequence generation and execution tools
  We provide our generation and execution tools so that researchers can generate their own dataset. We would like to ask the requestors to accept our terms of use before providing the tools to minimize the risk of misuse. Please see [3] for more details.

7.2 Usefulness of dataset

It is possible to identify the traces corresponding to attack methods in network and endpoint logs by focusing on the time in these logs and execution logs because there are executed codes and execution times in execution logs. This is why generated attack sequences and obtained logs can be treated as the ground truth in a dataset. However, they do not completely meet the requirement of reality because we do not include the characteristics of attacker behavior in attack sequences and logs. In generating attack sequences, we include Tactics and Techniques attackers used in terms of ATT&CK, but we do not use software attackers built such as malware and attack tools. When executing sequences, we do not include attackers' characteristics such as attacking slowly over a long period. Accordingly, it is not possible to reflect the characteristics of attackers' tools and behavior in developing a method to a degree that is effective in the real world with our dataset. However, we believe that the dataset is effective for education about targeted cyber attacks because it includes the characteristics of targeted cyber attacks such as attackers collect other hosts before Lateral Movement. Reality will be strengthened by integrating the characteristics of attacker behavior revealed from the research of observing targeted cyber attacks (i.e., [23]). We still have room for improvement in generating attack sequences. To make the sequences more diverse and real, it may be effective to select Tactic/Technique/Software probabilistically.

In this paper, the solution for the reproducibility problem is to unify the names of attack methods and their scopes in attack sequences. We could unify them by using ATT&CK. However, the Technique list used in attack sequence generation needs to be created manually by referring to incident

reports and security articles. Hence, in spite of the same information source, different sequences may be generated by researchers and analysts. To solve this problem, a technology for mapping information about attack methods in incident reports and security articles to ATT&CK, such as [6] may be suitable.

7.2.1 Ethics

In this study, we were concerned with the abuse of our tools, our generated sequences, and logs because they handle detailed attack methods and tool usage. Even if we publish attack sequences and logs as a dataset, however, malicious users cannot abuse them for attacks because they do not contain attack tools. We may share our generation and execution tools within cooperative research but we do not intend to release them. If the tools should be leaked or attackers should learn of our tools by referring to this paper, our tools would not be beneficial for attackers for the following two reasons. The first reason is that our tools requires targeted environment information. In regular use, users know this information because they prepare experimental environments themselves. However, attackers do not know the targeted environment information in most cases. Therefore, it will be unlikely that attackers will be able to use our tools. The second reason is that attackers cannot determine which attack sequences would be beneficial because our generation tool does not output whether a sequence is beneficial. If attackers can obtain the targeted environment information, abuse would be limited due to the second reason.

7.3 Limitations of dataset

If users evaluate their methods of investigating attack sequences in real time, they will need to prepare their experimental environment and execute sequences manually because the dataset we are preparing does not contain our generation tool, execution tool, and our experimental environment. They will need to manually assign Tactics and Techniques to attack sequences by referring to ATT&CK if they would like to generate new sequences or improve them. Binaries and executable codes used in the executable code database will need to be prepared in their experimental environment. They will also need to manually confirm whether generated or improved attack sequences will be chained logically.

8 Conclusion

In this paper, we proposed APTGen, a method of generating attack sequences of targeted cyber attacks, which are logically completed, and executing them for building a dataset for research of investigating them. We developed tools for generating and executing various attack sequences and showed that they can generate and execute variants of actual incidents. We published 800 generated attack sequences and logs as a dataset on our webpage [3]. We believe that these sequences and logs are effective for education about targeted cyber attacks, but not sufficiently meet the reality requirements for research and development of investigating attack sequences. In this paper, we do not evaluate the reality of generated attack sequences. Thus, one of future works is to test generated sequences against detection methods. We believe that reality of the dataset will be strengthened from research on observing targeted cyber attacks. We plan to receive feedback after publishing a dataset. We will then improve the dataset and consider its suitability other than the research of investigating attack sequences.

References

[1] Advanced Persistent Threat Groups. https://www.fireeye.com/current-threats/apt-groups.html.

[2] APT29 dissected—How Russian hacking operations are built on Active Directory – Javelin Networks Blog. https://jblog.javelin-networks.com/blog/apt29-dissected-how-russian-hacking-operations-are-built-on-active-directory/.

[3] APTGen: An Approach towards Generating Practical Dataset Labelled with Targeted Attack Sequences. https://ipsr.ynu.ac.jp/aptgen/index.html.

[4] BRONZE BUTLER Hacker Group Targets Japanese Enterprises. https://www.secureworks.com/research/bronze-butler-targets-japanese-businesses.

[5] Groups | MITRE ATT&CK®. https://attack.mitre.org/groups/.

[6] Mitre-attack/tram. https://github.com/mitre-attack/tram.

[7] MITRE ATT&CK®. https://attack.mitre.org/.

[8] MITRE ATT&CK® EVALUATIONS. https://attackevals.mitre.org/adversary-emulation.html.

[9] Public Report of the Committee of Inquiry (COI) into the cyber attack on Singapore Health Services Private Limited Patient Database. https://www.mci.gov.sg/pressroom/news-and-stories/pressroom/2019/1/public-report-of-the-coi.

[10] Redcanaryco/atomic-red-team. https://github.com/redcanaryco/atomic-red-team.

[11] Report on the illegal access to AIST's information system. https://www.aist.go.jp/pdf/aist_j/topics/to2018/to20180720/20180720aist.pdf. Japanese only.

[12] Dark South Korea Total War Review. https://eromang.zataz.com/2013/04/02/dark-south-korea-total-war-review/, 2013.

[13] REDBALDKNIGHT/BRONZE BUTLER's Daserf Backdoor Now Using Steganography - TrendLabs Security Intelligence Blog. https://blog.trendmicro.com/trendlabs-security-intelligence/redbaldknight-bronze-butler-daserf-backdoor-now-using-steganography/, 2017.

[14] CAIDA: Center for Applied Internet Data Analysis. Anonymization Tools Taxonomy. https://www.caida.org/tools/taxonomy/anontaxonomy.xml.

[15] Andy Applebaum, Doug Miller, Blake Strom, Chris Korban, and Ross Wolf. Intelligent, Automated Red Team Emulation. In *Proceedings of the 32Nd Annual Conference on Computer Security Applications*, ACSAC '16, pages 363–373. ACM, 2016.

[16] Bitdefender. A Closer Look at MiniDuke. https://labs.bitdefender.com/wp-content/uploads/downloads/2013/04/MiniDuke_Paper_Final.pdf.

[17] Stevens Le Blond, Adina Uritesc, Cédric Gilbert, Zheng Leong Chua, Prateek Saxena, and Engin Kirda. A look at targeted attacks through the lense of an NGO. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 543–558. USENIX Association, 2014.

[18] NISC (National center of Incident readiness and Strategy for Cybersecurity). Investigation results of the cause related to japan pension service's personal data leak incident. https://www.nisc.go.jp/active/kihon/pdf/incident_report.pdf. Japanese only.

[19] Paul Cichonski, Tom Millar, Tim Grance, and Karen Scarfone. Computer Security Incident Handling Guide : Recommendations of the National Institute of Standards and Technology. Technical Report NIST SP 800-61r2, National Institute of Standards and Technology, 2012.

[20] SANS ICS E-ISAC. Analysis of the cyber attack on the ukrainian power grid. https://ics.sans.org/media/E-ISAC_SANS_Ukraine_DUC_5.pdf, 2016.

[21] F-Secure. THE DUKES7 YEARS OF RUSSIAN CYBERESPIONAGE. https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-Secure_Dukes_Whitepaper.pdf.

[22] G. Falco, A. Viswanathan, C. Caldera, and H. Shrobe. A Master Attack Methodology for an AI-Based Automated Attack Planner for Smart Cities. *IEEE Access*, 6:48360–48373, 2018.

[23] Brown Farinholt, Mohammad Rezaeirad, Paul Pearce, Hitesh Dharmdasani, Haikuo Yin, Stevens Le Blond, Damon McCoy, and Kirill Levchenko. To Catch a Ratter: Monitoring the Behavior of Amateur DarkComet RAT Operators in the Wild. In *Security and Privacy (SP), 2017 IEEE Symposium On*, pages 770–787. IEEE, 2017.

[24] FireEye. HAMMERTOSS: Stealthy Tactics Define a Russian Cyber Threat Group. https://www2.fireeye.com/rs/848-DID-242/images/rpt-apt29-hammertoss.pdf.

[25] Sean Gallagher. How they did it (and will likely try again): GRU hackers vs. US elections. https://arstechnica.com/information-technology/2018/07/from-bitly-to-x-agent-how-gru-hackers-targeted-the-2016-presidential-election/, 2018.

[26] Grant Ho, Aashish Sharma, Mobin Javed, Vern Paxson, and David Wagner. Detecting credential spearphishing in enterprise settings. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 469–485. USENIX Association, 2017.

[27] Joerg Hoffmann. Simulated Penetration Testing: From "Dijkstra" to "Turing Test++". In *Twenty-Fifth International Conference on Automated Planning and Scheduling*, 2015.

[28] Eric M. Hutchins, Michael J. Cloppert, and Rohan M. Amin. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Leading Issues in Information Warfare & Security Research*, 1(1):80, 2011.

[29] Japan Pension Service. Report on investigation results. https://www.nenkin.go.jp/files/kuUK4cuR6MEN2.pdf. Japanese only.

[30] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[31] Ministry of Health, Labor and Welfare. Verification report by the verification committee for japan pension service's data leak incident through unauthorized access. https://www.mhlw.go.jp/file/05-Shingikai-10201000-Daijinkanbousoumuka-Soumuka/0000095309.pdf. Japanese only.

[32] Suneel Randhawa, Benjamin Turnbull, Joseph Yuen, and Jonathan Dean. Mission-Centric Automated Cyber Red Teaming. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*, ARES 2018, pages 1–11. Association for Computing Machinery, August 2018.

[33] Secureworks. The full picture of sophisticated cyber-attacks targeting Japanese companies. https://www.secureworks.jp/~/media/Files/JP/Reports/Secureworks-Bronze-Butler-Report.ashx. Japanese only.

[34] Blake E. Strom, Andy Applebaum, Douglas P. Miller, Kathryn C. Nickels, Adam G. Pennington, and Cody B. Thomas. MITRE ATT&CK^{TM} : Design and Philosophy. 2018.

[35] Molra J West-Brown, Don Stikvoort, Klaus-Peter Kossakowski, Georgia Killcrece, and Robin Ruefle. Handbook for computer security incident response teams (CSIRTs). Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, 2003.

[36] Joseph Yuen. Automated Cyber Red Teaming. Technical Report DSTO-TN-1420, DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION EDINBURGH (AUSTRALIA) CYBER AND ELECTRONIC WARFARE DIV, 2015.