Real-time detection tool of attacks leveraging Domain Administrator privilege

The University of Tokyo
Wataru Matsuda, Mariko Fujimoto, Takuho Mitsunaga

1. Introduction

In Advanced Persistent Threat (APT) attacks, attackers who can intrude into an organization network tend to stay inside the network or repeat intrusion multiple times until they are able to accomplish their goals. When Active Directory(AD), a centralization management system for Windows computers, is in place, attackers try to get Domain Administrator's privilege which is the highest privilege of Active Directory environment. Attackers who can get the Domain Administrator's privilege likely create a backdoor that disguises itself as a legitimate account called "Golden Ticket", in order to obtain long-term administration privilege. The use of the Golden Ticket indicates that Active Directory environment is under the full control of the attackers, thus requiring immediate detection and appropriate countermeasures. However, detecting Golden Ticket is quite difficult since attackers tend to leverage legitimate accounts or legitimate tools/commands provided by Microsoft in order to avoid detection.

We introduce a real-time detection tool of attack activities abusing Domain Administrator privilege including Golden Tickets using Domain Controller's Event logs.
Our tool consists of the following steps to reduce false detection rate and help immediate response.

Step1(Signature based detection): Firstly, analyze Event logs with logical tool focusing on the characteristics of the attack activities.
Step2(Machine Learning): Analyze with anomaly detection using unsupervised machine learning and detect suspicious commands as anomaly which attackers tend to use.
Step3(Real-time alert): If attack activities are detected, raise real-time alert using Elastic Stack.

The purpose of using both of Signature based detection and Machine Learning is reduction of false detection rate. "Machine Learning" can reduce false positive rate through analyzing the result of "Signature based detection". Machine Learning can extract anomaly behaviors compared with normal operation logs, so we can easily find out whether the result of "Signature based detection" is false positive or not. As a result, it is possible to detect attacks with high accuracy by using the proposed tool even if legitimate Domain Administrator's account is leveraged.

2.2. The Golden Ticket

A Golden Ticket is a TGT that has a proper signature created by the attacker. TGT is signed by the password hash of krbtgt account, however, attackers that have exploited the domain administrative privileges could obtain the password hash of krbtgt, enabling them to create a TGT with legitimate signature. A tool to attack AD environments called mimikatz enables attackers to easily create a TGT with a significantly long term of validity (defaulted to ten years) to any given account in offline environment. Attackers tend to create a Golden Ticket for the domain administrator account. Offline in this context refers to any standalone computer that does not belong to the AD domain, and any environment that cannot communicate with the DC. The extended expiration limit of the Golden Ticket enables the attacker to continuously use it even after the password for the account he/her disguised is changed. Furthermore, since the Golden Ticket has a legitimate signature, it is difficult to differentiate it from a normal TGT. Since the detection of a Golden Ticket attack is difficult, these countermeasures are often delayed or never done, leading to increasing numbers of incidents and damages.

4. Proposed tool

In this research, we propose a new tool for detecting attacks against AD through focusing on characteristics of APT attacks and statistics of commands/tools which attackers tend to use. There are some stages of attacks against AD, we propose a tool for detecting attacker's activities with Domain Administrator's privilege.

Our tool consists of the following steps.

Step1(Signature based detection): Firstly, analyze Event logs with logical tool focusing on the characteristics of the attack activities.

Step2(Machine Learning): Analyze with anomaly detection using unsupervised machine learning and detect suspicious commands as anomaly which attackers tend to use.

Figure 3 shows the flow of Step 1 and Step2.

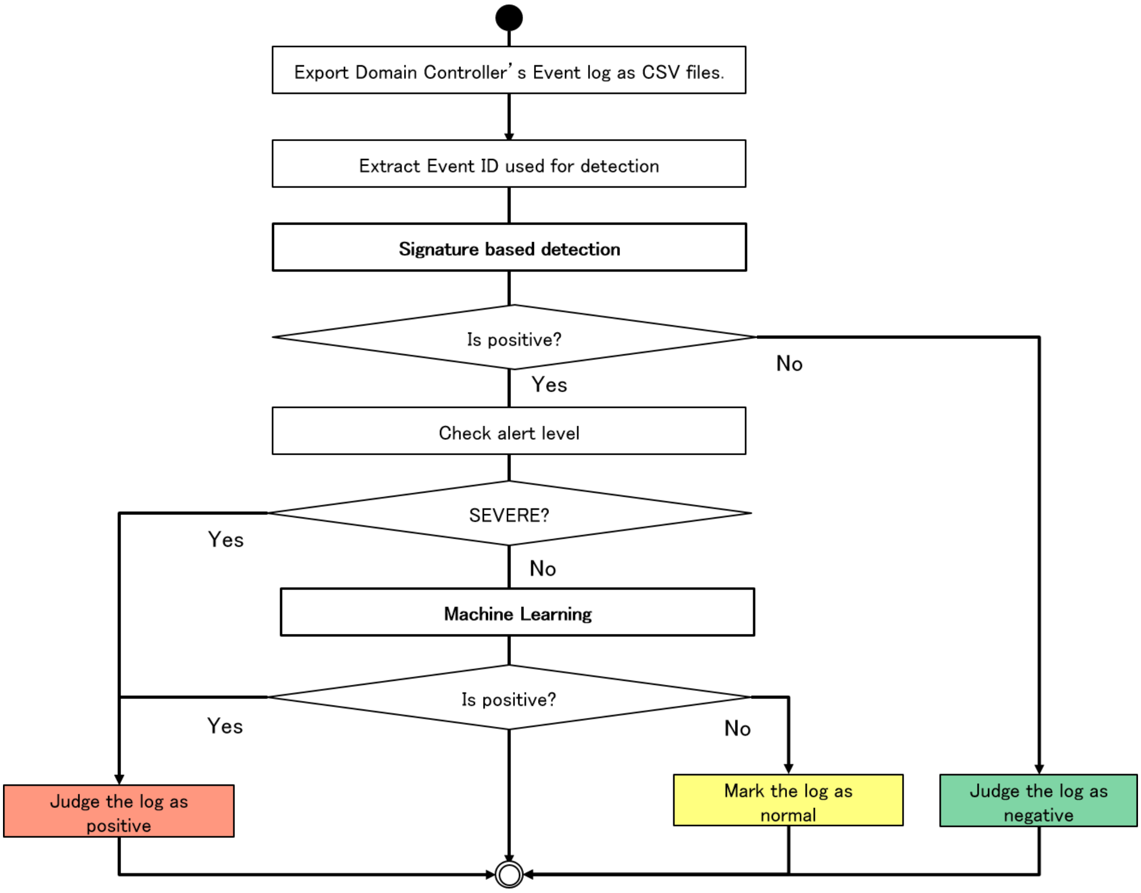
Export Domain Controller’s Event log as CSV files.
Extract Event ID used for detection
Signature based detection
Is positive?
No
Yes
Check alert level
SEVERE?
Yes
No
Machine Learning
Is positive?
Yes
No
Judge the log as positive
Mark the log as normal
Judge the log as negative

Figure 1. Steps of the proposed tool

Step3(Real-time alert): In order to realize real-time detection, Event logs are transferred to Elastic Stack which is an open source log analysis platform in real time. If attack activities are detected, send alert mail to the security administrator. Figure 4 shows the flow of all steps of the proposed tool.

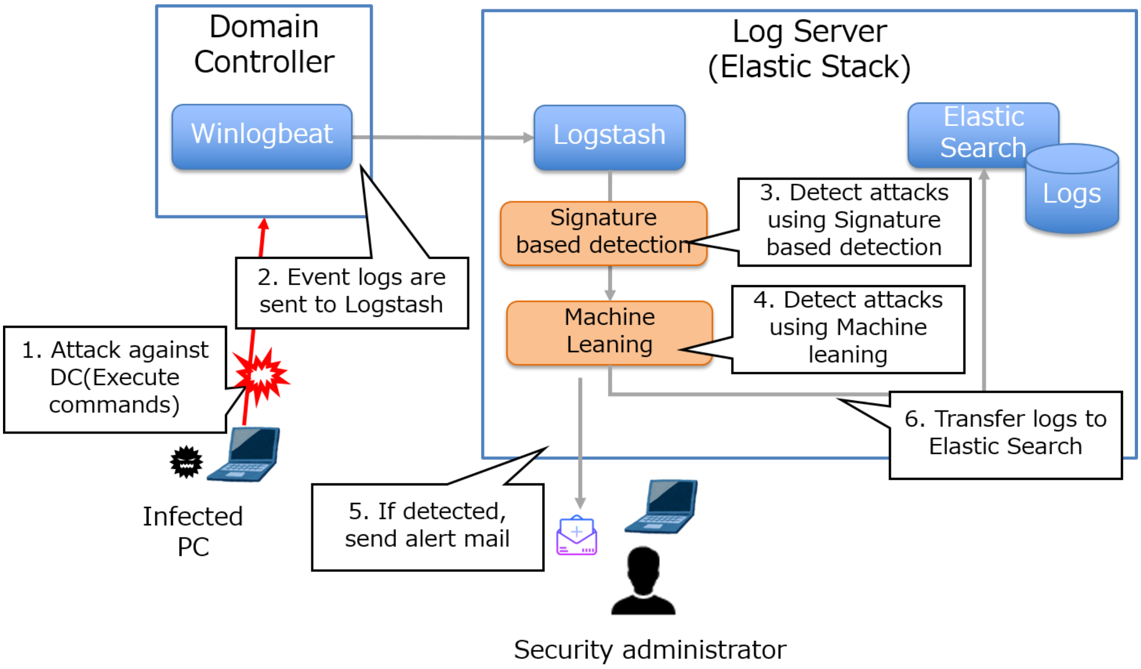
Domain Controller
Winlogbeat
Log Server (Elastic Stack)
Logstash
Elastic Search
Logs
Signature based detection
3. Detect attacks using Signature based detection
Machine Leaning
4. Detect attacks using Machine leaning
2. Event logs are sent to Logstash
1. Attack against DC(Execute commands)
Infected PC
6. Transfer logs to Elastic Search
5. If detected, send alert mail
Security administrator

Figure 4. Real-time alert using Elastic Stack

4.1. Event ID used for detection

In AD environment, DC uniformly handles the authentication of all users and computers, and logs related to authentication are stored in the DC's Event log. We use only Event logs (authentication log and process execution log) of DC for detection. Our proposed tool is practical since it uses only built-in Windows Event logs, so it is relatively easy to implement in an operating environment.

We use event ID related with Kerberos authentication and process creation shown in Table 1 and Table 2.

Table 1. Event ID used for detection

Event ID	description	consideration for detection	Logical Detection	Anomaly Detection
4672	Special privileges assigned to a new login	Information of accounts that have domain administrative privileges are recorded.	Use	Not Use
4674	An operation was attempted on a privileged object	Logged the process when the specified user exercised the special privileges.	Use	Use

4688	A new process has been created	Logged all processes executed	Use	Use
4768	A Kerberos authentication ticket (TGT) was requested	This event is recorded upon a TGT request. Therefore, when a Golden Ticket is used, this event is not recorded.	Use	Not Use
4769	A Kerberos service ticket was requested	When a service is accessed using a TGT including the Golden Ticket, this event is recorded.	Use	Not Use

Table 2. Data column in each Event ID used for detection

Column Name	4672	4674	4688	4768	4769
Account Name	Use	Use	Use	Use	Use
Client Address	-	*	*	Use	Use
Process Name	-	Use	Use	-	-
Object Name	-	Use	-	-	-
Service Name	-	Use	Use	Not Use	Not Use

*: Use information in Event ID:4769. Event ID 4674 and 4688 have no information of source IP address. The proposed tool specifies the Source IP address from Event ID 4769 just before Event ID 4674/4688 for each accounts. Because there is a high possibility that Service Ticket is requested before command/tools execution.

In Event ID:4688, information regarding execution of all processes such as process name including normal privilege users is recorded. On the other hands, In Event ID: 4674, specific processes executed with special privileges is recorded. The reason why we use both Event ID: 4674 and 4688 for detection is as follows.

- Information of temporary exe file(%SystemRoot%PSEXESVC.exe) which created on the destination computer when it accessed remotely by Psexec is recorded in Event ID:4674. The information is useful for detection of Psexec since the temporary file name is constant even attackers changed the file name.

- It is specific condition that execution of commands is logged in Event ID: 4674. For instance, when execute commands on a remote computer by using Psexec or wmic with loaded credentials on a source computer's memory. Attackers tend to load the stolen credentials in the same way to use the Golden Ticket and access to the target computer remotely.

4.2. Signature based detection

In order to minimize false negative rate, "Signature based detection" uses multiple indicators focusing on characteristics of attacks against AD especially attacks with the Golden Ticket. It detects a log as positive if a log matches any single indicator. If you have a list of Domain Administrator's accounts, we use the list (admin list) for detection.

Indicators for Signature based detection:

A) If a ST request event (Event ID: 4769) is recorded without a prior TGT request event (Event ID: 4768).

B) If administrative privilege use is recorded in Event ID:4672 which is not included in admin list.

C) Path information of executed commands and processes recorded in EventID:4674 is not system directory (such as “c:¥windows”)

D) If any specific commands that attackers tend to use is recorded in events with Event ID: 4674, 4688. we create a blacklist of commands which attackers tend to use (command blacklist) introduces in reference [C].

The specific detail of the detection algorithm is showed in Figure 5.

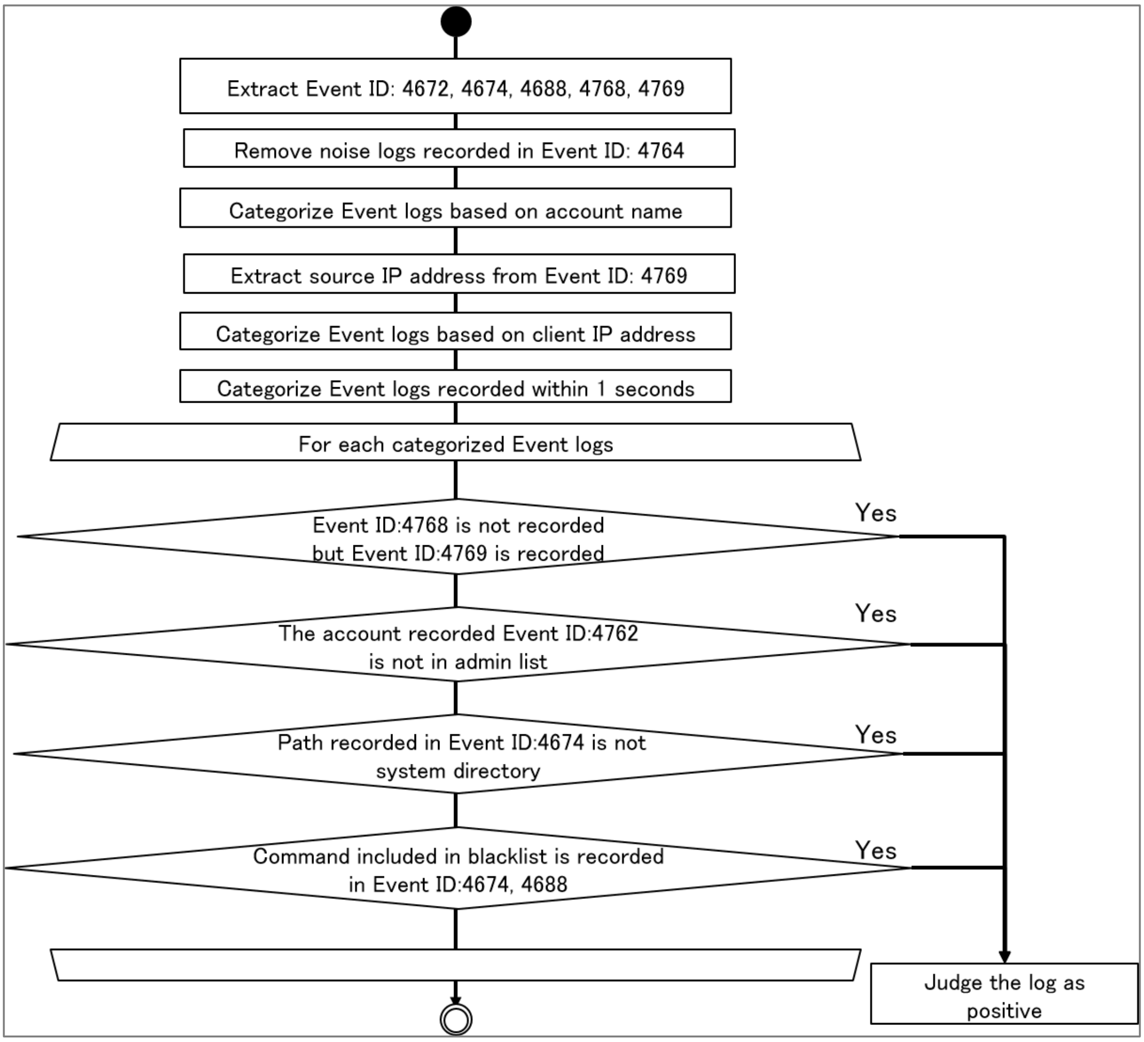
Extract Event ID: 4672, 4674, 4688, 4768, 4769
Remove noise logs recorded in Event ID: 4764
Categorize Event logs based on account name
Extract source IP address from Event ID: 4769
Categorize Event logs based on client IP address
Categorize Event logs recorded within 1 seconds
For each categorized Event logs
Event ID:4768 is not recorded but Event ID:4769 is recorded
Yes
The account recorded Event ID:4762 is not in admin list
Yes
Path recorded in Event ID:4674 is not system directory
Yes
Command included in blacklist is recorded in Event ID:4674, 4688
Yes
Judge the log as positive

Figure 5. The algorithm of Signature based detection

Analyzed events are exported in a result file. Alert level is placed on the logs which are judge as positive for the next step (Anomaly Detection).

Table 3. Alert level

Level	description
SEVERE	Matches any single indicator A OR B OR C
	Matches indicator D AND more than 80% of commands in blacklist are used
WARNING	Matches indicator D AND more than 20% of commands in blacklist are used
NOTICE	Matches indicator D AND less than 20% of commands in blacklist are used

Regarding indicator D, we judge based on the rate of used commands in the blacklist.
You should define appropriate percentage depending on your operation environment.
Next, we analyze the logs which are judged as positive and alert level is WARNING or NOTICE by Signature based detection.
Finally, we use Anomaly Detection to reduce false positive because the blacklist contains commands which also operators might use in regular operations such as “ipconfig”.

4.3. Anomaly Detection using unsupervised learning

In “Signature based detection”, a lot of false positive can be occurred depending on your operating environment (e.g. A Domain Administrator regularly use a part of commands which attackers also use) since it detects a log as positive if a log matches any single command included in the blacklist. On the other hand, “Machine Learning” can reduce false positive rate through analyzing the result of “Signature based detection”. Machine Learning detect anomaly behaviors through comparing with normal operation logs, so we can easily find out whether the result of “Signature based detection” is true positive or false positive.
Machine Learning gives computer systems the ability to "learn" with data without being explicitly programmed and recognizes pattern. It's divided into supervised and unsupervised learning. Supervised learning requires that the outputs are already known and that the data used for training should be labeled with correct answers. On the other hand, unsupervised learning doesn't require labels of correct answers.
We use unsupervised learning, because it is difficult to label the correct answers for attack detection, since users cannot differentiate logs recorded by attacks.

In Anomaly Detection, we analyze Event logs related to process (Event ID: 4674, 4688) which are judged as positive and alert level is WARNING or NOTICE by Signature based detection.

5. Verification of proposed tool

5.1. Verification tool

We verify whether attacks against Active Directory can be correctly detected using the proposed tool. The verification environment is shown in Table 4.

Table 4. Verification environment

	OS	Number of computers
DC	Windows Server 2008 R2	1
File server	Windows Server 2008 R2	1
Client Computer	Windows 7 (x64)	42

We conduct mock APT attack against the AD assuming that a legitimate Domain Administrator’s account is leveraged (e.g. steal Domain Administrator’s privilege, create the Golden Ticket for a legitimate Domain Administrator’s account, and steal information using the Golden Ticket). Commands and tools used during the attack is shown in Table 5. There is a possibility that attackers use other commands, however we use the least commands in order to accomplish the attack.

Table 5. Commands and tools used during the attack

Type	Commands and tools used during the attack
Attack tool	mimikatz
Tool provided by Microsoft	psexec
Built-in Windows command	wmic
Built-in Windows command	klist
Built-in Windows command	ipconfig
Built-in Windows command	hostname
Built-in Windows command	netstat
Built-in Windows command	net
Built-in Windows command	copy
Built-in Windows command	schtasks

We also conduct mock operations for verification of false detection. We verify two patterns of operations: operation only using GUI tools (GUI operation) and operation using both GUI and CLI tools including commands which attackers also use (GUI and CLI operation). We perform operations showed in Table 6 after login by a compromised Domain Administrator's account.

Table 6. Verification pattern

Pattern	Operations
GUI operation	Logon to DC using RDP. Then execute the following operations using GUI tools. Add users Delete users Password reset Check event viewer Share folder access Alter group policy setting
GUI and CLI operation	Logon to DC using RDP or Psexec. Then execute the following commands. ipconfig ping tasklist copy systeminfo net netstat whoami dir schtasks wmic wusa netsh sc

5.2. Verification of Anomaly Detection

We use unsupervised learning using scikit-learn (Python machine learning library) and verify detection rate depends on the algorithms shown in Table 7.

Table 7. Algorithms for verification of Anomaly Detection

Algorithm	Summary
One-Class SVM	An unsupervised algorithm that learns a decision function for novelty detection: classifying new data as similar or different to the training set.
IsolationForest	The IsolationForest 'isolates' observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature.
Local Anomaly Factor (LOF[FM1])	An unsupervised anomaly detection tool which computes the local density deviation of a given data point with respect to its neighbors.

5.2. Verification result

As the result, we found out "One-Class SVM" performed the highest detection rate. So, in this paper we describe the result using "One-Class SVM".[FM2]

The detection rate of the proposed tool is shown in Table 8.

Table 8. Detection rate of the proposed tool

Operation Type	Recall	Precision	F_Value
GUI operation	1.0	0.96	0.98
GUI and CLI operation	1.0	0.89 [FM3]	0.94

The result indicates it is possible to detect attacks with high accuracy by using the proposed tool even if legitimate Domain Administrator's account is leveraged and

The domain administrator regularly used some of commands which attackers tend to use. "GUI operation" especially achieve high detection rate, on the other hand, detection rate for "GUI and CLI operation" relatively tends to be lower compared with "GUI operation".

Remarks on the verification result

- We verified the deference of detection rate depending on Event ID (4674 and 4688). We found out that Event ID 4674 achieved higher detection rate, but recorded commands or tools were limited. Event ID 4674 records only "ipconfig", "hostname", "netstat" and "psexec" as far as our verification. Therefore, if attackers use other commands for attacking, false negative is occurred using only Event ID 4674.On the other hand, Event ID 4688 records more commands than Event ID 4674. We suggest using not only Event ID 4674 but also Event ID 4688 for detection in order to reduce false negative when attackers use commands which is not recorded in the Event ID 4674.

- False positive can be occurred if legitimated domain administrators use the commands which in not used in the daily operation. However, you can find out whether the result is false positive or not through monitoring logs in a short period and raise alert since administrators can easily compare the alert with their real operations.

- Duration of gathering Windows event logs for machine learning depends on environment. In our environment, we required the Windows event logs for about one week to achieve sufficient recall and precision.

6. Consideration for implementation

The followings are recommendations aspects of operation in order to detect attacks effectively using the proposed tool.

- Minimize the number of accounts who have Domain Administrator's privilege, and the computers which domain administrators use.

- Enter ID/password of the Domain Administrator account every time when you access the DC using Psexec because a trace of access using above login tool is not recorded in Event ID:4674. In case of Single Sign-On login (a login tool using loaded credentials on memory of the source computer), a trace of access is recorded in Event ID:4674 as same as attacker's access using malicious authentication ticket such as Golden Ticket. If Single Sign-On login is used in daily operation it is difficult to detect attacks as an anomaly since Event ID:4674 is recorded regularly. On the other hand, in case of login with ID /password, it is easy to detect Event ID:4674 recorded by attacks since Event ID:4674 is rarely recorded in the clean environment.

- Save the evidences of maintenance operations with Domain Administrator's privilege (e.g. date, used account and computer, operations). These evidences help administrators to judge whether detected operations are false positive or not.

1. Summary

It is difficult to detect attacks against AD since attackers tend to leverage legitimate accounts or tools. In this research, we will propose a tool for detecting attacks with combination of Signature based detection and Anomaly Detection focusing on Windows event logs related to Kerberos authentication and process creation. As a result, it is possible to detect attacks with high accuracy even if legitimate Domain Administrator's account is leveraged. In addition, even if the file name of attack tools were changed, our tool can detect the new file name as an anomaly.

Finally, the proposed tool is practical detection tool for mitigating damages of APT attacks since it uses only built-in Windows Event logs, so it is relatively easy to implement in running environments.

Reference

[A] Shingo Abe, Detecting Lateral Movement in APTs, https://www.first.org/resources/papers/conf2016/FIRST-2016-105.pdf

[B] JPCERT Coordination Center, Detecting Lateral Movement through Tracking Event Logs, https://www.jpcert.or.jp/english/pub/sr/ir¥_research.html

[C] JPCERT Coordination Center, Windows Commands Abused by Attackers, http://blog.jpcert.or.jp/2016/01/windows-commands-abused-by-attackers.html

[D] CERT-EU, Protection from Kerberos Golden Ticket, https://cert.europa.eu/static/WhitePapers/CERT-EU-SWP_14_07_PassTheGolden_Ticket_v1_1.pdf

[E] Institute of Information Industry Taipei, AD2: Anomaly Detection on AD Log Data for Insider Threat Monitoring,

[F] Markus Goldstein, Enhancing Security Event Management Systems with Unsupervised Anomaly Detection,

[G] Finding Advanced A*acks and Malware With Only 6 Windows Event ID's , https://conf.splunk.com/session/2015/conf2015_MGough_MalwareArchaelogy_SecurityCompliance_FindingAdvnacedAttacksAnd.pdf

[H] SANS Institute, Detecting Security Incidents Using Windows Workstation Event Logs

[I] One-class SVM with non-linear kernel (RBF), http://scikit-learn.org/stable/auto_examples/svm/plot_oneclass.html

[J] One-class SVM versus Elliptic Envelope versus Isolation Forest versus LOF, http://scikit-learn.org/stable/modules/anomaly_detection.html#one-class-svm-versus-elliptic-envelope-versus-isolation-forest-versus-lof

[K] IsolationForest example, http://scikit-learn.org/stable/auto_examples/ensemble/plot_isolation_forest.html#sphx-glr-auto-examples-ensemble-plot-isolation-forest-py

[L] Anomaly detection with Local Anomaly Factor (LOF), http://scikit-learn.org/stable/auto_examples/neighbors/plot_lof.html#sphx-glr-auto-examples-neighbors-plot-lof-py