electronics

MDPI

Article

Active Directory Attacks—Steps, Types, and Signatures

Basem Ibrahim Mokhtar [1], Anca D. Jurcut [2], Mahmoud Said ElSayed [2,*] and Marianne A. Azer [1,3]

[1] School of Information Technology & Computer Science, Nile University, Cairo 12566, Egypt
[2] School of Computer Science, University College Dublin, D04V1W8 Dublin, Ireland
[3] Computers and Systems Department, National Telecommunication Institute, Cairo 12577, Egypt
* Correspondence: mahmoud.abdallah@ucdconnect.ie

Abstract: Active Directory Domain is a Microsoft service that allows and facilitates the centralized administration of all workstations and servers in any environment. Due to the wide use and adoption of this service, it has become a target for many attackers. Active Directory attacks have evolved through years. The attacks target different functions and features provided by Active Directory. In this paper, we provide insights on the criticality, impact, and detection of Active Directory attacks. We review the different Active Directory attacks. We introduce the steps of the Active Directory attack and the Kerberos authentication workflow, which is abused in most attacks to compromise the Active Directory environment. Further, we conduct experiments on two attacks that are based on privilege escalation in order to examine the attack signatures on Windows event logs. The content designed in this paper may serve as a baseline for organizations implementing detection mechanisms for their Active Directory environments.

Keywords: Active Directory (AD); Advanced Persistent Threats (APT); attacks; golden ticket attack; Kerberos; pass the hash; security; silver ticket attack

check for updates

Citation: Mokhtar, B.I.; Jurcut, A.D.; ElSayed, M.S.; Azer, M.A. Active Directory Attacks—Steps, Types, and Signatures. *Electronics* **2022**, *11*, 2629. https://doi.org/10.3390/electronics11162629

Academic Editor: Andrei Kelarev

Received: 15 July 2022
Accepted: 17 August 2022
Published: 22 August 2022

1. Introduction

The Microsoft domain environment is one of the most critical systems in a corporate network [1]. The Windows domain directory services implementation known as Active Directory (AD) is used to offer services for user and identity management, authentication, and policy administration. It is the most used implementation of directory services and is utilized across the Windows ecosystem. Active Directory offers a variety of services. Among these services are [2]:

1. Domain services mediate communication between users and domains, save consolidated data, and offer search and login functions.
2. Certificate services issue, maintain, and distribute secure certificates.
3. Open (LDAP) protocol-compatible directory-enabled apps are supported by lightweight directory services.
4. Directory federation services enable single sign-on (SSO) for users to log in to a variety of online apps in a single session.
5. Rights management guards digital content that is subject to copyright by prohibiting its unlawful use and dissemination.
6. Domain name resolution is performed via the DNS service.

Active Directory (AD) is the cornerstone for both the offense and defense teams' daily operations and system administrators [3]. The fact that it provides and controls access to every resource in the organization attracts malicious actors and makes it a favorite target for cyber-attacks and especially Advanced Persistent Threat (APT) attacks [4]. The following explains the elements of the nomenclature as established by the United States Air Force (USAF) [5]. Advanced: the adversary is knowledgeable with intrusion tools and tactics and is capable of developing unique exploits. Persistent: the attacker plans to complete a mission, receive orders, and strike particular targets. Threat: the adversary is well coordinated, well funded, and motivated. Table 1 presents a detailed description of the main features associated with the APT, such as the APT nomenclature meaning [5], signs [6], lifecycle [7], techniques [8], types of targets comparison [5] with malware [8], and detection and protection measures [5].

Table 1. Advanced Persistent Attacks' main characterizing attributes and detection and protection measures.

Attribute	Main Features	Description
Nomenclature meaning	Advanced	The adversary is knowledgeable with intrusion tools and tactics and is capable of developing unique exploits.
	Persistent	The attacker plans to complete a mission, receive orders, and strike particular targets.
	Threat	The adversary is well coordinated, well funded, and motivated.
Signs	Players	Attacks are frequently carried out by actors who have a specific goal in mind. These actors usually have the support of nation-states or groups funded by corporations. APT28, OilRig, and Deep Panda are a few examples of groups.
	Goals	The goal is to continuously acquire intelligence while undermining target capabilities. This data theft or sabotage may be performed for strategic or political reasons.
	Timeliness	Attacks prioritize gaining access quickly and keeping it there for a prolonged period of time. Over the course of an attack, attackers frequently return to an infiltrated system many times.
	Resources	Planning and carrying out APT attacks requires a lot of resources. This covers the following: time, development and security skills, and hosting.
	Risk tolerance	Attackers are more likely to concentrate on specific targets rather than launch wide strikes. APT attackers are also more watchful to avoid detection or to cause unusual system activity.
	Methods	APT assaults frequently make use of complex methods that have a need for security knowledge. Rootkits, DNS tunneling, social engineering, and rogue Wi-Fi are a few examples of these methods.
	Attack source	APT assaults can come from many different places and can happen during an attack meant to divert security personnel. Before deciding on an entry point, attackers frequently take the time to completely map out a system's vulnerabilities.
	Attack value	The size of the target or the size of the attack operations are both considered to be components of attack value. APTs tend to target larger enterprises more frequently than smaller ones.
	Go-around detection tools	It is possible for APT assaults to go beyond typical detection methods, which depend on signature-based detection. Attackers achieve this by employing cutting-edge strategies such as fileless malware or tactics that let them hide their tracks.
Lifecycle	Initial reconnaissance	The phase in which the attacker researches a target, monitors the network, and accumulates data for use in subsequent attacks.
	Initial compromise	When an attacker successfully runs malware on one or more hosts within the target systems.
	Establish foothold	This is the phase in which the attacker takes control of newly compromised systems and keeps it that way.
	Escalate privileges	Increased access to systems and their data is attained at the escalate privileges stage by using various techniques including password hash dumping.
	Internal reconnaissance	During this phase, the attacker probes the compromised system to better understand its surroundings.
	Move laterally	The process by which an attacker uses his access to move across systems in a compromised environment.
	Maintain presence	The attacker secures ongoing access to the environment through techniques such as installing backdoors.
	Complete mission	This is the point at which the attacker achieves his primary objective, which is often data exfiltration from the target environment.

Table 1. *Cont.*

Attribute	Main Features	Description
Techniques	Social engineering	A user's information systems are compromised using social engineering. Instead of involving aleatory assaults on systems, this strategy targets individuals with privileged access, coercing them into disclosing personal information in order to carry out a harmful attack.
	Spear-phishing	This tactic focuses on one particular organization in an effort to obtain user passwords, financial information, or other private data.
	Watering hole	In terms of cyberespionage, it is comparable to spear-phishing. The attacks are customized to the victim's needs. Attackers attempt to learn information about the victim by taking into account his or her own interests in order to do this.
	Drive-by download	Using this method, malicious software is unintentionally downloaded and executed when a malicious website is accessed.
Types of targets	Government and public sector	Services are interrupted by government organizations, governmental policy, and private information leaks.
	IT	This involves IT service interruptions and members' private details being disclosed.
	Financial sector	This involves Financial Information Leakage in the Sector and Leaks of members' private information.
	Energy and electricity	Gas and electricity supplies are cut off and members' private information is leaked.
	Medical and health treatment	This is the disruption of health services and the disclosure of private medical data that is sensitive.
	Traffic and logistics	This involves system interruptions of the transportation and logistics networks.
	Manufacturing industry	This is IT system disruption in the manufacturing industry and the disclosure of business trade secrets.
Comparison with malware	Definition	APT attacks are smart, targeted, and well organized. Malware is harmful software that is used to attack and disable any system (for example, ransomware).
	Attacker	APT attackers can be government officials and organized crime organizations. Malware attackers are crackers.
	Target	APT attacks target organizations engaged in diplomacy, the information technology sector, and other industries. Malware targets any computer, personal or professional.
	Purpose	APT attacks aim to harm a specific target or filter private information, while malware attackers seek self-acknowledgment.
	Attack lifecycle	APT attacks attempt to maintain perseverance while trying various methods. Malware attacks come to an end when security measures discover it (e.g., anti-virus software).
Detection and protection measures	Email screening	The majority of APT attempts uses phishing to obtain first access. Email filtering and blocking malicious links or attachments within emails might help to thwart these intrusion attempts.
	Endpoint security	APT attacks entail the compromise of endpoint devices. Advanced anti-malware protection and Endpoint Detection and Response can assist in identifying and responding to an endpoint breach by APT attackers.
	Access control	Effective authentication safeguards and careful account management—paying particular attention to privileged accounts—can lessen the dangers of an APT.
	Examining traffic, user, and entity activity	This can assist in identifying penetrations, lateral movement, and exfiltration at various stages of an APT attack.

Most attacks start by infecting one single PC in the environment and then the attacker starts to target high-privileged accounts such as domain admins. Domain admins are targeted due to their ability to access nearly any resource in the organization. Usually, attackers try to employ many persistence techniques to stay in the environment with high privilege as long as possible. As the world is going rapidly to digitize every business and service, it becomes more crucial to employ security measures to prevent and defend against cyber-attacks. As per Kaspersky [9], the cost of a data breach is USD 1.41 M for enterprises and USD 108 K for small and medium businesses. With these facts and statistics in mind, the concern for Active Directory security is growing and many research efforts have been proposed to show the risk of Active Directory attacks and techniques to defend against them. This paper aims to highlight the techniques used to elevate privileges in Active Directory environments. In addition, we list mitigation techniques to detect, minimize, and avoid these types of attacks. Furthermore, we conduct experimental work on two AD attacks. The first is the pass the hash attack, and the second is the Kerberoasting attack.

The contributions of this paper are as follows:

1. Summarize the main characterizing attributes of the Advanced Persistent Threats by exploring their meaning, signs, lifecycle, techniques, types of targets, comparison with malware, as well as protection and detection mechanisms.
2. Illustrate the typical attack lifecycle for Active Directory and describe the different techniques.
3. Show the signatures for pass the hash attack and Kerberoasting attack.
4. Show how Kerberos protocol is exploited in various stages of the attack lifecycle.

The remainder of this paper is organized as follows. Section 2 provides a brief background on the Kerberos Authentication mechanism used in the Active Directory and targeted in most attacks. Section 3 presents the lifecycle of a typical attack on AD environments. In Section 4, we classify and survey the work done in defending and securing AD. Section 5 presents our experimental work. Finally, conclusions and future work are presented in Section 6.

2. Background

This section presents background about the workflow of Kerberos authentication protocol and the security issues of Kerberos.

2.1. Kerberos Authentication Workflow

Kerberos is the authentication protocol used in AD environments. It offers authentication for both client and server. Entities utilizing this protocol to authenticate and access resources are called principals. The service responsible for authenticating and authorizing principals is called the Key Distribution Center (KDC). The KDC comprises two subservices: the authentication server (AS) responsible for authentication and the ticket-granting service (TGS) responsible for issuing tickets. The KDC exists and runs in any writable domain controller (DC). Studying Kerberos's steps and workflow is crucial in understanding all AD attacks. Requesting access to a resource in Active Directory must follow the below procedure summarized in Figure 1 [10].

1. The client hashes the user's password. This hash is used as the secret key to secure communication between the client and the KDC.
2. The client sends an encrypted timestamp with the secret key to the AS, and the AS verifies that the client knows the user's password by decrypting the timestamp with the hash of the user's password existing in the AD database. If the decryption is performed successfully, this implies that the client knows the correct user's password.
3. The AS replies to the above request by sending two sections of information:
   (a) The encryption key used for subsequent requests to the KDC, which is encrypted with the user's hash.
   (b) The ticket-granting ticket (TGT) containing information about the user. This TGT is encrypted with the krbtgt account password which is known only by the TGS. This TGT is forwarded to the TGS in future requests for any service.
4. Using the TGT received previously, the client creates a request for a specific service and sends the request along with the TGT to the TGS.
5. The TGS decrypts the TGT and the request, verifies its legitimacy, and sends back a reply containing two parts to the client:
   (a) One part is a service ticket encrypted with the service secret key aimed to be sent to the server. This ticket contains the user's group information, a session key to communicate with the client, and a timestamp.
   (b) This part contains the session key to be used between the client and the server. This part is encrypted with the key received from the AS's reply in step 3.
6. The client sends a request for the remote server coupled with the service ticket received in the previous step. The server accepts this request if it decrypts the service ticket successfully with the key shared between it and the KDC, which implies that this request is authorized by the KDC.

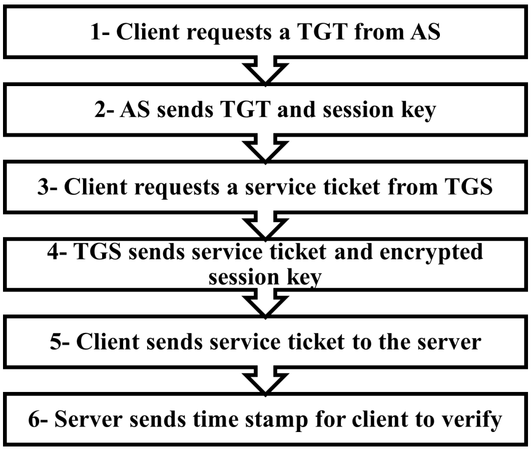

1- Client requests a TGT from AS
2- AS sends TGT and session key
3- Client requests a service ticket from TGS
4- TGS sends service ticket and encrypted session key
5- Client sends service ticket to the server
6- Server sends time stamp for client to verify

Figure 1. Kerberos Authentication Workflow. Adapted with permission from Ref. [10].

2.2. Kerberos Security Issues

The main discriminating feature of Kerberos is its stateless nature [11]. This feature means that the KDC with its two entities AS and TGS does not keep information about previous sessions [11]. All the information required by the TGS to process the requests is included in the TGT [11]. Since the TGT is encrypted by the krbtgt account password, the two entities which can decrypt the TGT are the AS, which issues the TGT, and the TGS, which issues service tickets after receiving the TGT [12]. These facts lead to two important conclusions. The first is that the krbtgt password is the most significant in the AD environment, and the second is that any information in the TGT is trusted, given that it is encrypted with the krbtgt password [13]. One more significant feature of the krbtgt account is that its password is rarely changed or expires, which means that if it is exposed to an attacker the system will be exposed for a very long time until the krbtgt password is changed [14].

3. Active Directory Attack Phases

Attacks on AD environments typically pass by specific stages and follow a certain path. The existing research efforts aiming to illustrate AD attacks assume that exploitation has happened and the attacker has a foothold in the environment [4,15]. In this paper, we follow the same approach, and we study the attacks on AD systems after obtaining the initial foothold. Once an adversary has access to one user or one machine, he starts to enumerate the domain to obtain all possible information to laterally move through the environment and escalate his privileges. The attacker then moves to the next stage, which is hunting for local or admin privileges. After obtaining admin privilege, the attacker looks for ways to persist in the environment as long as possible in order to achieve its objectives such as data exfiltration, denial of service, or any other target. The main stages in the AD lifecycle are shown in Figure 2.

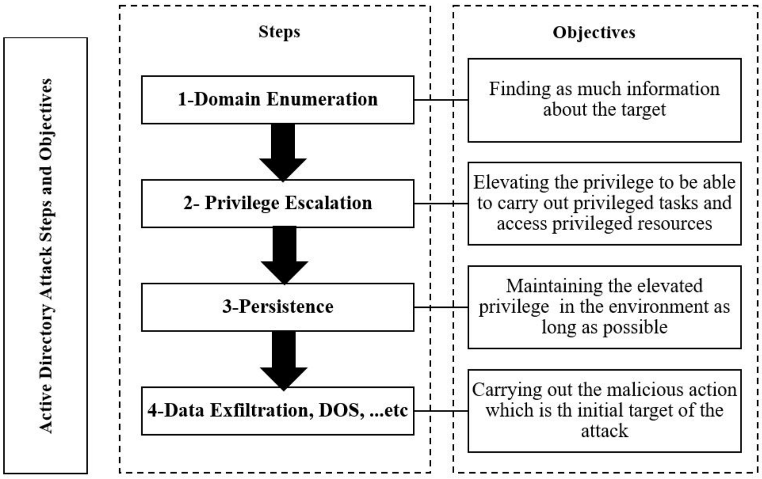

Active Directory Attack Steps and Objectives
Steps
1-Domain Enumeration
2- Privilege Escalation
3-Persistence
4-Data Exfiltration, DOS, ...etc
Objectives
Finding as much information about the target
Elevating the privilege to be able to carry out privileged tasks and access privileged resources
Maintaining the elevated privilege in the environment as long as possible
Carrying out the malicious action which is th initial target of the attack

Figure 2. Active Directory attack steps.

3.1. Domain Enumeration

Once an attacker has gained a foothold in an AD environment, the end goal is to elevate his access by targeting one of the privileged accounts such as domain admins or enterprise admins [16]. To achieve this goal, the attacker tries to perform domain enumeration to know the weak points or vulnerabilities which he can abuse to gain privileged access. Unfortunately, it is easy to perform most domain reconnaissance operations with any regular user without elevated privileges [16]. Attackers can use several tools to perform domain reconnaissance, of which some are native Microsoft tools and others are made by security researchers. The following is an overview of four known tools that are used in domain enumeration.

1. **Net.exe:** The net.exe is a built-in tool in Windows that can be used to carry out tasks on groups, users, accounts, policies, etc. [17]. Using the net.exe tool, the attacker can see all the attributes of users and groups and hunt for critical groups such as AD domain admins and can then see information about the user's group membership.
2. **Active Directory module:** Active Directory [18] allows administrators to query and make changes to Active Directory with PowerShell. It comprises a set of PowerShell cmdlets that can retrieve a lot of information about the current domain [19]. While this module needs remote server administration tools (RSAT) installed [19], there are ways in which the attacker can leverage to import this module into any workstation [20].
3. **Powerview:** Powerview is part of the Powersploit framework, and it is a PowerShell tool used in domain enumeration that can act as a replacement for Windows net commands [21]. It can provide the same functionality as the AD module. It also has some extra functions which the attacker can leverage to identify the locations in the network where specific users are logged in to [22].
4. **Bloodhound:** Bloodhound is a GUI tool used by red and blue teams to visualize AD domains and to facilitate the identification of complex attack paths and gain deep knowledge about privilege relationships in AD systems [23].

It is worth mentioning that the output of the reconnaissance phase is very broad and the information which can be retrieved is huge and depends on the time spent on this phase. Attackers usually grab information about the current domain, current forests, privileged groups such as domain or enterprise admins, interesting Access Control List (ACL) entries, domain structure, group policies applied, sensitive files, shares available, and machines that have active sessions for specifically targeted users [24].

3.2. Privilege Escalation

The second stage of a successful attack on an AD environment is escalating privileges, since in several attacks the intruders hunt for local admin privilege to facilitate moving to the domain admin. In this section, we go through the techniques and attacks used to escalate privileges to domain admin, as they are more dangerous and can control the entire domain if exploited successfully.

1. **Pass the hash attack:** The pass the hash (PTH) attack has been evolving for years. It depends on hash extraction from a compromised machine and using this hash to create tokens that allow access to sensitive resources or hosts across the domain. While Microsoft has been implementing many security measures in each new Windows version to prevent PTH attacks, new techniques evolve continuously to carry out this attack successfully [25].
   Windows hosts store credentials in the form of NT hashes [25]. In PTH attacks, adversaries use the fact that NTLM hashes can be used for authentication without using the user's password. PTH involves two stages: (1) the first stage is extracting the hashes of certain users depending on their availability and (2) the second stage is creating tokens using extracted hashes by tools such as Mimikatz [26]. It is important to note that PTH can give attackers privileges equivalent to the user's password without having to know the actual password. Attackers usually try to extract high-privileged accounts' passwords and use them in PTH attacks to escalate their privilege. The most common technique to carry out stage (1) is to dump the credentials in the memory of the Local Security Authority Subsystem Service (LSASS) process. LSASS stores information in its memory about all accounts that are used actively including the NT hash [27]. These hashes are used to provide a single sign-on experience to users in order to avoid entering the password each time the user wants to access a resource [28]. Extracting the hash of a domain admin account provides the attacker with the ability to access the same resources granted to the domain admin user without knowing its password. The most secure measure to protect against PTH attacks is to protect the memory of the LSASS process and prevent accessing it [29]. This feature is implemented in Windows Defender Credential Guard (WDCG) which makes the LSASS process not accessible to malicious applications [29]. While this feature prevents hash extraction from LSASS memory, other ways can be used to extract the hashes [30].
2. **Kerberoasting attack:** Kerberoasting is an attack relying on the fact that some services run under normal user accounts. If a service runs under a user account, the service ticket provided by AD for this service is encrypted by the NTLM hash of the user's password. Grabbing the ticket and brute forcing it allows access to the plaintext password of the target user. The service principal name (SPN) attribute of a user indicates if a service is running under this user account. The procedure for performing a Kerberoasting attack is shown in Figure 3, and it is carried out as follows:
   The attacker scans the AD environment for any user who has the SPN attribute defined as PowerShell, LDAP queries, or any custom tool such as Powersploit. After grabbing the target account, the user asks the AD for a service ticket using the SPN value of the target account. Exporting the ticket to the disk and brute forcing it reveals the user's password, which is the final stage. Another variation of the Kerberoasting attack is the targeted Kerberoasting attack, which can be of two types:

   - The first is where the attacker grabs the AS-REP Kerberos messages which are encrypted by the user's hash. For this attack to happen, the UserAccountControl settings must contain the "Don't require Kerberos pre-authentication" setting to be enabled.
   - The second is when the attacker has a high privilege to set the SPN value of any user to any dummy service and then request a service ticket for this service which can then be cracked to obtain the user password.

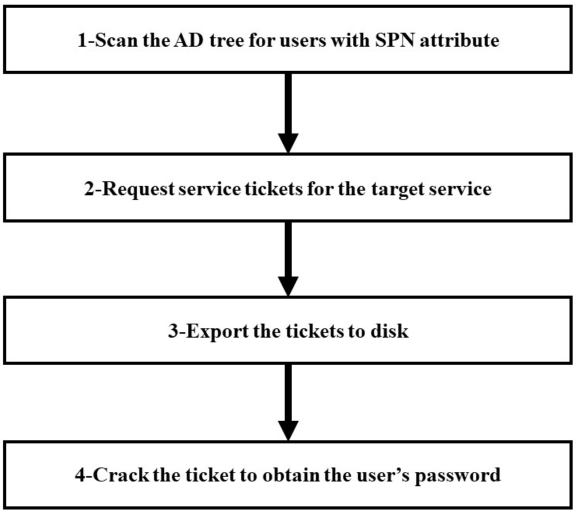

1-Scan the AD tree for users with SPN attribute
2-Request service tickets for the target service
3-Export the tickets to disk
4-Crack the ticket to obtain the user's password

Figure 3. Kerberoasting attack steps.

Mitigations for this attack include enforcing a strong and complex password policy, enabling Kerberos encryption rather than RC4, and limiting the privileges for service accounts [1].

3. **Constrained and unconstrained delegation abuse:** Delegation is an Active Directory feature that permits the impersonation of an account by users or computers [31]. Enabling this feature is available through the delegation tab on a user or computer account [32]. Through this tab, constrained and unconstrained delegation can be enabled and customized.

3.3. Unconstrained Delegation

Unconstrained delegation is the permission for a computer to impersonate any service. Once it is enabled, any time a user connects to this computer, their TGT is stored in the computer memory for later use [33]. The attacker compromises a computer and tricks any privileged user such as the domain admin to log in on the compromised computer. The attacker then extracts the TGT of the privileged user and uses it to access any service on the domain. The previous scenario is just a simple attack, and other sophisticated attacks may occur by abusing this feature in different ways [34]. Two rights need to be given for a user to be able to manage delegation [33]. The first is the `SeEnableDelegationPrivilege` user right, which gives the user the privilege to enable delegation for a certain computer. The second is the right to update `msDS-AllowedToDelegateTo` and `User AccountControl` attributes for a computer. These attributes contain the delegation settings of a computer. An attacker can use Powerview or the Active Directory module to enumerate and know which user has the right to delegate and know which computers have delegation enabled. `Mimkatz` is another tool that can be used to export the tickets on the compromised computers. Common mitigations for this attack include allowing only certain SPNs for delegation and to place privileged users in protected groups to prevent their TGTs from being used in delegation.

1. **Constrained delegation:** Unlike unconstrained delegation, constrained delegation provides access to a specific set of services listed in the]`msDS-AllowedToDelegateTo` attribute on the user configuration. The steps for abusing the constrained delegation are illustrated in Figure 4 and are as follows. The attacker compromises an account with constrained delegation enabled by dumping its hash, using Mimikatz for example. The attacker then can request a TGT for the compromised account and then a TGS to access any service listed on the `msDS-AllowedToDelegateTo` attribute as any privileged user.

One interesting fact about this attack is that the SPNS is not checked when requesting TGS, which leads the attacker to access any service running under the same account. For example, if the CIFS service is listed in the `msDS-AllowedToDelegateTo` attribute then the attacker can access any service which is running under the same account as the CIFS service [35]. The above fact can lead to the compromise of the entire domain if exploited successfully.

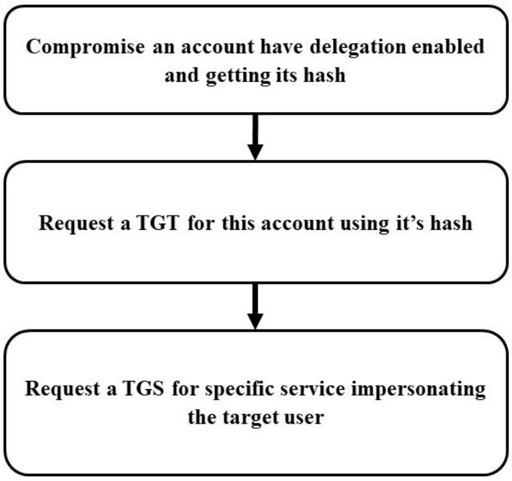

Compromise an account have delegation enabled and getting its hash
Request a TGT for this account using it's hash
Request a TGS for specific service impersonating the target user

Figure 4. Constrained Delegation Abuse Flow.

Mitigations for constrained delegation abuse typically put critical accounts in the protected users group or mark the account as sensitive and cannot be delegated on the account tab [36].

2. **Abusing DNS admins' privileges:** Abusing DNS admins' privileges is a technique where a DNS admin member can escalate their privileges to system privilege if he has the rights to configure server-level plugin DLL and to restart the DNS service [36]. By injecting a malicious DLL executing a reverse shell to the attacker machine into the dns.exe service and restarting the service, the attacker can have a shell on the DNS server as a system account which is the highest privileged account.
While the DNS admins are not granted the necessary privileges by default, it is common for DNS admins to be given the right to restart the DNS service and to inject the DLLs in the `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\DNS\Parameters\ServerLevelPluginDll` registry.
The stages for performing this attack are shown in Figure 5 and are illustrated as follows: The attacker compromises a machine having a ticket for a DNS admin user and having a local admin privilege on it. The attacker uses the ticket to inject the malicious DLL file containing a reverse shell in the ServerLevelPluginDll registry. Next step is to force a restart for the DNS service. Once the DNS service is restarted, the dns.exe executes the malicious DLL as the system user and returns a reverse shell to the attacker as the system account. If the DNS server in the environment is the same domain controller server, the above attack leads to the compromise of the entire domain. Common mitigations for this attack are monitoring the DNS service status and the ServerLevelPluginDll registry for any changes and limiting the privilege to load DLL files to only specific users, not all the DNS admins, if required.

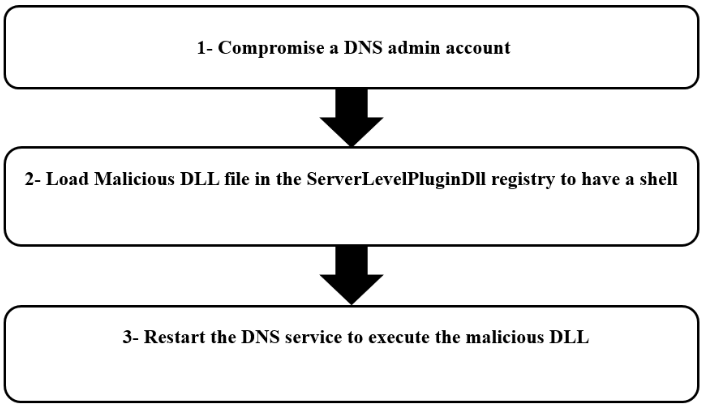

1- Compromise a DNS admin account
2- Load Malicious DLL file in the ServerLevelPluginDll registry to have a shell
3- Restart the DNS service to execute the malicious DLL

Figure 5. DNS Admins' Privileges Escalation Flow.

3. **Domain persistence** The next step after escalating privileges in an AD attack is to persist as long as possible with an elevated privilege in order to achieve the attack target. While there are many techniques of postexploitation persistence in the AD system, some of the techniques are common, while others are rare and have many limitations. In this section, we illustrate the common attacks by which an adversary persists in the AD environment. We go through the golden and silver ticket attacks in detail, as they are more common and give brief information about some other tactics which are less common and have more limitations.

    (a) Golden ticket attack: Golden tickets are user-created TGTs which are used to give the attackers access to specific resources. Intruders use the TGTs to obtain service tickets from the domain controller without verifying the contents of the TGT [37]. In Kerberos protocol, the TGT is used to prove to the KDC service on the domain controller that the user is authenticated to the DC. The TGT is encrypted with the krbtgt account NTLM hash. The way the KDC verifies the integrity of the TGT is by trying to decrypt it with the krbtgt hash. Once it is decrypted successfully, the KDC grants the user the service tickets without verifying the contents of the TGT unless the TGT's age is older than 20 min [38]. Given the above behavior, an attacker who has domain admin privileges to the DC can follow the steps in Figure 6 to dump the krbtgt hash and use it to forge a TGT impersonating any user and persisting with high privileges for a very long time. Interesting facts about this attack are that the attacker can impersonate any user, even if the user does not exist in the domain on any machine joined or not joined to the domain, and also that the krbtgt password is rarely changed [38]. Once this attack is carried out successfully, the only way to stop it is to change the krbtgt account password twice. The dumping and generation of golden tickets are usually conducted using the Mimikatz tool, which provides the ability to modify the security identifier (SID) history attribute in the TGT. The SID is an identifier that is used by the AD system to verify the privilege of a user, and the SID history is an attribute that includes more than one SID. This attribute can be exploited to include the SIDs of various domains and exploit the entire forest.

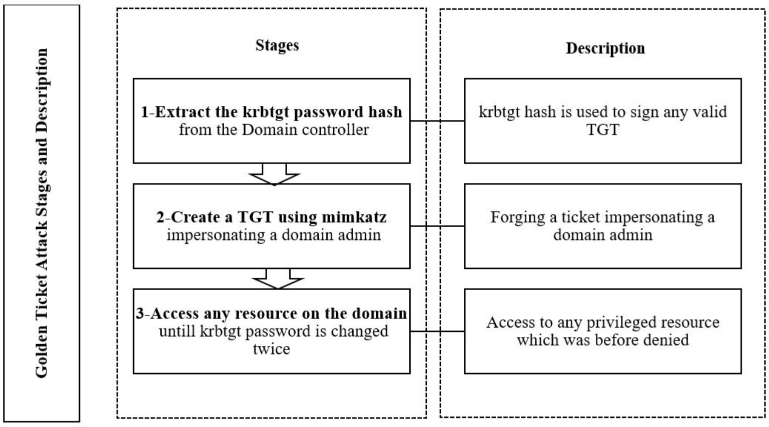

Golden Ticket Attack Stages and Description
Stages
Description
1-Extract the krbtgt password hash from the Domain controller
krbtgt hash is used to sign any valid TGT
2-Create a TGT using mimkatz impersonating a domain admin
Forging a ticket impersonating a domain admin
3-Access any resource on the domain untill krbtgt password is changed twice
Access to any privileged resource which was before denied

Figure 6. Golden Ticket Attack Stages.

(b) Silver ticket attack: Silver tickets are service tickets that are used to access certain services, unlike golden tickets which are TGTs providing access to all services. The flow for the silver ticket is presented in Figure 7. The attacker gets access to the NTLM hash of a service account and then uses a tool such as Mimikatz to forge a service ticket to access this service, impersonating any user.

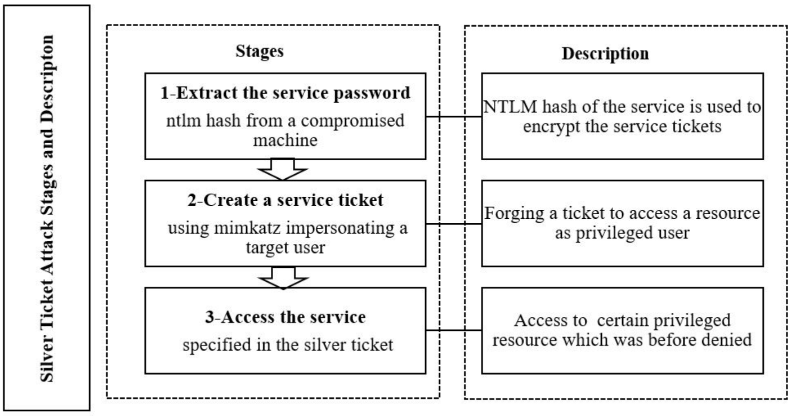

Silver Ticket Attack Stages and Descripton
Stages
Description
1-Extract the service password ntlm hash from a compromised machine
NTLM hash of the service is used to encrypt the service tickets
2-Create a service ticket using mimkatz impersonating a target user
Forging a ticket to access a resource as privileged user
3-Access the service specified in the silver ticket
Access to certain privileged resource which was before denied

Figure 7. Silver Ticket Attack Stages.

The authors in [39] showed how to use a silver ticket attack to compromise the domain controller again after the golden ticket attack has been stopped by resetting the krbtgt password twice. The attack is carried out against the host service of the domain controller to create a scheduled task to dump the credentials using Mimikatz and obtain the krbtgt password again. While the silver ticket attack has a narrower scope than the golden ticket attack, according to [39], it has a more dangerous effect, as it is easier to be performed and its technique does not involve communication with the domain controller, which makes it stealthier and harder to detect. The best way to prevent silver ticket attacks is to enforce security controls and use complex passwords for service accounts to minimize the probability of their exposure [39].

(c) Skeleton key attack: Skeleton key is an attack carried out against the domain controllers. The adversary targets the Local Security Authority Subsystem Service (LSASS) process in the domain controller [40]. The LSASS process is responsible for the entire system of authentication in the Active Directory. The attacker patches the LSASS process to create a master password to work for any account. Although the attack is not compatible with Windows Server 12 or later, it still possesses a great risk, as a lot of companies still use outdated versions of Windows [40]. The procedure for carrying out the skeleton key attack is as follows. The attacker compromises the domain controller and acquires the domain admin privileges. He then runs mimikatz.exe "privilege::debug" "misc::skeleton" exit on the domain controller. After running successfully, a master password is created which enables login with any account with password Mimikatz. Upon running, the skeleton key carries out the tasks shown in Table 2.

i. Check for one of the following compatible 64-bit versions. The malware does not work with 32-bit Windows versions or with Windows Server versions beginning with Windows Server 2012 (6.2).
ii. Use the SeDebugPrivilege function to acquire the necessary elevated privileges to write to the (LSASS) process. This process controls user account authentication.
iii. Enumerate available processes to acquire a handle to the LSASS process.
iv. Obtain addresses for authentication-related functions that should be patched.
    A. CDLocateCSystem—placed in cryptdll.dll.
    B. SamIRetrieveMultiplePrimaryCredentials—placed in samsrv.dll.
    C. SamIRetrievePrimaryCredentials—placed in samsrv.dll.
v. Perform OS-specific modifications using the global variable set during the compatibility check in step 1.
vi. Use the OpenProcess function to get a handle to the LSASS process.
vii. Reserve the required memory space to alter and patch the LSASS process's memory.
viii. Patch relevant functions depending on the operating system:
    A. CDLocateCSystem (all compatible Windows versions).
    B. SamIRetrieveMultiplePrimaryCredentials (only Windows 2008 R2 (6.1)).
    C. SamIRetrievePrimaryCredentials (all compatible Windows versions other than Windows 2008 R2 (6.1)).

After rebooting the DC, the LSASS process memory is refreshed and the master password is no longer valid. In addition, for this attack to operate successfully without error, the skeleton key needs to be installed on all domain controllers to answer any logon attempt successfully [41]. The best prevention for this attack is to limit the number of domain admins and ensure they cannot log in to less protected machines where their credentials can be dumped [29].

(d) DSRM persistence: The directory services restore mode (DSRM) account is a local administrator account existing on every domain controller. The DSRM password is set at the time of promoting the DC and it is seldom changed [42]. The main function of the DSRM account is to provide the administrator with a backdoor to the database to recover or repair any failure. While this account does not provide access to the domain or its services by default, the attacker can manipulate its configuration to have local admin access to the domain controller and compromise the entire domain [43]. The attacker performs this attack using Mimikatz by dumping the SAM database of the domain controller and performing the pass the hash attack using the NTLM hash acquired to maintain a valid TGT, providing access to all domain services. The mitigation technique for this attack is to make sure that the registry key DsrmAdminLogonBehavior does not exist or its value equal to 1 to disable logging in with the DSRM account [42].

(e) SSP persistence attacks: The security support provider (SSP) is an API used by Windows to carry out the authentications of Windows login. It is a DLL file that provides security packages to other applications. This DLL stacks itself up in LSA when the system starts, making it a start-up process. After it is loaded in LSA, it can access all the Windows credentials. An SSP attack can be carried out using Mimikatz by issuing the MemSSP command, and this command registers a malicious SSP on a Windows host [44]. The injected SSP logs all the logged-on user's passwords in cleartext [44]. Once an attacker has domain admin privileges to a domain controller, he can issue this malicious command to obtain the passwords of another privileged account who has logged on to the domain controller. To detect this attack, the `c:\Windows\System32` path for a log file called mimilsa.log file which is produced by Mimikatz should be monitored [44].

(f) Access Control List abuse: Access rights for any object in the AD environment are provided through an Access Control Entry (ACE). An ACE defines the access or audit permission granted for a specific user or group. An ACL is an attribute for any object in AD. Each ACL contains a group of ACEs defining the right for each user to access this object [45]. Modifying the ACLs of certain objects in the AD tree can give the attacker access to resources that cannot be accessed by default. In this section, we illustrate uses by cyber criminals to modify the ACLs of objects in common techniques in AD to persist in the environment and gain high privilege.

Table 2. Skeleton Key Attack Operations.

Skeleton Key Attack Step	**Comment**
1- Search for a 64-bit version that is compatible.	The malware is incompatible with 32-bit Windows. versions or with Windows Server versions beginning with Windows Server 2012 (6.2).
2- Utilize the SeDebugPrivilege command.	This procedure is in charge of user account authentication. to gain the necessary higher rights to write to the (LSASS) process.
3- List all existing procedures.	This is done to get a hold on the LSASS process.
4- Find the addresses of the functions that are linked to authentication and need to be patched.	Examples of these functions are: - CDLocateCSystem — placed in cryptdll.dll. - SamIRetrieveMultiplePrimaryCredentials — placed in samsrv.dll. - SamIRetrievePrimaryCredentials — placed in samsrv.dll.
5- Make changes that are particular to the OS.	This is carried out with the help of the global variable that was set during the compatibility check in step 1.
6- Use the OpenProcess function.	This is performed to get a handle to the LSASS process.
7- Set aside the needed memory space to change and fix the LSASS process's memory.	This is conducted to alter and patch the LSASS process's memory.
8- Patch relevant functions.	This depends on the operating system: a- CDLocateCSystem (all compatible Windows versions). b- SamIRetrieveMultiplePrimaryCredentials (only Windows 2008 R2 (6.1)) c- SamIRetrievePrimaryCredentials (all compatible Windows versions other than Windows 2008 R2 (6.1)).

3.4. Abusing AdminSDHolder ACL

AdminSDHolder is an object in the AD tree whose distinguished name is (cn = adminsdholder, cn = system, dc = domain, dc = com). The function of this object is to provide a security template for AD objects which are members of privileged groups. Being very critical, any accidental or intentional modification of the ACL of these groups poses a great risk to the entire AD environment. The AdminSDHolder ACL overwrites each ACL of these privileged groups every 60 min to maintain the tight-security ACLs imposed by the administrators on these objects. The default ACLs for AdminSDHolder are as follows: Authenticated Users: Read; SYSTEM: Full Control; Administrators: Modify; Domain Admins: Modify; Enterprise Admins: Modify.

The attack is conducted by adding a generic total permission to a certain user in the ACL of AdminSDHolder. The attacker then waits for 60 min while the ACL is propagated to the protected groups. Once the ACL is applied, the target user is now part of the domain admins group and can modify its members [46]. The detection of this attack can be accomplished by continuously monitoring the AdminSDHolder ACL and monitoring the protected group members' addition of modification [46].

3.5. Remote Access Methods for ACL Abuse

There are several remote access methods available in the AD system. Each remote access method is controlled by many ACLs which can be abused by attackers to provide remote privileged access for intruders.

The most common remote access methods are Windows Management Instrumentation (WMI) and PowerShell remoting (PS remoting). The two methods give AD administrators the ability to issue a command and control the AD machines and feature remotely. In this section, we illustrate in brief how the ACLs of the above methods can be abused to maintain persistence in the AD environment.

- WMI ACL abuse: WMI is the Microsoft implementation of web-based enterprise management (WBEM). Data in WMI are grouped into WMI classes. For example, there is a WMI class for logical drives (Win32_LogicalDisk), and there is a class for running processes (Win32_Process). WMI classes are then grouped into WMI namespaces. Most of the WMI classes exist under the root\cimv2 WMI namespace, but there are other namespaces (for example, root\MicrosoftExchangev2) that contain WMI classes related to a specific Microsoft application. To find data, an administrator can navigate through the hierarchy. For example, if there is a need to find the amount of free disk space on a logical disk, navigation should be performed as follows.
  ```
  root\cimv2 namespace>Win32_LogicalDisk WMI>
  ```
  A WMI namespace is just a container for WMI classes. By keeping different WMI classes in different WMI namespaces, Windows allows users to specify different access permissions for different WMI classes. Connecting to a remote computer using WMI requires the correct distributed component object model (DCOM) settings and WMI namespace security settings to be enabled for the connection. ACLs exist for DCOM endpoints and for WMI namespaces to ensure authorized users access the resources only [47]. An adversary having domain admin privileges can modify the ACLs of the DCOM endpoint and the WMI namespaces to allow connecting to different computers using WMI. For example, the below PowerShell command allows WMI access to ops-mssql computer for lab users without being a member of any privileged group [47].
  ```
  Set-RemoteWMI -Compute
  rName ops-mssql -SamAccountName labuser -Ve
  rbose
  ```
  It is worth mentioning that WMI ACL modification does not leave any event logs, but 4624 and 4634 logs appear when using WMI access [47].
- PS remoting ACL abuse: PowerShell remoting is a Windows feature that allows interactive sessions with remote computers, running commands or scripts, and starting a persistent session with a remote computer [48]. By default, administrator privilege is required to use PowerShell remoting. However, an attacker having domain admin privileges can modify the ACL of the PowerShell remoting endpoint to give a user read, write, and execute permissions. Once these permissions are granted, the user can connect to remote computers and run commands or scripts without having domain admin privilege [47]. As for WMI access, accessing machines using PS remoting does not leave any logs except 4264 and 4634 when logging in and out.

Table 3 summarizes the AD attacks stages and describes the main attacks introduced in this section.

Table 3. Active Directory Attacks Stages and Techniques Description.

Stages	Techniques	Description
Domain enumeration	Net.exe	This allows attackers to see all the attributes of users and groups and hunt for critical groups such as AD domain admins and can then see information about the user's group membership.
	AD module	This allows administrators to query and make changes to Active Directory with PowerShell.
	Powerview	This has some extra functions which the attacker can leverage to identify the locations in the network where specific users are logged into.
	Bloodhound	This helps to visualize AD domains and to facilitate identifying complex attack paths and gain deep knowledge about privilege relationships in AD systems.
Privilege escalation	Pass the hash	It depends on hash extraction from a compromised machine and using this hash to create tokens that allow access to sensitive resources or hosts across the domain.
	Kerberoasting	This relies on the fact that some services run under normal user accounts. The service ticket provided by AD for this service is encrypted by the NTLM hash of the user's password. Grabbing the ticket and brute forcing it allows access to the plaintext password of the target user.
	Delegation abuse	Delegation is an Active Directory feature that permits the impersonation of an account by users or computers. Constrained and unconstrained delegation can be enabled and customized. Constrained delegation provides access to a specific set of services listed in the msDS-Allowed To Delegate to attribute on the user configuration. Unconstrained delegation is the permission for a computer to impersonate any service. Once it is enabled, any time a user connects to this computer, their TGT is stored in the computer memory for later use.
	Abusing DNS admins' privileges	Abusing DNS admins' privileges is a technique by which a DNS admins member can escalate their privileges to system privileges if he has the rights to configure server-level plugin DLL and to restart the DNS service.
Domain persistence	Golden ticket	Golden tickets are user-created TGTs which are used to give the attackers access to specific resources. Intruders use the TGTs to obtain service tickets from the domain controller without verifying the contents of the TGT.
	Silver ticket	Silver tickets are service tickets that are used to access certain services. The attacker gets access to the NTLM hash of a service account and then uses a tool such as Mimikatz to forge a service ticket to access this service, impersonating any user.
	Skeleton key	Skeleton Key is an attack carried out against the domain controllers. The adversary targets the Local Security Authority Subsystem Service (LSASS) process in the domain controller. The LSASS process is responsible for the entire system of authentication in the Active Directory. The attacker patches the LSASS process to create a master password to work for any account.
	DSRM persistence	The main function of the DSRM account is to provide the administrator with a backdoor to the database to recover or repair any failure. The attacker can manipulate its configuration to have local admin access to the domain controller and compromise the entire domain.
	SSP persistence	The security support provider (SSP) is an API used by Windows to carry out the authentications of Windows login. It is a DLL file that provides security packages to other applications. This DLL stacks itself up in LSA when the system starts, making it a start-up process. After it is loaded in LSA, it can access all the Windows credentials.
	ACL abuse	Modifying the Access Control Lists of certain objects of the AD tree can give the attacker access to resources that cannot be accessed by default.

4. Related Work

In this section, we review and classify the existing research work in the Active Directory security, detection, and prevention of various AD attacks.

4.1. Defending against Credential Theft Attacks

The authors in [49] conducted three experiments to see the effects of disabling a compromised user, disabling a compromised computer, and rebooting the system. They concluded that disabling a compromised user or computer does not have an effect if the issued tickets are still valid. Contrary to the first two methods, the third method which is rebooting the system yields the disabled account to be banned from logging on in Windows 2012 and later only. A study of PTH attacks was presented in [27,50], where the authors illustrated the full cycle of PTH attack. In [27], the authors concluded that multilayer security defense is the most useful approach to mitigate PTH attacks. They also provided some measures such as limiting local admin privileges to users and using Windows firewall to prevent lateral movement, but they confirmed that these measures are not one hundred percent successful and a security assessment should be conducted regularly to enforce and test the security measures imposed. In [27], three defense mechanisms were provided to prevent PTH attacks, the most important of which is dividing the authorities and not allowing every admin access to all resources, which limits the attacker's ability to compromise all the network after compromising a single admin account.

4.2. Assessing Active Directory Security and Vulnerabilities

In [51], a study of security issues in the AD environment was presented. The authors concluded that the Active Directory is an important piece in any environment allowing greater control over business resources. They listed common vulnerabilities in the AD environment and stressed that these vulnerabilities are critical and should be hardened to protect the business assets and avoid any compromise. In [3], a penetration testing methodology for the Active Directory was presented. The scheme called MSDEPTM has ten stages including all the actions that could be performed by an attacker. The authors compared their proposed method with other penetration testing standards and showed the advantages. They also provided the defenses which can be deployed to protect from various AD attacks.

4.3. Active Directory Attack Detection

The authors in [52] proposed a technique to detect attacks involving domain administrator accounts using Windows event logs. They proved that their scheme is efficient in detecting such attacks. However, a drawback of their scheme is that it detects attacks based on an explicit list of commands which are run by attackers. This method is limited, and attackers can use a different command to carry out their malicious behaviors. Another method was proposed in [3] to detect APT attacks using machine learning techniques. The authors used unsupervised machine learning, as the data set for the attacks is huge and it is difficult to differentiate benign from malicious behavior. They also used event logs as a data source and emphasized that detecting attacks using process creation is efficient in detecting APT attacks in AD environments. The limitation of their approach is that their technique can produce false positives if the real administrator uses the same commands as attackers.

4.4. Defending against Account Lockout Attacks

The authors in [53] studied and surveyed account lockout attacks. They emphasized that most top companies have an exposed web portal that can be targeted in account lockout attacks. They compared the different approaches deployed to defend against this attack and found that the user-supplied tokens and middle box tokens are the most efficient at preventing this attack. The middle box in this research is referring to firewalls or proxies intercepting the traffic.

5. Experimental Work and Analysis

In this section, we launched two privilege escalation attacks. The first was the pass the hash attack [50], and the second was the Kerberoasting attack [25]. The goal of these experiments was to assess and see any signatures of these attacks in the Windows event logs to assist in fast detection and response in case of any intrusion. The section is organized as follows: In Section 5.1, the lab setup and methodology are presented, and then each attack is presented in a separate section followed by its results in Sections 5.2 and 5.3.

5.1. Lab Setup and Methodology

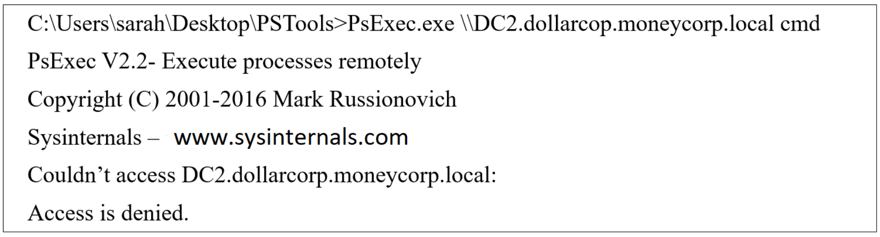
Our scope in this lab was not to assess the Windows defense against the two attacks; rather, we aimed to perform the attacks without any constraints to be able to assess the detection mechanism in a Windows event viewer. With this in mind, we performed these attacks with the Windows defender disabled and without applying any necessary patching to the operating system. We utilized a Dell T5500 workstation (Dell Technologies, Austin, TX, USA) as underlying hardware in our lab. The software tools and components used in the lab are listed in Table 4.

Table 4. Software Tools of the Experimental Work.

Tool	Origin	Description
VMware Workstation 14	VMware, Inc., Palo Alto, CA, USA	A hypervisor used to host the test machines.
WindowsServer 2016	Microsoft Corp., One Microsoft Wasy, Redmond, Washington, DC, USA	The server providing the domain controller rule.
Windows10 Enterprise	Microsoft Corp., One Microsoft Wasy, Redmond, Washington, DC, USA	Client machine.
WindowsPowerShell	Microsoft Corp., One Microsoft Wasy, Redmond, Washington, DC, USA	Shell used to launch different tools.
Mimikatz 2.0	BenjaminDelpy, Security Researcher, France	Tool used to extract the tickets and launch different attacks.
WindowsEvent Viewer	Microsoft Corp., One Microsoft Wasy, Redmond, Washington, DC, USA	Windows default tool to log events.
PSEXEC	Microsoft Corp., One Microsoft Wasy, Redmond, Washington, DC, USA)	A tool for running command over the network in Windows.
FileZilla Server	Open Source FTP server	FTP server used in Kerberoasting attack.
Hashcat	Open Source	Tool used for cracking passwords.
Powerview	Matt Graeber	Active Directory enumeration tool.

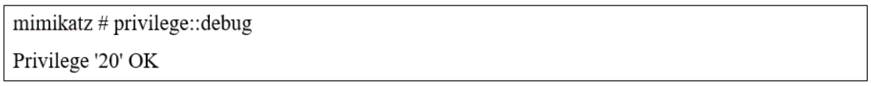
5.2. Pass the Hash Attack

In our experiment, we assumed that an attacker had compromised the Windows 10 client machine named Student-Machine and compromised the user account named Sarah, which had local administrative privileges on Student-Machine. We also had the Hassan admin account, which is a domain admin logged onto the compromised machine. The attack was carried out as follows. The attacker elevated his privileges using Privilege::debug command to be able to access the LSA process and dumped its hashes. He issued sekurlsa::logonpasswords to dump the NTLM hashes in the LSASS memory. He then found the NTLM hash of hassan_admin account, which was used to carry out the pass the hash attack and escalate his privileges. The command used to pass the hash and obtain a TGT for the domain admin is `sekurlsa::pth/user: hassan_admin/domain:dollarcorp.moneycorp.local /ntlm:e19ccf75ee54e06b06a5907af13cef42`. We assured the escalation of privileges by running the psexec command on the domain controller to open a CMD, and the command ran successfully, which indicated that we then had domain admin privileges. Figures 8–12 show the output of the commands issued on the compromised machine. Figure 8 shows that initially trying to run psexec on the domain controller gave permission denied, which was expected, as the user still had local admin on the client machine only. Figure 9 shows that the privileges had been elevated in Mimikatz and the tool was able to dump the NTLM hash from memory. Figure 10 shows the hassan_admin user NTLM hash in the LSASS memory. Figure 11 shows the output of the pass the hash command, which shows that a ticket with admin privilege had been issued successfully. Finally, Figure 12 shows that `PSEXEC` command had been run successfully on the domain controller.



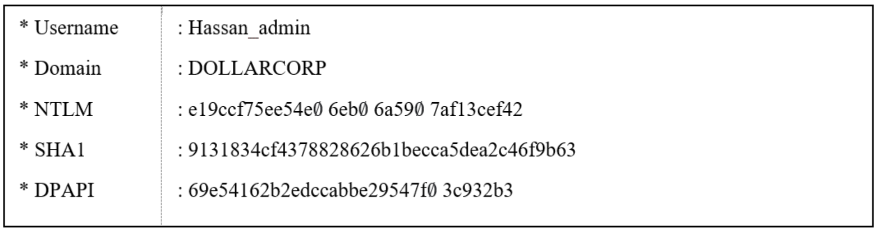
```
C:\Users\sarah\Desktop\PSTools>PsExec.exe \\DC2.dollarcop.moneycorp.local cmd
PsExec V2.2- Execute processes remotely
Copyright (C) 2001-2016 Mark Russionovich
Sysinternals – www.sysinternals.com
Couldn’t access DC2.dollarcorp.moneycorp.local:
Access is denied.
```

Figure 8. Access Denied on Domain Controller.



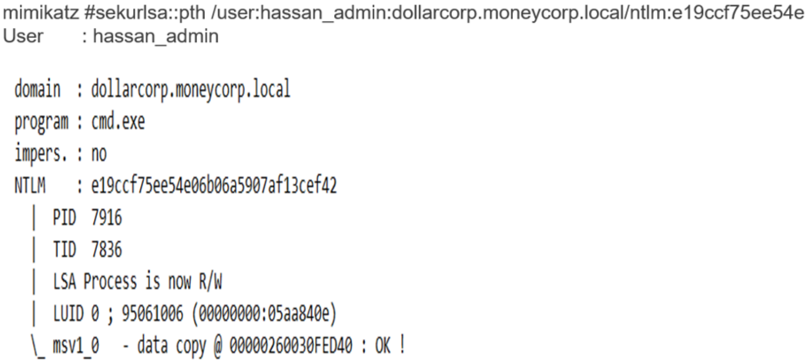
```
mimikatz # privilege::debug
Privilege '20' OK
```

Figure 9. Elevated Privileges on Mimikatz.



```
* Username    : Hassan_admin
* Domain      : DOLLARCORP
* NTLM        : e19ccf75ee54e0 6eb0 6a590 7af13cef42
* SHA1        : 9131834cf4378828626b1becca5dea2c46f9b63
* DPAPI       : 69e54162b2edccabbe29547f0 3c932b3
```

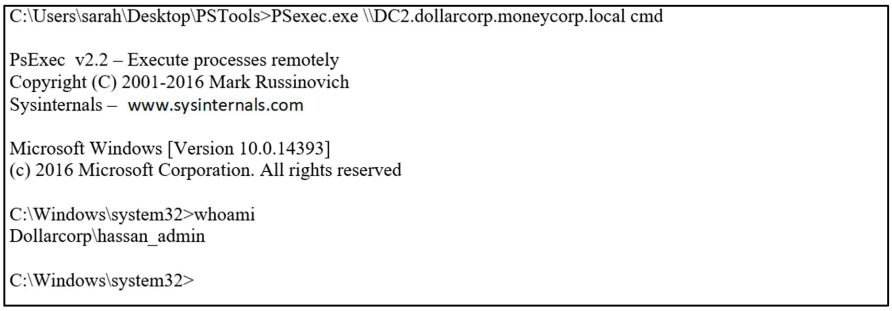
Figure 10. Admin NTLM Hash Dumped Successfully.



```
mimikatz #sekurlsa::pth /user:hassan_admin:dollarcorp.moneycorp.local/ntlm:e19ccf75ee54e
User      : hassan_admin

 domain  : dollarcorp.moneycorp.local
 program : cmd.exe
 impers. : no
 NTLM    : e19ccf75ee54e06b06a5907af13cef42
   |  PID  7916
   |  TID  7836
   |  LSA Process is now R/W
   |  LUID 0 ; 95061006 (00000000:05aa840e)
   \_ msv1_0   - data copy @ 00000260030FED40 : OK !
```

Figure 11. Admin Ticket Issued Successfully.



```
C:\Users\sarah\Desktop\PSTools>PSexec.exe \\DC2.dollarcorp.moneycorp.local cmd

PsExec  v2.2 – Execute processes remotely
Copyright (C) 2001-2016 Mark Russionovich
Sysinternals – www.sysinternals.com

Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved

C:\Windows\system32>whoami
Dollarcorp\hassan_admin

C:\Windows\system32>
```

Figure 12. Domain Controller Compromise.

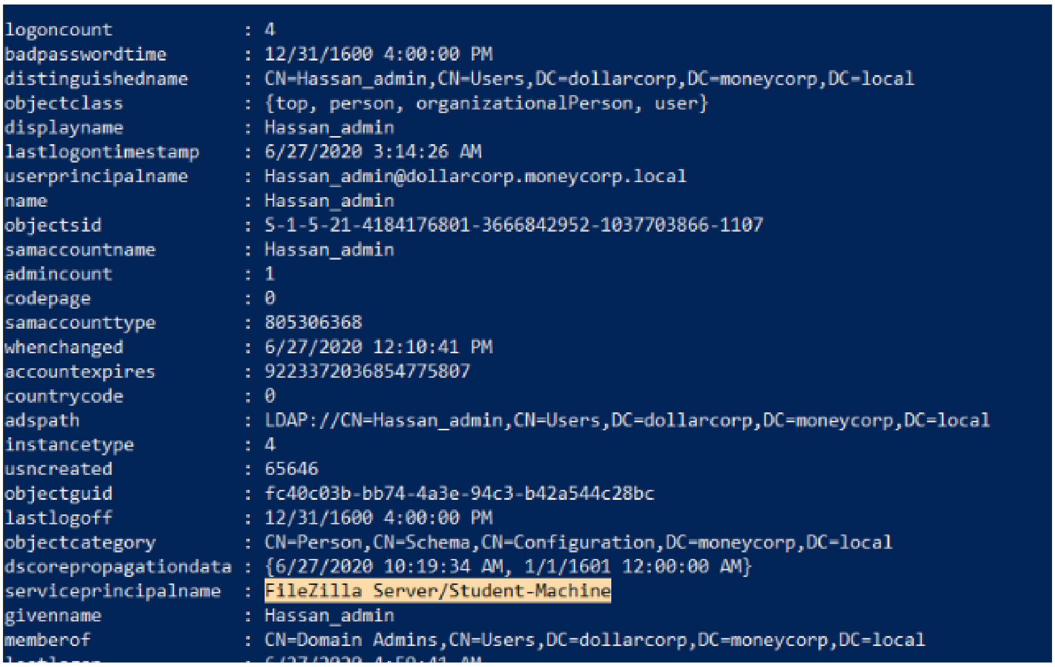
Investigating the logs of the compromised host, the target host, and the domain controller gives insights about the events logged. Upon examining the logs, it can be seen that the logs are the same for normal authentication and ticket creation on the source and the domain controller. However, there are a couple of logs which differ and can be spotted as signs of the pass the hash attack. The first is log event 4624, which differs in the logon type and has the value 9. The second is log event 4672, which indicates special privileges assigned to the initial user who launched the attack. The events on the source, target host, and the domain controller are presented in Table 5.

Table 5. Events logged for pass the hash attack.

Source Host	Target Host	Domain Controller
4648—A logon was attempted using explicit credentials.	4624—An account was successfully logged on (logon type 3, NTLM).	4776—The computer attempted to validate the credentials for an account.
4624—An account was successfully logged on **(logon type = 9; logon process = Seclogo)**.	4672—Special privileges were assigned to new logon.	
4672—Special privileges were assigned to new logon **(logged-on user, not impersonated user)**.		

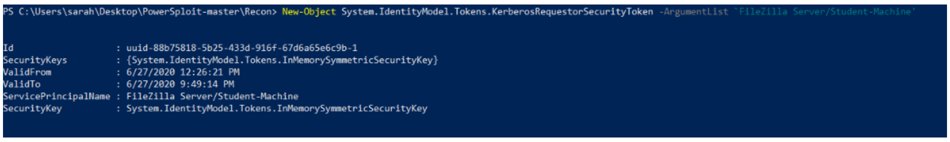
5.3. Kerberoasting Attack

To simulate this attack, an FTP server was set up on the client machine, and its service was started using hassan_admin account, which was a domain admin account. The SPN value of the account was set to the name of the service, which was File Zilla server/ Student-Machine, so that the admin account was targeted for Kerberoasting. The attack was launched as follows: The Get-NetUser -SPN command from Powerview was used to scan the domain for accounts which had the SPN attribute. The output showed the target admin account along with its SPN value which was the Fille Zilla server. PowerShell was then used to request a service ticket for the File Zilla server. The ticket was encrypted with the password of the service account, which was the domain admin in this case. The command Kerberos::list/export was issued to export the current tickets in the memory to the disk. Afterwards, an offline crack for the service ticket was issued using hashcat to obtain the admin password successfully. Figures 13–15 illustrate that the outputs of the commands ran on the compromised machine. Figure 13 shows the SPN value for the target admin account. Figure 14 shows that the service ticket was issued successfully for the File Zilla server. Finally, Figure 15 shows the service ticket exported using Mimikatz. As the Kerberoasting attack belongs to password-cracking attacks, no useful logs were found on the workstation or domain controller that could indicate a privilege escalation attack. This is because after cracking the service account the attacker gains the password and can log in normally without any weird behavior, which makes it harder to detect these types of attacks. However, it is recommended to use complex and long passwords for service accounts and limit the privilege of these accounts to reduce the attack surface for this attack.



```
logoncount            : 4
badpasswordtime       : 12/31/1600 4:00:00 PM
distinguishedname     : CN=Hassan_admin,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local
objectclass           : {top, person, organizationalPerson, user}
displayname           : Hassan_admin
lastlogontimestamp    : 6/27/2020 3:14:26 AM
userprincipalname     : Hassan_admin@dollarcorp.moneycorp.local
name                  : Hassan_admin
objectsid             : S-1-5-21-4184176801-3666842952-1037703866-1107
samaccountname        : Hassan_admin
admincount            : 1
codepage              : 0
samaccounttype        : 805306368
whenchanged           : 6/27/2020 12:10:41 PM
accountexpires        : 9223372036854775807
countrycode           : 0
adspath               : LDAP://CN=Hassan_admin,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local
instancetype          : 4
usncreated            : 65646
objectguid            : fc40c03b-bb74-4a3e-94c3-b42a544c28bc
lastlogoff            : 12/31/1600 4:00:00 PM
objectcategory        : CN=Person,CN=Schema,CN=Configuration,DC=moneycorp,DC=local
dscorepropagationdata : {6/27/2020 10:19:34 AM, 1/1/1601 12:00:00 AM}
serviceprincipalname  : FileZilla Server/Student-Machine
givenname             : Hassan_admin
memberof              : CN=Domain Admins,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local
```

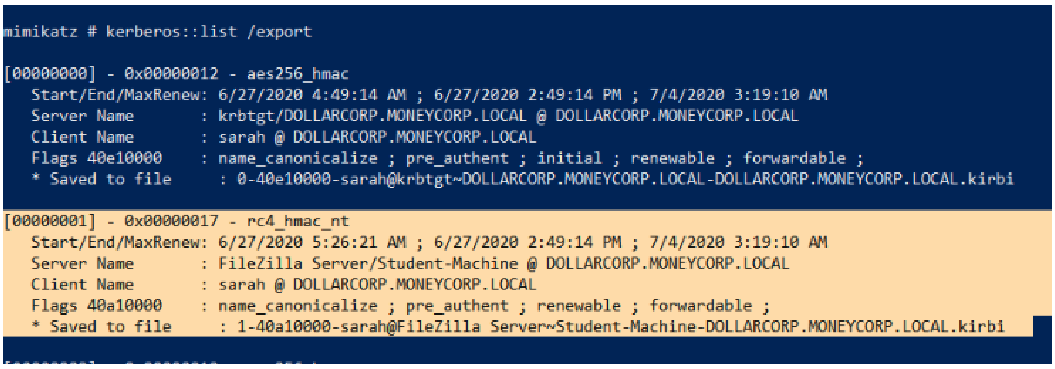
Figure 13. Admin Account has SPN Set.



```
PS C:\Users\sarah\Desktop\PowerSploit-master\Recon> New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList `FileZilla Server/Student-Machine'


Id                   : uuid-88b75818-5b25-433d-916f-67d6a65e6c9b-1
SecurityKeys         : {System.IdentityModel.Tokens.InMemorySymmetricSecurityKey}
ValidFrom            : 6/27/2020 12:26:21 PM
ValidTo              : 6/27/2020 9:49:14 PM
ServicePrincipalName : FileZilla Server/Student-Machine
SecurityKey          : System.IdentityModel.Tokens.InMemorySymmetricSecurityKey
```

Figure 14. Service ticket issued successfully.



```
mimikatz # kerberos::list /export

[00000000] - 0x00000012 - aes256_hmac
   Start/End/MaxRenew: 6/27/2020 4:49:14 AM ; 6/27/2020 2:49:14 PM ; 7/4/2020 3:19:10 AM
   Server Name       : krbtgt/DOLLARCORP.MONEYCORP.LOCAL @ DOLLARCORP.MONEYCORP.LOCAL
   Client Name       : sarah @ DOLLARCORP.MONEYCORP.LOCAL
   Flags 40e10000    : name_canonicalize ; pre_authent ; initial ; renewable ; forwardable ;
   * Saved to file     : 0-40e10000-sarah@krbtgt~DOLLARCORP.MONEYCORP.LOCAL-DOLLARCORP.MONEYCORP.LOCAL.kirbi

[00000001] - 0x00000017 - rc4_hmac_nt
   Start/End/MaxRenew: 6/27/2020 5:26:21 AM ; 6/27/2020 2:49:14 PM ; 7/4/2020 3:19:10 AM
   Server Name       : FileZilla Server/Student-Machine @ DOLLARCORP.MONEYCORP.LOCAL
   Client Name       : sarah @ DOLLARCORP.MONEYCORP.LOCAL
   Flags 40a10000    : name_canonicalize ; pre_authent ; renewable ; forwardable ;
   * Saved to file     : 1-40a10000-sarah@FileZilla Server~Student-Machine-DOLLARCORP.MONEYCORP.LOCAL.kirbi
```

Figure 15. Service Ticket Exported to Disk.

6. Conclusions and Future Work

Active Directory attacks have become more risky and sophisticated and they can compromise the entire environment. In this paper, we presented the typical AD attack lifecycle in detail. We summarized the main characterizing attributes of the Advanced Persistent Threats by exploring their meaning, signs, lifecycle, techniques, types of targets, comparison with malware, as well as protection and detection mechanisms. We also illustrated the Kerberos authentication workflow, which is abused in most AD attacks. The most common AD attacks were discussed. Additionally, an overview of the existing detection and mitigation mechanisms was provided. Further, two privilege escalation attacks were analyzed in our experimental work. The first is the pass the hash attack, and the second is the Kerberoasting attack. The goal of these experiments was to assess and investigate any signatures of these attacks in the Windows event logs to assist in fast detection and response in case of any intrusion. For future work, we plan to propose and simulate new detection and mitigation mechanisms for different types of Active Directory attacks.

Author Contributions: Conceptualization, B.I.M. and M.A.A.; methodology, B.I.M. and M.A.A.; software, B.I.M.; validation, B.I.M. and M.A.A.; formal analysis, B.I.M. and M.A.A.; investigation, B.I.M., M.S.E. and M.A.A.; resources, B.I.M.; data curation, B.I.M. and M.A.A.; writing—original draft preparation, B.I.M., A.D.J. and M.A.A.; writing—review and editing, B.I.M., A.D.J., M.S.E. and M.A.A.; visualization, B.I.M., A.D.J., M.S.E. and M.A.A.; supervision, A.D.J., M.S.E. and M.A.A.; project administration, B.I.M., A.D.J., M.S.E. and M.A.A.; funding acquisition, A.D.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the University College Dublin (UCD), School of Computer Science, Dublin, Ireland, grant number 13/RC/2077.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: This article does not contain any studies with human participants or animals performed by any of the authors.

Data Availability Statement: Data in this research paper will be shared upon request made to the corresponding author.

Conflicts of Interest: All authors declare that they have no conflict of interest for the presented work.

References

1. Kotlaba, L.; Buchovecká, S.; Lórencz, R. Active Directory Kerberoasting Attack: Detection using Machine Learning Techniques. In Proceedings of the 7th International Conference on Information Systems Security and Privacy (ICISSP 2021), Online, 11–13 February 2021; pp. 376–383.
2. Gkotsis, P. Creating a Windows Active Directory Lab and Performing Simulated Attacks. Master's Thesis, University of Piraeus, Piraeus, Greece, 2021.
3. Pektaş, A.; Başaranoğlu, E. Practical Approach For Securing Windows Environment: Attack Vectors And Countermeasures. In Proceedings of the 7th International Conference on Information Systems Security and Privacy (ICISSP 2021), Online, 11–13 February 2021; pp. 376–383.
4. Matsuda, W.; Fujimoto, M.; Mitsunaga, T. Potential use of prostate specific membrane antigen (PSMA) for detecting the tumor neovasculature of brain tumors by PET imaging with 89Zr-Df-IAB2M anti-PSMA minibody. In Proceedings of the 2018 IEEE Conference on Application, Information and Network Security (AINS), Langkawi, Malaysia, 21–22 November 2018; pp. 60–65.
5. Jeun, I.; Lee, Y.; Won, D. A practical study on advanced persistent threats. In *Computer Applications for Security, Control and System Engineering*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 144–152.
6. Advanced Persistent Threat (APT) Attacks. Available online: https://www.cynet.com/advanced-persistent-threat-apt-attacks/ (accessed on 30 July 2022).
7. Fireeye Advanced Threat Report 2013: FireEye Labs. 2013. Available online: https://www2.fireeye.com/rs/fireye/images/fireeye-advanced-threat-report-2013.pdf (accessed on 30 July 2022).
8. Quintero-Bonilla, S.; Martín del Rey, A. A new proposal on the advanced persistent threat: A survey. *Appl. Sci.* **2020**, *10*, 3874. [CrossRef]
9. Kaspersky. Kaspersky's 2019 IT Security Economics Report. Available online: https://go.kaspersky.com/rs/802-IJN-240/images/GL_Kaspersky_Report-IT-Security-Economics_report_2019.pdf (accessed on 9 September 2021).
10. Steiner, J.G.; Neuman, B.C.; Schiller, J.I. Kerberos: An Authentication Service for Open Network Systems. In Proceedings of the Usenix Winter, Dallas, Texas, USA, 9–12 February 1988; pp. 191–202.
11. Alva, D.; Benjamin, D. Abusing Microsoft Kerberos. Available online: https://www.blackhat.com/docs/us-14/materials/us-14-Duckwall-Abusing-Microsoft-Kerberos-Sorry-You-Guys-Don't-Get-It.pdf (accessed on 9 September 2021).
12. Github. BloodHoundAD. Available online: https://github.com/BloodHoundAD/BloodHound (accessed on 13 September 2021).
13. Will Schroeder. PowerSploit. Available online: https://github.com/PowerShellMafia/PowerSploit/tree/master/Recon (accessed on 13 September 2021).
14. Cybersecurity Bits, Bobs. Active Directory Domain Enumeration. Available online: https://mlcsec.com/active-directory-domain-enumeration/# (accessed on 13 September 2021).
15. Motero, C.D.; Higuera, J.R.B.; Higuera, J.B.; Montalvo, J.A.S.; Gómez, N.G. On Attacking Kerberos Authentication Protocol in Windows Active Directory Services: A Practical Survey. *IEEE Access* **2021**, *9*, 109289. [CrossRef]
16. Diogenes, Y.; Ozkaya, E. *Cybersecurity—Attack and Defense Strategies: Infrastructure Security with Red Team and Blue Team Tactics*; Packt Publishing Ltd.: Birmingham, UK, 2018.
17. White, S. Net.exe. Available online: https://docs.microsoft.com/en-us/windows/win32/winsock/net-exe-2 (accessed on 13 September 2021).
18. Ebad, S.A. Lessons learned from offline assessment of security-critical systems: the case of microsoft's active directory. *Int. J. Syst. Assur. Eng. Manag.* **2022**, *13*, 535. [CrossRef]
19. Microsoft. Active Directory. Available online: https://docs.microsoft.com/en-us/powershell/module/activedirectory/?view=windowsserver2019-ps (accessed on 13 September 2021).
20. Melnick, J. How to Create New Active Directory Users with Powershell, SysAdmin Magazine, June 2019. Available online: https://blog.netwrix.com/2018/06/07/how-to-create-new-active-directory-users-with-powershell/ (accessed on 13 January 2022).
21. Fletcher, D.R., Jr. Cyber Threat Intelligence Uses, Successes and Failures: The SANS 2017 CTI Survey. Available online: www.sans.org/reading-room/whitepapers/analyst/cyber-threat-intelligence-uses-successes-failures-2017-cti-survey-37677 (accessed on 14 March 2017).
22. TNelson; Kettani, H. Open source powershell-written post exploitation frameworks used by cyber espionage groups. In Proceedings of the 2020 3rd International Conference on Information and Computer Technologies (ICICT), San Jose, CA, USA, 9–12 March 2020; pp. 451–456.
23. Lemmens, M. BloodHound—Sniffing Out the Path Through Windows Domains. Available online: https://www.sans.org/blog/bloodhound-sniffing-out-path-through-windows-domains/ (accessed on 13 September 2021).
24. Myllyla, J.; Costin, A. Reducing the Time to Detect Cyber Attacks: Combining Attack Simulation with Detection Logic. In Proceedings of the Conference of Open Innovations Association FRUCT (FRUCT Oy, 2021), Oulu, Finland, 27–29 October 2021.
25. Rights, R.F. *Use Offense to Inform Defense. Find Flaws before the Bad Guys Do*; SANS Institute: Rockville, MD, USA, 2015.
26. El-Hadidi, M.G.; Azer, M.A. Traffic Analysis for Real Time Applications and its Effect on QoS in MANETs. In Proceedings of the 2020 15th International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt, 15–16 December 2020; pp. 1–6.

27. Dimov, D.; Tzonev, Y. Pass-the-hash: One of the most prevalent yet underrated attacks for credentials theft and reuse. In Proceedings of the 18th International Conference on Computer Systems and Technologies (2017), Ruse, Bulgaria, 23–24 June 2017; pp. 149–154.
28. Roobol, S.; Offerman, N.; de Laat, C.; van de Wouw, D.; Huijgen, A. *Development of Techniques to Remove Kerberos Credentials from Windows Systems, M.Sc*; Security and Network Engineering, School of Computer Science, University of Amsterdam: Amsterdam, The Netherlands, 2019.
29. Badhwar, R. Advanced Active Directory Attacks and Prevention. In *The CISO's Next Frontier*; Springer: Midlothian, VA, USA, 2021; pp. 131–144.
30. Ah-Fat, P.; Huth, M.; Mead, R.; Burrell, T.; Neil, J. Effective detection of credential thefts from windows memory: Learning access behaviours to local security authority subsystem service. In Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020), San Sebastian, Spain, 14–15 October 2020; pp. 181–194.
31. Higgs, C. Authorisation and Delegation in the Machination Configuration System. *LISA* **2008**, *8*, 191–199.
32. Warren, J. Unconstrained Delegation Permissions. Available online: https://stealthbits.com/blog/unconstrained-delegation-permissions/ (accessed on 10 September 2021).
33. De Clercq, J.; Grillenmeier, G. *Microsoft Windows Security Fundamentals: For Windows 2003 SP1 and R2*; *Elsevier*: Amsterdam, The Netherlands, 2011; ISBN 9780080491882.
34. Amador, M.; Bagwell, K.; Frankel, A. A note on interval delegation. *Econ. Theory Bull.* **2018**, *6*, 239. [CrossRef]
35. Suman, B.; Justin, H. Configuring Kerberos Delegation for Group Managed Service Accounts. Available online: https://docs.microsoft.com/en-us/windows-server/security/group-managed-service-accounts/configure-kerberos-delegation-group-managed-service-accounts (accessed on 10 September 2021).
36. Kevin, J. Constrained Delegation Abuse: Abusing Constrained Delegation to Achieve Elevated Access. Available online: https://blog.stealthbits.com/constrained-delegation-abuse-abusing-constrained-delegation-to-achieve-elevated-access/ (accessed on 10 September 2021).
37. Markoff, J. Attack of the zombie computers is growing threat. *New York Times* **2007**, *157*, 1.
38. Soria-Machado, M.; Abolins, D.; Boldea, C.; Socha, K. Kerberos golden ticket protection. *Mitigating Pass-the-Ticket Act. Dir. CERT-EU Secur. Whitepaper* **2014**, *7*, 2016.
39. Metcalf, S. Red vs. Blue: Modern Active Directory Attacks, Detection, & Protection. Available online: https://www.blackhat.com/docs/us-15/materials/us-15-Metcalf-Red-Vs-Blue-Modern-Active-Directory-Attacks-Detection-And-Protection.pdf (accessed on 10 September 2021).
40. Liu, J.; Akhtar, N.; Mian, A. Adversarial training for commonsense inference. *IEEE Trans. Neural Netw. Learn. Syst. Rev.* **2020**, *47*, 777–780.
41. Tramèr, F.; Papernot, N.; Goodfellow, I.; Boneh, D.; McDaniel, P. The space of transferable adversarial examples. *arXiv* **2017**, arXiv:1704.03453.
42. Barker, S. *White Paper* ©; Copyright Quest® Software, Inc.: Aliso Viejo, CA, USA, 2007.
43. Boger, T. Directory Services Restore Mode (DSRM), & Protection. Available online: https://searchwindowsserver.techtarget.com/definition/Directory-Services-Restore-Mode-DSRM (accessed on 13 September 2021).
44. Warren, J. Stealing Credentials with a Security Support Provider (SSP). Available online: https://stealthbits.com/blog/stealing-credentials-with-a-security-support-provider-ssp/ (accessed on 13 September 2021).
45. Jacobs, M.; Satran, M. How Access Control Works in Active Directory Domain Services. Available online: https://docs.microsoft.com/en-us/windows/win32/ad/how-access-control-works-in-active-directory-domain-services (accessed on 13 September 2021).
46. Metcalf, S. Sneaky Active Directory Persistence #15: Leverage AdminSDHolder & SDProp to (Re)Gain Domain Admin Rights. Available online: https://adsecurity.org/?p=1906 (accessed on 13 September 2021).
47. Mittal, N.; RACE—Minimal Rights and ACE for Active Directory Dominance. Available online: http://www.labofapenetrationtester.com/2019/08/race.html (accessed on 13 September 2021).
48. Wheeler, S.; Wilson, C. Running Remote Commands. Available online: https://docs.microsoft.com/en-us/powershell/scripting/learn/remoting/running-remote-commands?view=powershell-7 (accessed on 13 September 2021).
49. Nichols, J.A.; Taylor, B.A.; Curtis, L. Security resilience: Exploring windows domain-level defenses against post-exploitation authentication attacks. In Proceedings of the 11th Annual Cyber and Information Security Research Conference (2016), Oak Ridge, TN, USA, 5–7 April 2016; pp. 1–4.
50. Jadeja, N.; Vaghasia, M. Analysis and Impact of Different Mechanisms of Defending Pass-the-Hash Attacks. In *Cyber Security*; Springer: Singapore, 2018; pp. 179–191.
51. Binduf, A.; Alamoudi, H.O.; Balahmar, H.; Alshamrani, S.; Al-Omar, H.; Nagy, N. Active directory and related aspects of security. In Proceedings of the 2018 21st Saudi Computer Society National Computer Conference (NCC) (IEEE, 2018), Riyadh, Saudi Arabia, 25–26 April 2018; pp. 4474–4479.

52. Fujimoto, M.; Matsuda, W.; Mitsunaga, T. Detecting apt attacks against active directory using machine leaning. In Proceedings of the 2018 IEEE Conference on Application, Information and Network Security (AINS), Langkawi, Malaysia, 21–22 November 2018; pp. 15–20.
53. Liu, Y.; Squires, M.R.; Taylor, C.R.; Walls, R.J.; Shue, C.A. Account Lockouts: Characterizing and Preventing Account Denial-of-Service Attacks. In Proceedings of the International Conference on Security and Privacy in Communication Systems, Orlando, FL, USA, 23–25 October 2019; pp. 26–46.