Crimson Publishers
Wings to the Research

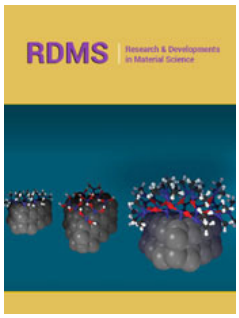
Review Article

Security Test of Active Directory Domain Services

Štefan Počarovský, Martin Koppl and Miloš Orgoň*

Institute of Electrical Engineering, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovicova 3, 812 19 Bratislava, Slovakia

ISSN: 2576-8840



***Corresponding author:** Miloš Orgoň, Institute of Electrical Engineering, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovicova 3, 812 19 Bratislava, Slovakia

Submission: February 15, 2023
Published: March 09, 2023

Volume 18 - Issue 3

How to cite this article: Štefan Počarovský, Martin Koppl and Miloš Orgoň*. Security Test of Active Directory Domain Services. Res Dev Material Sci. 18(3). RDMS. 000939. 2023.
DOI: 10.31031/RDMS.2023.18.000939

Copyright@ Miloš Orgoň*. This article is distributed under the terms of the Creative Commons Attribution 4.0 International License, which permits unrestricted use and redistribution provided that the original author and source are credited.

Abstract

In today's cyber-world, one of the most widely used security terms is the security of user accounts and their proper authentication process. We are in a digital age where every user of information systems has a digital identity [1]. If a particular user wants to access a particular service, he has to authenticate himself first and only on the basis of that the user will be granted rights to the service. A 2014 study says that the average user of web services had approximately 25 web accounts [2]. However, it is now estimated that there are approximately 80 web accounts per user, and some form of service identity authentication must be implemented for each of them [3]. In medium and large enterprise computer networks, services are used to centrally manage users, for example, using a Windows server role - "Active Directory Domain Services". This role uses the "Kerberos" authentication protocol.

Keywords: Active directory domain services; Ticket granting ticket; Golden ticket; Kerberos; "krbtgt" account

Introduction

If we want to manage a large group of users and their access in large computer networks, Active Directory Domain Services [4] is used to do this. These are companies that use an infrastructure built on Microsoft infrastructure. Active Directory Domain Services is an implementation of Lightweight Directory Access Protocol (LDAP) directory services. This service allows policies to be centrally managed, applying critical updates across a company's computing structure, where settings are stored in a centrally organized database. Kerberos is the name of the authentication protocol used by this service. It is the one that allows users to be authenticated on insecure computer networks using symmetric cryptography, but it needs a third party to authenticate. Among other things, it provides SSO (Single Sign-On), i.e., the user does not have to continuously enter his login credentials when requesting a network service. However, there are a lot of hacker attacks here and in this article, we will discuss the "GOLDEN TICKET ATTACK".

Manage Accounts in Computer Networks

In computer networks that are built on a "WINDOWS" infrastructure, user management can be performed individually on each device or centrally. With non-centralised management, this is manageable up to 10 active devices on the network. If the network contains more active devices or users, it is necessary to switch to centralised user management for security reasons. It is with this type of user management that we get a much better overview of active accesses, active and inactive user accounts. You only need to block or allow a user in one place and there is no need to block them on every device.

Computer grouping management type-workgroup

A workgroup is a logical grouping of computers that are enrolled under the same workgroup name. All devices in a workgroup must be of type PEER-TO-PEER and each device in the network must maintain its own local SAM (Security Accounts Manager) database [5]. This means that the same users must be created on each device, but this does not allow the use of this system in large enterprise networks, as the management of such an infrastructure would be extremely difficult and complicated. To ensure network functionality, the same usernames and passwords must be created on each PC, which limits security, among other things. If an

attacker want to get access to the database of login accounts and passwords, he would only need access to any computer on the network. At the same time, if we want to deny a user access to a network service, we would have to deny access on every computer where that user was created, which is both organizationally difficult and time consuming. A simple representation of a workgroup is shown in (Figure 1).

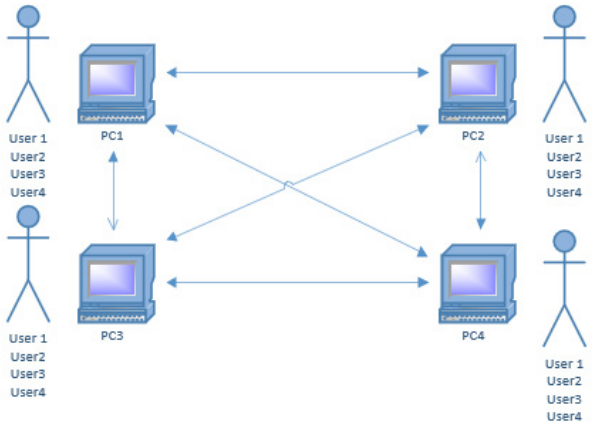


Figure 1: A simple representation of a workgroup in windows infrastructure.

Computer grouping management type-active directory domain services

The complexity of managing user accounts is addressed by the ADDS (Active Directory Domain Services) role [4]. It is a hierarchically structured system and stores all information about objects on the network in its database, allowing system administrators to easily find and use this information. However, the data store contains not only users and hashes of their passwords, but also a list of PCs, printers, servers, and various other volumes, etc. This data is stored in the "Ntds.dit" file on the domain controller, that is, on the server(s) where the ADDS service is implemented. Security is integrated with ACTIVE DIRECTORY by means of login authentication and object access control. Using this system, authenticated users can access authorized resources throughout the network.

Active Directory also includes:

a. Schema -defines the object classes, the constraints and limits of the object instances, and the format of the object names.

b. Global Catalog -contains information about each object in a domain, allowing users and administrators to find information regardless of which domain in the directory actually contains the data.

c. Query and index mechanism -objects can be found or published by network users or applications.

d. Replication service -distributes data in the domain controller over the network. Any data change is distributed to all domain controllers in the network.

To ensure that specific tasks are performed, to ensure consistency of structure, and to eliminate potential conflicts in the Ntds.dit database, ADDS uses 5 FSMO roles that domain controllers share with each other [6]. This means that the domain controllers do not replicate it across the network, but there is only one for the entire forest.

They are:

a) Schema Master,

b) Domain Naming Master,

c) Primary Domain Controller (PDC) Emulator,

d) Infrastructure Master,

e) Relative ID (RID) Master.

In the (Figure 2) is a simple representation of the ADDS role services and their user accounts.

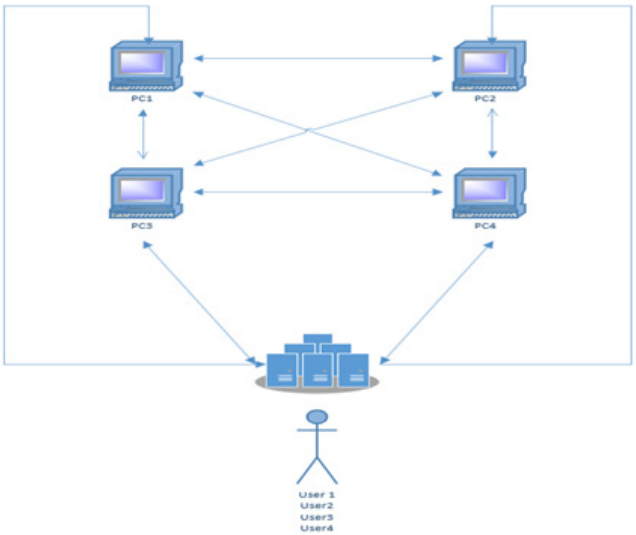


Figure 2: A simple representation of the active directory domain services in windows infrastructure.

In contrast to the WORKGROUP logical grouping of computers, the individual objects (users, among others) are created on the domain controller. When a network service is requested by an active device on the network, the active device itself first verifies the permissions of the requester on the domain controller and then grants or denies the service to the requester. There is no need to have databases of objects on each active element, as there was with the workgroup, but all this data is just in one database on the domain controller. Of course, in the case of multiple domain controllers on the network, that database is replicated to the other domain controllers. ADDS uses the KERBEROS authentication system for this authentication method.

Protocol kerberos: The Kerberos authentication network protocol [7] is used for Single Sign-On (SSO) in large organizations. It is a sign-on where a user U authenticates against his computer P and this computer will authenticate against the organization's servers with its user login. This protocol is based on the principles

Copyright © Miloš Orgoň

of symmetric cryptography, where the secret key K plays the role of both a proof and an authentication token, i.e., K=DF=OF, where:

D_F -proof ticket, where the PC will prove its identity and its rights in the system,

O_F -authentication ticket, is the data that the system will use to authenticate the computer's identity.

We will refer to the individual Kerberos protocol parties as the computer P and user U, the factor server as FS, the accreditation server as AS, and the destination server as CS. The process is as follows. After user U authenticates on computer P, user U gains access to the long-term proof ticket, i.e., the K_{PF} key. The computer then requests the short-term proof ticket K_1 from the factor server. With this key, the computer P can authenticate itself to the credential server AS and obtains from it a one-time ticket, which is the key K_2. With this K_2 key, computer P can authenticate itself against the target server CS whenever it needs a service (e.g. access to shared files). Thus, each PC is assigned a K_{PF} key depending on the authenticated user. The FS factor server has access to all K_{PF} PC_s (depending on the users in the network), it has access to the K_{FA} key, which is the key for the encrypted transfer of short-term tickets towards the AS accreditation server, and at the same time it has access to the K_{AC} key, which is the key of the K_{AC} target server. This is used to encrypt the transmission to the individual CS destination servers. A better explanation is provided by (Figure 3); [8].

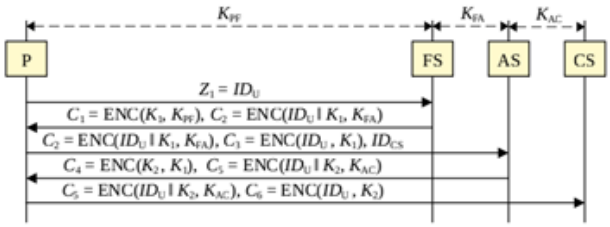


Figure 3: Principle of kerberos protocol, taken from Kryptografie okolo nás [8].

Each user U authenticates to his computer P with an access password. Using this password, the computer P can retrieve the K_{PF} proof ticket of user U. The K_{PF} is stored in encrypted form on disk and the decryption key is obtained by a derivation function from the given password. Computer P then sends a request $Z_1=ID_U$ to the FS factor server to allocate a short-term proof ticket, where the short-term proof ticket is actually the secret key K_1. The factor server verifies the existence of a user with ID_U, then uses the user's KPF key, generates a random key K1, and sends an allocation cryptogram $C_1=ENC(K_1,K_{PF})$ and a confirmation cryptogram $C_2=ENC(IDU \| K_1,K_{PF})$ to computer P. The cryptogram $C_1=ENC(ID_U \| K1,K_{PF})$ is used to generate the random key K_1. Computer P decrypts the cryptogram C_1 and obtains the proof ticket $K_1=DEC(C1,K_{PF})$. It stores the confirmation cryptogram C_2. The key KFA is accessed by computer P, the factor server FS and the accreditation server AS, i.e., we can use the cryptogram $C_2=ENC(ID_U \| K_1,K_{PF})$ to confirm that user U with ID_U has received the proof or verification ticket K_1 from the factor server FS.

If user U will use the service of the target server CS, it will use the following procedure. It is necessary for the computer P to first send to the accreditation server AS the cryptogram C_2, the proof cryptogram $C_3=ENC(ID_U,K_1)$ and at the same time the ID of the target server IDCS. According to this identifier, the accreditation server determines that the requestor is interested in the service provided by the CS server. The accreditation server then decrypts the confirmation cryptogram C_2 with the key K_{FA} , resulting in $ID_U \| K_1=DEC(C_2,K_{FA})$. For the accreditation server AS, this is a trusted confirmation from the FS server that a proof ticket has been assigned to the user U with the ID_U. Then the AS accreditation server verifies that this is indeed the user with the ID_U. It must be the case that by decrypting the proof cryptogram C_3, the AS server obtains the same ID as that given in the confirmation cryptogram C_2. Thus, if $DEC(C_3,K_1)=ID_U$, then the AS accreditation server will generate a one-time proof ticket for user U, which is key K_2. Then it sends the allocation cryptogram C_4 to the host $P=ENC(K_2,K_1)$ and the confirmation cryptogram $C_5=ENC(ID_U \| K_2,K_{AC})$.

The user decrypts the allocation cryptogram C_4, in this way he finds out the value of the ticket $K_2 =DEC(C_4,K_1)$ that has been allocated to him. He then forwards the confirmation cryptogram C_5 to the target C_S, also with the associated proof cryptogram $C_6=ENC(ID_U,K_2)$. The target C_S server first decrypts with the key K_{AC}, the cryptogram C5, and thus obtains $DEC(C5,K_{AC})=ID_U \| K_2$. This is an acknowledgement from the accreditation server that the user with ID_U has been assigned key K_2 as a proof ticket. This key is then used to decrypt the proof cryptogram C_6 and should be given the same identifier as was given in the decrypted cryptogram C_5. Thus, if $DEC(C_6,K_2)=ID_U$, then the network service requester is indeed requester U. This section concludes the role of the Kerberos protocol. The target C_S passes the data to the domain controller, then the domain controller discovers the rights of the user ID, and based on these rights, it grants or withholds the requested service to user U.

Golden Ticket Attack

If a user wants to access a service, such as accessing shared files, a printer, a database, etc., they must first prove their identity and permissions. Kerberos in ADDS acts as a third party and issues a TGT (Ticket Granting Ticket) [9,10] that vouches for the user's identity. The queried service then evaluates the given TGT. The Kerberos protocol provides many benefits that help provide security and convenience in authentication; without it, users would constantly have to enter passwords in clear text if they wanted to communicate with a network service. However, care must be taken at all the times to ensure that the protocol in question is not exploited to create a Golden TGT and gain administrative access privileges across the entire network that operates under ADDS [11].

What is TGT and golden ticket

MIMIKATZ [12] is a tool used by security researchers, it is publicly available and can be used for malicious purposes. If an attacker manages to hack an ADDS administrator account, MIMIKATZ can create a special KERBEROS TGT that has the following basic properties:

Copyright © Miloš Orgoň

a) Golden ticket is a method to generate a TGT of an arbitrary user in ADDS, then the attacker can impersonate anyone in the domain.

b) Golden ticket can also be created offline.

c) Golden ticket can be generated for any lifetime.

d) Event logs do not distinguish the use of a legitimate TGT ticket from a golden ticket, so there is no rule to ensure its use.

Description of test environment for a golden ticket attack

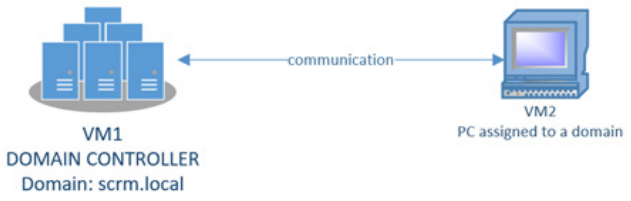


Figure 4: Proposed structure of attack simulation environment.

We created an environment to simulate a "Golden attack" under Hyper-V, where we generated a virtual machine (Gen2 type). On this virtual machine we installed "Windows server Datacenter edition 2016", where we simultaneously implemented the role ADDS which we set up, we generated a new forest named "scrm.local". We created a domain user with login "user.hack" of type "user". This user does not have access to the files on the domain controller's disk. We then created a second virtual machine on which we also installed "Windows Datacenter Edition 2016" and connected the server to the domain (not as a domain controller). While the PC to domain in Active Directory has 2 types of admin account (local admin, domain admin), the domain controller itself has only one type of admin account. The structure of this environment is shown in (Figure 4).

```
PS C:\Users\user.hack> cmd
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\user.hack>pushd \\DCTEST\C$
Access is denied.
```

Figure 5: Blocked access to data of domain controller.

Next, we will work exclusively under the "user.hack" account, which we have created as an account with user rights in the domain. After logging into VM2 under the "user.hack" account and using the SAMBA protocol, we wanted to log into VM1 (the domain controller) and download the NTDS.dit database where all the data and objects of the "scrm.local" domain are stored. Since we did not have permissions, the domain controller blocked our access, as can be seen in (Figure 5). We then listed the whoami /groups assigned to user "user.hack" using the whoami/groups command, as shown in (Figure 6). In this figure, we can clearly see that the user is not assigned to any domain admin group.

```
scrm\user.hack
PS C:\Users\user.hack> whoami /groups

GROUP INFORMATION
-----------------

Group Name                                 Type             SID          Attributes
========================================== ================ ============ ==================================================
Everyone                                   Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled group
BUILTIN\Administrators                     Alias            S-1-5-32-544 Group used for deny only
BUILTIN\Users                              Alias            S-1-5-32-545 Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\REMOTE INTERACTIVE LOGON      Well-known group S-1-5-14     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\INTERACTIVE                   Well-known group S-1-5-4      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization             Well-known group S-1-5-15     Mandatory group, Enabled by default, Enabled group
LOCAL                                      Well-known group S-1-2-0      Mandatory group, Enabled by default, Enabled group
Authentication authority asserted identity Well-known group S-1-18-1     Mandatory group, Enabled by default, Enabled group
Mandatory Label\Medium Mandatory Level     Label            S-1-16-8192
PS C:\Users\user.hack> _
```

Figure 6: Account user hack exists in these groups.

Creating of golden ticket

To create a golden ticket we need the following information:

a) NT hash of the domain "krbtgt" account,

b) the name of the targeted domain account (usually Domain admin or Enterprise admin),

c) the name of the associated domain,

d) the SID of the domain.

The MIMIKATZ utility, which was developed to prove that Microsoft's authentication protocols are vulnerable to attack, is used for the Golden Attack. It was developed by Benjamin Delpy and is still used today for security analysis of systems in the ADDS environment. The name of the domain under test and the SID is easy to obtain in the command line with the command „whoami / all" a „whoami /fqdn", so:

SID domény: S-1-5-21-2397579404-2857638458-883868324

Názov domény: scrm.local

The biggest problem was getting the hash of the "krbtgt" account. We used the aforementioned MIMIKATZ utility and with the command lsadump::dcsync /domain.scrm.local /user:krbtgt we found, among other things, the RC4 hash of the account „krbtgt", that value is: d0beb8ddb48fddd6a708ef14704e99d0. The output of the command is shown in the (Figure 7).

Copyright © Miloš Orgoň

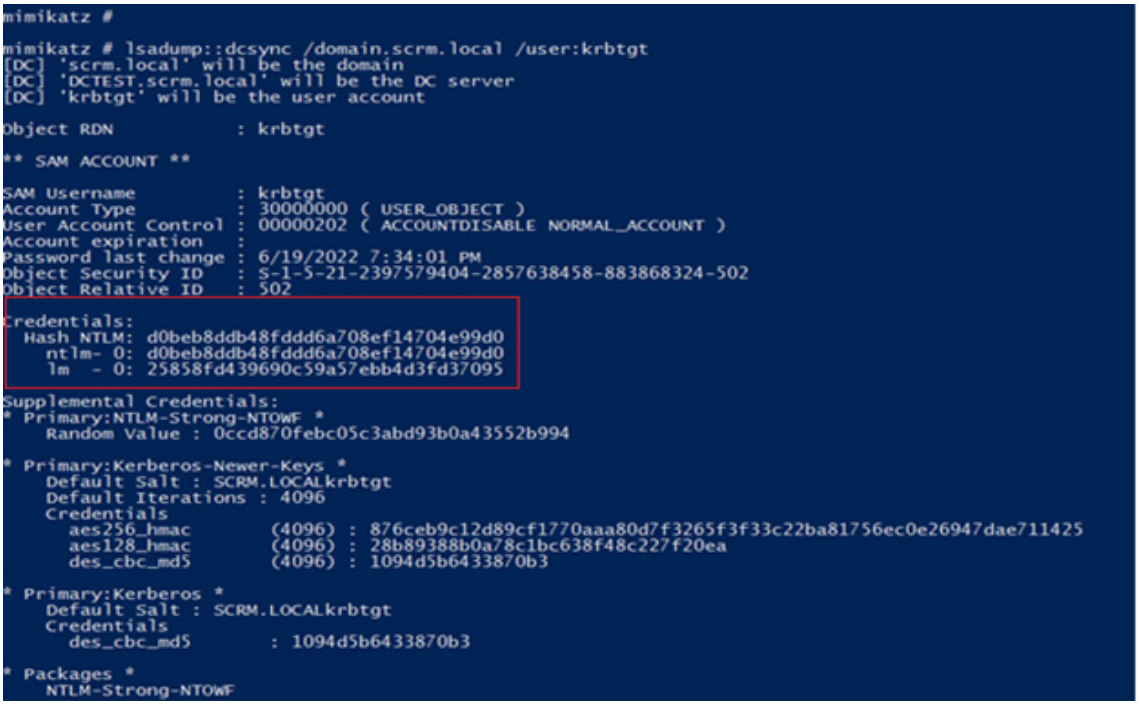


Figure 7: Command for finding the RC4 hash of „krbtgt“ account.

Once we had all the necessary data, we generated a TGT, which we signed with the hash “krbtgt” of the account, thus obtaining the so-called Golden Ticket. To generate it, we used the following command:

kerberos::golden /domain:scrm.local /sid :S-1-5-21-2397579404-2857638458-883868324 / rc4:d0beb8ddb48fddd6a708ef14704e99d0 /id:500 /user:user.hack

The output (confirmation of TGT generation) can be seen in (Figure 8). We then passed the given ticket to the current session using the command: kerberos::ptt ticket.kirbi (Figure 9).

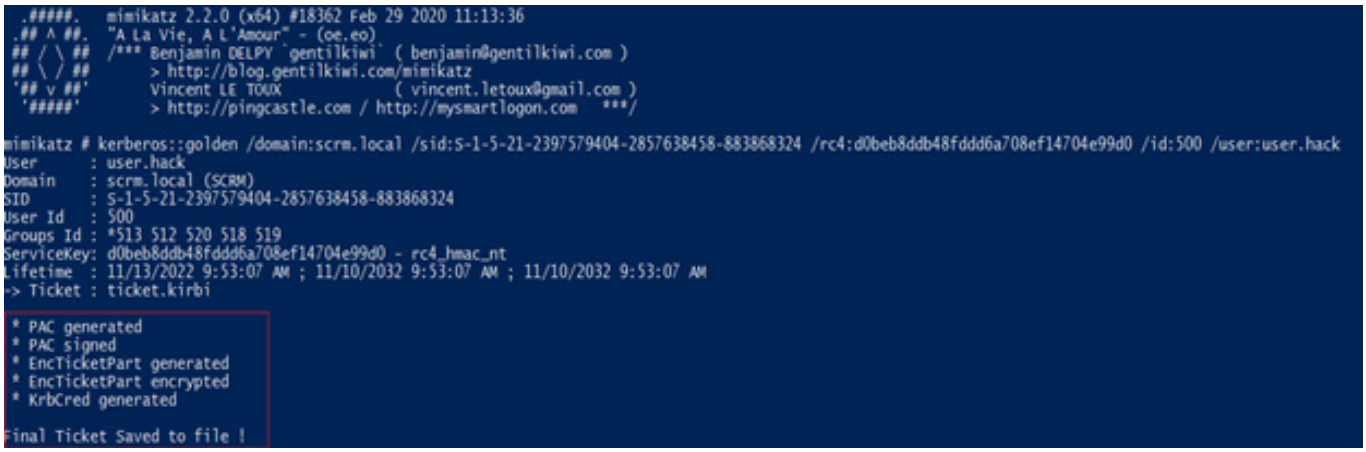


Figure 8: Listing when the golden ticket is generated.

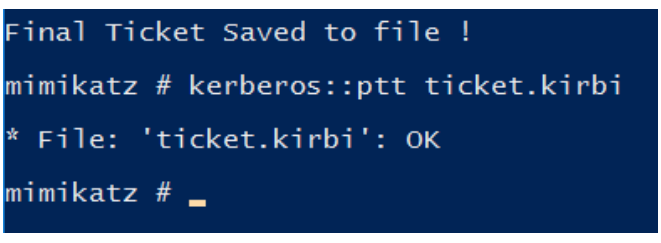


Figure 9: Assigning a ticket to the current session.

Copyright © Miloš Orgoň

Then we tried to reconnect using the SAMBA protocol to the domain controller (named "DCTEST") and tried to download the NTDS.dit database, which we have already done with the given TGT. In Figure 10 we can see that with the account "user.hack", which has no domain administrator privileges, we logged on to the domain controller and got up to the directory that hosts the NTDS.dit database.

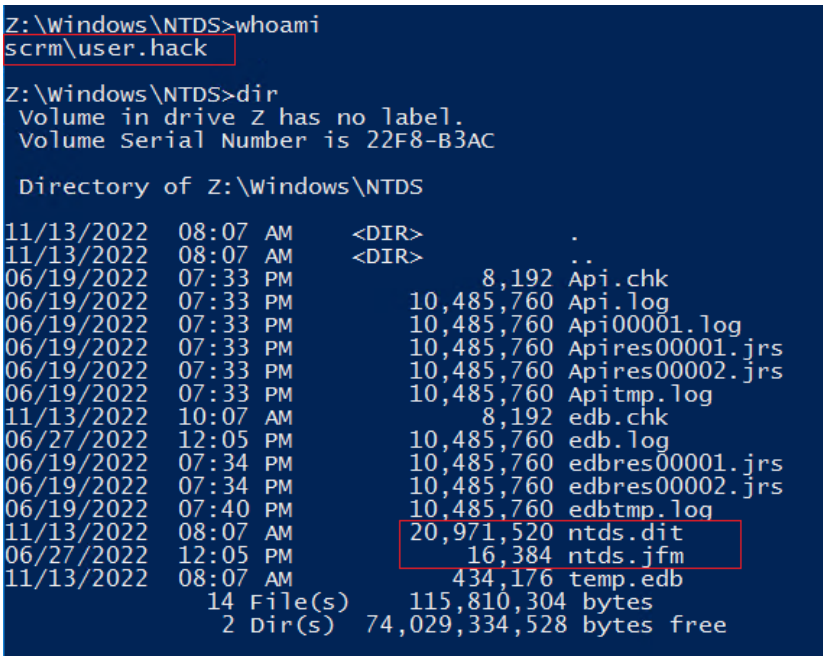


Figure 10: To log in to domain controller by user account using the samba protocol.

Conclusion

In this publication, we looked at the Active Directory Domain Services role and wanted to implement a Golden Ticket Attack, which we did. We created a TGT ticket that was signed with the account hash "krbtgt" and then spoofed it into the system. We created the Golden TGT ticket for 10 years and with it we were able to access the NTDS.dit database. In this database, there are stored domain objects, user accounts and their password hashes. Administrators can minimize this type of attack by periodically resetting the password of the "krbtgt" account, always at least 2 times in a row, because the system always remembers the last 2 hashes. At the same time, administrators must minimize logging into the system with an administrator account (not a local account and not a domain account).

Acknowledgement

This is a part of research activities conducted at Slovak University of Technology Bratislava within the scope of the project KEGA 034STU-4/2021

"Utilization of Web-based Training and Learning Systems at the Development of New Educational Programs in the Area of Optical Wireless Technologies ". This publication was based on work from the COST Action NEWFOCUS CA19111, supported by COST (European Cooperation in Science and Technology).

References

1. Alessandro P, Scardaci D, Liampotis N, Spinoso V, Grenier B, et al. (2020) Authentication, authorization, and accounting. In: Zhao Z, Hellström M (Eds.), Towards Interoperable Research Infrastructures for Environmental and Earth Sciences, Springer, Newyork, USA.
2. Das A, Bonneau J, Caesar M, Borisov N, Wang XF (2014) The tangled web of password reuse. In: Jonsson J, Kalisky B (Eds.), Public-Key Cryptography Standards (PKCS), Internet Engineering Task Force, Fremont, California, USA.
3. (2020) New research: Most people have 70-80 passwords. Newswire, London, UK.
4. Desmond B, Richards J, Allen R, Lowe-Norris AG (2013) Active directory. (5th edn), O'Reilly Media, Sebastopol, California, USA.
5. Microsoft (1993) Microsoft workgroup add-on for windows, user's guide, Microsoft Corporation, Bothell, Washington, USA.
6. Hunter LE (2005) Active directory field guide. (1st edn), Apress, New York, USA, pp. 352.
7. Neuman C, Yu T, Hartman S, Raeburn K (2005) The kerberos network authentication service (V5). Network Working Group, Internet Engineering Task Force, Fremont, California, USA, pp. 1-138.
8. Karel B (2019) Cryptography is all around us, CZ.NIC, Prague, Czech Republic, Europe.
9. Conrad E, Misenar S, Feldman J (2015) CISSP studz guide. (3rd edn), Elsevier, Amsterdam, Netherlands.
10. https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/dn408190(v=ws.11)
11. Grillenmeier G (2021) Now's the time to rethink Active Directory security. Network Security 2021(7): 13-16.
12. Brabety S (2020) Penetration testing mit mimikatz. (edn), O'Reilly Media, Sebastopol, California, USA.

Copyright © Miloš Orgoň