Kerberos V5

Franck Jeannot

Montréal, Canada, July 2023, Y760, v1.0

Abstract

This article reviews various implementations of the Kerberos protocol, with a focus on identifying some associated security issues, vulnerabilities, and weaknesses. It provides an intoduction analysis of notable attacks such as the Golden Ticket and Silver Ticket attacks, which exploit the trust Kerberos places in its tickets to gain unauthorized access to resources. It also discusses the threat posed by attackers using the Kerberoasting technique to crack the passwords of service accounts in a Windows domain. Despite these vulnerabilities, the Kerberos protocol remains a crucial tool for authenticating client-server applications and verifying users' identities in an untrusted network. Understanding these vulnerabilities is essential for developing more secure systems and mitigating potential threats.

Keywords: PAC, KDC, TGT, PKINIT, Kerberos, krb5-1.21, Golden Ticket, Silver Ticket, Kerberoasting, authentication, service accounts, vulnerabilities, TGS, PtT (pass the hash), KRBTGT, wireless implementations

1. Introduction

Kerberos V5 is a credential-based network authentication protocol, designed [1] to provide strong authentication for client/server applications using symmetric (secret) key cryptography [1]. This protocol relies on a trusted third party, known as the Key Distribution Center (KDC), to negotiate shared session keys between clients and services, thereby facilitating mutual authentication.

On the 5th Jun 2023, was released version 1.21 of Kerberos 5 (krb5-1.21)[2]. Beginning with release 1.20, the Key Distribution Center (KDC) started including minimal

[1] https://github.com/krb5/krb5.git

[2] https://web.mit.edu/kerberos/krb5-1.21

PACs [3] in tickets instead of **AD-SIGNEDPATH** authdata [4]. This change was made to improve the security and efficiency of the protocol. S4U requests [5] (protocol transition and constrained delegation) must now contain valid PACs in the incoming tickets.

Beginning with release 1.21, service ticket PACs will contain a **new KDC checksum buffer** [6]. This change was implemented to mitigate a hash collision attack against the old KDC checksum, enhancing the security of the protocol. However, this update has implications for mixed environments: if only some KDCs in a realm have been upgraded to versions 1.20 or 1.21, the upgraded KDCs will reject S4U requests containing tickets from non-upgraded KDCs and vice versa. This could potentially disrupt service in environments where not all KDCs are upgraded simultaneously [2].

Kerberos is widely used to authenticate service requests between trusted hosts across untrusted networks, such as the internet. It ensures that both the client and server verify each other's identity before transmitting application data. The client must specify the principal name of the server, and the server's identity must match that principal name exactly. If there's a mismatch or if the client specifies NULL for the server's principal name, the authentication process fails, preventing unauthorized access [1].

Kerberos is a computer network security protocol that authenticates service requests between two or more trusted hosts across an untrusted network, like the internet. It uses secret-key cryptography and a trusted third party for authenticating client-server applications and verifying users' identities [3].

The Kerberos V5 protocol provides a mechanism for mutual authentication between a client and a server before application data is transmitted between them [4]. The Kerberos protocol requires mutual authentication and supports it remotely. The client must specify the principal name of the server, and the server's identity must match that principal name exactly. If the client specifies NULL for the server's principal name or if the principal name doesn't match the server, the call will fail [5].

[3] Privilege Attribute Certificate

[4] The AD-SIGNEDPATH data is used to verify the integrity of the transited field in a ticket. The transited field is used to list the domains that have been traversed in the authentication process

[5] S4U stands for "Service-for-User". It's a part of the Kerberos protocol that allows a service to obtain a ticket on behalf of a user. There are two types of S4U operations: S4U2self (also known as protocol transition) and S4U2proxy (also known as constrained delegation).

[6] https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-pac/3341cfa2-6ef5-42e0-b7bc-4544884bf399

2. PKINIT

PKINIT [7] is an acronym for *Public Key Cryptography for Initial Authentication in Kerberos*. It is a set of protocol extensions for Kerberos[8] that seamlessly integrates public key cryptography into the initial authentication exchange. This integration is achieved by utilizing asymmetric-key signature and/or encryption algorithms in pre-authentication data fields [6].

The primary motivation behind the development of PKINIT was to overcome the limitations of the traditional Kerberos protocol, which relied heavily on symmetric key cryptography. While symmetric key cryptography offers advantages in terms of computational efficiency, it suffers from key distribution problems. In a large network with numerous clients and services, securely distributing and managing symmetric keys can become a significant challenge.

PKINIT addresses this challenge by leveraging the strengths of public key cryptography. In PKINIT, a client proves its identity to the Key Distribution Center (KDC) by signing a timestamp with its private key. The KDC, which has a copy of the client's public key, can verify the signature to authenticate the client. This eliminates the need for the client and KDC to share a symmetric key in advance.

Furthermore, PKINIT enhances the security of the Kerberos protocol by providing stronger resistance against password guessing attacks. In the traditional Kerberos protocol, an attacker could potentially compromise a client's password by capturing the client's authentication traffic and performing an offline dictionary attack. With PKINIT, this attack is no longer feasible because the client's private key is never transmitted over the network.

However, it's important to note that while PKINIT improves the security and scalability of Kerberos, it also introduces new challenges. For instance, the KDC must now manage a database of public keys or certificates, which can be a complex task in a large network. Additionally, the use of public key cryptography introduces additional computational overhead, which can impact the performance of the Kerberos protocol.

3. PAC and vulnerabilities

The **PAC** stands for Privilege Attribute Certificate. It's a Microsoft extension to the Kerberos protocol. The PAC contains various types of authorization data, including the user's group membership data and other information. When the KDC issues a ticket, it includes the PAC in the ticket. This means that when a client

[7] https://web.mit.edu/kerberos/krb5-1.13/doc/admin/pkinit.html?highlight=pkinit

[8] https://datatracker.ietf.org/doc/html/rfc4556

presents a ticket to a server, the server can extract the PAC from the ticket and use it to determine the client's access rights.

There have been some known attacks on the Kerberos protocol after 2022. In November 8, 2022, Windows released updates to address security bypass and elevation of privilege vulnerabilities with Privilege Attribute Certificate (PAC) signatures. This security update addresses Kerberos vulnerabilities where an attacker could digitally alter PAC signatures, raising their privileges [7]. Another vulnerability addressed by the November 8, 2022 Windows updates is related to Authentication Negotiation by using weak RC4-HMAC negotiation [8] (refer to CVE-2022-37967 [7] and CVE-2022-37966 [8]).

4. Linux Implementations of Kerberos

On Linux, the **krb5-pkinit** package contains the PKINIT plugin, which allows clients to obtain initial credentials from a Key Distribution Center (KDC) using a private key and a certificate [9]. This package is part of the larger Kerberos V5 software suite available for Linux systems, which provides a comprehensive implementation of the Kerberos protocol.

The Kerberos V5 software suite includes libraries for Kerberos 5, utilities for managing Kerberos databases, user and administrative utilities, and a PAM (Pluggable Authentication Modules) module for integrating Kerberos authentication with other system services [9]. The suite also includes the KDC itself, which can be run as a service on a Linux server to provide authentication services for a network of clients.

The implementation of Kerberos in Linux is highly configurable, allowing for a wide range of deployment scenarios. For example, the krb5.conf configuration file allows administrators to specify the default realm, the locations of KDCs and admin servers, logging settings, and many other options [10]. The kadmin utility provides a command-line interface for managing the Kerberos database, including adding and deleting principals, changing passwords, and modifying policy settings [11].

In addition to the standard Kerberos V5 software suite, there are also several other Kerberos-related packages available for Linux. These include krb5-user, which provides basic user programs such as kinit, klist, and kdestroy; krb5-doc, which provides the Kerberos V5 Installation Guide and other documentation; and krb5-kdc-ldap, which allows the KDC to use an LDAP directory as its database backend [12].

Despite the robustness of the Kerberos implementation in Linux, it is not without its challenges. For example, integrating Kerberos with other authentication systems can be complex, and managing Kerberos tickets can be difficult for end users. However, with careful planning and configuration, Kerberos can provide a powerful and secure authentication solution for Linux environments.

5. Implementations for Wireless systems

Wang et al. (2010) [13] proposed a new key establishment scheme for wireless sensor networks. They employed **LU Composition techniques** [9] for mutual authenticated pairwise key establishment and integrated LU Matrix with Elliptic Curve Diffie-Hellman for anonymous pathkey establishment. This scheme was found to be efficient in achieving group key agreement and management [13].

Kerberos authentication has been studied and implemented in wireless systems also, particularly in wireless sensor networks. Siddique (2012) [14] proposed an authentication mechanism in wireless sensor networks using the Kerberos authentication scheme. This scheme provides a centralized authentication server, known as the Key Distribution Center (KDC), which authenticates users by providing them with a ticket to grant requests to the base station [14].

In another study, Pak Song-Ho et al. (2016) [15] proposed a PKINIT_AS Kerberos V5 authentication system that uses public key cryptography. They also discussed a method to implement the gssapi_krb authentication method and secure Internet service using it in IPSec VPN [15].

[9] LU Composition is a method to break down a complex matrix into simpler parts for easier calculations

6. Simplified Sequence Diagram and Golden Ticket attack

The simplified flow to get a service ticket with a focus on the PKINIT portion is as below :

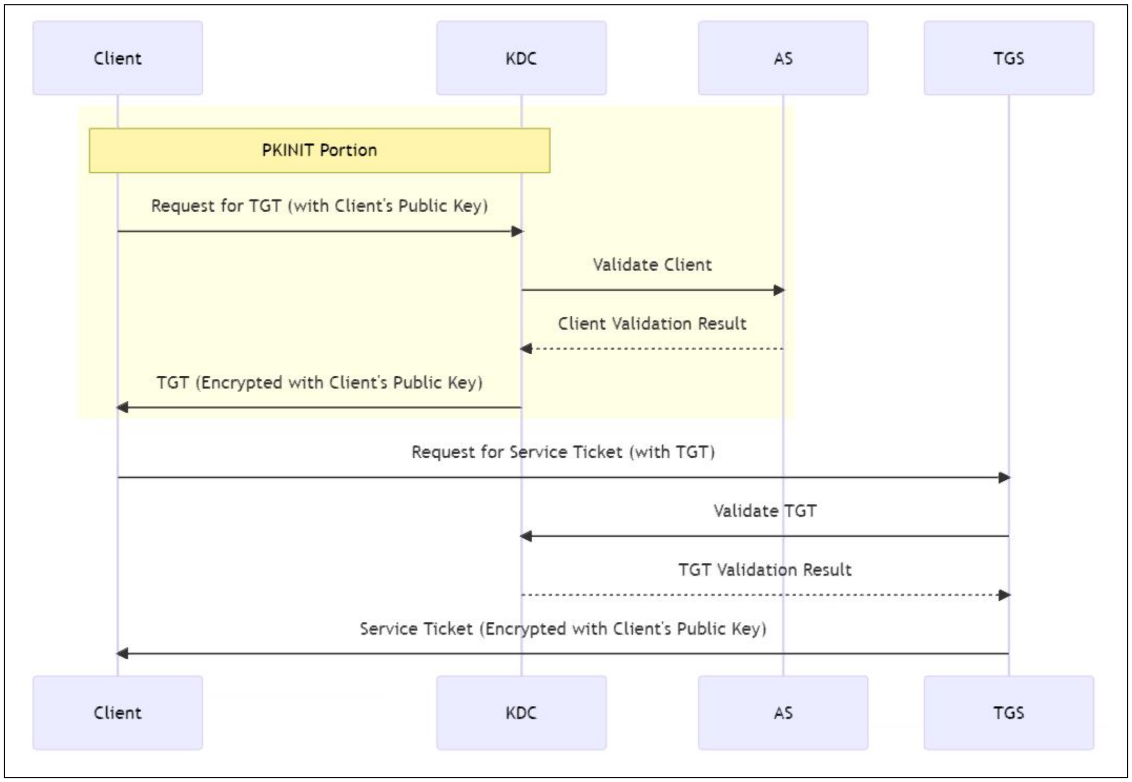
Client
KDC
AS
TGS
PKINIT Portion
Request for TGT (with Client's Public Key)
Validate Client
Client Validation Result
TGT (Encrypted with Client's Public Key)
Request for Service Ticket (with TGT)
Validate TGT
TGT Validation Result
Service Ticket (Encrypted with Client's Public Key)
Client
KDC
AS
TGS

Figure (1): Kerberos protocol with the PKINIT portion

6.1. Golden ticket attack

A "Golden Ticket" attack is a type of attack on the Kerberos protocol where an attacker with domain admin rights can compromise the KRBTGT account[10]. Using the KRBTGT account, a malicieous actor can create a Kerberos ticket granting ticket (TGT) that provides authorization to any resource and set the ticket expiration to any arbitrary time. This fake TGT is called a "Golden Ticket" and allows attackers to achieve network persistence [16] [17] [18] [19].

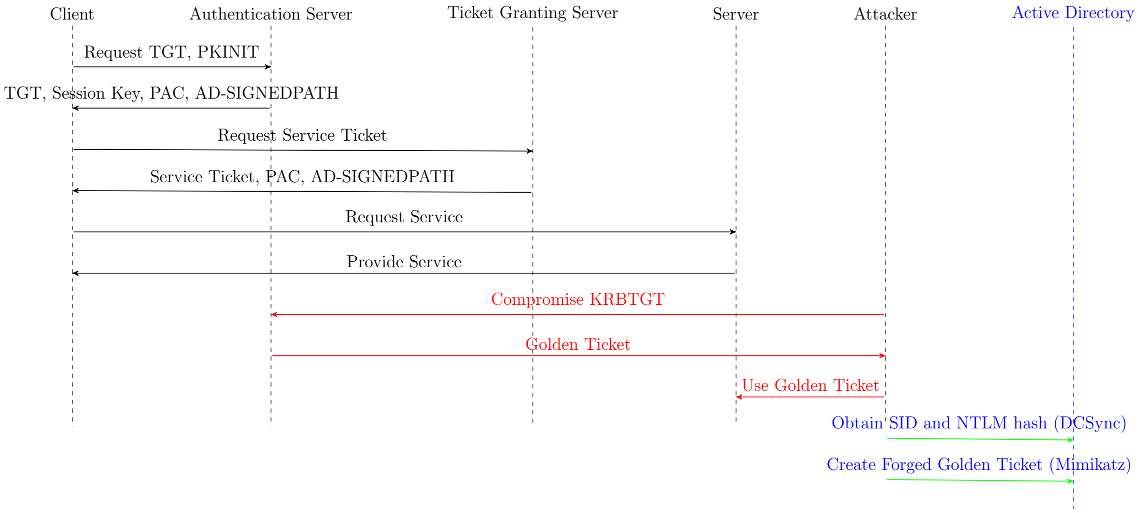

Figure (3): Golden ticket : simplified sequence

Figure (2): T1558.001 - Steal or Forge Kerberos Tickets, Golden Ticket

Its security is crucial as its compromise can lead to the creation of valid tickets for any user, granting unauthorized access to resources.

[10] The KRBTGT (Key Distribution Center Service) Account is a built-in account in the Kerberos protocol used to encrypt and sign all Kerberos tickets within a domain. It's essentially the "master key" for the Kerberos Ticket Granting Ticket (TGT) service.

A threat actor with a valid KRBTGT account hash can create a forged Golden Ticket using an open-source tool such as Mimikatz. Actors may also use DCSync, a Mimikatz feature, to obtain the security identifier (SID) of the KRBTGT account and NTLM hash using the. Threat actors then use these hashes to create their Golden Ticket and potentially run a **Pass the Ticket (PtT) attack** [20], moving laterally within an organization's AD environment.

A **Golden Ticket attack** and a **Silver Ticket attack** are two different types of attacks that exploit the Kerberos authentication system :

1. **Golden Ticket Attack:** This attack involves the creation of a Kerberos Ticket Granting Ticket (TGT) by an attacker who has gained access to the Key Distribution Center Service Account (KRBTGT) hash. The attacker can create a TGT with any user privileges, including domain administrator rights, and set the ticket's lifetime to an arbitrary length. This allows the attacker to maintain persistence in the network and access any resource in the domain.
2. **Silver Ticket Attack:** This attack [21] is more targeted and involves the creation of a Kerberos service ticket (TGS) by an attacker who has gained access to a service account's NTLM hash. The attacker can create a service ticket with any user privileges for a specific service on a specific server.

In both cases, the attacks exploit the trust that the Kerberos protocol places in its tickets, allowing attackers to impersonate any user and gain unauthorized access to resources.

7. Implementation of Kerberos protocol in Mermaid language

In mermaid code we have below a simplified version of kerberos protocol :

```
participant C as Client
participant AS as Authentication Server
participant TGS as Ticket Granting Server
participant S as Server
C->>AS: Request TGT
AS-->>C: TGT and Session Key
C->>TGS: Request Service Ticket
TGS-->>C: Service Ticket
C->>S: Request Service
S-->>C: Provide Service
```

8. Kerberos V4 and V5: history

Version 4 of Kerberos protocol was primarily designed by Steve Miller and Clifford Neuman [1] and ended in 2006 due to various vulnerabilities [11] [12]. Clifford Neuman and John Kohl published [22] version 5 in 1993 with the intention of overcoming existing limitations and security problems.

Let's review the sequence diagram of the original V4 version : K is the key and $K_{c,s}$ the ticket. Once the authentication is established, the client and server share a common session key $K_{c,s}$, which has never been transmitted over the network without being encrypted. Also, in version 4, message **2** is $\{K_{c,tgs}, n, \{T_{c,tgs}\}\, K_{tgs}\}\, K_c$, and message 4 is $\{K_{c,s}, n, \{T_{c,s}\}\, K_s\}\, K_{c,tgs})$. In version 4, the 5 steps to get a tickets were :

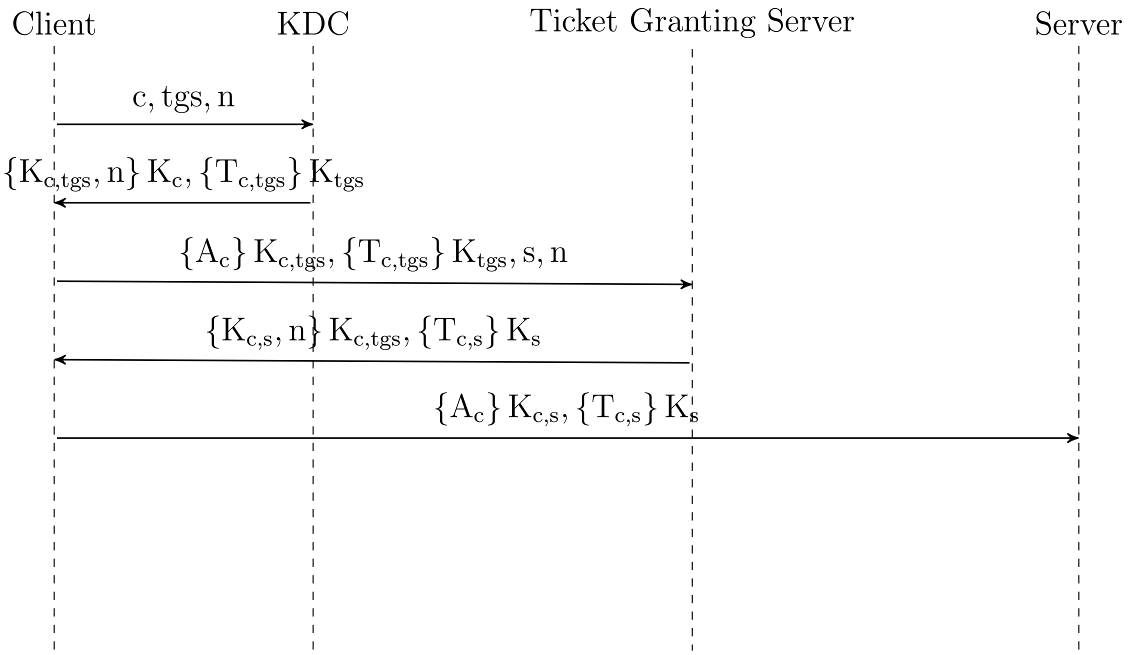
Client
KDC
Ticket Granting Server
Server
c, tgs, n
$\{K_{c,tgs}, n\}\, K_c, \{T_{c,tgs}\}\, K_{tgs}$
$\{A_c\}\, K_{c,tgs}, \{T_{c,tgs}\}\, K_{tgs}, s, n$
$\{K_{c,s}, n\}\, K_{c,tgs}, \{T_{c,s}\}\, K_s$
$\{A_c\}\, K_{c,s}, \{T_{c,s}\}\, K_s$

Figure (4): Kerberos V4 original simplified sequence, (inspired from [22])

Version 5 appeared as RFC 1510 [13], which was then made obsoleted by RFC 4120 [14] in 2005.

[11] https://web.mit.edu/kerberos/krb4-end-of-life.html

[12] http://web.mit.edu/tlyu/papers/krb4peril-ndss04.pdf

[13] http://www.faqs.org/rfcs/rfc1510.html

[14] https://datatracker.ietf.org/doc/html/rfc4120

9. Kerberoasting

Kerberoasting [23] [24] [25] is a technique used by attackers to crack the passwords of service accounts in a Windows domain. In a typical Kerberos setup, a Ticket Granting Service (TGS) ticket is used to authenticate to services in a domain. When a user requests access to a service, the Key Distribution Center (KDC) issues a TGS ticket, which is encrypted with the service account's password.

In a Kerberoasting attack [15], an attacker with access to a valid Kerberos ticket (even a low-privileged user) can request a TGS ticket for any service. Since the TGS ticket is encrypted with the service account's password, the attacker can then take this ticket and attempt to crack it offline, without any risk of detection. If the service account's password is weak, the attacker can crack the password and gain unauthorized access to the service :

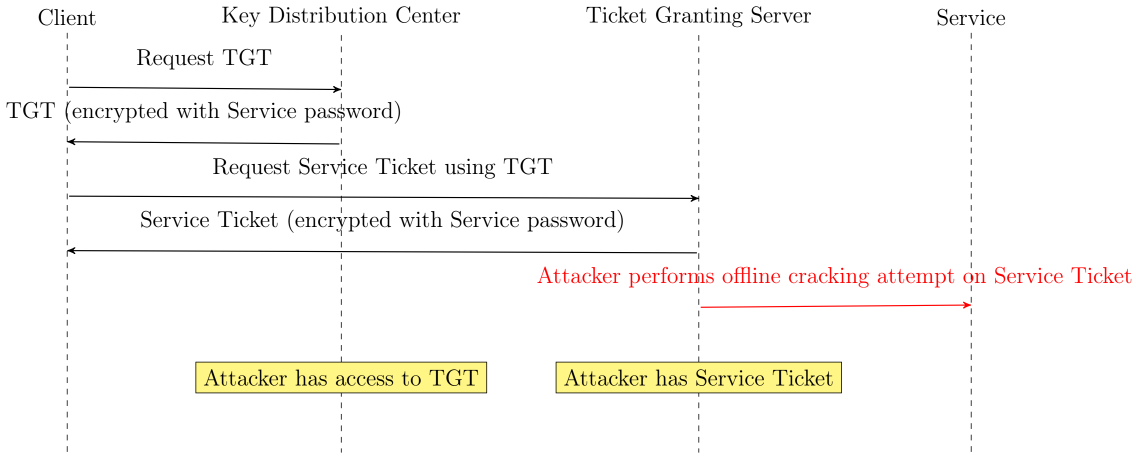
Client
Key Distribution Center
Ticket Granting Server
Service
Request TGT
TGT (encrypted with Service password)
Request Service Ticket using TGT
Service Ticket (encrypted with Service password)
Attacker performs offline cracking attempt on Service Ticket
Attacker has access to TGT
Attacker has Service Ticket

Figure (5): Kerberoasting attack simplified diagram

[15] https://attack.mitre.org/techniques/T1558/003/

References

[1] Neuman, B. Clifford and Ts'o, Theodore, Kerberos: An authentication service for computer networks, IEEE Communications Magazine 32 (9) (1993) 33–38.
URL https://ieeexplore.ieee.org/document/312841/

[2] Kerberos 5 Release 1.21, https://web.mit.edu/kerberos/krb5-1.21/.

[3] Simplilearn, Understanding Kerberos: How It Works and Authentication Explained!, https://www.simplilearn.com/what-is-kerberos-article (2023).

[4] Microsoft, Microsoft Kerberos Protocol Extensions, https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-kile/b4af186e-b2ff-43f9-b18e-eedb366abf13 (2019).

[5] Microsoft, Kerberos v5 protocol, https://docs.microsoft.com/en-us/windows/win32/com/kerberos-v5-protocol (2019).

[6] Zhou, Jianying and Gollmann, Dieter, Public key cryptography and password protocols: the multi-step BPKI, in: European Symposium on Research in Computer Security, Springer, 2005, pp. 114–129.
URL https://link.springer.com/chapter/10.1007/11555827_8

[7] Kb5020805: How to manage kerberos protocol changes related to cve-2022-37967, https://support.microsoft.com/en-us/topic/kb5020805-how-to-manage-kerberos-protocol-changes-related-to-cve-2022-37967-997e9acc-67c5-48e1-8d0d-190269bf4efb.

[8] Kb5021131: How to manage the kerberos protocol changes related to cve-2022-37966, https://support.microsoft.com/en-gb/topic/kb5021131-how-to-manage-the-kerberos-protocol-changes-related-to-cve-2022-37966-fd837ac3-cdec-4e76-a6ec-86e67501407d.

[9] MIT, Kerberos V5 Installation Guide (2021).
URL https://web.mit.edu/kerberos/krb5-1.12/doc/admin/install_kdc.html

[10] MIT, krb5.conf - Kerberos 5 configuration file (2021).
URL https://web.mit.edu/kerberos/krb5-1.12/doc/admin/conf_files/krb5_conf.html

[11] MIT, kadmin - Kerberos 5 database administration program (2021).
URL https://web.mit.edu/kerberos/krb5-1.12/doc/admin/admin_commands/kadmin_local.html

[12] Debian, Debian – details of package krb5-user in buster (2021).
URL https://packages.debian.org/buster/krb5-user

[13] Wang, Eric Ke and Hui, Lucas C. K. and Yiu, S. M., A new key establishment scheme for wireless sensor networks (2010). arXiv:1004.0591.
URL https://arxiv.org/abs/1004.0591

[14] Q. Siddique, Kerberos authentication in wireless sensor networks (2012).
URL http://arxiv.org/abs/1203.0640v1

[15] Pak, Song-Ho and Pak, Myong-Suk and Jang, Chung-Hyok, A method to Implement the Kerberos User Authentication and the secured Internet Service (2016). arXiv:1604.08799.
URL https://arxiv.org/abs/1604.08799

[16] Persistence and privilege escalation security alerts, https://learn.microsoft.com/en-us/defender-for-identity/persistence-privilege-escalation-alerts.

[17] Jeff Warren, Complete Domain Compromise with a Golden Ticket Attack, https://blog.netwrix.com/2022/08/31/complete-domain-compromise-with-golden-tickets/ (2022).

[18] Golden Ticket Attack: Detecting and Preventing, https://frsecure.com/blog/golden-ticket-attack/.

[19] How kerberos golden ticket attacks are signaling a greater need for identity-based security, https://www.sentinelone.com/blog/how-kerberos-golden-ticket-attacks-are-signaling-a-greater-need-for-identity-based-security/ (2022).

[20] Use Alternate Authentication Material: Pass the Ticket, https://attack.mitre.org/techniques/T1550/003/.

[21] T. Grippo, H. A. Kholidy, Detecting Forged Kerberos Tickets in an Active Directory Environment (2022).
URL https://arxiv.org/abs/2301.00044

[22] J. Kohl, B. Neuman, T. Ts'o, The Evolution of the Kerberos Authentication Service, IEEE Computer Society Press, Los Alamitos, CA, 1994.
URL https://www.cs.fsu.edu/~awang/courses/cop5611_s2023/kerberos.pdf

[23] J. Garcia, A. Sopelana, M. Urueña, Kerberoasting: A critical analysis of attacks and defenses, in: 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), IEEE, 2019, pp. 1–9.

[24] B. Hansen, Kerberoasting: A practical guide, in: 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), IEEE, 2019, pp. 1–8.

[25] V. Shastri, What is a kerberoasting attack?, https://www.crowdstrike.com/cybersecurity-101/kerberoasting/ (2023).