Understanding Kerberos Part 1: The Basics

Jochen Kressin · 2024-05-27

In the realm of network security, ensuring the authenticity of users in a multi-user network environment is paramount. Kerberos, a robust authentication protocol, serves as the backbone for enabling secure access to multiple services over a network without requiring repeated credential submissions by the user (Single Sign On). This article delves into the details of Kerberos, explores its most common applications, and discusses its advantages and disadvantages over other authentication methods like SAML (Security Assertion Markup Language) and OIDC (OpenID Connect).

What is Kerberos?

Kerberos is an authentication protocol that uses secret key cryptography to confirm the identities of users and their eligibility to access various services within a network. Developed in the 1980s at MIT as part of Project Athena, Kerberos was designed to provide strong authentication for client/server applications.

Watch the video in the online version of this article.

How Does Kerberos Work?

The operation of Kerberos revolves around a Client/Server model and involves a trusted third-party component known as the *Key Distribution Center* (KDC), which consists of two parts: the *Authentication Server* (AS) and the *Ticket Granting Server* (TGS). Here's a step-by-step breakdown of the Kerberos authentication process:

- **Authentication Request**: When a user (client) attempts to access a service, they submit an authentication request to the AS. This request includes the user's identifier and the service they wish

to access, securely hashed with the user's password.

- **AS Response**: If the user is found in the KDC's database, the AS responds by sending back a TGT (Ticket-Granting Ticket), encrypted using a secret key derived from the user's password.
- **TGT to TGS**: The client decrypts the TGT using their password, and if access to the service is required, the client sends a service request to the TGS, including the TGT and the identifier of the requested service.
- **Service Ticket**: The TGS validates the TGT and if it's valid, issues a service ticket to the client. This ticket is encrypted with the service's secret key.
- **Accessing the Service**: The client then sends this service ticket to the service server. The server decrypts the ticket using its own secret key, verifies it, and grants access to the client.

This sequence ensures that the user's password need not be sent over the network during any of these transactions, significantly enhancing security.

The operation of Kerberos revolves around a client/server model

They communicate over an unsecure network, thus there is a need for a trusted 3rd party: Key distribution center (KDC)

The KDC comprises of two servers: the Authentication Server (AS) and the Ticket Granting Server (TGS)

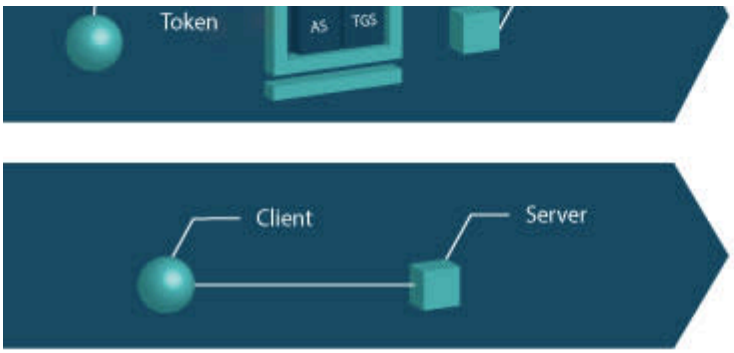
The CLIENT send a request to the AS: "My user id is XXX and I need a TICKET for SERVICE xyz"

The AS based on the user ID retrieves the PASSWORD from the database, this is how the USER is VERIFIED

Ticket Granting Ticket (TGT) encrypted with a secret key derived from the user's password is sent back to the CLIENT

The CLIENT sends the TGT to the TGS alongside the shared secret key and a REQUEST. The TGT decrypts the ticket using a shared secret key with the AS.

TGS sends the CLIENT a TICKET encrypted with the service's secret key.



The CLIENT sends the TICKET to the SERVICE SERVER

SERVICE SERVER decrypts the TICKET with the it's secret key verifies it, and grants access to the CLIENT

Primary Uses of Kerberos

Kerberos is predominantly used in secure network environments where HTTP requests are frequent and need to be authenticated quickly and securely. Common implementations of Kerberos include:

- **Enterprise Environments**: Integration with Active Directory for Windows domains.
- **Network File Systems**: Particularly with protocols like NFS that require each access request to be authenticated.
- **Email Services**: Securing SMTP, IMAP, and POP access.
- **Database Access**: Authentication for database servers like Oracle or SQL Server.

Benefits of Kerberos

Kerberos provides several benefits as a single sign-on authentication protocol:

- **Strong Security**: By using secret-key cryptography, it avoids the pitfalls of transmitting passwords over the network, mitigating the risk of password theft.
- **Scalability**: It handles a large number of requests without significant degradation in performance, making it suitable for large enterprises.
- **Reduced Password Fatigue**: As a single sign-on technology, it reduces the need for multiple passwords and user credentials, simplifying user access to services.
- **Interoperability**: Kerberos is supported across various platforms including Unix, Windows, and several network applications, providing broad compatibility.

Where there is light, there is Shadow

While Kerberos is a powerful and secure authentication protocol that offers numerous benefits, it also has several disadvantages and limitations that can impact its deployment and operation in certain environments.

Complexity of Setup and Maintenance

Kerberos requires a centralized KDC, which needs to be properly installed, configured, and maintained. This setup can be complex and requires careful planning and expertise:

- **Single Point of Failure**: The KDC represents a single point of failure within the network. If the KDC goes down, new authentication requests cannot be processed, preventing user access to services.

- **Scalability Issues**: While Kerberos itself is scalable, the KDC can become a bottleneck in large environments with many simultaneous authentication requests.

Sensitive Time Synchronization

Kerberos protocol heavily relies on time stamps to prevent replay attacks, where an old authentication request is resent to trick the system. This requirement means:

- **Time Synchronization**: All client machines, as well as the servers, must have closely synchronized clocks within the network. If clocks are not properly synchronized, legitimate authentication requests may be rejected.

Limited Offline Capabilities

Kerberos requires continuous access to the KDC for authentication which can be a significant limitation in environments where network connectivity is intermittent or unreliable:

- **Network Dependency**: Users must be able to communicate with the KDC to access services, which can be problematic in distributed environments or in situations where network connectivity is compromised.

Cross-Domain Authentication Complexity

While Kerberos supports cross-domain authentication, configuring inter-realm trust can be complex and cumbersome. This complexity increases with the number of realms involved:

- **Trust Relationships**: Each Kerberos realm must establish a trust relationship with other realms, which can complicate the setup and increase the administrative overhead.

Troubleshooting and Diagnostics

Diagnosing issues within a Kerberos-enabled environment can be difficult due to the protocol's opaque error messages and complex interactions between clients, the KDC, and servers:

- **Complex Diagnostics**: Identifying the root cause of authentication failures can be challenging, requiring detailed logs and expertise in Kerberos' intricate mechanisms.

Conclusion

Kerberos stands out as a powerful, secure, and efficient authentication protocol well-suited to complex network environments. Its ability to provide a secure single sign-on experience helps organizations streamline their security protocols and user access management.

However, its deployment and ongoing management can be complex and may pose challenges in environments not well-suited to its operational requirements. Organizations considering Kerberos must weigh these disadvantages against its security benefits and consider whether alternative solutions like SAML or OIDC might better meet their needs, especially in predominantly web-based or cloud-focused environments.

This article was published on the Search Guard blog: https://search-guard.com/blog/kerberos-elasticsearch-searchguard-pt1/
Search Guard is Security and Alerting for Elasticsearch and Kibana — search-guard.com · Start a free trial