Kerberos and Windows Security: History

medium.com/@robert.broeckelmann/kerberos-and-windows-security-history-252ccb510137

Robert Broeckelmann 1 июня 2018 г.

Robert Broeckelmann

Cerberus /

In previous posts, we explored various identity protocols including OAuth2, OpenID Connect, SAML2 profiles, WS-Trust, and WS-Federation. This post continues our exploration of identity protocols by looking at the Kerberos v5 authentication protocol and its use in Microsoft Windows authentication. I don't have nearly as much experience with Kerberos and Windows authentication as I do with the other identity protocols mentioned above, but Microsoft Windows is the front door to the network for many users (especially in the corporate scene). That being said, I'm writing this post on a laptop running Linux and most of my coworkers are using Macs. Regardless of those anomalies, Kerberos is something that needs to be understood for many identity scenarios. In the corporate/enterprise space, Kerberos (through Windows) is often the first thought for achieving Single Sign On. Grant it, it's effectiveness outside the corporate LAN (including VPN connections of various types) becomes questionable.

Unlike the other protocols that we've explored, Kerberos is not HTTP(S)-based. The first public release of Kerberos v4 predates the HTTP protocol by about two years. Likewise, the Kerberos protocol predates SSLv2 (first publically used version) by seven years. Thus, places where encryption are required within the Kerberos protocol are not relying upon transport layer encryption (as we know it today). In fact, it is fascinating that the Kerberos protocol has remained relevant this long. This is very likely due to its ubiquitous use in Microsoft Windows.

Several years ago, I was working at a client site where Active directory Federation Services (AD FS v2.0, henceforth referred to as ADFS) was used heavily. We seemed to constantly run into issues associated with ADFS having a Kerberos ticket associated with every SAML2 token that ADFS issued (whether created via WS-Trust, WS-Federation, or SAML Browser Profile). Legacy issues at this client surrounding domain controller configuration and architecture were causing scalability issues as the amount of traffic against Kerberos-related services increased dramatically as ADFS became a critical component of dozens of systems including an Enterprise Service Bus (ESB) that serviced tens of millions of SOAP calls a day (on some days, but always millions). It became an extreme disadvantage to have Kerberos tickets backing all the SAML2 assertions that were being issued by the Identity Provider — ADFS. This isn't a knock against Microsoft, Windows, or ADFS. It was a problem born of decades of bad architecture decisions within this organization — they probably aren't alone.

This series of posts ties the world of those domain controllers into our ongoing discussion of identity protocols in their real world.

History of Kerberos

I always like to spend some time exploring the history of a protocol before we jump into the technical details. The Kerberos authentication protocol was originally developed at MIT for Project Athena. The project goals included integrating[5]:

- Networked file systems (the original remote file subsystem was discontinued because of viable alternatives at the time, they started using , similar to)
- A unified graphical environment (they developed)
- Naming convention service (think , but Athena developed)

Obviously, it was a different era; these are common technologies that are taken for granted today (Q2, 2018)— not so much in the early 1980s. By 1988, Project Athena had achieved all of these goals. The SSO component resulted in Kerberos.

Kerberos used the Needham–Schroeder symmetric key protocol from 1978 (a symmetric key exchange protocol) as a starting point. The Kerberos v4 protocol was first publically described in this 1988 Usenex conference paper. That date makes Kerberos one of the oldest identity-related protocols in common use today. Compare it to SAML (2002), SAML2 (2005), WS-Trust (2007), WS-Federation (2003), OAuth2 (2010), and OpenID. Connect (2014).

Kerberos is the “most widely deployed system for authentication and authorization in modern computer networks” [6]. Given Kerberos is the basis of Microsoft Windows (which is widely deployed itself) security, this is easy enough to believe.

In Greek Mythology, Kerberos was the three-headed guard dog of Hades. Insert obligatory commentary about the Kerberos authentication protocol requiring a “third-party” (the Key Distribution Center) to handle communication between a client and service.

The current version of the protocol is v5. It was released in 1993 (and updated in 2005). A brief Kerberos version history is below.

v1-v3 — Development started in 1983. It was never used outside of MIT. It was a limited R&D implementation to figure out how things should work. I didn’t find much information on exactly what was in each of these versions—will update in the future if I find it.

v4 — Primarily designed by Steve Miller and Clifford Neuman. Some non-technical details:

- Also part of .
- Released on January 24th, 1989.
- Adopted by several operating system vendors.
- MIT provided a reference implementation.
- Until 2000, US export control restrictions on encryption technology (namely DES) prevented Kerberos from legally being used outside of the US.
- Eric Young, at Bond University of Australia, created a DES implementation that he used to add encryption to a Kerberos implementation that had all encryption removed.
- Another non-US Kerberos 4 implementation, KTH-KRB, was developed in Sweden.
- v4 was announced in 2006 due to vulernabilities in the DES algorithm (which was the only encryption algorithm supported by v4).

I’m not going to cover the technical details of v4 in this post. Check out this link for more info on v4.

v5 — Published in 1993 by Neuman and Kohl with the intention of overcoming v4 limitations and security problems (see link above). Version 5 appeared as RFC 1510. The following specs are related Kerberos v5

- —Kerberos v5
- RFC1964 — Kerberos Version 5 Generic Security Service Application Programming Interface (GSS-API) Mechanism. This defines a standard set of functions (API library) that Kerberos implementations should use.

The following features were added to Kerberos v5:

- Pre-authentication
- On-the-wire protocol based on

- Changes to pre-salt algorithm.
- Delegation
- Support for forwarding, renewing and postdating tickets.
- Replay cache
- Support for cross-realm authentication
- Extensible encryption types

v5-update — In 2005, the Internet Engineering Task Force (IETF) Kerberos working group released RFC 4120, which added numerous features and updated various cryptographic standards to keep Kerberos current and relevant. BThis included:

- — Encryption and Checksum Specifications for Kerberos 5
- — Advanced Encryption Standard (AES) Encryption for Kerberos 5
- — new edition of the Kerberos V5 spec: "The Kerberos Network Authentication Service (V5)"
- — new edition of the (GSS-API) spec: "The Kerberos Version 5 Generic Security Service Application Program Interface (GSS-API) Mechanism: Version 2."

Some of the issues with cryptographic standards that needed to be addressed are described here.

The enhancements included (from here):

- Deprecate the use of , RC4-HMAC-EXP, and other weak cryptographic algorithms in Kerberos.
- Introduced use of AES.
- Provide additional clarity where ambiguity existed in RFC 1510
- Implementers cannot automatically assume forward compatibility.
- Outlines the basis for backwards compatibility.
- Existing Kerberos message formats cannot readily be extended by adding fields to the ASN.1 types
- Many RFC 1510 implementations ignore unknown authorization data elements. Depending on these implementations to honor authorization data restrictions may create a security weakness.
- With the exception of the INVALID flag, clients MUST ignore ticket flags that are not recognized
- KDCs MUST ignore KDC options that are not recognized.
- Because new KDCs will ignore unknown options, clients MUST confirm that the ticket returned by the KDC meets their needs.
- New additions to the AP-REQ message
- Additional clarity about how to prevent replay type attacks without having to rely on other mechanisms like NONCE.

RFC4120 has since been updated by RFCs 4537, 5021, 5896, 6111, 6112, 6113, 6649, 6806, 7751, 8062, 8129. It has been nearly thirteen years since the latest Kerberos spec was published; so, it is easy enough to imagine that numerous details have had to be updated.

Kerberos v5 provides an extension mechanism that allows enhancements to be added without breaking backwards compatibility. There have been numerous extensions added over the years. Many of them are listed here.

There are several efforts that have supported Kerberos over the years including:

Historical downloads of the MIT implementation of Kerberos can be found here.

History of Kerberos-Based Windows Authentication

In modern Microsoft Windows, domain-based authentication of users and hosts is performed through Kerberos. Kerberos v5 (RFC1510) was introduced to the platform in Windows Server 2000; it replaced NTLM (Windows NT LAN Manager) as the default authentication option. NTLM is still used on standalone systems (ie, not domain joined).

Microsoft decided not to use RFC1964 (GSSPAI) in the original Kerberos implementation; their implementation, Security Support Provider Interface (SSPI), uses a similar set of functions to GSSAPI, but with extensions and Windows-specific data types. All Windows protocols use the Windows Authentication API (SSPI). The following SSPs (Security Support Providers) are available in Microsoft Windows:

- (NT Login Manager)
- (SChannel)
- Distributed Password Authentication (DPA)
- (PKU2U)

This post is about Kerberos and to a slightly lesser extent Kerberos in Microsoft Windows. So, we are not going into anymore detail on the other SSPs here; please refer to the Microsoft documentation referenced above for more information.

With the release of Windows Vista in November, 2006, Microsoft updated the Kerberos implementation to support RFC4120 (and associated RFCs, including replacing DES with AES).

Summary

That concludes our brief history of Kerberos. Kerberos v5 is one of the older identity protocols in common use today (initially introduced to the public in 1988). Kerberos came out of Project Athena at MIT. It is especially common in Microsoft Windows; though, it is supported by many popular platforms. In our next post, we will look at the Kerberos v5 protocol details.

Image: Cerberus /