Kerberos for Internet-of-Things

IETF89

Thomas Hardjono
MIT Kerberos & Internet Trust Consortium
February, 2014
Contents

• Kerberos Protocol Overview
• Kerberos in Devices
  – DOCSIS & PacketCable
  – Intel AMT
• Kerberos for IoT (pros & cons)
• History of Kerberos
Kerberos Protocol Overview
Kerberos Protocol Messages

1. AS-REQ
2. AS-REP
3. TGS-REQ
4. TGS-REP
5. AP-REQ
6. AP-REP

User

Kerberos Client

Kerberos Server

Authentication Server (AS)

Ticket Granting Server (TGS)

Service
Needham-Schroeder Protocol

\[ A \rightarrow S : A, B, Na \]
\[ S \rightarrow A : \{Na, B, K_{ab}, \{K_{ab}, A\}K_{bs}\}K_{as} \]
\[ A \rightarrow B : \{K_{ab}, A\}K_{bs} \]
\[ B \rightarrow A : \{Nb\}K_{ab} \]
\[ A \rightarrow B : \{Nb - 1\}K_{ab} \]

- S is Server (KDC)
- A and B are Client and Service
- N is nonce
- K is the shared symmetric key
Basic Flows

- Long term symmetric keys:
  - Client and KDC share unique long-term key
  - Service and KDC share unique long term key
- Long term keys used to establish session-keys
  - Used to encrypt Tickets & Authenticators
  - Ticket-Granting-Ticket (TGT) and Service Ticket
- Authenticator:
  - Encrypted by Client to provide Proof-of-Possession (POP) to intended recipient
### What’s Inside a Ticket Granting Ticket

<table>
<thead>
<tr>
<th>Ticket (Ticket Granting Ticket (TGT))</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ticket version No.</td>
<td>[tkt-vno]</td>
</tr>
<tr>
<td>Issuing Server’s Realm</td>
<td>[realm]</td>
</tr>
<tr>
<td>Server’s Principal Name</td>
<td>[sname]</td>
</tr>
<tr>
<td>Encrypted Ticket Part</td>
<td>[enc-part]</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Ticket Flags</td>
<td>[flags]</td>
</tr>
<tr>
<td>Session Encryption Key</td>
<td>[key]</td>
</tr>
<tr>
<td>Client’s Realm</td>
<td>[crealm]</td>
</tr>
<tr>
<td>Client’s Principal Name</td>
<td>[cname]</td>
</tr>
<tr>
<td>Transited Realms</td>
<td>[transited]</td>
</tr>
<tr>
<td>Time of initial authentication</td>
<td>[authctime]</td>
</tr>
<tr>
<td>Start Time of ticket (opt)</td>
<td>[starttime]</td>
</tr>
<tr>
<td>Expiration Time of ticket</td>
<td>[endtime]</td>
</tr>
<tr>
<td>Max renew time of ticket (opt)</td>
<td>[renew-till]</td>
</tr>
<tr>
<td>Client Host Addresses (opt)</td>
<td>[caddr]</td>
</tr>
<tr>
<td>Restrictions (opt)</td>
<td>[authorization-data]</td>
</tr>
</tbody>
</table>
What’s Inside a Service Ticket

<table>
<thead>
<tr>
<th>Field</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ticket version No.</td>
<td>[tkt-vno]</td>
</tr>
<tr>
<td>Issuing Server’s Realm</td>
<td>[realm]</td>
</tr>
<tr>
<td>Server’s Principal Name</td>
<td>[sname]</td>
</tr>
<tr>
<td>Encrypted Ticket Part</td>
<td>[enc-part]</td>
</tr>
<tr>
<td>Ticket Flags</td>
<td>[flags]</td>
</tr>
<tr>
<td>Session Encryption Key</td>
<td>[key]</td>
</tr>
<tr>
<td>Client’s Realm</td>
<td>[crealm]</td>
</tr>
<tr>
<td>Client’s Principal Name</td>
<td>[cname]</td>
</tr>
<tr>
<td>Transited Realms</td>
<td>[transited]</td>
</tr>
<tr>
<td>Time of initial authentication</td>
<td>[authtime]</td>
</tr>
<tr>
<td>Start Time of ticket (opt)</td>
<td>[starttime]</td>
</tr>
<tr>
<td>Expiration Time of ticket</td>
<td>[endtime]</td>
</tr>
<tr>
<td>Max renew time of ticket (opt)</td>
<td>[renew-till]</td>
</tr>
<tr>
<td>Client Host Addresses (opt)</td>
<td>[caddr]</td>
</tr>
<tr>
<td>Restrictions (opt)</td>
<td>[authorization-data]</td>
</tr>
</tbody>
</table>
## What’s Inside an Authenticator

<table>
<thead>
<tr>
<th>Authenticator version No.</th>
<th>[authenticator-vno]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client’s Realm</td>
<td>[realm]</td>
</tr>
<tr>
<td>Client’s Principal Name</td>
<td>[cname]</td>
</tr>
<tr>
<td>Checksum (opt)</td>
<td>[cksum]</td>
</tr>
<tr>
<td>Client’s time/microsecs</td>
<td>[cusec]</td>
</tr>
<tr>
<td>Client’s current time</td>
<td>[ctime]</td>
</tr>
<tr>
<td>Sub-key (opt)</td>
<td>[subkey]</td>
</tr>
<tr>
<td>Sequence Number (opt)</td>
<td>[seq-number]</td>
</tr>
<tr>
<td>Authorization data (opt)</td>
<td>[authorization-data]</td>
</tr>
</tbody>
</table>

© 2007-2014 The MIT Kerberos & Internet Trust Consortium. All Rights Reserved.
Kerberos in Devices & Embedded Systems
DOCSIS & Packet Cable

Call Management Server

TFTP Server

Provisioning Server

Key Distribution Center

Service Provider Network

IPsec ESP secures NCS protocol

Kerberized Key Management

Config File

authenticated with hash delivered via secured SNMPv3

SNMPv3 security

Kerberized Key Management

Kerberos/PKINIT

MTA

CM

CMTS

DOCSIS Security

Packet Cable 1.5 (PKT-SP-PROV1.5-I04-090624)
DOCSIS & Packet Cable

Kerberos/PKINIT

AS Request

MTA

AS Reply

KDC

Kerberized Key Management for SNMPv3 and IPsec

App Server (Prov Server or CMS)

AP Request

MTA

AP Reply

Encrypted Session Key

Kerberos Ticket

KDC RSA Signature

KDC Certificate

KDC DH Public Value

Service Provider CA Certificate

... Digitaly Signed by: Service Provider Root

Key Management Data – key lifetime, should client rekey?

Chosen Ciphersuite

Sequence #

Subkey

Application Specific Data - IPsec or SNMPv3

SHA-1 HMAC

Packet Cable 1.5 (PKT-SP-SEC1.5-I03-090624)
Kerberos in Intel® AMT

• Active Management Technology (AMT)
  – Manageability technology for Intel platforms (hardware, firmware, software)

• Out-of-Band Manageability:
  – OS-independent (i.e. Pre-OS) & runs on auxiliary power
  – Remote boot (Serial-over-LAN)
  – Remote diagnostics & firmware updates (pre-OS boot)
  – Remote OS repairs
  – Bound to PC hardware (difficult to tamper)

• IT Administrator must be authenticated by AMT device before performing AMT operations remotely
Kerberos in Intel® AMT

- Kerberized Service has local ACLs containing privileges per SID.
- Shares a service key with KDC
- AMT device decrypts & validates ticket & authenticator.
- AMT device decodes PAC data structure (authorization)

- AD schema extension for AMT Class
- Service Principal Name and Service Key established for each registered AMT platform
Other Embedded Case Studies

• See TeamF1 presentation:
  – Data Center authentication
  – VPN termination device
  – Industrial automation

Kerberos.org and Other Links

• Needham-Schroeder paper (1978):
  – Also see Denning-Sacco paper (1981) *CACM* 24 (8): 533–535

• Kerberos RFC 4120:

• MIT Code Base distribution (now Rel 1.12)

• Kerberos APIs documentation:

• Some Guides:
Kerberos in Constrained Devices
Kerberos for IoT: the Pros

• Well understood protocol (cf. Needham-Schroeder)
• Symmetric-key approach suits constrained devices
  – Long-term keys can be installed by device manufacturer
  – Symmetric key operations cheaper/faster
  – Kerberos flows can be optimized for IoT devices
• Integration with directories a well-trodden path
• Open source code (20+ years)
  – MIT code written in C – several generations of coders
  – Active dev community
Kerberos in IoT: the Cons

• RFC4120 will put you to sleep…😊
• No initial key distribution protocol
  – Use PKINIT (RFC6112) or similar
• Good C programmers hard to come-by
Kerberos History & Status
A Brief History of Kerberos

• Kerberos was developed as the Authentication engine for MIT’s Project Athena in 1987:
  – Became IETF standard in 1993 (RFC1510) – now RFC4120
• MIT’s release of Kerberos as open source in 1987 led to rapid adoption by numerous organizations
• Kerberos now ships standard with *all* major operating systems
  – Apple, Red Hat, Microsoft, Sun, Ubuntu
• Serves tens of millions of enterprise users:
  – Microsoft has been using Kerberos as the default authentication package since Windows 2000
  – Windows Logon used daily by millions of users.
  – Used in DOCSIS CableModems for device authentication.
  – Used for embedded systems security
• Kerberos has been *hugely* successful
MIT Kerberos: Timeline & Milestones

1983 - MIT Project Athena Started

1988 - Paper on MIT Kerberos at USENIX’88

1993 - IETF RFC1510 published

1999 - Microsoft Windows 2000 uses Kerberos

2000 - CableLabs uses Kerberos for Cable Modems

2007 - Huge adoption of Kerberos by Finance industry, Defense, Cable, etc

2007 - MIT Kerberos Consortium Founded

2013 - MIT KIT expansion
MIT Kerberos in Commercial Products

- Google
  - Enterprise Search Appliance (GSA)
- Cisco:
  - Cisco IOS - Rel. 11.2 +
  - NAC Appliance
  - ASA5000 & VPN3000 series.
- Intel:
  - VPro II Platforms (AMT)
- Red Hat:
  - Enterprise Linux & FreeIPA
- Sun/Oracle:
  - Solaris 8 to 10 and Solaris Nevada
- Yahoo
  - Hadoop infra
- Juniper:
  - Network Admission Control
- SAP R3
- NetApp:
  - Kerberized NFS
- F5 Networks:
  - BIG-IP ADC
- Other Open Source OS:
  - Ubuntu
  - Debian
Kerberos in Browsers: SPNEGO

(1) HTTP GET resource.html

(2) HTTP 401 – Denied: WWW-Authenticate: Negotiate

(3) Kerberos TGS_REQ
Request Service Ticket for
HTTP/mypages.mywebsite.com@mywebsite.com

(4) Kerberos TGS_REP
Service Ticket granted for
HTTP/mypages.mywebsite.com@mywebsite.com

(5) HTTP GET Authorization: Negotiate w/SPNEGO Token

(6) HTTP 200 – OK WWW-Authenticate w/SPNEGO
response + resource.html

Web Server (Kerberized Service)

(e.g. mypages.mywebsite.com)