STORAGE DEVELOPER CONFERENCE
SDC 22 Fremont, CA September 12-15, 2022
BY Developers FOR Developers

Kerberos/Authentication Updates in Samba

Status Update within Samba 4.16/4.17

Stefan Metzmacher <metze@samba.org>

Samba Team / SerNet

2022-09-14

Check for Updates

- Check for an updated version of this presentation here:
- https://samba.org/~metze/presentations/2022/SDC/

(draft)

Topics

- SambaXP 2020
- Security updates
- Testing improvements
- MIT KDC improvements
- Updated Heimdal snapshot
- Pending Heimdal based Fixes
- Future Updates
- How you can reliable change a machine password
- Questions? Feedback!

SambaXP 2020

- Also see my SambaXP 2020 Talk
- https://samba.org/~metze/presentations/2020/SambaXP/
- It explains/shows a lot of details of how Kerberos works

Security updates

- In November 2021 we fixed a lot security problems
  - Mostly related to name based races
- See Andrew's SambaXP 2022 Talk which explains the details

Security updates

- In November 2021 we fixed a lot security problems
  - Mostly related to name based races
- See Andrew's SambaXP 2022 Talk which explains the details

Testing improvements

- In 2020 we introduced python based protocol tests for krb5
  - We're able to generate any possible request PDU
  - and verify all fields of the response PDU of the KDC
  - The initial infrastructure consisted of 3498 lines
  - (including autogenerated asn code)
- Now in 2022 these tests have been expanded a lot
  - We're now at ~ 21k lines!
  - These new tests helped a lot exploring and fixing the security problems
- Catching regressions is important when changing the KDC code
  - The amount of tests should be able to prevent major regressions
  - However there's still a lot of potential for new/additional tests

Testing improvements

- In 2020 we introduced python based protocol tests for krb5
  - We're able to generate any possible request PDU
  - and verify all fields of the response PDU of the KDC
  - The initial infrastructure consisted of 3498 lines
  - (including autogenerated asn code)
- Now in 2022 these tests have been expanded a lot
  - We're now at ~ 21k lines!
  - These new tests helped a lot exploring and fixing the security problems
- Catching regressions is important when changing the KDC code
  - The amount of tests should be able to prevent major regressions
  - However there's still a lot of potential for new/additional tests

Testing improvements

- In 2020 we introduced python based protocol tests for krb5
  - We're able to generate any possible request PDU
  - and verify all fields of the response PDU of the KDC
  - The initial infrastructure consisted of 3498 lines
  - (including autogenerated asn code)
- Now in 2022 these tests have been expanded a lot
  - We're now at ~ 21k lines!
  - These new tests helped a lot exploring and fixing the security problems
- Catching regressions is important when changing the KDC code
  - The amount of tests should be able to prevent major regressions
  - However there's still a lot of potential for new/additional tests

MIT KDC improvements

- The MIT-KDC code for the active directory dc got support for:
  - PKINIT (certificate/smartcard authentication)
  - S4U2Self (enable an application service to obtain a Kerberos service ticket on behalf of a named user)
  - S4U2Proxy (including resource based constrained delegation RBCD)
  - Propagation of Asserted Identity SIDS: S-1-18-1 vs. S-1-18-2
- We still hide the MIT-KDC feature behind '–with-experimental-mit-ad-dc'
  - The Heimdal based KDC is still the preferred choice
  - The new features require MIT krb5 1.20, which got released on 2022-05-26
  - But the python tests give us an overview what's still missing (and it's getting less and less)

MIT KDC improvements

- The MIT-KDC code for the active directory dc got support for:
  - PKINIT (certificate/smartcard authentication)
  - S4U2Self (enable an application service to obtain a Kerberos service ticket on behalf of a named user)
  - S4U2Proxy (including resource based constrained delegation RBCD)
  - Propagation of Asserted Identity SIDS: S-1-18-1 vs. S-1-18-2
- We still hide the MIT-KDC feature behind '–with-experimental-mit-ad-dc'
  - The Heimdal based KDC is still the preferred choice
  - The new features require MIT krb5 1.20, which got released on 2022-05-26
  - But the python tests give us an overview what's still missing (and it's getting less and less)

Updated Heimdal snapshot (Part 1)

- Samba 4.15 had basically the same Heimdal snapshot as 4.0
  - We did the last import from upstream in 2011
  - Only fixed important bugs
- Samba 4.16 imported a fresh Heimdal snapshot
  - We still have custom patches, but rebased
  - We try to create upstream pull requests before we integrate patches
  - But we may not wait for the changes to accepted upstream
- The new Heimdal internal APIs are much more flexible:
  - It's much easier to hook our AD KDC logic into the core code
  - Hopefully we require less custom changes for future features

Updated Heimdal snapshot (Part 1)

- Samba 4.15 had basically the same Heimdal snapshot as 4.0
  - We did the last import from upstream in 2011
  - Only fixed important bugs
- Samba 4.16 imported a fresh Heimdal snapshot
  - We still have custom patches, but rebased
  - We try to create upstream pull requests before we integrate patches
  - But we may not wait for the changes to accepted upstream
- The new Heimdal internal APIs are much more flexible:
  - It's much easier to hook our AD KDC logic into the core code
  - Hopefully we require less custom changes for future features

Updated Heimdal snapshot (Part 1)

- Samba 4.15 had basically the same Heimdal snapshot as 4.0
  - We did the last import from upstream in 2011
  - Only fixed important bugs
- Samba 4.16 imported a fresh Heimdal snapshot
  - We still have custom patches, but rebased
  - We try to create upstream pull requests before we integrate patches
  - But we may not wait for the changes to accepted upstream
- The new Heimdal internal APIs are much more flexible:
  - It's much easier to hook our AD KDC logic into the core code
  - Hopefully we require less custom changes for future features

Updated Heimdal snapshot (Part 2)

- Support for Kerberos FAST was added:
  - This brings Kerberos request armoring
  - It can tunnel ticket requests and replies that might be encrypted with a weak password inside a wrapper built with a stronger password, say from a machine account.
  - We don't support Compound Identity with FAST yet
- FAST is used by Heimdal and MIT by default if possible
  - But not for Authetication Ticket requests (AS-REQ/REP)
    - Pre-Authentication with weak passwords is not protected
  - Only for Service-Tickets requests (TGS-REQ/REP)
- Windows clients do not use FAST by default
  - Windows (at least) 2012 DCs, as well as explicit GPO settings, are required
  - We announce ourself only as Windows 2008R2

Updated Heimdal snapshot (Part 2)

- Support for Kerberos FAST was added:
  - This brings Kerberos request armoring
  - It can tunnel ticket requests and replies that might be encrypted with a weak password inside a wrapper built with a stronger password, say from a machine account.
  - We don't support Compound Identity with FAST yet
- FAST is used by Heimdal and MIT by default if possible
  - But not for Authetication Ticket requests (AS-REQ/REP)
    - Pre-Authentication with weak passwords is not protected
  - Only for Service-Tickets requests (TGS-REQ/REP)
- Windows clients do not use FAST by default
  - Windows (at least) 2012 DCs, as well as explicit GPO settings, are required
  - We announce ourself only as Windows 2008R2

Updated Heimdal snapshot (Part 2)

- Support for Kerberos FAST was added:
  - This brings Kerberos request armoring
  - It can tunnel ticket requests and replies that might be encrypted with a weak password inside a wrapper built with a stronger password, say from a machine account.
  - We don't support Compound Identity with FAST yet
- FAST is used by Heimdal and MIT by default if possible
  - But not for Authetication Ticket requests (AS-REQ/REP)
    - Pre-Authentication with weak passwords is not protected
  - Only for Service-Tickets requests (TGS-REQ/REP)
- Windows clients do not use FAST by default
  - Windows (at least) 2012 DCs, as well as explicit GPO settings, are required
  - We announce ourself only as Windows 2008R2

Pending Heimdal based Fixes (Part 1)

- Usage of previous passwords should not update badPwdCount
  - It happens when working on multiple hosts with cached passwords
  - It's already fixed for NTLM authentication
  - But Kerberos Pre-Authentication results in ACCOUNT_LOCKED_OUT
  - https://bugzilla.samba.org/show_bug.cgi?id=14054
  - This merge request has fixes for the problem
  - https://gitlab.com/samba-team/samba/-/merge_requests/664
- There are important S4U2Proxy fixes for Windows consumers
  - We need to use the correct decryption key for enc-authorization-data
  - https://bugzilla.samba.org/show_bug.cgi?id=13131
  - We need to use the correct authtime for the PAC
  - https://bugzilla.samba.org/show_bug.cgi?id=13137
  - This merge request has fixes for the problem
  - https://gitlab.com/samba-team/samba/-/merge_requests/2458

Pending Heimdal based Fixes (Part 1)

- Usage of previous passwords should not update badPwdCount
  - It happens when working on multiple hosts with cached passwords
  - It's already fixed for NTLM authentication
  - But Kerberos Pre-Authentication results in ACCOUNT_LOCKED_OUT
  - https://bugzilla.samba.org/show_bug.cgi?id=14054
  - This merge request has fixes for the problem
  - https://gitlab.com/samba-team/samba/-/merge_requests/664
- There are important S4U2Proxy fixes for Windows consumers
  - We need to use the correct decryption key for enc-authorization-data
  - https://bugzilla.samba.org/show_bug.cgi?id=13131
  - We need to use the correct authtime for the PAC
  - https://bugzilla.samba.org/show_bug.cgi?id=13137
  - This merge request has fixes for the problem
  - https://gitlab.com/samba-team/samba/-/merge_requests/2458

Pending Heimdal based Fixes (Part 2)

- We should announce PA-SUPPORTED-ETYPES like windows:
  - We should announce strong encryption types, even if no related key is stored
  - It means a ticket can have a stronger session key type than decryption key type
  - https://bugzilla.samba.org/show_bug.cgi?id=13135
  - This merge request has fixes for the problem
  - https://gitlab.com/samba-team/samba/-/merge_requests/2459

Future Changes

- Compound Identity Support together with Claims Support
  - The new Heimdal KDC APIs will make it easy to add new PAC buffers
  - It's also easy to check with PA-Data elements are used by the client
- Given the client support for FAST in Heimdal and MIT
  - winbindd could be changed to use armoring krb5 auth for pam_winbind
  - It would prevent krb5 pre-auth with weak passwords on the wire

Future Changes

- Compound Identity Support together with Claims Support
  - The new Heimdal KDC APIs will make it easy to add new PAC buffers
  - It's also easy to check with PA-Data elements are used by the client
- Given the client support for FAST in Heimdal and MIT
  - winbindd could be changed to use armoring krb5 auth for pam_winbind
  - It would prevent krb5 pre-auth with weak passwords on the wire

How you can reliable change a machine password (Part 1)

- Windows passwords are UTF-16 with up to 255 characters
  - From there the UTF-8 version is calculated for Kerberos
  - It's also the input for MD4() in order to generate the NTHASH
  - Machine passwords should be as strong as possible
- First we tried completely random passwords:
  - The length is random between 128 and 255 characters
  - Each character is a random 32-bit codepoints
  - =>https://bugzilla.samba.org/show_bug.cgi?id=12262
    - After a password change Kerberos may no longer works
    - The conversation of passwords was wrong depending on 'unix charset'
    - As Heimdal/MIT libraries don't support compound UTF-16
- Then we limited the characters to 16-bit codepoints
  - This avoids compound UTF-16 characters
  - We also verify all conversations and may fallback to ascii for invalid characters

How you can reliable change a machine password (Part 1)

- Windows passwords are UTF-16 with up to 255 characters
  - From there the UTF-8 version is calculated for Kerberos
  - It's also the input for MD4() in order to generate the NTHASH
  - Machine passwords should be as strong as possible
- First we tried completely random passwords:
  - The length is random between 128 and 255 characters
  - Each character is a random 32-bit codepoints
  - =>https://bugzilla.samba.org/show_bug.cgi?id=12262
    - After a password change Kerberos may no longer works
    - The conversation of passwords was wrong depending on 'unix charset'
    - As Heimdal/MIT libraries don't support compound UTF-16
- Then we limited the characters to 16-bit codepoints
  - This avoids compound UTF-16 characters
  - We also verify all conversations and may fallback to ascii for invalid characters

How you can reliable change a machine password (Part 1)

- Windows passwords are UTF-16 with up to 255 characters
  - From there the UTF-8 version is calculated for Kerberos
  - It's also the input for MD4() in order to generate the NTHASH
  - Machine passwords should be as strong as possible
- First we tried completely random passwords:
  - The length is random between 128 and 255 characters
  - Each character is a random 32-bit codepoints
  - =>https://bugzilla.samba.org/show_bug.cgi?id=12262
    - After a password change Kerberos may no longer works
    - The conversation of passwords was wrong depending on 'unix charset'
    - As Heimdal/MIT libraries don't support compound UTF-16
- Then we limited the characters to 16-bit codepoints
  - This avoids compound UTF-16 characters
  - We also verify all conversations and may fallback to ascii for invalid characters

How you can reliable change a machine password (Part 2)

- We tried to store the new password locally first
  - In the past we had problems with ctdb failing to store the password after the remote change
  - =>https://bugzilla.samba.org/show_bug.cgi?id=12782
    - There are DCs with RefusePasswordChange=1 returning WRONG_PASSWORD
    - That way we destroyed the join
- We now store 3 or 4 password generations
  - older, old, current and optionally next
  - Before trying a remote change we store the 'next_change' password (if not already existing)
  - Then we check which password the remote DC currently holds
    - If the server already knows about the 'next_change', we just finish the pending change.
    - If the server only knows about our old or older password, we abort the change, in hope replication latency will fix things up later
  - We try the remote change and store the result

How you can reliable change a machine password (Part 2)

- We tried to store the new password locally first
  - In the past we had problems with ctdb failing to store the password after the remote change
  - =>https://bugzilla.samba.org/show_bug.cgi?id=12782
    - There are DCs with RefusePasswordChange=1 returning WRONG_PASSWORD
    - That way we destroyed the join
- We now store 3 or 4 password generations
  - older, old, current and optionally next
  - Before trying a remote change we store the 'next_change' password (if not already existing)
  - Then we check which password the remote DC currently holds
    - If the server already knows about the 'next_change', we just finish the pending change
    - If the server only knows about our old or older password, we abort the change, in hope replication latency will fix things up later.
  - We try the remote change and store the result

How you can reliable change a machine password (Part 2)

- We tried to store the new password locally first
  - In the past we had problems with ctdb failing to store the password after the remote change
  - =>https://bugzilla.samba.org/show_bug.cgi?id=12782
    - There are DCs with RefusePasswordChange=1 returning WRONG_PASSWORD
    - That way we destroyed the join
- We now store 3 or 4 password generations
  - older, old, current and optionally next
  - Before trying a remote change we store the 'next_change' password (if not already existing)
  - Then we check which password the remote DC currently holds
    - If the server already knows about the 'next_change', we just finish the pending change.
    - If the server only knows about our old or older password, we abort the change, in hope replication latency will fix things up later.
  - We try the remote change and store the result

How you can reliable change a machine password (Part 2)

- We tried to store the new password locally first
  - In the past we had problems with ctdb failing to store the password after the remote change
  - =>https://bugzilla.samba.org/show_bug.cgi?id=12782
    - There are DCs with RefusePasswordChange=1 returning WRONG_PASSWORD
    - That way we destroyed the join
- We now store 3 or 4 password generations
  - older, old, current and optionally next
  - Before trying a remote change we store the 'next_change' password (if not already existing)
  - Then we check which password the remote DC currently holds
    - If the server already knows about the 'next_change', we just finish the pending change.
    - If the server only knows about our old or older password, we abort the change, in hope replication latency will fix things up later.
  - We try the remote change and store the result

How you can reliable change a machine password (Part 2)

- We tried to store the new password locally first
  - In the past we had problems with ctdb failing to store the password after the remote change
  - =>https://bugzilla.samba.org/show_bug.cgi?id=12782
    - There are DCs with RefusePasswordChange=1 returning WRONG_PASSWORD
    - That way we destroyed the join
- We now store 3 or 4 password generations
  - older, old, current and optionally next
  - Before trying a remote change we store the 'next_change' password (if not already existing)
  - Then we check which password the remote DC currently holds
    - If the server already knows about the 'next_change', we just finish the pending change.
    - If the server only knows about our old or older password, we abort the change, in hope replication latency will fix things up later.
  - We try the remote change and store the result

How you can reliable change a machine password (Part 3)

- Even with only validated 16-bit codepoint passwords we are not safe
  - =>https://bugzilla.samba.org/show_bug.cgi?id=14984
    - Changing the password via an RODC we likely destroy the join
    - RODC/RWDC PasswordUpdateForward handling via NetrLogonSendToSam ignores errors
    - Passwords longer than ~ 127 characters get INVALID_PARAMETER, most likely 256 bytes vs. 256 (UTF-16) characters
- We now finally match Windows
  - We're using a fixed length of 120 characters
  - It means password changes work against RODCs now
- It is so important to match Windows as close as possible
  - This is just one example
  - But we had a lot of similar cases in the last 20 years
  - It's really important otherwise we're constantly hitting untested code
    - In Windows itself
    - Other vendors are only testing against Windows

How you can reliable change a machine password (Part 3)

- Even with only validated 16-bit codepoint passwords we are not safe
  - =>https://bugzilla.samba.org/show_bug.cgi?id=14984
    - Changing the password via an RODC we likely destroy the join
    - RODC/RWDC PasswordUpdateForward handling via NetrLogonSendToSam ignores errors
    - Passwords longer than ~ 127 characters get INVALID_PARAMETER, most likely 256 bytes vs. 256 (UTF-16) characters
- We now finally match Windows
  - We're using a fixed length of 120 characters
  - It means password changes work against RODCs now
- It is so important to match Windows as close as possible
  - This is just one example
  - But we had a lot of similar cases in the last 20 years
  - It's really important otherwise we're constantly hitting untested code
    - In Windows itself
    - Other vendors are only testing against Windows

How you can reliable change a machine password (Part 3)

- Even with only validated 16-bit codepoint passwords we are not safe
  - =>https://bugzilla.samba.org/show_bug.cgi?id=14984
    - Changing the password via an RODC we likely destroy the join
    - RODC/RWDC PasswordUpdateForward handling via NetrLogonSendToSam ignores errors
    - Passwords longer than ~ 127 characters get INVALID_PARAMETER, most likely 256 bytes vs. 256 (UTF-16) characters
- We now finally match Windows
  - We're using a fixed length of 120 characters
  - It means password changes work against RODCs now
- It is so important to match Windows as close as possible
  - This is just one example
  - But we had a lot of similar cases in the last 20 years
  - It's really important otherwise we're constantly hitting untested code
    - In Windows itself
    - Other vendors are only testing against Windows

Questions? Feedback!

- Stefan Metzmacher, metze@samba.org
- https://www.sernet.com
- https://samba.plus

Slides: https://samba.org/~metze/presentations/2022/SDC/