Secure HTTP Usage

How hard can it be?

Akademy 2019

Volker Krause

vkrause@kde.org
@VolkerKrause

Transport Security

- Each network connection to a remote server needs to be encrypted
- TLS (formerly known as SSL)
  - Using “https:” URLs
  - Enabling TLS on non-HTTP sockets
- What could possibly go wrong?

Forgetting TLS

- Common on secondary features or docs
  - Links to websites
  - Download of additional content
  - Content sharing
- D19996: Static checking tool as ECM test
- tcpconnect/Wireshark/etc for runtime tests

But a good server redirects to https!

- Are we following that redirect correctly?
- Would we also follow a redirect to http?
- Are we considering HTTP Strict Transport Security (HSTS)?

QNetworkAccessManager

- Redirects off by default (!)
- QNetworkRequest::NoLessSafeRedirectPolicy
- HSTS is off by default (!)
- HSTS state is not persisted by default (!)
- Cookie persistence left to application

KIO

- Redirects work by default
- Follows redirects from https to http (!)
- HSTS is not supported at all (!!)
- Cookie persistence on by default (!)

TLS Errors

- Many ways this can go wrong:
  - Unknown or self-signed certificate
  - Too weak encryption
  - Capture portals
- Only allow override for user-defined hosts, hard fail on internal REST API calls etc

KIO::SslUi

- Allows to show TLS errors to the user and offer a way to (persistently) override
- Choice applied session-wide
- Used by default by KIO
- Does ~~not support QNAM~~ support QNAM since KF 5.62

Testing TLS Errors

- https://badssl.com
- Test servers with all kinds of TLS scenarios
- Focused on browser testing, but useful even for non-HTTP tests
- Demo

Non-HTTP TLS

- QSslSocket
- KTcpSocket
  - Defaults to deprecated/unsecure SSLv3, needs to be explicitly set to current standards
- Both supported by KIO::SslUi

Going Forward

- KNAM: a QNAM with secure defaults
- ~~Extend SSL error handler for QNAM~~
- Rebase DAV support on KNAM
- Rebase KIO HTTP on KNAM
- Phase out KTcpSocket in favor of QSslSocket

Conclusion

- We have all the building blocks, but assembling them correctly is too error prone
- Tools for verifying/testing this could use improvement and more wide-spread use
- Doing this right is important!

Questions?

KF6 BoF: Monday 09:30 U1-04

References

- Demo application: https://invent.kde.org/vkrause/http-demo
- HTTP URL checker: https://invent.kde.org/vkrause/http-check
- tcpconnect: https://github.com/iovisor/bcc
- QNetworkAccessManager: https://doc.qt.io/qt-5/qnetworkaccessmanager.html
- QSslSocket: https://doc.qt.io/qt-5/qsslsocket.html
- KIO: https://api.kde.org/frameworks/kio/html/index.html
- HSTS: https://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security
- TLS error tests: https://badssl.com