Changes in file: qauthenticator.cpp

```
static QByteArray qNtlmPhasel();
static QByteArray qNtlmPhase3(QAthenticatorPrivate *ctx, const QByteArray& phase2data);
fif qT_COMPIG(asp) // SSPI
        static OByteArray (NtlmFhase3(QAuthenticatorPrivate *ctx, const OByteArray6 phase2data);

#if OI_CONFIG(espx) // SSPI
static bool q SSPI_library_load();

#static DByteArray qSspitartup(CAuthenticatorPrivate *ctx, CAuthenticatorPrivate::Hethod
                                                                                                                                                                                                                                                               QStringView host);

=static QByteArray qSspiContinue(QAuthenticatorPrivate *ctx, QA
               | OStringView host, CByteArrayView challenge = ());
| Static Cool qGsaspiTestGetCredentials (OStringView host);
| Static CByteArray qGsaspiStartup(CAuthenticatorFrivate *ctx, OStringView host);
| Static CByteArray qGsaspiStartup(CAuthenticatorFrivate *ctx, OStringView host);
| OSTRING CAUTHER CONTINUE CONTIN
                                   methodString = "NTLM";
if (challenge.isEmpty()) {
CONFIG(sspi) // SSPI
                                                                                                                                                                                                                                                                                        methodString = "NTLM";
if (challenge.isEmpty()) {
CONFIG(sspi) // SSPI
                                             G(smp1) // SSPI
OByteArray phaselToken;
if (user.isEmpty()) ( // Only pull from system if no user was specified in
authenticate.
                                                                                                                                                                                                                                                                                                  OPair<OByteArray, bool> phaselToken;
if (user.isEmpty()) { // Only pull from system if no user was specified in
                                             authenticator

phaselToken = qSspiStartup(this, method, host);
) clso if (!q_SSPI_library_load()) {

// Since we're not running qSspiStartup we have to make sure the library is
                                                                                                                                                                                                                                                                                                  authenticator
phaselToken = q5spiStartup(this, method, host);
) else if (fq.SSF] library load()) (
// Since we're not running d5spiStartup we have to make sure the library is
                                                      qwarning("Failed to load the SSPI libraries");
return "";
                                                                                                                                                                                                                                                                                                           qWarning("Failed to load the SSPI libraries");
return "";
                                                                                                                                                                                                                                                                                               )
if ('phaselToken.first.isEmpty()) {
    response = phaselToken.first.toBase64();
    phase = Phase2;
} else
                                           )
if (!phaselToken.isEmpty()) {
    response = phaselToken.toBase64();
    phase = Phase2;
} else
                                                    response = qNtlmPhasel().toBase64();
if (user.isEmpty())
phase = Done;
else
                 #endif
                                                                                                                                                                                                                                                                - #endif
                                                                                                                                                                                                                                                                                                         response = qNtlmPhasel().toBase64();
if (user.isEmpty())
    phase = Done;
else
                                         else {
FIG(ssp1) // SSFI

OByteArray phase3Token;
if (ssp1WindowsHandles)
phase3Token = dSsp1Continue(this, method, host, OByteArray::fromBase64(
challenge));
if (lphase3Token.isEmpty()) {
    response = phase3Token.toBase64();
    phase = Done;
} else
                                                                                                                                                                                                                                                                                             } else {

#if QT_CONFIG(ssm
               #endif
                                                     response = qNtlmPhase3(this, QByteArray::fromBase64(challenge)).toBase64(); phase = Done;
                                                                                                                                                                                                                                                                                                         response = qNtlmPhase3(this, QByteArray::fromBase64(challenge)).toBase64(); phase = Done;
                                              challenge = "";
                                                                                                                                                                                                                                                                                                  challenge = "";
     phaselToken = qGssapiStartup(this, host);
                                                                                                                                                                                                                                                                                                phaselToken = qGssapiStartup(this, host);
518
519 - #endif
                                            if (!phaselToken.isEmpty()) {
    response = phaselToken.toBase64();
    phase = Phase2;
} else (
    phase = Done;
    return "";
                                                                                                                                                                                                                                                                                               if (!phaselToken.first,isEmpty()) {
    response = phaselToken.first,toBase64();
    phase = Phase2;
} else {
    phase = Done;
    return "";
                                  } else {
    OByreArray phase3Token;
    OByreArray phase3Token;
    ONFIG (seps) // SSST |
    if (seps) // SSST |
    if (seps) // SSST |
    challenge);
    challenge);
    // GSSAFI |
    challenge);
                                                                                                                                                                                                                                                                                       IG(sspi) // SSFI

if (sspi)%indowsHandles)
phaesToken = qSspiContinue(this, method, host, QByteArray::fromBase64(challenge)):
NFIG(gssapi) // GSSAFI
                                                                                                                                                                                                                                                                                                if (gsapiHandles)
phase3Token = qGssapiContinue(this, QByteArray::fromBase64(challenge));
                                                                                                                                                                                                                                                                                               if (!phase3Token.first.isEmpty()) {
    response = phase3Token.first.toBase64();
    if (phase3Token.second) {
        phase = Done;
    }
}
                                             if (!phase3Token.isEmpty()) {
   response = phase3Token.toBase64();
                                                    phase = Done;
                                                                                                                                                                                                                                                                                                         }
challenge = "";
                                             challenge = "";
} else {
                                                                                                                                                                                                                                                                                               challeng
) else {
    if (phase3Token.second) {
        phase = Done;
}
                                                      phase = Done
                                                                                                                                                                                                                                                                           atic OPair(OByteArray, bool) qSspiStartup(OAuthenticatorPrivate *ctx, OAuthenticatorPrivate::Method method, OStringView host)
           QStringView host)
                                                                                                                                                                                                                                                                               QPair<QByteArray, bool> result;
result.first = "";
result.second = false;
                                                                                                                                                                                                                                                                                if (!q_SSPI_library_load())
    return result;
                            if (!q_SSPI_library_load())
    return QByteArray();
                                                                                                                                                                                                                                                                               TimeStamp expiry: // For Windows 9x compatibility of SSPI calls
                           TimeStamp expiry; // For Windows 9x compatibility of SSPI calls
                                                                                                                                                                                                                                                                               if (!ctx->ssniWindowsHandles)
                                             SECENS_CRED_OUISOURD, HMILDER, MSERMEN ? GRACH : HMILDER, HMILDER, HMILDER, GCTX->SSPIWINDOWSHANDLES->CREDHANDLE, GEXPITY
                                                                                                                                                                                                                                                                                               Scorms_ckbr_corbound, muliper, useAuch : muliper, muliper, muliper, fctx->sspiWindowsHandles->credHandle, &expiry
                           );
if (secStatus != SEC_E_OK) (
ctx->sspiWindowsHandles.reset(nullptr);
return QByteArray();
                                                                                                                                                                                                                                                                             );
if (secStatus != SEC_E_OK) (
    ctx->ssp:WindowsHandles.reset(nullptr);
    return result;
                                                                                                                                                                                                                                                                 4
                                                                                                                                                                                                                                                                              return qSspiContinue(ctx, method, host);
                            return qSspiContinue(ctx, method, host);
                                                                                                                                                                                                                                                                #static OPaircOByteArray, bools of spiContinue (ORUTHERLAND)

QAuthenticatorFrivate::Method method,
QStringView host, QByteArrayView challenge)
             ==static QByteArray qSspiContinue(QAuthenticatorPrivate *ctx, QAuthenticatorPrivate::Method method,
                           OByteArray result;
SecBuffer challengeBuf;
SecBuffer responseBuf;
SecBufferBesc challengeBesc;
SecBufferBesc challengeBesc;
SecBufferBesc responseDesc;
unsigned long attrs;
TimeStamp expiry; // For Windows %x compatibility of SSPI calls
                                                                                                                                                                                                                                                                             //ObyteArray result;
SecBuffer challengeBuf;
SecBuffer responseBuf;
SecBufferDeec challengeBus;
SecBufferDeec challengeBus;
SecBufferDeec responseDeac;
unsigned long attrs;
TimeStamp expiry; // For Windows 9x compatibility of SSPI calls
bool isBone = false;
QPair<QByteArray, bool> result;
                           if (!challenge.isEmpty())
                                                                                                                                                                                                                                                                             if (!challenge.isEmpty())
```

| Acceptatus != SEC_I_COMPLETE_AND_CONTINUE_46 secStatus != SEC_I_CONTINUE_NEEDED) {
| pSecurityFunctionTable=>FreeCredentialsHandle(scux->sspiWindowsHandles->credHandle);
| pSecurityFunctionTable=>DeleteSecurityContext(scux->sspiWindowsHandles->ctxHandle);
| pSecurityFunctionTable=>DeleteSecurityContext(scux->sspiWindowsHandles->ctxHandle);
| pSecurityFunctionTable=>DeleteSecurityContext(scux->sspiWindowsHandles->ctxHandle);
| ctx->sspiWindowsHandles.reset(nullptr);
| ctx->sspiWindowsHandles.reset(nullptr);
| tistor_I_continue_NeeDed 64 iscor_I_complete_And_continue_(scux->sspiWindowsHandles->ctxHandle);
| ctx->sspiWindowsHandles.reset(nullptr);
| tistor_I_continue_NeeDed 64 iscor_I_complete_And_continue_(scux->sspiWindowsHandles->ctxHandle);
| tistor_I_continue_NeeDed 64 iscor_I_continue_NeeDed 64 is