Credential Management and Secure Single Login for SPKM

- Client-Server
- Generic Security Service
- Simple Public Key Mechanism
- Credential Management
- Secure Single Login
Credential Management and Secure Single Login for SPKM

The Players

Client

Mr. BadGuy

Server
Credential Management and Secure Single Login for SPKM

GSS_Init_sec_context
(..., Cred_C, S, ...)

GSS_Accept_sec_context
(..., T, Cred_S)

GSS_Init_sec_context
(..., T, ...)

Client

Server
Credential Management and Secure Single Login for SPKM

![Diagram of Kerberos Authentication System]

- **Client**
- **Server**
- **Authentication Server**
- **Ticket Granting Service**

Key Diagram Components:
- TGS-req
- S-req
- Service-req
- S
- TGS

© Detlef Hühnlein, huehnlein@secunet.de
SPKM (3-Way-Auth)

1. Rc, \( M = Rc|S|C \), \( F = \text{Sig}(h(M), Sc) \)

2. \( M, F, \) Pc

3. Verify

4. Rs, Kcs, \( N = Rs|Rc|Kcs \)
   \( G = \text{Enc}(N, Ps), \) \( H = \text{Sig}(h(N), Ss) \)

5. \( G, H, \) Ps

6. Verify, decrypt \( G, I = \text{Enc}(Rs, Ps) \)

7. I

8. \( Rs' = \text{Enc}(I, Ss) = Rs? \)

© Detlef Hühnlein, huehnlein@secunet.de
Credential Management for SPKM

"The key management employed in SPKM is intended to be as compatible as possible with both X.509 and PEM, since these represent large communities of interest and show relative maturity in standards."
Credential Management and Secure Single Login for SPKM

Credential Management

PSE

secunet

Michaela May

or

SW-PSE

X.500

© Detlef Hühnlein, huehnlein@secunet.de
## Credential Management and Secure Single Login for SPKM

### Multiple Connections

- Keep PSE accessible (for a long time)
- Enter PIN to open PSE for every connection
- Secure Single Login

### Credential Management

<table>
<thead>
<tr>
<th></th>
<th>Usability</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keep PSE accessible (for a long time)</td>
<td>😊</td>
<td>😞</td>
</tr>
<tr>
<td>Enter PIN to open PSE for every connection</td>
<td>😞</td>
<td>😊</td>
</tr>
<tr>
<td>Secure Single Login</td>
<td>😊</td>
<td>😊</td>
</tr>
</tbody>
</table>

© Detlef Hühnlein, huehnlein@secunet.de
Secure Single Login

<table>
<thead>
<tr>
<th>Kerberos</th>
<th>SPKM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get TGS-Ticket with limited lifetime to authenticate</td>
<td>Generate and (self) certify Public Key Pair with limited lifetime to authenticate</td>
</tr>
</tbody>
</table>
X.509 v3 - Certificates

- version
- serialNumber
- signature
- issuer
- validity
- subject
- subjectPublicKeyInfo
- ... extensions
- not before
- not after
- signature
- algorithm
X.509 v3 / PKIX - Extensions

- **SubjectAltName**
- **IssuerAltName**
- **Basic Constraints**
  - Boolean: `cA`
  - Integer: `PathLenConstraint`
- **Key Usage**
  - BitString:
    - (0) `digitalSignature`
    - (1) `nonRepudiation`
    - (2) `keyEncipherment`
    - (3) `dataEncipherment`
    - (4) `keyAgreement`
    - (5) `keyCertSign`
    - (6) `cRLSign`
    - (7) `encipherOnly`
    - (8) `decipherOnly`
- **Name Constraints**
  - `GenSubtree` permittedSubtrees
  - `GenSubtree` excludedSubtrees
- **ExtendedKeyUsage**
  - OID
  - KeyPurposeId

Examples:
- `id-kp-serverAuth`
- `id-kp-clientAuth`
- `id-kp-codeSigning`
- `id-kp-emailProtection`
Credential Management and Secure Single Login for SPKM

new Key Purposes:
- id-kp-SignTempCert
- id-kp-Temporary

### permanent

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issuer</td>
<td>CA</td>
</tr>
<tr>
<td>validity</td>
<td>u-notBefore</td>
</tr>
<tr>
<td>subject</td>
<td>User</td>
</tr>
<tr>
<td>subjectAltName</td>
<td>User-alt</td>
</tr>
<tr>
<td>issuerAltName</td>
<td>CA-alt</td>
</tr>
<tr>
<td>Keyusage</td>
<td>critical=TRUE, digitalSignature, nonRepudiation</td>
</tr>
<tr>
<td>ExtKeyUsage</td>
<td>critical=FALSE (id-kp-SignTempCert)</td>
</tr>
<tr>
<td>Basic Constraints</td>
<td>critical=TRUE, cA=FALSE</td>
</tr>
</tbody>
</table>

### temporary

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issuer</td>
<td>User</td>
</tr>
<tr>
<td>validity</td>
<td>t-notBefore</td>
</tr>
<tr>
<td>subject</td>
<td>User</td>
</tr>
<tr>
<td>subjectAltName</td>
<td>User-alt</td>
</tr>
<tr>
<td>issuerAltName</td>
<td>User-alt</td>
</tr>
<tr>
<td>Keyusage</td>
<td>critical=TRUE, digitalSignature</td>
</tr>
<tr>
<td>ExtKeyUsage</td>
<td>critical=TRUE, id-kp-Temporary</td>
</tr>
<tr>
<td>Basic Constraints</td>
<td>critical=TRUE, cA=FALSE</td>
</tr>
</tbody>
</table>
Credential Management and Secure Single Login for SPKM

Verification Procedure

1. PKIX conform
   - Issuer = subject
   - issuerAlt = subjectAlt
   - validity.T-notBefore > validity.U-notBefore
   - validity.T-notAfter < validity.U-notAfter
   - KeyUsage critical = TRUE
digitalSignature = TRUE
   - ExtKeyUsage (id-kp-SignTempCert)

   not present

2. PKIX conform

   KeyUsage critical = TRUE
digitalSignature = TRUE
   ExtKeyUsage (id-kp-SignTempCert)

   Issuer = subject
   issuerAlt = subjectAlt
   validity.T-notBefore > validity.U-notBefore
   validity.T-notAfter < validity.U-notAfter
   KeyUsage critical = TRUE
   nonRepudiation = FALSE
   keyCertSign = FALSE
   cRLSign = FALSE
   ExtKeyUsage critical = TRUE
   id-kp-Temporary is present
   Basic Constraints critical = TRUE
cA = FALSE
### Efficiency (Estimate)

<table>
<thead>
<tr>
<th>Security</th>
<th>Usab.</th>
<th>Time Efficiency (1024 Bit Mult.)</th>
<th>Space Efficiency (Byte)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Once</td>
<td>Session</td>
</tr>
<tr>
<td>Single Login</td>
<td>😞😊</td>
<td>/</td>
<td>/</td>
</tr>
<tr>
<td>Multiple Login</td>
<td>😊😞</td>
<td>/</td>
<td>/</td>
</tr>
<tr>
<td>SSLLogin - RSA</td>
<td>😊😊</td>
<td>/</td>
<td>108315</td>
</tr>
<tr>
<td>SSLLogin - DL (naive)</td>
<td>😊😊</td>
<td>116000</td>
<td>1267</td>
</tr>
<tr>
<td>SSLLogin - DL (prec.)</td>
<td>😊😊</td>
<td>116517</td>
<td>675</td>
</tr>
</tbody>
</table>
http://www.ietf.org/internet-drafts/draft-huehnlein-credman-spkm-00.txt

By Hans Schupp, GMD, Darmstadt, Germany

& me