Seminar: Information and System Security

Opportunistic protected login: Next step in traditional password based user authentication

Rakesh Pandit
MSc student, rakesh.pandit@cs.helsinki.fi
Overview

- Problem?
- Protected login
- Opportunistic protected login
- TLS – origin bound certificates
Content

- Goals and Threat model
- Simple authentication
- TLS client authentication
- TLS OBC authentication
- Protected first login
- Framework evaluation claims
- Open problems
- Conclusion
• Goals and Threat model
  • Simple authentication
  • TLS client authentication
  • TLS OBC authentication
  • Protected first login
  • Framework evaluation claims
  • Open problems
  • Conclusion
Goals

- Public key cryptography
- No change – user experience
- Fallback – opportunistic
- Reduce unprotected logins
Threat model (includes)

- Man in the middle attack
- Phishing attacks (steal username/password)
Proposal

Server

TLS-OBC

Client (Browser)

User

PROTECTED LOGIN

Mobile device

Secure first login
• Goals and Threat model
• **Simple authentication**
• TLS client authentication
• TLS OBC authentication
• Protected first login
• Framework evaluation claims
• Open problems
• Conclusion
Simple authentication
Simple authentication

1. Handshake
Simple authentication

1. Handshake
2. Server -> Client (certificate)

Server

Cert-S

ACCEPT

Client
Simple authentication

1. Handshake
2. Server -> Client (certificate)
3. TLS connection
Simple authentication

1. Handshake
2. Server -> Client (certificate)
3. TLS connection
4. Client -> Server (password)

Username/password
Simple authentication

1. Handshake
2. Server -> Client (certificate)
3. TLS connection
4. Client -> Server (password)
5. Server -> Client (Cookie)
Simple authentication

1. Handshake
2. Server -> Client (certificate)
3. TLS connection
4. Client -> Server (password)
5. Server -> Client (Cookie)

Cookie = username/password
TLS MITM Attack

1. Handshake
1. Handshake

a. Client -> Attacker Hello,

b. Attacker -> Server Hello,

c. Server -> Attacker Hello,

d. Attacker -> Client Hello,
TLS MITM Attack

1. Handshake
2. Server -> Client (certificate)
TLS MITM Attack

1. Handshake
2. Server -> Attacker (certificate-S)
3. Attacker -> Client (certificate-A)
TLS MITM Attack

1. Handshake
2. Server -> Attacker (certificate-S)
3. Attacker -> Client (certificate-A)
4. TLS connection
TLS MITM Attack

1. Handshake
2. Server -> Attacker (certificate-S)
3. Attacker -> Client (certificate-A)
4. TLS connection
5. Client -> Attacker (username/pass)
6. Attacker-> Server (username/pass)
TLS MITM Attack

1. Handshake
2. Server -> Attacker (certificate-S)
3. Attacker -> Client (certificate-A)
4. TLS connection
5. Client -> Attacker (username/pass)
6. Attacker-> Server (username/pass)
7. Server -> Attacker (Cookie)
8. Attacker -> Client (Cookie)
TLS MITM Attack

Connection established
• Goals and Threat model
• Simple authentication
• TLS client authentication
• TLS OBC authentication
• Protected first login
• Framework evaluation claims
• Open problems
• Conclusion
TLS Client Authentication

Cert-C

Server

Client
TLS Client Authentication

Server

Cert-C

Client

Cert-C
TLS Client Authentication
MITM Attack – TLS Client Authentication

1. Client -> Attacker (Cert-C)
MITM Attack – TLS Client Authentication

1. Client -> Attacker (Cert-C)
2. Attacker -> Server (Cert-A)
MITM Attack – TLS Client Authentication

1. Client -> Attacker (Cert-C)
2. Attacker -> Server (Cert-A)
3. Cert-A != Cert-C (mismatch)
• Goals and Threat model
• Simple authentication
• TLS client authentication
• TLS OBC authentication
• Protected first login
• Framework evaluation claims
• Open problems
• Conclusion
TLS-OBC authentication

1. Client -> Server (OB-certificate)
TLS-OBC authentication

1. Client -> Server (OB-certificate)
2. Server -> Client (certificate)
TLS-OBC authentication

1. Client -> Server (OB-certificate)
2. Server -> Client (certificate)
3. TLS connection
TLS-OBC authentication

1. Client -> Server (OB-certificate)
2. Server -> Client (certificate)
3. TLS connection
4. Client -> Server (password)
TLS-OBC authentication

1. Client -> Server (OB-certificate)
2. Server -> Client (certificate)
3. TLS connection
4. Client -> Server (password)
5. Server -> Client (Cookie)
TLS-OBC authentication

Cookie = <value, HMAC\(_K\)(value+fingerprint)>

Value = old cookie
K = server secret key
Fingerprint = fingerprint of OB certificate
TLS-OBC authentication
TLS-OBC MITM attack

1. Client -> Attacker (OB-certificate)
TLS-OBC MITM attack

1. Client -> Attacker (OB-certificate)
2. Attacker -> Server (OB-certificate-A)
TLS-OBC MITM attack

1. Client -> Attacker (OB-certificate)
2. Attacker -> Server (OB-certificate-A)
3. Server -> Attacker (Cert-S)
TLS-OBC MITM attack

1. Client -> Attacker (OB-certificate)
2. Attacker -> Server (OB-certificate-A)
3. Server -> Attacker (Cert-S)
4. Attacker -> Client (Cert-A)
TLS-OBC MITM attack

1. Client -> Attacker (OB-certificate)
2. Attacker -> Server (OB-certificate-A)
3. Server -> Attacker (Cert-S)
4. Attacker -> Client (Cert-A)
5. TLS connection
TLS-OBC MITM attack

1. Client -> Attacker (OB-certificate)
2. Attacker -> Server (OB-certificate-A)
3. Server -> Attacker (Cert-S)
4. Attacker -> Client (Cert-A)
5. TLS connection
6. Client -> Attacker (Cookie)
7. Attacker -> Server (Cookie)
1. Client -> Attacker (OB-certificate)
2. Attacker -> Server (OB-certificate-A)
3. Server -> Attacker (Cert-S)
4. Attacker -> Client (Cert-A)
5. TLS connection
6. Client -> Attacker (Cookie)
7. Attacker -> Server (Cookie)
TLS-OBC MITM attack

Server

Attacker

TLS OBC mismatch!

Client

Cookie
TLS OBC advantages

- Prevent MITM attack
- Don't change user experience
- First login still an issue
• Goals and Threat model
• Simple authentication
• TLS client authentication
• TLS OBC authentication
• Protected first login
• Framework evaluation claims
• Open problems
• Conclusion
Protected first login

Server

Client

Mobile phone
Protected first login

1. username/password
Protected first login

1. username/password
2. Client -> server (user/pass) and login ticket
1. username/password
2. Client -> server (user/pass) and login ticket
3. login ticket and id assertion
1. username/password
2. Client -> server (user/pass) and login ticket
3. login ticket and id assertion
4. id assertion and cookie
Protected first login
Protected first login

Client → Username/password → Server
Protected first login

Key@device_address.device_type.certauth

Client

Username/password

Login ticket + certauthid

Server

TLS Channel ID
Origin
TLS OBC Support
Expiration time
Key etc
(signed and encrypted with master key)
Protected first login

Mobile phone

Client
Protected first login

Login ticket
(signed and encrypted with master key)

Metadata=origin, TLS channel ID etc

Assertion request
Protected first login

- Mobile phone
- Client

Login ticket
(signed and encrypted with master key)

Metadata=origin,
TLS channel ID etc

Assertion request

Id assertion
Protected first login (again)

1. username/password
2. Client -> server (user/pass) and login ticket
3. login ticket and id assertion
4. id assertion and cookie
• Goals and Threat model
• Simple authentication
• TLS client authentication
• TLS OBC authentication
• Protected first login
• Framework evaluation claims
• Open problems
• Conclusion
Framework evaluation claims

- Quasi-nothing-to-carry-benefit
- Recovery-from-loss benefits
• Goals and Threat model
• Simple authentication
• TLS client authentication
• TLS OBC authentication
• Protected first login
• Framework evaluation claims
• Open problems
• Conclusion
Open problems

- Bluetooth – distance vs delay
- TLS-OBC proposal still in flux
- Tablets/smartphones usage
Attacks (not in threat model!)
Conclusion

• No user interaction?
• Open problems related to bluetooth
• Promising improvement
Thank you