Man-in-the-Middle-Angriff auf SSH2

- Studienarbeit von David Gümbel und Markus Geyer

Übersicht

- Asymmetrische Kryptoverfahren
  - RSA
  - Diffie-Hellman und DSS
- Man-in-the-Middle-Angriff
  - Prinzip
  - Real Life: dsniff, ettercap & Co.
- SSH-Protokoll
  - Architektur
  - Implementierung in J2SSH
  - Angriff mit jmitm2

Asymmetrische Kryptoverfahren

- Zwei Schlüssel
  - Öffentlich: nur Verschlüsselung und Signaturüberprüfung
  - Privat: nur Entschlüsselung und Signierung
- Probleme
  - Authentizität
  - Performanz
- Hybride Verschlüsselung

RSA

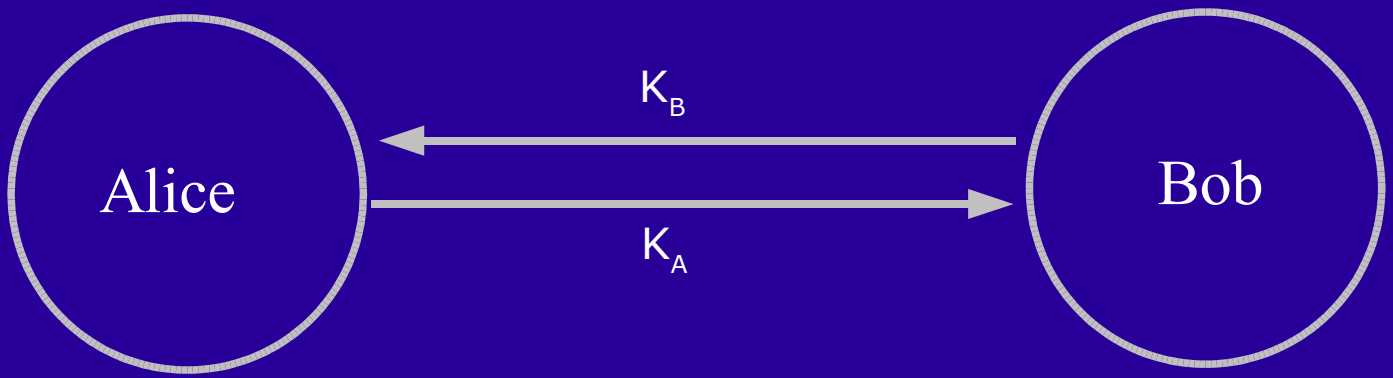
- 1977: Rivest, Shamir, Adleman
- Verschlüsselung und Signatur
- Basierend auf Faktorisierungsproblematik

Diffie-Hellman und DSS

- 1976: Diffe, Hellman
- reines Schlüsselaustauschverfahren
- basierend auf Diskreter-Logarithmus-Problem
- Session-Keys
- darauf aufbauend: ElGamal, DSS

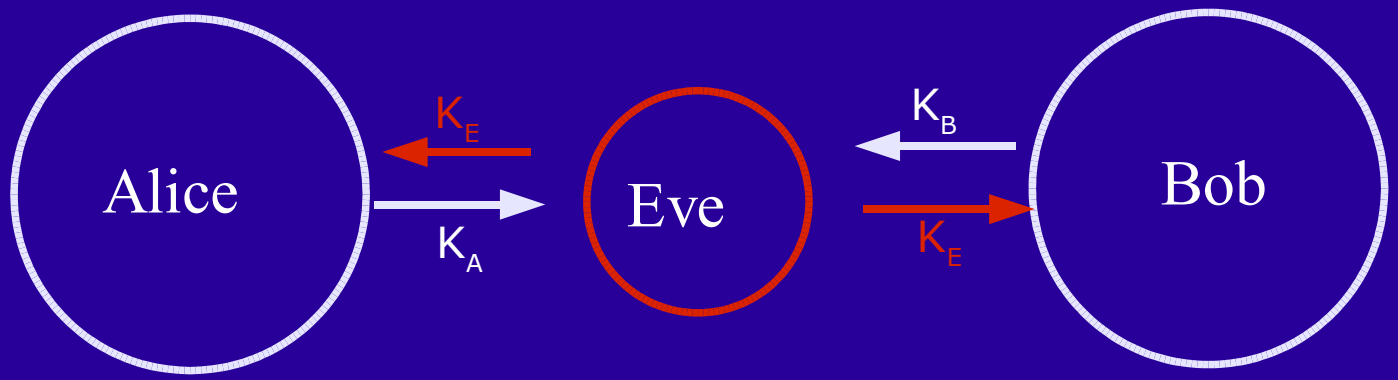
Man-in-the-Middle-Angriff

- Alice und Bob wollen kommunizieren
- daher Schlüsselaustausch i.d.R. über unsicheres Medium



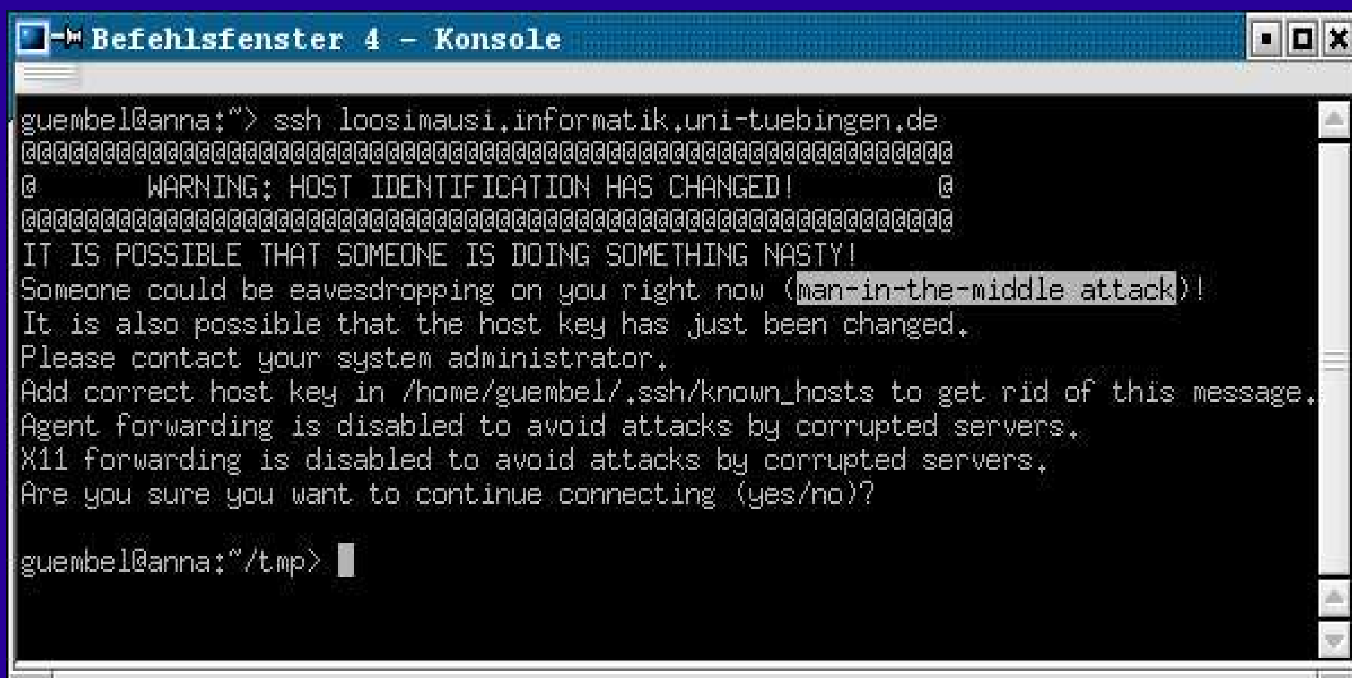
Man-in-the-Middle-Angriff

- Angreiferin Eve hat Kontrolle über den Nachrichtenkanal
- Eve sendet eigenen Public Key als fremden an Alice bzw. Bob



dsniff, ettercap & Co.

- Praktisch leicht durchführbar
- Für übliche Protokolle wie SSH1 und SSL: Toolsammlung dsniff automatisiert Angriff



SSH

- Secure Shell Protocol
- Features
  - Sicherer Datenkanal
  - Authentifizierung (Paßwort, Public Key, ...)
  - Integrität sichergestellt
  - remote shell, remote command execution
  - Port-Forwarding
  - X11-Forwarding

SSH – Historie

- 1995: T. Ylönen – SSH1
  - basiert auf RSA, DES
  - Designschwächen
  - angreifbar durch dsniff, ettercap, ...
- 1998 erste Implementierung SSH2
  - Designschwächen beseitigt
  - basiert auf Diffie-Hellman, DSA, AES
- weit verbreitet
- SSH.com, OpenSSH, J2SSH, ...

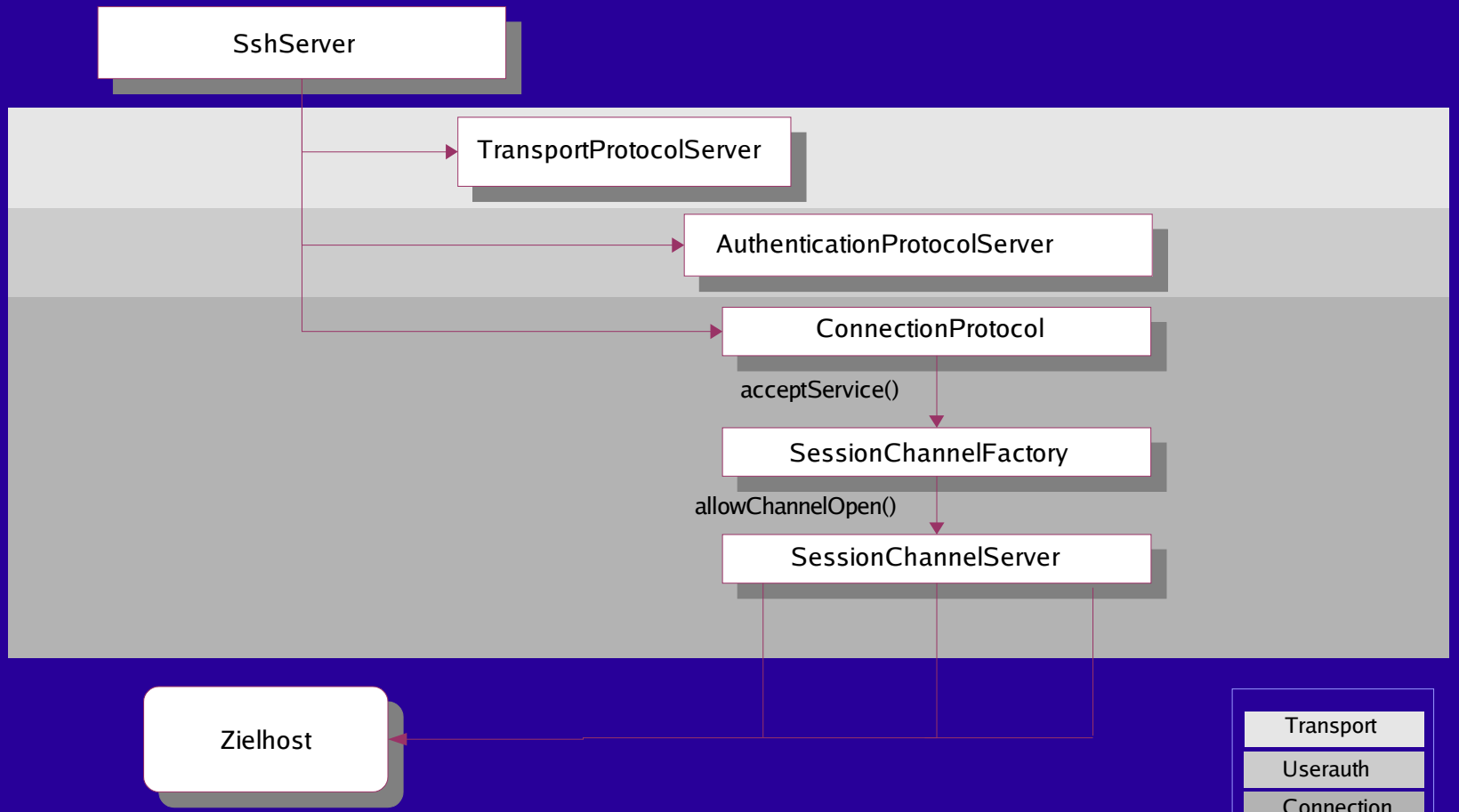
SSH2 – Architektur

- Drei Schichten
  - Transport Layer
    - verschlüsselter Datenkanal Client<->Server
    - Server-Authentifizierung per Server-Public-Key
  - User Authentication Layer
    - Benutzer-Authentifizierung per Paßwort, Public Key, Smartcard, ...
    - aufbauend auf Transport Layer
  - Connection Layer
    - beinhaltet „Sessions“
    - bietet X11- und Port-Forwarding, Shell etc.

SSH2 – Implementierung in J2SSH

- vollständiges SSH2-Protokoll (Client und Server)
- Java-basiert
- 256 Klassen
- Open Source (LGPL)
- http://www.sshtools.com

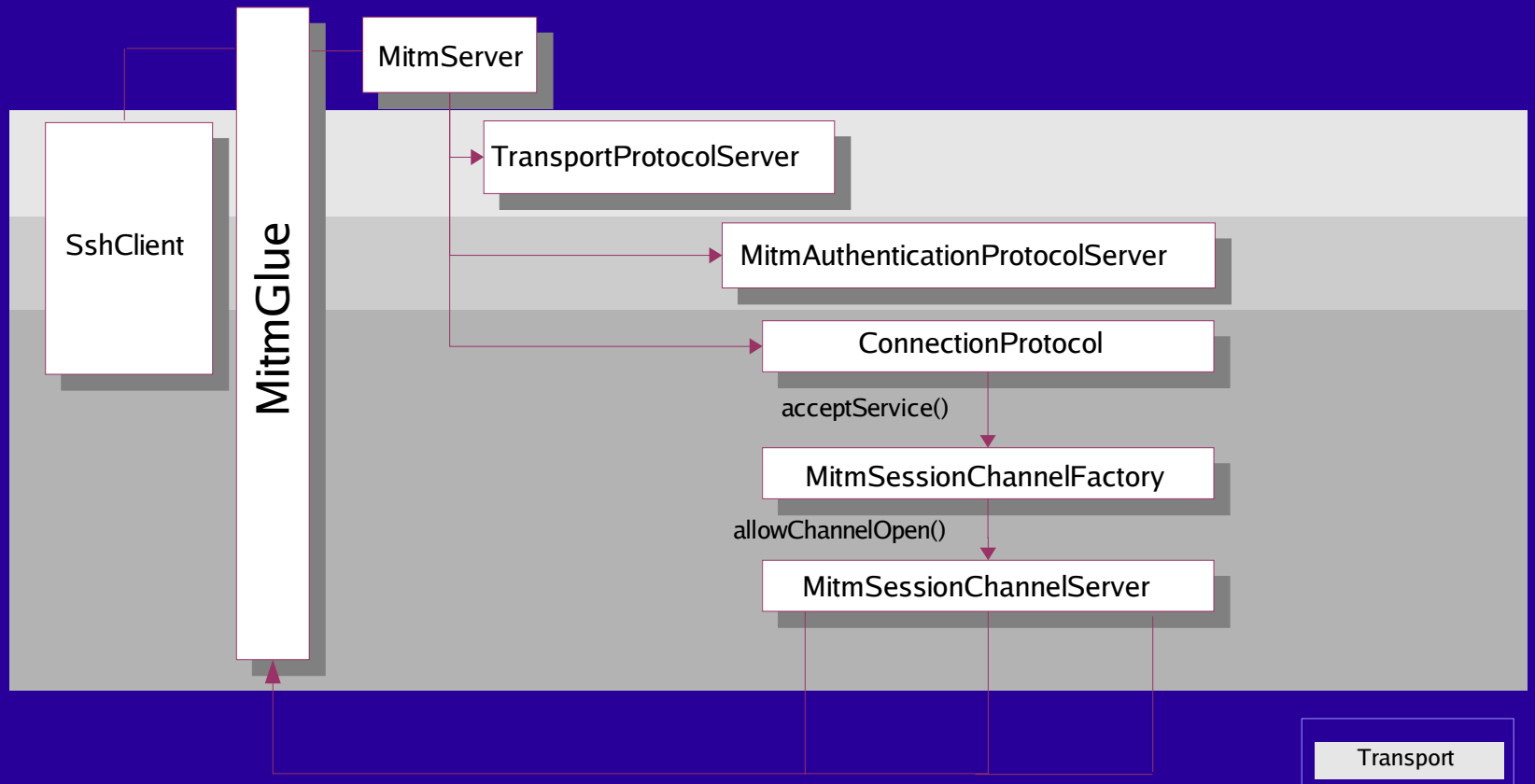
SSH2 – Implementierung in J2SSH



SSH2 – Angriff mit jmitm2

- dsniff
  - Layer 3 umleiten (ARP)
  - (DNS-Spoofing)
- jmitm2
  - eingehende SSH-Verbindungen abfangen
  - eigene SSH-Verbindungen aufbauen
  - Benutzerdaten, Ein- und Ausgaben durchreichen
  - Logging von Paßwörtern

jmitm2 – Architektur



jmitm2 – Features

- objektorientiert, basierend auf J2SSH
- plattformunabhängig
- XML-basierte Konfiguration
- mehrere Angriffe gleichzeitig möglich
- flexibles, ausführliches Logging (via Log4J)
- leicht erweiterbar
  - Session übernehmen
  - command injection
  - Ausgaben fälschen
  - Sitzungs-Logging

So what?

- immer brav Keys überprüfen :-)
- bereits für Schulungszwecke genutzt (ASAP-COM GmbH)
- Source, JavaDoc:
  - http://www.david-guembel.de/jmitm2.html
- Studienarbeit:
  - Bibliothek