Kurs 1866

KE 1: Sicherheit in der Informationstechnik

Angriffsziele / Schutzziele

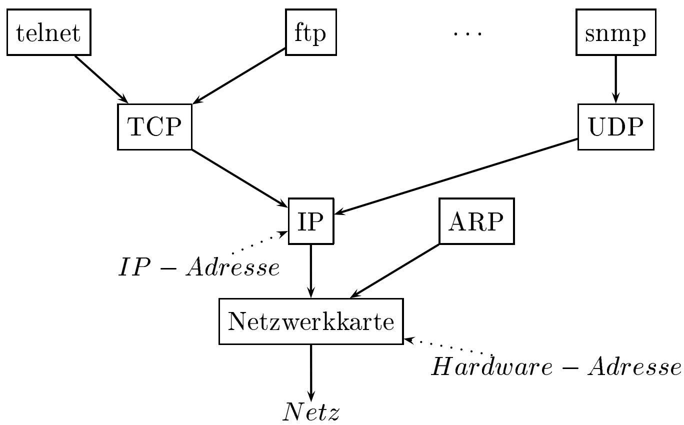
- Vertraulichkeit: Daten sind nur für befugte Personen zugänglich.
- Integrität: Daten sind korrekt und unveränderlich.
- Authentizität: Daten stammen vom vorgeblichen Erzeuger.
- Verfügbarkeit: Daten können von befugten Personen gelesen / bearbeitet werden.
- Anonymität

Netze

- Sterntopologie
- Ringtopologie
- Bustopologie

Ethernet ist ein Busnetz, dessen Sicherheitslücken durch sniffing in lokalen Netzen durch Switche begrenzt werden können.

Internetprotokoll



Konkrete Gefahren

- Schadensprogramme
  - Viren
  - Trojaner
  - Würmer
- Passwortmissbrauch
  - Hacking
  - Falsche Athentisierung

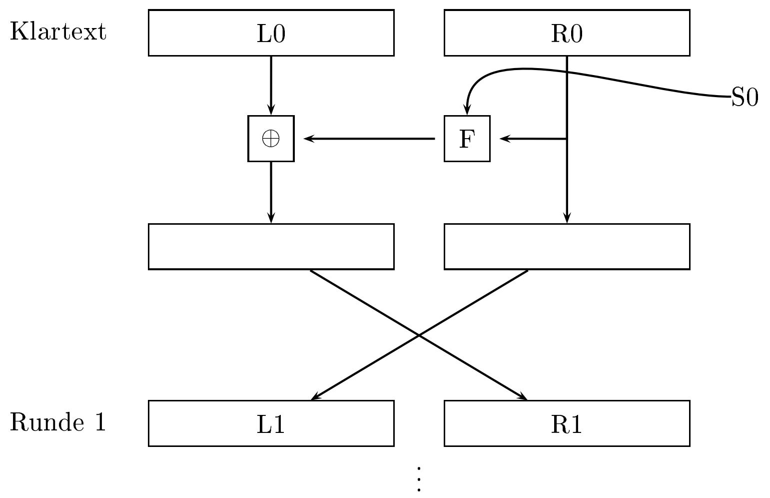
KE 2: Verschlüsselung und Digitale Signaturen

Verschlüsselungsverfahren

- symmetrische Verfahren (private key)
  - 1 Schlüssel für Ver- und Entschlüsselung
  - relativ schnell zu berechnen
  - relativ sicher (moderne Verfahren) bei ausreichender Schlüssellänge
  - Negativ: Schlüssel muss geheim bleiben
- asymmetrische Verfahren (public key)
  - 2 unterschiedliche Schlüssel für Ver- und Entschlüsselung
  - 1 Schlüssel (public key) kann veröffentlicht werden
  - Negativ: relativ langsam zu berechnen

Wichtiger symmetrische Verschlüsselungsalgorithmus:

Feistel Schema Grundlage für DES, IDEA usw.



Vorteil: Für Ver- und Entschlüsselung wird nur eine Einwegfunktion benötigt, die nicht invertierbar sind muss, da der selbe Algorithmus in gleicher Reihenfolge auch für die Entschlüsselung genutzt wird, wobei lediglich Cipher- und Plaintext ausgetauscht werden.
Nachteil: Nur linkes Teilwort wird jeweils verarbeitet.

Asymmetrische Verschlüsselungsverfahren

RSA (Rivest, Shamir, Adleman)
Grundproblem: Primfaktorzerlegung großer Zahlen

- Seien p, g zwei ähnlich große *Primzahlen*, $n = p \cdot g$ und m die *Nachricht* als Zahl mit der Eigenschaft $m \leq n$. Dann gilt $\forall k : m^{k(p-1)(q-1)+1} \bmod n = m$.
- Öffentlicher Schlüssel:
$$n, e \text{ (encrypt) teilerfremd zu } (p-1)(q-1)$$
- Privater Schlüssel:
$$d \text{ (decrypt) mit } e \cdot d \bmod (p-1)(q-1) = 1$$
so dass
$$e \cdot d = k(p-1)(q-1) + 1$$
ist.

- Verschlüsselung:
$$\mathrm{Crypt}(m) = m^e \bmod n$$
- Entschlüsselung: Mit $c = \mathrm{Crypt}(m)$ gilt dann
$$\mathrm{DeCrypt}(c) = c^d \bmod n$$

Da die Multiplikation kommutativ ist, kann die Ver- und Entschlüsselung auch umgekehrt, also erst mit dem privaten und dann mit dem öffentlichen Schlüssel erfolgen.

El Gamal, Diffie-Hellmann Grundproblem: Diskreter Logarithmus in Restklassenringen bzw. multiplikativen Gruppen.

Hash-Funktionen

Abbildungsfunktion H einer großen Ursprungsmenge auf eine kleine Zielmenge. Voraussetzungen für die Kryptographie dabei sind u.a.

- Breite Streuung
- Funktion ist eine Einwegfunktion (leicht zu berechnen aber schwer zu invertieren)
- Schwache Kollisionsresistenz: Zu gegebenem M_1 kann kein $M_2 \neq M_1$ gefunden werden, so dass $\mathrm{H}(M_1) = \mathrm{H}(M_2)$ gilt.
- Starke Kollisionsresistenz: Es können keine $M_1 \neq M_2$ gefunden werden, so dass $\mathrm{H}(M_1) = \mathrm{H}(M_2)$ gilt.

Bekannte Verfahren sind *MD5* mit 128-Bit Hashwert und *SHA-1* mit 160-Bit Hashwert.

Digitale Signaturen

- Signierung: Nachricht → Hashwert → Verschlüsselung des Hashes
- Verifizierung: Verschlüsselter Hash → Entschlüsselung und abschließender Vergleich mit dem Hash der Nachricht.

Eingesetzter Standard u.a. *DSS* (Digital Signature Standard) mit El-Gamal (diskreter Logarithmus) als Grundlage.

Zertifizierung öffentlicher Schlüssel

- Zertifikat ist öffentlicher Schlüssel eines Benutzers, der mit dem öffentlichen Schlüssel der Zertifizierungsstelle entschlüsselt werden kann.
- *Trust Center* oder auch *Certification Authority* (CA)
- Oberste CA-Stelle in Deutschland ist die RegTP
- Veröffentlichung von öffentlichen Schlüsseln zur Kontrolle außerhalb der Computertechnik notwendig, z.B. in Zeitungen, Fernsehen bzw. im Bundesanzeiger. Veröffentlichung nicht des gesamten Schlüssels sondern nur des „Fingerprints“ (MD5 oder SHA-1) zur besseren Kontrolle durch menschlichen Vergleich.

KE 3: Benutzersicherheit im Internet

PGP / GnuPG

- Hybrides Verschlüsselungsverfahren
- Erzeugung eines *Session-Key*s
- Verschlüsselung des session keys mit asymmetrischen Verfahren wie RSA, DSS oder Diffie-Hellmann
- Verschlüsselung der eigentlichen Nachricht mit symmetrischen Verfahren wie IDEA, CAST-128 oder 3DES mit dem session key
- Eventuell Erzeugung einer Signatur mit SHA-1

Sicheres Surfen

Stichworte:

- SSL, sitzt zwischen TCP und HTTP
- Anonymisierer wie Remailer, Rewebber usw.

Weitere Stichworte

SSH, JavaScript, Java, ActiveX

KE 4: Anbietersicherheit im Internet

Sichere Server

- Minimale Systeme
- Aktuelle Programmversionen und Patches
- Intergritätsprüfung
- Log Dateien kontrollieren
- Korrekte Backups mit sicherer Aufbewahrung

Firewall

- Paketfilter auf IP- und Portebene

  Nachteil: Keine benutzerspezifischen Regeln möglich
- Stateful Inspection Filter

  Paketfilter mit „Gedächnis“, so dass z.B. ACK-Pakete nur nach vorherigem und dazugehörigem SYN-Paket akzeptiert werden.
- Application Level Gateway = Proxy
- Intrusion Detection Systeme (IDS)
  - Anomalie Erkennung
  - Signatur Analyse

Organisatorische Sicherheitsmaßnahmen

Planung → Realisierung → Schulung → Kontrolle

Kurs 1867

KE 1: Angriffe auf Rechner oder Netze

Sniffer

- Mitlesen aller Datenpakete, da Ethernet ein Bus-System ist.
- Funktioniert nur bei Hubs.
- Bei Switches gesperrt, da Peer to Peer Verbindung.

Spoofing

- IP-Spoofing: Vortäuschen falscher MAC bzw. IP-Adresse
- DNS-Spoofing:
  - Manipulation auf Opfer-System notwendig oder
  - (vorübergehende) Vortäuschung eines DNS-Servers.

Weitere Angriffsziele

- Buffer overflow
- URL-Hacking

Hacker-Tools

- Host-Scanner
  - Untersuchung eines Rechners auf Schwachstellen
    - Zugriffsrechte
    - Besitzer
    - Integrität / digitale Signaturen
    - Version der Software
  - Programme: COPS und TIGER
- Network-Scanner
  - Programme: SAINT

Angriffe auf Verschlüsselungen

- Brute Force
  Schutzmöglichkeiten:
  - maximale Fehlversuche begrenzen
  - Zeitschranke für Wiederholungen setzen
  - Intrusion Detection Systeme
  - Vergrößerung des Schlüsselraums

- Lineare Kryptoanalyse / Known Plaintext Attack
  sehr kompliziert, aber es ist eine bestimmte Anfälligkeit der S-Boxen im DES zu erkennen.
- Differenzielle Kryptoanalyse / Chosen Plaintext Attack
  DES ist unanfällig gegen Differenzielle Kryptoanalyse, war IBM zum Zeitpunkt der Konstruktion bekannt
- Primfaktorzerlegung
  - Primzahlen p und q sollten sich nicht zu stark unterscheiden, sonst arbeitet *number field sieve* zu effizient
  - Primzahlen sollten nicht zu nah beieinander liegen, sonst können sie mittels der *Fermat*-Faktorisierung faktorisiert werden.
  - $(p+1)$ und $(p-1)$ sollten möglichst wenige „kleine" Primteiler haben. Andernfalls ist der *Polland*–$(p+1)$ bzw. $(p-1)$–Algorithmus sehr effizient.

KE 2: Benutzersicherheit

eCommerce

- Business to Consumer (B2C)
- Business to Business (B2B)

Bezahlverfahren

- Wichtige Voraussetzungen:
  - Bequemlichkeit
  - Akzeptanz
  - Transaktionskosten verhältnismäßig
  - Sicherheit
    - Vertraulichkeit
    - Integrität
    - Authentizität
    - Verfügbarkeit
- Verfahren
  - Nachnahme
  - Überweisung
    Neue Form: „Stackbox", Überweisung on demand über JavaApplets des Verkäufers mit TAN
  - Lastschrift
  - Kartenzahlung
    - Kundenkarte
    - ec-Karte
    - Kreditkarte
      Neues Verfahren: „Secure Electronic Transaktion" (SET)
      - durch Signatur unter Einschaltung der Bank des Kunden und der des Händlers
      - Verschlüsselung der Bestellung mit öffentl. Schüssel des Händlers
      - Verschlüsselung der Zahlungsanweisung mit öffentl. Schlüssel der Bank des Händlers
  - Paybox
    Zahlungsauthorisierung über Mobiltelefon des Kunden mit Geheimzahl/PIN

Schutz des privaten PC

- Anti Viren Software
- (Personal) Firewall
- „Vernünftige“ Benutzerverwaltung
- Ausschalten unnötiger und sicherheitsrelevanter Dienste (z.B. Remotezugriff)

Biometrie

- Stimme
- Unterschrift
- Fingerabdruck
- Auge
  - Iris
  - Augenhintergrund
- Gesichtserkennung
- Handgeometrie

KE 3: Anbietersicherheit

VPN

Grundprinzip: Tunneln von Nachrichten in verschlüsselter Form

- Prinzipiell auf allen ISO/OSI Schichten möglich
  - Allgemein einsetzbar aber nur auf Ebenen 2 und 3
  - Klassisches VPN auf Ebene 3
- Generic Routing Encapsulation (GRE) auf Ebene 2
  Spezielle Erweiterung des Point to Point Protocol (PPP)
  - Microsoft Point to Point Encapsulation (MPPE)
  - Layer 2 Tunneling Protocol (L2TP)
- Hauptsächlich IPSec für VPN auf Ebene 3 (IP)
  - Im Wesentlichen definiert in
    - IP Authentication Header (AH)
    - IP Encapsulating Security Payload (ESP)
  - 2 Modi:
    - *Transportmodi*: IP Header bleibt erhalten. Es wird lediglich neuer Messge Authentication Code (MAC) erzeugt und zwischen IP Header und Nutzlast eingefügt (AH):

IP Header	IP Nutzlast	$\Longrightarrow$	IP Header	AH	IP Nutzlast

      Zusätzlich kann die Payload verschlüsselt werden (ESP):

IP Header	IP Nutzlast	$\Longrightarrow$	IP Header	ESP Header	IP Nutzlast	ESP Trailer

        - *Tunnelmodi*: IP Paket wird vollständig verschlüsselt und mit neuem IP Header versehen.

IP Header	IP Nutzlast	$\Longrightarrow$	IP Header neu	ESP Header	IP Header alt	IP Nutzlast	ESP Trailer

    - IPSec ist immer nur für eine Transportrichtung definiert.
    - Problem ist der Schlüsselaustausch. Gemäß IPSec existieren zwei Prinzipien:
        - *Manueller Schlüsseltausch*
        - *Automatischer Schlüsseltausch* z.B. mittels DSS, RSA oder Diffie-Hellmann.

Intrusion Detection System (IDS)

- Host based IDS: Wird auf dem zu überwachenden PC installiert.
- Network based IDS: Eigenständiger PC, der im promiscous mode den kompletten Netzverkehr überwacht und analysiert.
- Beispiel:
    - Host based IDS auf 1. Firewall zum Internet und allen öffentlich erreichbaren Servern in der DMZ.
    - Network based IDS in der DMZ und im lokalen Netz.
- Probleme: IDS sind selber PC mit allen Unzulänglichkeiten wie
    - Absturzgefahr
    - Angriffs- und Hackinggefahr
    - Beschränkter Platzbedarf für mitgeschnittene Pakete
    - Verarbeitungsgeschwindigkeit bei Gigabit und schnelleren Netzen
- Programme
    - Host based IDS
        - Programm *Tripwire*: Initialisierungslauf über alle zu überwachende Dateien und anschließend regelmäßiger Vergleich auf Änderungen.
          Nachteil: Sehr lange Laufzeit
    - Network based IDS
        - Überwachung der
            - *IP Header* hauptsächlich überwachte Bits/Flags: Header Length, Fragment Offset, Time To Live, Source Address, Destination Address.
            - *TCP Header* hauptsächlich überwachte Flags: SYN-, ACK- und FIN-Flags.
            - *UDP Header* analog TCP entsprechende UDP-Flags.
            - *ICMP Messages* gesammelte Infos über ein Netzwerk
        - Programme *TCPdump, Snort*
        - Vorteil von Snort: Funktion wie TCPdump aber zusätzlich kann der Paketinhalt, die Payload, analysiert werden.
        - Überwachung von verdächtigen Paketen
            - Ungewöhnliche Spezifikationsausnutzung (Daten, die bereits im Handshaking-Verfahren beim Verbindungsaufbau übersandt werden)
            - fragmentierte Pakete (unzulässige Reservierung von Ressourcen für DOS)
            - Pakete an Broadcast-Adressen
            - Undefinierte Kombination von Statusbits
            - Pakete an Dienste (Ports), die nicht angeboten werden (Portscan)
- Reaktion auf IDS-Alarm

- Automatische Reaktion sehr problematisch (z.B. Gefahr des IP-Spoofings und damit Reaktion auf falschen Angreifer)
- Manuelle Reaktion
  - Ruhig bleiben, genau (potenziellen) Schaden analysieren und begrenzen
    Problem beim Abschalten oder Netztrennung, da dann keine weiteren Ermittlungsmöglichkeiten auf Verursacher mehr möglich.
  - Gegenmaßnahmen ergreifen wie z.B.
    - Firewall anpassen
    - Organisation ändern (Zuständigkeiten, 4-Augen Prinzip, Password-Handling usw.)
    - Systeme jetzt sicherer wieder imn Betrieb nehmen
    - Schlussfolgerungen für Zukunft in Orga einarbeiten.

KE 4: Rahmenbedingungen und Software-Prozesse

Gesetzliche Rahmenbedingungen

- BGB, StGB, AGB, FernAbsG, VerbraucherKrG, BDSG, UrhG, SigG
- TKG, TDSV, TKÜV, UWG, HGB, KonTraG
- TDG, TDDSG, MDStV, HGB, AO, JuSchG, ProdukthaftungsG

Sichere Systeme

Da die meisten Sicherheitsmängel in Programmen (auch Betriebssystem-Programme) auftreten, ist die Programmierung sicherer Programme der größte Schutzfaktor. Die Einteilung erfolgt gemäß der fünf Phasen bei *Software–Engineering I.*

- *Analyse*: Potenzielle Risiken mit einplanen.
- *Design*: Umsetzung der erkannten Risiken in Architektur wie z.B.
  - Auswahl des Betriebssytems
  - Verteilung der Anwendung auf unterschiedliche Systeme
  - Kein Einsatz unsicherer Grundlagen (Datenhaltung ohne Schutzmaßnahmen)
- *Implementierung*:
  - Einsatz umfangreicher Testmöglichkeiten vorsehen
  - Debug-Anweisungen wie z.B. *assert* in C ausführlich nutzen
  - Gängige Fehler vermeiden und kontrollieren wie z.B.
    - Buffer overflow ⇒ Externe Eingaben alle prüfen
    - Stack overflow ⇒ Rekursions- und Schleifenende überprüfen
    - Beim Einsatz stark getypter Sprachen *alle* Variablenzuweisungen prüfen.
- *Test*:
  - Selbstlernende Testgrundlagen (alle bereits erkannten Fehler müssen vollständig zu den vorherigen Testdaten hinzugefügt werden)
  - Audits zur Qualitätskontrolle, um Analyse- und Designfehler erkennen zu können.
  - ISO 9000 enthält entsprechende Mindestanforderungen.
- *Betrieb*:
  - Technische Aspekte beachten (z.B. funktionieren Backups und können diese zurückgespielt werden?)
  - Organisatorische Aspekte vorsehen und beachten (z.B. werden Backpus angefertigt, auch im Urlaub, und sicher verwahrt?)
  - Ständige Überwachung beider genannten Punkte