Einführung in Security

HERAUSFORDERUNGEN IN DER SECURITY

- Sicherheit ist nicht wichtig, [den Leuten] ist wichtig, dass es funktioniert!
- Sicherheitsgruppe sagt immer Nein!
- Sicherheit ist ein Prozess(Bruce Schneier)

DEFINITION SICHERHEIT RISIKO

- „Sicherheit ist eine Sachlage bei der das Risiko nicht größer als das Grenzrisiko ist.“
- „Das Grenzrisiko ist das größte noch vertretbare Risiko eines bestimmten technischen Vorganges oder Zustandes.“
- „Eine absolute Sicherheit ohne jegliches Risiko gibt es weder in der Technik noch in der Natur“

Risiko = Schaden * Eintrittswahrscheinlichkeit

UNTERSCHIEDLICHE SICHERHEITSZIELE

- Vertraulichkeit
- Integrität
- Verfügbarkeit
- →❽CiA Triad❽ (Confidentiality, Integrity, Availability)

WEITERE DEFINITIONEN

- Bedrohung/Threat → Potenzielle Verletzung der IT-Sicherheit
- Angriff/Attack → Aktion, die eine Bedrohung wahr werden lässt
- Angreifer/Attacker → Subjekt, das einen Angriff durchführt
- Schwachstelle/Vulnerability → Sicherheitsfehler im System, welchen man für Angriff ausnützen kann
- Exploit → Software, die eine Schwachstelle ausnützt

Wenn jemand in ein System einbricht, nutzt dieser Angreifer Fehler in Prozessen, Technik oder Management aus um unberechtigt auf Daten zuzugreifen oder Aktionen auszulösen.

KATEGORISIERUNG VON ANGRIFFEN

- Unberechtigter Zugriff Auf Daten
    - Sniffing
    - Man in the Middle
- Täuschung/Akzeptanz von falschen Daten
    - Spoofing

    - Man in the Middle
- Unterbrechung der Funktionalität
    - Denial of Service(Dos)
    - Distributed Denial of Service(DDoS)
- Widerrechtliche Verwendung

MEHRERE PHASEN EINES ANGRIFFS

- Sammeln von Daten über das Angriffsziel
    - Zb. Verwendete Systeme, User-Kennung
- Ausnutzen von gefundenen Sicherheitsproblemen
    - Zb. SQL-Injection
- Aufrechterhaltung des Zugriffs
    - Anlegen eines eigenen Benutzer-Accounts
- Verwischen von Spuren
    - Löschen von Logfiles

KOSTENFAKTOR IT-SICHERHEIT

- Oftmals nur Investitionen in Sicherheit für das Management sichtbar
- Ergebnisse dieser Investition bei guter Arbeit oft unsichtbar(„passiert ja eh nix“)
- ROI-Return of Investment
- Aufwand/Nutzen schwierig zu messen
- Erfahrungswerte für
    - Kosten bei erfolgreichen Angriffen
    - Auftrittswahrscheinlichkeit von Angriffen

- Sicherheitsprobleme treten auf!
- Das Wissen um Motivation und Bedrohungspotenzial von AngreiferInnen hilft bei der Absicherung
- Bedrohungen und Angriffe in der IT ähnlich Bedrohungen und Angriffen abseits der IT
- Angriffe werden in Phasen unterteilt
- Es gibt viele unterschiedliche Herausforderungen in der IT-Security
- Security ist vielschichtig, Technische/Organisatorische Lösungen
- Sicherheitsziele, Schutzbedarf
- Kosten/Nutzen, Risiko

Kryptographie,Verschlüsselung,Key-MGM

KRYPTOGRAPHIE

- Wissenschaft von der Geheimhaltung von Nachrichten.
- Unverschlüsselter Text (Klartext, Plaintext) wird mittels einer Funktion, die durch einen Schlüssel (Key) parametrisiert wird, in den verschlüsselten Text (Ciphertext) umgewandelt.
- Komplexes, mathematisches Problem als Basis
- Schutzziele:
    - Vertraulichkeit → Verschlüsselung
    - Integrität → Prüfsumme/Hash
    - Authentizität → Signatur
- Unterscheidung in symmetrische und asymmetrische Kryptographie
    - Unterschiedliche Einsatzzwecke
    - Unterschiedliche Anforderungen an die Umgebung
- Kerckhoffs Prinzip: Sicherheit darf nur von Geheimhaltung des Schlüssels abhängen. Nicht von der Geheimhaltung des Algorithmus!

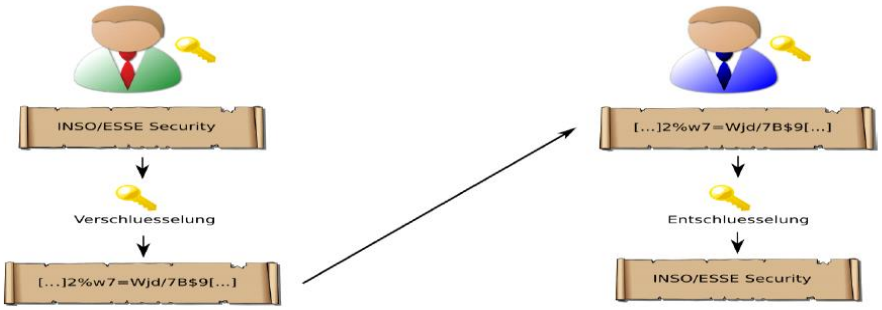
HASH-VERFAHREN

- Hash ist ein eindeutiger Fingerprint eines Dokuments

- 2 Docs mit unterschiedlichen Inhalten dürfen (in der Theorie) nicht denselben Fingerprint erhalten

- Bsps für Verfahren:
  - MD5, SHA-1 → gelten bereits als unsicher
  - SHA-2
  - SHA-3

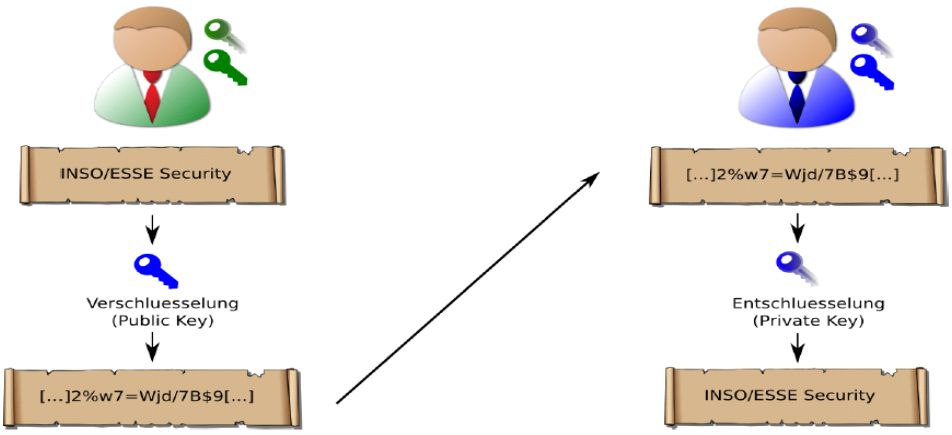
- Symmetrische Kryptographie



- Beispiele: Data Encryption Standard (DES), Triple-DES (3DES), Advanced Encryption Standard (AES)
- Problem ist Schlüsseltausch bei großer Anzahl an TeilnehmerInnen

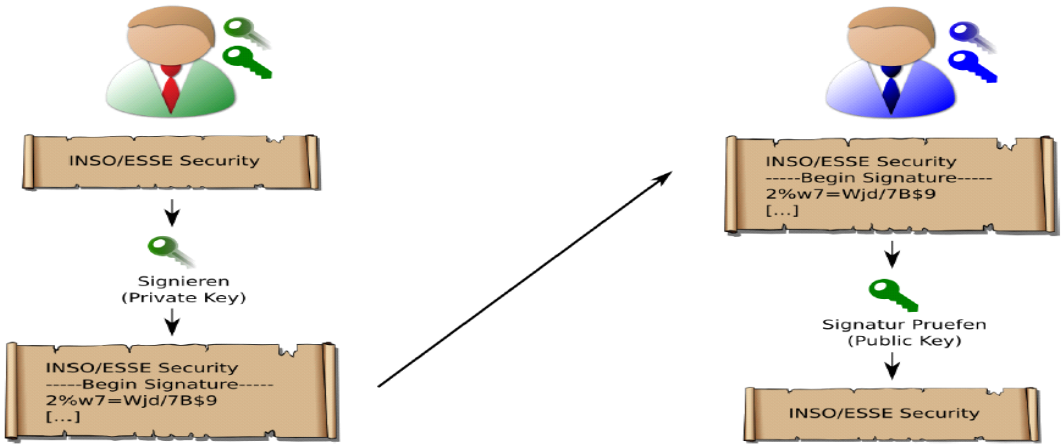
- Asymmetrische Kryptographie

Asymmetrische Kryptographie: Verschlüsselung



- Beispiele: Rivest-Shamir-Adleman (RSA), ElGamal

Asymmetrische Kryptographie: Signatur



Beispiele: Rivest-Shamir-Adleman (RSA), ElGamal

HYBRIDVERSCHLÜSSELUNG

- Kombination aus symmetrischer und asymmetrischer Verschlüsselung
- Asymmetrische Verschlüsselung → Rechenintensiv
- Symmetrische Verschlüsselung → Schlüsseltausch

- Ablauf:
    - Verschlüsselung der zu verschlüsselnden Daten mit einem neu erstellten symmetrischen Key
    - Verschlüsselung des soeben erstellten symmetrischen Keys mit einem public Key
    - Asymmetrisch verschlüsselten Symmetrischen Key an das symmetrisch verschlüsselte Doc anhängen

GÜLTIGKEITSMODELLE BEI SIGNATUREN

- Festlegung der Regeln für die Überprüfung einer Signatur
- Unterschiedliche Modelle vorhanden
    - Schalenmodell → Zum Zeitpunkt der Prüfung alle Zertifikate im Pfad gültig
    - Kettenmodell → Zum Zeitpunkt der Erstellung der Signatur war jeweils das signierende Zertifikat gültig
    - Hybridmodell → Zum Zeitpunkt der Erstellung der zur prüfenden Signatur waren alle Zertifikate im Pfad gültig

PROBLEME UND LÖSUNGEN BEI KRYPTOGRAPHIE

- Kryptographische Verfahren sind komplex → Complexity is the worst enemy of security (B. Schneier)
- Alterung von Algorithmen/Schlüssellängen
    - Angreifer kann Daten ohne Kenntnis des privaten Schlüssels entschlüsseln (zb. Mittels Brute-Force)
    - Erfordert regelmäßige Umschlüsselung von Daten unter Verwendung neuer Algorithmen/längerer Schlüssel

PRIVATE KEY

- Sichere Verwahrung des privaten Schlüssels → Sicherer Schlüsselspeicher
- Verlust des privaten Schlüssels → Verlust der verschlüsselten Daten
    - Backup des privaten Schlüssels erforderlich
    - Wenn nur für Signaturen eingesetzt → kein Backup erforderlich

PUBLIC KEY

- Integritätsschutz für Schlüssel
    - Angreifer kann Schlüssel manipulieren → Man in the Middle
- Schlüssel enthält keine Information zum Besitzer
    - Zuordnung zu einer Person schwierig
    - Verwendung von Zertifikaten mit zusätzlichen Infos zum Besitzer

PUBLIC KEY INFRASTRUCTURE

- Registration Authority (RA)
    - Registrierung neuer Anwender
        - Überprüfung der Identität
        - Weiterleitung an CA zur Ausstellung von Zertifikaten für PKI
    - Sperre von Zertifikaten/Benutzer
    - Definierte Protokollen für Zertifikatsantrag und –Austausch (Certificate Management Protocol)

- Certification Authority (CA)
    - Zuständig für Ausstellung und Verwaltung der Zertifikate

        - Endbenutzer Zertifikate werden durch CA-Zertifikat signiert
        - Erstellung/Signatur von Sperrinformationen
- Verzeichnisdienst (DIR)
        - Zugriff auf öffentliche Zertifikate
            - Zertifikate zur Prüfung von Signaturen erforderlich
        - Zugriff auf Zertifikat muss eindeutig sein
- Sperrinformationen
        - Vertraulichkeit des privaten Schlüssels erforderlich
            - Wenn nicht gegeben, kann private key nicht trusted werden
            - Auch sämtliche signed docs mit dem key nicht mehr trustworthy
        - Anwendung muss Sperrstatus eines empfangenen Zertifikats prüfen
- Zeitstempeldienste (TSA)
        - Service stellt signierte Zeitstempel aus
            - Dient als Nachweis, dass Daten zu diesem Zeitpunkt existierten
        - Ablauf
            - Client sendet Hash-Wert des Dokuments an TSA
            - TSA fügt Zeitstempel ein und signiert diese Kombination
            - Client fügt diese Info zum Doc hinzu
- Verwendung von PKI-Funktionalität
        - Abfrage von Zertifikaten aus dem Verzeichnisdienst
        - Signatur- und Pfadvalidierung
        - Signaturerstellungseinheit
        - Kryptographische Algorithmen
- Anwendungen
        - Mail-Client-Applikation
        - XML- oder Web-Service-Client-Applikation
- Zertifikate als Entität für die Vertrauensstellung
        - Zuordenbarkeit eines Public Keys zu einer Identität
        - Bestätigung wird durch Zertifizierungsstelle (CA) durchgeführt
        - Zertifikat ist im Rahmen der PKI öffentlich

- PGP = Pretty Good Privacy

CHIPKARTEN

- Speicherkarten
    - Keine Sicherheitsmerkmale für eine Zugriffskontrolle vorhanden
- Prozessorkarten
    - OS der Karte regelt Zugriff auf Daten
    - Möglichkeit Zugriff über Personal Identification Number (PIN) zu regen
    - Anwendung zb mit kryptographischen Schlüsseln
    - Private Key verlässt Chipkarte nicht
- USE: E-Card, SIM, Bankomatkarte
- Angriffe:
    - Side Channel Attacken (Stromverbrauch)
    - Physikalisch (Temperatur)
    - Man in the Middle (z.B Kartenterminal)
    - Social Engineering (Diebstahl, Ausspionieren des PINS)

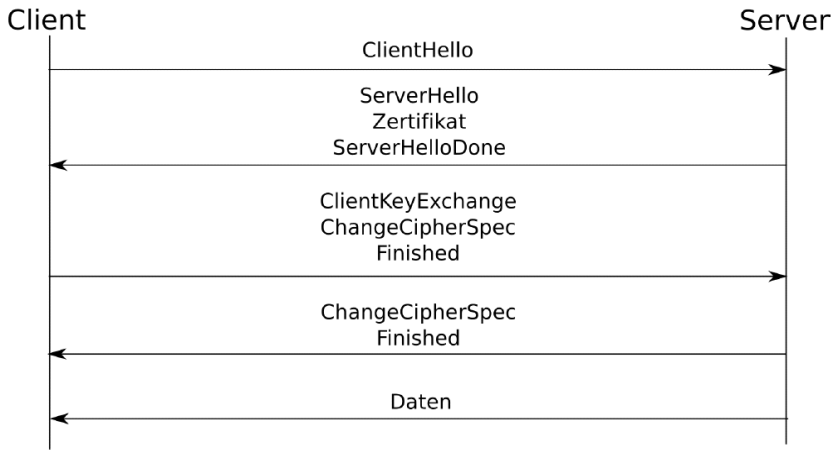
KARTENLESER

- Zum Lesen und Schreiben von Daten
- Untersch. Sicherheitsklassen
    - 1: Keine Sicherheitsmerkmale
    - 2: Gerät enthält PIN-Pad
    - 3:PIN-Pad und Display
    - 4:PIN-Pad, Display und eigenes Sicherheitsmodul zur Authentisierung des Geräts

Transport Layer Security (TLS)

- Nachfolger von SSL (Secure Sockets Layer)
- Oberhalb TCP Layer
- HTTPS → HTTP + TLS
- Verwendung
    - Authentifizierung
    - Verschlüsselung der Kommunikation

TLS – Kommunikationsablauf



Zusammenfassung

- Grundlagen zu Kryptographie (Verschlüsselung, Signatur)
- Bestandteile einer Public Key Infrastructure
- Aufbau und Verwendung von Zertifikaten
- Vertrauensmodelle
- Anwenderkomponente anhand von Firefox
- Herausforderungen und ausgewählte Lösungen bei Kryptographie

Netzwerksicherheit

EINFÜHRUNG

- Vernetzung Wichtig
- Kleine LANs – Internet
- Verschiedenste Systeme und Applikationen
- Probleme
    - Anzahl und Komplexität der beteiligten Systeme

- Attacken über das Netzwerk einfach
- Nachvollziehbarkeit von Angriffen schwierig → Anonymität

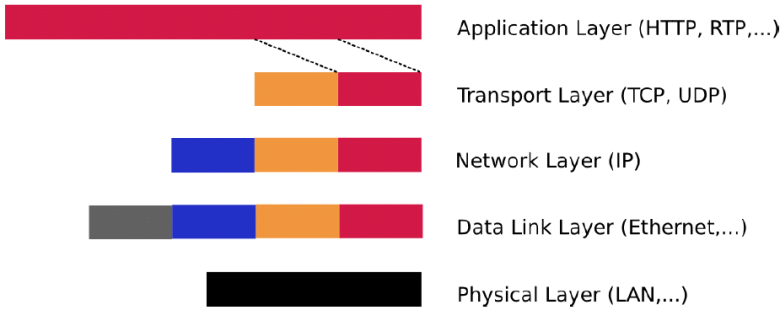
BSP FÜR TECHNISCHE UND NICHT TECHNISCHE ANGRIFFE IN NETZWERKEN

- Vertraulichkeit
  - Google Hacking
  - Sniffing
- Integrität
  - Spoofing
- Verfügbarkeit
  - Session Hijacking
  - DoS
  - DDoS

ISO/OSI-MODELL

- ISO- International Organization for Standardization
- OSI - Open Systems Interconnection
- Schichtenmodell
  - Reduzierung der Komplexität der Abhängigkeiten
  - Trennung der Aufgaben in einzelne Schichten
  - Schichten weitgehend unabhängig voneinander

Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer



Angriffe auf ausgewählte Schichten

- ARP
- IP

- ICMP
- TCP

ARP – ADDRESS RESOLUTION PROTOCOL

- Umwandlung von IP-Adressen in Hardware Adressen (z.B Ethernet MAC-Adressen)
- ARP Poisoning
    - Black Hole
    - Man in the Middle
- Spoofing(Program Disguising as another by falsifying data,thereby gaining an illegitimate advantage) von MAC-Adressen leicht möglich!

IP - Internet Protocol

- Unzuverlässig, verbindungslos
- Logische Netzwerk Adressen
- Routing
- x.x.x.x (zb. 192.168.1.1)
- Address Spoofing

ICMP – Internet Control Message Protocol

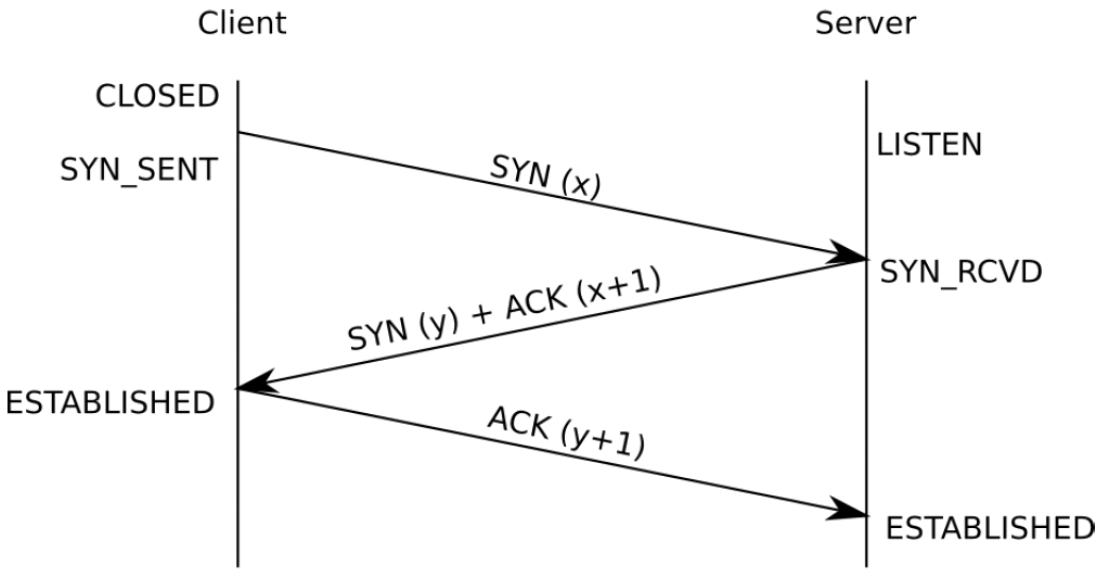
- Fehlermeldungen
    - Port Unreachable
    - Host Unreachable
    - Time Exceeded
- Diverse andere Informationen
    - Uhrzeit
    - Echo Request/ Echo Reply
- Smurf Attack → DDOS

TCP – Transmission Control Protocol

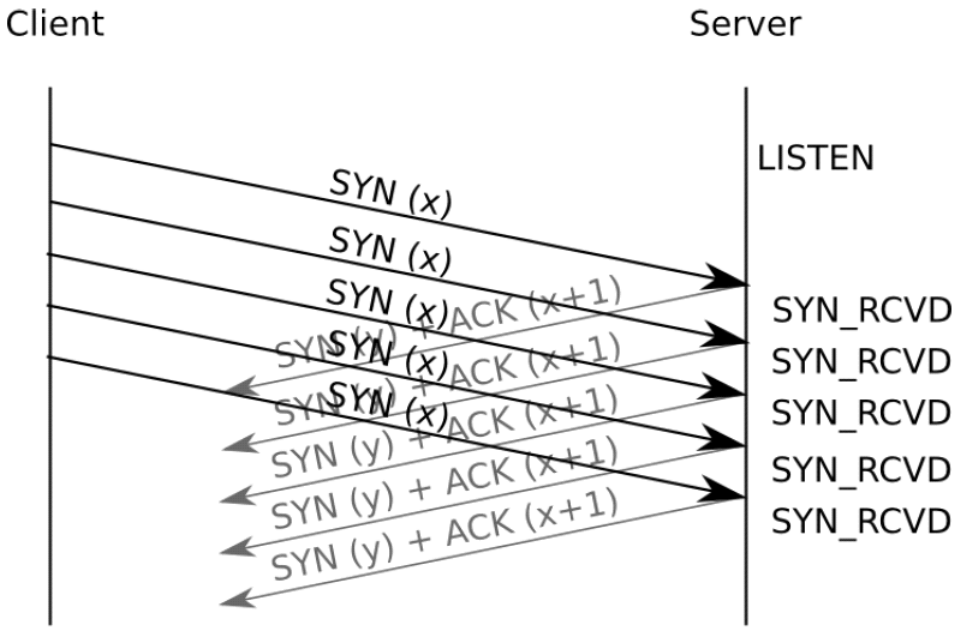
- Verbindungsorientiert (Three-Way-Handshake), verlässlich (Timeout, Retransmission)
- Exemplarische Zustände einer TCP Verbindung: Listen, Established, Closed

- Anwendung
    - WWW
    - SSH

TCP – Three-Way-Handshake



TCP – SYN Flooding



VORBEREITUNGEN FÜR ANGRIFFE AUF COMPUTERSYSTEME

- Nicht für jeden Angriff gleich
- Host Scan, Port Scan, OS Detection
- Scanning
    - Oft auffällig
    - Daher Slow/Stealth Scan
    - Ablenkung

WLAN (Wireless Local Area Network)

- Potenziell unsichere Netze
- Angreifer haben Zugriff auf Übertragungsmedium
- Hotels, Flughäfen..
- Angriffe
    - Sniffing
    - Angreifer spooft einen AP, Client
    - DOS

ABSICHERUNG VON WLAN

- WEP – unsicher
- WPA-WPA2 Gute Kennwörter erforderlich
- VPN

SNIFFING

- Netzwerkverkehr mitlesen
- Kabelgebundenes Netzwerk vs. WLAN
- Tcpdump
- Wireshark

LÖSUNGSANSÄTZE

- Absicherung der Clients/Server
- Sicherung des Übertragungswegs (internet,WLAN) via Virtual Private Network(VPN), SSL/TLS

- Zusätzliche Maßnahmen wie Intrusion Detection Systems, Intrusion Prevention Systems, Honeypots, -nets, -clients

Firewalls

- Unterbindung von unerlaubten Zugriffen
- Arten
    - Paketfilter
    - Proxy Firewall
- Positionierung von Firewalls an der Grenze zwischen zwei Netzwerkzonen(Zonenmodell)#

INTRUSION (Detection/Prevention) Systeme

- Intrusion Detection System (IDS)
    - Erkennung von Angriffen
        - Signaturen
        - Anomalie-Erkennung
    - Senden eines Alarms bei erkanntem Angriff
- Intrusion Prevention System (IPS)
    - Zusätzlich zur Erkennung, Automatische Ergreifung von Maßnahmen
        - zB Aktivierung einer Firewall Regel

HoneyPot und HoneyNet

- Honeypots sind eine ergänzende Sicherheitsmaßnahme
- Definition
    - „A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource. A security resource whose value lies in being probed, attacked or compromised“ (Spitzner)
- Kein Produktionssystem
- Kein legitimer Zugriff
- Honeynet ist ein Netzwerk von Honeypots
- Aufbau : wie eine Firewall bzw ein Gateway (Honeywall)
- Verbund von Honeypots mit diversen Diensten

Zusammenfassung und Ausblick

- Vernetzung fast alltäglich
- TCP/IP wurde ursprünglich nicht mit Blick auf Sicherheit spezifiziert
- Angriffe finden auf allen OSI-Ebenen statt
- Verschiedene Angriffstypen (DoS, Sniffing, Man in the Middle, Spoofing,...) bei verschiedenen Protokollen (ARP, IP, ICMP, TCP)
- „Kreative Anwendung" von Protokollen, Möglichkeiten, die nicht spezifiziert wurden
- Herstellung des „Vertrauens" fehlt
- Unterschiedliche Maßnahmen, um Netzwerksicherheit umzusetzen

Sicherheit in der Softwareentwicklung

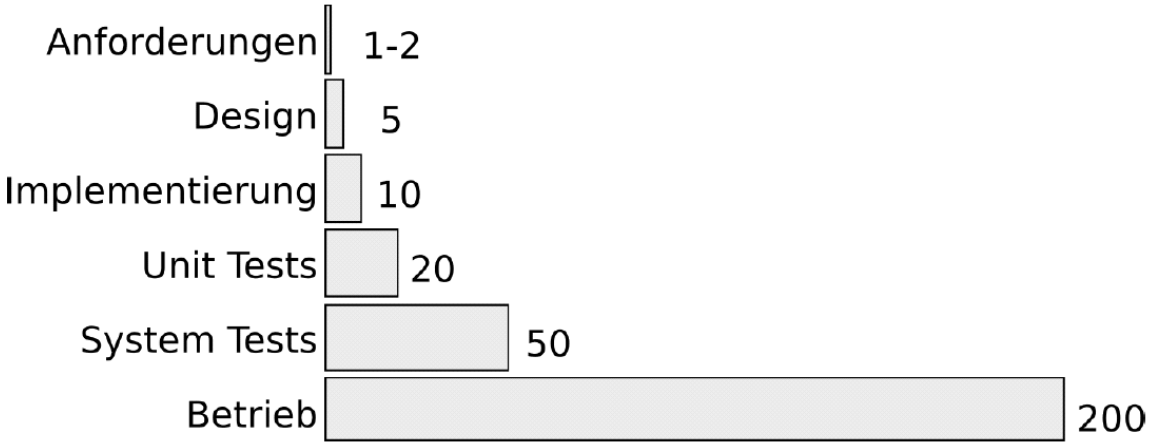
Als Software Development Lifecycle wird der gesamte Prozess der Softwareentwicklung bezeichnet

Entwicklungsphasen:

- Analyse/Anforderungen
- Entwurf
- Implementierung
- Test
- Betrieb

Sicherheit ist kein Feature –„Sicherheit ist ein Prozess" (Bruce Schneier) → es ist nicht ausreichend Sicherheit in einem der Schritte zu betrachten (z.B Verwendung einer Firewall)

Relative Kosten von Softwarefehlern

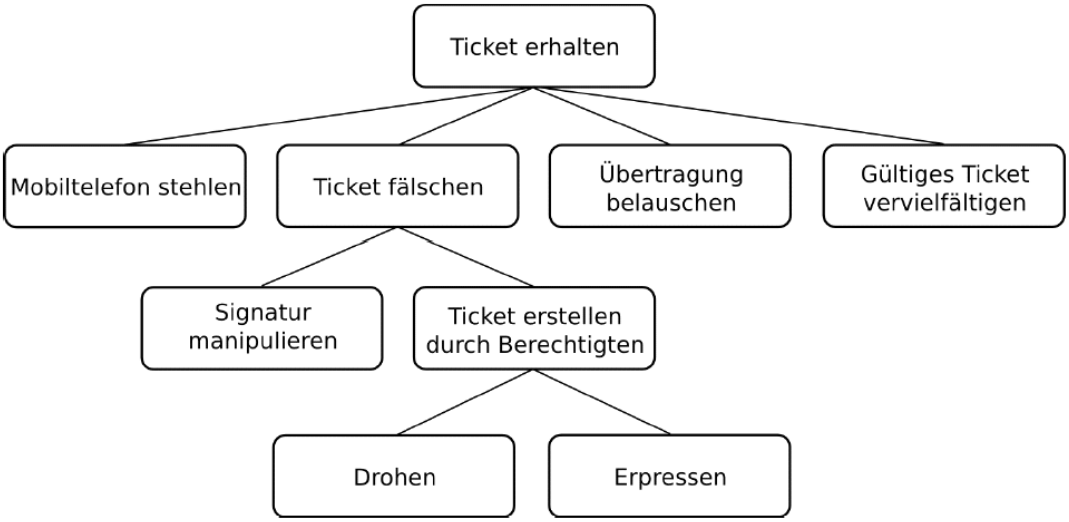


Z.B. wenn die Fehlerbehebung in der Phase der Anforderungen 1 EUR kostet, kann das 50 EUR in der Phase der System Tests kosten

SICHERHEIT IN DER ANALYSEPHASE

- Als Analyse wird die Erhebung und Aufbereitung der Sicherheitsanforderungen für das zu entwickelnde System bezeichnet.
- Funktionale und nicht-funktionale Sicherheitsanforderungen
- Unterschiedliche/Widersprüchliche Anforderungen von Stakeholdern(am System beteiligte Personen)
- Sicherheit wird als selbstverständlich angesehen und nicht explizit von den Stakeholdern gefordert
- Durchführung der Analyse
    - Ermittlung der Bedrohungen und Risiken
    - Werkzeuge zur Durchführung vorhanden, zb. Attack Trees /Threat Modelling
- Attack Trees
    - Technik zur Dokumentation von Bedrohungen
    - Identifizierung möglicher Bedrohungen mit iterativer Detaillierung
    - Knoten/Kanten können zur Bewertung der Eintrittswahrscheinlichkeit einer Bedrohung verwendet werden. Größere Systeme → schnell unübersichtlich!

Beispiel für Angriffsbäume/Attack Trees



- Threat Modelling
    - Methode von Microsoft zur Modellierung von Bedrohungen in der Software
    - Frühzeitige Beschäftigung mit den möglichen Bedrohungen
    - Bedrohung identifizieren – Denken wie ein Angreifer

SICHERHEIT IN DER ENTWURFSPHASE

- Entwurf des Systems
    - Abbildung von Sicherheitsanforderungen durch den Entwurf
    - Best Practices
        - Security Patterns stellen für immer wieder auftretende Herausforderungen im Sicherheitsbereich Entwurfslösungen zur Verfügung
        - Attack Patterns bescheiben Angriffe inklusive der Vorbedingunen welche erfüllt werden müsen
- 8 Design Principles
    - Einfachheit der Schutzmechanismen (Design komplex →Fehleranf↗llig)
    - Fail-Safe Defaults → Im Fehlerfahll auf einen sicheren Zustand zurück gehen
    - Vollständige Berechtigungsprüfung
    - Offenes Design → Sicherheit des Systems darf nicht auf der Geheimhaltung des Designs

oder der Implementierung beruhen. + Kerckhoffs Prinzip.

- Separation of Privilege → Zugriff auf Infos durch zb 2 kryptopgraphische Keys
- Minimum an Rechten verteilen
- Minimum an gemeinsamen Mechanismen (Abhängigkeiten minimieren)
- Psychologische Akzeptanz (Usability)

SICHERHEIT DER IMPLEMENTIERUNG

- Eine sichere Softwarearchitektur kann durch Programmierfehler unsicher werden
- Eine Software ohne Schwachstellen durch Programmierfehlen kann Schwächen durch die Architektur aufweisen
- Penetrate and Patch für sichere Software nicht ausreichend da Sicherheitsfehler nur im Betrieb gefunden und mit Patches behoben werden
- Patches behebn nicht immer die eigentliche Ursache sondern oft nur Symptome
- SQL Injections → eine Manipulation des SQL Statements durch Angreifer möglich
    - Gegenmaßnahmen
        - Jede Eingabe muss validiert werden + HTTP Header
        - Prüfung der Eingaben auf gültige Zeichen(Hochkomma evtl) → Whitelist Filtering Blacklist would be ausfiltern von möglichen Hochkommas
- Command Injection → Manipulation eines SQL Statements aber ein Systemaufruf(Programm) wird verwendet
    - Gegenmaßnahmen
        - Validierung der Eingaben
        - Beschränkung der Zugriffsrechte auf Dateien oder Rechtevergabe
- Cross Site Scripting (XSS) →Unvalidierte Benutzereingaben werden von der Software entgegen genommen und ausgegeben
    - Arten
        - Reflected XSS
        - Stored XSS
    - Gefahr
        - Session-Hijacking
        - Redirecting to other Sites
    - Maßnahmen
        - Alle Eingaben validieren vor dem erneuten Anzeigen in der Website

        - Client-seitige Verhinderung durch Browser

- XSS Reflected
    - Präperierte URL wird von Angreifer an Benutzer geschickt, Seite wird aufgerufen von Client Server liefert reguläre Seite mit extra Inhalt aus prepd URL, Browser wertet Rückgabe von Server aus, entweder direkt zum angreifer oder zu client und dann zum Angreifer
- XSS Stored
    - Angreifer fügt präperierten Inhalt ein. Persistente Speicherung des Inhalts durch Server. Seite wird aufgerufen von Client Server liefert reguläre Seite mit extra Inhalt aus prepd URL, Browser wertet Rückgabe von Server aus, entweder direkt zum angreifer oder zu client und dann zum Angreifer
- Cross Site Request Forgery(CSRF) → ist eine Design Schwachstelle
    - Durch präparierte URL kann Angreifer einen Benutzer zu einer Aktion am Server missbrauchen
    - Aktionen wie Logout, Einträge erstellen..
    - Maßnahmen
        - Shared Secrets zu jeder internen URL. → Geheimer Token den nur Client und Server kennen.
- Buffer Overflows → entstehen durch Programmierfehler, wenn keine oder falsche Längenüberprüfungen beim Schreiben in Buffer durchgeführt wird →✓berschreibt dadurch Speicherbereiche
    - Auswirkungen
        - Absturz der App durch Exceptions
        - Uncommon Systemverhalten
        - Keine Auswirkung, wenn der überschriebene Speicher nicht verwendet wird.
    - Gegenmaßnahmen
        - Use of Programmiersprachen mit Längenüberprüfung + Input Validierung

Zusammenfassung

- Sicherheit in allen Phasen der Entwicklung berücksichtigen
- Sicherheitsbewusstsein erzeugen
- Sicherheitsprobleme und Sicherheitsfeatures von Programmiersprachen berücksichtigen
- Security *immer* berücksichtigen – es werden immer wieder bereits Prototypen im Wirkbetrieb eingesetzt!
- Regelmäßige Analyse des Systems z.B. auf Grund neuer Angriffstechniken erforderlich → „Sicherheit ist ein Prozess“
- Für die Implementierung: *Input-Validierung!* (Ursache für viele Sicherheitsfehler)

Testing

"Testing is the process of executing a program with the intent of finding errors"

"Program testing can be a very effective way to show the presence of bugs, but is hopelessly inadequate for showing their absence"

WAS IST TESTEN?

- Überprüfen des aktuellen Zustands mit einem geplanten Zustand
- Validierung: Erzeugen wir das richtige Produkt?
- Vergleich: Wünsche des Kunden <--> Anforderungen
- Verifikation: Erzeugen wir das Produkt richtig?
- Vergleich: Anforderungen <--> Applikation
- Test erfolgreich, wenn Fehler gefunden wurden

HERAUSFORDERUNGEN BEIM SOFTWARE TESTING LAUT THOMPSON

- Unveröffentlichte Seiteneffekte
- Unbekannte Seiteneffekte
- Falsch implementierte Funktionalität

- Veraltete Funktionalität

FUNKTIONALE TESTS VS SICHERHEITSTESTS

- Funktionale Tests
- Spezifikation als Grundlage
- Erwarteter Output
- Sicherheitstests
- Spezifikation nicht nutzbar als Grundlage
- Suche nach ungewollter Funktionalität und Seiteneffekte

SECURITY FEATURES != SECURE FEATURES

- Sicherheitsrelevante Funtionalitäten
- Anforderungen mit sicherheitskritischen Aspekten
- Funktionale Fehler haben Auswirkung auf Sicherheit
- BSP: Authentifizierung
- Sichere Funktionen
- Sicherheitsfehler können in jeder Funktionalität auftreten
- Testen der Robustheit gegen Angriffe
- BSP: Fehlende Input Validierung

SICHERHEITSTESTS

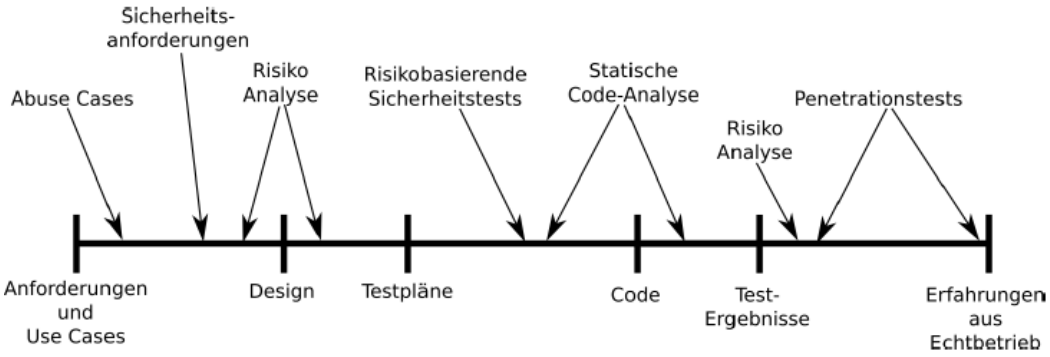
- 1k Zeilen Code --> fünf bis fünfzehn Bugs nach dem Testen
- oft unentdeckt
- keine Auswirkung auf restliche Funktionalität
- mögliche Sicherheitsfehler
- Test ist eine Momentumaufnahme und zeitlich beschränkt --> Angreifer hat beliebig Zeit
- Testen alleine reicht nicht.

DUCHFÜHREN

- Entwickler/Tester
- Fehlendes Wissen über Sicherheit und aktuelle Schwachstellen --> Schwachstellen werden leicht übersehen
- Verwendung des Systems aus Benutzersicht --> Funtkionen
- Sicherheitsexperte
- Fehlendes Wissen über Domäne und Implementierungsdetails

- Verwendung des Systems aus Angreifersicht --> Schwachstellen

Sicherheitstests im Lebenszyklus (ii)



WHITE BOX TESTS

- Wissen über Systemdetails wird für Tests herangezogen
- Verwenden von Source Code, Doc..
- Techniken : Statische Analyse, Code Reviews

BLACK BOX TESTS

- Details der Umsetzung werden für Tests nicht herangezogen
- Übergeben der Eingabedaten und Betrachtung der zurückgelieferten Ausgabe(Vergleich mit spezifizierter/erwarteter Ausgabe)
- Erforderlich wenn kein Zugriff auf die Implementierung vorhanden ist

GRAY BOX TESTING

- Kombination von White + Black Box Testing
- Bevorzugendes Verfahren für die Testdurchführung
- Datenvarianten anhand interner Details identifizieren (White Box)
- Tests mit ermittelten Datenvarianten durchführen (Black Box)

SECURE CODE REVIEW

- kostenintensiv --> aber sehr effektiv
- Top-Down Ansatz
- Ausgangspunkt: Wissen über Schwachstellen (Detailwissen über Code gering)

- Bottom-Up Ansatz
- Ausgangspunkt: Wissen über Code (Höherer Aufwand als bei Top-Down)
- Testfälle am Papier durchführen --> Überprüfung anhand von Coding Standards

STATISCHE CODEANALYSE

- Untersuchung des Codes ohne Ausführung
- Wird von Compilern genutzt
- Typprüfung: korrekte Verwendung von Typen
- Stilprüfung: Verwendung von Codierrichtlinien
- Fehlersuche: Ermittlung der Verwendung nicht initialisierter Variablen
- zusätzlich zu Source Code Review verwendbar

STATIC CODEANALYSE VARIANTEN

- Signaturbasiert
- Suchen nach Mustern wie sprintf, strcpy
- Kontrollfluss-Analyse
- Ausführungsreihenfolge der einzelnen Instruktionen

VULNERABILITY SCANNING

- Suchen nach Schwachstellen ohne weitere Ausnutzung dieser
- Bekannte Schwachstellen
- Ermitteln der Versionen
- Durchführung für jedes System wichtig

PENETRATION TESTING

- Sicherheitsüberprüfung
- Schwachstellen finden
- Ausnutzbarkeit der Schwachstellen zeigen --> Beweis für Kunden/Management
- Durchführung im Betrieb
- Berücksichtigung des gesamten Systems(OS,Datenbank..)
- Wichtig: Rechtliche Aspekte beachten!
- Abstürze oder Schäden am System unter Test
- Ist Auftraggeber berechtigt?
- Testdurchführung: Durchführung von Angriffen

FUZZING

- Random Testing
- Automatisiertes Ausführen mit zufällig generierten oder vordefinierten Werden
- Testen der Datenfelder, Struktur und Ablauf
- Zufällig generierte Daten
- Ableitung der WErte von typischen Angriffswerten (XSS, SQL Injection..) --> Fuzz-Vektor
- Fehlererkennung
- Veränderung eines Systems:Prozessorleistung, Fehlereinträge (LOGs)
- BSP HTTP Stateless, Sessions ermöglichen Zustandsspeicherung -->Muster in Session ID?

ARTEN

- Template/Block Fuzzer --> Für einfache Protokolle geeignet
- Mutation Based Fuzzer --> Verwendung bestehender daten und Abwandlung dieser
- Model Based Fuzzer --> Umsetzung eines komplexeren Ablaufs, zB Handshake

Zusammenfassung

- Sicherheit wird von den beteiligten Personen oft nicht verstanden
- Testen alleine reicht nicht aus → Sicherheit ist ein Prozess
- Priorisierung bei der Testdurchführung anhand Risk-Based Sicherheitstests
- Sicherheitstests im gesamten Lebenszyklus integrieren
- Viele Möglichkeiten für einen Angreifer
- Weiterentwicklung der Angriffstechniken → Anpassen der Testtools und regelmäßige Re-Test Durchführung
- "Sicherhheit kann nicht in ein System hineingetestet werden" (Michael Howard)

BEGRIFFE

- Authentisierung: Vorlage eines Nachweises zur Identifikation (zB Username/PW)
    - Herausforderungen:
        - Gültiger Zugang soll möglichst einfach sein
        - Kein bester Weg zur Authentisierung – Auswahl des Verfahrens je nach Einsatzzweck
    - Methoden:
        - **Wissensbasiert**: Textuelle/graphische Passwörter, PINs
        - **Besitz:** Tokens, Chipkarten
        - **Biometrisches Merkmal**: Iris, Fingerabdruck

- Authentifizierung: Überprüfung des Nachweises
    - Gründe:
        - Nachweis der Identität -Voraussetzung für Identifikation
        - Voraussetzung zur Zugriffskontrolle
        - Herstellung von Vertrauen
        -
- Autorisierung: Überprüfung der Rechte eines Users

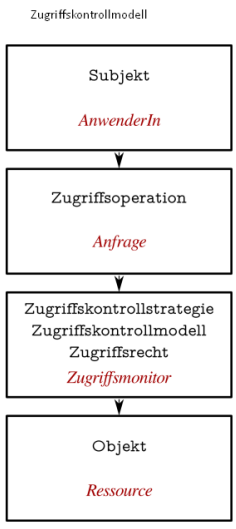
- **Wissensbasiert**/Passwörter:
    - Sicherheit abhängig von technischen und Organisatorischen Faktoren
    - Verlust schwer/zu spät erkennbar
    - Leicht zu merken vs. Schwierig zu erraten
    - REGELN:
        - Keine Default Passwörter verwenden
        - nicht aufschreiben
        - Pw soll vom Benutzer änderbar sein
        - Anzahl der Eingabe versuche beschränken
        - Prozess zur Änderung bei vergessenen Pws
    - Trotz Empfehlungen default Pw oft verwendet -> Sicheheitsbewusstsein stärken
    - Angriffe:
        - Raten
        - Social Engineering
        - Brute Force
        - Wörterbuch Attacken
        - Rainbow Tables
        - Sniffing
    - Abwehrmethoden:
        - Passwörter nur verschlüsselt abspeichern (Hash, Salt)
        - User-Training
        - Passwort Wechsel
        - Zeitliche Begrenzung der Gültigkeit der Pws
        - Fehlanmeldungen/letzter login anzeigen

- **Besitz:**

- Verlust schnell erkennbar
- Physischer Schutz notwendig
- Höhere Kosten als bei Passwörtern
- Methoden: Erzeugung von Einmal-Passwörtern (auf Basis von Uhrzeit, Counter,…)
- Chipkarten:
  - Gut: Privater Schlüssel verlässt Chipkarte nicht
  - Angriffe:
    - Side Channel Attacks
    - Physikalisch (Temperatur, Spannung)
    - Man in the Middle
    - Social Engineering

- Biometrisches Merkmal:
  - Körperliches Merkmal
  - Schwer/nicht zu ersetzen
  - Unterscheidung: Identifikation – Validierung
  - Fehlerrate (FAR: False Acceptance Rate, FRR: False Rejection Rate)
  - Höhere Kosten als Tokens oder Passwörter
  - Angriffe:
    - Spoofing (Hochauflösende Fotos z.B. 31C3, Modelierung von gefundenen Fingerabdrücken)
    - Replay Attacken
    - Umgehen der Sensoren
    - Brute Force

- Zugriffskontrolle: bedeutet Auditierung
  - Zwei unterschiedliche Fälle für Angriffe
    - AngreiferIn hat Zugriff auf Binärdaten ->Kryptografische Methoden erforderlich
    - Es existiert eine Schicht zwischen Angreiferin und Binärdaten -> Zugriffskontrolle
  - Kernfrage: Wer darf auf was wie zugreifen?
- Zugriffskontrollstrategie ist ein Regelwerk, das besagt was erlaubt/nicht erlaubt ist
- Zugriffskontrollmodell ist ein Formalismus, um eine Zugriffkontrollstrategie zu beschreiben
  - Verschiedene Zugriffskontrollmodelle für unterschiedliche Zugriffkontrollstrategien
  - Ein Modell soll einfach, ausdrucksstark, intuitiv, wartbar und umsetzbar sein.



BEGRIFFE
- Objekt: passiv, soll geschützt werden, Informationsträger
- Subjekt: aktive Elemente, die im Auftrag von AnwenderInnen Zugriff auf Informationen ausführen
- Zugriffsoperation: Art, um auf ein Objekt zuzugreifen (read, write, execute, …)
- Zugriffsrecht: Rechte für den Zugriff auf Dateien, Datenträger, Prozesse
- Schutzdomäne: Gruppierung von identischen Zugriffsrechten
- Zugriffsmonitor: setzt die Zugriffskontrollstrategie durch

GRUNDMODELL DER AUTORISIERUNG

- s ϵ S … Menge aller Subjekte
- o ϵ O … Menger aller Objekte
- a ϵ A … Menge aller möglichen Zugriffsarten

Eine Funktion f entscheidet ob der Zugriff von einem Subjekt s auf ein Objekt o, von der Art a zulässig ist. Diese Funktion wertet das Ergebnis des drei Tupels (s, o, a) (kann wahr oder falsch sein) aus.

Zugriffsmatrix:

		Objekt				
		α	β	γ	δ	ϵ
Subjekt	a	0	0	1	0	1
	b	1	1	1	1	0
	c	0	0	0	0	1
	d	0	1	1	1	0
	e	1	1	0	1	0
	f	0	1	0	0	1

- definiert die Zugriffsrechte
- bestimmt was ein Subjekt darf
- bestimmt was mit einem Objekt passieren darf
- Umsetzt direkt als Matrix langsam und ineffizient

Beispiele für Zugriffsrechte:

- Eigentümer: Uneingeschränkte Zugriffsrechte
- Gruppe: alle innerhalb einer Gruppe haben dieselben Rechte

DISCRETIONARY ACCESS CONTROL (DAC)

- Ziel: Der Besitzer legt die Berechtigungen auf Objekte selbst fest
- Zugriff auf Objekte hängt nur von der Identität ab
- Jedes Objekt hat einen Besitzer
- Der Benutzer kann Berechtigungen für Operationen an andere vergeben
- Die Feststellung wer welche Rechte hat ist aufwändig

ROLE BASED ACCESS CONTROL (RBAC)

- Ziel: Der Zugriff auf OBjekte wird durch die Rolle festgelegt
- Benutzer haben Rollen
- Eine Rolle ist eine Sammlung von Funktionen, die für eine Arbeit gebraucht werden
- Hierarchisches Rollenkonzept
- Benutzer können mehrere Rollen haben und Rollen wechseln
- Rollen die nur für eine Session gelten sollen können aktiviert werden.
- Rollen können andere Rollen ausschließen

MANDATORY ACCESS CONTROL (MAC)

- Ziel: Zentrale Festlegung der (maximalen) Zugriffsrechte von Benutzern
- Kontrolle des Informationsflusses
- Weitergabe von Rechten ist eingeschränkt möglich
- Objekte und Benutzer sind kategorisiert (öffentlich, vertraulich, geheim, …)
- Benutzer können nur auf ein Objekt zugreifen, wenn ihre Kategorisierung mindestens der des Objekts entspricht
- Objekte können nur mit Kategorien, die gleich- oder höherwertig der Kategorie des Benutzers ist, geschrieben werden
- Anwendung hauptsächlich im militärischen Bereich

HERAUSFORDERUNGEN:

- Umsetzung von tatsächlichen Berechtigungen
- Concept of Least Privilege
- Administration von Zugriffskontrolle

Zusammenfassung und Ausblick

- Identität, Authentisierung, Authentifizierung, Autorisierung
- Unterschiedliche Methoden zur Authentisierung
- Passwörter (Achtung, Passwortsicherheit im Lab!), Chipkarten, Biometrie
- Zugriffskontrolle
- Discretionary Access Control (DAC)
- Role-Based Access Control (RBAC)
- Mandatory Access Control (MAC)

- Zugriffskontrolle allein hilft jedoch bedauerlicher Weise auch nicht gegen alle Angriffe – IT-Sicherheit ist vielschichtig!

Organisatorische Sicherheit / Sicherheitsmanagement

TECHNISCHE vs. ORGANISATORISCHE SICHERHEIT

- Bestimmte Herausforderungen lassen sich besser bzw. nur durch organisatorische Maßnahmen lösen.

TECHNISCHE SICHERHEIT

- Absicherung Kommunikationswege(Lan,Wlan)
- Beispiele: Firewall, Virenscanner.

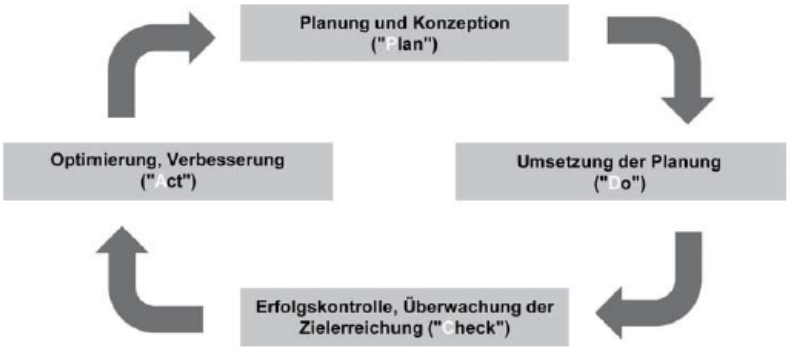
ORGANISATORISCHE SICHERHEIT

- Primär: Aufbau des IT-Sicherheits-Managements (Sicherheitsplanung)
- Formulierung einer IT-Sicherheitslinie (strategische Aussagen)
- Aufgaben:
- IT-Benutzer schulen und für Sicherheit sorgen
- Alle Maßnahmen und Veränderungen dokumentieren
- IT-Sicherheit muss vom Management, Administratoren und IT-Benutzern als fortlaufender Prozess verstanden und gelebt werden!
- Mitarbeiter schulen
- Betrifft sicheren Umgang mit IT-Systemen, als auch Unternehmensdaten
- Daten sichern
- Verlust von Daten kann zu Einschränkungen im Betrieb und wirtschaftlichen Schäden führen
- aktive Datensicherung-->Backup + regelmäßige Restores ob gesicherte Daten wiederhergestellt werden können
- Daten richtig löschen
- Unternehmensdaten dürfen Dritten nicht bekannt werden.
- EInfaches Löschen nicht ausreichend - Daten können wieder hergestellt werden
- spezielle Soft /Hardware zur Zerstörung zb thermisch
- Dokumentation
- Stetig aktuelle Dokumentation bildet of Grundlage für Problemlösung
- Aktualität,Updates
- Ständige Weiterentwicklung von Schadprogrammen
- Sicherheitsupdates, Wartung von Hardware, Virenschutz aktuell halten

- Überprüfung der Wirksamkeit von organisatorischen Sicherheitsmaßnahmen

Sicherheitsmanagementprozess

Das Sicherheitsmanagement kann dementsprechend als Prozess zum Aufbau, Betrieb sowie zur kontinuierlichen Prüfung, Steuerung und Fortentwicklung des Sicherheitsniveaus gesehen werden. Darstellung z.B. als PDCA-Modell



!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! (PDCA FRAGEN SIE IMMER)

- Sicherheitsverantwortliche Rollen werden durch Management festgelegt.Typische Rollen : IT-Sicherheitsbeauftragter, Datenschutzbeauftragter
- Größtes Risiko --> Fehlverhalten vom Management
- Policies enthalten Vorgaben, um diesem Risiko entgegenzuwirken, bzw. zu mindern

SECURITY POLICIES

- wird das angestrebte Sicherheitsniveau definiert, dass für die Erfüllung der Aufgaben des Unternehmens benötigt wird.
- Security Policy = Sicherheitsrichtlinie
- Inhalte: angestrebte IT-Sicherheitsziele/Stragien
- Anforderungen
- Vollständigkeit der Policies
- Policies müssen bekannt sein/verstanden und gelebt werden
- Policies befassen sich mit
- Verwendung von USB-Sticks
- Private Verwendung von Notebooks

- Fremde in Firmenräumen
- NormBsp ISO 27k Reihe --> spezifiziert Anforderungen an Informationssicherheits-Managementsysteme(ISMS)

IT GRUNDSCHUTZ

- "Kochbuch" für normales Schutzniveau
- Standard-Sicherheitsempfehlungen für Geschäftsprozesse, Anwendungen, & IT-Systeme
- Praktikabilität von IT-Sicherheitsanalysen nach IT-Grundschutz
- Kosteneffektive Erhöhung des IT-Sicherheitsniveaus

COBIT 5/ITIL

- Rahmenwerk für das Management und die Steuerung der Unternehmens-IT
- COBIT definiert nicht, wie Anforderungen umzusetzen sind, sondern primär was umzusetzen ist
- ITIL --> Richtlinien was man wie umsetzen soll
- Policies -> grundlegende Ziele, die eine Organisation erreichen möchte
- Prozesse -> erforderliche Umsetzung, um die Ziele zu erreichen
- Verfahren -> wer führt was wann aus, um die Ziele zu erreichen
- Arbeitsanweisungen -> Anweisungen um spezifische Aktionen auszuführen

SICHERHEITSKONZEPT BESCHREIBT

- notwendigen Maßnahmen zur Realisierung und Aufrechterhaltung eines angemessenen und definierten Sicherheitsniveaus für das Unternehmen
- 4 prinzipielle wichtige Fragen für das Sicherheitskonzept
- Was will ich schützen?
- Wogegen soll ich mich schützen?
- Wie kann ich diesen Schutz erzielen?
- Kann ich mir diesen Schutz leisten?

BC (BUSINESS CONTINUITY PLANNING)

- Business Impact Analyse: Risiken für jegliche Resourcen kritischer Prozesse analysieren und bewerten
- Business Continuity Strategien: für gesantes Unternehmen, dessen Prozesse und zugehörige Resourcen und Systeme
- BC - Operation: Festlegung von Reaktionen auf vordefinierte Notfallszenarien
- Umsetzen und Implementieren, Aufrechterhalten: Fortbildungs- Awareness- und Trainingsprogramme

Zusammenfassung

- Technische vs. organisatorische Sicherheit
- Sicherheitsmanagement und der dahinterliegende Sicherheitsmanagementprozess
- (Security-)Policies und deren Anwendung
- Standards: ISO 27k, IT-Grundschutz, COBIT, ITIL
- Sicherheitskonzepte: Ziel und Erstellung
- BC: Sinn und Vorgehensweise

Social Engineering

"Amateure hacken Systeme, Profis hacken Menschen"

SOCIAL ENGINEER

- Hohe soziale Fähigkeiten
- In der Lage, spontan auf sich veränderte Situationen zu reagieren
- Verlangen, sich mit Neuem zu beschäftigen und Neues zu erschaffen (altes langweilt ihn)

Menschliche Faktoren von Social Engineering

Verhalten	Beschreibung	Beispiel
Reziprozität	Man fühlt sich verpflichtet, Gefallen zurückzuzahlen	Nach einer Gratisprobe kauft man eine Ware
Konsistenz	Wir sind bestrebt, in Übereinstimmung mit früheren Verhalten zu handeln	Eine Zusage kann man schwer brechen
Soziale Bewährtheit	Wir orientieren uns an andere	Produkte werden als Top Seller beworben
Sympathie	Sympathischen Menschen hilft man lieber	Attraktive Modells werden in der Werbung eingesetzt
Autorität	Anweisungen von Autoritäten werden weniger in Frage gestellt	Ein Arzt gibt seinen Patienten eine Anweisung
Knappheit	Knappe Ressourcen haben einen höheren Wert	Von einer Ware ist nur mehr ein Stück lagernd

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! (Verhaltensmuster werden gern gefragt)

SOCIAL ENGINEERING ANGRIFF

- 1. Informationsbeschaffung
- 2. Beziehungen aufbauen
- 3. Beziehungen ausnutzen
- 4. Ausführung

Angriffsziel von Social Engineering

Ziel	Beschreibung/ Beispiel
Geld	Gestohlene Kreditkartennummern, bezahlte Industriespionage und vieles mehr
Ego	Das Unmögliche möglich machen, einzubrechen in als sicher geltende Institutionen oder eine persönliche Herausforderung zu meistern
Unterhaltung	Unterhaltung für einen gelangweilten Teenager, Kuriosität
Streitsachen	Rache, divergente politische oder ideologische Auffassungen
Zugehörigkeit	Zugehörigkeit zu elitären Hackergruppen und gegenseitiger Austausch von Know How und Berichten
Status	Je schwieriger es ist, ein Ziel zu erreichen, umso reizvoller ist es

!!!!!!!!!!!!!!!!!!!!!

Betriebssystemsicherheit

"Ein Betriebssystem umfasst die Programme eines digitalen Rechensystems, die zusammen mit den Eigenschaften der Rechenanlage die Grundlage der möglichen Betriebsarten des digitalen Rechensystems bilden und insbesondere die Abwicklung von Programmen steuern und überwachen"

ÜBLICHE SICHERHERHEITSZIELE

- Vertraulichkeit
- Integrität
- Verfügbarkeit

MAßNAHMEN ZUR ERHÖHUNG DER IT-SICHERHEIT IM OS

- Härtung von Systemen
- Firewalls
- Berechtigungssysteme

MINIMIERUNG DER ANZAHL VON ANGRIFFSVEKTOREN

- Grundgedanken für die Härtung eines Systems
- Ein nicht installiertes Service kann nicht angegriffen werden
- Ein nicht vorhandenes Tool kann nicht für einen Angriff verwendet werden.
- Ein nicht vorhandenes Recht kann nicht missbraucht werden.
- Somit nur Services/Tools installieren und Rechte vergeben**(Least Privilege)** die unbedingt notwendig sind
- Härten eines Systems bedeutet Entfernung aller Softwarebestandteile und Funktionen, die zur Erfüllung der Aufgabe durch das Programm nicht notwendig sind.
- Härten selten möglich.. Fehlendes Know-How / Budget / 3rd Party Software

HÄRTEN - UMSETZUNG IN DER REALEN WELT

- Erforderliche Funktionen festlegen
- System installieren
- Nicht erforderliche Software entfernen
- falls nicht löschbar --> Dienst deaktivieren
- Zugriffsrechte strenger setzen

BSP FÜR UNIX SECURITY KONZEPTE

- Berechtigungen (ls, chmod..)

- User, Group , Others
- Chroot --> change root: / wird geändert, Erhöhung der Sicherheit
- suid--> Viele Systemprogramme nutzen das suid-Bit
- Alle Benutzer sind diesen Sicherheitskonzepten unterworfen--> Ausnahme root

GRUNDLEGENDE STRATEGIE ZUR SICHERUNG VON WINDOWS

- Versiegeln
- Das Starten von Programmen ist standardmäßig verboten und nur für und nur für zugelassene Programme erlaubt
- Isolieren
- Ausgehender und eingehender Netzwerkverkehr ist grundsätzlich verboten und nur für festgelegte Ausnahmen erlaubt.
- Herabstufen
- Die automatische Nutzung von administrativen Rechten für adminstrative Konten ist deaktiviert

WINDOWS FIREWALL

- Überwacht und filtert gesamten Datenverkehr über TCP/IP
- Ist eine Stateful Inspection Firewall
- Für alle Verbindungen, welche der PC nach außen initiiert, ist eine Rückverbindung nach innen zulässig
- Port Filter possible

USER ACCOUNT CONTROL

- Nutzung von administrativen Rechten erst durch Zustimmungsabfrage
- basiert auf MIC + Access Tokens

WINDOWS ACCESS TOKEN

- Nach erfolgreicher Anmeldung wird Windows-Zugriffstoken(Token um zu bestimmen ob rechte or not) generiert

MANDATORY INTEGRITY CONTROL (aka Integrity Level)

- Definieren das maximale Zugriffsrecht, dass ein Prozess auf ein Windows-Objekt ausüben darf

BACKDOORS,ROOTKITS

"You cant trust code you didnt totally create yourself. No amount of source-level verification will protect you from using untrusted code"

- Ständige Problematik von Hintertüren
- Absicht --> Illegal

- Rootkit (Access to a computers area that wouldnt be otherwise allowed)
- Start direkt zu Beginn des Systems
- Änderung von tiefgreifenden OS-Functions --> unbemerkt vom Benutzer)

Zusammenfassung

- Angriffe auf Betriebssysteme finden statt
- Unterschiedliche Beispiele für Betriebssysteme
- Sicherheitszielen von Betriebssystemen
- Maßnahmen zur Erhöhung der IT-Sicherheit bei Betriebssystemen
  - Linux
  - Windows
- Härtung von Systemen
- Berechtigungssysteme
- Backdoors, Rootkits