Security for System Engineering

Sichere Programmierung

Ein- & Ausgabevalidierung: Format
Datentyp,
Wertebereich
Überprüfen auf Schadinformationen

Fällen: Blacklist, Whitelist
Serialisieren: Escapen von Sonderzeichen (Integrität erhalten!)

Clientvalidierung kann umgangen werden!

Bibliotheken + Frameworks: sichere Funktionen verwenden
existierende Frameworks verwenden
statische Codeanalyse verwenden

Exception Handling: throw early, catch late
kein sensibles Infos ausgeben!

Logging: Frameworks verwenden
Einheitlichen Aufbau
+ alle sicherheitsrelevante, Authorisation, Datenmanipulation, Session Timeout

Problematik: Datenschutz, Speicherort, Löschfristen
Sensible Informationen
Nachvollziehbarkeit

Vererbung: Sicherheitsrelevante Klassen finalizen!

Programmaufbereitung: Verständlichkeit, Integrität, Authentizität stärken!

Code Obfuscation: Aufwand für Dekompilierung / Reverse Engineering erhöhen
- Layout Obfuscation
- Control Obfuscation
- Data Obfuscation

Code Encryption: Dekompilierung verhindern, Code Manipulation verhindern
- Binärcode manipulieren: Überprüfungsmechanismus
- Quellcode manipulieren: Teile nachladen

Zugriff auf Code einschränken

Code Signing: Authentizität der Applikation / des Entwicklers
Verhindern von Code Manipulation
Rechtemanagement: Java Applets

Encrypted Configuration: Konfigurationsinhalt schützen
Manipulation verhindern

Netzwerk Sicherheit ... we had no idea that this would turn into a global and public infrastructure

ICMP: echo Scanning
smurf attack: spoofed echo to broadcast
Ping of Death
ICMP flood

IP: Ping of Death
Fragment Overlapp Attack: bypass firewall
Tiny Fragment Attack: bypass firewall

DHCP: Rogue Conductors

UDP: UDP flood
UDP Storm: chargen, echo
Amplification: NTP (monlist), DNS

DNS: DoS
Answer Spoofing

TCP: Low Rate DoS: keeping sessions open
TCP Session Poisoning: inject session data
Session Hijacking
Christmas Tree Packet

Scanning: ICMP Scan
TCP: Syn Scan, Fin Scan
UDP
nmap!

→ Fingerprinting

Netzwerkzonen: unterschiedliche Sicherheitsanforderungen

⇒ Firewalls between Zones
Packetfilter, Stateful Inspection, Proxy Firewall (Application Level)

↪ Umgehen von Firewalls durch Tunneling
↪ Application Layer Firewalls verstehen Protokolle
↪ Schädlig für Malware auf Firewall

Web Application Security

Webserver Arten: Statische Webserver
Dynamische Webserver
Reverse Proxy

HTTP Methoden: GET, POST, PUT, DELETE, OPTIONS, HEAD, TRACE
-> nicht verwendete Methoden deaktivieren

HTTP Authentifizierung: Basic: base64 encoding
Digest: hashing der credentials
-> nonce beeinflusst Digest Berechnung

Session Management: Cookies (HTTP Header)
URL Parameter
Hidden Form
-> Session Hijacking!

Injection Angriffe: Eingabedaten beeinflussen interpretierten Code
-> SQL Injection, Command Injection, XML Injection, XPath Injection

Alle Eingabedaten validieren, prepared Statements, Zugriffsrechte beschränken

Unvalidated Redirects: Zieladresse berechnet aus Eingabedaten
-> Malware, Phishing

Insufficient Transport Layer Protection:
unverschlüsselte Übertragung von sensiblen Informationen
- TLS verwenden, Secure Flag bei Cookies

Failure to Restrict URL Access: security by obscurity, missing function level access control
-> Rollenbasiertes Zugriffsmodell

Insecure Cryptographic Storage: Logfiles, Backups, sensitive data exposure
-> Verschlüsselung von Datenbank etc
-> Daten und Schlüssel separat speichern

Security Misconfiguration: keine Patches, Default Accounts
-> Guidelines definieren, Audits

Insecure Direct Object References: IDs in Eingabedaten, security by obscurity
keine Kontrolle am Server
-> Zugriffsmodell, keine direkten Referenzen in Eingabedaten

Broken Authentication and Session Management: Session Hijacking
-> Zufällige Session IDs, Time Outs, Sessions terminieren, SSL

Using Components with known Vulnerabilities:
-> Installation von Patches

Google Hacking: SQL Fehlermeldungen, Konfigurationsdateien, offene Admin Schnittstellen

Browser Security: Same Origin Policy, Port Blocking, Cookie Policies

IT Risikomanagement

Risiko: Eintrittswahrscheinlichkeit • Schadenshöhe, ergibt sich aus Schwachstelle + Bedrohung

Lebenszyklus Sicherheitsprozess: Plan → Do → Check → Act

Phasen Sicherheitsprozess:

Risikoidentifikation: Erkennen + Beschreiben, Erfassung Maßnahmen
Eingangsparameter wie Schwachstellen + Bedrohung
→ Brainstorming, Angriffsbäume

Risikobewertung: Schätzung Schadensschwere (verschiedene Kategorien)
Schätzung Eintrittswahrscheinlichkeit
Ergebnis: Risikolevel

Risikobehandlung / steuerung: Risikoakzeptanz durch Management
Risikovermeidung + Risikoverminderung
Risikotransferierung: Versicherung

Risikokontrolle: Bereitstellung relevanter Risikoinformationen
nachvollziehbare Dokumentation der Risikoverarbeitung
laufendes Risiko Reporting

IT Grundschutz: Was muss ich in meiner Organisation wie schützen?

- Herausgeber: BSI
- Basis für Informationssicherheit, „Hochbruch"

Sicherheitsprozess laut IT Grundschutz: Initiierung des Sicherheitsprozess
Erstellung Sicherheitskonzeption
Umsetzung Sicherheitskonzeption
Aufrechterhaltung + Verbesserung

Ablauf Erstellung IT Sicherheitskonzept:

Strukturanalyse: Gruppenbildung: Typ, Nutzerkündung, administrative Rahmenbed.
Erfassung Anwendungen: Zuordnung zu Geschäftsprozessen
Erfassung IT Systeme
Netzplanerstellung

Schutzbedarfsfeststellung: Schutzbedarfskategorien: normal, hoch, sehr hoch
Darstellung für Anwendungen: Schadensszenarien aus Anwendersicht
Darstellung für IT Systeme: Abhängigkeiten
Maximumprinzip, Kumulationseffekt, Verteilungseffekt
Darstellung für Räume: basierend auf Ergebnissen für Anwend + IT Systeme
Darstellung für Kommunikationsverbindungen: Netzplan als Grundlage

→ Ergebnisse bieten Ansatzpunkte für weiteres Vorgehen

Modellierung des Verbunds: Auswahl der Maßnahmen
Abbildung des betrachteten IT Verbunds auf IT Grundschutz Bausteine
Auswahl zutreffenden Maßnahmen aus Maßnahmenkatalog

→ Maßnahmen sind Bausteinen zugeordnet

Basis Sicherheitscheck: Soll - Ist Vergleich
Interviews zur Erläuterung der Zertifizierung
Umsetzungsstatus kontrollieren
Dokumentation der Ergebnisse des Vergleichs

Ergänzende Sicherheitsanalyse: bei hohem Schutzbedarf
wenn kein Baustein vorhanden

→ Zertifizierung nach ISO möglich

Mobile Security immer mehr Mobile Geräte in Verwendung: 2Mrd
sensible Daten, zwei Faktor Authentifizierung
Geolokation, Kamera, Mikrofon, Bankgeschäfte

Verbreitung: Repackaging: Dekompilieren, Payload hinzufügen, Veröffentlichen
Update Angriff: Payload durch Update hinzufügen
Drive by Download
Fake Apps

Android Sandboxing: Prozessisolation: unterschiedliche Benutzer
eindeutige AID
Dateisystemrechte: eigenes Verzeichnis für jede App

Berechtigungen: definiert in Android Manifest.xml
durch Benutzer bei Installation akzeptiert
low-level Zugriff über UID/GID geschützt (Kernel)
high-level über OS geschützt

Komponenten: Activity, Service, Broadcast Receiver, Content Provider

-> Inter Component Communication: Intents
- Explizite Intents
- Implizite Intents

-> Komponente public wenn, EXPORTED-flag oder Intent Filter definiert

Confused Deputy Attack:
Verwundbare App mit höheren Berechtigungen
Intent an Verwundbare App
-> continuation-intent in Facebook Login Activity

Collusion-Angriffe: gleiche UID oder Kommunikation
Berechtigungen mehrerer kleiner Apps vereinigt

Injection Angriffe via Intents: SQL, Command, XSS

Broadcast Theft

Activity Hijacking: Starten bösartiger Activity

Service Hijacking: Android wählt Empfänger randomisiert

Intent Spoofing: Social Engineering Angriff nach Broadcast Intent

-> Gegenmaßnahmen:
- Komponenten bedacht exportieren
- custom permissions
- Explizite Intents verwenden

-> Package Name der sendenden App überprüfen
nicht sinnvoll! Spoofing

Side Channel Angriffe: Gyroscope
Kamera
RAM Usage
Ultrasonic Sound

Unsicherer Netzwerkverkehr: SSL Verschlüsselung verwenden
Certificate Pinning in App (MITM)

Local Storage Optionen: Shared Preferences
Interner Speicher
Externer Speicher
SQLite Datenbank

Shared Preferences: Mode_World_Readable / Writable vermeiden
Externer Speicher: Daten nicht vertrauenswürdig

keine sensiblen Daten auf mobilen Geräten!
-> wenn nicht anders möglich: Verschlüsseln
-> Schlüssel aus Benutzerpasswort oder Keystore / Keystore Klassen

Intrusion Detection + Intrusion Prevention

Ziele: Aufrechterhaltung der IT Sicherheit während des Betriebs
möglichst frühe und genaue Erkennung von unerwünschten Aktivitäten
→ Schadensreduzierung bei Angriffen

Intrusion Detection System: Erkennung von Angriffen + Alarm
Intrusion Prevention System: Erkennung von Angriffen + Ergreifung von Abwehrmaßnahmen
→ Zusammengefasst IDPS

Aufbau: Ereigniskomponente: Aufnahme Daten über Sensoren
Sicherungs- Absicherungskomponente: Protokollierung
Analysekomponente: Analyse gesammelter Daten + Auswertung, Alarm
Monitor + Aktionskomponente: Auswertung + Durchführung von Aktionen

Arten: Host Based IDPS: Endsystem; Filesystem, Prozesse
Network Based IDPS: gesamter Traffic, Portscans, Hostscans
Network Behaviour Analysis IDPS
Wireless IDPS

→ Positionierung NIDS: passive, in-line

Erkennungsmethoden: signature based: bekannte Signaturen
anomaly based: unbekannte Angriffe, false positive, Definition normales Verhalten
stateful protocol analysis

Honeypot + Honeynet: ausgenutzte Sicherheitsmaßnahme
leichte Analyse von authentischem Traffic

2 Arten: geringe Interaktivität: Simulation
hohe Interaktivität: reale Systeme

Erkennung neuer Angriffsmuster: Signaturverbesserungen
Umleitung verdächtigen Traffic in Honeypots

Angriffe auf IDPS: Tiny Fragment
Fragment Overlap
Stealth Scan

Maßnahmen bei Angriff: Alarm
Netzwerkverbindung Reset
Abänderung von Paketen
Blockieren des gesamten Traffics

Security + Usability

Motivation: big part of security failures due improper configuration errors
users do not understand implications of security
→ SSL warnings

Consequences: users ignoring security warnings
users disable / circumvent security mechanisms
users not using security mechanisms
users do not understand consequences of insecure actions

Usability (Jakob Nielsen): Learnability, Efficiency, Memorability, Errors, Satisfaction

Methods: Heuristic Evaluation
multiple Evaluators inspect a system individually
no users required
can be used early in project

Cognitive Walkthrough: inspect single tasks instead of whole system
find critical path to complete task (user perspective)
reflect each step

User Testing: test with real users in controlled environment
- within subject testing: each user conducts all tasks
- between subject testing: user groups conduct subset of tasks

System Usability Scale: quick and dirty post-test questionnaire
10 questions
not diagnostic: something wrong, but what?

Password + Authentication: what user knows (knowledge based)
what user possesses (token based)
what user is (biometric)

- pass algorithms
- graphical passwords: shoulder surfing
- passfaces: biased

Problematic properties of security:

- Unmotivated user property: security is secondary goal
- Abstraction property: unintuitive
- Lack of feedback property: summarized feedback inadequate
detailed feedback too complicated
- Barn door property: once something was compromised, even go back
- Weakest link property: exploration might be dangerous

Mobile Challenges

First Contact: WPS
→ Insecure: Brute force attacks
Validity of PIN is checked in two halves
Device sticks with PIN
PIN calculation by MAC

Local User Authentification
- Biometrical: Face unlock, Fingerprint
- Pin, Password
- Security Gesture
→ Security patterns: traced (starting point)

Guidelines (Ka-Ping Yee): used for design + evaluation
→ goal: minimize likelihood of undesired events
make sure tasks are accomplished correct and easily

1. provide most comfortable way to do tasks with least granting of authority
2. grant authority to others with user actions actions indicating consent
3. offer ways to reduce others authority
4. maintain awareness of others authority
5. maintain awareness of users own authority
6. protect user of agents that manipulate his authority
7. enable user express safe security policies that fit his tasks
8. draw distinctions among objects and actions relevant to the task
9. present objects and actions distinguishable
10. indicate consequences of decisions that the user is expected to make

Accessability: subset of Usability