Zusammenfassung Introduction to Security

January 12, 2022

1 Einführung

1.1 Herausforderungen

Meist ist den Nutzern von Produkten wichtiger, dass es so funktioniert wie es soll und nicht wie **sicher** es ist.

Das Sicherheitsteam muss Lösungen oft ablehnen. Dabei ist es wichtig Alternativen zeigen zu können.

Sicherheit ist ein laufender Prozess und daher muss man sich laufend darum kümmern.

Oft ist IT Sicherheit erst ein Thema wenn etwas passiert. Der Grund sind hohe Kosten für das Testen auf **Nicht-Funktionalität**.

Programmieren ist einfach zu erlernen, ohne entsprechendes Wissen werden aber leicht Sicherheitslücken eingebaut.

Die Komplexität moderner Systeme macht es schwer, dass diese den erforderlichen Sicherheitsrichtslinien genügen. Oft wachsen Systeme auch laufend und werden entsprechend dem Leitsatz "*never touch a running system*" nicht gewartet aus Angst etwas zu zerstören.

Auch die zunehmende Vernetzung stellt macht es schwer Sicherheit zu gewährleisten. Dadurch dass die meisten Geräte in unserem Umfeld online sind und über drahtlose Standards kommunizieren, bieten sich Angreifern viele neue kreative Angriffswege.

1.2 Definitionen

Begriff	Erklärung
Zero-Day-Angriff	unbekannte Angriffe gegen die noch keine Gegenmaßnahmen ergriffen worden sind
Bug Bounties	Programme in denen Firmen zahlen wenn Menschen Sicherheitslücken finden und diese nicht nutzen
Weakest Link	schwächste Sicherheitsmaßnahme, also am einfachsten angreifbar
Sicherheit	Sachlage bei der das Risiko kleiner als das Grenzrisiko ist, wobei das **Grenzrisiko** das größte noch vertretbare Risiko für ein System ist. Alles darunter wird als **Restrisiko** bezeichnet.
CIA-Triad	die drei Sicherheitsziele • Vertraulichkeit • Integrität • Verfügbarkeit
Vertraulichkeit	Informationen können nur durch Berechtigte eingesehen werden
Integrität	es können keine unauthorisierten Veränderungen an Daten vorgenommen werden
Verfügbarkeit	Berechtigte können in einer definierten Zeit den Dienst nutzen.
Bedrohung	Potenzielle Verletzung der IT Sicherheit
Angriff	Aktion, die eine Bedrohung wahr werden lässt
Angreifer	Subjekt, das einen Angriff durchführt
Schwachstelle	Sicherheitsfehler im System die für einen Angriff ausgenutzt werden kann
Exploit	Software, die eine Schwachstelle ausnützt

1.3 Angriff

Die Ziele der CIA-Triad sind nicht immer gleich wichtig, daher wird meist für jedes Ziel ein Schutzbedarf in Form von Kategrien angegeben. Etwa:

- normal

- hoch
- sehr hoch
- gering
- ...

Angriffskategorien

Unberechtigter Zugriff	etwa durch Sniffing oder MIM
Täuschung / Akzeptanz	eine Empfängerin erkennt manipulierte Daten nicht als solche. Etwa durch Spoofing (als jemand anders ausgeben)
Unterbrechung der Funktionalität	Angriff auf das Sicherheitsziel Verfügbarkeit. Etwa durch (D)DoS Angriffe
wiederrechtliche Verwendung	etwa durch Command Injection

1.4 Angriffsphasen

Zunächst werden möglichst viele Informationen über das Angriffsziel gesammelt. Dazu können verwendete Systeme, Mitarbeiternamen, Software-Versionen, etc sein. Quellen können etwaige Online Infos, Papierunterlagen im Müll und ähnliches sein.

Die Informationen über verwendete Systeme und deren Versionen können benutzt werden um bereits gefundene Sicherheitslücken auszunutzen. Das kann etwa mittels SQL Injection passieren.

In der letzten Phase kann versucht werden den bestehenden Zugriff für weitere Zugriffe aufrecht zu erhalten. Dazu kann beispielsweise ein eigener Account angelegt werden.

Zuletzt wird oft versucht Spuren zu verwischen, etwa durch Löschen von Logfiles.

1.5 Absichern

Das Durchführen einer Risiko und Bedrohungsanalyse gibt einen Überblick über die Angriffsoberfläche. Dabei werden mögliche Bedrohungen nach den folgenden Kategorien

- Zielobjekt (was soll angegriffen werden)
- Urherber (wer greift an)
- Motivation (warum wird angegriffen)
- Wahrscheinlichkeit (wie wahrscheinlich erfolgt ein Angriff)
- Auswirkungen / Kosten (welche Auswirkungen hat ein Angriff)

Durch die Risikoanalyse/bewertung wird jeder Bedrohung ein Risiko zugeordnet und kann gegenüber dem Grenzrisiko bewertet werden. Anschließend kann versucht werden Risiken zu mitigieren durch treffende Sicherheitsmaßnahmen. Wenn nach einer Restrisikoabschätzung das Restrisiko kleiner ist als das Grenzrisiko sollte der Prozess von neuem beginnen um auf neue Bedrohungen zu prüfen.

Das Bedrohungspotenzial kann von verschiedenen Faktoren abhängen:

- Skill Level: Hacker vs Script Kiddies
- White Hat vs Black Hat Hacker (gut vs böse)
- Budget: Konkurrent vs Script Kiddies
- Zeit: Menschen die das als Hobby machen werden sich wahrscheinlich länger damit beschäftigen

1.6 Sicherheit vs Kosten

Die Erfolge die durch IT Sicherheit bewirkt werden, sind meist unsichtbar und daher für das Management nur schwer nachvollziehbar. Um dem Management zu zeigen, was sie für die Ausgaben bekommen wird mit *Return of Investment* gearbeitet. Dazu müssen Informationen zu Kosten im Falle eines Angriffs geschätzt werden. Dabei gilt es viele verschiedene Faktoren einzubeziehen, etwa:

- Erfahrungswerten aus der Branche
- Service Level Agreements geben an wie verfügbar ein Service sein muss (eventuelle Pönalzahlungen)
- Gesetzesverstöße gegen etwa die DSGVO
- CEOs können sich bei grob fahrlässigen Verstößen haftbar machen
- etc

Es kann nun eine Kosten Nutzenanalyse aufgestellt werden etwa in Form eines Graphen wie in Bild 1. In der Praxis entscheidet man sich für ein optimales Kosten/Nutzenverhältnis, welches der Tiefpunkt der Gesamtkosten ist. Dabei sollte der Gesamtwert der zu schützenden Güter nicht geringer sein als die Gesamtkosten.

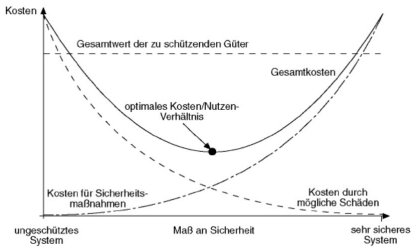

Figure 1: Kosten-Nutzen Analyse

1.7 Maßnahmen

- Security früh berücksichtigen
- genau definierte Aufgaben und Schnittstellen in der Architetktur
- bei kryptographischen Lösungen auf existierende Standards und etablierte Lösungen setzen
- zusätzlich zu den funktionalen Tests auch Sicherheitstests
- technische als auch organisatorische Lösungen zur Umsetzung

2 Kryptographie

Oft kann man sich bei der Kommunikation im Internet nicht sicher sein, ob man wirklich mit dem eigentlichen Gesprächspartner kommuniziert. Um ein gewisses Vertrauen herzustellen kann Kryptographie verwendet werden. Das Ziel dieser ist die Geheimhaltung von Nachrichten. So wird ein Plain Text in einen Cipher Text umgewandelt. Die Basis dessen ist eine Mathematische Funktion die einfach zu berechnen ist, aber die Umkehrfunktion beinahe unmöglich zu erstellen ist.

Es wird zwischen symmetrischer und asymmetrischer Kryptographie unterschieden. Weiters gilt das Prinzip, dass die Sicherheit nur durch die Geheimhaltung des Schlüssels gegeben sein darf und nicht durch die Geheimhaltung des Algorithmus.

2.1 Hash

beschreibt einen eindeutigen Fingerprint eines Dokuments mit fixer Länge. Dabei sollten Dokumente mit unterschiedlichem Inhalt nicht den gleichen Hash erhalten können. Die Hashfunktionen sind Einwegfunktionen, solche die extrem schwer umkehrbar sind. Ein Beispiel für ein Hashverfahren ist SHA-3.

2.2 Verschlüsselung

Zur Vertrauensgarantie kann kann Ver- und Entschlüsselung verwendet werden. Es wird unterschieden zwischen

- symmetrischer Verschlüsselung: Ver- und Entschlüsseln benötigt den gleichen Schlüssel
- asymmetrischer Verschlüsselung: Ver- und Entschlüsseln benötigen verschiedene Schlüssel (public, private)

Bei der asymmetrischen Verschlüsselung besitzen alle Sender einen öffentlichen Schlüssel des Emphfängers. Dieser muss nicht geheim gehalten werden. Eine mit diesem Schlüssel verschlüsselte Nachricht kann nur mit dem privaten Schlüssel des Empfängers entschlüsselt werden.

2.3 Signieren

Beim Signieren wird ein Hash des Plain Textes durch den private Key verschlüsselt

2.4 Hybrid Verschlüsselung

Da die asymmetrische Verschlüsselung sehr rechenintensiv ist und die symmetrische das Schlüsseltauschproblem hat können beide kombiniert werden. Ablauf:

1. Erzeugen eines symmetrischen Schlüssels und verschlüsseln des Plain Textes
2. Der symmetrische Schlüssel wird mit dem öffentlichen Schlüssel des Empfängers verschlüsselt
3. das asymmetrische verschlüsselte Schlüssel wird an den Ciphertext angehängt

Da nur der Schlüssel verschlüsselt werden muss, ist der Rechenaufwand vernachlässigbar.

2.5 Speichern von Schlüsseln

Schlüssel werden oft auf Chipkarten gespeichert. Dabei werden folgende Formen unterschieden

Speicherkarten	keine Sicherheitsmerkmale, der reine Besitz reicht aus um Daten zu lesen
Prozessorkarten	haben einen eigenen Prozessor und Betriebssystem welches den Zugriff auf die Daten regelt. Oft wird die Eingabe eines Pins gefordert

Üblicherweise verlässt der Schlüssel die Karte nicht und kryptographische Operationen finden auf der Karte selbst statt.

Allerdings können auch Chipkarten angegriffen werden. So versuchen Side Channel Angriffe basierend auf der Implementierung eines kryptographischen Algorithmus mit Infos die man beim Ablauf des kryptographischen Algorithmus erhält Informationen zu dem Schlüssel zu finden. Dazu zählen der Stromverbrauch oder auch die erforderliche Rechenzeit.

Eine andere Möglichkeit ist es die Karte zu einem Verhalten zu zwingen in der sie angreifbar ist. Etwa durch erhöhte Temperaturen und Spannungen. Auch kann durch einen MitM Angriff auf ein Kartenterminal andere Daten zum Signieren gesendet werden Außerdem kann auch noch Social Engeneering verwendet werden (Klauen, Erpressung...)

2.5.1 Kartenterminals

Unterscheidung verschiedener Sicherheitsklassen:

- Terminals der Sicherheitsklasse 1 haben keine besonderen Merkmale
- Sicherheitsklasse 2 hat zumindes ein Pin Pad um den Code direkt einzugeben
- Sicherheitsklasse 3 hat ein Pin Pad und einen Bildschirm der nicht durch MitM angreifbar ist
- Sicherheitsklasse 4 hat zusätzlich noch ein Sicherheitsmodul zur Authentisierung des Geräts

Neben den oben genannten Features sollten auch physikalische Schutzmaßnahmen gegen Manipulation eingebracht werden.

2.6 Public Key Infrasturcture

2.6.1 Registration Authority

Hier melden sich Anwender an und müssen je nach Sicherheitslevel verschiedene Identitäsnachweise erbringen. Nach der Identitätsprüfung wird der öffentliche Schlüssel an die Certification Authority weitergeleitet. Sollte ein von einer CA ausgestelltes Zertifikat ungültig werden oder ein privater Schlüssel nicht mehr sicher genug sein, kann die Ra auch die Zertifikate / User sperren.

2.6.2 Certification Authority

Durch die Identitäsgarantie durch die RA kann die CA garantieren, dass ein öffentlicher Schlüssel zu einer gewissen Person gehört. Weiters signiert die CA die Daten und stellt ein Zertifikat aus, das in Vereichnisdiensten veröffentlicht werden kann.

Die CA bietet auch die Möglichkeit der Schlüsselerstellung direkt an der CA bzw Karten auszustellen.

Es ist dabei wichtig, dass der private Schlüssel der CA geheim bleibt, da sonst die gesamte CA kompromittiert ist.

2.6.3 Kontrolle von gültigen Zertifikaten

Einem Schlüssel ist nicht anzusehen ob er geperrt wurde. Daher muss ein zugehöriges Zertifikat und Zertifikate selbst auch von einem Dienst beüglich seines Sperrstatus kontrolliert werden. Zwei Protokolle zum Handlen von Sperrinformationen sind Certification Revocation List (CRL) die alle geperrten Zertifikate auflistet oder das Online Certificate Status Protocol (OCSP) in dem die Gültigkeit eines Zertifikates von einem Server abgefragt wird.

2.6.4 Verzeichnisdienst

Das DIR bietet Zugriff auf die verschiedenen Zertifikate die von einer CA ausgestellt wurden. Diese Zertifikate werden werden zur Prüfung von Signaturen und sogar zum Verschlüsseln von Daten erforderlich. Die Suche nach einem Zertifikat muss dabei eindeutig sein, etwa über den DNS Namen.

2.6.5 Time Stamping Authority

Der Zeitstempeldienst bestätigt, dass ein Zertifikat zu einem bestimmten Zeitpunkt existiert hat indem er einen Hash eines Dokuments signiert

2.6.6 Zertifikat

Sie dienen der Zuordenbarkeit eines öffentlichen Schlüssels zu einer Identität. Das Zertifikat ist öffentlich via Verzeichnisdienst verfügbar. Dabei sind Informationen wie die zeitliche Gültigkeit eines Zertifikates, die ausstellende CA und den Einsatzzweck des Schlüssels. Standards sind etwa:

- X.509
- Card Verifiable Certificate (CVC)

2.6.7 X.509 PKI

Besitzt einen hierarchischen Aufbau mit einer Root CA unter dieser in mehreren Schichten weitere Sub CAs sein können. Die letzte Stufe stellen die Endbenutzerzertifikate dar. Die Vertrauensstellung erfolgt meist über die Root CA der direkt vertraut wird und den Endbenutzerzertifikaten wird inderekt über das Vertrauen in die Root Ca vertraut da diese durch diese signiert sind. So können große Bereiche leicht verwaltet werden.

2.6.8 Gültigkeitsmodelle von Signaturen

Definition: Die Zertifikatskette ist die Kette von der Root Ca über die Sub CAs bis zum Endbenutzerzertifikat.

Drei unterschiedliche Modelle:

Schalenmodell	zum Zeipunkt der Prüfung müssen alle Zertifikate im Pfad gültig sein
Kettenmodell	zum Zeitpunkt der Erstellung der Signatur muss nur das signierende Zeritifikat gültig sein
Hybridmodell	zum Zeitpunkt der Erstellung der zu prüfenden Signatur müssen alle Zertifikate im Pfad gültig gewesen sein

2.7 Web of Trust

Neben der PKI gibt es noch das Web of Trust, eine direkte Vertrauenstellung, da es keine CA gibt. Es werden öffentliche Schlüssel für email Adressen getauscht und vergleicht die Fingerprints auf einem zweiten Kanal. Grundsätzlich ist es wie Personen die verschiedenen Personen vertrauen, weil sie Personen vertrauen, die diesen vertrauen.

2.8 Pretty Good Privacy

PGP ist eine Freeware Alternative bei der zusätzlich zu den öffentlichen Schlüsseln auch noch zusätzliche Infos wie Namen oder Fotos gespeichert werden. Zusammen ergeben sie dann einen PGP /GPG Key

2.9 TLS

Trnasport Layer Security wird verwendet um HTTP Verbindungen zu sichern (HTTPS) und ist der Nachfolger von SSL. Verwendet werden kann TLS für

- Authentifizierung (einseitig oder beidseitig)
- Verschlüsselung der Kommunikation

Allerdings ist die Verschlüsselung zweier Komponenten nur auf OSI Layer 4 möglich. Auf höheren Ebenen kann das mit PGP sichergestellt werden.

Hier muss um eine TLS Session aufzubauen ein Handshakeverfahren durchgeführt werden.

2.9.1 Cipher Suite

Es gibt dann auch Cipher Suiten die verfügbare kryptographische Methoden für die Kommunikation festlegen. Server und Client legen mit dieser fest, welche sie unterstützen. Für eine erfogreiche TLS Session ist die Festlegung auf eine Cipher Suite notwendig. Gibt es keine gemeinsame wird auch keine Verbindung aufgebaut, gibt es mehrere, wird die stärkste verwendet.

Teile einer Cipher Suite

- Alg für Schlüsselaustausch

- Alg für Authentifizierung
- Alg für Verschlüsselung
- Alg für Hashfunktion

es gibt ein Bild für den Aufbau euner Verbindung, das hier einfügen

3 Netzwerksicherheit

3.1 Einführung

Das Internet war nie als globales Angebot gedacht und das hat Auswirkungen auf dessen Sicherheit. So ist die Zeit bis zu einem Angriff meist sehr kurz, vor allem durch automatische Programme wie etwa Würmer. Dadurch, dass heute viele Geräte eine Internetanbindung haben, geraten auch diese ins Visier von Angreifern. Gerade weil die Lebensdauer die der Update-Policy zumeist um Jahre übersteigt. Aus Security Sicht ergeben sich damit einige Probleme:

- Anzahl und Komplexität der Systeme erhöt deren Fehleranfälligkeit und steigert die Auswirkung von Sicherheitslücken
- Attacken über das Netzwerk sind leicht und auch aus sicherer Distanz auszuführen
- Nachvollziehbarkeit von Angriffen ist schwierig, vor allem wenn die Angreifer aus anderen Ländern stammen
- Anonymität im Netz durch verschiedene Mechanismen wie etwa VPN erschwert die Tätersuche
- Angriffsmethoden leicht und schnell verfügbar etwa in entsprechenden Foren
- Angriffe im Netzwerk oft Basis für weitere Angriffe auf das gesamte System

Mögliche Angriffe sind etwa:

- Google hacking - durch Eingabe bestimmter Suchbegriffe können Webseiten gefunden werden, wo z.B. Passwörter im SRCcode stehen
- Convert Channels - Kanäle über die unauffällig Informationen geschickt werden können
- Sniffing - belauschen von Netzwerktraffic
- Spoofing - Vortäuschen von Informationen
- Session Hijacking - übernehmen einer fremden Session, etwa TCP
- (D)DoS - (Distributed) Denial of Service

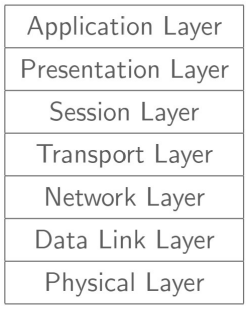

Figure 2: ISO / OSI Schichtmodell

3.2 ISO / OSI Modell

ISO- International Organization for Standardization

OSI - Open Systems Interconnection

ISO / OSI beschreibt ein Schichtenmodell (Bild 2), wobei die Aufgaben der Schichten klar voneinander getrennt sind, unabhängig voneinander funktionieren und klar definierte Schnittstellen zwischeneinander haben.

Auch das TCP / IP Modell wird fasst die Schichten 5-7 des ISO / OSI zur Applicationsschicht und die Schichten 1-2 zur Netzwerkschicht zusammen. **Bild hier einfügen**

3.3 ARP - Adress Resolution Protocol

Verknüpft die logischen Adressen aus Layer 3 mit den physischen Adressen aus Layer 2 verknüpft. Etwa IP Adressen zu Mac Adressen.

Oft werden sogenannte ARP Caches verwendet um ARP Anfragen zu beschleunigen. Es gibt keinen Integritätsschutz für ARP-Replys und so können Angreifer bestimmten IP Adressen bestimmte MAC Adressen zuordnen. Dadurch wird der ARP Cache vergiftet: ARP-Poisoning Zwei verschiedene Arten:

Black Hole	Man in the Middle
Angreifer ordnen die eigene MAC Adresse einer IP Adresse zu. Dadurch werden die Pakete nicht an das eigene Ziel weitergeleitet und es entsteht ein schwarzes Datenloch	Angreifer erhalten alle Pakete können diese lesen und speichern bzw verändern und dann an das eigentliche Ziel schicken. Ein Empfänger kann das nur schwer erkennen bzw verhindern

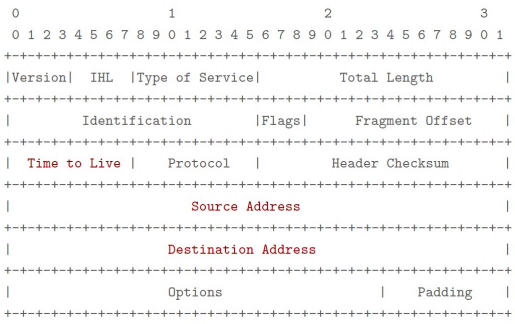

Figure 3: IPv4 Header

3.3.1 Man in the Middle Angriff

Rechner 1 will mit Rechner 2 kommunizieren. Der Angreifer sendet einen ARP Reply mit der MAC Adresse des Angriffsrechners und der IP Adresse von Rechner 2. Dadurch denkt Rechner 1, dass er Pakete an die MAC Adresse des Angriffsrechners schicken muss umd die Ip Adresse von Rechner 2 zu erreichen. So erhält der Angreifer Pakete die eigentlich für Rechner 2 bestimmt wären. Für einen MitM Attack können die Pakete weiter zu dem korrekten Empfänger geleitet werden. Dabei werden als Source IP Adresse die von Rechner 1 und als MAC Adresse die eigene verwendet. Somit erhält der Angreifer auch die Pakete von Rechner 2.

3.4 Internet Protocol

IP ist ein unzuverlässiger Standard: es wird nicht kontrolliert ob ein Empfänger überhaupterreichbar ist. Es gibt auch keine Kontrolle ob Pakete verloren gegangen sind und auch die Reihenfolge wird nicht geprüft. Um solche Kontrollen müssen sich Protokolle anderer Schichten kümmern.

IP arbeitet auf Layer 3 von ISO und mit logischen Netzwerkadressen die auf Hosts konfiguriert werden. Weiters findet das Routing auf IP Ebene statt.

Es gibt mehrere Typen von IP Adressen:

- Hostadressen für nur einen Host
- Broadcastadressen für alle Hosts in einem Netzwerk
- spezielle Adressen

Als Nachfolger von dem immer noch weit verbreiteten IPV4 ist IPv6 der viel komplexer aber auch viel sicherer mit einem größeren Adressbereich ist

3.4.1 IP Header

Von dem Header in Bild 3 kann ein Host alle Felder beliebig ändern. So kann durch manipulation der *Source Address* ein sogenanntes Address Spoofing erreicht werden. Dabei sieht es für einen angegriffenen Host so aus, als ob das Paket von einer anderen IP Adresse kommt.

3.4.2 ICMP

wird verwendet um Fehlermeldungen und Informationen zu kommunizieren

- Port Unreachable
- Host Unreachable
- Time Exceeded (IP Paket hat zu viele Router passiert -¿ TTL)
- Uhrzeit
- Echo Request / Reply

3.4.3 Traceroute

ist ein Befehl mit dem das Routing von IP Paketen nachvollzogen werden kann. Benutzt das TTL Feld im IP Header.

1. senden von Echo Request mit TTL 1 → erster Host dekrementiert TTL auf 0 und sendet ICMP Time Exceedet (daher kennen wir dessen IP Adresse)
2. senden von Echo Request mit TTL 2 ...
3. solange bis Zielsystem erreicht wurde

3.4.4 ICMP Smurf Attacke

Basiert auf IP Adress Spoofing und hat zum Ziel, dass ein Opfer mit Paketen überschwemmt wird

1. Angreifer sendet ICMP Echo Request an Broadcast Adresse mit IP Adresse des Opfers
2. Alle Hosts senden eine ICMP Echo Reply
3. Opfer erhält massig Pakete
4. DoS

Damit das nicht passiert, lassen die meisten Router / Systeme keine ICMP Echo Requests an die Broadcast Adresse zu

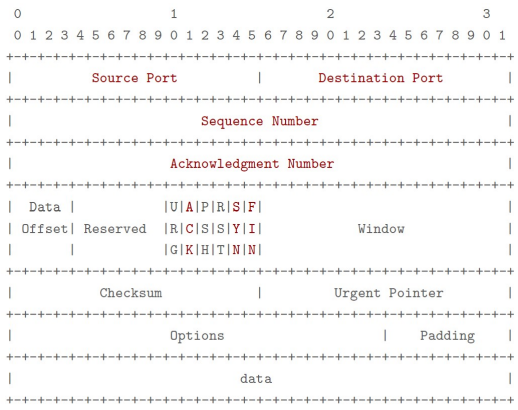

Figure 4: TCP Header

3.5 TCP

TCP ist ein zuverlässiger verbindungsorientierter Standard:

- Aufbauen einer Verbindung → Zielhost muss verfügbar sein: Three Way Handshake
- Alle Pakete kommen in der richtigen Reihenfolge an
- Verlorene Pakete werden erkannt und neu gesendet → Zielhost bestätigt das Ankommen von Paketen
- Flow Control: Zielhost gibt bekannt wie ausgelastet er ist und somit die Übertragungsgeschwindigkeit ändern

TCP Verbindungen haben verschiedene Zustände wie etwa *listen, established, closed*.

3.5.1 Header

Für uns wichtige Felder sind die rot markierten Felder in Bild 4. Beim Aufbau einer TCP Verbindung verbindet sich der Client von seiner *IP Source Address* und *Acknowledgement Number* auf die *IP Destination Address* und einen *Destination Port*.

Landattacke

Eine DoS Attacke die durch einen Implementierungsfehler möglich ist

1. senden eines SYN Pakets, in dem Ziel und Source Adresse und Port gleich sind

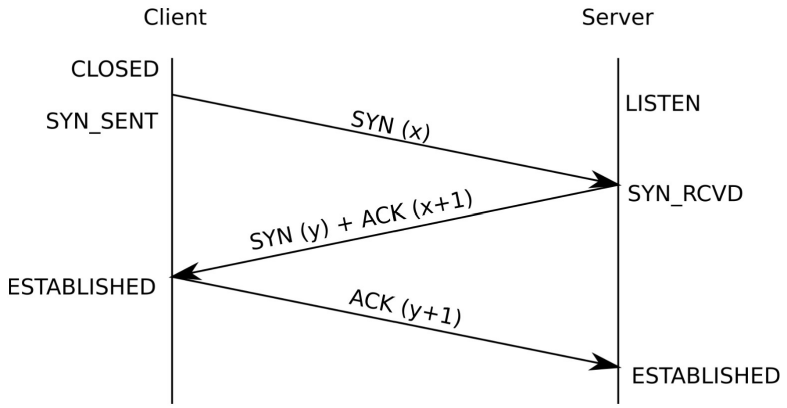

Figure 5: Der 3 Way Handshake

2. der Zielhost sendet daraufhind ein SYN/ACK Paket and den Absender, also sich selbst
3. dieses wird durch einen Implementierungsfehler als SYN Paket erkannt und zurück zu Schritt 2
4. **Racecondition!**

3.5.2 Three Way Handshake

1. client sendet SYN Paket mit Sequence Number x
2. host empfängt SYN Paket und senden ein SYN/ACK Paket mit eigener Sequence Nummer y
3. der host hat damit den Erhalt aller Pakete bis zur Sequence Nummer x
4. der host ist damit bereit für das Paket mit Sequence Nummer $x+1$
5. ohne diese Bestätigung wird erneut ein SYN Paket von dem client geschickt
6. nach der Bestätigungsnachricht sendet der client ein ACK Paket und somit ist die Verbindung aufgebaut

TCP SYN Flooding

Wiederholtes senden von SYN Paketen an einen Server. Dieser speichert die involvierenden Sequence Nummern und Acknowledgement Nummern. Der Angreifer speichert keine Daten und überflutet so den Speicher des Servers.

Verhindert werden kann dieses Vorgehen etwa durch verwenden eines SYN-Cookies. Dadurch muss ein Server im SYN_RCVD Zustand keine Daten zu einer Verbindung speichern. Der Server verwendet dazu eine spezielle Sequence Nummer um zu erkennen, ob bereits zuvor eine SYN Paket von diesem Client verschickt wurde.

3.6 Vorbereitung auf einen Angriff

Eine Möglichkeit um Informationen über ein Ziel einzuholen ist ein Host Scan. Dazu kann versucht werden eine TCP Verbindung aufzubauen indem über alle IP Adressen iteriert wird. Damit kann herausgefunden werden, welche Hosts verfügbar sind. Ein Port Scan über alle möglichen UDP / TCP Ports bringt Aufschluss über die Verfügbarkeit von Services. Durch Unterschiede in der Implementierung des TCP / IP Netzwerkstacks kann herausgefunden werden, welches Betriebssystem am Ziel installiert ist.

Bei einem Vulnerability Scan verbindet man sich mit Services und versucht aufgrund von Bannern bzw anderen Infos die Versionsnummer eines Services herauszufinden.

Die oben genannten Scans sind oft auffällig wenn davon zu viele Hosts in einem Netzwerk betroffen sind. Eventuell kann so etwas einen Alarm oder eine Firewall auslösen. Daher werden oft sogenannte *Slow Scans* oder *Stealth Scans* durchgeführt.

Slow Scans werden einfach sehr langsam durchgeführt, sodass nicht alle Hosts / Ports auf einmal gescannt werden.

Stealth Scans sind Scans die so durchgeführt werden, das sie in den log files nicht auffallen. z.B.: statt einem SYN Paket zu schicken ein FIN Paket

Zur Vorbereitung kann auch ein paralleler Angriff organisiert werden, wohingegen der eigentliche Angriff an einer anderen Stelle stattfindet

3.7 Wireless Netzwerk Angriffe

Wireless Netzwerke sind eher unsicher, da Angreifer auch aus der Distanz Zugriff auf das Übertragungsmedium besitzen. Auch wenn heutige WLAN Netze oft verschlüsselt sind ist es in öffentlichen (unverschlüsselten) meist möglich den Traffic zu sniffen udn analysieren. Mögliche Angriffe:

- Sniffing
- Spoofing eines Access Points: vorgeben ein Netzwerk zu sein durch gleichen Namen. Dadurch versuchen sich Clients mit diesem zu verbinden und übermitteln dabei Passwort und Username. Mittels Brute Force kann dann versucht das Passwort zu finden
- DoS etwa durch stören des Signals

3.7.1 Standards

WEP: veraltet, unsicher und leicht knackbar

WPA: WPA/WPA2 benötigen Passwörter die nicht über einfache Wörterbuchangriffe kanckbar sind, aber es gibt trotzdem Wege. Daher wurde WPA3 entwickelt

Tipps: um in öffentlichen Netzwerken Daten sicher übertragen zu können, werden VPNs verwendet. Das sind verschlüsselte Verbindungen bis zu einem gewissen Punkt. Daher sind auch generell End 2 End Verschlüsselungen empfohlen.

Standard Passwörter sollten geändert werden.

Netzwerknamen können uninteressant gemacht werden. das richt aber als alleinige Sicherheitsmaßnahme nicht aus.

3.8 Sniffing

beschreibt das Mitlesen von Netzwerktraffic. Es ist sowohl in kabelgebundenen als auch in kabellosen Netzwerken möglich. Für erstere wird ein physischer Anschluss an das Netzwerk benötigt und es kommt ganz auf den Teil des Netzwerks an mit dem man verbunden ist. Zusätzlich leiten Switches meist nur die Pakete weiter die für einen bestimmt sind und um mehr zu sehen braucht es einen Mirror Port für den aber zumeist root Berechtigungen notwendig sind.

mögliche Sniffing Werkzeuge sind

- tcpdump
- aircrack
- Wireshark

3.9 Schutzmaßnahmen

- Sicherheitsupdates installieren
- Netzwerk in Segmente gemäß ihren Aufgaben unterteilen
- Sicherung des Übertragungswegs (VPN TLS)
- Verschlüsseln und Signieren von Nachrichten
- Verwenden von Intrusion Detection Systemen oder Intursion Prevention Systemen oder Honeypots

Firewall

Soll unerlaubte Zugriffe unterbinden, schwierig ist es aber zu unterschieden, welche Zugriffe erwünscht und welche unerwünscht sind. Arten:

Paketfilter	analysieren Pakete aufgrund der MAC Adresse/ IP Adresse unf Ports und entscheiden ob es verworfen werden sollte
Stateful Inspection	Kennen den Status einer Verbindung (z.B. ob 3 Way Handshake durchgeführt wurde) → weitere Entscheidungsgrundlage
Proxy Firewall	Firewall die auf Applikationsebene agiert und Anfragen als Proxy weiterleitet.

In der Regel werden Firewalls zwischen zwei Netzwerkzonen platziert. Somit kann etwa durch einen Zonenplan innerhalb eines Netzwerks festgelegt werden, welche Zonen wie miteinander kommunizieren können. Zwischen zwei Firewalls gibt es eine *Demilitatrised Zone* die Services anbietet die von externen und internen Netzwerken kommen können.

IDS / IPS

Intrusion Detection und Intrusion Prevention Systeme sollen potentielle Angriffe erkennen und entsprechende Maßnahmen ergreifen. Dazu wird ein einfacher Stringvergleich zwischen bekanntem Angriffsverhalten und aufgezeichnetem Angriffsverhalten durchgeführt. Bei einer Übereinstimmung wird ein Alarm ausgelöst. Es gibt auch eine Anomalieerkennung die aufgezeichnete Muster mit erlernten Mustern vergleicht. Nach der Erkennung können dann automatische Maßnahmen ergriffen werden wie etwa die Aktivierung einer speziellen Firewallregel.

Honeypots

zählen zu den ergänzenden Sicherheitsmaßnahmen und sind nur zur Erkennung von Angriffen da und des Informationssammelns über die Angreifer. Sie haben keinen produktiven Nutzen und sollen nur angegriffen werden.

4 Sicherheit in der Softwareentwicklung

4.1 Einführung

Der Entwicklungsprozess kann durch den **Software Development Lifecycle** beschrieben werden:

- Analyse / Anforderungserhebung
- Entwurf
- Implementierung
- Test

- Betrieb

Es ist allerdings wichtig, dass der Sicherheitsaspekt in jedem Entwicklungsschritt berücksichtigt wird und auch nach der Entwicklungsphase das System konsequent an neue Bedrohungen angepasst wird. Nachrägliches Absichern gegen ein Sicherheitsproblem ist auch viel aufwendiger, da bestehendes teilweise neu aufgearbeitet werden muss. Das steigert sich bei jeder Entwicklungsphase um einiges.

Um ein Programm sicher zu entwickeln, sollen folgende Punkte erfüllt sein:

- ein Vorgehensmodell wählen, das IT Sicherheit berücksichtigt (etwa SDL oder CLASP)
- Aktivitäten und Verantwortlichkeiten für Sicherheitsprozesse definieren
- – wer ist für die Risiko und Bedrohungsanalyse zuständig
  – Sicherheitsschulungen zu sicherer Softwareentwicklung
  – bestimmen von Mitarbeitern die sich um Einhaltung der Sicherheitsprozesse kümmern
- bei Vorgehensmodellen die IT Security nicht unterstützen sollten Sicherheitsprozesse in diese integriert werden

4.2 SDLC - aus IT Security Sicht

4.2.1 Analyse

Bezeichnet die Erhebung und Aufbereitung der Sicherheitsanforderungen an das System. Darunter fallen folgende Punkte:

funktionale / nicht funktionale Sicherheitsanforderungen

funktional	ein System muss sicherstellen, dass ein Passwort das des eingegeben Benutzernamens ist
nicht funktional	auch wenn Sonderzeichen vorkommen oder der Eingabestring lang ist muss die PW Routine korret arbeiten (kein Absturz)

widersprüchliche Anforderungen von Steakholdern

Beispiel: es wird geforder nur sichere Verschlüsselungen zu verwenden aber es wird auch genau ein unsicheres Verschlüsselungsprotokoll gefordert

Anforderungen durch Normen / Gesetze: diese dürfen nicht verletzt werden und müssen umgesetzt werden

Evaluierungen bei sicherheitskritischen Systemen: Berücksichtigungen

der Evaluierungsanforderungen (etwa Protection Profiles)

Generell sehen Auftraggeber Sicherheit als selbstverständlich. Daher sollte von Anbeginn an dieses Thema angesprochen werden und Ressourcen dafür eingeplant werden

Weiters müssen in der Analyse die *Bedrohungen und Risken* ermittelt werden und die *Angriffsoberfläche* vollständig erfasst werden. Auch müssen die *Schutzziele* für verarbeitete Daten festgelegt werden. Dazu werden im Folgenden zwei Werkzeuge vorgestellt:

Angriffsbäume

Risiken werden in einer Baumstruktur dargestellt (Bild 6) Zu dem Graphn

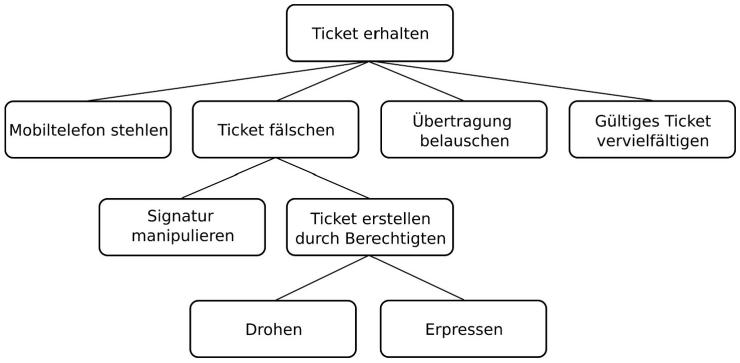

Figure 6: Angriffsbaum

können auch Gewichtungen hinzugefügt werden in Form von Eintrittswahrscheinlichkeiten. Geeignet ist ein solcher Baum für kleine und mittelgroße Projekte. Der Baum kann in einem gewünschten Detailheitsgrad erstellt werden.

Threat Modelling

Beschreibt eine Methode zur Modellierung von Bedrohungen in Software (von Microsoft). Dazu sollen sich Entwickler in die Position eines Angreifers versetzen um Schwachstellen zu finden. Dadurch wird eine frühzeitige Beschäftigung mit Bedrohungen gefördert.

4.2.2 Entwurf

Bildet die Sicherheitsanforderungen ab und dabei sollen die folgenden Punkte berücksichtigt werden:

- Berücksichtigen von Best Practices -¿ verwenden bekannter Maßnahmen

- Sicherheitsmuster die für sich wiederholende Herausforderungen Lösungen bereitstellen
- Angriffsmuster die Angriffe beschreiben inklusive Vorbedingungen
- Fehler in Analyse und Entwurfsphase können zu Problemen in der Architektur führen

Beispiel Best Proactice: **8 Design Principles**(Tabelle 1)

Einfachheit der Schutzmaßnahmen	Komplexität ≡ Fehleranfälligkeit, Aufteilen eines Systems in kleinere leichter testbare Teile, Sicherheitsmaßnahmen einfach und überschaubar, damit Fehler gut erkannt werden.
Fail Safe Defaults	ein System soll standardmäßig im sicheren Zustand sein, damit im Fehlerfall darauf zurückgegriffen werden kann Bsp: standardmäßig Zugriffe ablehnen
vollständige Berechtigungsprüfung	Prüfung sämtlicher Zugriffe und auf invalide Zugriffe entsprechend reagieren
Offenes Design	Sicherheit darf nicht auf der Geheimhaltung des Designs oder der Implementierung beruhen
4 Augen Prinzip	Informationen / Aktivitäten sollen durch mehrere Parteien geschützt sein, Bsp: mehrere Schlüssel
Least Priviledge	Benutzer sollen nur die erforderlichen Rechte besitzen um Risiken zu minimieren
Minimum an gemeinsamen Mechanismen	Komponenten sollen unabhängig voneinander agieren können, → keine gemeinsamen Dateien, Variablen, etc
Psychologische Akzeptanz	Personen die diese Sicherheitsmechanismen anwenden müssen auch von ihnen akzeptiert werden → sollen angenehm benutzbar sein (UI. . .)

Table 1: 8 Design Prinziples

4.2.3 Implementierung

Eine an sich sichere Architektur kann durch einen Implementierungsfehler unsicher werden. Daher sollte man sich an mehreren Guides orientieren (Tabelle 2)

Auflistung von häufigen Sicherheitsfehlern	verwenden von OWASP oder CWE um bekannte Sicherheitsschwachstellen einzusehen
Programmierrichtlinien	es existieren für viele Sprachen Secure Coding Guidelines die häufige Programmierfehler vorbeugen
Sicherheitskonzepte von Programmiersprachen	viele Sprachen bieten automatisierte Tools um Sicherheitsprobleme vorzubeugen. Beispiele: Längenprüfung und Speicherverwaltung. Wichtig ist auch zu wissen wenn eine Sprache das nicht unterstützt
Code Review	andere Personen als die Entwickler sehen sich den Code an und prüfen Ihn auf Vorgaben, Best Practices und Sicherheitsfehler

Table 2: Guides für Implementierungen

Viele Unternehmen halten sich an dem Ansatz *Penetrate and Path* fest. Dabei werden Sicherheitsfehler die im Betrieb gefunden und gemeldet werden durch Live Patches beseitigt und sonst keine Sicherheitstests durchgeführt. Das ist nicht ausreichend, da nicht gemeldete Fehler nicht ausgebessert werden und ein Patch meist nur den Effekt eines Fehlers ausbessert und nicht das eigentliche Problem. Außerdem können Angreifer aus den Patches Angriffe ableiten und diese auf nicht gepatchte Systeme anwenden.

Ein häufiges Problem auf Implementierungsebene ist die **SQL Injection** die aus der Übung bekannt ist. Dabei kann ein Angreifer versuchen durch *Hochkommas, Tautologien* oder *Union* Befehlen Daten aus den Tabellen ziehen oder Login Masken zu umgehen. Ähnlich funktioniert eine Command Injection. Verhindert werden kann so etwas durch Eingabenvalidierung mittels black- (nicht empfohlen) oder whitelisting von Zeichen, verwenden von Funktionen der Sprache und Risikominimierung durch Least Priviledge.

Eine andere Attacke durch fehlende Validierung ist **Corss Site Scripting**. Sie kann auf Webanwendungen angewendet werden und ist für Nutzer unsichtbar. Durch den Angriff kann etwa eine Session gehijacked werden oder Nutzer werden auf andere Seiten weitergeleitet. Zwei Arten (Tabelle 3) Verhindert werden kann XSS durch Eingabenvalidierung, Kodierung von Benutzereingaben oder Firewalls die verhindern, dass Antworten nicht an die ursprüngliche Domain weitergeleitet werden.

Cross Site Request Forgery bewirkt, dass durch eine präparierte URL ein Opfer zur Durchführung einer Aktion in einer Webanwendung missbraucht wer-

Reflected XSS	1. senden einer präparierten URL an das Opfer 2. Opfer öffnet den Link 3. Server liefert die reguläre Seite mit den Inhalten aus der URL 4. Browser wertet die Rückgabe aus und Daten werden an den Angreifer übermittelt
Stored XSS	1. für einen späteren Angriff wird ein präparierter Inhalt auf einer Website hinzugefügt und vom Server gespeichert 2. Opfer folgt dem Link im Browser 3. Server liefert die reguläre Seite mit den Inhalten aus der URL 4. Browser wertet die Rückgabe aus und Daten werden an den Angreifer übermittelt

Table 3: Arten von XSS

den kann. z.B Ausloggen oder sogar ändern des Passworts. Der Angreifer kann daher im Namen eines Nutzers mit dem Server agieren. Verhindern kann man CSRF durch Shared Secrets. Das ist ein vertraulicher Token den nur Client und Server kennen und zu internen URLs und Formularen hinzugefügt wird. Bei jeder Anfrage wird dabei das Token übermittelt und geprüft. Bei unverschlüsselten Verbindungen ist dieser Mechanismus allerdings nutzlos, da das Token gestohlen werden kann. Eine andere Methode sind nötige Bestätigungen für Aktionen wie etwa "wollen sie wirklich das Passwort ändern?"

Buffer Overflows treten auf wenn keine oder eine falsche Längenüberprüfung beim Schreiben in einen Buffer durchgeführt wird. Dadurch können die folgenden Punkte eintreten:

- Absturz
- ungewöhnliches Systemverhalten
- Ausführung von eingeschleustem Code
- nichts, falls der Speicherbereich nicht weiter benutzt wird

Sie treten meist in Low Level Sprachen auf und können verhindert werden durch

- Eingabevalidierung und Längenprüfung
- Verwendung von Sprachen die das automatisch machen
- Vermeidung von Funktionen die auf die Korrektheit der Eingabe vertrauen

4.2.4 Testen

Weiteres in dem kommenden Kapitel

4.2.5 Betrieb

Im Betrieb zeigt sich ob die Sicherheitsanforderungen erfüllt und richtig umgesetzt worden sind. Es ist wichtig de Sicherheitsanforderungen während des Betriebs zu erfüllen. Dazu gibt es oft Service Level Agreements, die eine Definition und Verfügbarkeitsanordnungen beinhalten. Zusätzlich sollte ein System gemonitored werden um Sicherheitsprobleme während des Betriebs zu entdecken. Auch müssen Reaktionen auf Sicherheitsvorfälle im Vorfeld definiert werden (Wer wird verständigt bei Vorfällen, wie an Feiertagen, etc). Auch dazu gibt es wieder Best Practices wie etwa die ITIL.

5 Testing

Ein System ist niemals "fertig" getestet da Software meist kontinuierlich weiterentwickelt wird, insbesondere in Hinsicht auf Security. Selbst wenn ein System

zu einem gewissen Zeitpunkt eine sehr gute Testabdeckung hat, sind durchschnittlich 5 bis 15 Bugs in 1000 Zeilen Programmcode enthalten. Dabei handelt es sich meist um solche die nur durch gezielte Suche entdeckt werden können und Sicherheitsverletzungen darstellen. Diese Fehler treten auf, da auch beim Testen Vorbedingungen gelten (Konfiguration, Umgebung, etc). Bei der kleinsten Änderung dieser könnten schon neue Fehler aufgedeckt werden. Daher ist es nie absolut sicher, dass ein System fehlerfrei ist. Oft ist es auch Zeitdruck oder Kosten (Kosten Nutzen Kurve) die Beschränkungen auf die Testkapazitäten auflegen.

5.1 Was bedeutet Testen

Ein System in einen gewissen Zustand versetzen (durch Tätigkeiten, Interaktionen) und anschließend mit einem erwarteten Zustand verglichen.

Es gibt dabei zwei Kategorien:

- Validierung: entspricht ein Produkt den gewünschten Anforderungen → erzeugen wir das richtige für den Kunden
- Verifikation: erfüllt das Produkt die gewünschten Anforderungen → erzeugen wir es richtig für den Kunden

Wichtig ist auch der Begriff der *Testabdeckung*: abgedeckte Codezeilen, jeder Pfad im Programm....

5.2 Testarten

In der Praxis zwei verschiedene Personenkreise:

Entwickler / funktionaler Tester

Entwicklern fehlt oft notwendiges Wissen bzw Erfahrung im Security Bereich und führen die Tests stark auf die Funktionalität und Qualität fokussiert durch. Wenn solche Personen Sicherheitstests durchführen führt das dann oft dazu, dass Schwachstellen nicht erkannt werden.

Sicherheitsexperten

Sicherheitsexperten fehlt oft das spezifische Wissen zum Kontext eines Testobjekts oder Implementierungsdetails. Sie haben einen guten Überblick über die Bedrohungslage ähnlicher Systeme und können daher den Sicherheitsprozess gut optimieren und strukturieren. Die Programme werden auch aus Angreifersicht getestet und in einem besonderen Mindset, sodass auch schwer auffindbare Schwachstellen gefunden werden.

5.3 Merkmalsräume

Stellen eine Charakterisierung der Testarten dar Tabelle 4

Prüfkriterium	Inhaltliche Orientierung des Testfalls: **Was** wird geprüft
Prüfebene	Durchführungszeitpunkt jedes Testfalls: **Wann** wird geprüft
Prüfmethodik	Durchführung des Testfalls: **Wie** wird getestet

Table 4: Merkmalsräume

5.3.1 Prüfkriterium

Das **Prüfkriterium** wird nochmals unterteilt in Tabelle 5

funktionale Sicherheitstests	**nicht funktionale Sicherheitstests**
• Spezifikation als Grundlage • Tests sind in sich geschlossen und geben eine eindeutige Antworten auf die Testfrage • Funktionale Fehler deuten auf nicht funktionale Eigenschaften der Sicherheit hin • zur *Verifikation* eingesetzt	• Spezifikation nicht als alleinige Grundlage • Analog zu Use Cases werden Abuse Cases eingebaut (aus der Sicht des Angreifers) • Testfälle greifen ineinander und beeinflussen sich gegenseitig • Fragen beschäftigen sich mit Robustheit der einzelnen Komponenten

Table 5: funktionale / nicht funktionale Tests

5.3.2 Prüfebene

Auch die Prüfebene wird unterteilt in die verschiedenen Projektphasen. Jede von diesen Phasen hat bestimmte Eigenschaften und Prüfobjekte. So wird in einer frühen Phase ein Penetrationstest wenig Sinn machen. Trotzdem sollte mit

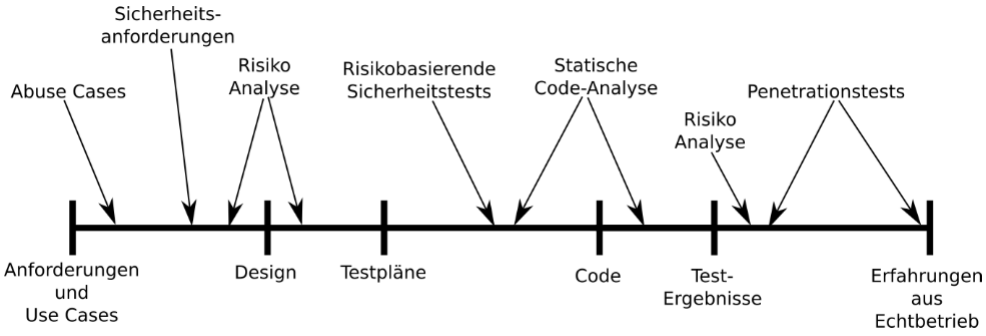

Figure 7: Tests und wann sie durchzuführen sind

dem Testen möglichst früh begonnen werden. Das ist billiger und zeigt früher Fehler auf. Wann welche Tests durchgeführt werden ist in Bild 7 zu sehen. Noch eine Auflistung:

- Abuse cases - nach der Erstellung der Anforderungen
- Sicherheitsanforderungen - folgen aus Abuse cases
- Risikoanalyse - wird ab der Feststellung der Anforderungen in jedem Schritt weiter iterativ aufgebaut
- Risikobasierte Sicherheitstests - werden nach der Risikoanalyse konzipiert und ausgeführt
- statische Codeanalyse - sobald Code vorhanden ist
- Penetrationstests - sobald lauffähige Komponenten vorhanden sind möglich

5.3.3 Prüfmethodik

Unterscheidung von den 3 Testarten:

- White Box
- Black Box
- Grey Box

White Box

Der Tester bekommt umfassende Informationen über das System aus denen er Inputmuster gewinnen kann die erfolgversprechende sind. Diese Testart ist sehr effizient aber nicht realitätsgetreu.

Black Box

Der Tester hat keine Informationen zu dem Zielsystem (realitätsnah). Der Tester versucht dabei durch manipulierte Input Werte falsches Verhalten zu erzwingen. Dazu werden sogenannte *Äquivalenzklassen* erstellt. Oft werden *Grenzwertanalysen* durchgeführt die meist sehr realitätsnah sind.

Grey Box

Eine Kombination aus White und Black Box verbindet die Vorteile beider Modelle allerdings abgeschwächt.

5.3.4 Testtechniken

statische Codeanalyse

Untersuchung des Codes ohne ihn auszuführen. Das kann manuell oder automatisiert erfolgen. Das *Secure Code Review* ist der manuelle Vorgang, Code in Hinblick auch Sicherheit und Fehler zu prüfen. Sie ist meist sehr langsam und kostenintensiv, allerdings auch sehr effektiv und es können viele logische Fehler gefunden werden. Durchgeführt werden kann diese

- Top Down: detailliertes Wissen über aktuelle Schwachstellen sind erforderlich. Wissen über den Code wird im Rahmen des Reviews aufgebaut.
- Bottom Up: detailliertes Wissen über den Code und bekannte Schwachstellen ist vorhanden und es wird versucht mit dadurch kritische Stellen auszumachen
- Walkthrough: Testfälle werden nur am Papier durchgeführt um potentielle Informationsflüsse und Berechnungen nachvollziehen zu können (meist nur Teilkomponenten)
- Inspection: Überprüfung von Codeschnippsel durch Checklisten mit den Coding Standards

automatisierte statische Codeanalyse mittels Compiler

Meist stark vom Tool abhängig aber der Ablauf ist meist gleich:

1. Code wird dem Tool als Input zugewiesen in Form eines Modells
2. Mittels Datenbank mit Sicherheitswissen wird eine Analyse gestartet
3. Gefundene Violations werden ausgegeben

Es gibt mehrere Techniken:

- Signaturbasiert: Aufspüren von Mustern im Code

- Kontrollflussanalyse: Kontrollfluss wird analysiert und potentielle fehlerhafte Flüsse werden erkannt
- Datenflussanalyse: Nachvollziehen aus welchen Quellen Daten kommen und welche Verbraucher diese verwenden. Daten die aus nicht vertrauenswürdigen Quellen stammen können so nachvollzogen werden.

Fuzzing

Beballern eines Systems mit Random Werten. Die Tests können unterschieden werden:

- Input-Erstellungstechnik
  - Generation Based Fuzzer: Input in jeder Iteration neu generiert
  - Mutation Based Fuzzer: ableiten der neuen Inputs aus den alten
- Smartness: Unterscheidung des Targets
  - Random Fuzzer: einfachste Form, gesamte Bereiche eines Ziels werden gefuzzt
  - Template Fuzzer: in dem Template sind die zu fuzzenden Bereiche markiert
  - Block Fuzzer: zu fuzzender Block wird ausgewählt
  - Evaluation based Fuzzer: Fuzzer lernt aufgrund des Protokolls und Outputs wiederholte Anfragen und passt sein Vorgehen an

Fuzzing müssen meist massenhaft durchgeführt werden um einen großen Wertebereich abzudecken. Fehler können erkannt werden an Veränderungen im System als da wäre: Prozessorleistung, Log Größe oder andere Parameter.

Penetrationstest Hierbei schlüpft der Tester in die Rolle des Angreifers um ein System auf Schwachstellen zu prüfen. Dafür werden möglichst reale Bedingungen erstellt (gute Kopie des Originalsystems) und das System auf allen OSI Layern getestet. Auch Social Hacking wird manchmal angewandt. Es ist besonders wichtig bei den Tests kreativ und vor allem rechtlich abgesichert zu sein. Dazu gibt es klare Rules of Engagement die abklären ob ein Auftraggeber berechtigt ist einen solchen Test durchführen zu lassen, ob man ein System abstürzen lassen darf und so weiter. Darauf aufbauend muss es für jeden Penetrationstest eine Permission to Attack geben vom Auftraggeber. Beispiele für Rahmenbedingungen sind:

- Was ist das Ziel des Tests
- Welche Komponenten werden getestet
- Welche Typen von Angriffen werden in Betracht gezogen
- Abbruchbedingungen für Test

- Informationsbasis der Tester
- verwendete Methodik und Techniken

Penetrationstests sind meist sehr detailliert auf ein Zielsystem zugeschnitten. Um einen Penetrationstest richtig zu planen gibt es mehrere Phasen:

- Pre Engagement - Planung
- Intelligence Gathering
- Threat Modelling - Bedrohungsanalyse
- Vulnerability Analysis - Schwachstellenanalyse
- Exploitation - aktive Ausnutzungsphase
- Post Exploitation - Auswirkungsanalyse der Schwachstelle
- Reporting - Berichterstattung

Bei Penetrationstests ohne einen Auftrag kann bei einer Veröffentlichung 2 Vorgehensweisen unterschieden werden:

- Responsible Disclosure: Schwachstelle wird zuerst an den Hersteller gemeldet um diesem Zeit für einen Fix zu geben
- Full Disclosure: direkt alle Einzelheiten werden veröffentlicht und zwingt den Hersteller zu einer schnellen Handlung

6 Identität Authentifizierung und Autorisierung

6.1 Definitionen

Identität: wer jemand ist

Authentisierung: Vorlage eines Identitätsnachweises

Authentifizierung: Überprüfen eines Identitätsnachweises → gibt es diese Person

Autorisierung: Überprüfung ob eine Person zu einer Aktion berechtigt ist

Zutritt: Betreten abgegrenzter Bereiche

Zugang: Nutzung von IT Systemen

Zugriff: Nutzung von Informationen auf einem IT System

Objekt: zu schützender Informationsträger

Subjekt: aktives Element die Zugriffe ausführt

Zugriffsoperation: Art um auf ein Objekt zuzugreifen

Zugriffsrecht: Recht des Zugriffs auf ein Objekt

Schutzdomäne: Gruppierung von identischen Zugriffsrechten

Zugriffsmonitor: setzt die Zugriffskontrollstrategie durch

6.2 Authentisierung

Eine Kontrolle des Zutritts / Zugangs oder Zugriffs ist erst sinnvoll, wenn bekannt ist wer das haben will. Dabei ist es wichtig, dass Berechtigte einfach Zugriff erlangen können aber Unberechtigte zuverlässig abgewiesen werden. Authentisierung kann mittels Wissen (Passwort), Besitz (Keycard) oder biometrische Merkmale passieren.

Passwörter sind einfach zu implementieren, daher günstig und bieten hohe Flexibilität (leicht zu ändern). Allerdings ist ein Verlust meist schwer oder zu spät erkennbar, da ein Passwort auch von mehreren Personen mehr oder weniger unerkannt benutzt werden kann. Passwörter sollten einfach zu merken sein und gleichzeitig schwer knackbar sein. Komplexität kann durch eine umfangreiche Grundmenge an möglichen Zeichen gegeben sein. Gleichzeitig ist es wichtig ein Login Feld von Eingaben mit nicht erlaubten Zeichen zu schützen. Auch kann Komplexität erreicht werden durch eine ausreichende Länge der Passwörter. Default Passwörter sollten vermieden und nicht irgendwo notiert werden. Die Anzahl kann der Eingaben sollte beschränkt werden um Brute Force keine Chance zu lassen. Es muss auch einen Mechanismus geben, wenn jemand sein Passwort vergisst.

Angriffe auf Passwörter sind etwa Raten oder Bruteforce mit Wortlisten oder einfach allen möglichen Kombinationen. Auch können Rainbowtables verwendet werden die Passwörter mit dazugehörigem Hash speichern. Das geht aber nur, wenn die Passwörter keinen Salt besitzen (zufällig generierter String der für den Hash angehängt wird). Social Engineering ist eine weitere Möglichkeit (Erpressung...). Passwörter sind auch oft in Script files gespeichert direkt auf Websiten. Sniffing kann auch verwendet werden um eine Passwortübertragung aufzuzeichnen und anschließend das PW zu cracken.

Schützen kann man sich durch sicheres Speichern der Passwörter als Hashes mit Salt und nicht alle Benutzer sollen auf die PW Datei Zugriff haben. Außerdem können PW Checker verwendet werden die PW auf Güte prüfen. In Firmen sollten Mitarbeiter regelmäßige Schulungen besuchen. Eine andere Möglichkeit ist, Lookout Mechanismen zu verwenden, sodass etwa nach 3-maliger falscher Eingabe eine Zeit vergehen muss, bis ein neuer Versuch gestartet werden kann.

Ein PW sollte sofort geändert werden wenn der Verdacht besteht, dass ein Passwort gestohlen wurde. Informationen zu (Fehl)Anmeldungen helfen Benutzern zu erkennen ob jemand anderer das PW verwendet. Die Verwendung eines PW-Safes wird empfohlen.

Besitz kann auch als Authentisierung verwendet werden. Dazu muss ein physischer Gegenstand vorhanden sein. Dieser sollte angemessen gegen Diebstahl gesichert werden und nicht einfach zu kopieren sein.

Chipkarten sind das klassische Beispiel. Auf diesen werden kryptographische Schlüssel sicher gespeichert und Operationen werden direkt auf ihnen ausgeführt. Manchmal werden kryptographische Operationen auch aus Gründen der Geschwindigkeit außerhalb der Karte ausgeführt. Der Schlüssel wird unter Vertraulichkeit aufgebracht und wenn die Karte entsorgt werden soll unter derselben gelöscht werden. Wichtig ist auch, dass keine Pin Codes auf diesen notiert werden sollten.

Mögliche Angriffe auf Keycards sind etwa Social Engeneering, Man in the Middle oder etwa Messung von Stromverbrauch oder Rechenzeit um Rückschlüsse auf den privaten Schlüssel zu ziehen. Außerdem kann die Karte physikalisch über die Temperatur und die Spannung beeinflusst werden.

Die letzte Möglichkeit ist die **biometrische Erkennung**. Diese sind kaum ersetzbar aber die Weitergabe ist im Allgemeinen nicht möglich. Biometrische Erkennung ermöglicht sowohl die Identifizierung als auch die Validierung von Personen. Ein Nachteil sind die hohen Kosten.

Ein Thema ist einerseits die False Acceptance Rate und andererseits die False Rejection Rate. Je geringer die FRR ist desto höher ist die FAR. Dabei ist zu bedenken, dass verschiedene Menschen verschieden gut auf diese Systeme ansprechen

Mögliche Angriffe sind etwa Spoofing durch hochauflösende Fotos von Fingerabdrücken oder Gesichtern. Wenn ein Angreifer auf die Strecke zwischen Sensor und Backend zugreifen kann sind Replay Attacken wenn der Traffic mitgeschnitten wird. Sensoren melden manchmal nur erfolgreich oder nicht erfolgreich. Das kann verwendet werden, um diese Nachricht einfach an das Backendsystem weiterzuleiten. Brute Force geht auch, wenn man genug Menschen dazu bewegen kann ihren Finger draufzuhalten.

6.3 Zugriff

Zugriff auf Ressourcen sollten nur Berechtigte erhalten. Unberechtigte Zugriffe sollten verhindert oder aufgezeichnet werden. Es sollte auch regelmäßig geprüft werden, ob Berechtigungen noch aktuell sind oder bereits entzogen worden sind.

Zugriffskontrollen können etwa auf Betriebssystemen, Datenbanken oder Webservern durchgeführt werden. Die Kernfrage dabei ist, wer auf was zugreifen

darf. Dazu gibt es eine **Zugriffskontrollstrategie** die besagt was erlaubt ist und was nicht. Weiters gibt es ein **Zugriffskontrollmodell** welches die Strategie formal beschreibt. Ein Modell sollte möglichst einfach, ausdrucksstark und leicht zu warten sein.

Zugriffsrechte können in einer Zugriffsmatrix (Benutzer x Objekte) definiert werden. Das ist aber langsam und ineffizient. Daher werden Subjekte zu Rechtegruppen zusammengefasst. Alle Subjekte in einer solchen Gruppe haben die gleichen Zugriffsrechte und können auch leicht verschiedenen Gruppen zugeordnet werden

6.3.1 Zugriffskontrollmodelle

Discretionaey Access Control

In diesem Modell legt der Besitzer eines Objekts die Berechtigungen selbst fest. Die Zugriffe auf dieses sind rein von der Identität abhängig und der Besitzer kann Rechte an andere User vergeben. Die Verwaltung ist dabei aber ziemlich aufwendig. So müssen etwa für neue Mitarbeiter alle Rechte für jedes Objekt neu vergeben werden. Auch eine Prüfung welche Benutzer welche Rechte haben ist ziemlich aufwendig.

Role Based Access Control

Rechte werden über Gruppen verwaltet. Jeder User bekommt eine Rolle zugewiesen über diese der Zugriff auf Objekte festgelegt ist. Dabei können Benutzer mehrere Rollen besitzen, auch Vererbung in einem hierarchischen System ist möglich. Manche Rollen können auch andere ausschließen. Dieses Konzept nennt man *Separation of Duty*

Mandatory Access Control

Für alle Benutzer werden zentral die Zugriffsrechte von Benutzern festgelegt. Dabei geht es vor allem um den Informationsfluss. Objekte und Benutzer erhalten intern Einstufungen wie öffentlich, vertraulich oder streng geheim. Nutzer können dabei auf ein Objekt zugreifen, wenn ihre Einstufung mindestens der des Objekts entspricht. Die Anwendung dieses Modells liegt hauptsächlich im militärischen Bereich.

7 Organisatorische Sicherheit und Sicherheitsmanagement

Die primäre Aufgabe des IT Managements ist die der Sicherheitsplanung. Dazu müssen Richtlinien definiert werden. Diese müssen mit Unterstützung der Geschäftsführung umgesetzt werden. Dabei sind folgende Aufgabenstellungen wesentlich:

- Eindeutige Definition von Verantwortlichkeiten inc Vertretungsregelungen und Kommunikationswegen
- IT Benutzer müssen regelmäßig zum Thema Sicherheit geschult werden
- Sicherheitsmaßnahmen und Änderungen müssen dokumentiert werden
- Regelmäßige Prüfungen ob Sicherheitsmaßnahmen eingehalten werden
- Sicherheit sollte als fortlaufender Prozess verstanden und gelebt werden

7.1 Schulungen von Mitarbeitern

Mitarbeiter müssen eine ausreichende Einführung und Schulung bekommen. Sie sollen verstehen warum IT Sicherheit wichtig ist und wie sie dazu beitragen können. In Schulungen sollen der sichere Umgang mit IT Systemen und Daten erläutert werden. Diese sollen regelmäßig durchgeführt werden, da Mitarbeiter die Inhalte schnell vergessen.

7.2 Backups

Datenverlust kann zu erheblichen Schäden führen. Daher ist eine aktive Datensicherung erforderlich. Dazu gibt es einen Backupplan in dem definiert ist was wie oft gesichert wird. Auch in den Backups müssen Vertraulichkeit, Integrität und Authentizität sichergestellt werden. Mit steigender Wichtigkeit der Daten steigt auch die Wichtigkeit der Sicherungen. Es muss auch regelmäßig getestet werden, ob Daten korrekt gesichert werden und der Restore funktioniert.

7.3 Sensible Daten

Wenn sensible Daten (Schlüssel, personenbezogen, ...) gespeichert werden, müssen diese nach der Verwendung sicher gelöscht werden. Das kann direkt nach dem Gebrauch passieren, aber auch viel länger. Es kommt immer wieder vor, dass ausrangierte Festplatten mit unverschlüsselten internen Daten entsorgt werden. Das reine Löschen dieser Daten ist dabei meist nicht ausreichend, da diese oft wiederherstellbar sind (nur Verzeichnisseintrag gelöscht) Daher muss spezielle Hard und Software verwendet werden um Festplatten zu zerstören. Dazu sollten externe spezialisierte Unternehmen konsultiert werden.

7.4 Sicherheitsprozess

Da sich Bedrohungen laufend ändern, müssen die Sicherheitsmaßnahmen hinterherziehen. Damit die Sicherheit sichergestellt ist muss Organisatorisch sichergestellt werden, dass neue Angriffe erkannt bewertet und auf sie entsprechend reagiert wird. Sind technische oder organisatorische Änderungen nötig sollen diese zeitnah umsetzbar sein.

Es muss also Prozesse geben die sicherstellen, dass auf neue Risiken entsprechend reagiert wird

Um die Prozesse richtig umzusetzen ist es wichtig zu wissen, welche Assets wo, von wem und für was eingesetzt werden. Dazu ist die akkurate Dokumentation wichtig. Diese sollte folgende Punkte umfassen:

- Welche Sicherheitsmaßnahmen gelten für die Assets
- Wie sind sie konfiguriert (schnelles wiederaufsetzen nach Ausfall)
- Servicenummern der IT Lieferanten und Dienstleister (schnelle Supportkontaktierung)
- Vertragliche Vereinbarungen mit Kunden bzw Lieferanten, SLAs
- On und Offboarding von Mitarbeitern
- Wer welche Berechtigung haben darf und braucht
- Überprüfungen der Einhaltung und Wirksamkeit von Maßnahmen (auch durch externe testbar - Penetrationstests)

Die Dokumentation wird im Idealfall regelmäßig kontrolliert und (meist automatisch) aktualisiert.

7.5 Sicherheitsmanagementprozess

Der erste Schritt ist die Zuteilung von Sicherheitsverantwortlichen-Rollen. Dabei ist klar festgelegt, wer welche Aufgaben (Sicherheit) hat. Die wichtigste Rolle ist die des **Chief Information Security Officer - CISO** die durch das Projektmanagement zugeteilt wird. Weitere Rollen sind etwa:

- Information Security Manager (Koordination und Ansprechpartner)
- Information Risk Manager (Erkennung, Bewertung und Management von Risiken)
- IT Sicherheitsbeauftragter (Umsetzung: technisch und organisatorisch)
- Datenschutzbeauftragter

7.6 Security Policies

Helfen bei der Umsetzung von IT Sicherheit in eiem Unternehmen und muss von der Geschäftsführung unterstützt und gelebt werden. Diese legen technische und Organisatorische Maßnahmen, Verhaltensrichtlinien und Verantwortlichkeiten fest. Enthalten sind auch

- angestrebten Sicherheitsziele
- verfolgte Sicherheitsstrategie

Anforderungen

Sie sollten vollständig aber verständlich und nicht zu detailliert sein, da Mitarbeiter diese sonst nicht lesen und unterschreiben ohne zu wissen, um was es geht. Sie sollen so gestaltet werden, dass Mitarbeiter sie annehmen und befolgen. Damit dies sichergestellt ist sollte regelmäßig geprüft werden ob sie richtig angewendet werden, wenn nicht wieso und anschließend überarbeitet werden.

Oftmals verhalten sich Mitarbeiter unbeabsichtig falsch und daher enthalten Policies Beschreibungen wie sicher und korrekt mit der IT Umgegangen wird.

Mögliche Policies sind:

- Datenschutz
- Anlage / Deaktivierung von Accounts
- Verwendung von USB Sticks
- ...

7.7 Plan-Do-Check-Act

Beschreibt ein Modell für einen Sicherheitsmanagementprozess.

Planung

In dieser Phase findet die Planung und Installation eines Sicherheitsmanagements anhand der Risiko, Sicherheits und Unternehmenspolitik sowie betrieblichen und gesetzlichen Anforderungen statt. Es wird ein Vorgehensmodell für die Sicherheits und Risikopolitik definiert. Dazu werden die Sicherheitsanforderungen erhoben und anschließend entsprechende Richtlinien erstellt. Anhand dieser werden dann Sicherheitskonzepte entwickelt und Maßnahmen geplant.

Do

Diese Phase beschreibt den Betrieb unter Einhaltung der Richtlinien und das Monitoring des Sicherheitsmanagements. Hier erfolgen Schulungen und Messungen und Überwachung um Sicherheitsvorfälle zu erkennen. Es werden auch Daten gesammelt um Ist-Soll Vergleiche durchzuführen.

Check

Hierbei wird das Sicherheitsmanagement regelmäßig auf Wirksamkeit und Verbesserungspotential geprüft. Möglichkeiten dafür sind:

- Tests
- Audits
- Übungen
- Sicherheitsprüfungen

Der Vorgang wird dokumentiert, analysiert und bewertet. Die Ergebnisse werden berichtet und dienen als Grundlage für die nächste Phase.

Act

Beschreibt die Verbesserung die aufgrund der Ergebnisse der Check Phase gewonnen wurde, sowie neue Erkenntnisse und Gesetze. Das Management priorisiert Bedarfe und erstellt eine Roadmap zur Verbesserung. Dann beginnt der Kreis von neuem.

7.8 Schaffen von Vertrauen

Geschieht meist durch Einhaltung von Normen und Standards. Diese sind auch oft gefordert für verschiedene Aufträge. Normen / Standards können etwa nach einzelenen Aspekten oder Gruppierungen dieser geprüft werden:

- Betriebsprozesse
- Entwicklungsprozesse
- Fertigungsprozesse
- technische Maßnahmen

7.8.1 ISO 27001

Der Standard im IT Sicherheitsbereich für Sicherheitsmanagement.

Die Norm beschreibt Anforderungen an Informationssicherheitsmanagementsysteme (ISMS). Das ist eine Sammlung von Prozessen und Sicherheitsmaßnahmen um IT Sicherheit zu erreichen.

Es wird dabei auf die Errichtung, Umsetzung, Betrieb, Überwachung, Überprüfung, Aufrechterhaltung und Verbesserung eines ISMS eingegangen.

Sie ist anwendbar für Organisationen jeglicher Art, Größe und Ausprägung bzw Teilen davon.

Die Einführung eines ISMS ist ohne Vorarbeit sehr aufwendig. Dabei müssen nämlich Aspekte wie im Folgenden berücksichtigt werden.

- Umfang
- Prozeduren und Maßnahmen zur Pflege eine ISMS
- Methoden zur Risikobewertung
- Dokumentation
- ...

ITIL baut auf dieser Norm auf und ist eine Sammlung an Best Practices aus Vorgehen, Funktionen und Rollen. Es ist Prozessbasiert, dient also dazu den operativen Betrieb zu optimieren.

7.8.2 Baukastenmodell

Das BSI hat ein Modell entwickelt, den **IT Grundschutz**, das Bausteine, Gefährdungen unf Maßnahmen beinhaltet. Diese können nach Anleitung zusammen und umgesetzt werden. Der Vorteil dabei ist, dass kleinere bzw mittelgroße Projekte ohne Sicherheitsabteilung dadurch leicht ein annehmbares Schutzniveau erhalten.

7.9 Sicherheitskonzepte

Die Inhalte eines Sicherheitskonzepts sind Maßnahmen zur Realisierung und Aufrechterhaltung eines Sicherheitsniveaus für ein Projekt / Unternehmen. Dazu werden folgende Fragen beantwortet:

- Was will ich schützen
- Wogegen soll ich mich schützen
- Wie kann ich mich schützen
- Kann ich mir den Schutz leisten

7.9.1 BSI-Standard 100-2

Beschreibt Schritte zur Erstellung eines Sicherheitskonzepts nach IT Grundschutz. Der Ausgangspunkt ist eine IT Infrastruktur:

- Organisation
- Infrastruktur
- IT Systeme
- Anwendungen
- Mitarbeiter

Anschließend werden die folgenden Schritte durchgeführt:

1. Strukturanalyse
2. Feststellen des Schutzbedarfs für die festgestellten Assets
3. Modellierung: Auswählen von Sicherheitsmaßnahmen - Soll Ist Vergleich: welche Maßnahmen sind bereits vorhanden und welche braucht es noch
4. Umsetzung der Maßnahmen
5. Sicherheitsanalyse: für welche Assets braucht es eine Risikoanalyse weil deren Schutzbedarf höher ist.
6. Anwenden der Standard und erweiterten Sicherheitsmaßnahmen für geprüfte Assets

7.9.2 Business Continuity

Wenn das IT System über einen gewissen Zeitraum ausfällt führt das zu Verlusten wie etwa weil:

- Mitarbeiter können nicht arbeiten
- Dienste sind nicht verfügbar
- Fuhrpark steht still

Um dem Vorzubeugen ist ein Plan zur möglichst schnellen Wiederherstellung des Normalzustands nötig. Das heißt dann **Business Continuity**.

8 Sicherheitsfaktor Mensch

Technische Sicherheitsmechanismen können leicht umgangen werden wenn Social Engineering Methoden angewandt werden. Darunter wird die Kunst verstanden Menschen gegen den eigenen Willen zu beeinflussen.

8.1 Social Engineer

In der Regel ist eine solche Person nicht an Zerstörung interessiert, sondern am unberechtigten Zugang zu Informationen. Er greift nicht Computersysteme an, sonderen die Menschen die diese bedienen. Meist sind sie gewandte Redner, kann gut auf Menschen eingehen und tritt überzeugend auf. Er kann sich Informationen auf vielfältige Weise beschaffen und weist hohe Geduld auf, da seine Angriffe auch mehrere Jahre dauern können. Charaktereigenschaften sind:

- schnelle und spontane Anpassung an extreme Veränderungen
- Ablehnung von Routine und Tradition
- Tendenz von alten Sachen gelangweilt zu sein

- Verlangen neues zu erfahren und zu erreichen.

Er nutzt Konstanten menschlichen Verhaltens zur Beeinflussung aus. So etwa **Reziprozität** das das Gefühl Gefälligkeiten zurückzahlen zu müssen, beschreibt.

Auch **Konsitenz** kann genutzt werden. Das bedeutet, das Menschen die Entscheidungen getroffen haben, ungern später entgegensätzliche Entscheidungen treffen. Der Social Engineer verwicklet sein Opfer in eine Vielzahl von Entscheidungen die subsequent immer größere Tragweite haben. Dadurch ist es für ein Opfer sehr schwer Anfragen abzulehen.

Eine andere Möglichkeit ist die **soziale Bewährtheit**. Dazu wird ein Opfer in eine stressige Entscheidungssituation versetzt und die Information mitgeteilt, wie sich seine Kollegen angeblich verhalten haben.

Sympathie ist auch eine entscheidende Waffe, da sympathischen Menschen wahrscheinlicher geholfen wird. Da Sympathie durch einfaches zeit miteinander verbringen entstehen kann (Ähnlichkeiten finden) ist sie sogar relativ leicht anzuwenden.

Weiters kann **Autorität** einen Durchbruch für einen Angreifer bedeuten. Ein gutes Beispiel ist der Arzt der seine Patienten nicht ausreden lässt. Der Social Engineer nimmt für einen Angriff eine entsprechend autoritäre Rolle ein und nutzt den Umstand, dass Entscheidungen und Weisungen von Personen höherer Autorität weniger in Frage gestellt werden.

Schlussendlich kann er das Prinzip der **Knappheit** nutzen. Er setzt das Opfer unter Druck indem er für die Entscheidungsfindung nur sehr wenig Zeit zulässt und die negativen Konsequenzen bei einer falschen Entscheidung betont. Alternativen die durch den Angreifer aufgezeigt werden, wirken nun wesentlich attraktiver.

8.2 Der Angriff

Der Angriff besteht aus vier Phasen

8.2.1 Informationsbeschaffung

Der Angreifer verwendet unterschiedliche Techniken um Informationen über sein Ziel zu beschaffen. Das können sowohl öffentliche Informationen (Social Media, etc) als auch Detailinformationen sein. Detailinformationen sind vor allem Information über die Mitarbeiter deren privates Leben, Hobbies usw. Diese Informationen können leicht über soziale Medien recherchiert werden.

Während dieser Phase eignet sich der Angreifer auch ein geeignetes Fachvokabular an um anschliesend glaubhaft eine Rolle einnehmen zu können (siehe Catch me if you can). Weiters muss er die Unternehmenskultur kennen und begreifen. Wenn er viele Vorgänge einer Organisation kennt, kann sich ein Angreifer als Firmenmitglied ausgeben.

8.2.2 Beziehungen aufbauen

Der Angreifer begiebt sich in eine entsprechende soziale Position um Beziehungen aufzubauen. In seiner angenommenen Identität unterstützt er das Opfer und baut ein Vertrauensverhältnis auf. Diese Phase kann sehr kurz oder auch sehr lang dauern.

8.2.3 Beziehungen ausnutzen

Der Angreifer benutzt seine Beziehungen um Informationen zu bekommen, wie etwa Urlaubszeiten, Zugangsdaten und mehr. Auch kann er Menschen in dieser Phase dazu bringen, unrechtmäßige Handlungen durchzuführen wie das Anlegen eines Benutzeraccounts.

8.2.4 Ausführung

Sobald ein Angreifer die Zugriffsmöglichkeit hat, kommt es zum eigentlichen Angriff bei dem etwa eine bestimmte Information gestohlen wird, oder sich selbst Geld überwiesen wird.

8.3 Arten

8.3.1 Technology based

Die bekannteste Form ist das Phishing durch Mails von potentiell vertrauenswürdigen Absendern mit der Aufforderung einer Handlung. Dabei wird oft auf eine Website verwiesen die einen genauen Nachbau einer eigentlichen Website darstellt nur mit einem gewissen Clou.

8.3.2 Human based

Hier gibt es eine direkte Kommunikation mit dem Opfer, meist über Telefon. Dabei werden di zuvor diskutierten menschliche Verhaltensweisen ausgenutzt. Human based Social Engineering ist fast immer mit Identitätsdiebstahl verbunden → der Angreifer gibt sich als eine andere Person aus.

8.3.3 Reverse Social Engineering

Hier wird auf die Reziprozität gebaut. Der Angreifer kreiert eine Notsituation für das Opfer. Die Situtation ist so erstellt, dass das Opfer bei dem Angreifer Hilfe sucht und dieser freundlich und kompetent hilft. Das entstandene Vertrauensverhältnis kann der Angreifer später ausnutzen.

8.4 Social Engineering Model of Protection

Bestehend aus drei Verteidigungsschichten muss ein Angreifer sämtliche durchdringen um Informationen zu erhalten.

Die äußerste Schicht ist die Informationspolitik nach außen. Dadurch sollen Recherchen der Angreifer erschwert werden. In der Folge können Angriffe verhindert oder leichter entdeckt werden.

Die mittlere Schicht sind bewusstseinsbildende Maßnehmen der Mitarbeiter. Durch Schulungsprogramme sollen die Gefahren einer Social Engineering Attacke aufgezigt werden und ein Bewusstsein für die Sicherheitsrichtlinien geschaffen werden.

Die unterste Schicht sind systematische Sicherheitsmaßnahmen die sowohl physisch als auch technisch sein, wie etwa Spam Filter.

Es können auch noch mehr Schichten eingezogen werden die aus Kombinationen der Bereiche Organisation, Technik und Menschen bestehen können.

9 Betriebssystemsicherheit

Betriebssysteme sind hetzutage sehr vielfältig aufgrund der hohen Einsatzbereiche von Geräten. Grundsätzlich wird versucht die üblichen Sicherheitsziele als da wären, Vertraulichkeit Integrität und Verfügbarkeit zu erfüllen. Dazu können folgende Maßnahmen ergriffen werden:

- Berechtigungssystem
- Firewalls
- Speicherverwaltung
- Verschlüsselung von Filesystemen
- Jails / Sandboxes
- Härtung von Systemen

9.1 Härtung von Betriebssystemen

Die Grundgedanken sind:

- Ein nicht installiertes System kann nicht angegriffen werden
- Ein nicht vorhandenes Tool kann nich tfür einen Angriff verwendet werden
- Ein nicht vorhandenes Recht kann nicht missbraucht werden

Man versucht also ein minimales System zu erhalten um auch die Angriffsfläche zu minimieren → Leas Priviledge. Eine etwas formalere Definition:

> Härten bedeutet in der IT Sicherheit die Entfernung aller Softwarebestandteile und Funktionen, die zur Erfüllung der vorgesehenen Aufgabe durch das Programm nicht zwingend notwendig sind.

Härtung ist auch auf Systemen die keine Betriebssysteme sind möglich aber kein Ersatz für andere Sicherheitsmaßnahmen sondern nur ein Zusatz.

9.1.1 Ideales gehärtetes System

- Auswahl sicherer Hardware
- Auswahlt erforderlicher Services
- Auswahl geeigneter Admin Utilities falls erforderlich
- Auswahl eines geeigneten, gut supporteten, gut konfigurierbaren Betriebssystems
- Auswahl eines Konzepts zum System Monitoring um Angriffe zu erkennen

9.1.2 Real gehärtetes System

Eine ideale Härtung ist in der realen Welt kaum möglich. Gründe dafür werden im Folgenden behandelt:

Meist fehlt Know-How um eine sichere Konfiguration bis ins letzte Detail durchzuführen, denn eine ideale Härtung umfasst alle Details eines Systems. Ein so tiefgreifendes Verständnis ist schwer zu erlangen.

Oft fehlen auch Zeit und Geld um ein System ideal zu konfigurieren.

Bei 3rd Pary Software sind auch meist gar nicht alle Details konfigurierbar.

Um trotzdem ein gut gehärtetes System zu erhalten, können die folgenden Schritte ausgeführt werden:

1. Festlegen der erforderlichen Funktionen
2. Installieren des Systems
3. Nicht erforderliche Software entfernen oder zumindest deaktivieren
4. Zugriffsrechte streng setzen
5. Konfigurationen von Diensten überprüfen
6. Default Passwörter ändern

Am Beispiel von Linux: Es gibt eine Paketverwaltung (apt, pacman, ...) die Softwarepakete (de)installiert. Diese kümmert sich auch um Abhängigkeiten von Paketen, das aber nicht immer optimal. So werden oft unwichtige zusätzliche Pakete installiert, die aber oft einfach deinstalliert werden können. Ansonsten kann versucht werden sie am starten zu hindern.

9.2 Least Priviledge

Das Konzept des Least Priviledge heißt auch, dass es privilegierte und nicht privilegierte Accounts geben muss. In den meisten Systemen kann kurzzeitig höhere Berechtigung angefortert werden (`root`).

`root` hat unter Linux besondere Berechtigungen, daher ist `root` access meist das Ziel von Angreifern. Die Berechtigung sind etwa:

- Lesen beliebiger Dateien und Ordner
- Änderen der Zugriffsberechtigungen
- Wechsel der User ID und dessen Rechten
- Prozesssteuerung: Ändern der Prioritäten `kill` ...
- Ändern von Systemparametern
- ...

9.2.1 Einige technische Eigenheiten von Linux

Mithilfe des `suid` Bits können unberechtigte User Programme mit zusätzlich den Rechten des Beitzers aufrufen. Wenn dies aber ein fehlerhaftes Programm ist und es zu Buffer Overflows kommt, können Angreifer so mit `root` beliebigen Code ausführen. Daher sollte kontrolliert werden, ob alle Programme die das `suid` gesetzt haben, dies auch wirklich brauchen.

Der Befehl `chroot` kann das Wurzelverzeichnis für ein Programm geändert werden. So denkt ein Programm das im Pfad `/srv/www/` ausgeführt wird nach `chroot` dass es in `/` ausgeführt wird und sieht alles andere nicht. Mit `root` Rechten kann aus einer `chroot` Umgebung ausgebrochen werden. Daher ist es sinnvoll in so einer "Jail" Umgebung die Anzahl an Tools zu minimieren.

Wenn Programme dynamisch gelinkt werden, müssen alle erforderlichen Bibliotheken in dem Directory sein, da sie sonst nicht ersichtlich sind. Statisch gelinkte Applikation funktionieren wie gehabt..

Alle Systeme bringen Firewall Funktionalität mit, mit `iptables` oder `nftables` als Backend. Frontendmäßig gibt es mehrere Möglichkeiten. Sie filtert den Datenverkehr über den TCP Stack.

`ssh` ist auch auf Linux Standard mitlerweise um Remote Zugriff auf Systeme zu ermöglichen.

9.3 Sicherheitsmaßnahmen anhand von Windows

9.3.1 Starten von Programmen

Programme werden standardmäßig unter Nutzerrechten ausgeführt. Services hingegen sind von Nutzersitzungen unabhängig. Diese können daher auch ohne Nutzersitzung aktice sein. Daher sind sie ein gutes Ziel für Angriffe. Zur Minimierung der Angriffsfläche kann daher der *App Locker* verwendet werden um nur jene Programme für Nutzer freizugeben, die er auch wirklich benötigt.

9.3.2 Netzwerktraffic

Der Netzwerktraffic sotte mithilfe einer Whitelist (nur zugelassene Verbindungen) abgesichert werden. Auch dadurch wird die Angriffsfläche vermindert. Windows besitzt eine Firewall die den gesamten Datenverkehr über das TCP/IP Protokoll überwacht. Vom Typ her ist es eine *Sateful Inspection Firewall*. Dazu werden ein und ausgehende Pakete einer Session zugeordnet. Wenn die Pakete zu einer legitimen Verbindung gehören, werden sie zugelassen anderenfalls werden sie verworfen.

9.3.3 Administratorrechte

In Windows sind diese schon standardmäßig deaktiviert. Das heißt, dass Nutzer prinzipiell mit einem Nutzerkonto operieren. Eine Verwendung von Administratorrechten muss explizit angeordnet werden (Run as Administrator). Dadurch werden Programme initial nur mit Nutzerrechten gestartet.

9.3.4 Updates

Nur aktuelle Versionen des OS bieten akkuraten Schutz. Zusätzlich sollte der *Windows Defender Advanced Threat Protection* aktiviert werden. Die Verwendung von Sandboxen findet auch in immer mehr Applikationen Einzug und erschweren Angreifern den Zugriff auf das restliche System.

9.4 Zugriffssteuerung anhand von Windows

9.4.1 Login

Sie basiert auf der Rechtezuordnung zu bestimmten Nutzerkonten. Zunächst muss ein Nutzer seine Credentials im Login Screen eingeben. Diese werden über den *CredentialProviderDLL* gehashed und mittels *Winlogon* an den Local Security Authority Subsystem Service (LSASS) weitergegeben.

Das LSASS prüft die Kontorechte und ob das gewählte Konto die gewählte Login Art ausführen darf. Anschließend werden die Credential Hashes gekapselt und mit den gespeicherten Hashes aus der Security Accounts Manager (SAM) Datenbank abgeglichen. Abschließend stellt LSASS ein entsprechendes Nutzertoken (bzw Admin) aus. Dieses wird an den *Winlogon* Prozess zurückgegeben und mit der Nutzersitzung verbunden. Dieses Token wird für jeden Prozess /

Datenzugriff herangezogen um zu bestimmen ob der Nutzer für die entsprechende Aktion berechtigt ist.

Das LSASS benötigt für seine Arbeit verschiedene Datensätze zu Benutzerkonten. Für deren Verwaltung gibt es zwei verschiedene Ansätze:

- Lokale Nutzerverwaltung: das System erledigt alle Aufgaben für die Benutzerkontenverwaltung selbst. Die SAM Datenbank enthält dabei sämtliche Nutzerkonten und deren Credentialhashes. Daher war sie in der Vergangenheit ein beliebtes Angriffsziel für Cracking Angriffe. Diese Version wird bei isolierten Client PCs eingesetzt
- Active Directory: Eignen sich für Netzwerk PCs. In diesem Modus werden Nutzer Gruppen und Credentials am *Domain Controller* vorgehalten und über das Kerberos Protokoll auf diesem geprüft.

9.4.2 Dateisystem

Nach dem Login interagieren Nutzer mit dem Dateisystem NTFA. Dieses regelt Zugriffsberechtigungen intern mittels *NTFS Rechten*. Es existiert also ein Zugriffsschutz bereits auf Dateiebene. Jeder Datei wird dabei eine Access Control List zugeordnet die angibt, welche Nutzer bzw Gruppe Zugriff hat.

9.4.3 Berechtigungen

In Windows werden diese in Kontorechte und Privilegien unterteilt. Kontorechte umfassen dabei die unterschiedlichen Berechtigungen zum Login des Nutzerkontos. Diese regeln die Anmeldemodalitäten für ein bestimmtes Konto. Sie werden nur beim Login durch das LSASS geprüft.

Die Privilegien regeln den Zugriff für Nutzer und Gruppen auf spezielle Operationen und Aktionen nach dem Login. Dazu wird nach dem Login über das Nutzertoken dynamisch geprüft, ob ein Nutzer berechtigt ist eine Aktion durchzuführen. Die meisten dieser Rechte müssen von einem Administrator freigegeben werden (Least Priviledge).

Berechtigungen können auch an Gruppen von Nutzern erteilt werden. Die Gruppen können durch Admins erstellt, bearbeitet, zugewiesen und gelöscht werden.

9.4.4 Steuerung / Prüfung von Nutzerprivilegien

Das Zentrale Element hierbei ist das *Windows Access Token*. Nach dem positiven Login wird es durch das LSASS erstellt und an eine Nutzersession gebunden. Wenn ein Nutzer Admin Rechte hat wird parallel dazu ein zweites Admin Token erstellt. Jeder Prozess der in dieser Nutzersitzung gestartet wird, bekommt eine Kopie des Tokens, erbt also die Nutzerrechte.

Wenn der Active Directory Modus aktiviert ist, werden anstelle der Windows Tokens Kerberos Tickets genutzt, die analog verwendet werden.

Da verschiedenen Prozesen mehr oder weniger vertraut wird, werden entsprechende Mechanismen benötigt. Dazu gibt es die *Mandatory Integrity Control* die Prozessen Security Identifier zuweist, die das Vertrauen in diese widerspiegeln. Mögliche Vertrauensstufen:

- System: höchste Stufe und nur Systemprozessen/nutzern vorbehalten
- High: hoch und nur durch Admins zugreifbar, Prozesse auf diesem Vertrauenslevel können nicht durch Standardbenutzer beeinflusst werden.
- Medium: Stufe der Standardbenutzer
- Low: Prozesse die eine erhöhte Angriffsfläche bieten
- Untrusted: verwendet für anonyme Anmeldungen verwendet und Prozesse können nur im Arbeitsspeicher existieren.

Die Benutzerkontensteuerung bietet die Möglichkeit einem Standardnutzer Ressourcen mit hoher Vertrauensstufe aufzurufen. Dazu wird bei einem Standardnutzer ein Popup mit Aufforderung zur Eingabe von Admin Credentials bzw bei einem Admin Konto eine ja nein Frage.

In Firmennetzwerken können Einstellungen an mehrere Windows Clients mittles *Group Policy Objects* verteilt werden. Sie enthalten Anweisungen zur Konfiguration und werden innerhalb eines Active Directory automatisiert verteilt. Die GPOs werden regelmäßig verteilt. Es gibt:

- Lokale GPOs: Ausnahmekonfigurationen und durch das AD verteilt
- Globale GPOs: betreffen sämtliche angebundene Clients.

9.4.5 Logging

Windows legt unterschiedliche Logdateien an wie etwa der *Eventlog*. In dieser können Anwendungsereignisse oder etwa sicherheitsbezogene Ereignisse ausgelesen werden.

9.5 Backdoors / Rootkits

Unter Backdoors werden Wege verstanden auf dem Dritte unbemerkt auf das System zugreifen können. Diese sind meist unebkannt und illegal. Durch eine Backdoor können geschützte Daten eingelesen bzw verändert werden ohne, dass dies bemerkt wird.

Rootkits sind Programme die beim Start eines Systems mit gestartet werden. Dabei versuchen sie tiefgreifende OS Funktionen so zu ändern, dass die Rootkits unbemerkt von den Benutzern ausgeführt werden.

10 Sicherheitskonzepte von macOS

Grundsätzlich ist MacOS Closed Source, d.h. dass man keinen Zugriff auf den Source Code hat. Informationen sind daher stark von Apple selbst abhängig.

10.1 Updatezyklus

Das OS bekommt jährlich einen Versionssprung mit größeren Features und Design-Änderungen. Außerdem werden über Point Releases mit Security-Patches und Fehlerbehebungen versorgt. In der Regel werden Macs sieben Jahre lang mit Versionsupdates versorgt und drei weitere Jahre mit Security-Patches.

OSX hat ähnlich zu Windows einen Hybrid Kernel der eine Mischung aus einem monolithischem Kernel (ein großes Executable) und einem Microkernel ist. Es ist ein UNIX basiertes System mit Darwin als Kernkomponente.

10.2 Secure Enclave

Beschreibt einen isolierten Bereich auf einem eignes dafür bestimmten Prozessor. Bei Intel Macs ist die Secure Enclave extern des Prozessors und bei M1 Macs im Chip enthalten aber doch abgegrenzt. Das Ziel der Abgrenzung sind *Side Channel Attacken* (Kapitel **??**sec:2)).

Außerdem läuft der Prozessor auf einer niedrigen Taktrate die Clock und Power Attacken verhindern soll.

Auch der Speicher der Secure Enclave ist vom Anwendungsprozessor isoliert. Dazu wird beim Start des Systems ein zufälliger Schlüssel generiert. Alle Daten die die Secure Enclave schreibt werden mit diesem verschlüsselt und es wird ein Message Authentification Code (MAC) generiert. Nur wenn dieser MAC gültig ist, können Daten wieder entschlüsselt werden.

Weiters ist ein dezidierter True Random Number Generator enthalten der Vorteile für die Schlüsselerstellung bringt. Für die asymmetrische Verschlüsselung gibt es auch einen extra Block, den Public Key Accelerator (PKA) und für die symmetrische die AES-Engine.

Die Enclave beinhaltet auch einen sicheren nicht flüchtigen Speicher aus Entropiegründen bei der Generierung von Schlüsseln.

Die Aufgaben sind im Großen und Ganzen die Verschlüsselung und der Schutz der Benutzer Daten. Dazu zählt auch die Festplattenverschlüsselung Filevault. Sie beinhaltet auch die Biometrie-Informationen für den Login.

Dadurch, dass der Prozessor abgegrenzt ist und man keinen direkten Zugriff auf deren Daten hat, ist es für Dritte deutlich schwerer Angriffe durchzuführen.

10.3 Apple File System - APFS

Es ist speziell auf Flash-Speicher optimiert und der Aufbau besteht aus Containern und Volumes. Container können mit Partitionen verglichen werden und können mehrere Volumes enthalten.

Bei einer Standard Installation besteht das File System aus mehreren Volumes.

- System Volume: read only, hier befindet sich das Betriebssystem. Dadurch kann das System im laufenden Betrieb nicht von Dritten modifiziert werden
- Signed System Volume: in späteren Versionen wurde das SV weiters durch eine Signierung geschützt. Der Kernel Prüft während der Laufzeit, ob Inhalte unverändert sind und gestattet nur dann die Ausführung
- Data Volume: enthält die User Daten und User Programme und ist durch File Vault geschützt
- Preboot Volume: Informationen die für den Boot Prozess wichtig sind.
- VM Volume: Speicher für Swap Files falls der Arbeitsspeicher ausgeht
- Recovery Volume: enthält das Recovery OS mit Fetplattendienst und Terminal

10.4 File Vault

File Vault ist die Festplattenverschlüsselung und verwendet den Advanced Encryption Standard mit Xor Encrypt Xor (XEX) und operiert auf der Ebene von Volumes. Die Secure Enclave kümmert sich dabei nur um die Schlüsselverwaltung. Die Daten werden durch einen von System generierten Schlüssel gesichert. Dieser ist aber noch nicht geschützt. Daher wird bei der Aktivierung von Filevault der Schlüssel durch einen einzigartigen Hardwareschlüssel (Hardware UID) und einen generierten Schlüssel nochmals verschlüsselt. Der generierte Schlüssel (Key Encryption) ist selbst durch die Login Daten und den Hardwareschlüssel gesichert. Weiters gibt es einen xART Schlüssel der verhindert, dass die Festplatte nur mehr durch die Hardware UID entschlüsselt werden kann.

Standardmäßig ist auf Macs die Festplatte sowieso verschlüsselt (nur Hardware UID). Damit bei einer nachträglichen Aktivierung von Filevault nicht die gesamte Festplatte nochmal verschlüsselt werden muss, wird der Volumeschlüssel noch einamal mit dem Key Encryption Schlüssel verpackt (Bild 8).

10.5 Keychain und Find My

Der Passwortmanager im Apple System heißt Keychain und bietet die Möglichkeit über die Apple ID mit anderen Geräten synchronisiert zu werden.

Die Find My Funktion ermöglicht das Auffinden verloren gegangener Geräte über Internet und Bluetooth und muss dazu nicht einmal sichtbar eingeschaltet sein. Durch Find My kann auch ein Gerät remote gelöscht werden.

Das Activation Lock soll Diebstähle verhindern, da das Gerät nur durch die

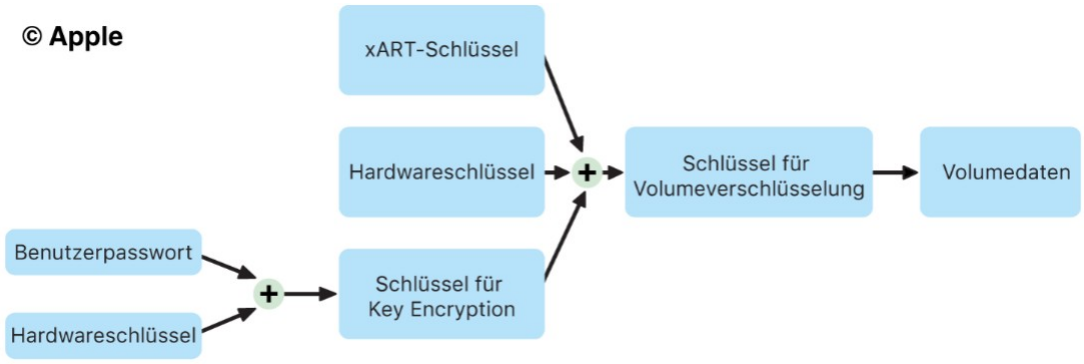

Figure 8: File Vault

Eingabe der Apple ID entsperrt und erneut aktiviert werden kann. Der Apple Support kann das aber umgehen wenn man beweisen kann, dass man sich selbst ausgesperrt hat.

10.6 Private Relay

Die Idee hinter der Technologie ist, dass keine Partei gleicheitig weiß Wer man ist und Wohin man sich verbindet. Die IP Adresse ist für den Netzanbieter sichtbar. Die abgerufene Ressource ist allerdings weder von Apple noch von dem Netzbetreiber sichtbar, weil sie verschlüsselt ist (erstes Relay).

Das zweite Relay generiert eine temporäre IP Adresse, kennt aber die des Users nicht, da es nur verschlüsselte Anfragen von Seiten des Users über das erste Relay gibt. Die Ressource wird nun abgerufen und nimmt den gleichen Weg zurück.

Die Technologie ist der eines VPN recht ähnlich, allerdings ist es weniger flexibel und nur von Safari unterstützt.

10.7 Intelligent Tracking Prevention

Diese Option verhindert Profiling im Internet. Dabei wird mittels on device Machine Learning Tracking erkannt und blockiert. In dem durch Safari erstellten Privacy Report werden die Trackingversuche festgehalten. Weiters soll es gegen Browser Fingerprinting helfen.

10.8 XD, ASLR, Secure Boot, Malware Erkennung, Downgrade Verbot und Systemressourcen-Zugriff

XD (Executable Disable) kann Bereiche im Speicher als nicht ausführbar markieren und der Prozessor verweigert dann die Ausführung.

ASLR (Address Space Layout Randomization) schützt vor Memory Corruption Bugs in dem auszuführende Programmblöcke an zufälligen Addressen gespeichert werden. Dadurch ist es schwer Schadcode an den angrenzenden Addressen abzulegen.

Die System Integrity Protection ist ein Mechanismus um spezifische Speicherorte auf read only zu setzen. Ist standardmäßig aktiv, kann aber deaktiviert werden.

Die eingebaute signaturbasierte Malwareerkennung heißt XProtect.

Das Zurücksetzen auf ältere Versionen ist verboten um vor Downgrade Angriffen zu schützen.

Zugriffe auf Systemressourcen werden von Betriebssystem überwacht.

10.9 Programme installieren, Berechtigungen und Sandboxing

Es können einerseits über den Mac App Store (Reviewed von Apple) als auch über das Internet mittels Gatekeept Technologie Programme geladen werden. Diese Technologie prüft deren Authentizität durch Überprüfungen mit der Code Signatur die mit einem Apple Developer Certificate durchgeführt werden können.

Die Berechtigungen können ziemlich fein granular eingestellt werden. So kann für jede Applikation geregelt werden, welche Ressourcen verwendet werden können.

Sandboxing schützt nicht vor nicht autorisiertem Zugriff durch Anwendungen.

10.10 Ältere OSX Versionen nutzen

Apple löst manchmal einfach Features auf, wie etwa Support für 32 bit Anwendungen. Daher sind Nutzer mit spezifischen Tools manchmal auf ältere Versionen angewiesen. Offiziell gibt es keine Supportzeiträume aber tendenziell sind die Vorgängerversionen sicher nutzbar, allerdings nur die n-1te Iteration. Daher gilt troztdem, immer die aktuellste Version nutzen.