

Zusammenfassung Informationssicherheit

Mathis Hermann

December 13, 2022

1 Gesetzliche Grundlagen

Schweizer Recht ist uneingeschränkt anwendbar, wenn Täter, Opfer und Tatort in der Schweiz liegen. Im Internet ist das ein Problem.

1.1 Gesetzestexte

Es gibt keum für das Internet spezifische Gesetzestexte in der Schweiz. Es gibt aber viele Gesetzestexte, die im Zusammenhang mit dem Internet angewendet werden: 1) Obligationenrecht, 2) Strafgesetzbuch, 3) Datenschutzgesetz, 4) Fernmeldegesetz, 5) Urheberrechtsgesetz – (Liste nicht abschliessend)

Dabei können auch diverse Unterschiede zwischen CH und EU-Recht gefunden werden.

Recht	CH	EU
Widerrufsrecht	Freiwillig (10 Tage)	14 Tage
Lieferfrist	Keine Obergrenze	maximal 30 Tage; länger kann verabredet werden
Gewährleistung	2 Jahre	2 Jahre
Bestell-Button	Keine Vorgabe	Muss als letzter Punkt des "Vertrages" auf der Seite stehen und mit "Zahlungspflichtig bestellen" <i>eindeutig</i> beschriftet sein.
Preise	Tatsächlich zu bezahlen- der Preis in CHF inkl. aller nicht frei wählbarer Zuschläge jeglicher Art. Einheiten und Verrech- nungssätze müssen klar ersichtlich sein.	Gesamtpreis ein- schliesslich aller Steuern und Abgaben.

Obligationenrecht Definiert Formvorschrift: Es braucht im Allgemeinen keine besondere Form für das Zustandekommen eines Vertrages.

Alle Parteien haben Rechte und Pflichten:

- Vertrag muss eingehalten werden
- Haftbarkeit bei versäumten Pflichten
- Kein Recht ohne Verpflichtungen

Datenschutzgesetz Gilt für das Bearbeiten von Daten natürlicher und juristischer Personen. Informationen können mit Verweis auf Datenschutz Gericht nicht vorenthalten werden.

- Personendaten – alle Angaben, die sich auf eine bestimmte oder bestimmbar Person beziehen
- Betroffene Personen – natürliche oder juristische Personen, über die Daten bearbeitet werden
- Besonders Schützenswerte Personendaten – Daten über:

- die religiösen, weltanschaulichen, politischen, oder gewerkschaftlichen Ansichten oder Tätigkeiten
- die Gesundheit, Intimsphäre oder Rassenzugehörigkeit
- Massnahmen der sozialen Hilfe
- Administrative oder strafrechtliche Verfolgungen und Sanktionen

Jede Person kann vom Inhaber einer Datensammlung Auskunft darüber verlangen, ob Daten über sie bearbeitet werden. Lässt der Inhaber der Datensammlung Personendaten durch einen Dritten bearbeiten, so bleibt er auskunftspflichtig. Die Auskunft ist in der Regel schriftlich, in Form eines Ausdrucks oder einer Fotokopie sowie kostenlos zu erteilen.

Fernmeldegesetz Es ist eine Meldepflicht für Fernmeldeanlagen definiert – *Fernmeldedienst*: fernmeldetechnische Übertragung von Informationen für Dritte; Senden und Empfangen von Informationen über Leitungen oder Funk etc.

WLAN-Accesspoint, Mailserver, Chatserver Gameserver mit Chatfunktion, Powerline-LAN sind grundsätzlich meldepflichtig – Paragraph 2 bestimmt Ausnahmen.

Urheberrechtsgesetz Werke sind, unabhängig von ihrem Wert oder Zweck, geistige Schöpfungen der Literatur und Kunst, die individuellen Charakter haben. Urheber:in ist Person, welche das Werk geschaffen hat. Urheber:in hat das ausschliessliche Recht über das zu bestimmen, ob, wann und wie das Werk verwendet wird.

Wird in einem Arbeitsverhältnis bei Ausübung dienstlicher Tätigkeiten sowie in *Erfüllung vertraglicher Pflichten* ein Computerprogramm geschaffen, so ist Arbeitgeber:in allein zur Ausübung der ausschliesslichen Verwendungsbefugnisse berechtigt.

1.2 revDSG

Per Juni 2023 wird eine Revision des Datenschutzgesetzes eingeführt:

- Nur noch natürliche Personen sind geschützt
- Genetische und biometrische Daten werden besonders schützenswerte Daten
- Grundsätze "Privacy by Design" und "Privacy by Default" werden eingeführt

- Informationspflicht gilt für jede Form von Personendaten
- Ein Verzeichnis der Bearbeitungstätigkeiten wird zur Pflicht
- Pflicht zur raschen Meldung bei Verletzung der Datensicherheit

1.3 Praxis

Impressumspflicht gilt typischerweise sobald eine Website gewerblich oder kommerziell ist. Impressum muss leicht auffindbar sein und Vorname und Name (bzw. Firma), Adresse und E-Mail-Adressen beinhalten.

Abschliessend Informatiker:innen unterliegen beim Schreiben eines Programmes oder Aufsetzen eines Servers im Minimum immer der schweizerischen Gesetzgebung. Es existieren viele Gesetze, welche einfach zu übertreten sind, aber nicht übertreten werden dürfen.

Im Internet müssen je nachdem auch Gesetze von Drittstaaten berücksichtigt werden.

Gesetzestexte wurden nicht für Laien und nicht von technischen Fachleuten geschrieben. Gesetzestexte folgen nicht immer der Vernunft oder Logik.

Publikationen können helfen Zweifelsfälle zu beantworten.

2 Sicherheitsvokabular

2.1 Das Sicherheitsvokabular

- *Confidentiality / Vertraulichkeit* – gewährleistet, dass Daten nur für befugte Entitäten zugänglich sind.
- *Integrity / Integrität* – gewährleistet, dass Daten, Programme oder Funktionen in unveränderter Form belassen werden (immer gemessen am Sollzustand).
- *Availability / Verfügbarkeit* – gewährleistet, dass eine Funktion immer dem Funktionsbezüger zur Verfügung steht.
- *Authenticity / Authentizität* – Echtheit, Überprüfbarkeit, Vertrauenswürdigkeit eines Objektes.
- *Non-Repudiation / Nicht-Abstreitbarkeit* – Es kann nachgewiesen werden, dass eine Handlung tatsächlich stattgefunden hat.
- *Accountability / Verbindlichkeit* – Konsequenzen einer Handlung sind rechtlich bindend.
- *Asset / Aktivposten* – Mit einem Aktivposten wird Wert verbunden; ist meistens subjektiv
- *Threat / Bedrohung* – mögliche Gefahr, die eines der Assets in seinen Schutzzielen beeinträchtigen könnte
- *Vulnerability / Schwäche* – Zustand oder Begebenheit, durch die ein Asset gefährdet sein könnte
- *Single Loss Expectancy / Schadensausmass* – Wert, der die Höhe des einzelnen Schadensereignisses bemisst
- *Probability of Occurence / Eintrittswahrscheinlichkeit* – Faktor, der verkörpert mit welcher Wahrscheinlichkeit ein bestimmtes Ereignis oder ein bestimmter Schaden eintritt; Wird über einen bestimmten, längeren Zeitraum betrachtet
- *Risk / Risiko* – Produkt aus Schadensausmass und Eintretenswahrscheinlichkeit; Ereignis oder ein Effekt, welches nur mit einer gewissen Unsicherheit eintritt
- *Risk Strategy / Risikostrategien*

- *Risiko Vermeidung* – Vermeidung funktioniert häufig nicht
 - *Risikoreduktion / Risikooptimierung* – Reduktion ist das Ziel; Erfolgt eine Reduktion nach rein rechnerischen Grundsätzen, wird von *Risiko-Optimierung* gesprochen; Wenn das Risiko über den optimalen Grad gesenkt wird, spricht man von typischerweise von *Risikoreduktion*
 - *Risikotransfer* – Risiko auf grössere Entität transferieren wenn Risiko nicht mehr durch eine einzelne Entität abgefangen werden kann
- *Risikograph* – Risiken quantifizieren

Risikograph wird erstellt, um herauszufinden, wo und wie Risiken reduziert werden können. Der Graph ist gut zum optimieren (wo kann optimiert werden, wo kann Geld gespart werden, wo ist das Potenzial für Risikosenkung und wie viel Geld kann aufgewendet werden).

- Welche Risiken werden zusammengefasst?
- Welche Risiken werden aufgespalten? – Risikobetrachtung schön machen
- Es gibt Risiken, welche immer als Pakete auftreten:
 - Wahrscheinlichkeit, dass mehrere eintreten ist hoch
 - Rest des Systems muss kompensieren für ausgefallenes System
 - Kannibalisierung der Ressourcen

2.2 Authentication, Authorisation und Accounting

AAA-System bildet einen Grundpfeiler der IT-Sicherheit; Primäres Ziel ist *Confidentiality* zu gewährleisten. Vertraulichkeit und Verfügbarkeit von Daten und System gewährleisten

- *Authentication* – ist Person wer Person sagt sie ist
- *Authorisation* – darf Person, was Person macht
- *Accounting* – was macht Person (sammeln)

Authentication Benutzern / System / Prozess zweifelsfrei identifizieren; Identifizierung mit Überprüfung; Identifikation oft nicht geheim; Mit Username wird oft identifiziert und mit PW authentifiziert;

Mehrfaktor-Authentication – wird verwendet, wenn ein höheres Mass an Sicherheit gefordert ist; Nur sinnvoll, wenn nicht der selbe Faktor;

Authentifizierungsklassen:

- Something I know – e.g. Password
- Something I have – e.g. Mobile Phone, App, SMS, Streichliste
- Something I am – Fingerprint, Auge, Gesicht (einfach zu hacken)

Authorisation Es wird festgelegt, ob ein Benutzer Rechte hat um bestimmte Aktionen auszuführen; Es gibt mehrere Arten von wie die Authorisation gehandhabt werden kann. Typischerweise:

- *MAC* Mandatory Access Control – Rechte Aufgrund von Regeln; können verloren gehen
 - Position in Organigramm
 - Sicherheitsfreigabe
 - Zeichnungsberechtigungsstufe
- *DAC* Discretionary Access Control – Rechte werden vom Eigentümer eines Zielobjektes vergeben; Rechte werden vergeben, gehen nicht verloren
 - Eigentümer der Applikation oder Daten möchte es so haben
- *RBAC* Role-Based Access Control – Rechte werden aufgrund von Rollenzugehörigkeit vergeben; Rechte können verloren gehen
 - Aufgrund einer Rolle, die im Organigramm / aufgrund Tätigkeit zugewiesen wurde

Bei allen diesen Modellen wird einem Initiator bestimmte Rechte auf einem Zielobjekt gegeben

Accounting Informationen über die Ressourcenbenutzung sammeln (wer hat wann, was gemacht):

- Kapazitätsplanung
- Trendanalysen
- Verrechnung
- Revisionsfähigkeit

3 Verschlüsselung (Teil 1: Grundoperationen)

3.1 Symmetrische Verschlüsselung

- Gleicher Schlüssel zum Ver- und Entschlüsseln
- Einfach zu implementieren
- Typischerweise einfache, schnelle Operation (schneller als asymmetrisch)
- Kurze Schlüssel sind verhältnismässig sicher
- Praktisches Problem: Schlüssel muss sicher zwischen Sender und Empfänger ausgetauscht werden

Symmetrische Verfahren

- *RC4* gebrochen
- *DES* gebrochen
- *3DES* gebrochen
- *Camelia* 128+ bits
- *AES* 128+ bits

Einsatz Verschlüsselungen werden in verschiedenen Richtungen eingesetzt:

- Transportverschlüsselung
 - Ohne Authentifizierung – email
 - Mit einseitiger Authentifizierung – https: Informationen auf Website
 - Mit wechselseitiger Authentifizierung – banking: mit nicht-transferierbaren Tokens
- End-to-End Verschlüsselung
 - Schlüssel zu Informationen nur auf Zielgerät
 - nur mit wechselseitiger Authentifizierung mit nicht transferierbaren Tokens (Threema, Signal)

3.2 Asymmetrische Verschlüsselung

- Verschlüsselung mit Schlüsselpaar
- Verschlüsselung mit anderem Schlüssel als Entschlüsselung
- Aufwändige Mathematik und grosse Schlüssellängen werden benötigt (2048+ Bits) (ca. 1000x aufwändiger als symmetrisch)
- Austauschbarkeit der Schlüssel existiert, ist aber nicht überall vorhanden

Elliptische Kurven Punkte werden mit Linien verbunden und anschliessend gespiegelt. Benötigen wesentlich kürzere Schlüssel als RSA um sicher zu sein (geometrische Verschlüsselung).

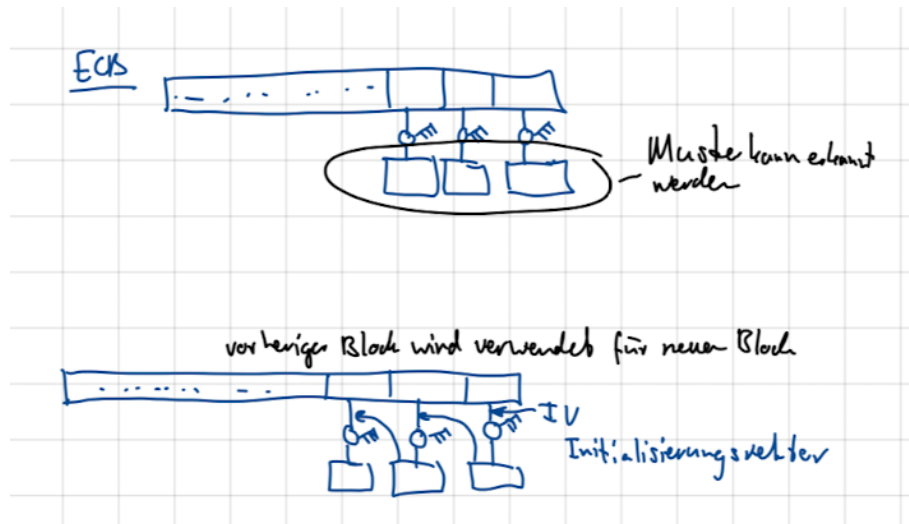
Verfahren

- *RSA und DSA*
- *ECC192* – gebrochen
- *RS2048* – vermutlich teilweise gebrochen
- *RSA4096+* – Ein mathematischer Körper wird als Basis errechnet; anfällig für Quantencomputer
- *ECC256+*

Ciphers, Modes, Paddings bei Block-Ciphern

- *Cipher* – Legt fest wie etwas verschlüsselt wird; Schlüsselgrösse gleich Blockgrösse, wenn nicht anders spezifiziert
- *Mode* – Legt fest, wie mehrere aufeinander folgende Blöcke von Daten verschlüsselt werden (ECB –
 - Electronic Code Block Mode: Block für Block wird mit dem selben Schlüssel verschlüsselt; Muster kann erkannt werden.
 - Vorheriger Block wird zusammen mit Block verwendet und mit Schlüssel verschlüsselt. Erster Block wird mit Initialisierungsvektor genommen

- *Padding* – Legt fest, wie beim letzten Block signalisiert wird, welche Bytes noch benutzt sind und welche nicht (01 bei 1 Byte, 0202 bei 2 Byte etc.)



Ciphern

- *AES* Schlüssellänge 256 verwenden (128 gebrochen); Blocklänge 128 andere existieren, werden aber seltener verwendet
- *Camelia* Feistelcipher mit Schlüssellänge 192, 256 (128 potenziell gebrochen); Blocklänge 128 Bit
- *Cast-256* ungebrochen; Schlüssellänge 256 (128, 169 192, 224 potenziell gebrochen); Blockgröße 128 Bit

Hashing Einwegfunktion um Daten zu einer nicht rückverfolgbaren Zeichenfolge zu formen. Wird mit oder ohne Salt verwendet (Salt wird vor das Dokument gesetzt, welche unabhängig von den Daten ist). Ein Hash sagt nichts über die Eigenschaft eines Dokumentes aus, ist aber einzigartig.

Eine Rainbow-Table enthält alle möglichen Kombinationen von Hashes.

Hashing Verfahren

- *MD5* – gebrochen
- *RIPE-MD 160* – gebrochen
- *SHA1* – gebrochen

- *SHA2/3 224* – vermutlich gebrochen
- *RIPE-MD 256+*
- *SHA2/3 256+*

3.3 Zertifikate

Komponenten von Zertifikaten:

- *Identifikation* – Felder *SubjectDN* und *SubjectAlternateName*
- *Öffentliche Schlüssel* – öffentlicher Teil eines asymmetrischen Schlüsselpaares; kann zum Prüfen von verschlüsseltem Material oder zum Verschlüsseln von Material an den Zertifikatsanbieter dienen
- *Gültigkeitsdauer* – Startzeitpunkt und Endzeitpunkt
- *Verwendungszweck* – Benutzung kann eingegrenzt werden; e.g. nur für Signatur von Emails oder Authentisierung von Web-Clients

Key- und Truststore Jede Applikation, die mit Zertifikaten arbeitet hat idR 2 Stores und folgenden Inhalt:

- *Keystore* – Zertifikate und den privaten Schlüssel
- *Truststore* – Trust-Anker (Root-Zertifikate)



4 Verschlüsselung (Teil 2: PKI und andere Vertrauensmodelle)

Hybride Verschlüsselung und PFS *Perfect Forward Secrecy* gewährleistet, dass beim Bruch eines Langzeitschlüssels, die Kommunikation nicht retrospektiv entschlüsselt werden kann.

4.1 Signieren

Vom Dokument wird ein Hash generiert, welcher mit dem privaten Schlüssel verschlüsselt wird. Das Dokument wird unverschlüsselt zusammen mit dem verschlüsselten Hash an den Empfänger geschickt. Der Empfänger kann mit dem öffentlichen Schlüssel des Versenders den Hash prüfen.

Signieren mit OpenSSL

Signieren mit OpenSSL	
Erzeugen eines selbstsignierten Zertifikates mit einem RSA2048 -Schlüssel:	
1	<code>\$ openssl req -nodes -x509 -sha256 -newkey rsa:4096 -keyout "\${whoami}_sign.key" -out "\${whoami}_sign.crt" -days 365 -subj  "/C=CH/o=FINW/L=Windisch/OU=IMVS/CN=\${whoami}s_Signatur_Schluesself"</code>
Erstellen einer Signatur-Datei mit einem sha256 Hashing:	
1	<code>\$ openssl dgst -sha256 -sign "\${whoami}_sign.key" -out meinfile.txt.sign meinfile.txt</code>
Prüfen der Signatur (dazu wird selbstverständlich das Zertifikat benötigt):	
1	<code>\$ openssl dgst -sha256 -verify <(openssl x509 -in "\${whoami}_sign.crt" -pubkey -noout) -signature meinfile.txt.sign  meinfile.txt</code>

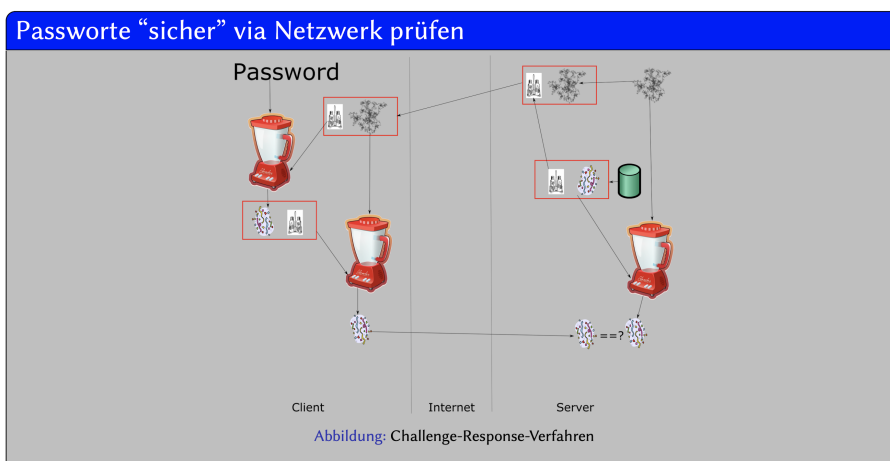
Signieren mit PGP

Signieren mit PGP	
Ein Schlüssel wird wie folgt erstellt:	
1	<code>\$ gpg --full-generate-key</code>
Anschliessend signieren wir die Mitteilung auch hier:	
1	<code>\$ gpg --detach-sign --output meinfile.txt.pgp.sign --armor meinfile.txt</code>
Eine Überprüfung erfolgt analog dazu mittels:	
1	<code>\$ gpg --verify meinfile.txt.pgp.sign meinfile.txt</code>

Signieren einer Mitteilung

Signieren einer Mitteilung	
Eine Datei signieren Sie mit:	
\$	<code>gpg --sign --armor examplefile</code>
Das Ergebnis ist Binär und suggeriert deshalb eine Verschlüsselung. ist aber nicht so. Ehrlicher ist deshalb:	
\$	<code>gpg --clearsign examplefile</code>
oder mit einer "abgesetzten Signatur";	
\$	<code>gpg --detach-sign --armor examplefile</code>
Um die Unterschrift der Datei zu prüfen verwenden Sie:	
\$	<code>gpg --verify examplefile.asc</code>

Passworte "sicher" via Netzwerk prüfen



Zero Knowledge Proof (ZKP) Prüfer (P / Peggy) beweisen den Besitz eines Passwortes, ohne dass die verifizierende Instanz durch den Prüfvorgang zusätzliche Kenntnisse über das Passwort erhält. Drei Phasen:

- Peggy legt ein Commitment vor – mehrere richtige Antworten über ein Zusammenhang
- Victor fordert die Aufdeckung einer Antwort aus dem Commitment – Resultatsets überlappen sich
- Peggy deckt die Antwort auf

5 Zertifikate

Begriffe

- *Zertifikat* – Überprüfbarer Schlüssel der einem Verwendungszweck dient; Meistens ein unterschriebener Public Key
- *Certification Authority (CA)* – Instanz, die Zertifikate ausstellt
- *Registration Authority (RA)* – Instanz, bei der Zertifikate beantragt werden können und die den Antrag prüft
- *Certification Revocation List (CRL)* – Liste von Zertifikaten, die nicht mehr gültig sind
- *Validation Authority (VA)* – Instanz, die Zertifikate prüft; Überprüfung ist ein vollautomatischer Prozess

5.1 Public Key Infrastructure (PKI)

Eine PKI ist ein System, dass digitale Zertifikate ausstellen, verteilen und prüfen kann.

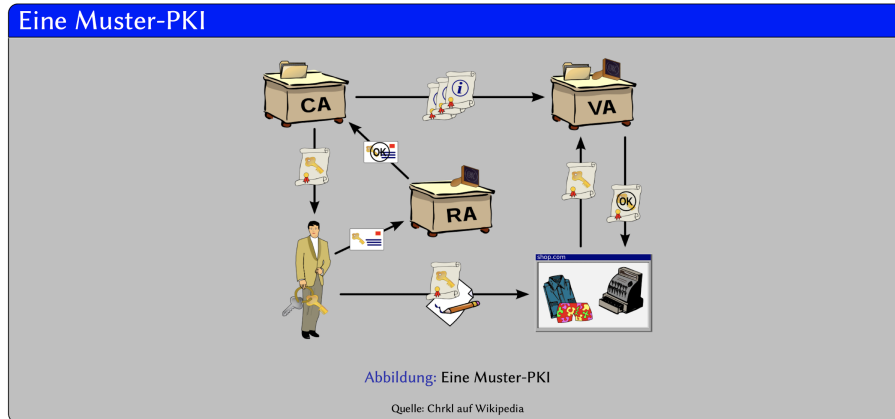
Vertrauensmodelle:

- *Hierarchisch* – Vertrauensstellung in eine CA. Jedes von ihr unterschriebene Zertifikat ist für die im Zertifikat angegebenen Aktionen gültig. Die CA unterschreibt ihr eigenes Zertifikat.
- *Web of Trust* – Zertifikate drücken sich gegenseitig ihr Vertrauen aus; jedem selber überlassen, wie Vertrauen definiert ist (e.g. PGP)
- *Cross Zertifizierung* – Zwei CA unterschreiben ihr Zertifikat; Vertrauensstellung nicht immer eindeutig geregelt und kann von Fall zu Fall variieren

Dokumente

- *Certificate Policy (CP)* – Beschreibt die Anforderungen an die CA und ihre Arbeitsweise; dient dritten zur Analyse der Vertrauenswürdigkeit
- *Certificate Practice Statement (CPS)* – Beschreibt die praktische Umsetzung des CP Dokumentes und die Praxis; unter welchen Bedingungen ein Zertifikat ausgestellt wird; Wenn CPS nicht öffentlich ist, gibt es noch ein Policy / PKI Disclosure Statement (PDS), welches den öffentlichen Teil enthält

PKI



5.2 Verschlüsselte Verbindungen

Aufbau von Verbindungen kann verschiedene Fehler darstellen:

- Nicht vertrauenswürdige Root-Zertifikate
- Abgelaufene oder zurückgezogene Zertifikate
- Verwendung eines Zertifikates, das nicht dafür zugelassen ist

Vertrauensverhältnis mit OpenSSL analysieren

Diagnose mit OpenSSL

OpenSSL stellt sowohl die Möglichkeit einen (toten) Server bereitzustellen, als auch einen Client (ähnlich wie `telnet`).

```
1 $ openssl s_client -connect gwerder.net:443 -tlsextdebug
```

Wenn Sie wissen möchten ob ein bestimmter Algorithmus vom Server akzeptiert wird, dann können Sie diesen auch direkt mit `-cipher` spezifizieren.

```
1 $ openssl s_client -connect gwerder.net:443 -cipher AES256-SHA
```

Mit etwas "schwarzer Bash-Magie" können Sie sogar Testen welche Algorithmen der Server nicht nur anbietet, sondern auch akzeptiert:

```
1 $ for i in $(openssl ciphers 'ALL:eNULL' | sed -e 's/:/_/g'); do echo -n "$i" | openssl s_client -connect gwerder.net:443 -cipher $i 2>/dev/null >/dev/null && echo "$i";done
```

Die Möglichkeiten sind hier unbegrenzt...

Es können auch Server-Verbindungen erstellt werden, um einen Client zu diagnostizieren – *Ein Server benötigt immer ein Zertifikat*

Diagnose mit Openssl für TLS-Clients

Ein generischer TLS-verschlüsselter Server:

```
1 $ openssl s_server -accept 7000 -cert gwerder.net.pem -CAfile gwerder.net_ca_chain.pem -key gwerder.net.key
```

Ein primitiver HTTP-Fileserver:

```
1 $ openssl s_server -accept 443 -cert gwerder.net.pem -CAfile gwerder.net_ca_chain.pem -key gwerder.net.key -WWW
```

Ein Echoserver mit Umkehrfunktion:

```
1 $ openssl s_server -accept 443 -cert gwerder.net.pem -CAfile gwerder.net_ca_chain.pem -key gwerder.net.key -rev
```

Diagnose mit Openssl mit OCSP

Die OCSP URL extrahieren

```
1 $ openssl s_client -connect gwerder.net:443 2>&1 < /dev/null | sed -n '/-----BEGIN/,/-----END/p' | openssl x509 -noout -ocsp_uri -in - ✓
```

Ausserdem die Liste der CA-Zertifikate extrahieren (das Root-CA-Zertifikat kann oder kann nicht enthalten sein)

```
1 $ openssl s_client -connect gwerder.net:443 -showcerts 2>&1 < /dev/null | sed -n '/-----BEGIN/,/-----END/p'
```

Einen OCSP-Request Absetzen:

```
1 $ openssl ocsp -issuer chain.pem -cert gwerder.net -resp_text -url http://ocsp.digicert.com
```

So ganz nebenbei: OpenSSL lässt sich auch als OCSP-Server verwenden (offiziell nicht zu Produktions-Zwecken)

```
1 $ openssl ocsp -index demoCA/index.txt -port 8080 -rsigner ocspSigning.crt -rkey ocspSigning.key -CA rootCA.crt -text -out log.txt & ✓
```

5.3 Security Tokens

Security Tokens sind ein wichtiger Faktor bei Mehrfaktor-Authentifizierung

- *Zeitsynchronisierte Tokens* – Präsentieren einen OTP aus ggw. Zeit und einem Geheimnis
- *CR-Basierte Tokens* – Eingabe einer Challenge erfolgt über eine Schnittstelle; Challenge wird mit dem Schlüssel der Person "signiert" und zurückgesandt
- *Usage Based Tokens* – Eine Art "Streichliste"; Bei jedem Druck auf eine Taste wird der nächste Code freigegeben; vorhergehende Codes werden automatisch invalidiert

Verbinden von Security Tokens Security Tokens sind in verschiedenen Varianten verfügbar

- Online – via USB; NFC; Fest auf einem Gerät verbaut
- Offline – Eingabe über Tastatur oder via Telefon

Wenn ein Token online ist, ist bei sicheren Tokens eine manuelle Aktion nötig um den Authentisierungsvorgang zu imitieren

Kritik

- *Verluste oder Diebstahl* – Verlust oder Diebstahl grundsätzlich kein Problem; muss mit den entsprechenden Massnahmen begleitet werden
- *Man-in-the-Middle* – Die meisten Smarttokens überprüfen nicht ihren Peer-Partner; Authentisierungs-Tokens sind anfällig für MITM wenn nicht anderweitig abgesichert

6 Technische und nicht-technische Angriffe

Es gibt mehrere Möglichkeiten Ereignisse zu klassifizieren.

Abwehrstrategien:

- Angriff verhindern
- Angriff erschweren
- Angriff ableiten (intern oder extern)
- Schaden kontrollieren
- Angriff entdecken (während und danach)
- Vom Angriff erholen

6.1 Nicht-Technische Angriffe

Angriffe, die ein Schutzziel beeinträchtigen können und als Ziel keine technische Anlage haben. Ziele sind Menschen, Prozesse, Zustände.

Dumpster Diving vertrauliche Informationen werden über Abfall, Altpapier, ausrangierte Hardware oder vergleichbare Kanäle besorgt; Sonderform über Webarchiv in der Vergangenheit publizierte Informationen beschaffen

Spying / Eavesdropping Informationen werden beschafft, indem Beobachtungen angestellt werden, ohne dass Entitäten manipuliert werden. Beobachtungen können visuell, akustisch, technisch sein; *Eavesdropping* ist, wenn Informationen über das Mitschneiden von legitimen Datenflüssen beschafft werden

Social Engineering Menschen manipulieren um Sicherheitssperren zu umgehen. Mithilfe:

- Impersonation
- Tailgating
- Phishing / Vishing / Whaling
- Popup Fenster
- Interessante Applikationen

- Hoaxing

Angriffskanäle – Email, Telefon, Soziale Medien, SMS, Brief, direkt *Gegenmassnahmen* – Ausbildung aller Beteiligten; Etablieren von geeigneten Prozessen

6.2 Technische Angriffe

Klassifikation nach

- Schutzziel
- Strategie
- Schweregrad des potentiellen Schadens
- Wahrscheinlichkeit
- Komplexität

Gegenmassnahmen

- Zugriff auf die Systeme limitieren – *verhindern*
- Angriffsfläche gering halten – *erschweren, kontrollieren*
- Schadaktivitäten erkennen – *entdecken*
- Potentiellen Schaden gering halten – *ableiten, erholen*

Phasen Angriffe werden selten isoliert sondern koordiniert geführt. Phasen von Attacken 1) Reconnaissance, 2) Exploitation, 3) Reinforcement, 4) Covering Tracks

(Distributed) Denial-of-Service Bei der DoS-attacke geht es darum, ein Ziel vorübergehen oder bleibend vom Netzwerk zu schädigen. Die einfachsten Angriffe verursachen Überlastungen in einem Teilsystem. Fortgeschrittene Attacken zerstören Daten / Programme oder Hardware.

Man in the Middle Dritter klinkt sich in Konversation ein; lässt sich durch *Mutual Authentication* verhindern. Bei Man in the Browser kontrolliert der Angreifer einen der Endpunkte (nach der abgesicherten Verbindung).

7 Angriffe und Szenarien

Verschiedene Arten von Exploits: technische Möglichkeit, Schwachstellen eines Programms oder Hardware auszunutzen. Viele Arten von Exploits: 1) Remote Code Execution, 2) Local Privilege Escalation, 3) Information Disclosure / Manipulation.

Alle Exploits können Attribute aufweisen. Für Angreifer sind Exploits am attraktivsten, die öffentlich nicht bekannt sind.

Responsible Disclosure Schwachstellen werden oft einer *Responsible Disclosure* unterworfen. Anfänglich wird nur publiziert, dass eine Schwäche in einem System oder Software hat und wie gravierend sie ist. Details werden dem Hersteller oder Verantwortlichen zugänglich gemacht.

Gegenseitig dazu ist *Full Disclosure*, wenn Informationen schnellstmöglich breit zugänglich gemacht werden. Somit haben Täter und Opfer den selben Wissensstand.

Zero Day Exploit Schwachstellen, die durch die Entdeckung durch eine Malware von der Öffentlichkeit entdeckt werden.

7.1 Angriff

- *(Cache-)Poisoning* – mittels falscher Informationen Datenströme umleiten: DNS-Poisoning, ARP-Spoofing, DHCP-Poisoning
- *Erpressung* – Typischerweise *Ransomware*; Existiert auch über den Verkauf von gebrauchten Festplatten
- *Privilege Escalation* – Sicherheitsbarrieren umgehen; Übernahme von Rechten hochprivilegierter Prozesse oder durch Überwindung von Sicherheitsbarrieren

Webbasierte Angriffe Häufigste Angriffe sind: 1) Drive-by-Infektionen, 2) Injection Attacken (Cross-Site-Scripting, SQL-Injektion)

- *Persistent* – In einem Blog, Kommentarfunktion, Gästebuch; e.g. über Upload von Files, Verlinkung von ungeprüften Quellen
- *Non-Persistent* – Einfachste Attacke; untergeschobener Link meistens im Zusammenhang mit einem Phishing-Mail
- *DOM-Based* – Untergeschobener Link wird aus dem DOM des Browsers geladen

Directory / Path-Traversal-Attack Lokale Files werden ausserhalb des Server-Roots geladen und übermittelt über 1) Backreferencing, 2) Logische Links.

CVSS Scores Geben den Schweregrad einer gefundenen Schwachstelle an

- *Basis Metrik* – Grundsätzliches über Exploit
- *Zeitbasierte Metrik* – Ändern sich über die Zeit
- *Umfeld-Metrik* – Firmenspezifisch / Setup-Spezifisch

8 Malware

8.1 Angriffe Entdecken

Indikatoren Zwei Arten: 1) Indikatoren für einen Angriff (IOA) und 2) Indikatoren für eine Kompromittierung. Ist grundsätzlich nicht das Selbe.

Häufigste Gefahr: Verschlüsselnde Ransomware – verschlüsselt alles bevor sie entdeckt werden kann. Höchste Gefährdung.

Die Indikatoren sind abhängig vom eigenen System.

Indikatoren für einen Angriff (IOA) Bei Angriff: *Sammeln* und dann *Analysieren*.

- Versagen eines Dienstes oder Zerstörung eines Assets
- Geänderte Leistung eines Subsystems
- Geändertes Verhalten eines Systems oder Subsystems

Sind nur Indikatoren und müssen nicht erfolgreich sein.

Indikatoren für eine Kompromittierung (IOC) Bei Angriff: *Isolieren* (Netzwerktechnisch nicht mehr erreichbar machen – unabhängig machen von anderen Systemen) und dann Systemzustand *einfrieren* (Snapshot erstellen und sicherstellen; Festplatten stromlos machen; Ausschalten und nicht herunterfahren – e.g. Stromstecker ziehen).

GANZ WICHTIG: Was muss bei einer Kompromittierung eines Systems gemacht werden?

- Andere Zustände eines Systems oder Subsystems
- Geänderte Leistung eines Subsystems
- Geändertes Verhalten eines Systems oder Subsystems
- Wertänderung von Assets durch externe Einflüsse

Bei Kompromittierung muss sofort gehandelt werden, da der Schaden immer grösser werden kann.

Firewall als Angriffsdetektion (und Verteidigung) Zum Netzwerkschutz werden Firewalls häufigst verwendet.

Aufgaben Firewall:

- Portfiltering
- Netzwerkverkehr untersuchen nach verdächtigen Aktivitäten
 - Injektions-Attacken
 - Fehlschlagende Loginversuche
 - Virensignaturen

Firewall macht ihre Aufgaben so gut sie kann und benötigt viele Kenntnisse über die Infrastruktur. Schwach, wenn Firewall nicht genug über Infrastruktur weiss. Meisten Firewalls machen nur Portfiltering.

Firewall muss nicht nur eingerichtet werden. Die Logs auswerten etc. muss auch gemacht werden – Expertenwissen benötigt. In der FW-Administration müssen die korrekten Personen angeheuert sein, damit diese korrekt behandelt wird.

Monitoring als Angriffsdetektion Monitoring erfasst typischerweise IOC und IOA.

- Typische IOAs
 - Logs
 - Fehlschlagende Loginversuche
 - Anormale Aktivitäten – Prozesse, die nicht mehr laufen / neu hinzugekommen sind
- Typische IOCs
 - Vorhandene files mit SUID-Flags – Wenn das File ausgeführt wird, läuft es mit den Rechten des File-Eigentümers
 - Treiber, deren Prüfsumme unbekannt sind – Pakete enthalten: Files, Install-Script, Prüfsummen für jedes File (prüfen ob das Paket korrekt entpackt wurde); Files identifizieren, welche (nicht) aus einer vertrauenswürdigen Quelle kommen

Privilege-Escalation: Berschaffung von höheren Privilegien

Intrusion Detection System Laufen und halten Angriffe innerhalb eines Sicherheitsperimeters fest. Dienen dazu, verdächtige Gegebenheiten aufzudecken. Wird zum *Perimeterschutz* verwendet. Ist passiv – beobachtet und meldet es weiter.

untersucht Informationsquellen nach IOCs oder IOAs; Intrusion Prevention System (IPS) hat zusätzlich Möglichkeit Einbrüche zu unterbinden.

Systeme sind Query-Basierte Firewalls.

Honeypot als Angriffsdetektion Systeme, welche Angreifende gezielt anziehen (e.g. Server, welcher gezielt eine Vulnerabilität hat). Jeder Zugriff auf einen Honeypot ist ein Indikator im Perimeter drin.

Wenn der Honeypot nur als ein solcher konfiguriert ist, kann jeder Zugriff auf den Honeypot als versuchter Angriff gewertet werden.

Tarpit als Angriffsdetektion Normale Dienste oder Geräte, welche Angreifer identifizieren und diese möglichst lange beschäftigen (e.g. wird immer langsamer) – Ziel ist es Angreifer möglichst lange aufzuhalten.

Muss möglichst aussehen, wie ein echtes System.

Beispiele für Tarpit:

- IP-Level-Tarpit – es wird simuliert, dass in einem Netz, alle IP-Adressen von Hosts besetzt sind; alle Hosts müssen betrachtet werden
- SMTP-Tarpit – es wird gewartet, um Antworten zu senden; wenn nächste Anfrage früher kommt als Antwort, kann davon ausgegangen werden, dass Angreifer da ist
- Harvester-Tarpit – (Harvester-Bots beeinträchtigen – e.g. Email-Harvester) Auf Website Link platzieren, welcher ganz viele E-Mail-Adressen anzeigt; Adressen werden unendlich viel zufällig generiert – mit robot.txt kann definiert werden, was bots dürfen (Bots binden, wenn sie sich nicht daran halten)

Aufwand für einen potentiellen Angreifer wird massiv erhöht, ohne legitime Benutzer zu beeinträchtigen.

8.2 Das Vertrauen in die Plattform

Es muss sichergestellt werden, dass die Plattformen vertrauenswürdig sind. Dafür werden *Trusted Plattform Modules* (TPM) verwendet. Mit einer TPM kann ein System verschlossen werden (e.g. Spielekonsolen). TPM kann (und wird) misbraucht werden, um das Digital-Rights-Movement zu umgehen.

Einleitung Es kann eine MITM-Attacke über die Plattform ausgeführt werden; Plattform muss zuerst überprüft werden, bevor ihr vertraut werden kann

Aufgaben eines Trusted Plattform Modules Beim Starten eines Computers wird das BIOS/UEFI geladen. Das TPM sorgt dafür, dass die CPU noch nicht gestartet wird. Das TPM berechnet Prüfsumme von UEFI, nimmt dessen Signatur überprüft sie. Wenn alles okay ist, kann das System gestartet werden.

Im TPM befindet sich:

- CA-Speicher, welcher nicht überschrieben werden kann
- hat eine HMAC (Möglichkeit zum Hashs erstellen)
- kann Signieren / Verify / decrypt / encrypt (RSA, ECC...)
- hat sichere Keystore (kann nicht ausgelesen werden)

TPM stellt sicher, dass keine MITM-Attacke ausserhalb des TPM möglich ist. Es stellt sicher, dass externe Kommunikation bis zur Plattform-Grenze immer sicher ist und dass aller Software vertraut werden kann.

Fähigkeiten der meisten TPM

Unified Extensible Firmware Interface (UEFI) ist ein teilweiser Ersatz für das BIOS. Es bietet:

- Vorteile
 - Unterstützung für Disks >2TB
 - Prozessorunabhängige Plattform für Treiber
 - Modularer Aufbau (Inhalt des UEFI kann modifiziert werden)
 - Integrierte Unterstützung eines Bootmanagers (Anstelle der Bootsektoren)
- Nachteile
 - Inhalt vom UEFI kann modifiziert werden (Sicherheitstechnisch ein Nachteil)
 - Wenn nur noch speziell signierte Software gestartet werden kann (secure boot), kommt einem "Vendor Lock" gleich.

Sicherheit einer virtuellen Trusted Plattform Bei der Rowhammer-Attacke wird die Volatilität der Bits im dRAM ausgenutzt. Somit könnte ein Bit geändert werden im dRAM und diese ausgenutzt werden. Es kann mit Glück der private key für ssh verändert werden und somit die Sicherheit dessen verkleinern. Ein TPM kann einen solchen HW-Angriff nicht abwehren.

Trusted Plattform vs. TPM TPM und TP sind nicht dasselbe. Ein TPM stellt eine sichere Plattform in einem nicht vertrauenswerten System zur Verfügung für Teilaufgaben, während TP eine vollständige Plattform ist, die eine Dienstleistung einem Benutzer zur Verfügung stellt.

SIM-Karte eines Mobiltelefons stellt sicher, dass die Identität des Telefons nicht von einer Drittpartei übernommen werden kann.

9 Sicherheit von HW / Plattform / Virtualisierung

9.1 Sicherheit im OS

Alles was ein Betriebssystem bietet, tut, verbietet, oder gezielt unterlässt um die Schutzziele zu gewährleisten.

Wahrnehmbare Sicherheitsaspekte im Betriebssystem

- Login
- Dateirechte – OS unterschiedlich
- Prozessrechte – Bei allen Systemen ungefähr gleich sichtbar
- Signierung – Viele Treiber können nicht installiert werden, wenn sie nicht signiert sind
- Verschlüsselung
- Warnung bei Gefahrenmomenten – Warnungen wenn ein potenziell gefährliches Programm ausgeführt wird

Viele Betriebssysteme bieten auch eine Gefahr für die Privatsphäre; Meistens verbergen sich diese hintern schönen "Features" – e.g. Cortana hört immer zu, Handschrifterkennung, Texterkennung, *Update-Empfehlung*.

Updateempfehlung Windows empfängt von einem Gerät, welche Software (Version) installiert ist und sendet eine Update-Version. Client exponiert sich somit und Microsoft weiss, welche Software (-Version) auf welchem Gerät installiert ist.

Es ist schwierig etwas vor dem Hersteller des OS zu verbergen. Daten werden gesammelt, und in Features gepackt, welche den Nutzenden als "schöne" Funktionen angepriesen werden.

9.2 Gegenmassnahmen

Nicht direkt sichtbare Sicherheitsaspekte des OS

- *CPU-Domains* – Je nach Programm, das läuft, läuft die CPU auf verschiedenen Berechtigungsstufen (Ring 0 alles verfügbar Kernel Mode, Ring 3 User Mode)

- *Sandboxes* – In einer Sandbox ist man alleine; isolierten Bereich, innerhalb dessen jede Massnahme keine Auswirkung auf die äussere Umgebung hat
- *Protected-Mode* – Möglichkeit um Prozesse voneinander abgrenzen; einzelner Speicherbereich für Prozess definieren (Prozess wird abgeschaltet, wenn Speicher überwachsen wird)
- *ESP* – Daten in Speicherbereich, welche nicht als Code ausgeführt werden dürfen
- *ASLR* – Sorgt dafür, dass der Inhalt des Speichers zufällig angeordnet wird (gegen Buffer Overflows)
- *SMEP* – Kernel-Mode-Programme dürfen im Userland-Speicher keinen Code ausführen (gegen zweite Phase von Buffer Overflows)
- *SMAP* – Kernel-Mode-Programme können nicht auf Userland-Speicher zugreifen
- *Canaries* – Detektieren, wenn die Buffergrenzen zwischen Variablen verletzt wurden; nicht gut, wenn diese Prozesse abgeschossen werden
- *Watchdogs* – Überwachen die korrekte Funktion von Betriebssystem-Diensten; NMI – non-maskeable-interrupts

Einfache Erhöhung der Rechte *Windows* – Mit User Account Control (UAC) kann punktuell ein Prozess mit Administratorrechten ausgeführt werden. Das Programm wechselt in einen geschützten Modus und fragt den Benutzer, ob er sicher ist. Alle Userland-Prozesse werden eingefroren, damit nicht durch ein Programm automatisch angenommen werden kann.

Linux – Damit die root-Rechte möglichst fein vergeben werden können, bietet Linux "sudo" an. Es kann genau festgelegt werden, welcher Benutzer welche Programme ausführen kann.

Gefahren des Suchpfades Ein Verzeichnis in einem Suchpfad sollte nicht durch andere Benutzer beschreibbar sein, weil andere Benutzer sonst Programme, die im Suchpfad sind, übersteuern könnten und damit die Rechte des Benutzers kapern.

Verzeichnisse können mit dem selben Namen wie Systemprogramme enthalten, diese werden dann möglicherweise ausgeführt.

Hardening Vorgang des Absichern des Betriebssystems. Hardening-Aktivitäten:

- Patchen mit speziellen, sicherheitsfördernden Technologien
- Einführen von Passwortrichtlinien oder spezieller Authentisierung
- Deaktivieren und entfernen nicht benötigter Programme, Benutzer und Services
- Einschränken der lokalen Benutzer-Rechte
- Das verwenden einer lokalen Firewall

Nicht dazu gezählt (dennoch sicherheitsrelevant): 1) Einspielen sicherheitsrelevanter Patches, 2) Ausbilden von Benutzern

10 Sicherheit von OS / Software

10.1 Sicherheit bei Software

Häufige Fehler bei Software

- *Programme haben zu hohe Rechte* – z.B. weil mit admin-rechten entwickelt wurde
- *Programme prüfen Berechtigungen unzureichend* – Authentication-Bypass; Bei API wird der Schutz entfernt
- *Programme schützen ihre Funktionen unzureichend* – z.B. Frontend prüft / nicht redundante Implementierung im Backend

Prozesse, welche root-Berechtigungen brauchen: Start mit root, sobald gestartet, werden root-Berechtigungen entfernt (e.g. Apache Server) – Main-Prozess

Generelle Fehler von Programmierern

- Sicherheitsfeatures (UAC, Virenschutz) aus, weil sie hinderlich sind
- Zeitdruck – Features sind Endkunden meistens wichtiger als Sicherheit
- Wenig Verständnis für Sicherheit
- Verantwortung an Sicherheitsscanner delegieren – viele false-positive; Sicherheit nicht garantiert
- Sicherheits-Audit erst am Schluss, wenn das ganze Programm schon entwickelt wurde
- Programmierer denken in *Funktionen*, nicht in *Sicherheitsstufen*
- Verwendung von APIs oder Frameworks, ohne sich zu fragen, wer diese sonst ausführen kann – billion-laughs-Attake; xml external-entity Attacke

Sichere Sprachen Es gibt Sprachen, bei denen bestimmte Fehler wahrscheinlicher sind, aber keine Sprache ist sicher – jede Sprache kann sicher oder unsicher programmiert werden.

10.2 Watchdogs

Watchdogs 1) gewährleisten die Verfügbarkeit von Systemen und Programmen, 2) ergreifen korrigierende Massnahmen bei Fehlfunktion der zu überwachenden Komponente 3) werden anhand deren Fehlererkennung unterschieden:

- *Time-Out-Watchdog* – Watchdog meldet sich bei zu überwachender Software; wenn nach bestimmter Zeit (time-out) keine Antwort, wird (je nachdem) das System neu gestartet
- *Window-Watchdog* – Watchdog meldet sich bei zu überwachender Software; Antwort nach gewissem Zeitfenster erwartet; keine oder zu späte Antwort löst Massnahmen aus; Zu früh gesendete Antworten werden ignoriert
- *Smart-Watchdog* – Löst Operation aus und bewertet Ausgabe und handelt entsprechend

Reflektor-Attacken Angreifer sendet Daten an einen oder mehrere Empfänger und diese generieren Verkehr, der an das anzugreifende Ziel gesendet wird. Somit 1) verbergen sie die wahren Täter und 2) wird die Bandbreite der Attacke erhöht.

Smurf-Attake: Ping-Paket wird mit gefälschter Absender-Adresse an eine Broadcast-Adresse eines Netzwerkes gesendet. Schlecht konfigurierte Clients geben darauf Antwort, indem sie ein Paket an den vermeintlichen Absender (eigentliches Ziel) senden

Drive-By-Infektionen Dateien werden "unbeabsichtigt" auf einen Client heruntergeladen. Je nach Ort und Ausgestaltung werden diese Programme ausgeführt und setzen sich auf dem Client fest; Webbrowser – Verbreitung über angepasste Webseiten (e.g. ungepatchtes CMS)