MACsec

- MACsec ist ein Layer-2-Protokoll, das auf GCM-AES-128 basiert, um Integrität und Vertraulichkeit zu gewährleisten, und über Ethernet betrieben wird.

- Es kann den gesamten Datenverkehr in einem LAN, einschließlich DHCP und ARP, sowie den Datenverkehr von Protokollen höherer Ebenen sichern.

- Es ist eine Erweiterung von 802.1X, die sicheren Schlüsselaustausch und gegenseitige Authentifizierung für MACsec-Knoten bietet.

- IPsec (ein Sicherheitsprotokoll der Schicht 3) und TLS (ein Sicherheitsprotokoll der Schicht 4) bieten unterschiedliche Garantien und können je nach Anwendungsfall besser passen.

MACsec vs IPsec

- MACsec und IPsec arbeiten auf verschiedenen Netzwerkebenen.
- IPsec arbeitet mit IP-Paketen auf Schicht 3, während MACsec auf Schicht 2 mit Ethernet-Frames arbeitet.
- MACsec kann den gesamten DHCP- und ARP-Verkehr schützen, den IPsec nicht sichern kann.
- Auf der anderen Seite kann IPsec über Router hinweg funktionieren, während MACsec auf ein LAN beschränkt ist.

Public-Key-Zertifikat

- Überall wo asymmetrische Verschlüsselung eingesetzt wird, muss der öffentliche Schlüssel des Kommunikationspartners bekannt sein.
- Geht es dabei nur um die Sicherung der Vertraulichkeit, wird ein unbekannter oder falscher öffentlicher Schlüssel nur die vertrauliche Kommunikation sabotieren, was aber sofort bemerkt wird.
- Anders ist es, wenn die asymmetrische Verschlüsselung einen Kommunikationspartner authentisieren soll.

Public-Key-Zertifikat

- Wenn man den öffentlichen Schlüssel einer Person zum Verschlüsseln benutzt, dann sollte man auch sicher sein, dass dieser Schlüssel tatsächlich zur angegebenen Person gehört.

- Ein Public-Key-Zertifikat dient dazu, die Zugehörigkeit eines öffentlichen Schlüssels zu einem bestimmten Eigentümer zu bestätigen.

Public-Key-Zertifikat

- Durch ein Public-Key-Zertifikat können Nutzer eines asymmetrischen Kryptosystems den öffentlichen Schlüssel einer Identität (z. B. einer Person, einer Organisation oder einem IT-System) zuordnen.

- Ein Public-Key-Zertifikat enthält in der Regel eine ganze Reihe von Informationen, u a.:
  - den zu bestätigenden öffentlichen Schlüssel
  - den Eigentümer des Schlüssels (subject)
  - den Aussteller des Zertifikats
  - die benutzten kryptografischen Verfahren
  - die Gültigkeitsdauer des Zertifikats
  - eine digitale Signatur des Ausstellers zur Bestätigung aller Informationen

Public-Key-Zertifikat

- Der Aussteller eines Zertifikates wird als Zertifizierungsinstanz bezeichnet (certificate authority CA).
- Die Zertifizierungsinstanz sollte von einer vertrauenswürdigen Organisation oder Stelle (z. B. eine Behörde) betrieben werden, damit die Anwender sich auf die in den Zertifikaten enthaltenen Informationen verlassen können.
- Durch die digitale Signatur über das Zertifikat lässt sich die Authentizität und Integrität des digitalen Zertifikates überprüfen.
- Für diese Prüfung wird jedoch wiederum eine Zuordnung des Signaturschlüssels des Ausstellers zu seiner Identität, d. h. ein weiteres Zertifikat, benötigt.
- Diese Hierarchie von Zertifikaten bildet eine Public-Key-Infrastruktur (PKI).

Public-Key-Zertifikat

- Zertifikate haben bestimmte Gültikeitsdauer.
- Zertifikate für Schlüssel, die nicht mehr sicher sind, können und sollten vorzeitig gesperrt und die Sperrinformationen veröffentlicht werden.
- Vertrauenswürdigkeit eines Zertifikates hängt auch stark davon ab, wie schnell es gesperrt werden kann und wie zuverlässig und zeitnah die Sperrung veröffentlicht wird.
- Sperrungen werden über Certificate Revocation List (CRL), veröffentlicht.

Public-Key-Zertifikat

- Arten von Zertifikaten:
  - TLS/SSL Server Zertifikate: Ein Server muss beim erstmaligen Verbindungsaufbau ein Zertifikat vorlegen. Ein Client, der eine Verbindung zu diesem Server herstellt, führt den Validierungsalgorithmus für den Zertifizierungspfad aus
    - Der Eigentumer (subject) des Zertifikats stimmt mit dem Hostnamen überein.
    - Das Zertifikat ist von einer vertrauenswürdigen Zertifizierungsstelle signiert.

Public-Key-Zertifikat

- Arten von Zertifikaten:
  - TLS/SSL Client Zertifikate
  - E-Mail-Zertifikat: Im S / MIME-Protokoll für sichere E-Mails müssen Absender herausfinden, welcher öffentliche Schlüssel für einen bestimmten Empfänger verwendet werden soll. Sie erhalten diese Informationen aus einem E-Mail-Zertifikat.
  - Codesignaturzertifikate
  - Stammzertifikat (Root): Ein selbstsigniertes Zertifikat zum Signieren anderer Zertifikate. Wird auch manchmal als Vertrauensanker bezeichnet.

Public-Key-Zertifikat

- Zwischenzertifikate (Intermediate): Ein Zertifikat zum Signieren anderer Zertifikate. Ein Zwischenzertifikat muss von einem anderen Zwischenzertifikat oder einem Stammzertifikat signiert sein.

- Endgerät- oder Leaf-Zertifikat: Jedes Zertifikat, das nicht zum Signieren anderer Zertifikate verwendet werden kann. Beispielsweise sind TLS / SSL-Server- und -Client-Zertifikate, E-Mail-Zertifikate, Codesignaturzertifikate Endgerätezertifikate.

Public-Key-Infrastruktur (PKI)

- Ist eine Reihe von Rollen, Richtlinien, Hardware, Software und Verfahren, die zum Erstellen, Verwalten, Verteilen, Verwenden, Speichern und Sperren digitaler Zertifikate und zum Verwalten der Verschlüsselung mit öffentlichen Schlüsseln erforderlich sind.

- Der Zweck einer PKI besteht darin, die sichere elektronische Übertragung von Informationen für eine Reihe von Netzwerkaktivitäten wie E-Commerce, Internet-Banking und vertrauliche E-Mails zu ermöglichen.

- Dies ist für Aktivitäten erforderlich, bei denen einfache Kennwörter eine unzureichende Authentifizierungsmethode darstellen und ein strengerer Nachweis erforderlich ist, um die Identität zu bestätigen.

Public-Key-Infrastruktur (PKI)

Eine PKI besteht aus:

- Zertifizierungsstelle (Certificate Authority, CA): die die digitalen Zertifikate speichert, ausstellt und signiert.
- Registrierungsstelle (Registration Authority, RA): die die Identität von Entitäten überprüft, die die Speicherung ihrer digitalen Zertifikate bei der Zertifizierungsstelle anfordern.
- Verzeichnisdienst (Directory Service): Ein sicherer Ort, an dem Schlüssel gespeichert und indiziert werden.
- Validierungsdienst (Validation Authority, VA): Ein Dienst, der die Überprüfung von Zertifikaten in Echtzeit ermöglicht wie OCSP.
- Certificate Policy: In diesem Dokument beschreibt die PKI ihr Anforderungsprofil an ihre eigene Arbeitsweise. Es dient Dritten zur Analyse der Vertrauenswürdigkeit und damit zur Aufnahme in den Browser.

Digitale Zertifikate typen

zwei Haupttypen:

- Domain Validated Certificates (DVC)
- Extended validation Certificates (EVC)

- Der Unterschied zwischen den beiden Typen besteht im Grad des Vertrauens in das Zertifikat, das mit einer strengeren Validierung kommt.

- Die bereitgestellte Verschlüsselungsstufe ist identisch.

Digitale Zertifikate typen

Domain Validated Certificates (DVC):

- ist ein digitales X.509-Zertifikat, das normalerweise für TLS (Transport Layer Security) verwendet wird, bei dem die Identität des Antragstellers durch den Nachweis einer gewissen Kontrolle über eine DNS-Domäne überprüft wurde.
- Der Validierungsprozess ist in der Regel vollautomatisch und stellt die günstigste Form des Zertifikats dar.
- sind ideal für die Verwendung auf Websites, die Inhalte bereitstellen, und werden nicht für vertrauliche Daten verwendet.

Digitale Zertifikate typen

Extended validation Certificates (EVC):

- zertifikate, deren Ausgabe an strengere Vergabekriterien gebunden ist. Dies bezieht sich vor allem auf eine detaillierte Überprüfung des Antragstellers durch die Zertifizierungsstelle.

- Genutzt werden die Zertifikate meist, um Webanwendungen per HTTPS zu sichern und den Anwendern vor dem Hintergrund von Phishing-Angriffen eine zusätzliche Sicherheit zu geben, etwa beim Online-Banking.

- Sie sind in der Regel teurer als DVC, da sie manuell validiert werden müssen.

Digitale Zertifikate typen

EVC-Vergabekriterien:

- Um EV-SSL-Zertifikate ausstellen zu dürfen, müssen sich die Zertifizierungsstellen selbst einer Überprüfung unterziehen.
- Feststellung der Identität und der Geschäftsadresse des Antragstellers.
- Sicherstellung, dass der Antragsteller ausschließlicher Eigentümer der Domain ist oder eine exklusive Nutzungsberechtigung hat.
- Ein EVC kann für Behörden, Kapitalgesellschaften, Personengesellschaften, …. Ausgestellt werden.

Digitale Zertifikate typen

- Im Allgemeinen ist ein Zertifikat gültig für die Verwendung für einen einzelnen fully qualified domain name (FQDN).
- Beispiel ein Zertifikat, das für die Verwendung für www.mydomain.com erworben wurde kann nicht für die Verwendung für mail.mydomain.com oder www.otherdomain.com verwendet werden.
- Wenn wir jedoch mehrere Subdomains sowie den Hauptdomainnamen sichern müssen, können wir ein Wildcard-Zertifikat erwerben.
- Ein Wildcard-Zertifikat deckt alle Subdomains unter einem bestimmten Domainnamen ab.

Digitale Zertifikate typen

- Beispielsweise kann ein Wildcard-Zertifikat für *.mydomain.com verwendet werden für:
  - mail.mydomain.com
  - www.mydomain.com
  - ftp.mydomain.com
- Es kann aber nicht verwendet werden, um sowohl mydomain.com als auch myotherdomain.com zu sichern.

Digitale Zertifikate typen

- Um mehrere verschiedene Domainnamen in einem einzigen Zertifikat abzudecken, müssen wir ein Zertifikat mit SAN (Subject Alternative Name) erwerben.

- In der Regel können wir damit zusätzlich zum Hauptdomänennamen vier weitere Domänennamen sichern. Beispielsweise könnten wir dasselbe Zertifikat für Folgendes verwenden:
  - www.mydomain.com
  - www.mydomain.org
  - www.mydomain.net
  - www.mydomain.co
  - www.mydomain.co.de

Programmsicherheit

- Wir wissen, dass Sicherheit ein gewisses Maß an Vertrauen voraussetzt, dass das Programm die erwartete Vertraulichkeit, Integrität und Verfügbarkeit gewährleistet.

- Wie können wir aus der Sicht eines Programms oder eines Programmierers eine Softwarekomponente oder ein Codefragment betrachten und deren Sicherheit bewerten?

- Eine Möglichkeit, die Sicherheit oder Qualität zu bewerten, besteht darin, die Benutzer aufzufordern, die Merkmale von Software zu benennen, die zur allgemeinen Sicherheit beitragen (unterschiedliche Antworten).

Programmsicherheit

- Frühe Arbeiten zur Computersicherheit basierten auf dem Paradigma " penetrate and patch ".
  - Der Druck, ein bestimmtes Problem zu beheben, ermutigte dazu, sich nur auf den Fehler selbst und nicht auf dessen Kontext zu konzentrieren.
  - Der Fehler hatte häufig nicht offensichtliche Nebenwirkungen an anderen Orten als dem unmittelbaren Bereich des Fehlers.
  - Die Behebung eines Problems verursachte häufig einen Fehler an einer anderen Stelle.
  - Der Fehler konnte nicht ordnungsgemäß behoben werden, da die Systemfunktionalität oder -leistung darunter leiden würde

Programmsicherheit

Arten von Fehlern:

- Vorsätzliche Fehler
  - Bösartig (malicious)
  - nicht bösartig (nonmalicious)

- unbeabsichtigte Fehler
  - Validierungsfehler
  - Domänenfehler
  - unzureichende Identifizierung und Authentifizierung
  - Grenzbedingungsverletzung (boundary condition)

Programmsicherheit

nicht bösartig (nonmalicious) Programmfehler:

Pufferüberlauf (Buffer Overflows):

- Ein Pufferüberlauf entspricht dem Versuch, zwei Liter Wasser in einen Ein-Liter-Behälter zu füllen.
- Ein Puffer (oder ein Array oder eine Zeichenfolge) ist ein Bereich, in dem Daten gespeichert werden können. Ein Puffer befindet sich im Speicher. Da der Speicher endlich ist, ist die Kapazität eines Puffers endlich.
- Aus diesem Grund muss der Programmierer in vielen Programmiersprachen die maximale Größe des Puffers deklarieren, damit der Compiler diesen Speicherplatz freigeben kann.

Programmsicherheit

- Ein Pufferüberlauf tritt auf, wenn mehr Daten in einen Puffer fester Länge gestellt werden, als der Puffer verarbeiten kann.

- Die zusätzlichen Informationen, die irgendwo abgelegt werden müssen, können in den angrenzenden Speicherbereich überlaufen und die in diesem Bereich gespeicherten Daten beschädigen oder überschreiben.

- Dieser Überlauf führt normalerweise zu einem Systemabsturz, bietet einem Angreifer jedoch auch die Möglichkeit, beliebigen Code auszuführen oder die Codierungsfehler zu manipulieren, um böswillige Aktionen auszulösen.

Programmsicherheit

Pufferüberlauf-Angriff Beispiel[code]:

```
#include <stdio.h>
#include <string.h>

int main(void)
{
   char buff[15];
   int pass = 0;

   printf("\n Enter the password : \n");
   gets(buff);

   if(strcmp(buff, „InfoSec"))
   {
      printf ("\n Wrong Password \n");
   }
   else
   {
      printf ("\n Correct Password \n");
      pass = 1;
   }

   if(pass)
   {
     /* Now Give root or admin rights to user*/
      printf ("\n Root privileges given to the user \n");
   }

   return 0;
}
```

- Betrachten wir ein Programm, das ein Benutzerkennwort anfordert, um dem Benutzer Zugriff auf das System zu gewähren.
- In dem Code werden durch das richtige Kennwort dem Benutzer die Root-Berechtigungen erteilt.
- Wenn das Kennwort falsch ist, werden dem Benutzer vom Programm keine Berechtigungen erteilt.

Programmsicherheit

Pufferüberlauf-Angriff Beispiel:

- Lassen Sie uns das Programm mit dem richtigen Passwort ausführen:

```
$ ./bfrovrflw

 Enter the password :
InfoSec

 Correct Password

 Root privileges given to the user
```

- Das funktioniert wie erwartet. Die Passwörter stimmen überein und Root-Rechte werden vergeben.

- Aber wissen Sie, dass in diesem Programm die Möglichkeit eines Pufferüberlaufs besteht?

Programmsicherheit

Pufferüberlauf-Angriff Beispiel:

- Die gets () - Funktion überprüft die Array-Grenzen nicht und kann sogar Zeichenfolgen schreiben, deren Länge größer ist als die Größe des Puffers, in den die Zeichenfolge geschrieben wird.

```
$ ./bfrovrflw

 Enter the password :
hhhhhhhhhhhhhhhhhhhh

 Wrong Password

 Root privileges given to the user
```

- Im obigen Beispiel hat das Programm auch nach Eingabe eines falschen Kennworts so funktioniert, als hätten Sie das richtige Kennwort eingegeben.

Programmsicherheit

Pufferüberlauf-Angriff Beispiel:

- Der Angreifer hat eine Eingabe mit einer Länge gegeben, die größer ist als der Puffer, wodurch ein Pufferüberlauf entsteht, der den Speicher der Ganzzahl "pass" überschreibt.

- Trotz eines falschen Passworts wurde der Wert von "pass" ungleich Null, und daher wurden dem Angreifer Root-Berechtigungen gewährt.

- Um einen Pufferüberlauf zu vermeiden, sollten Entwickler von C / C ++ - Anwendungen Standardbibliotheksfunktionen wie gets, scanf und strcpy,die die Grenzen nicht prüfen meiden.

Programmsicherheit

Schadcode (Malicious Code):

- Schädlicher Code kann alles tun, was jedes andere Programm kann.Schreiben einer Nachricht auf einem Computerbildschirm, Beenden eines laufenden Programms, Erzeugen eines Sounds oder Löschen einer gespeicherten Datei.

- Bösartiger Code kann auch vorerst überhaupt nichts tun, Es kann so gepflanzt werden, dass es unentdeckt schlummert, bis ein Ereignis den Code zum Handeln veranlasst.

- Der Auslöser kann eine Uhrzeit oder ein Datum sein, ein Intervall (zum Beispiel nach 30 Minuten), ein Ereignis (zum Beispiel, wenn ein bestimmtes Programm ausgeführt wird).

Programmsicherheit

Arten von schädlichem Code:

- Bösartiger Code ist die allgemeine Bezeichnung für unerwartete oder unerwünschte Auswirkungen in Programmen oder Programmteilen, die von einem Agenten verursacht werden, der vorsätzlich Schaden anrichten will.

- Diese Definition schließt aus unbeabsichtigte Fehler, den Zufall, bei dem sich zwei gutartige Programme zu einem negativen kombinieren bewirken.

- Der Agent ist der Verfasser des Programms oder die Person, die dessen Verbreitung veranlasst.

Programmsicherheit

Virus:

- ist ein sich selbst verbreitendes Computerprogramm, das bösartigen Code an andere nicht bösartige Programme weitergeben kann.
- Der Begriff "Virus" wurde geprägt, weil das betroffene Programm wie ein biologisches Virus wirkt: Es infiziert andere gesunde, indem es sich an das Programm bindet und es entweder zerstört oder mit ihm koexistiert.
- Ein Virus kann entweder vorübergehend (transient) oder ansässig (resident) sein.

Programmsicherheit

Virus:

- Ein vorübergehendes Virus hat ein Leben, das vom Leben seines Wirtes abhängt. Der Virus wird ausgeführt, wenn das angehängte Programm ausgeführt wird, und wird beendet, wenn das angehängte Programm beendet wird (Während seiner Ausführung hat der vorübergehende Virus möglicherweise seine Infektion auf andere Programme übertragen).

- Ein ansässiges Virus lokalisiert sich im Speicher, dann kann es aktiv bleiben oder als eigenständiges Programm aktiviert werden, auch nachdem das angehängte Programm beendet wurde.

Programmsicherheit

Trojanisches Pferd:

- Ein Trojanisches Pferd bezeichnet man ein Computerprogramm, das als nützliche Anwendung getarnt ist, im Hintergrund aber ohne Wissen des Anwenders eine andere Funktion erfüllt.
- Trojanische Pferde können über jeden Weg auf einen Computer gelangen, Datenträger, Netzwerkverbindungen wie das Internet und E-Mails.
- Die Verbreitung des Trojanischen Pferdes erfolgt durch den Anwender des Computers selbst.
- Für die Verbreitung mittels bsp. E-Mails wird meistens ein Computerwurm verwendet, der das Trojanische Pferd transportiert.
- Der Trojaner selbst wird dadurch, dass er sich augenscheinlich verbreitet, jedoch nicht zu einem Virus.

Programmsicherheit

Logikbombe:

- ist ein Computerprogrammteil, der nach Eintreten bestimmter (logischer) Bedingungen schädliche Aktionen auslöst.
- Auslösende Bedingungen können etwa das Erreichen eines bestimmten Datums.
- Im Gegensatz zu einem Virus kann eine Logikbombe nicht per se andere Dateien infizieren bzw. sich selbst reproduzieren. Umgekehrt sind die Bedingungen, die ein Virus oder einen Wurm sich verbreiten lassen, nicht spezifisch genug, um als Logikbombe bezeichnet zu werden.

Programmsicherheit

Logikbombe (Beispiel):

- Der Stuxnet-Virus bestand aus mehreren Teilen, u. a. auch aus einer logischen Bombe.
- Bedingungen z.B. Prüfung der Frequenz des Zentrifugenmotors zwischen 807 Hz und 1210 Hz und Datum in bestimmtem Bereich.
- Zentrifugen zur Uran-Anreicherung durch gezielte Frequenzveränderungen beschädigen.

Programmsicherheit

Backdoor:

- Eine Funktion in einem Programm, mit der jemand auf das Programm zugreifen kann, nter Umgehung der normalen Zugriffssicherung.
- Software (meist durch einen Trojaner heimlich installierte) , die einen entsprechenden Fernzugriff auf den Computer ermöglicht.
- Eine Variante besteht darin, in einem System fest vorgegebene, nur dem Ersteller des Systems bekannte Passwörter oder andere versteckte Funktionen einzubauen, die einen Zugriff ohne die sonst übliche Authentifizierung ermöglichen.

Programmsicherheit

Wurm:

- Ist ein Programm, das Kopien von sich selbst über ein Netzwerk verbreitet.
- Der Hauptunterschied zwischen einem Wurm und einem Virus besteht darin, dass ein Wurm über Netzwerke funktioniert und sich ein Virus auf jedem Medium ausbreiten kann.
- Darüber hinaus verbreitet der Wurm Kopien von sich selbst als eigenständiges Programm, wohingegen der Virus Kopien von sich selbst als Programm verbreitet, das an andere Programme angehängt oder in andere Programme eingebettet ist.

Programmsicherheit

Rabbit:

- Viren oder Würmer, die sich ohne Einschränkung selbst replizieren, mit der Absicht, einige Computerressourcen zu erschöpfen. Ein rabbit erstellt möglicherweise Kopien von sich selbst und speichert sie auf der Festplatte, um die Festplatte vollständig zu füllen.

Programmsicherheit

Wie verbreiten sich Viren:

- Es muss aktiviert werden, indem es ausgeführt wird.

- Beispielsweise das SETUP-Programm, das Sie auf Ihrem Computer gestartet haben. Möglicherweise werden Dutzende oder Hunderte anderer Programme aufgerufen, einige auf dem Distributionsmedium, andere befinden sich bereits auf dem Computer, andere im Speicher. Wenn eines dieser Programme einen Virus enthält, kann der Viruscode aktiviert werden.

- Es ist jedoch möglich, dass die Ausführung ohne menschliches Eingreifen erfolgt, z. B. wenn die Ausführung durch ein Datum oder den Ablauf einer bestimmten Zeitspanne ausgelöst wird.

- Anhängen an einer E-Mail-Nachricht.

Programmsicherheit

Infektionsarten:

- Companion Viren:
  - Infizieren nicht die ausführbaren Dateien selbst, sondern benennen die ursprüngliche Datei um und erstellen eine Datei mit dem ursprünglichen Namen, die nur das Virus enthält.
  - Oder sie erstellen eine Datei mit ähnlichem Namen, die vor der ursprünglichen Datei ausgeführt wird.
  - Unter MS-DOS gibt es beispielsweise Companion-Viren, die zu einer ausführbaren EXE-Datei eine versteckte Datei gleichen Namens mit der Endung „.com“ erstellen, die dann nur das Virus enthält.
  - Wird in der Kommandozeile von MS-DOS ein Programmname ohne Endung eingegeben, sucht das Betriebssystem zuerst nach Programmen mit der Endung „.com“ und danach erst nach Programmen mit der Endung „.exe“.

Programmsicherheit

Infektionsarten:

- Überschreibende Computerviren:
  - Die einfachste Form von Viren, wegen ihrer stark zerstörenden Wirkung aber am leichtesten zu entdecken.
  - Wenn ein infiziertes Programm ausgeführt wird, sucht das Virus nach neuen infizierbaren Dateien und überschreibt entweder die ganze Datei oder nur einen Teil mit einer benötigten Länge.
  - Die Wirtsdatei wird dabei irreparabel beschädigt und funktioniert nicht mehr oder nicht mehr korrekt, wodurch eine Infektion praktisch sofort auffällt.

Programmsicherheit

Infektionsarten:

- Prepender:
  - Diese Art von Computerviren fügt sich am Anfang der Wirtsdatei ein.
  - Beim Ausführen der Wirtsdatei wird zuerst das Virus aktiv, das sich weiterverbreitet oder seine Schadwirkung entfaltet.
  - Danach stellt das Virus im Arbeitsspeicher den Originalzustand des Wirtsprogramms her und führt dieses aus.
  - Außer einem kleinen Zeitverlust merkt der Benutzer nicht, dass ein Virus gerade aktiv wurde

Programmsicherheit

Infektionsarten:

- Appender:
  - Ein *Appender-Virus* fügt sich an das Ende einer zu infizierenden Wirtsdatei an und manipuliert die Wirtsdatei derart, dass es vor dem Wirtsprogramm zur Ausführung kommt.
  - Nachdem das Virus aktiv geworden ist, führt es das Wirtsprogramm aus, indem es an den ursprünglichen Programmeinstiegspunkt springt.
  - Diese Virusform ist leichter zu schreiben als ein Prepender, da das Wirtsprogramm nur minimal verändert wird und es deshalb im Arbeitsspeicher nicht wiederhergestellt werden muss.

Programmsicherheit

Infektionsarten:

- Verschleierung des Einsprungspunktes:
  - Viren, die diese Technik benutzen, suchen sich zur Infektion einen bestimmten Punkt in der Wirtsdatei, der nicht am Anfang oder am Ende liegt.
  - Da dieser Punkt von Wirt zu Wirt variiert, sind Viren dieses Typs relativ schwierig zu entwickeln, da unter anderem eine Routine zum Suchen eines geeigneten Infektionspunktes benötigt wird.
  - Der Vorteil für diesen Virentyp besteht darin, dass Virenscanner die gesamte Datei untersuchen müssten, um diese Viren zu finden.
  - im Gegensatz zum Erkennen von Prepender- und Appender-Viren, bei denen der Virenscanner nur gezielt Dateianfang und -ende untersuchen muss.
  - Sucht ein Virenscanner also auch nach diese Viren, benötigt er mehr Zeit. Wird der Virenscanner so eingestellt, dass er Zeit spart, bleiben diese Viren meist unentdeckt.