

E Angriffskonzepte, Systemschwachstellen

E Angriffskonzepte, Systemschwachstellen

Angriff: nicht-autorisierte Zugriff auf ein System

- Abwehrmaßnahmen
 - Authentisierung von Benutzern
 - abgesicherte Kommunikation zwischen Systemen (Verschlüsselung)
 - Vermeiden von Schwachstellen in Schnittstellenprogrammen (Buffer-Overflows, Fehlinterpretation von Parametern, ...)
- Ziele
 - nicht-autorisierte Benutzer fernhalten
 - Ausführung nicht-autorisierte Kommandos verhindern
 - nicht vorgesehene Verhalten von Programmen verhindern

1 Trojanische Pferde (2)

E.1 Angriffe von innerhalb des Systems

Beispiele

- Zinsberechnung
 - Programm zur Zinsabrechnung schreibt Rundungsreste einem Konto des Angreifers gut
- CAD-Demo
 - Demo-Programm durchsucht Festplatte nach Raubkopien und kodiert Funde in einem Bestellformular für Handbücher
- Gefälschte Rechnungen
 - Verbreitung meistens über Anhänge von Massen-E-Mails
 - enthalten → Viren oder Code, der weiteren Schadcode nachlädt
- Gefälschte Web-Seiten
 - Identitätsdiebstahl, Phishing
- Unterschieben manipulierter Werkzeuge bei schlechter PATH-Variable

E.1 Angriffe von innerhalb des Systems

E.1 Angriffe von innerhalb des Systems

- Angreifer hat erste Barriere überwunden oder ist Insider

1 Trojanische Pferde

- Programm, dessen implementierte Ist-Funktionalität nicht mit der angegebenen Soll-Funktionalität übereinstimmt
 - Programm erfüllt zwar die Soll-Funktion, besitzt aber zusätzliche, verborgene Funktionalität
 - ➔ Analogie zu dem hölzernen Pferd beim Kampf um Troja
 - Ziele:
 - öffnen von Hintertüren
 - Aufzeichnen / Manipulieren / Weitergeben von Daten
- Meist Manipulation von häufig benutzten Programmen
 - Editoren, Standard-Werkzeuge, Demo-Versionen von Produkten

1 Trojanische Pferde (3)

E.1 Angriffe von innerhalb des Systems

Abwehrmaßnahmen

- nur vertrauenswürdige Directories in Ausführungspfad (PATH)
- Signierte Anwendungen, Zertifikate für Web-Seiten
- Code-Inspektion
- Anwendungen möglichst immer mit minimalen Rechten ausführen
- keine unbekanntenen Anwendungen ausführen
- Sensible Daten (Passwörter, PINs, TANs, ...) möglichst nur auf externen Medien (Smartcard, verschlüsselter USB-Stick) speichern

2 Login-Spoofing

E.1 Angriffe von innerhalb des Systems

- Benutzer wird durch nachgebaute Login-Maske dazu verleitet, seine Login-Daten einzugeben
- Variante: nachgebaute Webseite für Login zu Bank / online-Shop / ...
- ▲ Abwehrmaßnahmen:
 - geschützte Login-Sequenz (CTRL-ALT-DEL)
 - zertifizierte Web-Seiten
- ↳ spezielle Art von Trojanischem Pferd

5 Pufferüberläufe

E.1 Angriffe von innerhalb des Systems

- alle Programme sind von solchen Angriffen von innen bedroht
 - Eingabedaten
 - Interpretation von Aufrufparametern
- alle Programme, die Daten vom Netzwerk lesen sind solchen Angriffen auch von außen ausgesetzt
- ▲ Problem: Daten werden über das Ende von lokalen Feldern hinaus gelesen
 - Feld ist auf dem Stack allokiert
 - Stack wächst von hinten, Felder von vorne
 - Schreiben über Feldgrenze hinweg erreicht irgendwann Ende des Stackframes
 - Rücksprungadresse!!!
 - siehe Lehrveranstaltung Systemprogrammierung, 8. Übung
- ↳ Variante: Return-oriented programming

3 Logische Bomben

E.1 Angriffe von innerhalb des Systems

- Mitarbeiter schleust manipuliertes Programm ein
 - so lange Mitarbeiter regelmäßig anwesend ist passiert nichts (regelmäßiges Login, spezielles "Beruhigungs-Kommando", Benutzerliste abprüfen, ...)
 - verläßt der Mitarbeiter das Unternehmen, geht nach einiger Zeit "die Bombe hoch"
 - Daten löschen / manipulieren
 - Daten verschlüsseln (kidnappen)

4 Versteckte Hintertüren

- Spezieller Code wird in Systemsoftware eingefügt, um normale Kontrollmechanismen zu umgehen
 - Vorsorge für späteren Zugriff auf Rechner
- ▲ Abwehr: Review-Prozess vor der Installation von Software

E.2 Angriffe von außerhalb des Systems

E.2 Angriffe von außerhalb des Systems

- Angreifer muss zunächst in das System eindringen
- ▲ potentielle Helfer und Schwachstellen
 - Benutzer
 - Software an Netzchnittstellen
 - Mobiler Code (Web-Seiten, Mail)

1 Viren

- ★ Biologie: Mikroorganismus, der auf lebende Wirtszelle angewiesen ist und fähig ist, sich zu reproduzieren
- ★ Computervirus: Befehlsfolge, die ein Wirtsprogramm zur Ausführung benötigt und sich reproduzieren kann

1 Viren (2)

■ Aufbau und Funktionsweise eines Virus

Viruskennung	<code>virus { 4711</code>
Infektionsteil	<code>while (suche nicht infiziertes Programm) { kopiere Virus in Programm }</code>
Schadteil	<code>if (Datum == Freitag der 13.) { formatiere Festplatte }</code>
Sprung	<code>goto Programmstart }</code>

■ Mutation: Virus verändert sich beim Kopieren

1 Viren (3)

■ Speicherung von Viren

- ▶ Code-Segment ausführbarer Programme
- ▶ Betriebssystem
- ▶ Boot-Sektor des Hintergrundspeichers

■ Bedrohung durch Viren

- ▶ Integrität
- ▶ Vertraulichkeit
- ▶ Systemverfügbarkeit

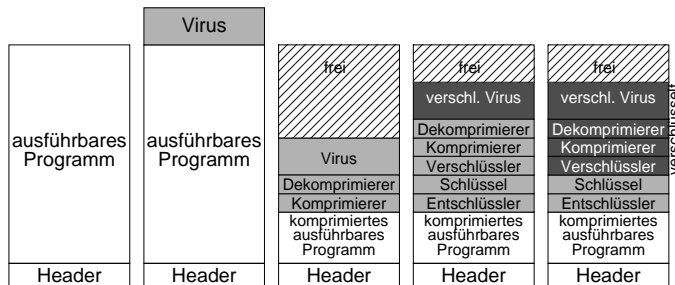
■ Abwehrmaßnahmen

- ▶ Integritätsprüfungen (Code-Inspektion)
- ▶ Signierte Software
- ▶ Quarantänestationen für neue Software
- ▶ Sorgfalt der Benutzer
- ▶ gutes Rechte-Management (Ausführung von SW mit minimalen Rechten)

2 Viren-Typen: Programm-Viren

■ Virus kopiert sich in eine ausführbare Datei

- ▶ verschiedene Alternativen



■ Strukturdaten der Datei (Modifikationszeitpunkt) werden wieder angepasst

3 Viren-Typen: Boot-Viren und speicherresidente Viren

■ Modifikation von Master-Boot-Record oder Boot-Sektor

■ Virus wird in der Boot-Phase resident in den Hauptspeicher geladen

- ▶ kopiert sich vor dem Start des Betriebssystems in den Speicher und bleibt dort resident (→ speicherresidente Viren)
- ▶ beeinflusst den Start des Betriebssystems (Freispeicherverwaltung)
- ▶ regelmäßige Aktivität z. B. durch Manipulation der Interrupt-Handler-Tabelle oder Abfangen von Systemaufrufen
- ▶ durch die Ausführung im Kernel-Modus beliebige Rechte
- ▶ durch Abfangen von Systemaufrufen und Kontrolle über Treiber auch Möglichkeiten zur Beeinflussung von Anti-Viren-Software

4 Viren-Typen: Makro- und Daten-Viren

- Kommandoprozeduren für Programme
 - z. B. Word oder Exel
 - werden in Kombination mit passiven Daten gehalten
 - passive Daten sind ohne die Makros oft nicht vernünftig nutzbar
- Sonderfall: Postscript-Daten
 - Postscript ist Turing-mächtige Programmiersprache
- ➔ Schadenspotential hängt von der Mächtigkeit der Makro-Sprache ab
 - besonders problematisch: Systemaufrufe durch Makros
- Schwachstellen in Bearbeitungs-Software für bestimmte Datenformate
 - z. B. gif-Daten
 - Probleme durch Pufferüberläufe und ähnliche Fehler
- ➔ Verbreitung häufig über E-Mail-Anhänge oder Web-Seiten

5 Viren: Antivirenmanagement

- Hauptprobleme
 - nachlässige Administratoren
 - nachlässige Benutzer
 - unzureichende Sicherheitskonzepte in Betriebssystemen (vor allem in früheren Microsoft-Systemen)
- Viren-Scanner
 - spezialisierte Unternehmen sammeln laufend Informationen
 - Viren werden zunächst isoliert (Infektion eines präparierten Programms - *goat file*)
 - Muster des Virus wird in Datenbank eingetragen und
 - Werkzeug durchsucht alle Programme nach Bytemustern oder Code-Sequenzen
 - regelmäßige Aktualisierung der Datenbank des Werkzeugs erforderlich

6 Viren: Antivirenmanagement (2)

- Heuristische Verfahren
 - Werkzeug durchsucht Programme nach typischen verdächtigen Codesequenzen
 - kann potentiell auch unbekannte Viren finden
 - Wettlauf der Viren-Erfinder mit den Werkzeug-Herstellern
- Aktivitätskontrolle
 - Werkzeuge oder Betriebssystemmechanismen überwachen Programme und erkennen atypisches bzw. Virus-typisches Verhalten (z. B. modifizierender Zugriff auf ausführbare Dateien)
- Monitoring
 - Werkzeuge überwachen wichtige Systemdateien und erkennen Modifikationen z. B. anhand von veränderten Hashes
 - wichtig: Informationen solcher Werkzeuge müssen vor dem Zugriff von Viren geschützt sein (externe Aufbewahrung, Verschlüsselung)

E.3 Würmer

- ★ Wurm: Ein ablauffähiges Programm mit der Fähigkeit zur Reproduktion
- Internet-Wurm (Robert T. Morris, 1988)
 - legte über Nacht ca. 6000 Sun- und VAX-Systeme mit Berkeley-UNIX lahm
 - Schwachstelle: bekannte Fehler in Berkeley-UNIX (sendmail, fingerd, rsh/ rexec), die über Internet unautorisierten Zugriff ermöglichten
 - Aufbau:
 - Ladeprogramm (l1.c, 99 Zeilen C-Code) und Wurm
 - l1.c wurde auf angegriffenem System kompiliert und gestartet, baute Verbindung zu angreifendem Rechner auf, lud Wurm und startete ihn
 - Wurm durchsuchte Routing-Tabellen nach weiteren Rechnern und versuchte diese zu infizieren (über rsh mit Benutzerabbildung, finger mit Pufferüberlauf, sendmail mit automatischer Ausführung eines Programms bei Mailempfang)

E.3 Würmer (2)

E.3 Würmer

- ... Internet-Wurm - Aufbau
 - Wurm versuchte außerdem Passwörter zu knacken und mit den Rechten der entspr. Benutzer auf anderer Rechner vorzudringen
- Fehler
 - Wurm überprüfte bei Zugang zu einem Rechner, ob dort der Wurm bereits aktiv war
 - nur in jedem 7. Fall wurde er trotzdem aktiv
 - Verhältnis 1:7 erzeugt zu viele Würmer
 - ↳ Rechner wurden mit den Würmern überschwemmt und brachen zusammen

E.3 Würmer (3)

E.3 Würmer

- Allgemeine Eigenschaften von Wurmern
 - Verbreitung vor allem über Internet
 - Angriffspunkte häufig Fehler in Betriebssystem-Netzwerkschnittstellen oder zentralen Systemprogrammen (mit Netzwerkschnittstelle oder im oft auch im Mail-Bereich)
 - Ausbreitung oft rasend schnell (bevor Patches verfügbar sind oder flächendeckend installiert sind)
- Bedrohungen
 - Integrität und Vertraulichkeit von Daten
 - vor allem Systemverfügbarkeit (teilweise auch als Nebeneffekt wenn Würmer außer Kontrolle geraten)
 - Schaden alleine durch Systemausfälle oft im Bereich 10 - 100 Mio. \$