

Cheatsheet

by Koll, Michael

Betriebssystemforensik (allgemein)

Betriebssystem

Architektur

Monolithisch (S.22)

Geschwindigkeit	schnell, minimaler Overhead; Funktionen optim. abgestimmt
Sicherheit	Risiko: ganzes BS im priv. Modus; Probleme einzelner Komp. Auswirkung auf ganzes BS
Speichereffizienz	Schlecht, ganzes BS im Speicher gehalten
Wartbarkeit, Erweiterbarkeit	Schlecht, da bei Änderungen viele Komponenten

Geschichtet (S.23)

Geschwindigkeit	Langsamer, da Funktionen Overhead, häufiger Kontextwechsel
Sicherheit	Teile des BS im User Mode, z.B. Treiber; Probleme Komponenten → BS
Speichereffizienz	Gut, einzelne Module dynamisch nachgeladen und entladen
Wartbarkeit, Erweiterbarkeit	Besser, da Änderungen meist nur bei einzelnen Komponenten

Mikrokern (S.24)

Geschwindigkeit	schlechte Performance, häufige Prozesswechsel und Interprozesskommunikation
Sicherheit	sicherheitskritischer Teil relativ klein; Dienste außerhalb Kern können Sicherheit und Stabilität nicht beeinflussen
Speichereffizienz	Gut, einzelne Module dynamisch nachgeladen und entladen
Wartbarkeit, Erweiterbarkeit	Sehr gut, einzelne Module können ausgetauscht werden (z.T. während Betrieb)

Vorteile virtuelles BS

Sandbox verbesserte Sicherheit durch Abschottung; bessere Ausnutzung des Systems durch mehrere VMs; herstellen kompatibler Laufzeitumgebungen

Ziele (S.12)

Unterstützung des Anwenders	Abstraktion der Hardware (Nummerierte Datenblöcke der HDD werden durch Reihenfolge, Verkettung und Verknüpfung zu Datei), Bereitstellen von Dienstfunktionen (Dateien öffnen, lesen, schreiben, schließen), Verbergen irrelevanter Details (Nummerierung Datenblöcke für Anwender nicht sichtbar)
Optimierung der Rechnerauslastung	Parallele Nutzung Rechnerkomponenten, mehrere Aufgaben quasiparallel
Zuverlässigkeit	Schutzmechanismus gegenseitig störender Prozesse, Abfangen von Ausnahmesituationen, Verhindern von blockierenden Prozessen
Portabilität	Programme auf verschiedenen Plattformen lauffähig
Nicht erfüllte Zuverlässigkeit	Prozess belegt zu viel Speicher, so dass andere Prozesse nicht ausgeführt werden können Abbruch mit Ctrl+C funktioniert nicht, da Signal auf Ignorieren steht Prozess zieht alle Prozessorleistung, so dass andere Prozesse blockiert sind (unfares Scheduling)

Aufgaben (S.14)

Programm- und Prozessverwaltung	Steuern, Erzeugen, Starten, Entfernen von Prozessen; Laden von Programmen von HDD in RAM; Leerlaufprozess; Kommunikation und Synchronisation von Prozessen
Anwenderschnittstelle	Kommandoebene, graphische Bedienoberfläche, Systemaufrufe zwischen BS und Programmen
Verwalten von Betriebsmitteln	Aufteilen der Betriebsmittel, Trennung Benutzerbereiche, Schutz, Prüfung Zugang
Verbindungen mit anderen Rechnern	

Begriffe

Parallel	Gleichzeitige Abarbeitung von Prozessen, jeder Prozess läuft auf eigener CPU
Quasiparallel	Abwechselnde Abarbeitung, alle Prozesse laufen auf gleicher CPU
Programm	besteht aus Vorschriften/Anweisungen in formaler Sprache; Ausführen zur Bewältigung bestimmter Aufgaben
Prozess	ablaufendes Programm mit konkreten Daten, besitzt Rechte, Registerinhalte und Speicher; Zustände running, ready oder waiting
Threads	Untereinheit von Prozessen, teilen sich denselben virtuellen Adressraum, Prozesswechsel schneller
Leerlaufprozess	Prozessor führt ständig Befehlszyklen aus, Leerlaufprozess verbraucht diese mit NOP-Anweisungen

Dateisystem

Zusammenhängende Belegung (S.104)

Belegungstabelle	Datei, Start, Länge
------------------	---------------------

Verteilte Belegung verkettete Listen (FAT) (S.105)

Belegungstabelle	Datei, Start
Hilfstabelle (FAT)	Verweis auf nächste Adresse, Dateieinde mit EOF

Verteilte Belegung mittels Index-Liste (S.106)

Belegungstabelle	Datei, Index-DU
Index-DU	Verweise auf DUs (falls zu lang Verweis auf weitere Index-DU)

Windows

Allgemein

Windows Stations, Desktops und Session (S.34)

Authentifizierung Session-orientiert, Session beinhaltet mehrere Stations, Stations beinhalten Desktops mit Fenstern und GDI-Objekten. Sicherheitsbeschreiber eines Objekts ist mit Station verbunden, darüber Kontrolle von Benutzer zum Desktop

Prozesse und Dienste

svchost.exe (Dienste) (S.138)

- mit `tasklist` laufende Prozesse mit Diensten auflisten (`tasklist -m svchost.exe -s`)
- mit `Process-Explorer` farblich gekennzeichnete Dienste → Properties → Services

- spezielle Programme wie z.B. svchost-Analyzer

Gestartete Dienste in Registry

HKLM\System\CurrentControlSet\Services als Unterschlüssel

laufende Prozesse PIDs und TIDs

mit Process Explorer; PID in Liste laufende Prozesse; TID Prozesseigenschaften → Threads

Registryzugriffe von Prozessen

Mit Process Explorer und Process Hacker; Möglichkeit über Process Monitor Registryzugriffe zu protokollieren (Software installieren → mit Process Monitor analysieren)

Ausgeführte Dienste

z.B. über msc (services) oder Registry (siehe oben)

Mandatorische Zugriffsregeln (S.153)

No-<Write Read>-Up	Kein schreibender/lesender Zugriff von Prozessen mit niedrigem Level auf Objekte mit höherem Level (gleiches Level zugelassen)
No-<Write Read>-Down	Kein schreibender/lesender Zugriff von Prozessen mit höherem Level auf Objekte mit niedrigerem Level (gleiches Level zugelassen)

Default: No-Write-Up (für alle Objekte), No-Read-Up (für Prozesse und Threads)

DACL (S.156)

Sicherheitsdeskriptor besteht aus Header, SID Besitzer, SID Gruppe, DACL, SACL

DACL besteht aus ACEs mit <Allow|Deny>, SID User, ACE-Bitmapp

Regeln DACL: Erst Einzel-ACE, dann Gruppe; Erst Verbote, dann Erlaubnisse; Reihenfolge von oben nach unten

Hinweis: Beim Ändern bzw. lesen aufpassen auf Gruppenzugehörigkeit (Jeder)

Festplatten und Drucker

Option 1	In regedit HKEY_LOCAL_MACHINE\SYSTEM exportieren, in RegRipper Report erstellen	ex-
Option 2	Systemwerkzeuge wie msinfo	

Forensische Anwendungsfälle

Suchen mit X-Ways

Nach Hexwert in Bild	Image einbinden, Datei nach hex-Wert durchsuchen
Nach ASCII-String in Dokument	Image einbinden, nach Text-Wert suchen mit ASCII-Codepage
Nach Unicode-String in Dokument	Image einbinden, nach Text-Wert suchen mit Unicode-Codepage
in docx-Datei	Image einbinden, Indexieren, Index nach Text-Wert durchsuchen mit ASCII- oder Unicode-Codepage

Carving

Carving-Programm durchsucht Dokument von Anfang nach Anfangssignatur, Markierung, Suchen Richtung Ende nach Endesignatur; Bereich dazwischen in Datei kopieren

Schattenkopie

Volume-Shadow-Copy-Service (VSS) hält Dateien in mehreren Versionen, Versionen können über Eigenschaften → Versionen eingesehen werden. Zur Analyse Schattenkopie mounten

Thumbs.db

Inhalte können mit Thumb.db-Viewer sichtbar gemacht werden (bildlich oder als Liste); Ungefähres Erscheinungsbild, Speicherort des Originals und Veränderungsdatum kann eingesehen werden

Überwachter Ordnerzugriff

(Details auf eigenem CheatSheet)

Angriffsmöglichkeiten prüfen, dazu:

Ist überwachter Ordnerzugriff aktiviert?	Windows Defender, Registry oder Gruppenrichtlinien
Standardverzeichnisse	Falls aktiviert, sind diese geschützt
Zusätzliche Verzeichnisse	Schauen ob Verzeichnis hinzugefügt (in Registry oder Windows Defender)
Erlaubte Anwendungen	Schauen ob Anwendungen erlaubt sind (in Registry)

Nutzung OneDrive

Anhaltspunkte zur Nutzung	
UserFolder	Schauen ob vorhanden
ClientFirstSignInTimestamp	Erster Login des Nutzers
UserCID	Falls vorhanden muss genutzt worden sein
Logdateien	Infos zu Anzahl Dateien, Up-/Downloadgeschwindigkeit, UserCID

UNIX

Systemzustand

Werkzeuge verwenden Informationen aus /proc-Verzeichnis	
Uptime	/proc/cpuinfo
Systemauslastung	/proc/stat
Speicherauslastung	/proc/meminfo
Version BS	/proc/version
Dateisysteme	/proc/filesystem

Windows 10-Forensik

Allgemein

Buildnummer

Aktuelle Buildnummer über `systeminfo` (cmd.exe) oder
HKLM\Software\Microsoft\Windows NT\CurrentVersion\
CurrentBuildNumber

Zuletzt verwendete Elemente

C:\Users\<username>\AppData\Roaming\Microsoft\Windows\
Recent

Überwacher Ordnerzugriff

Überwacht und blockiert den schreibenden Zugriff auf
vorhandene Dateien für nicht-vertrauenswürdige
Applikationen.

Aktivieren

Windows Defender Security Center → Einstellungen für Viren-
und Bedrohungsschutz → Überwacher Ordnerzugriff
oder

Gruppenrichtlinien: Computerkonfiguration/Administrative
Vorlagen/Windows/Windows Defender Antivir/Windows
Defender Exploit Guard/Überwacher Ordnerzugriff
oder

Registry (Besitzer vorher ändern): HKLM\Software\Microsoft\
Windows Defender\Windows Defender Exploit Guard\
ControlledFolderAccess\EnableControlledFolderAccess
(DWORD) = 0x01

Erlaubte Anwendungen

HKLM\Software\Microsoft\Windows Defender\
Windows Defender Exploit Guard\ControlledFolderAccess\
AllowedApplications
Hinzufügen mit (PS): Add-MpPreference
-ControlledFolderAccessAllowedApplications
«Anwendungspfad»

Geschützte Ordner

HKLM\Software\Microsoft\Windows Defender\
Windows Defender Exploit Guard\ControlledFolderAccess\
ProtectedFolders
Standardmäßig geschützte Ordner:
Documents\Pictures\Videos\Music\Desktop\Favorites
(<username> und Public)

Ereignisse

Einzusehen über EventVwr oder Powershell:
Get-WinEvent -LogName "Microsoft-Windows-Windows
Defender/Operational Where-Object {\$_.Id -in
1123,1124,5007}

Ereignis-IDs:

1123	Blockiertes Ereignis
1124	Überwachtes Ereignis (Auditmodus)
5007	Änderung von Einstellungen

Jumplists

Mehr Informationen als MRU/MFU:

- Dateiname, -pfad
- MAC Zeitstempel
- Name des Volumes
- Zeitlicher Verlauf von Down- und Uploads
- Informationen bleiben nach Löschen der Datei erhalten

Speicherort

Erstellt vom Betriebssystem: C:\User\<username>\AppData\
Roaming\Microsoft\Windows\Recent\AutomaticDestinations
Erstellt von Softwareanwendungen:
C:\User\<username>\AppData\Roaming\Microsoft\Windows\
Recent\CustomDestinations
Dateiname: <AppId>.<automatic|custom>Destinations-ms
Die AppId kann im ForensicsWiki nachgelesen werden https://www.forensicswiki.org/wiki/List_of_Jump_List_IDs

AutomaticDestination JL

Aufbau der Datei:

Header (32 Byte) mit Versionsnummer (3=Win10,
1=Win7/8), Anzahl Einträge, Anzahl gepinnte Einträge,
Zuletzt zugewiesene Entry-ID, Anzahl der Aktionen
DestList-Entry:

Prüfsumme	Fehlerhafter Eintrag wird nicht angezeigt
(New Birth)	Bei Änderung des Volumes geänderte New-
Volume-ID	ID
(New Birth)	Generiert aus Bootzeit, Sequenznummer und
Object-ID	MAC-Adresse. Bei Änderung des Volumes
	neue New-ID
NetBios Name	nbtstat -n
Entry ID	Fortlaufende Nummer
Access Timestamp	letzter Zugriff
Pinned Status	angepinnt (ja/nein)
Access Count	Zugriffszähler
variabel Unicode	vollständiger Pfad zur Datei
Länge Unicode	Länge Unicodepfad

CustomDestinations JL

einfachere Dateistruktur, zusammengesetzte
MS-SHLINK-Segmente
Anfang eines LNK-Segments: 4C 00 00 00 01 14 02 00 00
00 00 00 C0 00 00 00 00 00 00 46
Ende: AB FB BF BA

QuickAccess/Schnellzugriff

Angepinnte Einträge im Schnellzugriff des Explorers.
Dateiname 5f7b5f1e01b83767.automaticDestinations-ms

Tools

JumpListExt for	grafische Oberfläche, nicht mehr stabil in ak-
Windows 10	tuellen Versionen
JLECmd	JLECmd.exe -f <JLFile>
	(-html -csv -json) <targetDir> (-ld)

Windows 10 Applications

SystemApps

vorinstalliert, können nicht deinstalliert werden
C:\Windows\SystemApps\<appname>

WindowsApps

über Windows Store C:\Windows\WindowsApps\<appname>

Einstellungsdaten

C:\Users\<username>\AppData\Local\Packages\<appname>
Haupteinstellungen in Datei/Registry-Hive `settings.dat`

Anwendungsdaten

Gespeichert in ESE-DB-Datenbanken, Aufbau nicht
vollständig bekannt, teilweise möglich mit `ESEDatabaseView`
von Nirsoft

Build-in applications

Im Folgenden sind auf Windows bereits vorinstallierte
Programme aufgelistet, die forensisch verwertbare Information
bringen können, mit dem Namen, unter dem sie im
Konsolen-/Powershell-/„Ausführen“-/„Neuen Task
ausführen“-Fenster gestartet werden können:

certmgr Tool zum Verwalten der für den jeweiligen Benutzer verfügbaren Zertifikate.

control Systemsteuerung.

cipher Tool zum sicheren löschen von Daten, sodass sie nicht wieder herstellbar sind. Kann auch dafür verwendet werden, freien Speicherplatz auf der Festplatte zu löschen. Kann auch dafür verwendet werden, Dateien zu verschlüsseln.

diskmgmt Tool mit grafischer Oberfläche zum Verwalten von Datenträgern: Partitionen, Laufwerksbuchstaben und die Partitionstabelleart (MBR/GPT) von Datenträgern kann hiermit verändert werden

diskpart Kommandozeilentool, das ähnliche Funktionalität bietet wie diskmgmt.

eventvwr Tool zum Anzeigen diverser systemweiter Ereignisse. Entwickler von Dritt-Programmen können ihre Programme ebenfalls Ereignisse in die Ereignisanzeige schreiben lassen.

fsutil Stellt Funktionalitäten für Dateisystem-Operationen bereit.

gpedit Editor zum Bearbeiten von Richtlinien für einzelne Benutzer oder den ganzen Computer. Hier können Sicherheitseinstellungen vorgenommen werden aber auch Skripte hinterlegt werden, die beim Anmelden/Abmelden eines Nutzers oder auch beim Starten/Herunterfahren des Computers ausgeführt werden.

msconfig Bietet Konfigurationsmöglichkeiten für den Start des Systems und bietet darüber hinaus eine Anzeige zur Information, welche Dienste gerade ausgeführt werden und welche davon beim Systemstart gestartet werden.

msinfo32 Liefert ausführliche Informationen zu Treibern, angeschlossene Hardware, Druckaufträge, Systemvariablen, geladene Module, Dienste, etc.

perfmon Systemleistungs-Monitoring-Tool. Kann dazu benutzt werden, Statistiken über einzelne Prozesse und Eigenschaften einzelner Prozesse aufzuzeichnen.

regedit Editor für die Registry.

resmon Tool zum Monitoring von CPU, RAM, Prozessen, Netzwerkschnittstellen und Datenträgern.

secpol Editor zum Einstellen diverser Richtlinien. Es kann z. B. eingestellt werden, welche Ereignisse überwacht oder sogar unterbunden werden sollen.

taskschd Tool zum Anlegen von Aufgaben, die regelmäßig bzw. unter bestimmten Bedingungen ausgeführt werden.

WF Bietet Firewall-Konfigurationsmöglichkeiten

Witere tiefer im System verankerte Konsolenbefehle:

computerdefaults Festlegen von Standardprogrammen.

control Windows Features aktivieren oder deaktivieren.

appwiz.cpl „2

inetcpl.cpl Öffnet die Internetoptionen.

main.cpl Öffnet Mauseinstellungen.

Ncpa.cpl Öffnet das Netzwerkverbindungsmenü.

powercfg.cpl Öffnet die Energiesparoptionen.

sndvol Öffnet das Sound-Menü.

sysdm.cpl Systemeigenschaften öffnen (Umgebungsvariablen, Leistungsoptionen, Computername, etc.)

Scripts

Sicherstellen, dass eine Batch-Datei als Administrator gestartet wird:

```
if not "%1"=="am_admin" (powershell start -verb
```

Öffnen einer Konsole als Systemnutzer (muss als Administrator ausgeführt werden):

```
PsExec.exe -i -s -d CMD
```

Erlaube Ausführung von Powershell-Skripten:

```
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe Set-ExecutionPolicy -Scope "LocalMachine" -
```

Erlaube RDP-Verbindungen:

```
REG.exe ADD "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /f /v fDenyTSConnections /t REG_DWORD /d 0
```

Schalte das Speichern von Thumbnails aus:

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced]
"NoThumbnailCache"=dword:00000001
"DisableThumbnailCache"=dword:00000001

[HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\Explorer]
"DisableThumbsDBOnNetworkFolders"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced]
"NoThumbnailCache"=dword:00000001
"DisableThumbnailCache"=dword:00000001

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced]
"DisableThumbnailCache"=dword:00000001
"NoThumbnailCache"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced]
"DisableThumbnailCache"=dword:00000001
"NoThumbnailCache"=dword:00000001
```

Fast Startup und Ruhezustand

Datei: hiberfil.sys

Zustände

HIBR	Im Ruhezustand
RSTR	Wird fortgesetzt
WAKE	Nach Fortsetzung

Forensische Bewertung

- Änderung des Formats ab Win8
- Header bleibt auch nach Fortsetzen verfügbar
 - Daten nur zwischen Versetzen in Ruhezustand bis zur Fortsetzung
 - Vor Win8 zeitlich weit zurückreichende Daten
 - Sichern der hiberfil.sys im laufenden Zustand keine forensisch relevanten Daten
 - Größte Menge Daten **shutdown /h** runas '%0' am_admin & exit).
 - HIBR2BIN ermöglicht dekomprimieren der Daten im neuen Format
 - Fast Startup liefert keine interessanten Daten, da alle Applikationen beendet sind

Edge Browser / ESE-DB

Anwendungspfad
C:\Windows\SystemApps\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\MicrosoftEdge

ESE-Datenbank Transaktionsflow

1. Transaction in RAM (Log Cache)
2. Seiten aus DB in RAM (Page Cache)
3. Transaktion im RAM anwenden (LC→PC) (LC→Datei)
4. Aktualisieren der Datenbank
5. Datenbank aktualisieren

Dirty-DB
Datenbank, die nicht vollständig aktualisiert wurde.
V01.chk Zeitpunkt der Transaktion
CurrentVersion\Transaktionsdaten\Dateinamen
Wiederherstellung mit esentutl
esentutl /mh database.dat Überprüfung der Datenbank (Feld State=Dirty)
esentutl /fr database.dat Wiederherstellung der Datenbank (Feld State=Clean)

WebCacheV01.dat
Pfad
→C:\Users\<username>\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\AC\MicrosoftEdge\ (enthält v.a. Verweise und Speicherorte)
→C:\Users\<username>\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\AC\#!<number>\MicrosoftEdge\

Aufbau

Tabelle Containers

ContainerId	Referenz auf Tabelle Container_n
Directory	Pfad zum Verzeichnis mit zwischengespeicherten Daten
SecureDirectories	Zufällige Zeichenfolge, in 8er-Gruppen teilbar
Name	Containertyp (Cookies Content History ...)
PartitionId	Integritätslevel, (Protected= Internet=Low lokal=medium)

Tabelle Container_n

SecureDirectory	Unterverzeichnis im Cachepfad
Type	z.B. In PrivateModus (siehe Chivers)
AccessCount	Anzahl wie oft URL referenziert wird
<Timestamps>	Sync, Creation, Expiry, Modified, Accessed Time
URL	Quelle der Informationen
Filename	Name der Cachedatei

Cache-Speicherort ermitteln

SecureDirectories	in 8er-Blöcke aufteilen
SecureDirectory	zeigt auf x-ten Block (in Container_n)
Directory	Zeichenfolge anhängen

Zeitstempel

CreationTime	Erstellungszeit der Cachedatei/-objekt
ExpiryTime	vom Webserver vorgegeben, Cache wird ungültig
ModifiedTime	vom Webserver, Zeitpunkt der letzten Änderung der Ressource
AccessTime	Letzter Zugriff des Nutzers auf Datei

Werkzeuge

Fazit: Tools gute Unterstützung, manuell bringt mehr	
IECacheView	Zeigt Cachedateien von IE und Edge (Dateiname, -größe, -typ, URL, Zeitstempel, Cachedateipfad)
BrowsingHistoryView	Zeigt Browserverlauf mehrerer Browser

OneDrive

Anwendungspfad

C:\User\<username>\AppData\Local\Microsoft\OneDrive\

Registry

HKU\Software\Microsoft\OneDrive\	
.\	Version, UserFolder
.\Accounts\Personal	ClientFirstSignInTimestamp, UserID, UserFolder

Konfigurations- und Diagnostikdaten

Ausgehend vom One-Drive-Verzeichnis:	
.\logs\Personal\	Down-\Uploadgeschwindigkeit,
SyncDiagnostics.log	Ausstehende Down-\Uploads, verfügbarer Speicherplatz lokal, UserID (siehe REG), Anzahl Dateien und Verzeichnisse
.\settings\Personal\<userid>.dat	bisher kein Parser, mit Hexeditor Dateinamen einsehen
.\settings\Personal\<uploads downloads>.txt	Während Download temporär Daten wie Dateiname und User-CID

Logdateien

.\logs\Personal\	
*.aodl, *.odlsent, *.odl	enthalten Clientaktivitäten
Die Datei ObfuscationStringMap.txt	enthält verschleierte Dateinamen, die in den Logs gefunden werden können.
Mögliche Aktionen in den Logs:	
FILE_ACTION_ADDED	Datei lokal hinzugefügt
FILE_ACTION_REMOVED	Datei lokal entfernt
FILE_ACTION_RENAMED	Datei umbenannt

Arbeitsspeicher

Username und Passwort liegen im Klartext vor, nach Parameter &passwd= und &loginmft= suchen

Benachrichtigungen und Kacheln

Datenbank

C:\Users\<username>\AppData\Local\Microsoft\Windows\Notifications	
wpndatabase.db	Datenbank (Signatur 53 51 4C 69 74 65 20 66 6F 72 6D 61 74 20 33)
wpndatabase.db-wal	Write Ahead Log (Signatur 37 7F 06 82 oder 37 7F 06 83)
wpndatabase.db-shm	Shared Memory File, keine spezifische Signatur

SQLite-Datenbank mit WAL-Verfahren: Änderungen in Datei, bei Erreichen des Checkpoints (manuell oder automatisch) synchronisiert. WAL-Dateien bei der Untersuchung einbeziehen (PRAGMA wal_checkpoint).

Struktur und Inhalt

Relevante Tabellen in wpndatabase.db	
NotificationHandler	Anwendungen, die zu Benachrichtigungen berechtigt sind (Zuordnung über PrimaryID → AppID, GUID)
Notification	Benachrichtigungsinhalt → Payload

Kacheln

Datenbank wie Benachrichtigungen, Zeitstempel ArrivalTime und ExpiryTime Rückschlüsse auf Verwendung des Computers
Einige Anwendungen legen in dem DB-Verzeichnis Cacheordner an, die sehr lange zurückreichen

Cortana

%localAppData%\Packages\Microsoft\Microsoft.Windows.Cortana_cw5n1h2txyewy

Artefakte

→.\AppData\Indexed DB\IndexedDB.edb	11 Tabellen, Tabelle HeaderTable enthält createTime, lastOpenTime
→.\LocalState\ESEDatabase\CortanaCoreInstance\CortanaCoreDb.dat	[Veraltet] Geofences mit Standortdaten, Reminders benutzerspezifische Erinnerungen, Triggers LocationTriggers, TimeTriggers, ContactTriggers
→.\LocalState\DeviceSearchCache\	keine Dokumentation, Infos über Programmeinträgen, -aufrufen, Zeitstempel und JL-Einträge
→.\AC\INetCache\<randomnumber>	vollständige HTML-Seite von Suchen über Cortana
→.\AC\AppCache\<randomnumber>	HTML- und JavaScript Dateien für Cortana-Suche
→.\LocalState\LocalRecorder\Speech	Aufgezeichnete Sprachbefehle
→.\LocalState\Cortana\Uploads\Contacts	Falls Synchronisierung mit Android, Kontaktdaten und Mobilnummern
→9d1f905ce5044aee.	URLs die über Cortane-Suche ausgelöst wurden
→WebCacheV01.dat	URLs die über Cortana aufgerufen wurden
→%SystemDrive%\Windows\Prefetch\SEARCHUI.EXE-14F7ADB7.pf	Letzte Ausführungszeit(en)
→%SystemDrive%\Windows\appcompat\Programs\Amcache.hve	Erstellungs- und Änderungszeitstempel der Anwendung

Deaktivieren von Cortana

Parameter in HKLM\Software\Policies\Microsoft\Windows\Windows Search	
AllowCortana	dword:00000000
DisableWebSearch	dword:00000001
AllowSearchToUseLocation	dword:00000000
ConnectedSearchUseWeb	dword:00000000
ConnectedSearchPrivacy	dword:00000003

Registryforensik

Relative Pfade

%UserProfile%	Pfad zum derzeitigen Benutzerprofil
%SystemDrive%	Laufwerksbuchstabe, auf dem Windows installiert ist, i.d.R C:
%SystemRoot%	Pfad zum Windows Ordner, i.d.R. C:\Windows

Schlüssel & Werte

Ein Schlüssel enthält einen oder mehrere Werte sowie einen Zeitstempel des letzten Zugriffs

Jeder Wert hat 3 Felder:

Name	Eindeutig innerhalb eines Schlüssels
Typ	Datentyp des Wertes (s.u.)
Daten	kann leer oder null sein, Maximum 32767 Bytes, häufig in hexadezimaler Notation

Die wichtigsten Datentypen sind

REG_NONE	kein definierter Typ
REG_SZ	Fixe Länge und NULL-Char am Ende
REG_EXPAND_SZ	Variable Länge und NULL-Char am Ende
REG_BINARY	Binärdaten
REG_DWORD	Double-Word-Werte, häufig boolesche Werte
REG_LINK	Link
REG_MULTI_SZ	Liste von Strings

Struktur

Wurzelschlüssel

HKLM	HKEY_LOCAL_MACHINE	Hauptschlüssel
HKU	HKEY_HKU	Hauptschlüssel
HKCR	HKEY_CLASSES_ROOT	Verweis
HKCU	HKEY_CURRENT_USER	Verweis
HKCC	HKEY_CURRENT_CONFIG	Verweis

Verweise

HKCC	HKLM\SYSTEM\CurrentControlSet\Hardware Profiles\Current
HKCU	HKU\S-1-5-21-xxx (SID)
HKCR	HKLM\SOFTWARE\Classes

HKU

Nutzerspezifische Einstellungen und Informationen für jeden aktiv geladenen Benutzer (Standardprofile und angemeldete Profile, keine abgemeldeten Nutzer)

.DEFAULT	Einstellungen, die Windows nutzt, bevor ein Nutzer sich eingeloggt hat
S-1-5-18	well-known SID für LocalSystem-Benutzer
S-1-5-19	well-known SID für LocalService-Benutzer, lokale Dienste, die den LocalSystem-User nicht benötigen
S-1-5-20	well-known SID für NetworkService-Benutzer, Netzwerkdienste, die den LocalService-Benutzer nicht benötigen
S-1-5-21-[...]	SID des derzeit angemeldeten Benutzers (Link von HKCU)
S-1-5-21-[...]_Classes	Nutzerspezifische Dateiverknüpfungen

HKCU

Link auf HKU\[SID]

Spezifische Einstellungen und Informationen zum angemeldeten Benutzer (Umgebungsvariablen, Desktopeinstellungen, Netzwerkverbindungen, Drucker und Präferenzen)

AppEvents	Verknüpft Audiodateien mit Aktionen (z.B. Ton beim Öffnen eines Menüs)
Console	Daten zum Console-Subsystem (z.B. zum MS-DOS-Command-Prompt)
Control-Panel	Einstellungen der Systemsteuerung, u.a. regionale Einstellungen und Erscheinungsbild
Environment	Umgebungsvariablen, die Benutzer gesetzt haben
Keyboard-Layout	Installierte Tastaturlayouts
Network	Jeder Unterschlüssel ein Netzlaufwerk, Name des Schlüssels ist Laufwerksbuchstabe, enthält Konfigurationsdaten zum Verbinden
Printers	Präferenzen des Benutzers zum Drucken
Software	Nutzerspezifische Einstellungen zu installierten Programmen, je nach Programm Informationen zu Programmanbieter, Programm, Version, Installationsdatum und zuletzt zugegriffene Dateien. Ablage nach HKCU\Software\Programmanbieter\Programm\Version
Volatile Environment	Umgebungsvariablen, die beim Login definiert wurden

HKLM

Spezifische Einstellugen des lokalen Rechners, die für alle Benutzer geladen werden.

HARDWARE	Speichert HW-Daten beim Systemstart, wird bei jedem Start erstellt und mit Informationen über Geräte, Treiber und Ressourcen gefüllt
SAM	Lokale Windows-Sicherheitsdatenbank über Benutzer- und Gruppeninformationen (Link zu HKLM\SECURITY\SAM)
SECURITY	Lokale Windows-Sicherheitsdatenbank (inklusive SAM)
SOFTWARE	Einstellungen zu Applikationen des Rechners (und Microsoft-Applikationen)
SYSTEM	Informationen zur Systemkonfiguration (z.B. Gerätetreiber und Dienste). Derzeitiges Hardwareprofil ist Link von HKCC. Mehrere Sätze mit Schema ControlSetxxx. HKLM\SYSTEM\Select zeigt aktuelle verwendetes Profil in CurrentControlSet.

HKCR

Link auf HKLM\Software\Classes & HKU\[SID]_Classes

- Zuweisungen für Dateierweiterungen
- OLE-Datenbank
- Einstellungen für registrierte Anwendungen für COM-Objekte
- Nutzer- und systembasierte Informationen

Setzt sich aus HKLM\SOFTWARE\Classes und HKU\[SID]_Classes zusammen. Falls identischer Wert, hat HKCU Priorität.

Beispiel: Was soll passieren, wenn eine .pptx-Datei geöffnet wird. HKCR macht einen erheblichen Teil der Registry und des Systemverhaltens aus

HKCC

Link auf HKLM\System\CurrentControlSet\Hardware Profiles\Current

Link zu den Konfigurationsdaten des derzeitigen Hardwareprofils. Informationen werden bei jedem Booten neu erzeugt und daher nicht physisch in der Registry-Datei gespeichert.

System
Software

Hives

User-Profile-Hives in %UserProfile%\NTUSER.DAT

Alle anderen Hives und Dateien in %SystemRoot%\System32\config

HKU\.DEFAULT	DEFAULT
HKLM\SAM	SAM
HKLM\SECURITY	SECURITY
HKLM\SOFTWARE	SOFTWARE
HKLM\SYSTEM	SYSTEM

Schlüssel HKLM\HARDWARE mit dynamischen Hive, wird beim Systemstart erstellt aber nicht gespeichert

Liste zu Standard-Hive-Files:

HKLM\SYSTEM\CurrentControlSet\Control\hivelist
Liste User-Hives: HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList

SID & SAM

Liste der SIDs

HKLM\Software\Microsoft\WindowsNT\CurrentVersion\ProfileList
Pfad zu individuellen Profilen: ProfileImagePath

Aufbau der SID (S-1-5-21-[-...]-1002):

S Identifiziert den Schlüssel als SID
1 Revisionsnummer, Nummer der SID-Spezifikation
5 Autorität
21-[-...] Domänen-ID, identifiziert die Domäne oder den lokalen Computer, Wert ist variabel
1002 Benutzer-ID, relative ID (RID), >1000 für Profile die nicht standardmäßig generiert wurden

Informationen aus SAM

SAM\Domains\Account\Users\<Benutzernummer>\

F Enthält Informationen wie Datum der letzten Passwortänderung und Datum der letzten Anmeldung vom Nutzer mit der Id <Benutzernummer>

Wichtige Pfade

Systeminfo

HKLM\Software\Microsoft\ Windows Buildnummer
Windows NT\CurrentVersion/ (cmd: systeminfo)
CurrentBuildNumber

Autorun

HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce
HKLM\Software\Microsoft\Windows\CurrentVersion\Run
HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce
HKCU\Software\Microsoft\Windows\CurrentVersion\Run
Pfade in Run bei jedem Systemstart, RunOnce nur einmal

MRU

HKU\<SID>\Software \Microsoft \Windows \ CurrentVersion \Explorer
ComDlg32 Zuletzt ausgeführte Anwendungen und deren Pfade sowie geöffnete oder geänderte Dateien
RecentDocs Unterschlüssel mit Dateierweiterungen, zuletzt geöffnete Dateien diesen Typs
RunMRU Aufrufe, die via Run durchgeführt wurden
UserAssist Werte von Objekten, auf der Nutzer zugegriffen hat (z.B. Optionen der Systemsteuerung, Dateiverknüpfungen und Programme)

ROT13 verschlüsselt, es gibt mehrere MRU-Listen in unterschiedlichen Listen

Geschützter Speicher

HKU\<SID>\Software \Microsoft \ Protected Storage System Provider
Verschlüsselte Passwörter für viele Anwendungen (Outlook Express, MSN-Explorer oder Internet Explorer)

Autovervollständigung oder Passwort merken

Internet Explorer

HKU\<SID>\Software \Microsoft \Internet Explorer
Download Informationen zu Downloads
Main Benutzereinstellungen (Search Bars, Startseite, etc.)
TypedURLs Zuletzt besuchte Seiten (z.B. EMail, Onlinebanking)

Microsoft Edge nutzt
HKCU\Software\Classes\Local Settings\Software\ Microsoft\Windows\CurrentVersion\AppDataContainer\Storage\ microsoft.microsoftedge_xxxxxx\MicrosoftEdge

Netzwerke

WLAN

HKLM\Software\Microsoft\Windows NT\ Netzwerkgeräte
CurrentVersions/NetworkCards (Beschreibung und GUID)
HKLM/System/CurrentControlSet/ Details zum Netz-
Services/Tcpip/Parameters/ erkergerät (IP, Gate-
Interfaces/<GUID> way, Domain)

P2P

HKLM/System/ControlSet001/ Applikationen mit
Services/SharedAccess/Parameters/ erlaubtem Zugriff
FirewallPolicy/StandardProfile/ auf ausgehende
AuthorizedApplications/List Verbindungen

Angeschlossene Geräte

HKLM/System/Mounted Devices Liste aller Geräte, die im System gemountet wurden
HKCU\Software\Microsoft/ Mount eines Geräts bei
Windows\CurrentVersion\Explorer/ Nutzerlogin
MountPoints2
HKLM/System/CurrentControlSet/ Enthält für jede
Control/DeviceClasses DeviceClass-GUID

Unterschlüssel mit Geräten die verbunden waren oder sind.
DeviceInstance ist Pfad zu HKLM/System/CurrentControlSet/Enum. Durch Export Zeitstempel für ersten und letzten Zugriff

HKLM/System/CurrentControlSet/Enum/Geräte im System mit
<Enumerator>/<DeviceID> Gerätebeschreibung und IDs

HKLM/System/CurrentControlSet/Enum/Angeschlossene USB-
USBSTOR Geräte

Antiforensische Maßnahmen

Zeitstempel fälschen Prüfsumme häufig nur auf Inhalt (Tool <http://www.petges.lu/home/download>)
Pagefile.sys In HKLM/System/CurrentControlSet/Control/Session Manager/Memory Management den Wert ClearPagefileAtShutdown auf 1 setzen
Zeitstempel vermeiden HKLM/System/CurrentControlSet/Control/FileSystem Wert NtfsDisableLastAccessUpdate auf 1 setzen
Einträge löschen Verlauf IE oder zuletzt genutzte Dokumente
UserAssist abstellen HKU\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist Wert NoLog vom Typ DWORD mit Wert 1 erstellen

Tools

FTK-Imager Erstellung von Abbildern, Kopien der Hive-Files (Live) (Files → Obtain Protected Files)
Registry-Editor Importieren und Exportieren von Dateien, Struktur laden und entfernen, Verbinden mit der Registry eines Remotecomputers, Berechtigungen ändern, Registry durchsuchen
RegShot Änderungen in der Registry aufzeichnen (Erstellen eines ersten Abbildes und Vergleich mit einem zweiten)
Forensic Registry Editor (fred) Untersuchung und Bearbeitung von HIVE-Dateien, vorgefertige Berichtsvorlagen
RegRipper Extrahieren von spezifischen Informationen, Automatisierung durch Plugins und Profile
DCODE Decodieren von Zeitstempeln (<https://www.dcode.fr/timestamp-converter>)
Access Data Auslesen von Hive-Files (<https://accessdata.com/product-download/registry-viewer-1-8-0-5>)
Registry Viewer Auslesen von Hive-Files (<https://www.gaijin.at/dlregview.php>)

Netzwerkforensik

MAC-Adresse

Eine MAC-Adresse ist eine physikalische Adresse, die zur Adressierung von Netzwerkverkehr benutzt wird. Auch MAC-Adressen können gefälscht werden. Bei virtuellen Netzwerkkarten (wie sie z. B. in virtuellen Maschinen zum Einsatz kommen), sind MAC-Adressen frei wählbar. Eine MAC-Adresse ist 6 Byte lang.

Sniffing

Sniffing bezeichnet das Mitschneiden bzw. Analysieren von Netzwerkdatenverkehr. Dies kann im Wesentlichen entweder durch einen man-in-the-middle-Angriff erfolgen oder durch das allgemeine Mitlesen von Netzwerk-Datenverkehr (i. d. R. Ethernet oder WLAN), zu dem man physischen Zugang hat.

Tools

cURL	Einfaches Programm zum Senden von Netzwerk-Requests. Unterstützte Protokolle sind unter anderem HTTP, HTTPS, FTP und FTPS.
dig	Befehl zum Abfragen des Domain Name Systems (Alternative zu nslookup).
dsniff	Tools zum Sniffen von Passwörtern und Analysieren von Netzwerkdatenverkehr allgemein.
Ettercap	Tool zum Durchführen von Man-in-the-middle-Angriffen, beispielsweise mittels ARP-Spoofing.
filesnarf	Dateisniffer für NFS-Datenverkehr. (In dsniff enthalten.)
mailsnarf	Sniffer für Mails im Berkeley mbox format. (In dsniff enthalten.)
msgsnarf	Sniffer für ältere bekannte Chat-Messenger (ICQ, IRC, MSN Messenger usw.)
nmap	Etablierter Konsolen-basierter Portscanner.
OpenVAS	Etablierter Schwachstellen-Scanner.
Scapy	Tool zum Manipulieren von Paketen im Netzwerkverkehr.
urlsnarf	Sniffer für HTTP-Requests. (In dsniff enthalten.)
pcap	API für Sniffer, die von Tools wie Tcpdump, nmap usw. verwendet wird.
Tcpdump	Bekannter und verbreiteter Paketsniffer (Kommandozeilentool).
Wireshark	Etablierter Netzwerksniffer für Pakete verschiedener Protokolle

ARP

Das „Address Resolution Protokoll“ wird bei IPv4 benutzt, um von einer IP-Adresse die MAC-Adresse zu ermitteln, unter der sie zu erreichen ist. Das entsprechende Äquivalent von ARP für IPv6 ist das „Neighbor Discovery Protocol“ (NDP). Mittels „ARP -a“ kann man beispielsweise ARP-Zuordnungen unter Windows auslesen.

ARP-Spoofing

Als ARP-Spoofing bezeichnet man das Verteilen von ARP-Paketen bei denen die Kombination aus MAC-Adresse und IP-Adresse falsch ist. Empfänger solcher ARP-Pakete mit falschen Informationen übernehmen diese Informationen in aller Regel, ohne Prüfungen anzustellen.

Man-in-the-middle-Angriffe

Bei dieser Art von Angriffen schaltet sich der Angreifer netzwerktopologisch gesehen zwischen einem Server und sein Ziel. Dies kann oft relativ einfach mit ARP-Spoofing erreicht werden. Der man-in-the-middle kann den Netzwerkverkehr vom Ziel nun mitlesen.¹ Sofern der man-in-the-middle den Datenverkehr unverändert weiterleitet, merkt das Ziel in der Regel nichts von dem man-in-the-middle. Der Angreifer kann Datenverkehr auch unterdrücken oder verändert weiterleiten (z. B. für Phishing-Angriffe).

¹Dies bringt dem Angreifer nur für Netzwerkverkehr einen Vorteil, der unverschlüsselt vom Ziel gesendet/empfangen wird

Datenträgerforensik

Dateisysteme

	NTFS	exFAT	FAT32
Max. Größe	16EB	128PB	2TB
Max. Dateigröße	16TB	16EB	4GB
Max. Länge von Dateinamen	255	255	255
Anwendung	Windows, externe Datenträger	diverses	USB-Sticks

Tools

AccessData FTK Imager	Tool zum erstellen von Datenträger-Images.
Active@ Disk Editor	Tool zum direkten Anzeigen/Bearbeiten von Daten auf der Festplatte im Hex-Format.
dd	Tool zum Erstellen von Datenträgerimages.
Alternate-StreamView	GUI-basiertes Tool zum schnellen und einfachen Anzeigen von Alternate Data Streams.
exiftool	Umfangreiches Konsolen-basiertes Tool zum Anzeigen von EXIF-Daten von Bilddateien.
DiskDigger	Programm zum Wiederherstellen von gelöschten Dateien.
fdisk	Kommandozeilen-Programm zur Partitionierung von Datenträgern.
fsstat	Tool zum Anzeigen von Informationen über ein Dateisystem.
HxD	Einfacher Hex-Editor.
icat	Tool zum Anzeigen einer Datei basierend auf der inode-Nummber.
losetup	Konsolenbasiertes Tool für Linux zum Mounten von Partitionsimages.
mmls	Tool zum Auslesen der Partitionstabelle.
ntfswalker	Tool zum analysieren von NTFS-Partitionen.
OSFMount	GUI-basiertes Windows-Tool zum Mounten von Partitionsimages unter Windows.
Testdisk	Programm zum Wiederherstellen von gelöschten Dateien und Partitionen.
xxd	Konsolen-basiertes Tool für Linux zum Anzeigen des Hex-Dumps einer Datei.

Anderes

LUKS
Abkürzung für „Linux Unified Key Setup“. LUKS ist eine Erweiterung von dm-crypt und fügt den verschlüsselten Daten einen Header hinzu. Einen LUKS-Container erkennt man am Header. Dieser beginnt mit den Bytes „4C 55 4B 53 BA BE“. Ein LUKS-Container kann beispielsweise mit losetup eingebunden (gemountet) werden. Ein typischer Aufruf kann so aussehen:
sudo losetup -o 11071426702 /dev / loop3 myImage.img

Assembler

Allgemeines

²

Als Assembler bezeichnet man Computerprogramme, die Assemblerbefehle in Maschinencode übersetzt. Im Gegensatz zu Compilern von Hochsprachen übersetzen Assembler strikt die eingegebenen Befehle und interpretieren den den Eingangsquellcode kaum.

Register

Verwendung der Register

General purpose Register:

- eax: Zwischenwerte/Rückgabewerte bei Berechnungen
- ebx: Adressierungen (Base)
- ecx: Zählerregister (Counter)
- edx: I/O-Daten (Data)
- esi: Quelloperand-Speicheradresse für Stringoperationen (Source)
- edi: Zieloperand-Speicheradresse für Stringoperationen (Destination)

Special purpose Register:

- esp: Enthält die Adresse des obersten Stackelements (Stackpointer)

- ebp: Enthält die Adresse des aktuellen Stack-Frames
- eip: Enthält die aktuell auszuführende Instruktion (Instructionpointer)
- eflags: Enthält diverse Flags (Zeroflag, Overflow-Flag usw.)

Segment-Register:

- cs: Codesegment
- ds: Datasegment
- es: Extrasegment
- ss: Stacksegment

Verwendung der Flags

Die folgende Auflistung enthält die Flags, die im Flag-Register gespeichert sind.

- CF (Carry-Flag): Enthält den Übertrag aus einer vorangegangenen Operation
- PF (Parity-Flag): TODO
- AF (Adjust-Flag): TODO
- ZF (Zero-Flag): Ist 1, wenn das Ergebnis der letzten Operation 0 war.
- SF (Sign-Flag): TODO
- TF (Trap-Flag): TODO
- IF (Interrupt-Enabled-Flag): TODO

- DF (Direction-Flag): TODO
- OF (Overflow-Flag): Gibt an ob bei der letzten Operation ein Überlauf (oder „Unterlauf“) aufgetreten ist. Gewöhnlich definiert als $OF = in-carry^3 \text{ xor } out-carry^4$
- IOPL (IO-Privilege-Level): TODO
- NT (Nested-Task): TODO
- RF (Resume-Flag): TODO
- VM (Virtual-8086-Mode): TODO
- AC (Alignment-Check): TODO
- VIF (Virtual-Interrupt-Flag): TODO
- VIP (Virtual-Interrupt-Pending): TODO
- ID (Able to use CPUID instruction): TODO

Adressierungsarten

Befehle

Common Intermediate Language

mov

sub

call

²Dieses Cheatsheet bezieht sich hauptsächlich auf IA-32-Assembler

³Bezeichnet das Übertragsbit, das in die Vorzeichenstelle hineingeht

⁴Bezeichnet das Übertragsbit, das aus der Vorzeichenstelle hinausgeht

Reverse-Engineering

Tools

.NET Reflector	Programm zum Dekompilieren von .NET-Programmen.
IDA	Vollständiger Name: Interactive Disassembler. Von Microsoft entwickelter Disassembler, der Skripting erlaubt.
ildasm	Einfacher GUI-basierter Disassembler für PE-Anwendungen, die IL-Code enthalten.
OlllyDbg	Etablierter Debugger für 32-Bit Anwendungen auf Windows.
WinDbg	Debugger für Windows Kernel- und Usermode, der die Analyse von crash dumps und CPU-Register erlaubt.

Obfuscation

Obfuscation bezeichnet allgemein eine Veränderung des Programmcodes, um die Lesbarkeit bzw. das Reverse Engineering des Programms zu erschweren. Das Verhalten des Programs soll dabei gleich bleiben.⁵ In den folgenden Unterabschnitten werden einige Techniken zur Obfuscation beschrieben.

Function-Splitting

Beim Function-Splitting wird eine Funktion f „kopiert“ (im Folgenden f' genannt) (und dann im Idealfall an einer ganz anderen Stelle im Programm abgelegt und inhaltlich möglichst weiter obfuscatet, damit man möglichst schwer erkennen kann, dass die beiden Funktionen inhaltlich das gleiche machen). Wenn im bisherigen Programm f von 2 Stellen (im Folgenden g1 und g2 genannt) aus aufgerufen wird, dann wird der Programmcode prinzipiell dahingehend angepasst, dass g1 f aufruft und g2 f' aufruft. Es ist dadurch schwerer erkennbar, dass an dieser Stelle g1 und g2 inhaltlich die gleiche Funktion ausführen.

Function-Merging

Function-Merging ist im Prinzip das Gegenteil vom Function-Splitting: Wenn es zwei Funktionen f1 und f2 gibt, werden diese ersetzt durch eine Funktion f3. Die Parameter von f3 sind inhaltlich die Summe der Parameter von f1 und f2 und (je nach Implementierung) noch ein Parameter um zu entscheiden, ob der Algorithmus von f1 oder f2 ausgeführt werden soll, wenn f3 aufgerufen wird.

Junk-Code

Junk-Code bezeichnet Programmcode, der zur korrekten Programmausführung nicht erforderlich ist. Er dient lediglich dazu, einem Reverse-Engineer mehr Arbeit zu machen, da es nicht immer leicht erkennbar ist, ob Code Junk-Code ist oder nicht.

Fake-Loops

Als Fake-Loops werden Loops (for-Loops, while-Loops, etc.) bezeichnet, die den Anschein erwecken sollen, dass der Schleifeninhalt öfters ausgeführt wird. In Wirklichkeit wird der Inhalt der Schleife jedoch nur einmal oder womöglich auch gar nicht ausgeführt (z. B. wenn sie ausschließlich mit Junk-Code gefüllt ist).

Decompilierung

Beim Dekompilieren wird aus einem kompilierten Programm der Quelltext rekonstruiert. Die Ausgabe eines Decompilers ist beispielsweise C-Code. Dieser Vorgang ist nicht eindeutig und automatisches Decompilieren liefert oft nur bedingt brauchbare Ergebnisse.

Disassemblierung

Als Disassemblierung bezeichnet man einen Prozess, der aus einem kompilierten Programm die Maschinencode-Befehle in Assembler-Befehle zurück übersetzt. Dieser Vorgang ist in aller Regel relativ eindeutig und automatisiert durchführbar.

Verhinderung von Disassemblierung

Unaligned Branches

Maschinencode-Befehle haben keine einheitliche Länge. Dadurch können Opcodes in anderen Opcodes versteckt werden können. Wenn diese Eigenschaft ausgenutzt wird, kommen beim seriellen Disassemblieren möglicherweise andere Befehlsabfolgen zu Stande als bei der Ausführung des Programms.

Anti-Debug-Maßnahmen

int 3

Die „int 3“-Instruktion wird von Debuggern benutzt, um einen Breakpoint zu setzen/zur Laufzeit zu erkennen. Wenn „int 3“ im bereits im Programmcode aufgefunden wird, deutet das auf eine Anti-Debug-Maßnahme hin. „int 3“ kann durch „nop“ („No operation“-Instruktion) ersetzt werden, um „int 3“ beim Debuggen zu überspringen.

Angehängte Debugger abfragen

Es gibt die Funktionen, um direkt abzufragen, ob ein Debugger an das Programm angehängt ist. Im Wesentlichen sind dies: `-IsDebuggerPresent` `-CheckRemoteDebuggerPresent` Dass diese Funktionen benutzt werden, kann ein Indiz dafür sein, dass das Programm Debugging erschweren möchte. Ein Programm kann sich in dem Fall beliebig anders verhalten, wenn mit diesen Methoden festgestellt wird, dass ein Debugger angehängt ist.

Timestamp-Analyse

Bei normaler Programmausführung werden Funktionen relativ schnell hintereinander ausgeführt. Wenn die Ausführung einer Funktion sehr viel länger dauert als normalerweise, ist dies ein Indiz dafür, dass in der Zwischenzeit ein Breakpoint getriggert worden ist und somit das Programm offensichtlich gerade analysiert wird. Ein Programm kann sich in dem Fall anschließend beliebig anders verhalten.

Virtuelle Maschinen

Es ist relativ leicht, zu erkennen, ob ein Programm in einer virtuellen Maschine ausgeführt wird. Programme können sich dementsprechend beliebig anders verhalten, wenn sie in einer VM ausgeführt werden. Da heute vor allem im kommerziellen Bereich aber grundsätzlich viele Programme in VMs laufen (z. B. Webserver etc.), macht diese Anti-Debug-Maßnahme nur bei Programmen Sinn, die darauf ausgelegt sind, normalerweise nicht in einer virtuellen Maschine zu laufen (z. B. bei Desktoprechnern von Privatpersonen).

Libraries

MSVCRT.DLL

Enthält die Funktionen der C-Standard-Bibliothek für den von Microsoft entwickelten Visual C++ Compiler von Version 4.2 bis 6.0.

.NET-Programme

Reverse Engineering von .Net-Programme ist relativ einfach. Dies hat im Wesentlichen 2 Gründe:

- Die originalen Bezeichner von Funktionen etc. werden ins kompilierte Binary einbezogen/übernommen und können beim Dekompilieren wieder ausgelesen werden.
- Der .NET-Kompiler erzeugt generell Common-Intermediate-Language-Code, aus dem die Programmstruktur und damit der Source-Code generell relativ gut rekonstruiert werden können.

Es gibt deshalb Tools, die den Reverse-Engineering-Vorgang für .NET-Programme sehr leicht machen (siehe Tools-Abschnitt).

Verschiedenes

Intrinsische Funktion

Breakpoints

Breakpoints werden beim Debuggen dazu benutzt, um die Ausführung eines Programms an einer bestimmten Stelle zu pausieren. Es gibt folgende Arten von Breakpoints:

Hardware-Breakpoints

Software-Breakpoints

⁵Auch über Seiteneffekte im Verhalten sollte das obfuskierte Programm wenn möglich nicht vom „Originalprogramm“ unterscheidbar sein.

Kryptographie

Arten von Kryptographie

Symmetrische Kryptographie

Bei der symmetrischen Kryptographie wird beim Ver- und Entschlüsseln der gleiche (geheime) Schlüssel verwendet. Wird eine symmetrische Verschlüsselung zur Kommunikation verwendet, müssen daher alle Teilnehmer den Schlüssel kennen. Es muss darauf vertraut werden, dass die anderen Teilnehmer den Schlüssel (weder gewollt noch ungewollt) weitergeben. Beispiele für symmetrische Verschlüsselungsalgorithmen sind AES, DES, Blowfish und Serpent.

Asymmetrische Kryptographie

Bei asymmetrischen Verschlüsselungsverfahren gibt es einen privaten und einen öffentlichen Schlüssel. Mit beiden Schlüsseln kann man eine Nachricht verschlüsseln und anschließend mit dem jeweils anderen Schlüssel entschlüsseln. Der private Schlüssel darf bei Kommunikation mit asymmetrischen Verschlüsselungsverfahren niemals an irgendwen weiter gegeben werden, während der öffentliche Schlüssel an weitere Personen weiter gegeben werden kann/darf/muss, da dies technisch erforderlich ist.

Benutzung von asymmetrischer Kryptographie

Verschlüsselte Kommunikation

Bei verschlüsselter Kommunikation mittels eines asymmetrischen Kryptographie-Systems wird der öffentliche Schlüssel zur Verschlüsselung und der private Schlüssel zur Entschlüsselung einer Nachricht benutzt.

Beispiel:

A möchte eine Nachricht an B senden. B hat sich dafür ein Schlüsselpaar (bestehend auf öffentlichem und privatem Schlüssel) erzeugt. A verschlüsselt die Nachricht mit dem öffentlichen Schlüssel von B.

Signatur

reinterpret_cast

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum

urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Zeiger
Smart Pointer
POD-Typen
ELF-Format
CLR

Programmiersprachen

Arten von Programmiersprachen

General purpose languages

Mit GPL-Programmiersprachen können eine Vielzahl von Programmen entwickelt werden. Beispielsweise kann man mit GPL-Programmiersprachen oft sowohl Konsolenprogramme als auch Programme mit einer grafischen Benutzeroberfläche programmieren. GPL-Programmiersprachen beinhalten darüber hinaus typischerweise oft Elemente wie If,/else-, for-, while-Statements und erlauben das Programmieren und den Aufruf von „Unterfunktionen“. Ferner sind GPL-Programmiersprachen in aller Regel turing-vollständig.

Domain specific languages

DSL-Programmiersprachen sind nicht dafür geeignet, beliebige Softwareentwicklungs-projekte zu realisieren. Die Anwendbarkeit von DSL-Programmiersprachen sind auf (oft sehr wenige) „Domänen“ beschränkt. Beispiel: SQL. SQL-Statements sind dafür geeignet, um Daten aus einer Datenbank abzufragen, hinzuzufügen oder zu editieren. In dieser „Domäne“ hat SQL allen anderen Programmiersprachen einen wesentlichen Vorteil. Im Gegensatz zu GPL-Programmiersprachen ist SQL (wie auch alle anderen DSL-Programmiersprachen) allerdings ungeeignet, um etwa ein ausführbares Programm zu entwickeln.

Deklarative Sprachen

Ausführung des Programmcodes zur Laufzeit

Assembler

Ein Assemblierungsprogramm übersetzt Assembler-Befehle direkt in Maschinencode. Beim Umkehrvorgang (Disassemblieren) kann der Assemblercode weitestgehend wieder hergestellt werden. Direkt in Assembler zu programmieren ist heute nur noch selten gebräuchlich und nicht praxistauglich für große Softwareprojekte. Für spezielle Anwendungen kann es allerdings manchmal sinnvoll (oder sogar erforderlich) sein, Assemblercode zu schreiben. Anbieten tut sich dies bei speziellen Aufgaben, die hinsichtlich der Performance optimiert werden sollen. Da Programmierung in Assembler die hardware-nächste Art ist, zu programmieren, die in der Praxis auffindbar ist.

Kompilierung

Bei Sprachen wie C oder C++ wird der Programmcode kompiliert. Das bedeutet, dass der Programmcode vom Compiler in Maschinencode-Befehle übersetzt wird, wodurch

der Programmierer durch den Kompilier-Vorgang eine ausführbare Datei (z. B. .exe-Datei) erhält. Dies kann mit Aufrufen wie
gcc MyProgram.c -o MyProgram.exe
erfolgen. Programme von den meisten GPL-Programmiersprachen müssen vor der Ausführung kompiliert werden. Der Umkehrvorgang zur Kompilierung ist Dekompilierung. Es gibt Tools, zur Dekompilierung von Programmen, allerdings ist die Dekompilierung nicht eindeutig und bringt daher als Ergebnis nicht den Original-Programmcode eines Programms hervor. Dies kann im Wesentlichen folgende Gründe haben:

- Der Compiler optimiert den Programmcode in der Regel soweit wie möglich, um ihn beispielsweise performanter zu machen.
- Es gibt mehrere verschiedene Programmcode-„Varianten“ für verschiedene (wenn auch sehr ähnliche) Programmcode-Statements, die zum gleichen Maschinencode führen. Der Decompiler kann nur schwer feststellen, welche dieser Varianten im Original-Programmcode verwendet worden ist.
- Der Programmcode wurde obfuskiert. Hier werden beim Compilieren (oder ggf. auch davor oder danach) technische Maßnahmen angewandt, um es Dekompilieren und Reverse-Engineerern möglichst schwer zu machen, den Programmcode wieder herzustellen.

Interpretation

Bei Interpretation wird ein Programm nicht kompiliert, sondern der Programmcode selbst wird dem Anwender ausgeliefert. Programmiersprachen, deren Name auf „Script“ endet, sind in aller Regel Programmiersprachen, die interpretiert werden. Der Anwender braucht zum Ausführen des Programms einen Interpreter. Dieser „Interpretationvorgang“, der zur Ausführung des Programms führt, dauert länger als die Ausführung eines kompilierten Programms, da z. B. erst der Programmcode geparkt werden muss. Programme von DSL-Programmiersprachen werden oft interpretiert statt kompiliert. Browser haben daher z. B. immer einen Javascript-Interpreter und Datenbanken haben z. B. immer einen SQL-Interpreter. Dadurch dass der Programmcode selbst zur Ausführung gebraucht wird, sind Programme einer DSL immer „open source“⁶, bzw. genauer gesagt: Ein Anwender des Programms braucht (neben dem Interpreter) den Programmcode und kann ihn daher im Gegensatz zu kompiliertem Code beispielsweise theoretisch auf Fehler/Schadcode überprüfen. Dies wird jedoch dadurch erschwert, dass Programmcode von DSL-Programmiersprachen

oft „uglified“ und „minified“ an den Anwender herausgegeben wird, um Speicherplatz bei der Übertragung zu sparen⁷.

Skripte

Als Skript bezeichnet man eine Datei, welche beliebig viele (von Menschen lesbare) Befehle enthält, die bei Ausführung von einem geeigneten Interpreter interpretiert werden. Gängige Beispiele sind:

- Javascript-Dateien
- SQL-Dateien
- Kommandozeileninterpreter-Skripte (z. B. .bat-Dateien, .sh-Dateien; Diese Dateien brauchen keinen externen Interpreter, da der Interpreter direkt im Betriebssystem verfügbar ist)

Skriptdateien sind nicht zu verwechseln mit Konfigurationsdateien.

Zwischensprache

Einige Programmiersprachen (im Wesentlichen C#, VB.Net und Java) kompilieren den Code zwar scheinbar zu einer ausführbaren Datei, jedoch beinhaltet diese ausführbare Datei nicht direkt Maschinencode, sondern einen (für Menschen nicht lesbaren) Zwischencode (sogenannten Bytecode), der von einer installierten Laufzeitumgebung interpretiert und ausgeführt wird. Dieser Zwischencode ist stark optimiert um die Ausführungsgeschwindigkeit zu erhöhen, trotzdem sind Sprachen, deren Programme in eine Zwischensprache kompiliert wird⁸, langsamer als Sprachen, die direkt zu richtigem Maschinencode kompiliert werden.

Um .Net-Programme (Programme, die mit C# (oder anderen .Net-Programmiersprachen von Microsoft) geschrieben worden sind) auszuführen, wird das .Net-Framework benötigt, welches quasi nur auf Windows anwendbar ist⁹. Dies kann kostenfrei gedownloaded/benutzt werden und ist bei Windows 10 beispielsweise vorinstalliert.

Um Java-Programme auszuführen, wird eine installierte Java-Runtime-Environment benötigt. Diese ist für viele gängige Plattformen kostenfrei verfügbar.

Generationen von Programmiersprachen

Programmiersprachen erster Generation

Programmiersprachen erster generation sind quasi keine richtigen Programmiersprachen. Bevor die späteren Generationen von Programmiersprachen entwickelt worden sind, mussten Programmierer den Maschinencode selbst schreiben. Es gab keinen Compiler oder ähnliches. Die Programmierarbeit war äußerst aufwendig. Diese Verfahren sind heute in der Praxis nicht mehr gebräuchlich.

⁶Dies heißt natürlich nicht, dass Programme von DSL-Programmiersprachen immer Lizenzkostenfrei sind. Die Lizenz eines Programms einer DSL-Programmiersprache ist zu prüfen unabhängig davon, ob man dessen Programmcode besitzt oder nicht.

⁷Diese Optimierung wird speziell bei Javascript sehr extrem angewandt, um die Anzahl an Daten zu verkleinern, die ein Internetsurfer beim Aufruf einer Seite downloaden muss, ohne deren Semantik zu verändern.

⁸Das Wort „kompilieren“ wird im Allgemeinen auch für den Vorgang benutzt, in dem aus dem Programme von Sprachen wie C#, Java usw. in scheinbar ausführbare Dateien erzeugt werden.

⁹Es gibt mit „Net Code“ und „Net Standard“ Ansätze von Microsoft, um .Net-Programme auch auf anderen Plattformen lauffähig zu machen

Programmiersprachen zweiter Generation

Programmiersprachen dritter Generation bezeichnen üblicherweise Assembler-Programmiersprachen. Sie zeichnen sich dadurch aus, dass es ein Assemblierungsprogramm gibt, der die definierten für Menschen lesbaren Befehle in Maschinencode übersetzt.

Programmiersprachen dritter Generation

Programmiersprachen dritter Generation sind Sprachen wie C, Fortran, COBOL, Pascal, C#, C++. Sie wurden entwickelt, um dem Programmierer die mühselige Arbeit zu entwickeln, Assemblercode zu schreiben. Sie zeichnen sich dadurch aus, dass es einen Compiler gibt, der aus dem Programmcode Maschinencode oder Code einer Zwischensprache erzeugt.

Gängige Programmiersprachen

C

C ist relativ hardwarenah und sehr weit verbreitet. C bietet sich für Softwareprojekte an, die auf mehreren Plattformen ausgeführt werden sollen, da es für quasi jede heute gebräuchliche Plattform einen C-Compiler gibt. Es ist möglich, Assemblerprogramme in C zu integrieren, um z. B. bestimmte Aufgaben, bei denen Performance wichtig ist, direkt in Assembler programmieren zu können.

C++

C++ ist eine Erweiterung der Sprachspezifikation von C. Jedes C-Programm ist daher mit einem C++-Compiler kompilierbar. C++-Programme, sind, wenn sie C++-spezifische Features benutzen, umgekehrt nicht mit einem C-Compiler kompilierbar. C++ bietet diverse Features wie etwas Klassen/Mehrfachvererbung und ist eine relativ häufig eingesetzte Programmiersprache.

Javascript

Javascript (nicht zu verwechseln mit Java) wird überwiegend auf Websites eingesetzt, um diese interaktiv zu gestalten.

C#

C# ist eine verbreitete Programmiersprache von Microsoft, deren Syntax an C angelehnt ist und es Entwicklern relativ einfach machen soll, schnell auch komplexe Anwendungen zu entwickeln. C#-Programme werden beim kompilieren in Bytecode der Common Intermediate Language übersetzt und benötigen das .Net-Framework zur Ausführung. C# erlaubt die Verwendung von Bibliotheken, die in VB.Net geschrieben worden sind.

VB.Net

VB.Net (auch „Visual Basic .Net“, nicht zu verwechseln mit VB¹⁰) ist eine programmiersprache, die de facto die gleiche Mächtigkeit und den gleichen Zweck hat wie C#, syntaktisch aber auf VB basiert. Darüber hinaus ist VB.Net genau wie C#

auch von Microsoft entwickelt. VB.Net-Programme werden beim kompilieren in Bytecode der Common Intermediate Language übersetzt und benötigen das .Net-Framework zur Ausführung. Der Fokus bei VB.Net liegt auf eine Syntax, die an die menschliche Sprache angelegt ist, um es Programmieranfängern zu ermöglichen, Programmcode einer objektorientierten Programmiersprache einfacher lesen zu können. Programmcode von VB.Net und C# ist bis auf wenige Ausnahmen ineinander konvertierbar. Einige Ausnahmen bzw. Unterschiede zu C#:

- In VB.Net sind readonly-properties parametrisierbar, bei C# nicht.
- In VB.Net sind Funktionsnamen case-insensitive. (In C# sind sie case-sensitive.)

Langfristig plant Microsoft, die Feature-Entwicklung auf C# zu fokussieren und weniger Wert auf weitesgehende Konvertierbarkeit von C#- und VB.Net-Programmcode zu legen, sodass die Anzahl der Features, die C# hat und nicht in VB.Net nutzbar sind, in Zukunft steigen dürfte. Vb.Net erlaubt die Verwendung von Bibliotheken, die in C# geschrieben worden sind (eventuell treten Probleme wegen unterschiedlicher case-sensitivity auf).

VB

VB (auch „Visual Basic“, nicht zu verwechseln mit VB.Net) ist eine Skriptsprache und der Vorläufer von VB.Net. Syntaktisch liegt der Fokus auf eine einfache Lesbarkeit. VB-Skripte werden heute noch oft in Excel-Dokumenten verwendet.

Java

Java (nicht zu verwechseln mit Javascript) ist eine Programme, die in eine Zwischensprache kompiliert wird, ähnliche wie C#.

Typescript

Typescript ist eine Erweiterung von Javascript, die Elemente aus der objektorientierten Programmierung beinhaltet. Es soll damit für Entwickler möglich sein, die zunehmend komplexer werdende Dynamik von Websites mit Unterstützung von Elementen wie Klassen und Interfaces zu entwickeln, die es so in Javascript nicht direkt gibt. Typescript-Code wird bei Produktivsetzung in Javascript übersetzt („transkompiliert“).

Coffeescript

Coffeescript ist (ähnlich wie Typescript) eine andere Programmiersprache um Javascript-Code zu schreiben. Coffeescript legt den Fokus auf bessere Lesbarkeit und Vermeidung von Klammern. Coffeescript-Code wird bei Produktivsetzung in Javascript übersetzt („transkompiliert“).

Lua

Ist eine imperative Skript-Sprache, die in vielen Arten von Programmen (auch oft in Spielen) zum Einsatz kommt, um Dinge einfach konfigurierbar/skriptbar zu machen.

SQL

SQL ist eine Abfragefrage für Datenbanken. SQL gehört zu den domänenspezifischen Sprachen.

XPath

XPath ist eine Abfragesprache um Daten aus einem XML-Dokument auszulesen. XPath erfüllt analog anschaulich gesagt den gleichen Zweck wie SQL für Datenbanken. XPath gehört zu den domänenspezifischen Sprachen.

XQuery

XQuery ist eine Abfragesprache um Daten aus einem XML-Dokument auszulesen. XQuery ist mächtiger als XPath, da es wesentlich mehr Funktionen zur Verfügung hat. XQuery gehört zu den domänenspezifischen Sprachen. Ferner ist XQuery ferner turing-vollständig.

Python

Python ist eine Skriptsprache, die es ermöglicht, komplexe längere Programme zu schreiben, die vom Funktionsumfang her GPL-Programmiersprachen nahekommt. Python ist plattform-unabhängig und wird benutzt um diverse Sachen einafch und schnell skripten zu können.

PHP

PHP ist eine Skriptsprache, die vor z. B. oft bei Webservern eingesetzt wird, um den Inhalt einer Website beim Abruf dynamisch zu erzeugen (durch PHP-Skripte berechnen zu lassen).

Go

Go ist eine relativ neue ursprünglich von Google entwickelte Programmiersprache, die kompiliert wird und den Fokus darauf legt, Nachteile von C++ oder Java etwa bei hinsichtlich der Entwicklung für skalierbare Netzwerkdienste zu vermeiden.

Smalltalk

Smalltalk ist eine sehr alte objektorientierte Programmiersprache. In Smalltalk ist „alles“ ein Objekt, also auch Zeiger und „primitive Daten“ wie etwa Integer- und Boolean-Werte. Smalltalk wird in Bytecode kompiliert.

Objective-C

Objective-C ist eine Erweiterung von C. Objective-C-Kompiler können daher auch C-Programme kompilieren.

Swift

Swift ist eine von Apple entwickelte Programmiersprache, die primär für die Ausführung von Programmen auf Apple-Geräte konzipiert worden ist.

R

R ist eine domänenspezifische Programmiersprache, die für statistische Berechnungen konzipiert worden ist.

¹⁰Umgangssprachlich wird oft „VB“ bzw. „Visual Basic“ gesagt, wenn in Wirklichkeit „VB.Net“ bzw. „Visual Basic .Net“ gemeint ist.

