Cheatsheet IT-Forensik

by Matthias Bissinger, Marius Göcke, Michael Koll

Betriebssystemforensik (allgemein)

Betriebssystem

Architektur

Monolithisch

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

Geschichtet

`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

Mikrokernel

`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

`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 (unfaires Scheduling)

Aufgaben

`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

`Belegungstabelle`	Datei, Start, Länge

Verteilte Belegung verkettete Listen (FAT)

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

Verteilte Belegung mittels Index-Liste

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

Windows

Allgemein

Windows Stations, Desktops und Session

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)

- mit `tlist` laufende Prozesse mit Diensten auflisten (`tlist -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

`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

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
`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`

Überwachter Ordnerzugriff

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

Aktivieren

Windows Defender Security Center → Einstellungen für Viren- und Bedrohungsschutz → Überwachter Ordnerzugriff
`oder`
Gruppenrichtlinien: `Computerkonfiguration/Administrative Vorlagen/Windows/Windows Defender Antivir/Windows Defender Exploit Guard/Überwachter 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 -ControlledFolderAcessAllowedApplications «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 Versionssnummer (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-ID
`Volume-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

CustomDestionations 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 Explorer.
Dateiname `5f7b5f1e01b83767.automaticDestinations-ms`

Tools

`JumpListExt for Windows 10`	grafische Oberfläche, nicht mehr stabil in aktuellen 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 Datan, 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 Partitionstabellenart (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 runas '%0' am_admin & exit)
```

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_D
```

Schalte das Speichern von Thumbnails aus:

```
Windows Registry Editor Version 5.00

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

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

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer]
"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`
- 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)
4. Aktualisierte Daten in Logdatei (LC→Datei)
5. Datenbank aktualisieren

Dirty-DB

Datenbank, die nicht vollständig aktualisiert wurde.

`V01.chk`	Zeitpunkt der Transaktion
`*.log`	Transaktionsdaten, hexadezimale Dateinamen

Wiederherstellung mit esentutl

`esentutl /mh database.dat`	Überprüfung der Datenbank (Feld State=Dirty)
`esentutl /r database.dat`	Reparatur der Datenbank (Feld State=Clean)

WebCacheV01.dat

Pfade

→`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	Erstellungzeit 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,
                            UserCID, UserFolder
```

Konfigurations- und Diagnostikdaten

Ausgehend vom One-Drive-Verzeichnis:

.\logs\Personal\ SyncDiagnostics.log	Down-\Uploadgeschwindigkeit, Ausstehende Down-\Uploads, verfügbarer Speicherplatz lokal, UserCID (siehe REG), Anzahl Dateien und Verzeichnisse
.\settings\Personal\ <usercid>.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	Writhe 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	Benachrichtigunginhalt → 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 createdTime, 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. automaticDestinations-ms	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 zulegt 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\CurrenControlSet\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
`HLKM\SECURITY`	SECURITY
`HKLM\SOFTWARE`	SOFTWARE
`HLKM\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: `HLKM\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 NT/CurrentVersion/CurrentBuildNumber`	Windows Buildnummer (cmd: `systeminfo`)

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/AppContainer/Storage/microsoft.microsoftedge_xxxxxx/MicrosoftEdge`

Netzwerke

WLAN

`HKLM/Software/Microsoft/Windows NT/CurrentVersions/NetworkCards`	Netzwerkgeräte (Beschreibung und GUID)
`HKLM/System/CurrentControlSet/Services/Tcpip/Parameters/Interfaces/<GUID>`	Details zum Netzwerkgerät (IP, Gateway, Domain)

P2P

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

Angeschlossene Geräte

`HKLM/System/Mounted Devices`	Liste aller Geräte, die im System gemountet wurden
`HKCU/Software/Microsoft/Windows/CurrentVersion/Explorer/MountPoints2`	Mount eines Geräts bei Nutzerlogin
`HKLM/System/CurrentControlSet/Control/DeviceClasses`	Enthält für jede 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/<Enumerator>/<DeviceID>`	Geräte im System mit Gerätebeschreibung und IDs
`HKLM/System/CurrentControlSet/Enum/USBSTOR`	Angeschlossene USB-Geräte

Antiforensische Maßnahmen

`Zeitstempel fÃdlschen`	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/Contol/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 Registry Viewer`	Auslesen von Hive-Files (`https://accessdata.com/product-download/registry-viewer-1-8-0-5`)
`RegView`	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 bezeihnet das Mitschneiden bzw. Analysieren von Netzwerkdatenverkehr. Dies kann im Wesentlichen entweder durch einen man-in-the-middle-Angrif 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 einafch 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).

[1] Dies 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 analyiseren 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.

Alternate Data Streams

Bei NTFS-Systemen gibt es Alternate Data Streams (ADS). Obgleich es viele legitime Einsatzzwecke für ADS gibt, werden sie auch oft benutzt, um Daten zu verstecken. ADS sind Daten, die zu einer Datei hinzugefügt werden können, aber nicht Bestandteil von der Datei oder dessen Metadaten sind und standardmäßig nicht in Windows-Explorer etc. angezeigt werden. Eine Datei kann mehrere ADS haben. Ein ADS ist technisch gesehen eine Datei und der zu versteckende Inhalt wird in genau diese Datei geschrieben.
Beispiele:
Anlegen (Windows):

```
echo \$null > test.txt:hidden.txt
```

Durch diesen Befehl wird hidden.txt als Alternate Data Stream von test.txt angelegt. Falls test.txt nicht bereits existiert, wird diese Datei ebenfalls erstellt.
Finden (Windows):

```
dir /R
```

Der dir-Befehl ohne Argumente zeigt hidden.txt nicht an. Mit dem /R-Schalter hingegen wird hidden.txt aufgelistet.
Schreiben von Daten (Windows):

```
echo testcontent >test.txt
```

Mit diesem Befehl können beliebige Daten in test.txt geschrieben werden. Dies beeinflusst weder die Existenz noch den Inhalt von hidden.txt

```
echo hiddencontent > test.txt:hidden.txt
```

Mit diesem Befehl können beliebige Daten inhidden.txt geschrieben werden. Dies beeinflusst weder die Existenz noch den Inhalt von test.txt
Auslesen von Daten (Linux):

```
cat test.txt:hidden.txt
```

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] xor out-carry[4]
- IOPL (IO-Privilege-Level): TODO
- NT (Nested-Task): TODO
- RF (Resume-Flag): TODO
- VM (Virtuel-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

[2] Dieses Cheatsheet bezieht sich hauptsächlich auf IA-32-Assembler
[3] Bezeichnet das Übertragsbit, das in die Vorzeichenstelle hineingeht
[4] Bezeichnet 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.
`OllyDbg`	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-Engineerer 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 Assebler-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 Anit-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

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