A Szenario – Auftragsbeschreibung

Nach der Erstinstallation eine Windows-Client-Rechners wird man die von den Benutzern benötigten Anwendungsprogramme installieren und konfigurieren.

In diesem Szenario haben Sie von der Geschäftsführung den Auftrag erhalten, für die Standardanwendungen (u.a. aus Sicherheitsüberlegungen) nach Möglichkeit nur Open Source Software einzusetzen *(siehe weiter unten).*

Nach der Installation wird man die korrekte Funktion der Programme überprüfen sowie gegebenenfalls etwaigen Fehlermeldungen nachgehen – dazu ist eine Kontrolle über ein Programmablaufprotokoll hilfreich. Eine solche Überwachung ist auch angebracht, um verdächtigen Programmaktivitäten auf den Grund zu gehen ist – wir werden hierzu das Microsoft Tool *Process Monitor* einsetzen.

Ihr heutiges Protokoll sollte die vorgenommenen Schritte zur Vorbereitung sowie Installation, Konfiguration und Überwachung der Anwendungs-Software sowie die *Antworten auf alle gestellte Fragen* enthalten!

B Vorbereiten der VM

(1) Legen Sie für diese Übung einen neuen virtuellen PC auf der

Basis der vorbereiteten Windows 11 Standardinstallation (als *Linked Clone* … Snapshot nicht vergessen) an!

C Auswahl der Anwendungen

Bevor Sie mit der Installation beginnen können, werden Sie erst einmal geeignete Anwendungen wählen und testen müssen. Der (etwas steinige) Weg dieser Anwendungsauswahl wurde in unserem Szenario bereits vorgenommen. Im Folgenden finden Sie eine beispielhafte Mini-Liste von Softwarepaketen für Standardanwendungen, die den Anforderungen der Geschäftsführung genügen (diese Liste ist natürlich in der Regel viel länger):

Achtung: Wir werden **nicht** alle diese Anwendungen installieren, sondern nur **ein** Beispiel (nämlich Firefox)!

D Systemvorbereitung

(2) *Sicherung:* Erstellen Sie noch vor dem Start der VM zumindest

einen *VM-Snapshot* – für den Fall, das etwas mit der Installation schief geht … (in der Realität sollten Sie jetzt eine komplette [Systemabbild-Sicherung](https://support.microsoft.com/de-de/windows/sichern-und-wiederherstellen-in-windows-352091d2-bb9d-3ea3-ed18-52ef2b88cbef) *(Image Copy)* des Systems erstellen, zumindest jedoch einen Windows-*Wiederherstellungspunkt*).

Welchen (sinnvollen) Namen geben Sie Ihrem Snapshot?

E Anwendungsinstallation

(3) *Installationsplanung*: Im Rahmen dieser Übung werden Sie sich beispielhaft auf eine einzige Anwendung konzentrieren: Den beliebten Open-Source-Browser *Mozilla Firefox* in der neuesten, beschleunigten Version. Bevor so eine Software für den Produktiveinsatz installiert wird, sollte man einige Voraussetzungen für einen sicheren und legalen Betrieb überprüfen:

(a) Informieren Sie sich auf der Internetpräsenz von *Mozilla Firefox (Tipp: Die Hyperlinks anklicken!*) über:

- Systemanforderungen? *(System Requirements)*
- Aktuelle, stabile Version? *(Release Notes)*
- Installationsprozess?
- (Lizenzbestimmungen - *siehe nächster Punkt!*)

(b) Sofern – bis auf die Lizenz-Details – alle Fragen geklärt *(und dokumentiert)* werden konnten, laden Sie die aktuelle Version des Installationsprogramms herunter!

i) Während des Downloads überprüfen Sie jetzt bitte die Lizenzbestimmungen: Darf das Programm auch für kommerzielle Zwecke frei genutzt werden? (*Tipp:* EULA -> *Mozilla Public License 2 -> 2.1 Grants*)? ii) Was ist überhaupt ein(e) EULA? Welche Lizenz gilt für Firefox? *(Siehe zuvor!))* iii) Darf man überhaupt eine lokale Kopie z.B. am Schulserver anbieten? *(Tipp: Mozilla Public License - FAQ -> Q6)* (c) Sodann installieren Sie *Firefox* (am besten in der 64-Bit-Version) und machen ihn zu Ihrem Standard-Browser!

F Programmüberwachung mit Process Monitor

Programmablaufprotokoll: Wir wollen nun beispielhaft untersuchen, auf welche Windows-Ressourcen die neu installierte Anwendung (hier: Firefox) zugreift. Man macht das etwa beim *Troubleshooting*, wenn man feststellen möchte, ob es evtl. Zugriffsprobleme (falsche Rechte) auf bestimmte Dateien oder gar die Windows-Registry gibt, oder z.B. wenn man vermutet, dass einzelne Programme selbstätig eine Netzwerkverbindung auf einen fremden Server durchführen und dabei unwerwünscht Daten übertragen.

Microsoft stellt für solche Zwecke unter `live.sysinternals.com` das *Windows-Sysinternals*-Tool *Process Monitor* (`Procmon.exe`) bereit, das hilft, Zugriffe von laufenden Programmen (= Prozesse) auf das Dateisystem, auf die *Registry* sowie auf das Netzwerk offenzulegen und eventuelle Probleme zu erkennen. Auch kann man feststellen, ob das Programm eigenständig andere Programme startet. *Process Monitor* protokolliert jede dieser Operationen und deckt dabei z.B. fehlende Zugriffsrechte oder gar unerwünschte Zugriffsversuche bzw. Netzwerkaktivitäten auf. Mit anderen Worten: Mit *Procmon* können Sie recht gut feststellen, "was das Programm gerade macht"!

Hinweis: Falls *Process Monitor* auf Ihrem Labor-Windows-System *nicht* installiert ist, sollten Sie das Tool von Sysinternals herunterladen (ZIP-Datei in Ordner Ihrer Wahl auspacken, dann daraus einfach `Procmon.exe` aufrufen).

Komfortabler ist die Installation im Terminal (mit Administratorrechten), indem man die Anwendung mit dem *Windows-Paket-Manager* herunterladet:

```
winget source update
winget install sysinternals
```

Das Programm `winget` durchsucht Quellen wie den Microsoft Store nach der angegebenen Software, um sie herunterzuladen und zu installieren. Um die Paketdatenbank zu aktualisieren, verwendet man `winget source update`. Zusätzlich kann man mit dem Paket-Manager auch Software wieder deinstallieren (Parameter `uninstall`) oder auf dem aktuellen Stand halten (Parameter `upgrade`).

F.1 Tastenkünste

(4) Starten Sie `Procmon.exe` und beobachten Sie kurz die überwachten Operationen – mit welchem Tastaturkürzel kann man …

(a) Ein- und Ausschalten, ob der Bildschirminhalt laufend die neuesten Operationen anzeigt *(Autoscroll)*?

(b) die Überwachung *(Capture)* ausschalten, so dass keine Operationen mehr überwacht werden?

(c) das aktuelle, am Bildschirm angezeigte Protokoll löschen?

(d) die Überwachung wieder einschalten (Woran erkennt man in der Icon-Leiste, ob überwacht wird oder nicht?)?

(5) Wie kann man …

(a) nur Dateioperationen (Zugriffe auf das Dateisystem) **und** Registry-Zugriffe betrachten?

(b) ausschließlich Netzwerkaktivitäten sehen?

(6) Kann man auch die die übertragenen *Inhalte* (Daten in den Netzwerkpaketen) der beobachteten Netzwerkaktivitäten betrachten?

(7) Mit welchem Tastenkürzel (*Tipp: Tools → ...*) erhält man eine Übersicht über alle überwachten laufenden Programme (nennt man *Prozesse*)? Was bedeutet die Einrückung beim Programmnamen in dieser *Process Tree*-Darstellung?

F.2 Nur mit Filter

(8) Starten Sie einen Firefox-Browser und verwenden Sie die "Zielscheibe", um ausschließlich den gerade sichtbaren

Firefox-Browser-Prozess zu überwachen!

(8) Checken Sie die aktuellen Filter-Einstellungen:

(a) Welches Tastenkürzel bringt Sie eigentlich rasch zu den Filter-Einstellungen?

(a) Welche Filterregel (Tipp: Grün) sorgt nun dafür, dass nur Operationen des Firefox-Browsers überwacht werden?

(a) Wie wird standardmäßig verhindert, dass auch die Aktivitäten von *Procmon* selbst (`Procmon.exe`) angezeigt werden **(evtl. mit Screenshot zeigen)**?

(a) Wie kann man die Filterregeln auf den Ausgangszustand (alles zeigen bis auf wenige sinnvolle Ausnahmen) zurückstellen?

F.3 Big Brother

(10) Setzen Sie nun einen Filter, so dass nur Aktivitäten des Firefox-Browsers (*Process Name* `firefox.exe`) dargestellt werden - wie lautet die neue Filterregel? Testen Sie die Regel (es sollten nur Firefox-Ereignisse sichtbar sein)!

(11) Stoppen Sie die Überwachung, löschen Sie alle Ereignisse, beenden Sie Firefox und starten Sie wieder das Logging!

(12) Starten Sie den Browser, gehen Sie auf eine vertraute Web-Seite. Nach ein paar Sekunden stoppen Sie das Logging *(Capture)* wieder.

Jetzt werden wir die mitgeloggten Aktivitäten analysieren:

a) *Process Activity:* Welche Programme (bzw. wie oft) wurden offenbar nach dem Klicken auf das Firefox-Icon gestartet (*Tipp: Tools → Process Tree*)? Was fällt Ihnen auf? Wie lauten die tatsächlich aufgerufenen Kommandozeilen *(es reichen 2 Beispiele ;-)*?

b) *Network Activity:* Stellen Sie fest, mit welchen Netzwerkadressen sich der Firefox-Browser verbindet (*Tipp:* Übersicht über *Tools → Network Summary*)! Screenshot ins Protokoll! Untersuchen Sie ein paar Adressen: Insbesondere die nicht über DNS aufgelösten Adressen sind von Interesse. Haben Sie vielleicht doch einen Trojaner entdeckt *(Tipp: whois bzw. IP Lookup)*?

c) *File Activity I:* Wie kann man überprüfen, welche Dateien Firefox beim Starten geöffnet hat *(tabellarische Übersicht)*? Wie viele verschiedene Dateien und Ordner waren das eigentlich **(screenshot!)**?

d) *File Activity II:* Unter welchem Ordner **(absoluter Pfad ins Protokoll!)** wird offensichtlich die gesamte (wichtige) Benutzerprofilinformation des Browsers *für diesen User* (bei uns `junioradmin`) gespeichert? **Tipp:** Firefox kann mehrere `Profiles` für einen User verwalten...

e) *Registry Activity:* Stellen Sie fest, mittels welcher Registry-Einstellung Firefox nachfragt, mit welcher Sprache – etwa `en-US` – auf dieser Maschine bevorzugt gearbeitet wird (**Tipp:** Nach `Language` suchen, mit Strg-F immer weitersuchen bis `SUCCESS`)!

-> Kompletter Registry-Pfad `HKLM\...` oder `HKCU\...` ins Protokoll und den Zugriff mit einem Procmon-Screenshot im Protokoll und/oder Eduvidual dokumentieren!

Bonus: Wie kann man diese Information durch geschicktes Setzen der Filter zusammengefasst im *Registry Summary* erhalten **(Screenshot!)**?

F.4 Gedächtniskünste

(13) Manchmal möchte man das beobachtete Geschehen festhalten und später genauer auswerten - wie kann man **(ausprobieren!)** …

(a) den aktuellen, am Bildschirm angezeigten Log löschen?

(a) eine (früher) gespeicherte Log-Datei wieder einlesen?

(a) den aktuellen Überwachungsinhalt in eine Log-Datei schreiben (Speichern Sie testweise den aktuellen Inhalt in eine Datei IhrVorname_xxxx_Datum.PML auf dem Desktop!)? xxxx=Ihre Login-Nummer. Öffnen Sie die Datei mit einem Doppelklick. Was können Sie jetzt damit machen? Fügen Sie einen Screenshot des gesamten Fensters (mit Titelzeile) im Protokoll ein.

G Durchsicht der Autostart-Programme

Was noch fehlt ist ein Untersuchung, ob die neu installierte Anwendung sich selbst oder andere Komponenten beim Hochfahren des Rechners oder beim Neuanmelden automatisch startet. Dazu bietet sich das *Windows-Sysinternals*-Tool *Autoruns* (`Autoruns.exe`) an, das wirklich (fast) **alle** Möglichkeiten des Autostarts unter Windows erkennt (nicht nur den Weg über die `Run`-Einträge in der *Windows Registry* aus der gleichnamigen Übung).

(14) Starten Sie nach dem Entpacken *Autoruns* und machen Sie sich mit der Darstellung vertraut **(Screenshot)** – wie viele Einträge findet das Tool (so ca.)?

(15) Suchen Sie alle Einträge der Mozilla Corporation (die Firefox entwickeln) - welche Firefox-relevanten Auto-Start-Einträge finden Sie?

(16) Machen Sie den Vergleich: Wenn Sie *Autoruns* auf Ihrem eigenen PC oder Laptop laufen lassen (keine Gefahr ;-)) – wie schaut es da aus? Erschrecken

Sie nicht!

H Super-Bonus: Trojaner *analysieren*

(17) *Nur für Sehr Fortgeschrittene:* Laden Sie einen echten Virus oder Trojaner (etwa von `dasmalwerk.eu` oder einen "Trojaner-Simulator", etwa selbst gemacht mit `msfvenom` bzw. `Veil` unter Kali Linux…) auf eine entsprechend abgesicherte Windows-VM (bitte **nur** am Schul-PC!), wenn Sie sich trauen ;-)

und betrachten Sie das Programm mit *Process Monitor* beim Werken!

Viel Spaß!