CLOUDSUPPLIES

MILTON SECURITY WHITEPAPER

Härtungsmaßnahmen von Server und Anwendung

Zusammenfassung

MILTON ist ein zentrales System, dass mithilfe Künstlicher Intelligenz die Automatisierung des Betriebs von IT-Infrastrukturen ermöglicht. Dabei ist die IT-Sicherheit für die IT-Infrastruktur oberstes Gebot und ständiger Bestandteil bei der Entwicklung von MILTON. In diesem Dokument werden einzelne Maßnahmen zur Härtung, Sicherheitsvorgaben und Prozesse beschrieben und erläutert.

Inhaltsverzeichnis

Tabellen und Abbildungsverzeichnis

Dokumentenhistorie

Version	Datum	Änderung	Autor
1.0	10.10.2022	Initiale Erstellung	Ullrich Weichert
1.1	15.05.2023	Revision und kleinere Updates	Thomas Müller

Tabelle 1: Dokumentenhistorie

Allgemeines

MILTON wurde auf Basis unserer langjährigen Erfahrungen im Betrieb von IT-Infrastrukturen entwickelt. Darunter wurden Bedarfe von IT-Experten des Betriebs genauso berücksichtigt, wie auch die Belange des Geschäftsbetriebs an Stabilität, Performance und Verfügbarkeit.

Der #digitalAdmin MILTON wurde speziell für den automatisierten Betrieb von IT-Infrastrukturen entwickelt.

Dieser Anwendungszweck setzt weitgehende Verzahnung und Kommunikation in die IT-Infrastruktur voraus. Denn für die Erfüllung seiner Aufgabe benötigt MILTON den Zugriff auf das System oder die Applikation mit zumindest dem Berechtigungsset, welches für die Durchführung dieser Aufgabe erforderlich ist.

Soll MILTON abgelaufene Kennwörter für Benutzer in der IT-Infrastruktur im zentralen Benutzerkatalog automatisiert zurücksetzen, benötigt MILTON die Berechtigungen zum Auslesen der Benutzerinformationen bzw. dem Attribut des Benutzerkontostatus oder Kennwortstatus und zum Schreiben eines neuen Kennworts.

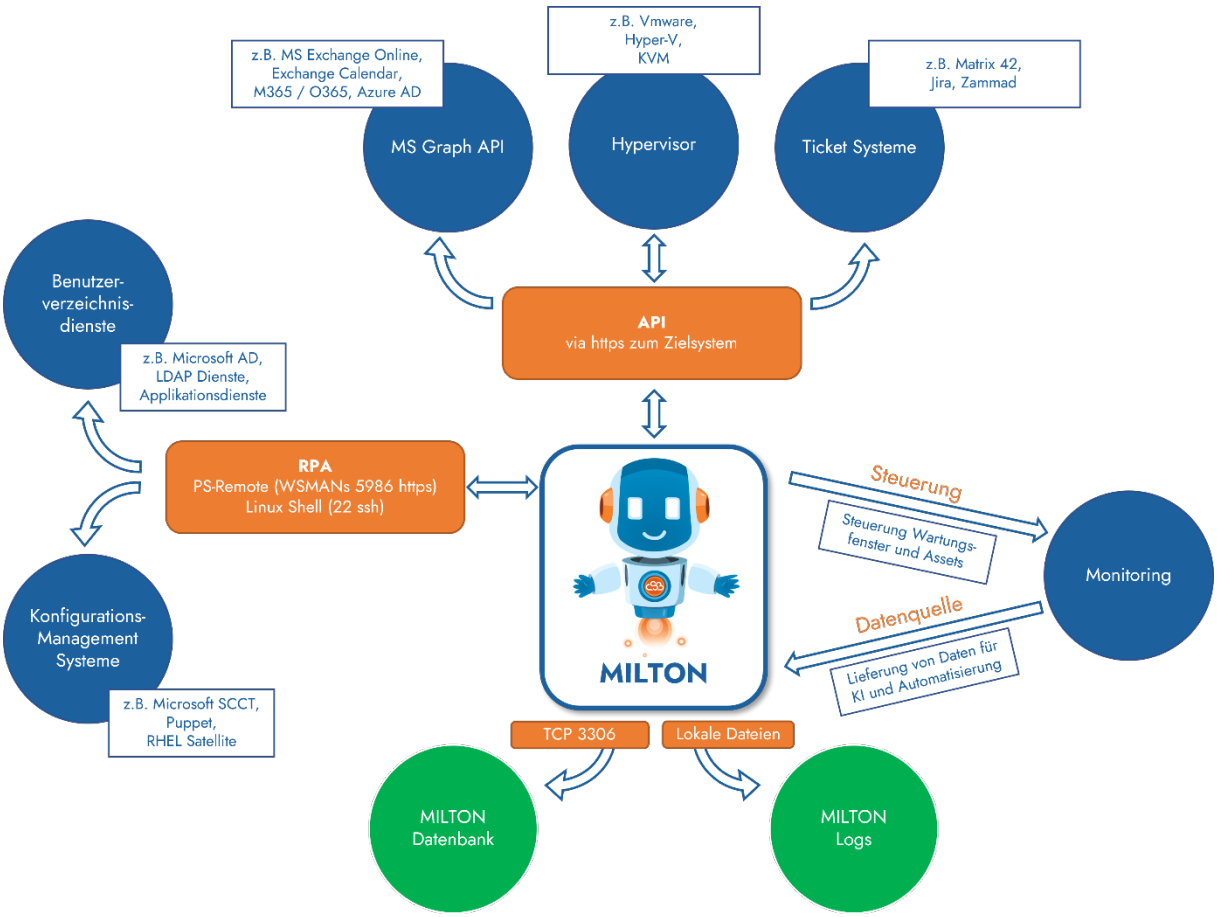

Abbildung 1: Dataflow Diagramm

Designprinzip

Grundsätzlich werden sämtliche Automatisierungstasks so entwickelt, dass diese nur jene Berechtigungen benötigen, die wirklich für die entsprechende Anforderung notwendig sind. Weiter ist für jeden Anwendungsfall ein eigenes Konto vorgesehen, so das eine Anhäufung an Berechtigungen auf einem einzelnen Benutzerkonto nicht notwendig ist.

Kommunikationswege bedienen sich grundsätzlich eines verschlüsselten Transportweges via SSL-Verschlüsslung.

Werden sensible Informationen innerhalb von MILTON verarbeitet oder gespeichert, wie zum Beispiel Benutzerkontoinformationen, so werden diese nur verschlüsselt in der Datenbank abgelegt.

Entwicklungsstandards

MILTON besteht aus vielen eigenentwickelten Bausteinen. Der Quell-Code von MILTON basiert auf einer sich stetig weiterentwickelnden Programmiersprache.

Die Entwicklung von MILTON findet in drei Stages bzw. Branches statt:

- **Entwicklung (DEV)**
  Sämtliche Entwicklungen finden nur in diesem Zweig statt.
- **Test**
  Im Test-Zweig werden sämtliche Neuerungen getestet und erst nach erfolgreichem Test in die Produktion überführt.
- **Produktion**
  Der Produktions-Zweig enthält das fertige Produkt sowie die veröffentlichten Versionen aus dem Test-Zweig.

Bevor eine Änderung aus einem Zweig in einen anderen Zweig überführt wird, wird ein Merge-Request gestellt, dieser stellt alle Änderungen zusammen und führt diese auf. Dieser Merge-Request muss dann im Vier-Augen-Prinzip freigegeben werden.

Schwachstellenmanagement

CLOUDSUPPLIES setzt einen hohen Anspruch an die Stabilität und Sicherheit von MILTON und der Plattform, auf der MILTON betrieben wird, weshalb wir sämtlichen entwickelten Quell-Code mehreren Stufen an Reviewphasen unterziehen. Bei der Zusammenführung von Änderungen wird der Programmcode geprüft:

- Ist die Coding-Governance eingehalten worden?

- Prüfen des Quell-Codes mit einem Schwachstellenscanner auf Sicherheitslücken und Schwachstellen (auch in importierten Bibliotheken).

Die Betriebsplattform respektive das Betriebssystem sowie die Datenbank werden gemäß BSI gehärtet. Die einzelnen Härtungsvorgaben können von Umgebung zu Umgebung variieren, wenn MILTON auf einem bereitgestellten System installiert wird. CLOUDSUPPLIES führt regelmäßig Tests innerhalb seines Labors bezüglich seiner Härtungsvorgaben durch und verwendet daraus resultierende Informationen für Updates der bei ausgerollten Systemen durch. Weiter werden regelmäßig interne Penetration Tests der Anwendung sowie dem Serversystem in der Laborumgebung durchgeführt.

Härtung

Die Härtung von Anwendungen und Betriebssystemen ist ein wichtiger Bestandteil bei der Absicherung vor Angriffen durch Dritte. Neben verschiedenen Konfigurationsanpassungen ist auch die Lokation von MILTON im Netzwerk der Betriebsumgebung ein wichtiger Punkt.

Härtung Betriebssystem

- Einsatz von `fail2ban` zum Schutz der Services SSH und der MILTON Weboberfläche vor Bruteforce Angriffen. `fail2ban` nutzt `iptables` um mehrfache fehlerhafte Logins von derselben Quell-IP-Adresse für eine gewisse Zeit zu sperren.
- Der Login für den Benutzer `root` via `x` oder `ssh` ist standardmäßig deaktiviert und nur via SUDO-Berechtigungen über entsprechende Benutzer nutzbar.
- Grundsätzlich wird eine Minimalinstallation des Betriebssystems verwendet und nur die notwendigen Pakete installiert. Das stellt sicher, dass die verwendeten Programme und Tools geprüft werden können.
- MILTON und sein Betriebssystem sind so konfiguriert, dass alle System- und Anwendungsupdates automatisch und ohne Einfluss vom Nutzer automatisch installiert werden.
- Das Sticky-Bit ist auf Linux Systemen eine Berechtigungseinstellung, die auf Dateien oder Verzeichnissen gesetzt wird. Häufig wird es gesetzt, um das Löschen dieser Dateien oder der Verzeichnisse zu unterbinden. Ist das Sticky-Bit auf Dateien gesetzt, die dem root-Benutzer gehören, so ist es unter Umständen möglich die aktuellen nicht privilegierten Benutzerrechte zum root-Kontext zu eskalieren. Deshalb prüfen wir regelmäßig die vorhandenen gesetzten Dateien mit Stickybit und die vergebenen SUDO-Berechtigungen.
- Damit eine Erkennung des Betriebssystems nicht ohne weiteres möglich ist, werden TCP-Timestamps unterdrückt und ungenutzte TCP Protokolle sind deaktiviert.

- Die Passwortkomplexität für sämtliche Benutzerkonten auf Betriebssystem- und Anwendungsebene ist auf folgende Parameter festgelegt.
    - Das Passwort muss mindestens 25 Zeichen lang sein
    - Große und kleine Buchstaben müssen enthalten sein
    - Es muss mindestens eine Zahl enthalten sein
    - Sonderzeichen müssen ebenfalls enthalten sein
    - Kennwörter laufen nicht ab
    - Passworthistorie ist für die letzten drei Passwörter aktiviert
    - Nach drei fehlerhaften Passworteingaben wird der Benutzer für 20 Minuten gesperrt

Härtung Anwendung

- Die Anwendung MILTON läuft im eigenen Userkontext und hat auf Systemebene nur die Berechtigungen, die für den Betrieb der Anwendung und zum Ausführen der RPA benötigt werden.
- Sämtliche Eingaben in die Weboberfläche sind gegen CSRF[1], Code Injections[2] und ungewollte Code Executions[3] gehärtet. Hier wird ein Sanitizing[4] (Bereinigung von Schadhaften Eingabeelementen) für jede Eingabe durchgeführt.
- Der Login ist zusätzlich gegen SQL Injections[5] gehärtet. Somit ist ein Auth-Bypass via SQL Injections nicht möglich.
- Die Ausführung von eigenen Scripten ist via dem MILTON Frontend nicht möglich.

[1] **Cross-Site-Request-Forgery** (CSRF) ist ein Angriff auf ein Computersystem, bei dem eine Anfrage auf eine Webanwendung durchgeführt wird. Diese Angriffe bedienen sich einem bereits angemeldeten Benutzer und ist in der Regel nicht ohne einen angemeldeten Nutzer möglich.

[2] Via **Code Injection** können in Anwendungen (schadhafter) Programmcode eingeführt werden. Diese werden dann in den Subroutinen der Anwendung ausgeführt und können zum Beispiel **Reverse-Shells** im Benutzerkontext der Anwendung öffnen.
Reverse-Shells ermöglichen eine Verbindungsaufnahme zu einem System, dass unter der Kontrolle des Angreifers steht und dient häufig der Command and Control Steuerung.

[3] Bei **Remote Code Executions** (RCE) wird (schadhafter) Code in die Anwendung geschleust und ausgeführt.

[4] Beim **Sanitizing** werden Eingaben vor Verwendung im Programm von Schadhaften Eingabeelementen bereinigt. So dass Remote Code Executions nicht mehr möglich sind.

[5] Wird in einer Anwendung SQL verwendet und die SQL Queries sind nicht entsprechend vorbereitet, sind **SQL-Injections** möglich, hierbei wird das SQL Query in der Anwendung manipuliert. Dabei ist von Login-Bypass bis hin zum Datenextrakt der gesamten Datenbank alles möglich.