Bundesamt
für Sicherheit in der
Informationstechnik

Deutschland
Digital•Sicher•BSI•

Konfigurationsempfehlungen zur Härtung von Windows 10 mit Bordmitteln

Version: 1.0

Bundesamt für Sicherheit in der Informationstechnik
Postfach 20 03 63
53133 Bonn
Telefon: +49 22899 9582-0
E-Mail: bsi@bsi.bund.de
Internet: https://www.bsi.bund.de

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Zusammenfassung

Dieses Dokument stellt das Ergebnis von Arbeitspaket 11 des Projekts „SiSyPHuS Win10: Studie zu Systemaufbau, Protokollierung, Härtung und Sicherheitsfunktionen in Windows 10“ dar. Das Projekt wird durch die Firma ERNW Enno Rey Netzwerke GmbH im Auftrag des Bundesamts für Sicherheit in der Informationstechnik (BSI) durchgeführt.

Ziel dieses Arbeitspakets ist die Erstellung eines umfassenden Härtungskonzeptes für die Konfiguration von Komponenten von Windows 10. Wie vom Bundesamt für Sicherheit in der Informationstechnik gefordert, ist Windows 10 LTSC 2019, 64 Bit in deutscher Sprache im Fokus dieses Dokuments.

2 Allgemeines

Aufbauend auf den in den Arbeitspaketen AP2 bis AP10 erarbeiteten Ergebnissen ist für Windows 10 eine Konfigurationsempfehlung zur Härtung erstellt worden, welche die Szenarien „normaler Schutzbedarf Domänenmitglied“ (ND), „hoher Schutzbedarf Domänenmitglied“ (HD) sowie „normaler Schutzbedarf Einzelrechner“ (NE) betrachtet.

Die Empfehlung richtet sich an fortgeschrittene Anwender und Administratoren und ist geeignet, die Konfigurationseinstellungen des Betriebssystems direkt umsetzen zu können.

2.1 Geltungsbereich

Das vorliegende Dokument und die darin enthaltenen Konfigurationsempfehlungen sind gültig für das Betriebssystem Microsoft Windows 10 Long Term Servicing Channel (LTSC), Version 2019. Die hierzu äquivalente Semi-Annual Channel (SAC) Version entspricht Windows 10, Version 1809 und ist zu Windows 10 LTSC Version 2019 hinsichtlich des Kernels und der Komponenten, die in beiden Versionen enthalten sind, funktionsgleich. Da LTSC-Versionen auf einen gleichbleibenden Funktionsumfang und Stabilität ausgelegt sind, werden von Microsoft keine Funktionsupgrades nach der Veröffentlichung bereitgestellt und Komponenten, die mit neuen Funktionalitäten versehen werden könnten, wurden entfernt. Zu den wichtigsten fehlenden Komponenten zählt der Edge-Browser, die virtuelle Assistentin Cortana und alle vorinstallierten Universal Windows Apps („Store-Apps“) inkl. dem Microsoft Store.

Die gegebenen Empfehlungen gehen von dem Szenario der Absicherung eines Standard-Büroarbeitsplatzrechners mit den zur Verfügung stehenden Bordmitteln des Betriebssystems aus. Dabei sollen zudem möglichst wenig Funktionseinschränkungen entstehen. Je nach Einsatzort und –Zweck des Systems (z.B. administrative Tätigkeiten, Schutzbedarf der verarbeiteten Informationen, etc.) müssen daher ggf. noch weitere Maßnahmen getroffen oder schärfere Konfigurationseinstellungen umgesetzt werden, wobei die hier gegebenen Empfehlungen als Grundlage dienen können.

2.2 Rahmenbedingungen

Die nachfolgend beschriebenen Konfigurationsempfehlungen basieren auf den im Projekt erarbeiteten Analyseergebnissen, auf Security Best Practices sowie auf langjähriger Expertise von ERNW. Zu allen Empfehlungen erfolgt ein Abgleich mit den in der Security Baseline für Windows 10 1809 (ms_sec_bl_1809, 2020) von Microsoft enthaltenen Einstellungen sowie dem Center for Internet Security (CIS) Benchmark (cis_win10_1809, 2019) für Windows 10 Enterprise (Version 1809). Abweichungen von der Security Baseline oder dem CIS-Benchmark werden im Dokument für die betroffenen Einstellungen erläutert und begründet. Sofern Empfehlungen im Dokument mit dem CIS-Benchmark oder der Security Baseline übereinstimmen, wird auf den entsprechenden Abschnitt im CIS-Benchmark bzw. auf die Security Baseline verwiesen, um das Auffinden der jeweiligen Einstellung in den anderen Publikationen zu erleichtern.

Bei der Erstellung dieses Dokuments folgte die Auswahl der konkreten Härtungsempfehlungen den folgenden Grundprinzipien zur Erhöhung der Systemsicherheit:

- Verhinderung von bekannten und verbreiteten Angriffsszenarien, die nach aktueller Kenntnislage aktiv ausgenutzt werden bzw. mit hoher Wahrscheinlichkeit ausgenutzt werden können.
- Verringerung der Angriffsfläche durch Deaktivierung nicht benötigter (oder veralteter) Funktionen und Komponenten.
- Verbesserung des Datenschutzes, indem Funktionen und Komponenten, die auf Cloud-Diensten basieren, deaktiviert werden.

- Verbesserung des Datenschutzes, indem soweit wie möglich die Übertragung von – nicht für die Funktionalität benötigten – Informationen an den Hersteller unterbunden wird.
- Minimierung von wichtigen Sicherheits- und Datenschutzentscheidungen sowie Auswahlmöglichkeiten durch den Benutzer.
- Erzwingen von bereits sinnvollen Standardeinstellungen, um eine Modifikation durch den Benutzer zu verhindern.

Es ist wichtig zu beachten, dass die empfohlenen Härtungseinstellungen nicht ohne ausführliche Vorabtests in einen produktiven Betrieb übernommen werden sollten, da einige Einstellungen generelle oder auch sehr spezifische (abhängig vom Anwendungsfall) Funktionseinschränkungen mit sich bringen können. Zudem wird darauf hingewiesen, dass nicht alle möglichen Konfigurationsparameter des Betriebssystems durch die Empfehlungen abgedeckt werden, sondern nur diejenigen ausgewählt wurden, welche für die Erhöhung der Systemsicherheit und die Erfüllung der vorgenannten Grundprinzipien relevant sind. Ähnliches gilt für Einstellungsempfehlungen, die nicht aus dem CIS Benchmark übernommen wurden, da ein Teil dieser Einstellungen über den Geltungsbereich dieses Dokuments hinausgeht.

Konfigurationsempfehlungen, die im Zusammenhang mit Protokollierung stehen, finden sich in der „Empfehlung zur Konfiguration der Protokollierung in Windows 10“ des Projekts SiSyPHuS (Arbeitspaket 10).

Gruppenrichtlinien-Objekte zu den Empfehlungen zur Konfiguration der Protokollierung (AP 10) und zur Härtung (AP 11) werden im Rahmen von Arbeitspaket 12 bereitgestellt.

2.3 Definition der Szenarien

Dieser Abschnitt definiert die in der Empfehlung betrachteten Nutzungsszenarien und die möglichen Auswirkungen auf Funktionalitäten des Betriebssystems.

2.3.1 Normaler Schutzbedarf Domänenmitglied (ND)

Die Härtungsmaßnahmen / Konfigurationsempfehlungen sind geeignet, das IT-System von ungezielten Angriffen und Infektionen mit verbreiteter Schadsoftware zu schützen. Zudem entstehen durch die Umsetzung der Härtungsmaßnahmen keine wesentlichen Einschränkungen bei der Nutzung von Funktionalitäten des IT-Systems. Die berücksichtigten Funktionalitäten sind unter „Auswirkungen auf Funktionalitäten im Betriebssystem“ (siehe Kapitel 2.4) aufgeführt.

2.3.2 Hoher Schutzbedarf Domänenmitglied (HD)

Die Härtungsmaßnahmen / Konfigurationsempfehlungen sind geeignet, das IT-System vor gezielten Angriffen und Infektionen mit individuell erstellter Schadsoftware zu schützen. Einschränkungen bei der Nutzung von Funktionalitäten des Betriebssystems sind hinnehmbar. Die berücksichtigten Funktionalitäten sind unter „Auswirkungen auf Funktionalitäten im Betriebssystem“ (siehe Kapitel 2.4) aufgeführt.

Hinweis: Die Implementierung von (HD) erfordert, dass sowohl die Einstellungen von (ND) als auch von (HD) angewendet werden.

2.3.3 Normaler Schutzbedarf Einzelrechner (NE)

Die Härtungsmaßnahmen / Konfigurationsempfehlungen sind geeignet, das IT-System von ungezielten Angriffen und Infektionen mit verbreiteter Schadsoftware zu schützen. Zudem entstehen durch die Umsetzung der Härtungsmaßnahmenkeine wesentlichen Einschränkungen bei der Nutzung von

Funktionalitäten des IT-Systems. Die berücksichtigten Funktionalitäten sind unter „Auswirkungen auf Funktionalitäten im Betriebssystem“ (siehe Kapitel 2.4) aufgeführt.

2.4 Auswirkungen auf Funktionalitäten im Betriebssystem

Zu jeder aufgeführten Konfigurationsempfehlung sind die entsprechenden Auswirkungen auf Funktionalitäten im Betriebssystem und Anwendungen mit Hilfe von Stichwörtern beschrieben. Hierbei sind insbesondere die Auswirkungen auf die folgenden Funktionalitäten betrachtet worden:

- lokales Arbeiten mit Office-Programmen
- Netzwerkfähigkeit des IT-Systems
- Multimediafähigkeit des IT-Systems
- Anzeige von Webseiten und aktiven Inhalten
- Möglichkeiten zur Fernwartung
- Enterprise-Deployment
- Integration des IT-Systems in Active Directory
- Ausführen von Programmen und Apps
- Installation von weiteren Anwendungen auf dem Betriebssystem

Sollten keine Auswirkungen oder Abweichungen vom Standardverhalten festgestellt worden sein, entfällt der jeweilige zur Konfigurationsempfehlung gehörende Unterpunkt.

3 Generelle Maßnahmenempfehlungen

Die folgenden Abschnitte beschreiben generelle Empfehlungen aus dem Bereich Informationssicherheit, die beim sicheren Betrieb eines Windows 10-Systems berücksichtigt werden sollten. Da diese nicht zwingend über technische Konfiguration (mit Bordmitteln) umsetzbar sind oder über den definierten Rahmen dieses Dokuments hinausgehen, werden diese Maßnahmen nur allgemein beschrieben.

3.1 Nutzung sicherer Quellen für Hard- und Software

Sowohl Hardware als auch Software kann manipuliert werden und somit Schadsoftware oder Hintertüren enthalten. Um die Vertrauenswürdigkeit eines IT-Systems zu gewährleisten, sollte grundsätzlich darauf geachtet werden, dass eingesetzte Hard- sowie Software aus bekannten und seriösen Quellen stammt. Konkret bedeutet dies:

- Eingesetzte Hardware sollte nur aus vertrauenswürdigen Quellen, idealerweise über den Hersteller selbst, bezogen werden. Dies gilt nicht nur für das Endgerät selbst, sondern auch für jegliche Hardware, die an dieses angeschlossen wird (wie z. B. externe Eingabegeräte, externe Datenträger, Docking-Stationen etc.).
- Software, die installiert wird, sollte nur aus vertrauenswürdigen Quellen bezogen werden. Wenn möglich, sollte Software über Bereitstellungen des Unternehmens bzw. der Organisation bezogen werden. Ist dies nicht möglich, sollte kommerzielle Software über einen ausgewiesenen Fachhandel (auch online) oder den Hersteller direkt akquiriert werden.
- Wird Open Source Software installiert oder Open Source Code (z. B. Skripte) ausgeführt, sollte der Entwickler identifiziert und die Software oder Skripte aus Quellen bezogen werden, die mit dem Entwickler assoziiert sind, wie z. B. dem zugehörigen GitHub Repository.
- Wird Software oder generell Code aus Online-Quellen heruntergeladen, sollte die Integrität der Daten geprüft werden. Dies kann durch eine Verifikation mit einem vom Hersteller bereitgestellten Hash-Wert erfolgen oder durch die Prüfung einer vorhandenen digitalen Signatur hinsichtlich Integrität der Datei und Authentizität des Herausgebers.
- Software sollte, wann immer möglich, über einen sicheren und verschlüsselten Kanal übertragen werden, bevor diese auf einem System zum Einsatz kommt. Dies ist vor allem zu beachten, wenn Daten über das Internet transferiert werden. Ziel ist es hierbei sog. Man-in-the-Middle-Angriffe zu verhindern (bei denen ein Angreifer Daten während des Transfers modifiziert) und eine Authentizitätsprüfung des Herausgebers zu ermöglichen, z. B. durch den Einsatz von Transport Layer Security (TLS).

3.2 Verwendung getrennter Standardbenutzerkonten und Administratorenkonten

Die typischen Tätigkeiten eines Standardbenutzers (z. B. Aufrufen von Webseiten, Abrufen von E-Mails, Bearbeiten von Dokumenten) bringen ein deutlich höheres Risiko für eine Kompromittierung mit sich, als es die Tätigkeiten eines Administrators tun. Um zu verhindern, dass ein Angreifer nach erfolgreicher Kompromittierung eines Benutzerkontos direkt weitreichende Kontrolle über ein System erlangen kann, sollten mindestens zwei unterschiedliche Konten angelegt werden. Ein Konto, welches über keine besonderen Berechtigungen verfügt und somit aus Angreifersicht wenig lohnend ist, sowie ein administratives Konto, dem die Berechtigungen zugewiesen werden, die für Administrationstätigkeiten notwendig sind. Das Standardbenutzerkonto wird daraufhin für die initiale Anmeldung und unprivilegierte Tätigkeiten genutzt, während das administrative Benutzerkonto nur zum Einsatz kommt, wenn höhere Berechtigungen genutzt werden müssen. Dies senkt die Wahrscheinlichkeit einer erfolgreichen Kompromittierung von Administratorenkonten und stellt eine zusätzliche Hürde für einen Angreifer dar.

Diese Empfehlung gilt dabei sowohl für lokale Benutzerkonten als auch für Domänenkonten in einer verwalteten Domänenumgebung. Ergänzend sollten Benutzerkonten, die nicht genutzt werden oder die nicht für den Betrieb des Systems notwendig sind, deaktiviert bzw. komplett entfernt werden.

Die Verwendung getrennter Benutzerkonten mit unterschiedlichen Berechtigungen wird unter Windows durch die Funktionalität der sog. Benutzerkontensteuerung begünstigt, da bei Aktionen, die erhöhte Privilegien erfordern, automatisch nach Zugangsdaten eines administrativen Benutzerkontos gefragt wird, wenn der angemeldete Benutzer diese nicht bereits besitzt.

3.3 Verwendung angemessener Kennwortrichtlinien

Einfache Kennwörter können von einem Angreifer mittels einer Passwort-Attacke (z. B. durch Brechen eines Hash-Wertes, um das zugehörige Klartext-Kennwort zu erhalten) berechnet oder manchmal sogar erraten werden (z. B. durch sog. Password Spraying, bei dem dasselbe Kennwort für alle Benutzer einer Umgebung zur erfolgreichen Authentifizierung getestet wird). Eine angemessen sichere Kennwortrichtlinie erfordert, bei der Wahl des Kennworts Anforderungen zu erfüllen, die eine Berechnung (oder das Erraten) des Kennworts deutlich erschweren. Generell sollte bei der Wahl des Kennworts folgendes beachtet werden: die Länge ist ausschlaggebender für die Passwort-Qualität als die Komplexität und durch diese zusätzliche Länge muss das Kennwort seltener geändert werden.

Je länger ein Kennwort ist, umso mehr Zeit benötigt ein Angreifer, das Kennwort z. B. aus einem Hash-Wert zu berechnen und die Wahrscheinlichkeit sinkt, dass ein Kennwort in wenigen Versuchen erfolgreich erraten werden kann. Das Erhöhen der Passwort-Komplexität bei geringer Länge würde hingegen die Kennwort-Qualität nur geringfügig erhöhen (siehe auch (ms_pass, 2020)). Bei ausreichender Passwort-Stärke muss das Kennwort seltener bzw. gar nicht geändert werden (siehe auch ORP.4.A23 in (bsi_it_gs_orp4, 2020)). Sollte jedoch eine Kompromittierung eines Kennworts vermutet werden, sollte das Passwort unabhängig von der noch ausstehenden Gültigkeitszeit unverzüglich geändert werden. Darüber hinaus sollten keine Kennwörter verwendet werden, die in bekannten und gängigen Passwortlisten enthalten sind.

Die in diesem Dokument empfohlene Konfiguration der Kennwortrichtlinie bezieht sich primär auf die lokale Konfiguration eines Windows-Systems und betrifft somit lokale Benutzerkonten. In einer verwalteten Domänenumgebung sollte die Kennwortrichtlinie auf Domänenebene durch Gruppenrichtlinien verbindlich für alle Systeme gelten und somit auch für alle Domänenbenutzer. Zusätzlich sollte es in einer Domänenumgebung eine granulare Unterscheidung zwischen Kontentypen geben, um für administrative Benutzerkonten und technische Dienstkonten eine striktere Kennwortrichtlinie vorschreiben zu können (z. B. durch sog. Fine-Grained Password Policies).

3.4 Sicheres Speichern von Kennwörtern

In Klartext auf dem System gespeicherte Passwörter (Textdateien, Skripte, Browser, ...) können von einem Angreifer, der über die Berechtigungen des angemeldeten Benutzers verfügt, auf einfache Weise ausgelesen werden. In einem Passwort-Manager können Passwörter verschlüsselt und vor unautorisiertem Zugriff geschützt abgelegt werden. Es wird daher empfohlen einen Passwort-Manager auf dem System einzusetzen. Nach der Installation erstellt der Benutzer eine Passwortdatenbank und wählt ein angemessen sicheres Masterpasswort (in Übereinstimmung mit einer vorher definierten Richtlinie). In der Passwortdatenbank sollten dann alle Passwörter hinterlegt werden, die der Benutzer zum Zugriff auf Webseiten, Freigaben, Applikationen etc. benötigt. Diese Passwörter sollen möglichst vom Passwort-Manager generiert werden und somit einzigartig sein. Durch die Eingabe des Masterpassworts kann auf die Passwörter in der verschlüsselten Datenbankdatei zugegriffen werden.

3.5 Keine Wiederverwendung von Kennwörtern

Wird dasselbe Kennwort für unterschiedliche Zwecke (z. B. Anwendungen, Computersysteme, Webseiten) verwendet, kann dieses auch an mehreren Stellen durch einen Angreifer kompromittiert werden. Eine

angreifbare Stelle reicht dabei aus, um anschließend Zugriff auf weitere, unabhängige Dienste zu erlangen, bei denen dasselbe Kennwort genutzt wird. Ein Passwort-Manager gewährleistet in diesem Fall nicht nur die sichere Verwahrung einer Vielzahl von Kennwörtern (siehe Kapitel 3.4), sondern kann auch genutzt werden, um zufällige und einzigartige Kennwörter pro Anwendung/Dienst zu generieren und abzuspeichern. Für jeden Dienst, der ein Passwort erfordert, sollte der Benutzer somit ein eigenes Kennwort erstellen und im Passwort-Manager sicher abspeichern. Dies gilt auch für unterschiedliche Benutzerkonten (z. B. Domänenbenutzerkonten), die von derselben Person verwendet werden.

3.6 Regelmäßige Aktualisierung der Firmware, des Betriebssystems und installierter Applikationen

Als essenzielles Bindeglied zwischen der Hardware des Geräts und dem Betriebssystem, wird der Firmware besondere Bedeutung beigemessen. Um die Sicherheit der Hardwareplattform, auf der später das Betriebssystem und unterschiedliche Anwendungen installiert werden, zu gewährleisten, sollte die Firmware des Geräts immer auf einem aktuellen Stand gehalten werden. Dafür ist es notwendig in regelmäßigen Abständen zu prüfen, ob ein neues Update vom Hersteller veröffentlich wurde. Zu dem Updateprozess sollte ebenfalls eine Überprüfung des hinterlegten kryptographischen Schlüsselmaterials bzw. der Zertifikate gehören, welche festgelegen, welche Firmware(-Updates) und Betriebssystem-Loader auf dem Gerät ausgeführt werden können. Hierbei geht es vor allem um die Prüfung der Informationen, die in der sog. Allowed Signature Database (db) und der Forbidden Signature Database (dbx) gespeichert sind, und die für den sicheren Bootvorgang (Secure Boot) genutzt werden.

Für das Windows-Betriebssystem und eigene Anwendungen stellt Microsoft in regelmäßigen Zyklen Updates bereit. Updates zu schwerwiegenden Sicherheitslücken werden in der Regel außerhalb des geregelten Zyklus veröffentlicht. Daher sollten neue Updates, Patches und Hotfixes für Betriebssysteme und Anwendungen von Microsoft schnellstmöglich nach Veröffentlichung über die Windows-Update Funktionalität des Betriebssystems installiert werden (konkrete Empfehlungen für die Konfiguration des Windows Update-Mechanismus werden in diesem Dokument gegeben). In verwalteten Domänen-Umgebungen kann zudem ein sogenannter Windows Server Update Service (WSUS) die Updates zentralisiert bereitstellen, um kontrollieren zu können, welche Systeme welche Updates erhalten.

Dritthersteller-Software, die zusätzlich auf dem System installiert wird, bietet einem Angreifer ebenfalls eine Angriffsfläche. Wie das Betriebssystem selbst, können auch installierte Anwendungen offene Sicherheitslücken aufweisen, die zur vollständigen Kompromittierung des gesamten Systems führen können. Standardisierte Anwendungen in einer verwalteten Umgebung (Browser, E-Mail-Programm, Dokumentenverarbeitung etc.) sollten durch eine zentralisierte Patch-Management-Lösung so bald wie möglich nach Veröffentlichung aktualisiert werden. Auf Einzelrechnern sollte in regelmäßigen Abständen manuell geprüft werden, ob neue Updates zur Verfügung stehen. Hierbei gibt es häufig keine definierten Zyklen, in denen Aktualisierungen veröffentlicht werden, und Update-Informationen sind in der Regel nicht standardisiert, was diesen Vorgang allgemein erschwert.

Grundsätzlich sollten für alle Updates die Empfehlungen aus Kapitel 3.1 berücksichtig werden (vor allem bei einer manuellen Aktualisierung), um die Integrität und Authentizität der Updatedateien bzw. der Quelle, aus der diese bezogen wurden, zu gewährleisten.

3.7 Installation ausschließlich notwendiger Applikationen und Betriebssystem-Komponenten

Jede zusätzliche Software-Komponente bietet eine zusätzliche Angriffsfläche und muss in regelmäßigen Zyklen (teils manuell) aktualisiert werden (siehe Kapitel 3.6). Daher sollten nur Anwendungen und Betriebssystemkomponenten installiert werden, die für die tatsächliche Tätigkeit des Benutzers bzw. für den Betrieb des Systems benötigt werden. Initial sollten vorinstallierte, aber nicht benötigte Anwendungen und Windows-Komponenten deinstalliert werden oder wenn dies nicht möglich ist, zumindest deaktiviert

werden (wie z. B. bei Windows-Diensten). Eine allgemeine Empfehlung für das Deaktivieren von Betriebssystem-Komponenten wird im Rahmen dieses Dokuments gegeben. Welche Anwendungen oder Funktionalitäten allgemein benötigt werden und somit installiert und aktiv sein müssen, muss jedoch individuell nach Einsatzzweck entschieden werden. Zusätzlich sollten in regelmäßigen Abständen die momentan installierten Anwendungen evaluiert werden und bei fehlender Notwendigkeit entfernt werden.

3.8 Verwendung von Festplattenverschlüsselung

Um die Vertraulichkeit von Daten, die im Dateisystem des Betriebssystems abgespeichert sind, zu sichern und nicht autorisierten Modifikationen von Daten entgegenzuwirken, sollte eine Software-Lösung zur Festplattenverschlüsselung eingesetzt werden. Dies ist vor allem dann relevant, wenn die physische Sicherheit nicht mehr gewährleistet werden kann, z. B. im Falle eines Diebstahls des Geräts. Dabei sollten nicht nur die Betriebssystem- und Daten-Partitionen verschlüsselt sein, sondern idealerweise die gesamte Festplatte. Zudem sollte das Verschlüsselungsprogramm, vor dem Start des Betriebssystems, eine Benutzer-Authentisierung durchführen (die sog. Prä-Boot-Authentisierung), um zu verhindern, dass Teile des kryptographischen Materials zur Entschlüsselung der Festplatte in den Arbeitsspeicher geladen und dort potenziell ausgelesen werden können. Der eingesetzte Verschlüsselungsalgorithmus sollte so gewählt werden, dass es einem Angreifer ohne Kenntnis des Verschlüsselungsschlüssels nicht trivial möglich ist die Festplatte zu entschlüsseln. Der eingesetzte Verschlüsselungsschlüssel sollte zufällig generiert sein und sicher aufbewahrt werden. Vor Aktivierung der Festplattenverschlüsselung sollte zudem ein Konzept zur Sicherung bzw. Wiederherstellung entwickelt werden, um den vollständigen Verlust der lokalen Daten zu verhindern, falls der Entschlüsselungsschlüssel verloren wird.

4 Konfigurationsempfehlungen

Auf Grund des Gesamtumfangs der Konfigurationsempfehlungen wurden alle Einstellungsempfehlungen, die auf Gruppenrichtlinien basieren, in ein separates Tabellen-Dokument ausgelagert. Hierdurch bleibt die Übersichtlichkeit des vorliegenden Dokuments gewahrt und die empfohlenen Einstellungsparameter können besser in Tabellenform dargestellt werden. Die vollständige Liste aller Härtungsempfehlungen auf Basis von Gruppenrichtlinien findet sich im Dokument „Konfigurationsempfehlungen zur Härtung von Windows 10 mit Bordmitteln: Gruppenrichtlinien-Einstellungen“. Darüber hinaus gibt es weitere Einstellungsempfehlungen, die im nachfolgenden Kapitel betrachtet werden.

5 Zusätzliche Konfigurationsempfehlungen

Die folgenden Abschnitte enthalten Konfigurationsempfehlungen, die nicht ausschließlich über Gruppenrichtlinien abgebildet werden können und zusätzlicher Erklärung bedürfen.

5.1 (HD) Windows Defender-Anwendungssteuerung

Windows 10 implementiert eine Funktionalität, die als konfigurierbare Code-Integrität bezeichnet wird (im Folgenden als Windows Defender-Anwendungssteuerung - Windows Defender Application Control – WDAC bezeichnet). WDAC soll das Ausführen von nicht-vertrauenswürdigem Code verhindern. „Nicht-vertrauenswürdiger Code“ sind Programme, deren Integrität und/oder Authentizität nicht verifiziert werden können (weil diese beispielsweise verändert oder aus nicht bekannten Quellen heruntergeladen wurden). WDAC verwendet hierzu benutzerdefinierte Kriterien, um ausführbare Dateien zu überprüfen, d. h. nur bestimmte zuzulassen. Diese Kriterien können Datei-Attribute beinhalten (z. B. Hash-Werte, Dateinamen). Die Kriterien werden in einer sogenannten WDAC-Richtliniendatei definiert. Diese Datei befindet sich zunächst im XML-Format (Extensible Markup Language) und wird anschließend in eine Binärdatei konvertiert, welche schlussendlich von WDAC gelesen und interpretiert wird.

WDAC wird durch die Gruppenrichtlinieneinstellung `Windows Defender-Anwendungssteuerung bereitstellen` unter `Computerkonfiguration\Administrative Vorlagen\System\Device Guard` aktiviert. Empfehlungen zum Konfigurieren von WDAC-Richtlinien finden Sie in (ERNW_WP7), Abs. 3.1.2, Abs. 3.1.3 und Abs. 3.1.4.

Die Folgenden Abschnitte beschreiben die zu berücksichtigen Elemente zum sicheren Einsatz von WDAC.

5.1.1 Signieren der WDAC-Richtlinie

Stellen Sie sicher, dass jede WDAC-Richtlinie digital signiert ist, um nicht autorisierte Richtlinienänderungen zu verhindern.

In Organisationen erfordert dies eine ordnungsgemäß verwaltetes Signaturzertifikat, welches idealerweise durch eine „Public Key Infrastructure“ (PKI) bereitgestellt wird. Die WDAC-Richtlinie selbst sollte auf einem dedizierten System signiert werden, um den Schutz des Signaturzertifikats und des Signaturprozess sicherzustellen. Des Weiteren sollte die signierte WDAC-Richtlinie nur über einen sicheren Kanal an die entsprechenden Zielsysteme verteilt werden.

Im Folgenden werden die Konfigurationsschnittstellen zum Signieren einer WDAC-Richtliniendatei aufgelistet.

5.1.1.1 Konfigurationsschnittstelle: PowerShell

Set-AuthenticodeSignature `Set-AuthenticodeSignature` ist ein PowerShell-Befehl mit dessen Hilfe Binärdateien digital signiert werden können.

```
Set-AuthenticodeSignature -FilePath $policy -Certificate $cert
```

- `FilePath` gibt den Pfad der Richtliniendatei (im Binärformat) an
- `Certificate` gibt das Signaturzertifikatobjekt an. Dieses Objekt kann z.B. mit Hilfe des PowerShell cmdlets `Get-PfxCertificate` erstellt werden (ms_pfx, 2020)

Detaillierte Informationen zum `Set-AuthenticodeSignature` PowerShell-Befehl finden Sie unter (ms_ps, 2020).

5.1.1.2 Konfigurationsschnittstelle: Windows-Anwendungen

SignTool.exe Bei der `SignTool.exe` Anwendung handelt es sich um ein Befehlszeilentool, mit dessen Hilfe unter anderem Binärdateien digital signiert werden können. `SignTool.exe` wird mit dem Software Development Kit (SDK) ausgeliefert.

`SignTool.exe sign /v /fd sha256 /f $cert /p $pass $policy`

- `sign` gibt an, dass eine Binärdatei digital signiert werden soll
- `/f` gibt den Dateipfad des Signaturzertifikats an (in PFX-Format)
- `$policy` gibt den Pfad der Richtliniendatei (im Binärformat) an
- `/fd` gibt den Dateihashwertalgorithmus zum Erstellen von Dateisignaturen an
- `/v` gibt an, dass ausführliche Ausgabe und Warnmeldungen angezeigt werden
- `/p` gibt das Kennwort zum Öffnen des Signaturzertifikats an

Detaillierte Informationen zum `SignTool.exe` finden Sie unter (ms_sign, 2020).

5.1.2 Anforderung für den sicheren Einsatz von WDAC

Unified Extensible Firmware Interface (UEFI)

Stellen sie sicher, dass die UEFI-Firmware aktiviert ist.

Die UEFI-Firmware bietet einen sicheren Speicher für relevante WDAC-Konfigurationsparameter und Daten z. B. zum Schutz der Integrität einer WDAC-Richtliniendatei. Das Speichern relevanter WDAC-Konfigurationsparameter und Daten in der Firmware bietet einen stärkeren Schutz vor Manipulationen durch unautorisierte Windows-Benutzer.

5.2 (HD, ND, NE) Virtualisierungsbasierte Sicherheit

Virtualisierungsbasierte Sicherheit (VBS) ist eine auf dem Microsoft Hypervisor-basierte Funktionalität, welche die traditionelle Windows Architektur in zwei streng voneinander isolierte Umgebungen aufteilt (im Folgenden als der sichere und normale Bereich bezeichnet). Die Aufteilung der Windows Architektur in einen sicheren und normalen Bereich ermöglicht nun erstmalig, dass Sicherheitskritische Funktionalitäten streng vom normalen Bereich getrennt werden können, was einen besseren Schutz vor unautorisierten Zugriffen ermöglicht. Dies umfasst z. B. das sichere Speichern und Verwalten von Windows Anmeldedaten oder das Überprüfen von auszuführenden Windows-Anwendungen. Es ist wichtig zu betonen, dass die Verwendung von VSM zum Schutz bestimmter Funktionalitäten ihre forensische Analyse (z. B. Analyse des Speicherinhalts oder Debugging) zu einer herausfordernden bzw. nicht durchführbaren Aufgabe macht.

VBS ist konzeptionell in zwei Komponenten unterteilt. Die VBS Kernkomponente, welche durch den Microsoft Hypervisor bereitgestellt wird und die sogenannten VBS Anwendungen, wie z.B. „Virtualization Based Protection of Code Integrity“ und „Credential Guard“.

VBS wird durch die Gruppenrichtlinieneinstellung `Virtualisierungsbasierte Sicherheit aktivieren` unter `Computerkonfiguration\Administrative Vorlagen\System\Device Guard` aktiviert.

Die folgenden Abschnitte beschrieben die zu berücksichtigen Konfigurationsempfehlung zum sicheren Einsatz der VBS Kernkomponente und der VBS Anwendungen.

5.2.1 VBS Kernkomponente

Die folgenden Einstellungen der VBS Kernkomponente erhöhen die Sicherheit der Systeminitialisierung von Windows 10 und schützen vor Plattformangriffen (z. B. Angriffe, die Funktionen auf Hardwareebene missbrauchen).

5.2.1.1 Sicherer Start und DMA-Schutz

Stellen Sie sicher, dass die folgende Gruppenrichtlinieneinstellung `Virtualisierungsbasierte Sicherheit aktivieren` unter `Computerkonfiguration\Administrative Vorlagen\System\Device Guard` auf folgenden Wert gesetzt ist: `Aktiviert`, und

- `Platform-Sicherheitsstufe auswählen` auf folgenden Wert gesetzt ist: `Sicherer Start und DMA-Schutz`.

Standardwert: Nicht konfiguriert

Sicherer Start (Secure Boot) ist ein Standard zum sicheren Booten von Computern, so dass der Computer nur Software lädt, welche der Computerhersteller und/oder der Betriebssystemhersteller als vertrauenswürdig erachtet. `Sicherer Start und DMA-Schutz` sollten auf Plattformen aktiviert sein, die den direkten Speicherzugriff (DMA) unterstützen. Dies sind Plattformen mit Geräten mit "input-output memory management units" (IOMMUs). Diese Richtlinienoption aktiviert Mechanismen zur Abwehr von DMA-basierten Angriffen und erfordert Hardware-Unterstützung.

5.2.1.2 Sichere Startkonfiguration

Stellen Sie sicher, dass die folgende Gruppenrichtlinieneinstellung `Virtualisierungsbasierte Sicherheit aktivieren` unter `Computerkonfiguration\Administrative Vorlagen\System\Device Guard` auf folgenden Wert gesetzt ist: `Aktiviert`, und

- `Sichere Startkonfiguration` auf folgenden Wert gesetzt ist: `Aktiviert`.

Standardwert: Nicht konfiguriert

Diese Einstellung stellt sicher, dass eine Plattform nur mit vertrauenswürdigem Code gestartet wird. Sie aktiviert die dynamic root of trust (DRTM)-Funktionalität, welche die Plattform vor Firmware-basierten Angriffen schützt.

5.2.2 VBS Anwendungen

Die folgenden Einstellungen der VBS Anwendungen erhöhen die Sicherheit des Windows-Benutzeranmeldeprozesses und des Integritätsüberprüfungsmechanismus.

5.2.2.1 Credential Guard

Stellen Sie sicher, dass die folgende Gruppenrichtlinieneinstellung `Virtualisierungsbasierte Sicherheit aktivieren` unter `Computerkonfiguration\Administrative Vorlagen\System\Device Guard` auf folgenden Wert gesetzt ist: `Aktiviert`, und

- `Credential Guard-Konfiguration` auf folgenden Wert gesetzt ist: `Mit UEFI-Sperre aktiviert`.

Standardwert: Nicht konfiguriert

Credential Guard verwendet VBS zum sicheren Speichern und Verwalten von Windows Anmeldedaten. Der Wert `Mit UEFI-Sperre aktiviert` aktiviert Credential Guard und weist Windows an, dass relevante Credential Guard-Konfigurationsparameter im sicheren Speicher des UEFIs abgelegt werden sollen.

5.2.2.2 Virtualisierungsbasierter Schutz der Codeintegrität

Stellen Sie sicher, dass die folgende Gruppenrichtlinieneinstellung `Virtualisierungsbasierte Sicherheit aktivieren` unter `Computerkonfiguration\Administrative Vorlagen\System\Device Guard` auf folgenden Wert gesetzt ist: `Aktiviert`, und

- `Virtualisierungsbasierter Schutz der Codeintegrität` auf folgenden Wert gesetzt ist: `Mit UEFI-Sperre aktiviert`.

Die Option `UEFI-Speicherattributtabelle erforderlich` ist nicht gesetzt.

Standardwert: Nicht konfiguriert

Virtualisierungsbasierter Schutz der Codeintegrität verwendet VBS zum sicheren Überprüfen von auszuführenden Windows Anwendungen. Der Wert `Mit UEFI-Sperre aktiviert` aktiviert Virtualisierungsbasierter Schutz der Codeintegrität und weist Windows an, dass relevante Virtualisierungsbasierter Schutz der Codeintegrität-Konfigurationsparameter im sicheren Speicher des UEFIs abgelegt werden sollen.

5.2.3 Anforderungen für den sicheren Einsatz von VBS

Unified Extensible Firmware Interface (UEFI)

Stellen sie sicher, dass die UEFI-Firmware aktiviert ist.

Die UEFI-Firmware bietet einen sicheren Speicher für relevante VSM-Konfigurationsparameter. Das Speichern relevanter VSM-Konfigurationsparameter in der Firmware bietet einen stärken Schütz von Manipulationen von unautorisierten Windows-Benutzer.

Trusted Platform Module (TPM)

Stellen Sie sicher, dass das TPM vorhanden und aktiviert ist.

Das TPM bietet einen sicheren Speicher für sicherheitskritische Daten, wie z. B. kryptographisches Material.

5.3 (HD, ND, NE) Trusted Platform Module

Das Trusted Platform Module (TPM) ist ein elektronischer Chip, der grundlegende Sicherheitseigenschaften auf Hardwareebene bereitstellt. Dieser ist hardwareseitig vor Manipulationen geschützt und kann als sicherer Speicher von Daten oder zum Schutz der Integrität eines Systems eingesetzt werden. Des Weiteren bietet der Chip einen Mechanismus, um kryptographisches Material sicher zu erzeugen und verwalten.

Das TPM ist ein Baustein, der übermittelte Befehle (sogenannte TPM-Befehle) annimmt und bearbeitet. Einige TPM-Befehle und -Funktionen sind so geschützt, dass sie nur von autorisierten Benutzern ausgeführt werden können. Die Benutzerautorisierung erfolgt durch Angabe eines Autorisierungswerts. Ein solcher Wert ist der *owner authorization value* (`OwnerAuth`), der das zentrale Authentifizierungsmerkmal für die Verwaltung des TPMs darstellt.

Die folgenden Abschnitte beschrieben die zu berücksichtigen Konfigurationsempfehlung zum sicheren Einsatz des TPMs.

5.3.1 Befehlsblockierung

Stellen Sie sicher, dass die folgende Gruppenrichtlinieneinstellung `Standardliste der blockierten TPM-Befehle ignorieren` unter `Computerkonfiguration\Administrative Vorlagen\System\Trusted Platform Module-Dienste` auf folgenden Wert gesetzt ist: `Deaktiviert`.

Standardwert: Nicht konfiguriert

Die Richtlinie konfiguriert die Zugriffssteuerung von Windows, um die Ausführung von TPM-Befehlen beschränken zu können. Der Wert `Deaktiviert` konfiguriert die Standardbefehlsblockierung und weist Windows an, dass nur bestimmte TPM-Befehle erlaubt sind.

5.3.2 Standardbenutzer-Sperrdauer und Standardbenutzer-Sperrschwelle (einzeln/gesamt)

Stellen Sie sicher, dass die folgende Gruppenrichtlinieneinstellung:

`Standardbenutzer-Sperrdauer` unter `Computerkonfiguration\Administrative Vorlagen\System\Trusted Platform Module-Dienste` auf folgenden Wert gesetzt ist: `Aktiviert`, und

- `Dauer zum Zählen von TPM-Autorisierungsfehlern (Minuten)` auf folgenden Wert gesetzt ist: 30;

`Standardbenutzer-Sperrschwelle(einzeln)` und `Standardbenutzer-Sperrschwelle(gesamt)` unter `Computerkonfiguration\Administrative Vorlagen\System\Trusted Platform Module-Dienste` auf folgenden Wert gesetzt ist: `Aktiviert` und

- `Maximale Anzahl an Autorisierungsfehlern pro Zeitraum` auf folgenden Wert gesetzt ist: 5.

Standardwerte: Nicht konfiguriert

Das TPM kann Autorisierungsversuche einschränken, um Brute-Force-Angriffe zu verhindern. Das TPM zählt die Anzahl der TPM-Autorisierungsfehler innerhalb eines Zeitraums (`Standardbenutzer-Sperrdauer`, `Dauer zum Zählen von TPM-Autorisierungsfehlern`) und sperrt sich selbst, wenn ein bestimmter Schwellenwert (`Standardbenutzer-Sperrschwelle`) erreicht wird. Wenn eine maximale Anzahl von fehlerhaften Autorisierungsversuchen überschreiten wurde (`Maximale Anzahl an Autorisierungsfehlern pro Zeitraum`), wechselt das TPM in den Sperrmodus. Im Sperrmodus können keine autorisierungspflichtigen Befehle mehr vom TPM verarbeitet werden.

Die obigen Werte haben sich als betriebsfähig erwiesen und bieten gleichzeitig einen angemessenen Sicherheitsvorteil.

5.3.3 Anforderungen für den sicheren Einsatz des TPMs

Windows-konforme TPM-Initialisierung

Führen Sie den PowerShell-Befehl `Enable-TpmAutoProvisioning` aus.

Das TPM sollte von Windows automatisch initialisiert werden, um automatisch eine sichere `OwnerAuth` zu generieren und ordnungsgemäß zu funktionieren. Die automatische TPM-Initialisierung durch Windows wird als `auto provisioning` bezeichnet. Der PowerShell-Befehl `Enable-TpmAutoProvisioning` aktiviert die automatische TPM-Initialisierung.

5.4 (HD, ND, NE) Windows-Telemetrie

Windows Telemetrie ist eine Komponente in Windows 10, die für die automatische Erhebung und Übertragung von Daten an eine von Microsoft betriebene Backend-Infrastruktur verantwortlich ist. Da Art und Umfang der erhobenen Daten, die Sicherheit ihrer Übertragung sowie ihre Speicherung und Verarbeitung im Telemetrie-Backend nicht vollständig bekannt sind, sollte die Windows Telemetrie unterbunden werden.

5.4.1 Deaktivierung von Telemetrie-Dienst und ETW-Sessions

Stellen Sie sicher, dass der Telemetrie-Dienst und die zugehörigen ETW-Sessions deaktiviert sind.

Sowohl für Unternehmensumgebungen als auch Endnutzer bietet unter Betrachtung typischer Betriebsanforderungen die Deaktivierung von Telemetrie-Dienst (`Benutzererfahrung und Telemetrie im verbundenen Modus`) und zugehörigen ETW-Sessions (Speicherbereiche, in die protokollierte Ereignisse geschrieben werden) das beste Verhältnis aus wirksamer Telemetrie-Unterbindung und operationellen Auswirkungen.

Die relevanten ETW-Provider (Entitäten, die Ereignisse protokollieren) schreiben ihre Daten in die ETW-Sessions `Autologger-DiagTrack-Listener` und `DiagTrack-Listener`. Diese Sessions sind die Quelle der Telemetrie-Daten. Durch die Deaktivierung der Sessions wird die Telemetrie-Datensammlung unterbunden. Um die beiden Sessions sowie die Übertragung von Telemetrie-Daten zu deaktivieren, muss zuerst der Dienst `Benutzererfahrung und Telemetrie im verbundenen Modus` deaktiviert werden. Dadurch wird die Initiierung der `DiagTrack-Listener` Session verhindert. Zusätzlich muss die `Autologger-DiagTrack-Listener` Session deaktiviert werden. Die Deaktivierung kann in der Registrierung vorgenommen werden; dazu muss der Wert des entsprechenden Registrierungsschlüssels auf „0“ gesetzt werden.

Für Umgebungen mit hohem Schutzbedarf (HD) sollten zudem zusätzlich die Empfehlungen für Netzwerk-basierte Maßnahmen berücksichtigt und umgesetzt werden (siehe Kapitel 3.2 in (ERNW_WP4_1)). Hierbei bietet sich die entsprechende Konfiguration eines DNS-Resolvers an, da diese in den meisten Unternehmensumgebungen und auch in Umgebungen versierter Endnutzer vorhanden sind und die benötigten Filterungsanforderungen effektiv und mit geringem Aufwand erfüllen, um Telemetrie-Kommunikation auf Netzwerkebene zu verhindern.

5.4.1.1 Deaktivierung Benutzererfahrung und Telemetrie im verbundenen Modus

Stellen Sie sicher, dass in den Gruppenrichtlinieneinstellungen der Dienst `Benutzererfahrung und Telemetrie im verbundenen Modus (DiagTrack)` unter `Windows-Einstellungen/Sicherheitseinstellungen/Systemdienste` auf folgenden Wert gesetzt ist: `Deaktiviert`.

Standardwert: Automatisch

Durch Deaktivierung des Dienstes `Benutzererfahrung und Telemetrie im verbundenen Modus`, wird die Initiierung der `DiagTrack-Listener` Session verhindert, die einen Teil der Quellen für Telemetrie-Daten darstellt, sowie eine Übertragung der protokollierten Daten an das Telemetrie-Backend unterbunden.

Alternativ können die folgenden Wege genutzt werden, um den Dienst zu deaktivieren:

Schnittstelle	Pfad/Befehl
Dienste (services.msc)	`Benutzererfahrung und Telemetrie im verbundenen Modus` →

Schnittstelle	Pfad/Befehl
	`Eigenschaften → Starttyp → Deaktiviert`
Registrierungs-Editor	`HKLM\SYSTEM\CurrentControlSet\Services\DiagTrack\ Start = 4`
PowerShell	`Set-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Services\ DiagTrack\ -Name Start -Value 4`

Tabelle 1: Deaktivierung Benutzererfahrung und Telemetrie im verbundenen Modus

5.4.1.2 Deaktivierung Autologger-Diagtrack-Listener

Stellen Sie sicher, dass der Registrierungswert `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\WMI\Autologger\AutoLogger-DiagTrack-Listener\Start` auf den Wert `0` gesetzt ist.

Standardwert: 1

Die `Autologger-Diagtrack-Listener` Session wird früh im Bootprozess initialisiert und speichert die protokollierten Daten in binärer Form im lokalen Dateisystem, bevor diese vom Telemetrie-Dienst nach dessen Start weiterverarbeitet werden. Um diese Protokollierung und lokale Speicherung der Daten zu unterbinden, sollte die `Autologger-Diagtrack-Listener` Session deaktiviert werden.

Alternativ können die folgenden Wege genutzt werden, um die Session zu deaktivieren:

Schnittstelle	Pfad/Befehl
PowerShell	`Set-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Control\WMI\Autologger\AutoLogger-Diagtrack-Listener\ -Name Start –Value 0`
Leistungsüberwachung (perfmon.exe)	`Datensammlersätze → Startereignis-Ablaufverfolgunssitzungen → AutoLogger-Diagtrack-Listener → Eigenschaften → Ablaufverfolgungssitzung → Haken bei **Aktiviert** entfernen`

Tabelle 2: Deaktivierung Autologger-Diagtrack-Listener

Detailliertere Beschreibungen zu den unterschiedlichen Konfigurationsmöglichkeiten der Telemetrie-relevanten Einstellungen sind dem Arbeitspaket 4.1 des SiSyPHuS-Projekts (ERNW_WP4_1) zu entnehmen.

5.5 PowerShell und Windows Script Host

5.5.1 PowerShell

Die Windows PowerShell stellt eine leistungsstarke Verwaltungsumgebung bereit. Über die Windows PowerShell kann auf eine Vielzahl von Ressourcen des Betriebssystems zugegriffen werden. Dazu zählt

beispielsweise die Windows-Registrierung, der Windows-Zertifikatsspeicher, die Windows-Umgebungsvariablen, das Dateisystem und eine Reihe von PowerShell-Ressourcen. Diese enorme Funktionsvielfalt stellt nicht nur dem Administrator eine große Anzahl von Werkzeugen bereit, die PowerShell wird auch von versierten Angreifern immer häufiger zur Durchführung komplexer Angriffe verwendet.

5.5.1.1 (HD, ND, NE) Deaktivierung von PowerShell Version 2.0

Stellen Sie sicher, dass PowerShell Version 2.0 deaktiviert ist.

Im Gegensatz zu PowerShell Version 5.1 (und aufwärts) werden in PowerShell Version 2.0 keine relevanten Sicherheitsfunktionalitäten implementiert, z. B. die Unterstützung der anti-malware scan interface (AMSI) oder die Protokollierung von PowerShell-Skriptblöcken (ERNW_WP8). Daher stellt die PowerShell Version 2.0 ein Sicherheitsrisiko dar und sollte deaktiviert werden.

5.5.1.1.1 Konfigurationsschnittstelle: PowerShell

Disable-WindowsOptionalFeature `Disable-WindowsOptionalFeature` ist ein PowerShell-Befehl mit dessen Hilfe PowerShell Version 2.0 deaktiviert werden kann.

```
Disable-WindowsOptionalFeature -Online -FeatureName
MicrosoftWindowsPowerShellV2Root
```

5.5.1.1.2 Konfigurationsschnittstelle: Windows-Features

Windows-Features ist ein Windows-Dienstprogramm zum Aktivieren oder Deaktivieren von Systemkomponenten.

Starten Sie die Windows-Anwendung `Systemsteuerung`. Klicken Sie auf `Programme`. Klicken Sie auf `Windows-Features aktivieren oder deaktivieren`. Im Dialogfeld `Windows-Features` deaktivieren Sie die Option `Windows PowerShell 2.0`. Übernehmen Sie die Änderungen mit `OK`.

5.5.1.2 (HD, ND, NE) Einschränkung der PowerShell-Skriptausführung

Stellen Sie sicher, dass Sie eine für Ihre Umgebung geeignete PowerShell-Ausführungsrichtlinie festlegen.

PowerShell unterstützt sogenannte Ausführungsrichtlinien („execution policies“). Mithilfe von Ausführungsrichtlinien können Benutzer die Bedingungen konfigurieren, unter denen PowerShell Skripte ausführt. Diese Richtlinien unterscheiden sich in ihrer Strenge und eignen sich für verschiedene Szenarien, in denen ein relativ hoher oder normaler Schutzbedarf besteht. Es ist wichtig zu betonen, dass die Ausführungsrichtlinien die versehentliche Ausführung von Skripten verhindern. Das hindert einen Angreifer aber beispielsweise nicht daran den Inhalt eines Skripts in eine PowerShell-Instanz zu kopieren und somit eine gesetzte Ausführungsrichtlinie zu umgehen (ms_ep, 2020).

Um den für jede Umgebung individuellen Kompromiss zwischen Sicherheit und Nutzbarkeit zu finden, bietet es sich an, die zu betrachtenden Ausführungsrichtlinien zu kategorisieren bzw. zu klassifizieren. Tabelle 3 gibt einen Überblick über die verschiedenen Ausführungsrichtlinien und die Szenarien, in denen sie geeignet sind (hoher Schutzbedarf - H; normaler Schutzbedarf – N).

Ausführungsrichtlinie	Beschreibung	Szenario
`Restricted`	Diese Ausführungsrichtlinie verbietet die Ausführung von Skripten.	H

Ausführungsrichtlinie	Beschreibung	Szenario
`AllSigned`	Diese Ausführungsrichtlinie erlaubt die Ausführung von Skripten, die von einem vertrauenswürdigen Herausgeber digital signiert wurden. Die Ausführung aller anderen Skripte ist nicht erlaubt.	
`RemoteSigned`	Diese Ausführungsrichtlinie erlaubt die Ausführung von Skripten, die von dem Computer stammen, auf dem die Richtlinie konfiguriert ist, wohingegen Skripts, die aus dem Internet stammen, von einem vertrauenswürdigen Herausgeber signiert werden müssen. Die Ausführung aller anderen Skripte ist nicht erlaubt.	N
`Unrestricted`	Diese Ausführungsrichtlinie erlaubt die Ausführung aller Skripte.	

Tabelle 3: PowerShell-Ausführungsrichtlinien

Abschnitt 5.5.1.2.1 und Abschnitt 5.5.1.2.2 bieten einen Überblick über die Schnittstellen zum Konfigurieren der oben aufgeführten Ausführungsrichtlinien.

5.5.1.2.1 Konfigurationsschnittstelle: PowerShell

Set-ExecutionPolicy `Set-ExecutionPolicy` ist ein PowerShell-Befehl mit dessen Hilfe PowerShell-Ausführungsrichtlinien aktiviert werden können.

`Set-ExecutionPolicy -ExecutionPolicy <Policy> -Scope <Scope>`

- `<Policy>` ist eine der in Tabelle 3, Spalte „Ausführungsrichtlinien“, aufgeführten Ausführungsrichtlinien (siehe auch (ms_expol, 2020))
- `<Scope>` gibt den Umfang der Auswirkungen der Ausführungsrichtlinie an, wie z. B:
  - `UserPolicy`: die Ausführungsrichtlinie wirkt sich auf den aktuellen Benutzer aus
  - `Process`: die Ausführungsrichtlinie wirkt sich auf die aktuelle PowerShell-Sitzung aus
  - `LocalMachine`: die Ausführungsrichtlinie betrifft alle Benutzer auf dem Computer, auf dem die Ausführungsrichtlinie konfiguriert ist.

Detaillierte Informationen zum `Set-ExecutionPolicy` PowerShell-Befehl finden Sie unter (ms_expol, 2020).

5.5.1.2.2 Konfigurationsschnittstelle: Gruppenrichtlinie

Ausführungsrichtlinien können mit der Gruppenrichtlinieneinstellung `Skriptausführung aktivieren` unter `Computerkonfiguration\Administrative Vorlagen\Windows PowerShell` konfiguriert werden.

Standardwert: Nicht konfiguriert (die Ausführung von Skripten ist nicht erlaubt, entspricht der Ausführungsrichtlinie `Restricted` – siehe Tabelle 3).

Mögliche Werte:

- `Nur signierte Skripts zulassen:` entspricht der Ausführungsrichtlinie `AllSigned` – siehe Tabelle 3
- `Lokale Skripts und remote signierte Skripts zulassen`: entspricht der Ausführungsrichtlinie `RemoteSigned` – siehe Tabelle 3

- `Alle Skripts zulassen`: entspricht der Ausführungsrichtlinie `Unrestricted` – siehe Tabelle 3.

5.5.1.3 (HD) Einschränkung der PowerShell-Skriptsprache (lokaler Computer)

PowerShell unterstützt das Konzept des Sprachmodus. Ein Sprachmodus bestimmt die zulässigen PowerShell-Befehle und Sprachelemente, die ein Benutzer ausführen darf. Der Sprachmodus ist eine PowerShell-Sitzungsvariable und daher sitzungsspezifisch (ERNW_WP8). Tabelle 4 gibt einen Überblick über alle PowerShell-Sprachmodi.

PowerShell-Sprachmodus	Beschreibung
`FullLanguage`	Dieser Sprachmodus erlaubt alle Befehle und Sprachelemente.
`RestrictedLanguage`	Dieser Sprachmodus erlaubt alle Befehle, schränkt jedoch die Ausführung von Skriptcode ein. Die Verwendung einiger Variablen und Operatoren ist zulässig. Skriptelemente wie Zuweisungsoperationen und Funktionsaufrufe sind nicht zulässig.
`NoLanguage`	Dieser Sprachmodus erlaubt alle Befehle, schränkt jedoch alle Sprachelemente ein.
`ConstrainedLanguage`	Dieser Sprachmodus erlaubt alle Befehle und Sprachelemente, beschränkt jedoch Befehle und Elemente basierend auf Datentypen. Eine umfassende Dokumentation der Funktionen dieses Sprachmodus finden Sie unter (ms_lm, 2020).

Tabelle 4: PowerShell-Sprachmodi

In lokalen PowerShell-Sitzungen ist der Standardsprachmodus `FullLanguage`. Windows erzwingt den `ConstrainedLanguage` Sprachmodus nur, wenn eine systemweite Anwendungssteuerungslösung wie Device Guard User Mode Code Integrity (UMCI) aktiviert ist. Wenn UMCI aktiviert ist, werden die PowerShell-Skripte, die nicht in der aktiven UMCI-Richtliniendatei angegeben sind, im Sprachmodus `ConstrainedLanguage` ausgeführt. Die in der bereitgestellten WDAC-Richtlinie angegebenen Skripte werden im Sprachmodus `FullLanguage` ausgeführt. Daher kann im Sprachmodus `FullLanguage` nur vertrauenswürdiger PowerShell-Code ausgeführt werden.

Aufgrund der hohen Einschränkung des `ConstrainedLanguage`-Sprachmodus und der UMCI-Richtlinien eignet sich dieser Sprachmodus nur für Szenarien, in denen ein sehr hoher Schutzbedarf besteht.

5.5.1.4 Sichere Verwendung von PowerShell-Remoting

PowerShell Remoting ist eine Funktionalität von PowerShell, mit der Benutzer über eine Netzwerkverbindung auf die PowerShell zugreifen können. Obwohl dies für Administrierungszwecke praktisch ist, können Angreifer es missbrauchen.

Stellen Sie sicher, dass PowerShell-Remoting nur aktiviert ist, wenn dies erforderlich ist.

PowerShell-Remoting ist in einigen Windows-Versionen (z. B. Windows Server 2012) standardmäßig aktiviert. Benutzer können PowerShell-Remoting mit dem PowerShell-Befehl `Enable-PSRemoting` aktivieren (ms_er, 2020).

5.5.1.4.1 Deaktivierung von PowerShell-Remoting

Die Deaktivierung von PowerShell-Remoting besteht aus mehreren Schritten. Dieser Abschnitt konzentriert sich auf PowerShell Version 5.1.

1. Führen Sie den PowerShell-Befehl `Disable-PSRemoting` aus.

Dieser PowerShell-Befehl deaktiviert den Remotezugriff auf alle PowerShell-Sitzungskonfigurationen. In Abschnitt 5.5.1.4.2 werden die PowerShell-Sitzungskonfigurationen erläutert.

2. (Option 1) Führen Sie die PowerShell-Befehle `Stop-Service WinRM -PassThru` und `Set-Service WinRM -StartupType Disabled -PassThru` aus.

Diese PowerShell-Befehle stoppen und deaktivieren den `WinRM`-Windows-Dienst. Dieser Dienst ermöglicht den Remotezugriff auf PowerShell über eine Netzwerkverbindung. In Windows ist dieser Dienst nicht nur für PowerShell-Remoting, sondern auch für die Serververwaltung erforderlich.

Wenn der `WinRM` -Dienst aus praktischen Gründen nicht deaktiviert werden kann, sollten seine PowerShell-„Listener“ (Netzwerkendpunkte, die PowerShell-Remoting ermöglichen) deaktiviert werden.

2. (Option 2) Führen Sie die PowerShell-Befehle `dir wsman:\localhost\listener` und `Remove-Item -Path WSMan:\localhost\listener\<Listener>` aus.

Der PowerShell-Befehl `dir wsman:\localhost\listener` gibt die Namen aller „Listener“ des `WinRM` -Dienstes aus. Der PowerShell-Befehl deaktiviert einen bestimmten „Listener“ (`<Listener>` ist ein Listenername).

3. Führen Sie den PowerShell-Befehl `Set-NetFirewallRule -DisplayName 'Windows Remote Management (HTTP-In)' -Enabled False -PassThru | Select -Property DisplayName, Profile, Enabled` aus.

Wenn PowerShell-Remoting mit dem PowerShell-Befehl `Enable-PSRemoting` aktiviert wird, werden Firewall-Ausnahmen erstellt, die den Remotezugriff auf den TCP-Port 5895 ermöglichen. Dieser Befehl deaktiviert die Firewall-Ausnahmen.

4. (optional, nur Einzelrechner) Stellen Sie sicher, dass der Registrierungswert `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\LocalAccountTokenFilterPolicy` auf den Wert `0` gesetzt ist.

Diese Einstellung stellt sicher, dass Benutzer keine Remotebefehle mit einem Administratorzugriffstoken ausführen können, ohne eine UAC (User Account Control)-Eingabeaufforderung auszulösen. Wenn PowerShell-Remoting mit dem PowerShell-Befehl `Enable-PSRemoting` aktiviert wird, können Benutzer Remotebefehle mit einem Administratorzugriffstoken ausführen, ohne eine UAC-Eingabeaufforderung auszulösen. Diese Einstellung wirkt sich nicht auf Domänenmitglieder aus.

5.5.1.4.2 Einschränkung von PowerShell-Remoting

Wenn PowerShell Remoting aktiviert sein soll, sollte es aus Sicherheitsgründen eingeschränkt werden. Just Enough Administration (JEA) ist ein PowerShell-Sicherheitsmechanismus, der das Prinzip der geringsten Berechtigungen implementiert. Es erstellt PowerShell-Instanzen (PowerShell-Sitzungen) zum Verwalten von Systemen mit einem begrenzten Satz verfügbarer PowerShell-Funktionalitäten. Dies sind die Mindestfunktionalitäten, die für die Durchführung bestimmter Systemverwaltungsaufgaben erforderlich sind.

JEA nutzt das PowerShell-Remoting, sodass Benutzer eine Verbindung zu einer PowerShell-Sitzung mit eingeschränkten Funktionalitäten herstellen können. Diese Sitzung wird als JEA-Endpunkt bezeichnet. Benutzer können mithilfe des PowerShell-Befehls `Enter-PSSession` eine Verbindung zum JEA-Endpunkt herstellen (ms_es, 2020).

Ein typischer Ansatz zur Konfiguration von JEA besteht aus mehreren Phasen. In den folgenden Abschnitten werden diese Phasen erläutert und ein minimales Beispiel für eine JEA-Konfiguration angegeben. Detaillierte Informationen zu JEA finden Sie unter (ms_jea, 2020).

Phase 1: Planung Die Konfiguration und Bereitstellung von JEA erfordert eine sorgfältige Planung. Ziel der ersten Phase ist es, das richtige Gleichgewicht zwischen Funktionalität und Sicherheit für ein bestimmtes Szenario zu finden. Zu diesem Zweck sollten zwei Fragen beantwortet werden:

- Welche Benutzer (oder Benutzergruppen) sollten Zugriff auf den JEA-Endpunkt haben?
- Welche PowerShell-Funktionalitäten sollten diesen Benutzern zur Verfügung gestellt werden?

Beispielsweise sollte die PowerShell-Instanz des Benutzers `test` eingeschränkt werden, damit nur die PowerShell-Befehle `Get-Help` und `Get-History` ausgeführt werden können.

Phase 2: JEA-Rollen festlegen In der zweiten Phase werden die zugelassenen PowerShell- Funktionalitäten für einen bestimmten Benutzer (oder eine Benutzergruppe) in Form von JEA-Rollen angegeben. JEA-Rollen werden in JEA- Rollenfunktionsdateien angegeben (Dateinamenserweiterung: `.psrc`).

Eine Rollenfunktionsdatei sollte in einem Ordner mit dem Namen `RoleCapabilities` gespeichert werden. Dieser Ordner sollte in einem Ordner eines PowerShell-Moduls gespeichert werden, z. B. in einem Unterordner, der unter `C:\Programme\WindowsPowerShell\Modules` gespeichert ist.

Wichtig: Der Zugriff auf den Ordner `RoleCapabilities` sollte restriktiv sein. Nur vertrauenswürdige Administratoren sollten auf den Ordner zugreifen und die Rollenfunktionsdatei ändern können. Andernfalls könnte ein nicht vertrauenswürdiger Benutzer die Rollenfunktionsdatei ändern und die von JEA auferlegten Einschränkungen umgehen.

Erstellen und bearbeiten Sie eine Rollenfunktionsdatei.

Mit dem PowerShell-Befehl `New-PSRoleCapabilityFile` kann eine leere Vorlagenrollenfunktionsdatei erstellt werden. Die erstellte Datei kann mit einem Texteditor bearbeitet werden. In der Rollenfunktionsdatei sollten, die in Phase 1 identifizierten PowerShell-Funktionalitäten angegeben werden. Abbildung 1 zeigt die Rollenfunktionsdatei für den Benutzer `test` (Dateiname: `testRole.psrc`).

```
@{

GUID = '0712c3ff-41cc-4dd8-9368-8e6fad3f90f1'

VisibleCmdlets = 'Get-Help', 'Get-History'

}
```

Abbildung 1: Die Rollenfunktionsdatei für den Benutzer test (testRole.psrc)

JEA unterstützt die Angabe zulässiger PowerShell-Funktionalitäten durch Schlüsselwörter, die in die Rollenfunktionsdatei geschrieben werden. Einige Schlüsselwörter sind:

- `VisibleCmdlets`: gibt die zulässigen PowerShell-Befehle an (cmdlets, siehe Abbildung 1: Die Rollenfunktionsdatei für den Benutzer test (testRole.psrc), (ERNW_WP8));
- `VisibleProviders`: gibt die zulässigen PowerShell-Provider an, z. B. `Registry`. PowerShell greift über Provider auf Systemressourcen zu;
- `VisibleExternalCommands`: gibt die im Dateisystem gespeicherten ausführbaren Dateien an, die über PowerShell ausgeführt werden können.

Detaillierte Informationen zu Rollenfunktionsdateien finden Sie unter (ms_rfd, 2020).

Speichern Sie die erstellte Rollenfunktionsdatei.

Wenn die Rollenfunktionsdatei fertiggestellt ist, sollte sie in einen Ordner mit dem Namen `RoleCapabilities` kopiert werden. Die PowerShell-Befehle in Abbildung 2: Speichern der

Rollenfunktionsdatei testRole.psrc : [1] erstellen einen Ordner für ein PowerShell-Modul mit dem Namen `testJEA` (`C:\Program Files\WindowsPowerShell\Modules\testJEA`); [2] registrieren das Modul; [3] erstellen den `RoleCapabilities` Ordner (Unterordner von `testJEA`) und kopieren die Datei `testRole.psrc` aus dem aktuellen Ordner in diesem Ordner.

```
$modulePath = Join-Path $env:ProgramFiles "WindowsPowerShell\Modules\testJEA"
New-Item -ItemType Directory -Path $modulePath

New-Item -ItemType File -Path (Join-Path $modulePath "testJEAFunctions.psm1")
New-ModuleManifest -Path (Join-Path $modulePath "testJEA.psd1") -RootModule "testJEAFunctions.psm1"

$rcFolder = Join-Path $modulePath "RoleCapabilities"
New-Item -ItemType Directory $rcFolder
Copy-Item -Path .\testRole.psrc -Destination $rcFolder
```

Abbildung 2: Speichern der Rollenfunktionsdatei testRole.psrc

Phase 3: Sitzungskonfiguration festlegen In der dritten Phase erfolgt die Zuordnung zwischen Benutzern und JEA-Rollen. Außerdem können Einstellungen für die durch JEA eingeschränkte PowerShell-Sitzung (d. h. JEA-Endpunkt) festgelegt werden. Diese Einstellungen werden in einer Sitzungskonfigurationsdatei angegeben (Dateinamenserweiterung: `.pssc`).

Erstellen und bearbeiten Sie eine Sitzungskonfigurationsdatei.

Mit dem PowerShell-Befehl `New-PSSessionConfigurationFile` kann eine leere Vorlagenkonfigurationsdatei erstellt werden. Die erstellte Datei kann mit einem Texteditor bearbeitet werden. Abbildung 3 zeigt die Sitzungskonfigurationsdatei für den Benutzer `test` (Dateiname: `testSession.pssc`).

```
@{

SchemaVersion = '2.0.0.0'

GUID = 'a73e1139-562e-459b-a3b1-63b2facfcf0e'

SessionType = 'RestrictedRemoteServer'

TranscriptDirectory = 'C:\Transcripts\'

RunAsVirtualAccount = $true

RoleDefinitions = @{ 'DESKTOP-ASRI2Q0\test' = @{ RoleCapabilities = 'testRole' } }

}
```

Abbildung 3: Die Sitzungskonfigurationsdatei für den Benutzer test (testSession.pssc)

JEA unterstützt die Angabe von Sitzungseinstellungen über Schlüsselwörter, die in die Sitzungskonfigurationsdatei geschrieben werden. Einige Schlüsselwörter sind:

- `RoleDefinitions`: gibt die Zuordnung zwischen Benutzern (oder Benutzergruppen) und JEA-Rollen an (siehe Abbildung 3). Das Schlüsselwort `RoleCapabilities` sollte den Namen der Rollenfunktionsdatei ohne die Dateinamenerweiterung angeben (`testRole` in Abbildung 3);
- `RunAsVirtualAccount`: gibt an, dass der Benutzer die von JEA zugelassenen PowerShell-Funktionalitäten mit Administratorrechten verwenden darf (siehe Abbildung 3);
- `LanguageMode`: gibt einen PowerShell-Sprachmodus an (siehe Tabelle 4);

- `SessionType`: gibt einen Sitzungstyp an. Im Kontext von JEA wird der Sitzungstyp `RestrictedRemoteServer` empfohlen (siehe Abbildung 3). Dieser Sitzungstyp konfiguriert die Sitzung so, dass nur die durch das Schlüsselwort `RoleDefinitions` angegebenen PowerShell-Funktionalitäten zulässig sind. Darüber hinaus wird der PowerShell-Sprachmodus `NoLanguage` erzwungen;
- `TranscriptDirectory`: weist PowerShell an, ein vollständiges Protokoll aller eingegebenen PowerShell-Befehle in einer Datei im angegebenen Ordner zu speichern. Dies wird empfohlen.

Detaillierte Informationen zu Sitzungskonfigurationsdateien finden Sie unter (ms_skd, 2020).

Registrieren Sie die Sitzungskonfiguration.

Wenn die Sitzungskonfigurationsdatei fertiggestellt ist, sollte die Sitzungskonfiguration unter einem bestimmten Namen registriert werden. Eine Sitzungskonfiguration kann mit dem PowerShell-Befehl `Register-PSSessionConfiguration` PowerShell registriert werden. Zum Beispiel registriert der PowerShell-Befehl:

`Register-PSSessionConfiguration -Name testJEAEndpoint -Path C:\ProgramData\JEAConfigurations\testSession.pssc -Force`

die Sitzungskonfigurationsdatei `testSession.pssc` (siehe Abbildung 3), die im Ordner `C:\ProgramData\JEAConfigurations` gespeichert ist, unter dem Namen `testJEAEndpoint`.

Phase 4: Testen Wenn die Sitzungskonfiguration registriert ist, ist der JEA-Endpunkt aktiv und sollte getestet werden.

Testen Sie den JEA-Endpunkt mithilfe des Befehls `Enter-PSSession` PowerShell.

Beispiel: Der PowerShell-Befehl `Enter-PSSession testhost -ConfigurationName testJEAEndpoint -Credential test` stellt eine Verbindung zum JEA-Endpunkt her, der auf dem Computer mit dem Namen `testhost` gehostet wird, und gibt die Sitzungskonfiguration mit dem Namen `testJEAEndpoint` (Parameter `ConfigurationName`) an. Die PowerShell-Sitzung wird für den Benutzer `test` eingerichtet (Parameter `Credential`). Wie in der Rollenfunktionsdatei angegeben, kann der Benutzer `test` nur die PowerShell-Befehle `Get-Help` und `Get-History` ausführen. Alle anderen PowerShell-Funktionalitäten sind nicht zulässig (siehe Abbildung 1: Die Rollenfunktionsdatei für den Benutzer test (testRole.psrc)).

5.5.2 Windows Script Host

Der Windows Script Host (WSH) stellt eine Laufzeitumgebung für Skriptsprachen dar (ERNW_WP8). Der Windows Script Host kann von Anwendern und Administratoren dazu verwendet werden, Aufgaben zu automatisieren.

5.5.2.1 (ND, NE) Ausführung von vertrauenswürdigen Skripten

Stellen Sie sicher, dass der Registrierungswert `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Script Host\Settings\TrustPolicy` auf den Wert `2` gesetzt ist.

Diese Einstellung verhindert die Ausführung von nicht signierten Skripten, also Skripte deren digitale Signaturen nicht überprüft werden können, oder von nicht vertrauenswürdigen Herausgebern signiert worden sind.

Standardmäßig ist die Ausführung aller Skripte zulässig. Wenn der Registrierungswert `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Script Host\Settings\TrustPolicy` nicht vorhanden ist, soll er erstellt werden.

Unsignierte Skripte können dadurch beispielsweise nicht mehr zur Administration verwendet werden.

5.5.2.2 (HD) Deaktivierung von Windows Script Host

Stellen Sie sicher, dass der Registrierungswert `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Script Host\Settings\Enabled` auf den Wert `0` gesetzt ist.

Diese Einstellung deaktiviert den Windows Script Host. Der Windows Script Host ist standardmäßig aktiv. Wenn der Registrierungswert `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Script Host\Settings\Enabled` nicht vorhanden ist, soll er erstellt werden.

Wenn Windows Script Host deaktiviert ist, können Skripte, deren Ausführung von WSH unterstützt wird, nicht ausgeführt werden. Dies schützt vor böswilligen Skripten, verhindert jedoch auch die Ausführung legitimer Skripte.

Es ist wichtig zu betonen, dass diese Einstellung nur die Ausführung von Skripten durch die Benutzeroberfläche blockiert. Die WSH-Kernfunktionalitäten sind weiterhin aktiv.

5.5.2.3 (HD) Deaktivierung von Windows Script Host-Remoting

Stellen Sie sicher, dass der Registrierungswert `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Script Host\Settings\Remote` auf den Wert `0` gesetzt ist.

Diese Einstellung deaktiviert das WSH-Remoting. WSH-Remoting ist standardmäßig deaktiviert. Wenn WSH-Remoting aktiviert ist, kann WSH Skripte auf einer Windows-Instanz ausführen, die über eine Netzwerkverbindung erreichbar ist. Auf diese Weise können Angreifer in diesen Windows-Instanzen schädliche Skripte ausführen.

5.6 (HD, ND, NE) Firmware

Die Firmware ist eine Software zur Steuerung der Hardware eines Computers. Der aktuelle Firmwarestandard ist UEFI (Unified Extensible Firmware Interface) und der ältere Firmwarestandard ist BIOS (basic input/output system). UEFI bietet mehr Funktionalitäten, einschließlich sicherheitsrelevanterer Einstellungen. Dieser Abschnitt konzentriert sich auf die sichere Konfiguration der UEFI-Firmware.

Da es mehrere Firmware-Hersteller gibt und einige Firmware-Einstellungen hardwareabhängig sind, fehlen möglicherweise einige Einstellungen, oder einige der in diesem Abschnitt aufgeführten Einstellungen sind möglicherweise nicht auf allen Plattformen verfügbar. Dieser Abschnitt konzentriert sich auf die am häufigsten verfügbaren Firmware-Einstellungen.

5.6.1 Administratorkennwort

Stellen Sie sicher, dass ein UEFI-Administratorkennwort festgelegt ist.

Das UEFI-Administratorkennwort verhindert das unbefugte Ändern von UEFI-Einstellungen. UEFI kann normalerweise so konfiguriert werden, dass dieses Kennwort bei jedem Systemstart oder auch (nur) beim Starten der grafischen UEFI-Benutzeroberfläche erforderlich ist.

5.6.2 Einschränkung der Bootreihenfolge

Stellen Sie sicher, dass die Liste der Geräte, von denen ein Betriebssystem gestartet werden kann (d. h. die Startreihenfolge), nur auf die erforderlichen Geräte beschränkt ist.

Diese Einstellung verhindert das Booten von Betriebssystemen von unbekannten Geräten. Ein solches Booten kann zum Diebstahl von Daten führen, wenn die Festplatte nicht verschlüsselt ist.

5.6.3 Deaktivierung von Legacy-Firmware-Funktionalitäten

Stellen Sie sicher, dass alle Legacy-Firmware-Funktionalitäten deaktiviert sind.

Diese Einstellung deaktiviert die Verwendung von Legacy-Firmware-Funktionalitäten (BIOS-Funktionalitäten), z. B. den Legacy-Startmodus ("boot mode") oder die Legacy-Bootoptionen ("boot options"). Legacy-Firmware-Funktionalitäten bieten nicht die von UEFI angebotenen sicherheitsrelevanten Funktionalitäten (z. B. sicherer Start, siehe Abschnitt 5.2.1.1).

5.6.4 Sichere Firmware-Updates

Stellen Sie sicher, dass nur vom Hersteller signierte Firmware-Updates installiert werden können.

Diese Einstellung verhindert die Installation von Firmware-Updates, die von einer unbekannten Quelle stammen.

Sie sicher, dass das Firmware-Rollback deaktiviert oder zumindest durch das Administratorkennwort geschützt ist (d. h. gesperrt ("locked"), siehe Abschnitt 5.6.1).

Diese Einstellung verhindert die Installation einer älteren UEFI-Firmware-Version.

Stellen Sie sicher, dass das Firmware-Update über Windows aktiviert ist.

Diese Einstellung stellt die automatisierte und vertrauenswürdige Installation der neuesten Firmware-Version sicher.

5.6.5 Sonstiges

Das UEFI ist eine technische Voraussetzung für mehrere sicherheitsrelevante Windows-Komponenten und -Funktionalitäten. In diesem Abschnitt werden einige UEFI-Einstellungen aufgeführt, die aktiviert werden müssen, damit diese Komponenten und Funktionalitäten ordnungsgemäß funktionieren.

Stellen Sie sicher, dass das TPM aktiviert ist.

Diese Einstellung aktiviert das TPM (siehe Abschnitt 5.3). Wenn es in der Firmware nicht aktiviert ist, kann Windows das TPM nicht verwenden.

Stellen Sie sicher, dass die CPU-Virtualisierungserweiterungen aktiviert sind.

Diese Einstellung aktiviert die CPU-Virtualisierungserweiterungen (z. B., Intel Virtualization Technology - VT-x). Wenn sie nicht aktiviert sind, sind die Sicherheitsfunktionen der virtualisierungsbasierten Sicherheit nicht funktionsfähig (siehe Abschnitt 5.2).

Stellen Sie sicher, dass der sichere Start (SecureBoot) aktiviert ist.

Diese Einstellung aktiviert den sicheren Start (SecureBoot). Wenn es in der Firmware nicht aktiviert ist, wird die sichere Start-Funktionalität nicht das Booten von Windows sichern.

Stellen Sie sicher, dass Ausführungsverhinderung ("execution prevention") aktiviert ist.

Diese Einstellung aktiviert die Ausführungsverhinderung ("execution prevention", NX). Die Ausführungsverhinderung verhindert den unbefugten Zugriff auf bestimmte Speicherbereiche. Daher schützt es vor der Ausführung von Schadsoftware.

Appendix

Werkzeuge

Werkzeug	Verfügbarkeit und Beschreibung
`Gruppenrichtlinienverwaltungs-Editor`	*Verfügbarkeit:* Verteilt mit Windows 10 *Beschreibung:* Ein Werkzeug zur Konfiguration von Gruppenrichtlinien.
`Registrierungs-Editor`	*Verfügbarkeit:* Verteilt mit Windows 10 *Beschreibung:* Ein Werkzeug zur Konfiguration der Registrierung.
`Dienste`	*Verfügbarkeit:* Verteilt mit Windows 10 *Beschreibung:* Ein Werkzeug zur Konfiguration von Systemdiensten.
`Leistungsüberwachung`	*Verfügbarkeit:* Verteilt mit Windows 10 *Beschreibung:* Ein Werkzeug zur Anzeige und Konfiguration von Leistungsdaten und Protokollen.

Referenzen

bsi_it_gs_orp4. (17. Juli 2020). Von https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKompendium/bausteine/ORP/ORP_4_Identit%C3%A4ts-_und_Berechtigungsmanagement.html abgerufen

cis_win10_1809. (22. November 2019). *CIS Microsoft Windows 10 Enterprise (Release 1809) Benchmark v1.6.1*. Von https://www.cisecurity.org/cis-benchmarks/ abgerufen

ERNW_WP10. (kein Datum). SiSyPHuS Win10 (Studie zu Systemaufbau, Protokollierung, Härtung und Sicherheitsfunktionen in Windows 10): Work Package 10.

ERNW_WP11. (kein Datum). SiSyPHuS Win10 (Studie zu Systemaufbau, Protokollierung, Härtung und Sicherheitsfunktionen in Windows 10): Work Package 11.

ERNW_WP12. (kein Datum). SiSyPHuS Win10 (Studie zu Systemaufbau, Protokollierung, Härtung und Sicherheitsfunktionen in Windows 10): Work Package 12.

ERNW_WP2. (kein Datum). SiSyPHuS Win10 (Studie zu Systemaufbau, Protokollierung, Härtung und Sicherheitsfunktionen in Windows 10): Work Package 2.

ERNW_WP4_1. (kein Datum). SiSyPHuS Win10 (Studie zu Systemaufbau, Protokollierung, Härtung und Sicherheitsfunktionen in Windows 10): Work Package 4.1.

ERNW_WP5. (kein Datum). SiSyPHuS Win10 (Studie zu Systemaufbau, Protokollierung, Härtung und Sicherheitsfunktionen in Windows 10): Work Package 5.

ERNW_WP7. (kein Datum). SiSyPHuS Win10 (Studie zu Systemaufbau, Protokollierung, Härtung und Sicherheitsfunktionen in Windows 10): Work Package 7.

ERNW_WP8. (kein Datum). SiSyPHuS Win10 (Studie zu Systemaufbau, Protokollierung, Härtung und Sicherheitsfunktionen in Windows 10): Work Package 8.

ms_applocker. (17. Juli 2020). Von https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-application-control/applocker/applocker-overview abgerufen

ms_ep. (17. Juli 2020). Von https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_execution_policies?view=powershell-5.1 abgerufen

ms_er. (17. Juli 2020). Von https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/enable-psremoting?view=powershell-5.1 abgerufen

ms_es. (17. Juli 2020). Von https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/enter-pssession?view=powershell-5.1 abgerufen

ms_expol. (17. Juli 2020). Von https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.security/set-executionpolicy?view=powershell-6 abgerufen

ms_jea. (17. Juli 2020). Von https://docs.microsoft.com/de-de/powershell/scripting/learn/remoting/jea/overview?view=powershell-5.1 abgerufen

ms_lm. (17. Juli 2020). Von https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_language_modes?view=powershell-5.1 abgerufen

ms_mss. (17. Juli 2020). Von https://blogs.technet.microsoft.com/secguide/2016/10/02/the-mss-settings/ abgerufen

ms_pass. (17. Juli 2020). Abgerufen am 17. Juli 2020 von https://docs.microsoft.com/en-us/archive/blogs/msftcam/password-complexity-versus-password-entropy

ms_pfx. (17. Juli 2020). Von https://docs.microsoft.com/de-de/powershell/module/microsoft.powershell.security/get-pfxcertificate?view=powershell-6 abgerufen

ms_ps. (17. Juli 2020). Von https://docs.microsoft.com/de-de/powershell/module/microsoft.powershell.security/set-authenticodesignature?view=powershell-6 abgerufen

ms_rfd. (17. Juli 2020). Von https://docs.microsoft.com/de-de/powershell/scripting/learn/remoting/jea/role-capabilities?view=powershell-5.1 abgerufen

ms_sec_bl_1809. (17. Juli 2020). Von https://www.microsoft.com/en-us/download/details.aspx?id=55319 abgerufen

ms_sign. (17. Juli 2020). Von https://docs.microsoft.com/de-de/dotnet/framework/tools/signtool-exe abgerufen

ms_skd. (17. Juli 2020). Von https://docs.microsoft.com/de-de/powershell/scripting/learn/remoting/jea/session-configurations?view=powershell-5.1 abgerufen

Abkürzungen

AMSI: anti-malware scan interface 21
BIOS: basic input/output system 28, 29
BSI: Bundesamt für Sicherheit in der Informationstechnik 5
ETW: Event Tracing for Windows 19
JEA: Just Enough Administration 24, 25, 26, 27
LTSC: Long Term Servicing Channel 5, 6
SAC: Semi-Annual Channel 6
UAC: User Account Control 24
UEFI: Unified Extensible Firmware Interface 15, 16, 17, 28, 29
UMCI: User Mode Code Integrity 23
VBS: Virtualisierungsbasierte Sicherheit 15, 16, 17
WSH: Windows Script Host 27, 28