



# Windows 10 セキュリティ評価支援報告書

Phase2

## CONTENTS

<b>1</b>	エグゼクティブサマリ	2
<b>2</b>	背景及び目的	2
<b>3</b>	評価要領	3
	3.1. 脆弱性攻撃対策	3
	3.1.1 メモリ破壊にかかわる脆弱性攻撃対策	3
	3.1.2 64bit版における DEPとASLRの拡張	5
	3.1.3 脆弱性攻撃対策の評価	6
	3.2. Pass-the-Hash攻撃対策	7
	3.2.1 Pass-the-Hash攻撃とは	7
	3.2.2 Pass-the-Hash攻撃対策	8
	3.2.3 Pass-the-Hash攻撃対策の評価	9
<b>4</b>	評価結果	11
	4.1. 脆弱性攻撃対策	11
	4.1.1 Supervisor Mode Execution Prevention (SMEP)	11
	4.1.2 Control Flow Guard (CFG)	12
	4.2. Pass-the-Hash攻撃対策	13
	4.2.1 Mimikatz	14
	4.2.2 gsecdump	15
	4.2.3 Pwdump7	16
	4.2.4 QuarksPwDump	17
<b>5</b>	むすび:Windows 10を利用する上での提言	18
	5.1. 脆弱性攻撃対策	18
	5.2. Pass-the-Hash攻撃対策	18



株式会社 FFRI

# 1 エグゼクティブサマリ

サイバー攻撃の深刻度は増しており、セキュリティ製品による対策だけではなく、OS等のシステムそのもののセキュリティレベルを維持することが重要となっている。このような観点からDoD(米国国防総省)などの高いセキュリティレベルを要求される組織でも導入が進められている<sup>1</sup>。マイクロソフト社のWindows 10のセキュリティ機能について評価を行った。Windows 10では、これまでに導入をされたセキュリティ機能に加え、ハードウェアレベルのセキュリティ機能、新たな脆弱性攻撃対策、Path-the-Hashと呼ばれる資格情報を奪取する攻撃への対策が導入されている。本調査においては、Windows 10のセキュリティレベルの高さを確認すると共に、2020年にサポートが終了するWindows 7は2世代前のシステムであり、現在の攻撃を防ぐことが難しいことを改めて確認することになった。

一方で、攻撃手法と対策技術は常に競争関係にあり、Windows 10を前提とした攻撃手法が継続的に研究開発されていることは無視できない。本調査では、これらの点を踏まえ組織におけるWindowsのセキュリティ対策として以下のフレームワークを推奨する。

1. セキュリティレベルの向上が確認された最新のOSやアプリケーションを利用する
2. 全てのPC・サーバーに対して適切な設定を適用する
3. 脆弱性を排除するためのセキュリティ更新を確実に実施する
4. 常に対策状況を把握し、新たに公表される脆弱性を評価し対応する
5. 攻撃が成功することを前提とした検知・対応の仕組みを構築する

また、近年のサイバー攻撃において鍵となっているPath-the-Hashについては、特に注意を払う必要がある。適切なアカウントを利用したWindows 10においては、現在の代表的なツールを使った攻撃を阻止できることが確認できた。しかし攻撃と対策は常に競争関係にあり、新たな攻撃が開発されると想定されるため、本調査では、Path-the-Hash攻撃の対策として以下のフレームワークを推奨する。

1. 原則としてユーザー権限のアカウントを利用する
2. 管理者権限が必要な際は、管理者グループ等を利用する(ビルドインアドミニストレータを利用しない)
3. Windows 10でCredential Guardを利用する
4. Restricted Administratorを活用する
5. ドメインユーザーの資格情報をキャッシュする数を制限する<sup>2</sup>
6. ローカルユーザーとドメインユーザーに同じパスワードを用いない
7. キットティング用のアカウントのパスワードをユニークにする(LAPSの利用)

1 <https://blogs.windows.com/japan/2016/02/19/us-department-of-defense-commits-to-upgrade-4-million-seats-to-windows-10/>

2 [https://technet.microsoft.com/ja-jp/library/mt629048\(v=vs.85\).aspx](https://technet.microsoft.com/ja-jp/library/mt629048(v=vs.85).aspx)

## 2 背景及び目的

2015年7月29日のWindows 10アップグレード版リリースから既に約1年半が経過しているが、Phase I 報告書<sup>3</sup>で言及した通りエンタープライズ市場におけるWindows 10の導入率は必ずしも高いとは言えない。

同報告書で記載したようにWindows 10にはWindows 7、8.1で発見、確認された様々な攻撃に対する新たな防御機能が搭載されており、セキュリティ面での導入の効果は高いと考えられる。そこで本書では以下の2つの観点で

Windows 7、Windows 10のセキュリティ機能の比較を行い、両環境における防御能力の比較評価を行う。また、その結果を踏まえてWindows 10を利用する上で考慮すべき事項について考察、提案する。

- 脆弱性攻撃対策
- Pass-the-Hash攻撃対策

3 [http://www.ffri.jp/assets/files/research/research\\_papers/windows10\\_security\\_ja.pdf](http://www.ffri.jp/assets/files/research/research_papers/windows10_security_ja.pdf)

# 3 3. 評価要領

サイバー攻撃への対策には、OS等のシステムそのもののセキュリティレベルを維持することが重要である。ここでは、脆弱性攻撃対策と、Path-the-Hash攻撃の概要を紹介するとともに、Windows 8.1までに実装された代表的な対策について解説する。

## 3.1. 脆弱性攻撃対策

本章では、マイクロソフト社の代表的な脆弱性攻撃対策を紹介する。これまでの取り組みに関しては、Exploit Mitigation Improvements in Windows 8<sup>4</sup>もしくはWindows 10 Mitigation Improvements<sup>5</sup>、などで細かく紹介されており、日本語ではBeyond Zero-day Attacks/ゼロデイ攻撃をめぐる攻防<sup>6</sup>として記事化されている。本章では、これらの資料を抜粋し

て以下に記載する。

- 4 [http://media.blackhat.com/bh-us-12/Briefings/M\\_Miller/BH\\_US\\_12\\_Miller\\_Exploit\\_Mitigation\\_Slides.pdf](http://media.blackhat.com/bh-us-12/Briefings/M_Miller/BH_US_12_Miller_Exploit_Mitigation_Slides.pdf)
- 5 <https://www.blackhat.com/docs/us-16/materials/us-16-Weston-Windows-10-Mitigation-Improvements.pdf>
- 6 <http://www.atmarkit.co.jp/ait/series/1568/>

### 3.1.1. メモリ破壊にかかわる脆弱性攻撃対策

脆弱性を悪用して悪意のコードを実行するために攻撃者は、プログラム実行の制御フローを乗っ取り、攻撃対象プロセスのメモリ上に配置した攻撃コードを実行させる必要がある。制御フローの乗っ取りにはスタック

ベースの脆弱性(図 1)やヒープベースの脆弱性(図 2)を使った攻撃があり、攻撃コードの実行には図 3のような命令ポインタの制御に記載の攻撃手法がある。

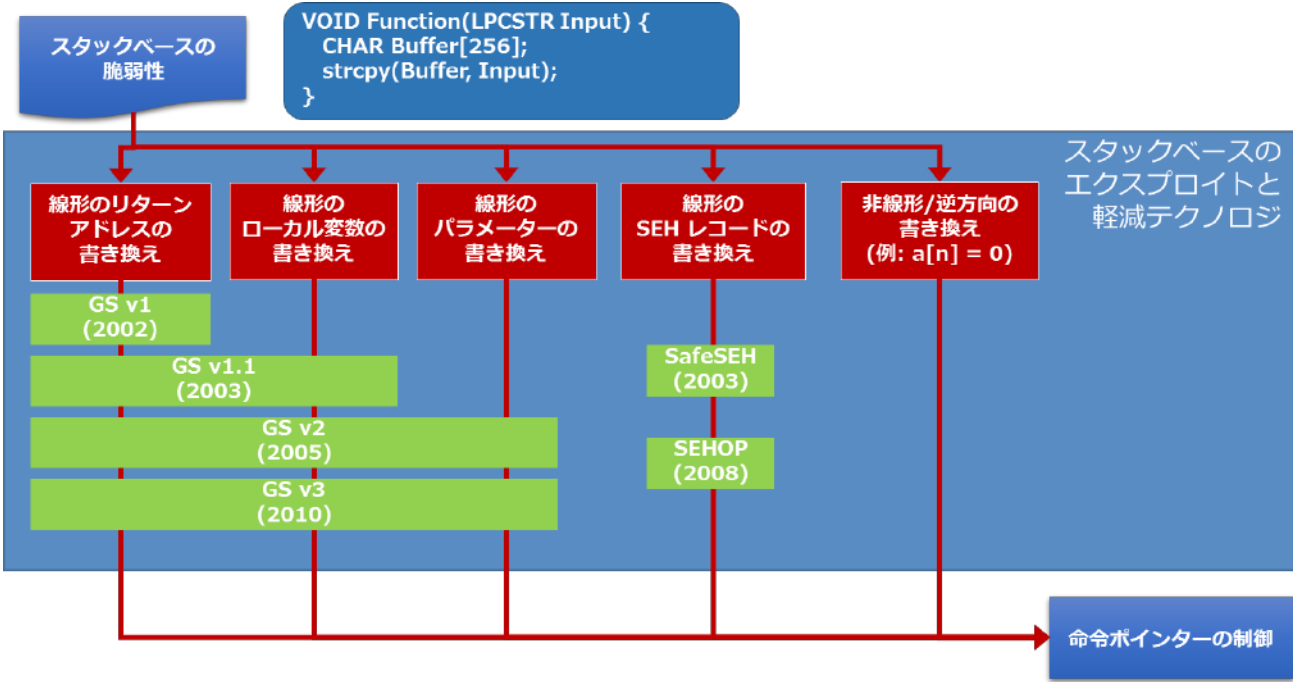
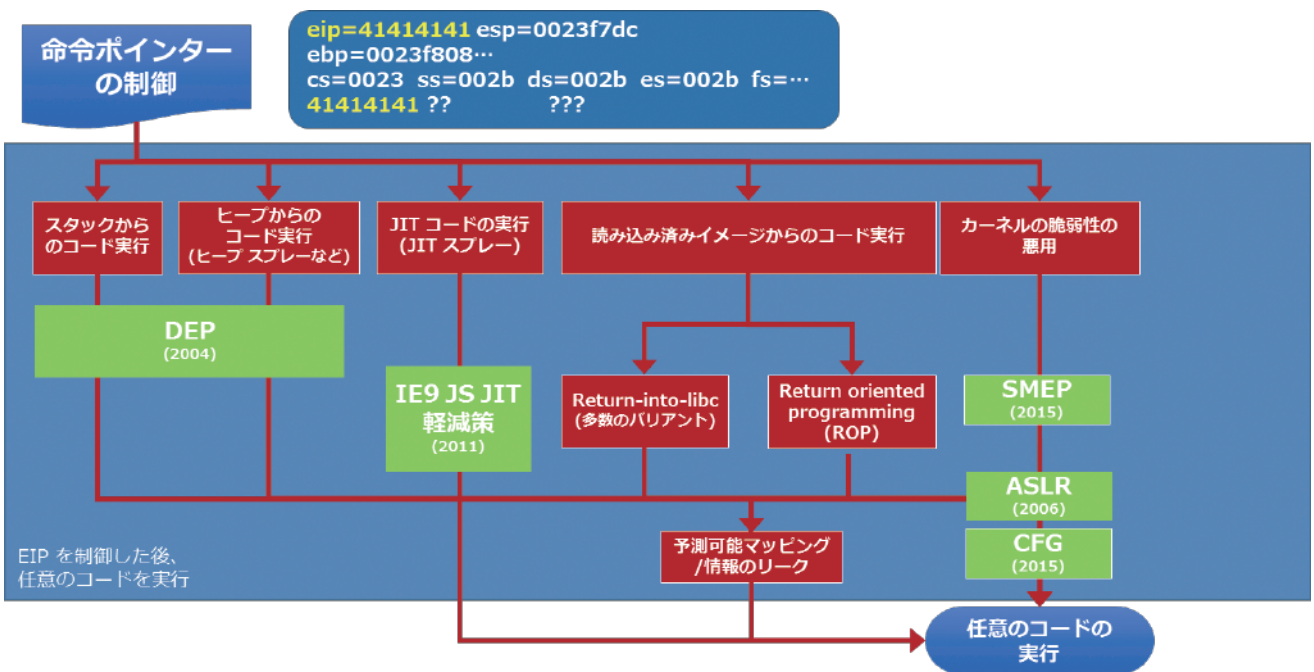
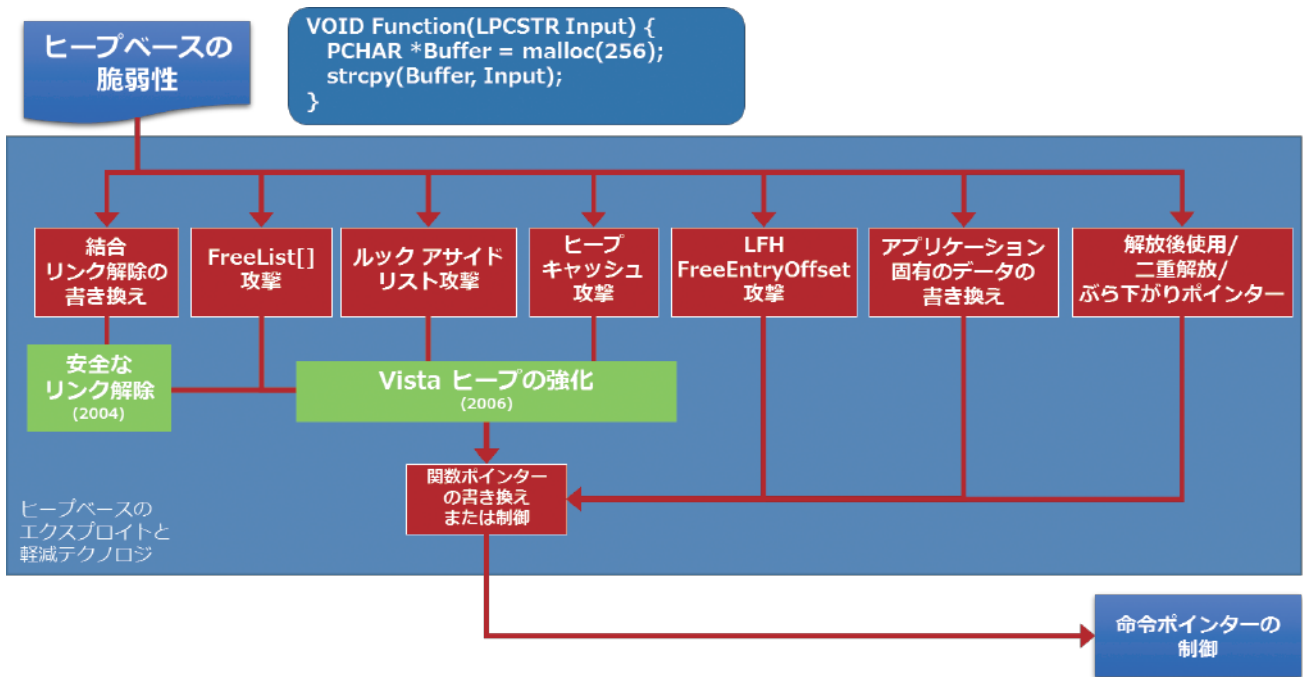


図 1 スタックベースの脆弱性とその代表的な対策



それぞれの攻撃カテゴリーには多くの手法があり、Windowsは、これに対応する様々な脆弱性攻撃対策を積み

重ねてきた歴史がある。以下にWindowsやWindowsの開発環境で実装された主要な対策機能を記載する。

表 1 Windowsで利用可能な脆弱性攻撃対策機能の一例

バッファのセキュリティチェック (GS)	スタック上に配置されるリターンアドレスの近隣にスタックガードと呼ばれる値を配置し、関数からのリターン前に値の検証を行うことでスタックの破壊を検出する(リターンアドレスの制御、乗っ取りを防ぐ)。
構造化例外処理上書き保護 (SEHOP)	構造化例外処理のリストの終端に特定の例外ハンドラー (FinalExceptionHandler)を設置し、例外発生時にリストを走査し、FinalExceptionHandlerに辿りつくか否かを検証することで例外ハンドラーの上書きを検出する(例外ハンドラーの制御、乗っ取りを防ぐ)。
安全な例外ハンドラーがあるイメージ (/SAFESEH)	プログラムのコンパイル時にプログラム中に含まれる例外ハンドラーの情報を実行ファイルに埋め込む。プログラム実行時に実行する例外ハンドラーを埋め込まれた情報と比較し、含まれていない例外ハンドラーの実行を禁止する(例外ハンドラーの制御、乗っ取りを防ぐ)。
データ実行防止 (DEP)	メモリ上のデータ領域に設置されたコードの実行を防止することで攻撃コードの実行を防ぐ。
アドレス空間配置のランダム化 (ASLR)	プロセスのモジュールレイアウト、スタック等のデータ領域のアドレスをランダム化することで攻撃コードの実行を困難にする。

### 3.1.2. 64bit版におけるDEPとASLRの拡張

64bit版の利用も重要な脆弱性攻撃対策となる点に注目する必要がある。Windows 8の64bit版では、カーネル

領域で大幅にDEP適用が拡張(図 4)されているほか、ハイエントロピーASLR(図 5)を導入している。

	x86 (PAE)		x64	
	Win7	Win8	Win7	Win8
Paged pool	X	X	NX	NX
Non-paged pool	X	X	X	X
Non-paged pool (NX)	N/A	NX	N/A	NX
Session pool	X	X	NX	NX
Image data sections	X	X	NX	NX
Kernel stacks	NX	NX	NX	NX
Idle/DPC/Initial stacks	X	NX	X	NX
Page table pages	X	NX	X	NX
PFN database	X	NX	X	NX
System cache	X	NX	X	NX
Shared user data	X	NX	X	NX
HAL heap	X	NX	X	NX

X = executable NX = non-executable

図 4 64 bitにおけるDEP拡張

領域別エントロピー (単位: ビット)	Windows 7		Windows 8		
	32 ビット	64 ビット	32 ビット	64 ビット	64 ビット (HE)
ボトムアップの割り当て (オプトイン)	0	0	8	8	24
スタック	14	14	17	17	33
ヒープ	5	5	8	8	24
トップダウンの割り当て (オプトイン)	0	0	8	17	17
PEB/TEB	4	4	8	17	17
EXE イメージ	8	8	8	17*	17*
DLL イメージ	8	8	8	19*	19*
非ASLR DLL イメージ (オプトイン)	0	0	8	8	24

\* ベース 4GB 未満の 64 ビット DLL は 14 ビット、4GB 未満の EXE には 8 ビットが適用される

図 5 ハイエントロピーASLRの効果

DEPの拡張、つまりNon-Executableの適用拡張により、攻撃者が攻撃コードを配置できる領域が更に制限される。このため攻撃者は悪意のあるコードを実行するためには、攻撃コード配置に関する新たな手法を用いる必要があり、脆弱性を使った悪意あるコード実行がより困難になった。

ハイエントロピーASLRとはWindows 8から搭載された、ASLRを拡張した機能である。ASLRはモジュールのメモリ上の配置をランダム化する機能であるが、そのアドレス配置をこれまでの8ビット（256通り）から、17ビット（約18万通り）～33ビット（約85億通り）に高めたもので、これにより攻撃を更に困難にする。

64bit化におけるユーザーランド側のメリットとして、ヒープスプレー攻撃が困難になっていることがあげられる。Windows 10 Mitigation Improvementsなどにあるように、近年のブラウザを狙った脆弱性攻撃ではヒープスプレー攻撃が使われることが主流となってきた。右図は32bitと64bitにおけるヒープスプレー攻撃の様子を示している。

ヒープスプレー攻撃とは、攻撃者がヒープメモリ上に大量のメモリを確保し、そこに攻撃コードを配置するという攻撃手法である。これにより攻撃対象コンピューターの正確なメモリレイアウトが分からなくても攻撃を成功させることができる。図 6にあるように、32bitでは

プロセスは2Gバイト分のアドレススペースしかないため、プロセスに割り当てられたメモリの大半をヒープスプレーデータで埋めることができる。これに対して64bitにおいては、1Tバイト分のメモリを確保したとしても、128Tバイトのアドレススペースのなかでカバーしている範囲は非常に限られていることから、ヒープスプレー攻撃を成功させることは困難となる。加えて、ヒープ領域のハイエントロピーASLR(24ビット)による、スタートアドレスの予測困難性も、ヒープスプレー攻撃をより難しいものにしてしている。

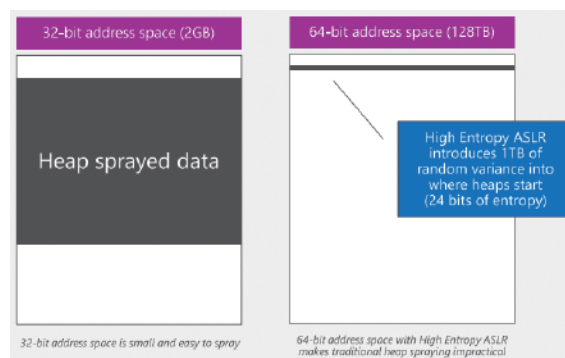


図 6 32bitと64bitのヒープスプレーの様子

### 3.1.3. 脆弱性攻撃対策の評価

Windows 10は、Windows 7と比較して多数の脆弱性攻撃対策機能が追加されているが、本項では、以下の機能について概要、メカニズム、効果について解説を行う。

- Supervisor Mode Execution Prevention (SMEP)
- Control Flow Guard (CFG)

## 3.2. Pass-the-Hash攻撃対策

単なるウィルス感染と考えられる事案が大きな問題として報道されるのは、一台のPCへの侵入を通じて、組織ネットワーク全体の制御が奪われ、機密情報の奪取やシステムの停止等の大きな被害につながっている

### 3.2.1. Pass-the-Hash攻撃とは

Pass-the-Hash攻撃とは資格情報を奪取する攻撃の一種であり、主に攻撃者が侵入した組織内ネットワークの侵害を拡大する目的で使われる。具体的にはパスワードそのものやパスワードをハッシュ化した情報を奪取する攻撃であり、Windowsの認証機能を提供するLSAプロセス内のメモリや、ディスク上(レジストリを含む)に保存された資格情報などのデータがターゲットとなる。この攻撃は、ローカル管理者権限を奪取した攻撃者が、他のコンピューターにログイン可能なユーザーアカウントの取得や、より高いドメイン管理者アカウントの取得を繰り返し、最終的にはシステム全体の管理権を奪取するために利用される。

より具体的なPass-the-Hash攻撃の実態およびその対策をまとめた資料として「Mitigating Pass-the-Hash (PtH) Attacks and Other Credential Theft, Version 1 and 2<sup>7</sup>」があるが、その中で紹介されている、典型的な攻撃を簡略化した図を以下に引用する。

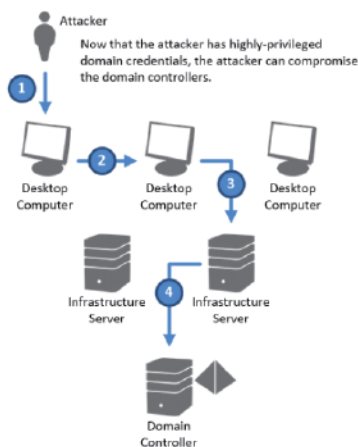


図7 攻撃者がシステム内を侵攻していく様子

上図の1のステップで、攻撃者は始めに攻略したコンピューター上でローカル管理者アカウントの権限を取得する。この初めのステップでは脆弱性攻撃やマルウェア

ためである。

このような攻撃の多くは Pass-the-Hashと呼ばれる攻撃手法を使用する。ここでは、現在のサイバー攻撃の鍵となる、Pass-the-Hash攻撃を取り上げる。

ア感染が使われる。その後に2や3にあるようなステップを踏みより高い権限の取得を試みる。特にステップ2でクライアントからクライアントへの水平方向へ攻撃者がログインしているのが特徴的であり、このような移動を行いながらシステムの探索行為が行われる。そして、その中でより高い権限を持つアカウントの認証情報を攻撃者に取得されると、ステップ3にあるような垂直方向のログインを許してしまう。攻撃者の垂直方向の動きも当然ながら、水平方向の動きに対しても注意する必要がある。このような動きの中で、他のコンピューターへログインする際に必要なアカウントの資格情報を、攻撃者はPass-the-Hash攻撃により奪取する。

実際のPass-the-Hash攻撃では、下図のようにコンピューターのメモリ上やファイルシステム上に保存されている資格情報が奪取される。これらの資格情報は例えばハッシュなどの形態であり、このハッシュが奪取されることでPass-the-Hash攻撃が行われる。奪取したハッシュは、他のコンピューターへのアクセスに使うこともできるし、またブルートフォース攻撃を行うことでパスワードを入手できる可能性がある。

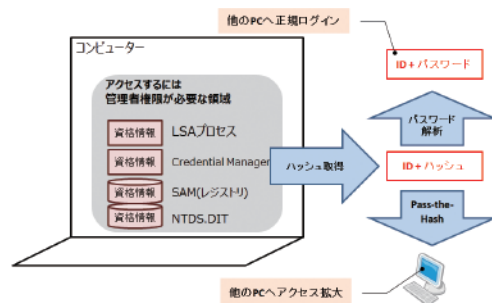


図8 Pass-the-Hashの仕組み

7 <https://www.microsoft.com/en-us/download/details.aspx?id=36036>

このメモリ上の資格情報が奪取されるケースでは、LSAプロセスというWindowsの認証などを行うプロセスが狙われる場合がある。LSAプロセスとはWindowsの中核となるプロセスであり、その性質上、当該コンピューターが持つローカルユーザーの資格情報や、ログオン中のユーザーの資格情報などを取り扱うため、そのメ

モリ内に資格情報を保持する必要がある。ファイルシステム上の資格情報が奪取されるケースでは、レジストリに保存されているローカルアカウントの資格情報や、キャッシュされたドメインユーザーの資格情報などが狙われる。

### 3.2.2. Pass-the-Hash攻撃対策

Windows 95で実装されたLMハッシュは、当時のコンピューターの処理能力を想定していることに加え、適切なハッシュ化が行われていないことが知られており、レインボーテーブルと呼ばれるデータベースを使って、瞬時にパスワードを解読できることが知られている。この問題の対策として、マイクロソフト社は、NTLMハッシュ、NTLM2ハッシュ、そしてKerberosの採用など、Pass-the-Hash攻撃対策を進め、Windows 8.1ではPass-the-Hash攻撃を困難にするアカウントグループを用意した(図9)。

Credential Guardは、このような背景を受けてWindows 10 EnterpriseおよびWindows Server 2016で導入された資格情報の盗難防止の機能である<sup>8</sup>。

Credential Guardは仮想化技術を用いてLSAプロセスを別のプロセスへ分離することにより、LSAプロセスに対するPass-the-Hash攻撃を防ぐ機能である。プロセスの分離は図10にあるようにWindows 10で導入されたVSM (Virtual Secure Mode)という仮想化の機能を使っている。なお、この機能によりLSAプロセス自体は防御されるものの、ファイルシステム上に保存されている資格情報は保護対象とならないことに留意する必要がある。実際のPass-the-Hash攻撃およびその対策については「Mitigating Pass-the-Hash (PtH) Attacks and Other Credential Theft, Version 1 and 2」にも詳しく記載されているので、併せてご覧いただきたい。

8 [https://technet.microsoft.com/ja-jp/library/mt483740\(v=vs.85\).aspx](https://technet.microsoft.com/ja-jp/library/mt483740(v=vs.85).aspx)

	Hashes		Tspkg	Wdigest	Kerberos	LiveSSP	3rdParty SSP
	LM	NT					
Windows 8.0							
Microsoft Account							
Local Account							
Domain Account							
Windows 8.1							
Microsoft Account			●	●			
Local Account			●	●			
Domain Account			●	●			
Protected Users							
RestrictedAdmin RDP							

● Off by default       No password data in memory  
 password data in memory

図9 Pass-the-Hash対策とグループ

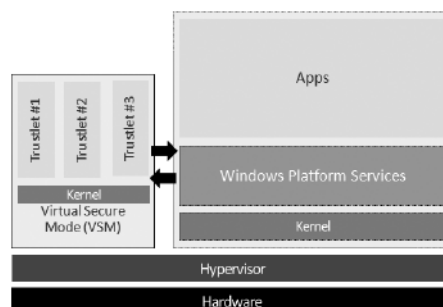


図10 Windows 10のVSMモデル



### 3.2.3. Pass-the-Hash攻撃対策の評価

Pass-the-Hash攻撃対策の機能を比較するため、以下の環境で実際の攻撃ツールを使って評価を行う。

1. Windows 7 Professional version 6.1.7601 (x86)
2. Windows 10 Enterprise version 10.0.14393 (x64)

Pass-the-Hash攻撃を行うツールとして以下の4つをそれぞれの環境で実行し、その結果を比較する。

- mimikatz<sup>9</sup>
- gsecdump<sup>10</sup>
- Pwdump7<sup>11</sup>
- QuarksPwDump<sup>12</sup>

それぞれのツールで実際に実行したコマンドとその期待する結果を以下に示す。

- 9 <https://github.com/gentilkiwi/mimikatz>
- 10 <http://download.openwall.net/pub/projects/john/contrib/win32/pwdump/gsecdump-0.7-win32.zip>
- 11 <http://download.openwall.net/pub/projects/john/contrib/win32/pwdump/pwdump7.zip>
- 12 <https://github.com/quarkslab/quarkspwdump>

表 2 調査対象のPass-the-Hashツール

ツール名	実際に実行したコマンド	期待する結果
Mimikatz	mimikatz.exe lsadump::cache	キャッシュされているユーザーの資格情報をディスク上から取得する。
gsecdump	Gsecdump -u -S	ログオン中のユーザーの資格情報をプロセスより取得する
Pwdump7	PwDump7.exe	ローカルユーザーの資格情報をディスク上から取得する。
QuarksPwDump	quarkspwdump -dhc	ローカルユーザーの資格情報をディスク上から取得する。

いずれもローカル管理者ユーザーで実行するが、mimikatzについては、更に高い権限を持つSystem権限での実行が前提となっているため事前に以下のようにpsexecコマンド<sup>13</sup>を実行しSystem権限を取得した上で実行した。

```
> psexec.exe -i -s \\127.0.0.1 cmd.exe
```

評価環境のWindows 10についてはCredential Guardを有効にした。以下にmsinfo32.exeの出力結果を示す。

なお、同様の評価としてJPCERT/CCの「Windowsの新セキュリティ機能を検証する:LSAの保護モードとCredential Guard」<sup>14</sup>がある。JPCERT/CCの評価ではWindows 10のビルド10586を使っているが、本評価ではAnniversary Update後のビルド14393を使っている。また、採用したツールは同じものであるが、コマンドライン引数は独自に選択をしている。

13 <https://technet.microsoft.com/ja-jp/sysinternals/pxexec.aspx>  
 14 [https://www.jpCERT.or.jp/magazine/acreport-lsa\\_protect.html](https://www.jpCERT.or.jp/magazine/acreport-lsa_protect.html)

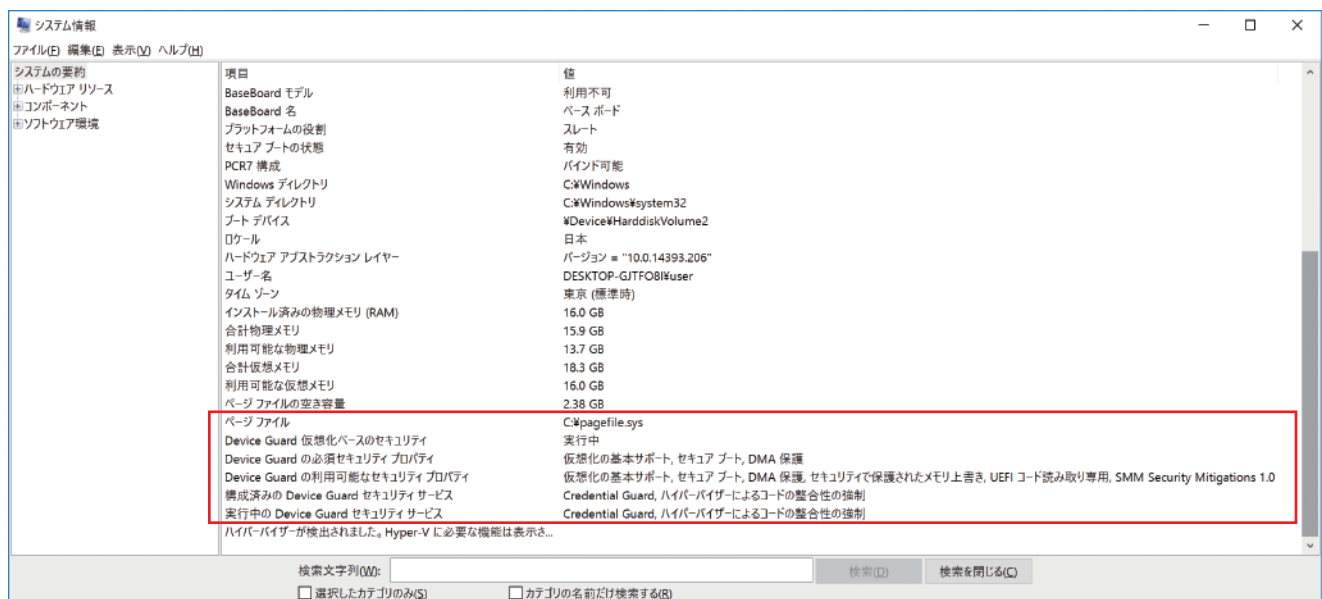


図 11 msinfo32.exeの出力

ここまで、Windows 8.1までに実装されたセキュリティへの取り組みについて解説をした。ここでは脆弱性対策としてSMEPとCFGについて解説をし、Path-The-Hash対策として実装されたCredential Guardに対する関連攻撃ツールの実行結果を解説する。

## 4.1. 脆弱性攻撃対策

Windows 10は、Windows 7に比べて多数の脆弱性攻撃対策機能が追加されているが、これらを回避する攻撃も少なからず確認されている。例えば、Pass-the-Hash攻撃等では、認証情報を取得するために、脆弱性を利用したシステム特権の奪取が試みられる。SMEPは、このような特権を奪取する攻撃の対策としてWindows 8で実装された技術である。また、DEPとASLRの導入が進んだことから、単純な手法ではデータセグメント・スタックセグメント上の攻撃コードの実行

が難しくなっているが、これを回避する手法としてROP (Return Oriented Program)が利用される。CFGはROP対策としてWindows 8.1 Previewで試験的に導入され、Windows 10において標準搭載された技術である。本稿では、SMEPとCFGについて、概要、メカニズム、効果について解説を行う。

- Supervisor Mode Execution Prevention(SMEP)
- Control Flow Guard (CFG)

### 4.1.1. Supervisor Mode Execution Prevention (SMEP)

近年の標的型攻撃では、Internet Explorer、Microsoft Office、Adobe Flash Player等のクライアントソフトウェアの脆弱性と、Windowsカーネルの脆弱性が併用される事例<sup>15</sup>が少なくない。

カーネル及びその関連コンポーネント等、OSと同じ特権で動作するソフトウェアの脆弱性が攻撃に悪用された場合、ログオンユーザーの権限やポリシー制御等のセキュリティ対策に関わらず、特権を奪われ、システムを掌握されるため、一般アプリケーションに対する脆弱性攻撃よりも深刻度が高い。Windows 8以降のWindowsでは、カーネルに対する攻撃の対策として、Intelの Ivy Bridge プロセッサで導入されたSMEPをサポートしている。

カーネルの脆弱性を悪用して任意コードの実行を行う際に取られる最も単純な手法は、悪意のコードをユーザー空間のメモリ上に配置し、カーネルの脆弱性を悪用してユーザー空間のコードを実行させる方法である。通常、各プロセスのコード、データ等はユーザー空間のメモリ上に配置され、カーネル及びその関連コンポーネントのコード、データ等は、カーネル空間のメモリ上に配置される。従来のWindowsにおいては、メモリ保護の仕組み上、アプリケーション等のユーザー権限で動作するプロセスからカーネル空間のメモリにはアクセスすることができず、

アクセスを試みた場合、一般保護例外が発生する。一方、特権モードで動作する処理(プロセス等)からユーザー空間のメモリ上にアクセスすることが可能であり、ユーザー空間のコードを実行することも可能であった。上記の攻撃は、この仕様を突いた手法である。

SMEPは、OSがCPUの特定のレジスタの値をセットすることで特権モードの処理実行時のユーザー空間のメモリ上のコード実行を禁止し、上記の攻撃手法を防止する。実際の攻撃に対するSMEPの効果については、マイクロソフト セキュリティインテリジェンスレポート(第20版)<sup>16</sup>でも紹介されている。その記載によるとPLATINUMという攻撃で使われたマルウェアのうちのひとつは、「Windows 10の環境では最新のパッチを適用していなくても攻撃は成立しなかったはずである」とある。PLATINUMは南アジアおよび東南アジアを狙った標的型攻撃であり、2009年から続いている攻撃とされている。そのうち2015年夏に使われたものがカーネルの脆弱性(CVE-2015-2546)を使ったもので、SMEPで防御するこ

15 <https://www.fireeye.com/content/dam/fireeye-www/blog/pdfs/twoforonefinal.pdf>

16 <https://www.microsoft.com/ja-jp/safety/resources/sir.aspx>

とができる。SMEPが有効なシステムで、この攻撃を成立させるにはSMEPを回避する必要がある。また、トレンドマイクロ セキュリティブログの「Windows カーネルモードの脆弱性「CVE-2014-4113」を検証<sup>17)</sup>」においても、当時のCVE-2014-4113の脆弱性を使った脆弱性攻撃をSMEPが防御しているという分析結果が報告されている。

SMEPを回避する方法<sup>18)</sup>も研究されているものの、攻撃

者に対して攻撃成立のために更なる技術的障壁となっていることは確かであり、上記のような実績からもSMEPはWindowsセキュリティレベルの向上に貢献しているといえる。

17 <http://blog.trendmicro.co.jp/archives/10140>

18 <https://www.coresecurity.com/corelabs-research/publications/windows-smep-bypass-us>

### 4.1.2. Control Flow Guard (CFG)

Control Flow Guard (CFG)は、Windows 8.1 Previewで試験的に導入され、その後、Windows 10においては標準搭載されたセキュリティ機能である。

プログラム実行時の関数呼び出しは、呼び出し先関数のアドレスを指定する直接呼び出しと、レジスタ等を介してアドレスを指定する間接呼び出しがある。近年の攻撃では、間接呼び出しを悪用することが増えている。典型的な手法としては、C++などの言語でクラスとして実装される仮想関数のポインターを書き換えることで攻撃コードの実行を試みる。Windows 10では、このような攻撃に対する緩和策としてCFGを実装した。

#### ■ 直接呼び出しの例

```
CALL DWORD PTR ds:0x12345678
```

#### ■ 間接呼び出しの例

```
CALL EAX
```

ヒープスプレー等の脆弱性攻撃では、脆弱性を突いて攻撃対象プロセスのメモリを破壊することで特定のレジスタの値を任意の値に操作する。上記のようにCALL EAXと言う命令が実行される前にEAXレジスタの値を操作し、攻撃コードが配置されたメモリ上のアドレスを指定することで攻撃コードを呼び出すことが可能となる。

CFGは、こうした攻撃に対処するためにプログラムのビルド時に間接的な関数呼び出しを抽出し、関数呼び出しの前にチェック関数の呼び出しを挿入する。また、プログラム中に含まれる間接関数呼び出しの情報を実行プログラムに埋め込む。チェック関数は、プログラム実行時に埋め込

まれた情報と呼び出す関数のアドレスを比較することで関数が信頼できるものか否かを確認し、信頼できないアドレスに対する間接関数呼び出しを検知した際に例外を発生させる。

下記の環境で実際に間接関数呼び出しが発生するサンプルプログラムをCFGが有効になる状態でビルドし、実際の動作を確認した。

#### ● Visual Studio 2015 Professional

- ▶ 最適化無効(/Od)
- ▶ C/C++ 追加のコマンドラインオプション /d2guard4
- ▶ リンカー追加のコマンドラインオプション /guard:cf

```
#include "stdafx.h"
void myfunc(void)
{
    printf("Hello world!\n");
}
int main()
{
    void(*fptr)() = NULL;

    fptr = myfunc;
    fptr();

    return 0;
}
```

上記のプログラムはmain関数内からmyfunc関数を呼び出す際に、関数ポインターfptr経由で呼び出しを行っており、最適化を無効にした状態でコンパイルを行うと、以

下のような間接関数呼び出し (call qword ptr [rsp+28h]) を行う機械語コードが生成される。

```

int main()
{
0000000013F691020 sub    rsp,38h
    void(*fptr)() = NULL;
0000000013F691024 mov    qword ptr [fptr],0

    fptr = myfunc;
0000000013F69102D lea   rax,[myfunc (013F691000h)]
0000000013F691034 mov    qword ptr [fptr],rax
    fptr();
0000000013F691039 mov    rax,qword ptr [fptr]
0000000013F69103E mov    qword ptr [rsp+28h],rax
0000000013F691043 mov    rcx,qword ptr [rsp+28h]
0000000013F691048 call  _guard_check_icall (013F691BFCh)
0000000013F69104D call  qword ptr [rsp+28h]

    return 0;
0000000013F691051 xor    eax,eax
}
0000000013F691053 add    rsp,38h
0000000013F691057 ret

```

また、前述の追加のコマンドラインオプションを指定することでCFGのチェック関数の呼び出しが間接関数呼び出しの直前に追加されていることが確認できる (call \_\_guard\_check\_icall)。\_\_guard\_check\_icall関数内部では、前述のビルド時に埋め込まれた間接関数呼び出しの情報とrcxレジスタ経由で渡された呼び出し先のアドレスを比較することで呼び出し先の関数が信頼できる正規の処理か否かの確認を行う。

このように、CFGはプログラムのビルド時点で正解情報を作成し、実行時に正解情報との比較検証を行うという点で前述の安全な例外ハンドラーのあるイメージ (/SAFESEH) と同様のアプローチであり、提供される保護機能を回避することが困難な機能だと言える。但し、その効果はあくまで間接関数呼び出しに限定されており、間接ジャンプ命令やreturn命令によって制御が奪われるケースでは効果がない点に留意する必要がある。

## 4.2. Pass-the-Hash攻撃対策

Pass-the-Hash攻撃対策の機能を比較するため、以下の環境で実際の攻撃ツールを使って評価を行った。

### 評価を実施したシステム

- ① Windows 7 Professional version 6.1.7601 (x86)
- ② Windows 10 Enterprise version 10.0.14393 (x64)

### 評価を実施したツール

- mimikatz
- gsecdump
- Pwdump7
- QuarksPwDump

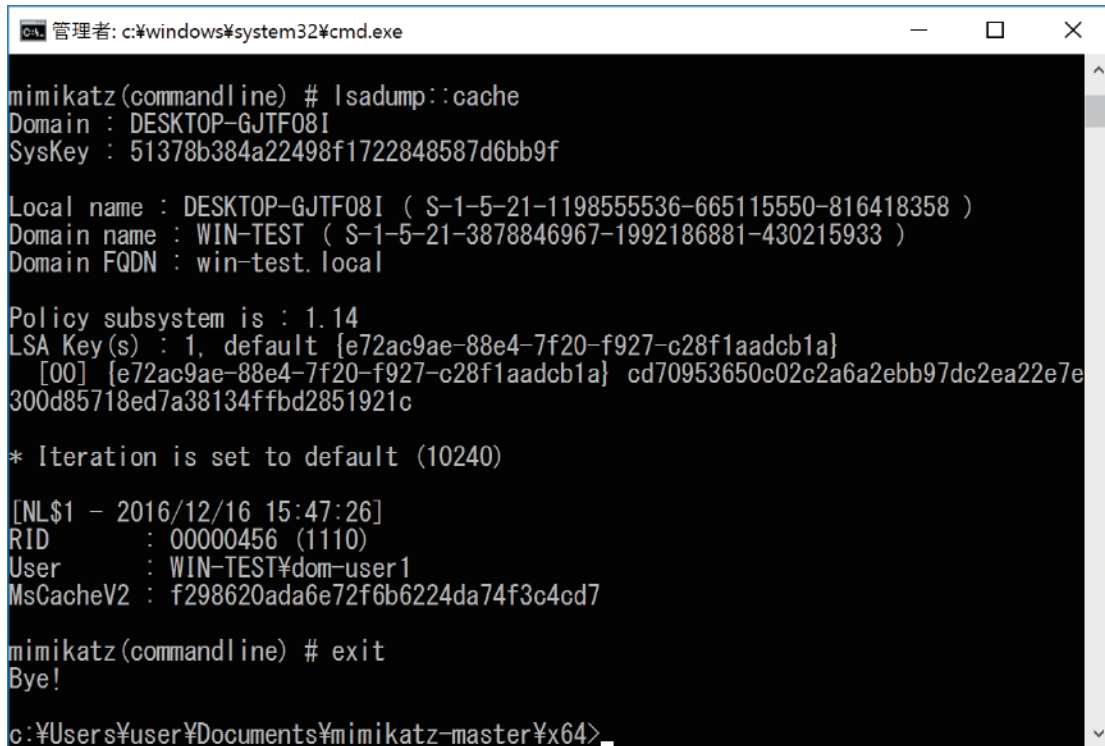
表 3 Pass-the-Hash攻撃対策の評価結果

ツール名	Windows 7	Windows 10	Windows 10 + Restricted Admin mode
Mimikatz	成功	成功	失敗
gsecdump	成功	失敗	失敗
Pwdump7	成功	失敗	失敗
QuarksPwDump	成功	失敗	失敗

### 4.2.1. Mimikatz

Windows 10においてもmimikatzの実行が成功しているが、これはmimikatzがCredential Guardが保護しないディスク(レジストリ)から資格情報を読み込んで

いるからである。Windows 10上での実行結果を以下に示す。



```
管理者: c:\windows\system32\cmd.exe
mimikatz(commandline) # lsadump::cache
Domain : DESKTOP-GJTF08I
SysKey : 51378b384a22498f1722848587d6bb9f

Local name : DESKTOP-GJTF08I ( S-1-5-21-1198555536-665115550-816418358 )
Domain name : WIN-TEST ( S-1-5-21-3878846967-1992186881-430215933 )
Domain FQDN : win-test.local

Policy subsystem is : 1.14
LSA Key(s) : 1, default {e72ac9ae-88e4-7f20-f927-c28f1aadcb1a}
[00] {e72ac9ae-88e4-7f20-f927-c28f1aadcb1a} cd70953650c02c2a6a2ebb97dc2ea22e7e300d85718ed7a38134ffbd2851921c

* Iteration is set to default (10240)

[NL$1 - 2016/12/16 15:47:26]
RID      : 00000456 (1110)
User     : WIN-TEST\dom-user1
MsCacheV2 : f298620ada6e72f6b6224da74f3c4cd7

mimikatz(commandline) # exit
Bye!
c:\Users\user\Documents\mimikatz-master\64>
```

図 12 mimikatz実行結果 (Windows10)

この出力結果より、WIN-TESTドメインのdom-user1ユーザーのハッシュが取得できることが分かった。このユーザーは事前に当該コンピューターにログインしたドメインユーザーである。Windowsは標準で、一度ログインしたユーザーの資格情報をキャッシュするようになっている。このキャッシュ機能により、オフライン状態でのドメインユーザーでのログインが可能となる。この結果から分かるように、一度、ログインしたユーザーの資格情報はキャッシュとしてディスクに保存されるため、Credential Guardが有効でも、Pass-the-Hash攻撃により奪取される可能性がある。

Windows 8.1以降では、「Mitigating Pass-the-Hash (PtH) Attacks and Other Credential Theft, Version 2<sup>19</sup>」でも推奨されているRestricted Adminモード等の

Pass-the-Hash攻撃対策を目的としたアカウントが追加されている(図 9)。Restricted Adminモードが有効になっているマシンに対してドメイン管理者のアカウントでリモートデスクトップで接続しても、そのログインしたマシン上でドメイン管理者の資格情報がキャッシュされない。以下のスクリーンショットでは、192.168.11.3のマシンに対してWIN-TEST\dom-admin1というドメイン管理者のアカウントでログインをしている。whoamiコマンドによりアカウントはWIN-TEST\dom-admin1であることが分かるが、手前のcmd.exe画面上で実行されているmimikatzではキャッシュされた資格情報を取ることができていないのが分かる。

19 <https://www.microsoft.com/en-us/download/details.aspx?id=36036>

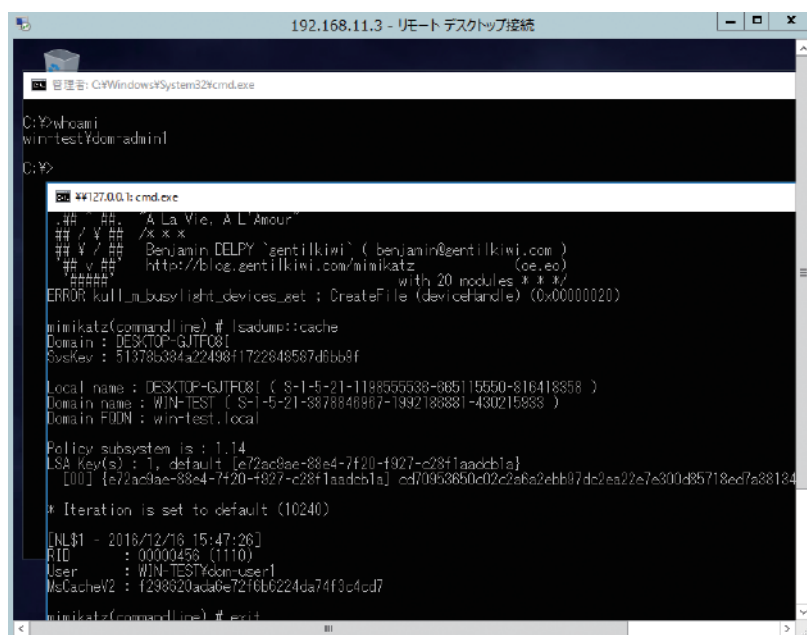


図 12 mimikatz実行結果 (Windows10)

この対策により、ドメイン管理者のアカウントでマルウェア等に攻略されてしまったマシンに対してリモートデスクトップでログインしても、キャッシュされた資格情報を取得される恐れがない。注意点としては、この対策を適用するにあたっての環境や構成に制限がある点と、この対策により発生する新しいリスクがある点が

あげられる。新しいリスクとしては、この対策を有効にしたマシンへはパスワードを知らなくても Pass-the-Hash攻撃で入手したハッシュだけでリモートデスクトップ接続ができるようになってしまう点があげられる。そのため、この対策の採用にはシステム全体の構成や設計を含めた検討が必要である。

#### 4.2.2. gsecdump

gsecdumpの実行結果を以下に示す。

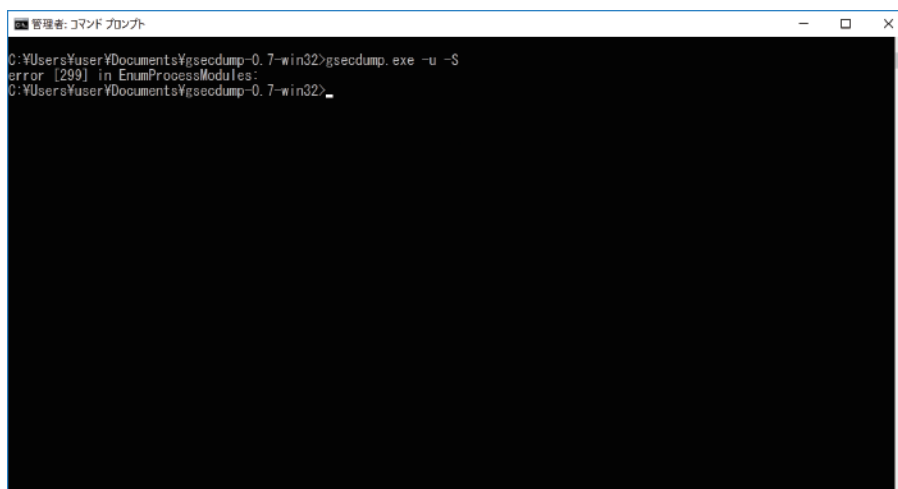
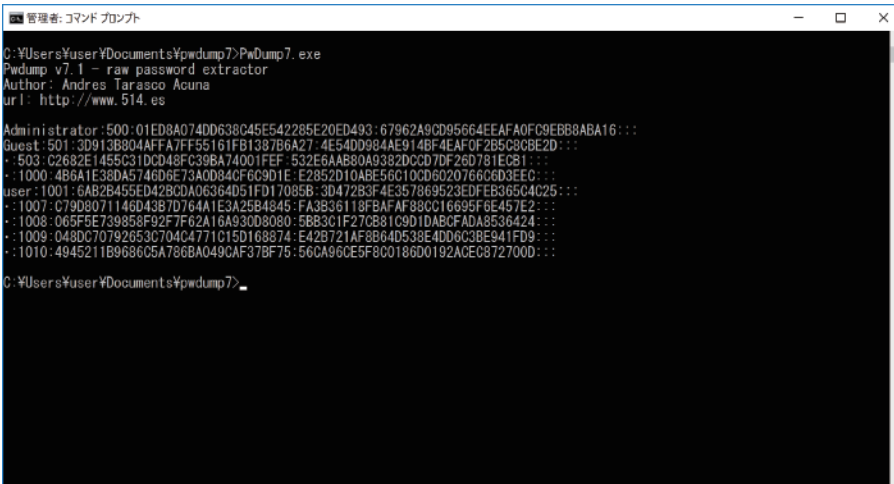


図 14 gsecdumpの実行結果 (Windows 10)

gsecdumpはCredential Guardが想定する通りの認証プロセスから資格情報を取得するタイプのツールである。このため、Credential Guardが有効なWindows 10において実行が失敗していると考えられる。これにより、Credential Guardは認証プロセスから資格情報を奪取するような攻撃に有効であることが分かった。

### 4.2.3. Pwdump7

Pwdump7の実行結果を以下に示す。



```
管理: コマンド プロンプト
C:\Users\User\Documents>pwdump7>PdDump7.exe
PdDump v7.1 - raw password extractor
Author: Andres Tarasco Acuna
url: http://www.514.es

Administrator:500:01ED8A074DD638C45E542285E20ED493:67962A9CD95664EEAFA0FC9EBB8ABA16:::
Guest:501:3D913B804AFFA7FF55161FB1387B6A27:4E54D9984AE914BF4EAF0F2D5C80BE2D:::
- 503: C2682E14E5931D0D48F938B47A001FEF:532E6A880A9382DCC07DF260781E081:::
- 1000: 486A1E38DA5746D6E73A0D24CF6C9D1E:E2852D10ABE56C10CD60207666D3EEC:::
user:1001:6A82B455ED428CD406364D51FD17085B:3D472B3F4E357869523EDFE8365C4C25:::
- 1007: C79D8071146D4387D764A1E3A25B4845:FA3836119FBFAFAF88CC16695F6E457E2:::
- 1008: 065F5E739858F92F7F62A16A930D8080:5B83C1F27C881C9D1DABCFADA8536424:::
- 1009: 048DC70792653C704C4771C15D168874:E42B721AF8B64D538E4DD6C3BE941FD9:::
- 1010: 4945211B9686C5A786BA049CAF37BF75:56CA96CE5F8C0186D0192AC0C872700D:::

C:\Users\User\Documents>pwdump7>
```

図 15 pwdump7実行結果 (Windows 10)

実行結果としてハッシュが取得できているように見えるが、これらは正しいハッシュではないために実行に失敗していると言える。Pwdump7は資格情報をディスクから取得するタイプのツールであるためハッシュが取得できているが、ディスクから取得した資格情報を正しく扱うことができていないため、誤ったハッシュを表示していると考えられる。これはCredential Guardによるものではなく、Pwdump7のWindows 10 Anniversary Update未対応による実行失敗と考えられる。

Windows 10 Anniversary Updateからディスク(レジストリ)に保存されている資格情報の暗号方式が変更された<sup>20</sup>。このためディスクから資格情報を奪取するタイプのツールは、この新方式に対応しないと資格情報を正しく読み込むことができない。なお、mimikatzの実行は成功しているが、これはWindows 10 Anniversary

Updateの新方式に対応しているためである。旧方式は、暗号処理の中でRC4という暗号を使うが、これは既に使わないようにするRFC7465<sup>21</sup>などが公開されており廃止に向かって進んでいる。今回の暗号方式の変更はこのような流れの中での仕様変更だと思われる。

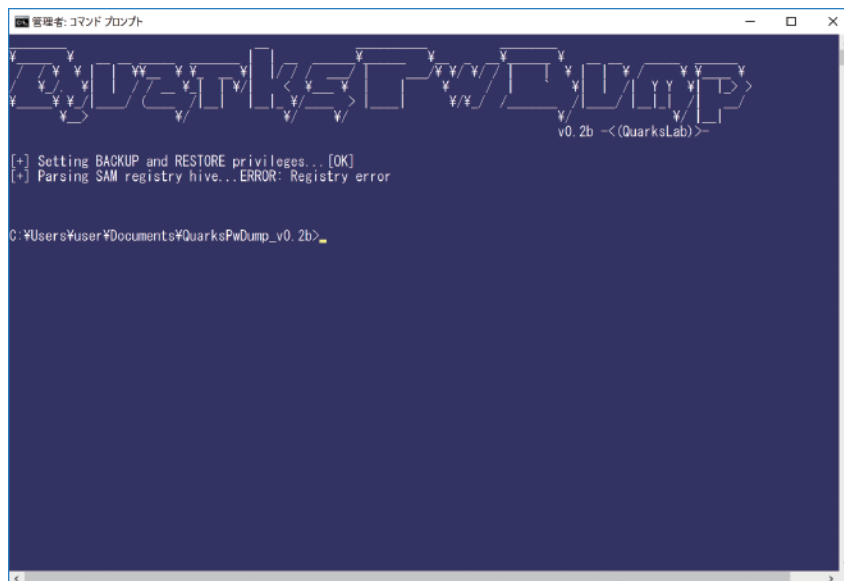
20 <http://www.passcape.com/index.php?section=blog&cmd=details&id=35>

21 <https://tools.ietf.org/html/rfc7465>



#### 4.2.4. QuarksPwDump

QuarksPwDumpの実行結果を以下に示す。



```
管理者: コマンド プロンプト
QuarksPwDump
v0.2b -<(QuarksLab)>
[+] Setting BACKUP and RESTORE privileges... [OK]
[+] Parsing SAM registry hive... ERROR: Registry error
C:\Users\User\Documents\QuarksPwDump_v0.2b>
```

図 16 QuarksPwDump実行結果

QuarksPwDumpもPwDump7と同様にディスクから資格情報を取得するタイプのツールであるが、レジストリ値の読み込みに失敗して実行を完了することができなかった。理由はPwDump7と同様にWindows 10 Anniversary Updateに対応していないためである。

# 5

## むすび：Windows 10を利用する上での提言

Windows 10は、Windows 7と比較して格段にセキュリティレベルが向上しているものの、これを破るための攻撃手法も継続的に開発されている。本節では、前述の評価結果を踏まえてWindows 10を利用する上で、安全性を高めるために考慮すべき事項について記載する。

### 5.1. 脆弱性攻撃対策

Windows 10における脆弱性対策については以下のフレームワークに集約することができる。

- (ア) セキュリティレベルの向上が確認された最新のOS/アプリケーションを利用する
- (イ) 全てのPC・サーバーに対して適切な設定を適用する
- (ウ) 脆弱性を排除するためのセキュリティ更新を確実に実施する
- (エ) 常に対策状況を把握し、新たに公表される脆弱性を評価し対応する
- (オ) 攻撃が成功することを前提とした検知・対応の仕組みを構築する

前述の通りWindows 10は新たな防御機能を搭載しており、脆弱性攻撃を成功させるための技術障壁は格段に高くなっているが、設定によってはWindows 10のセキュリティレベルを十分に活かすことができない。また、Phase 1の報告書でも記載した通り、脆弱性攻撃技術と

その対策技術はイタチごっこの関係にあり、これを破る攻撃も想定しなければならない<sup>22</sup>ことから、Windows 10においても、更新プログラムによる脆弱性の排除は、重要なセキュリティ対策となっている。

本稿で述べた対策の実装や更新プログラムの適用は、組織内のすべてのPCに対して確実に適用することが重要であることから、マイクロソフトが提供するActive Directoryのグループポリシーを利用したPCやサーバー管理の重要性が高まっている。セキュリティの対策状況を把握するためにも、PCや端末、その他のデバイスを包括的に管理するシステムを構築することが重要となっている。そして、常に状況を把握し、新たな脆弱性が公表された場合は適切な対策を取るようにする。そして、攻撃が成功する前提で、攻撃検知を行い対策を実施する仕組みを構築し運用する。

22 <https://www.coresecurity.com/system/files/publications/2016/05/Windows%20SMEP%20bypass%20U%3DS.pdf>

### 5.2. Pass-the-Hash攻撃対策

Windows 10においては、Credential GuardとRestricted Administratorを利用することで、代表的なツールによるPath-the-Hash攻撃を防げることが分かった。

一方で、Credential Guardはgsecdump等のLSAプロセスから資格情報を奪取するタイプのツールの実行を阻止することができることが確認できたが、ディスク上に保存されている資格情報の保護には課題が残った。これらの点を考慮した、Pass-the-Hash攻撃対策のフレームワークとして以下の対策を挙げることができる。

1. 原則としてユーザー権限のアカウントを利用する
2. 管理者権限が必要な際は、管理者グループ等を利用する（ビルドインアドミニストレータを利用しない）
3. Windows 10でCredential Guardを利用する
4. Restricted Administratorを活用する
5. ドメインユーザーの資格情報をキャッシュする数を制限する<sup>23</sup>
6. ローカルユーザーとドメインユーザーに同じパスワードを用いない
7. キットティング用のアカウントのパスワードをユニークにする（LAPSの利用）

23 [https://technet.microsoft.com/ja-jp/library/mt629048\(v=vs.85\).aspx](https://technet.microsoft.com/ja-jp/library/mt629048(v=vs.85).aspx)

前述のように、Credential Guardにより、LSAプロセス上の資格情報を保護することができるが、ディスク上に記録されるキャッシュやローカルユーザーの資格情報を保護することはできない。キャッシュされている資格情報にはドメインユーザーも含まれているため、こういったアカウントの資格情報が奪取されると攻撃者は他のコンピューターへログオンすることができるようになってしまう。これを避けるためには、ドメインユーザーの資格情報をキャッシュする数を制限<sup>24</sup>することが有効である。特に常にネットワークに接続しているコンピューターについては、資格情報をキャッシュしない設定にすることでこの攻撃を防ぐことができる。

ローカルユーザーの資格情報もディスク上に保存されているためCredential Guardの保護対象外である。ただし、PwDump7とQuarksPwDumpはWindows 10 Anniversary Updateに対応しておらず正しく動作しなかった。しかし、例えばmimikatzのようなWindows 10 Anniversary Updateに対応した攻撃ツールに対しては有効な対策が存在しないために、資格情報の盗難を前提とした運用が必要である。例えば、ローカルユーザーとドメインユーザーに同じパスワードを用いないことで、ローカルユーザーに対するPass-the-Hash攻撃が成功しても他のコンピューターに被害が広がらないようにするという対応が考えられる。

また、前述のように「Mitigating Pass-the-Hash (PtH) Attacks and Other Credential Theft, Version 2<sup>25</sup>」の35ページに記載のRestricted Adminモードを活用する対策も考えられる。Restricted Adminモードが有効になっているマシンに対してドメイン管理者のアカウントでリモートデスクトップをしても、そのログインしたマシン上でドメイン管理者の資格情報がキャッシュされない。

注意点としては、この対策を適用するにあたっての環境や構成に制限がある点と、この対策により発生する新しいリスクがある点があげられる。新しいリスクとしては、この対策を有効にしたマシンへはパスワードを知らなくてもPass-the-Hash攻撃で入手したハッシュだけでリモートデスクトップできるようになってしまう点があげられる。そのため、この対策の採用にはシステム全体の構成や設計を含めた検討が必要である。

また、パスワード管理上の問題としてローカル管理者のパスワードとして同じパスワードを多くのコンピューターで使ってしまうケースが考えられる。これは多数のコンピューターを同時にキッティングするような大

規模環境においてなど見られる。このような場合では、1台のコンピューターからローカル管理者のパスワードが盗まれると、仮にドメイン管理者のパスワードを別にしていたとしても、他のコンピューターへは各コンピューターのローカル管理者でログインをされてしまう。このような問題を解決する方法として、マイクロソフト社はLAPS<sup>26</sup>という無償のツールを提供している。LAPSを使うことにより各コンピューターのローカル管理者の管理をAD(Active Directory)で行うことができる。

24 [https://technet.microsoft.com/ja-jp/library/mt629048\(v=vs.85\).aspx](https://technet.microsoft.com/ja-jp/library/mt629048(v=vs.85).aspx)

25 <https://www.microsoft.com/en-us/download/details.aspx?id=36036>

26 <https://blogs.technet.microsoft.com/jpsecurity/2015/05/14/local-administrator-password-solution-laps/>

これらの対策を纏めた表を以下に示す。

表 4 Pass-the-Hash 対策まとめ

資格情報保存先	ユーザーの種類	対策
LSAプロセス内メモリ	ローカルユーザー、 ドメインユーザー	LSAプロセスより資格情報を奪取する攻撃には Credential Guardが有効である。
ディスク内(レジストリ等)	ローカルユーザー	<ul style="list-style-type: none"><li>● Windows 10 Anniversary Updateの適用により、一部のツールの実行を防ぐことができる。</li><li>● ローカルユーザーはドメインユーザーと同じパスワードを使用しない(ローカルユーザーのパスワード悪用を防ぐ)。</li><li>● LAPSなどを活用し、ローカル管理者のパスワードに共通パスワードを使わないといった管理を行う。</li></ul>
ディスク内(レジストリ等)	ドメインユーザー	<ul style="list-style-type: none"><li>● Windows 10 Anniversary Updateの適用により、一部のツールの実行を防ぐことができる。</li><li>● ドメインユーザーをキャッシュする数を制限する。</li><li>● Restricted Adminモードを使用する。</li></ul>



## Windows10セキュリティ評価支援（Phase2）報告書

---

2017-3-23初版



株式会社 FFRI

お問い合わせ先

株式会社 FFRI 経営管理本部 経営企画部 IR広報担当

TEL : 03-6277-1811 E-Mail : [pr@ffri.jp](mailto:pr@ffri.jp)

URL : <http://www.ffri.jp>