セキュアマルウェア分析アプライアンスのエアギャップモードの更新

内容

はじめに

このドキュメントでは、Secure Malware Analyticsアプライアンスのエアギャップモードを更新する手順について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- WindowsおよびUNIX/Linux環境でのコマンドラインによる入力の基礎知識
- マルウェア分析アプライアンスに関する知識

- Cisco Integrated Management Controller(IMC)の知識

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- Windows 10およびCentOS-8
- ルファス2.17
- C220 M4

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

ほとんどのセキュアマルウェア分析アプライアンスはインターネットに接続されているため、オンラインアップデートプロセスを使用します。ただし、一部のセキュアマルウェア分析アプライアンスは、内部ネットワーク内で厳密に維持されます。つまり、「エアギャップ」が発生します。アプライアンスを「エアギャップ」状態にしておくことは、アプライアンスの効果を下げることになるため、推奨しません。ただし、このトレードオフは、追加のセキュリティ要件や規制要件をサポートするために必要になる場合があります。

インターネットに接続していない状態でSecure Malware Analyticsアプライアンスを実行するユーザに対しては、このドキュメントで説明するオフライン更新プロセスを提供します。アップデートメディアは、要求に応じてSecure Malware Analyticsサポートから提供されます。詳細については、以下を参照してください。

メディア：Secure Malware Analyticsサポートは、Airgap（オフライン）アップデートメディアをISOとして提供します。適切なサイズのUSBメディアまたはHDD（ハードディスクドライブ）にコピーできます。

サイズ：サイズはアップデートメディアがサポートするバージョンによって異なりますが、多くの場合、ソースリリースと宛先リリースの間に新しいVMが導入されると、数十GBになる場合があります。現在のリリースでは、非同期ツールがVM関連の変更の差分更新に役立つため、およそ30 GBになる可能性があります。

アップグレードブートサイクル： airgapアップデートメディアがブートされるたびに、アップグレード先の次のリリースが判別され、その次のリリースに関連付けられたコンテンツがアプライアンスにコピーされます。アプライアンスの実行中に実行する必要がある前提条件チェックがリリースに存在しない場合、特定のリリースでもパッケージのインストールを開始できます。リリースに、このようなチェックや、このようなチェックを追加する可能性がある更新プロセスの一部に対するオーバーライドが含まれている場合、ユーザがOpAdminにログインしてOpAdmin > Operations > Update Applianceで更新を呼び出さない限り、実際には更新が適用されません。

インストール前フック：その特定のアップグレードにインストール前フックが存在するかどうか

に応じて、アップグレードが即座に実行されるか、アプライアンスが通常の操作モードに戻されます。これにより、ユーザーは通常の管理インターフェイスを使用して、手動でアップグレードを開始できます。

必要に応じて繰り返し：各メディアのブートサイクルでは、最終的なターゲットリリースへの1ステップのみがアップグレード（またはアップグレードの準備）されます。ユーザは、目的の宛先リリースにアップグレードするために必要な回数だけブートする必要があります。

制限事項

エアギャップ更新ではCIMCメディアはサポートされていません。

使用されるサードパーティコンポーネントのライセンスの制約により、UCS M3ハードウェアがEOL（サポート終了）を迎えると、1.xリリース用のアップグレードメディアは使用できなくなります。したがって、UCS M3アプライアンスは、EOLの前に交換またはアップグレードすることが重要です。

要件

移行：対象リリースのリリースノートに、次のバージョンがインストールされる前に移行が必須であるシナリオが含まれている場合、アプライアンスが使用不能状態にならないように、再起動の前にこれらの手順に従う必要があります。

注：特に、2.1.4よりも新しい最初の2.1.xリリースでは、複数のデータベース移行が実行されます。これらの移行が完了するまで続行することは危険です。詳細については、『Threat Gridアプライアンス2.1.5移行ガイド』を参照してください。

2.1.3より前のリリースからairgapアップグレードメディアを使用する場合、個々のライセンスから取得した暗号キーが使用されるため、アプライアンスごとにカスタマイズする必要があります。（2.1.3より前のオリジンバージョンをサポートするように構築されたメディアの場合、Secure Malware Analyticsではこれらのアプライアンスに事前にライセンスをインストールしておく必要があり、そのメディアが構築されたリストにないアプライアンスでは動作しません）。

リリース2.1.3以降の場合、airgapメディアは汎用であり、お客様情報は必要ありません。

はじめる前に

- バックアップ.更新を続行する前に、アプライアンスのバックアップを検討する必要があります。
- 新しいリリースへの更新を計画する前に、更新するリリースのリリースノートを参照して、バックグラウンドでの移行が必要かどうかを確認します
- アプライアンスの現在のバージョンを確認します： OpAdmin > Operations > Update Appliance
- すべてのThreat Gridアプライアンスドキュメント(リリースノート、移行ノート、セットアップと設定ガイド、管理者ガイド)に記載されているビルド番号/バージョン参照テーブルでSecure Malware Analyticsアプライアンスのバージョン履歴を確認します。

オフライン（エアギャップ）のSecure Malware Analyticsアプライアンスの更新

最初に、このページで利用可能なエアギャップのバージョンを確認します：アプライアンスバージョンのルックアップテーブル

1. TACサポートリクエストをオープンし、オフラインアップデートメディアを入手します。この要求には、アプライアンスのシリアル番号とアプライアンスのビルド番号が含まれている必要があります。

2. TACサポートは、インストールに基づいて更新されたISOを提供します。

3. ISOイメージをブータブルUSBに書き込みます。USBは、オフライン更新がサポートされている唯一のデバイス/方法であることに注意してください。

命名規則

更新されたファイル名は次のとおりです。TGA Airgap Update 2.13.2-2.14.0

つまり、このメディアは最小バージョン2.13.2を実行しているアプライアンスに使用でき、アプライアンスをバージョン2.14.0にアップグレードできます。

制限事項

- エアギャップ更新ではCIMCメディアはサポートされていません。
- 使用されているサードパーティコンポーネントのライセンスの制約により、1.xリリースのアップグレードメディアは、UCS M3ハードウェアがサポート終了(EOL)を迎えた後は使用できなくなります。したがって、UCS M3アプライアンスは、EOLの前に交換またはアップグレードすることが重要です。

Linux/MAC - ISOダウンロード

要件

次の項目に関する知識があることが推奨されます。

- ISOをダウンロードし、起動可能なUSBインストールドライブを作成するためのインターネットアクセスを備えたLinuxマシン。
- Airgapダウンロード手順は、Secure Malware Analyticsサポートから提供されます。
- GOプログラミング言語。ダウンロード
- .caibxインデックスファイル（TACサポートによって提供されるzipファイルに含まれる）
- Desync Tool（Secure Malware Analyticsサポートが提供するzipファイルに含まれる）

使用するコンポーネント

このドキュメントの情報は、CentOS Linuxリリース7.6.1810（コア）に基づくものです。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このド

キュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

設定

GOプログラミング言語のインストール

```
# wget https://dl.google.com/go/go1.12.2.linux-amd64.tar.gz
# tar -xzf go1.12.2.linux-amd64.tar.gz
# mv go /usr/local
```

desyncコマンドが失敗しない場合は、インストール後に次の3つのコマンドを実行します

```
# export GOROOT=/usr/local/go
# export GOPATH=$HOME/Projects/Proj1
# export PATH=$GOPATH/bin:$GOROOT/bin:$PATH
```

GOバージョンは、次の方法で確認できます。

```
# go version
```

DesyncコマンドによるISOのダウンロード

ステップ 1：desync.linuxおよび.caibxファイルを含む、Secure Malware Analyticsサポートが提供するZipファイルの内容を、マシン上の同じディレクトリにローカルにコピーします。

ステップ 2：ファイルを保存したディレクトリに移動します。

以下に例を挙げます。

```
# cd MyDirectory/TG
```

ステップ 3：pwdコマンドを実行して、ディレクトリ内にいることを確認します。

```
# pwd
```

ステップ 4：desync.linuxコマンドと.caibxファイルが含まれているディレクトリ内に移動したら、任意のコマンドを実行してダウンロード処理を開始します。

注：これらは、さまざまなISOバージョンの例です。Secure Malware Analyticsサポートの指示に従って、.caibxファイルを参照してください。

バージョン2.1.3から2.4.3.2 ISOの場合：

```
# desync extract -k -s s3+https://s3.amazonaws.com/threatgrid-appliance-airgap-update airgap-update-2.1
```

バージョン2.4.3.2から2.5 ISOの場合：

```
# desync extract -k -s s3+https://s3.amazonaws.com/threatgrid-appliance-airgap-update airgap-update-2.4
```

バージョン2.5 ~ 2.7.2ag ISOの場合：

```
# desync extract -k -s s3+https://s3.amazonaws.com/threatgrid-appliance-airgap-update airgap-update-2.5
```

ダウンロードが開始されると、経過表示バーが表示されます。

注：お客様の環境でのダウンロード速度とアップグレードメディアのサイズが、ISOの作成時間に影響する可能性があります。
ダウンロードしたファイルのMD5と、サポートが提供するバンドルで入手可能なファイルを比較して、ダウンロードしたISOの整合性を確認してください。

ダウンロードが完了すると、同じディレクトリにISOが作成されます。

USBをマシンにプラグインし、ddコマンドを実行してブータブルUSBドライブを作成します。

```
# dd if=airgap-update.iso of=/dev/<MY_USB> bs=64M
```

ここで、<MY_USB>はUSBキーの名前です（山カッコは省略）。

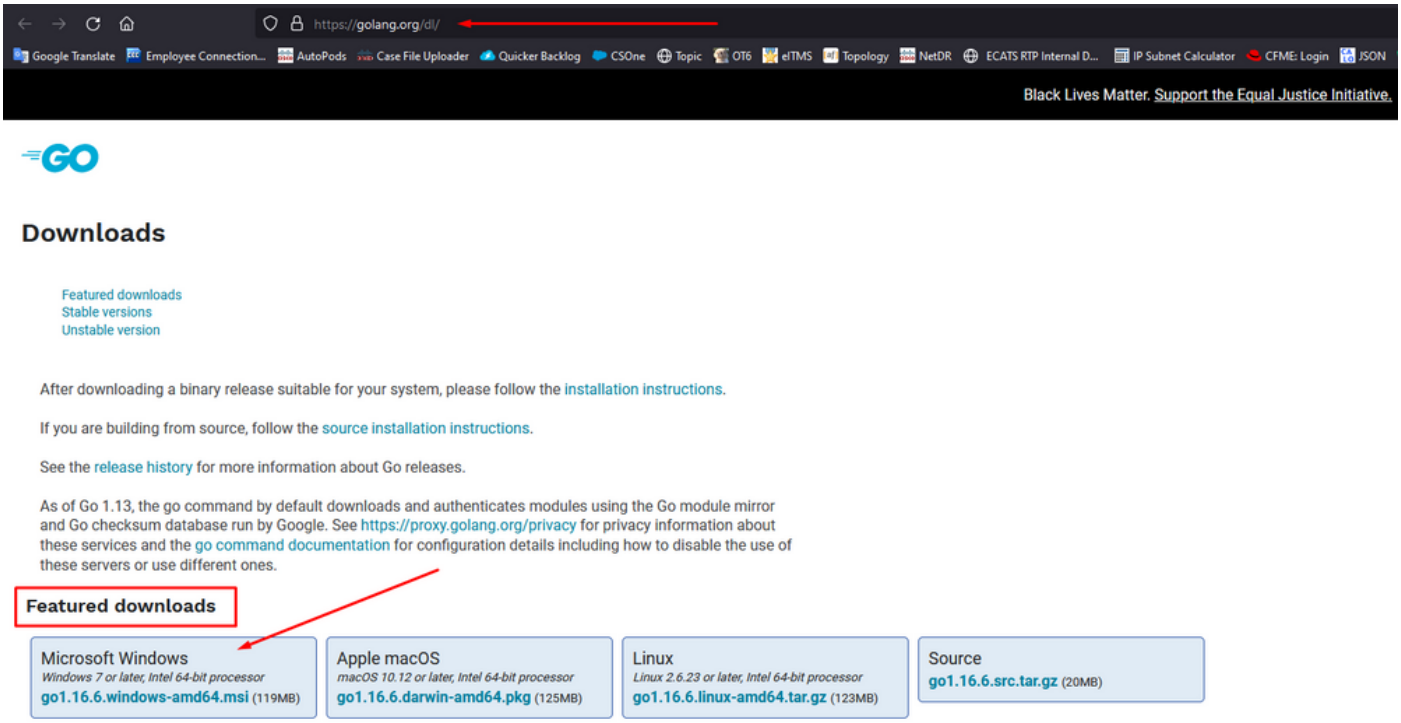
USBドライブを挿入し、アプライアンスの電源を入れるか再起動します。シスコのブート画面が表示されたら、F6キーを押してBoot Menuに入ります。

ヒント：
帯域幅に影響する可能性があるため、営業時間後またはオフピーク時にダウンロードを実行します。
ツールを停止するには、端末を閉じるか、Ctrl+c/Ctrl+zを押します。
続行するには、同じコマンドを実行してダウンロードを再開します。

Windows - ISOダウンロード

GOプログラミング言語のインストール

#1:必要なGOプログラミング言語をダウンロードします。https://golang.org/dl/からのインストール私の場合は、Featured Versionを選択します。CMDを再起動し、



確認するには、CMD runコマンドを閉じてから再び開きます。

```
go version
```

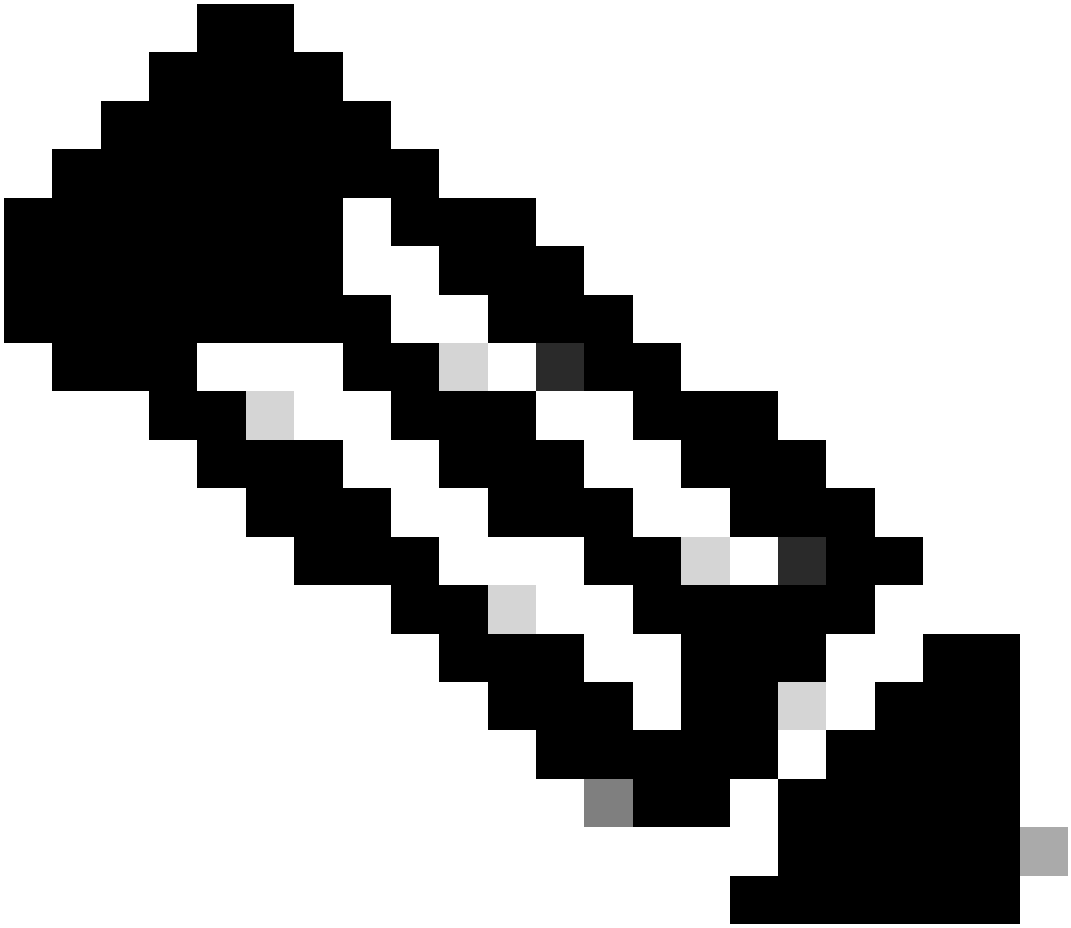
DesyncコマンドによるISOのダウンロード

#2:DESYNCツールをインストールします。コマンドを実行すると、多数のダウンロードプロンプトが表示されます。およそ2 ~ 3分後に、ダウンロードを実行する必要があります。

```
go install github.com/folbricht/desync/cmd/desync@latest
```

```
In case desync is not working using above command then change directory to C drive and run this command
```

```
git clone https://github.com/folbricht/desync.git
```



> 注: gitコマンドが機能しない場合は、https://git-scm.com/download/winからGitをダウンロードしてインストールできます。

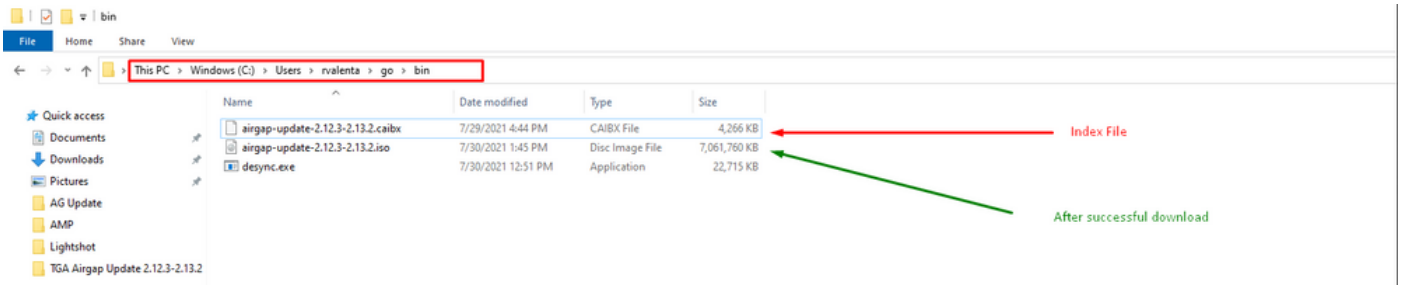
次に、次の2つのコマンドを1つずつ実行します。

```
cd desync/cmd/desync
```

```
go install
```

```
C:\Users\rvalenta>go install github.com/folbricht/desync/cmd/desync@latest
go: downloading github.com/folbricht/tempfile v0.0.1
go: downloading github.com/go-ini/ini v1.62.0
go: downloading github.com/minio/minio-go/v6 v6.0.57
go: downloading github.com/pkg/errors v0.9.1
go: downloading github.com/sirupsen/logrus v1.7.0
go: downloading github.com/spf13/cobra v1.1.1
go: downloading github.com/spf13/pflag v1.0.5
go: downloading golang.org/x/crypto v0.0.0-20201221181555-eec23a3978ad
go: downloading github.com/sirupsen/logrus v1.8.1
go: downloading gopkg.in/cheggaaa/pb.v1 v1.0.28
go: downloading github.com/spf13/cobra v1.2.1
go: downloading github.com/minio/minio-go v1.0.0
go: downloading cloud.google.com/go v0.72.0
go: downloading github.com/DataDog/zstd v1.4.5
go: downloading github.com/boljen/go-bitmap v0.0.0-20151001105940-23cd2fb0ce7d
go: downloading github.com/dchest/siphash v1.2.2
go: downloading github.com/hanwen/go-fuse v1.0.0
go: downloading github.com/klauspost/compress v1.11.4
go: downloading github.com/DataDog/zstd v1.4.8
go: downloading github.com/hanwen/go-fuse/v2 v2.0.3
go: downloading github.com/pkg/sftp v1.12.0
go: downloading golang.org/x/crypto v0.0.0-20210711020723-a769d52b0f97
go: downloading github.com/minio/minio-go v6.0.14+incompatible
go: downloading github.com/pkg/sftp v1.13.2
go: downloading github.com/pkg/xattr v0.4.3
go: downloading golang.org/x/sync v0.0.0-20201207232520-09787c993a3a
go: downloading google.golang.org/api v0.36.0
go: downloading github.com/hanwen/go-fuse/v2 v2.1.0
go: downloading golang.org/x/sync v0.0.0-20210220032951-036812b2e83c
go: downloading github.com/mattn/go-runewidth v0.0.9
go: downloading golang.org/x/sys v0.0.0-20201201145000-ef89a241ccb3
```

#3:Go — > bin locationに移動します。私の場合はC:\Users\rvalenta\go\binで、TAC提供の.caibxインデックスファイルにコピー/ペーストしました。



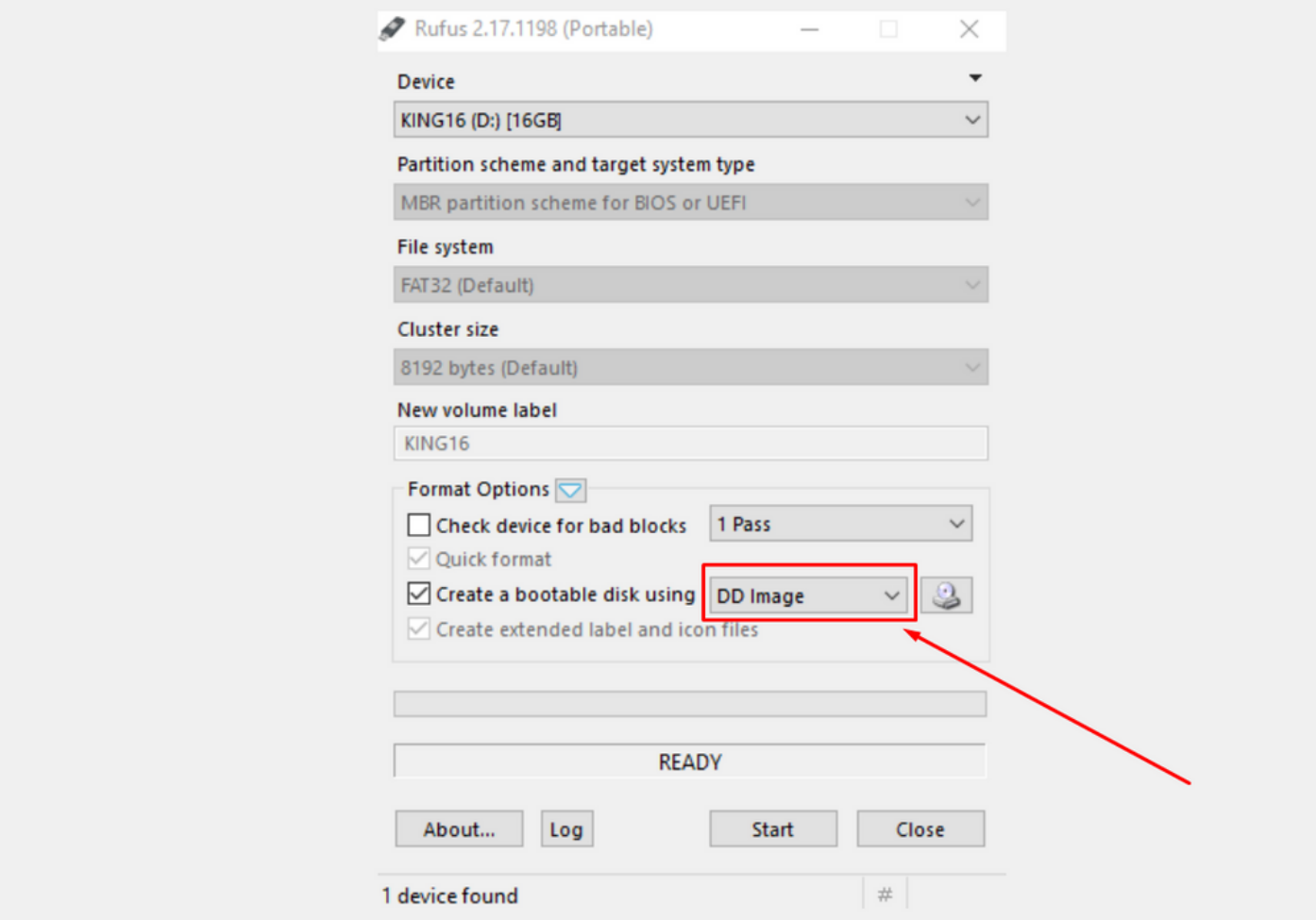
確認

#4: CMDプロンプトに戻り、フォルダgo\binに移動して、downloadコマンドを実行します。すぐにダウンロードが続行していることが表示されます。ダウンロードが完了するまで待ちます。これで、.ISOファイル全体が、以前コピーした.caibxインデックスファイルと同じ場所に作成されます

```
desync extract -k -s s3+https://s3.amazonaws.com/threatgrid-appliance-airgap-update airgap-update-2.12.
```

```
C:\Users\rvalenta>cd go

C:\Users\rvalenta\go>cd bin

C:\Users\rvalenta\go\bin>desync extract -k -s s3+https://s3.amazonaws.com/threatgrid-appliance-airgap-update airgap-update-2.12.3-2.13.2.caibx airgap-update-2.12.3-2.13.2.iso
Error: airgap-update-2.12.3-2.13.2.caibx: open ./airgap-update-2.12.3-2.13.2.caibx: The system cannot find the file specified.

C:\Users\rvalenta\go\bin>desync extract -k -s s3+https://s3.amazonaws.com/threatgrid-appliance-airgap-update airgap-update-2.12.3-2.13.2.caibx airgap-update-2.12.3-2.13.2.iso
[==========================================================================================================================================================================] 100.00% 16m52s

C:\Users\rvalenta\go\bin>_
```

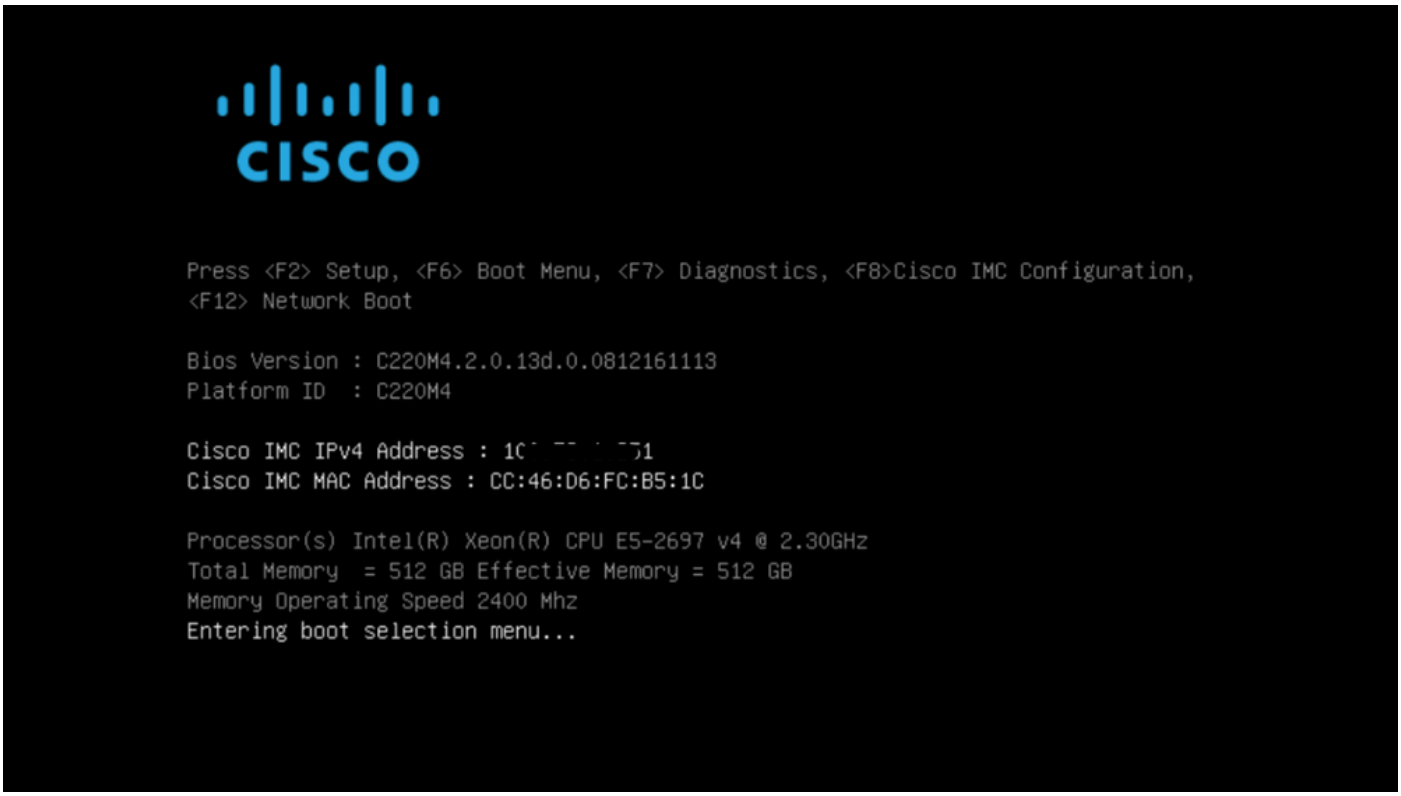
次に、RUFUSを使用してブート可能なUSBを作成します。これは、バージョン2.17を使用する上で非常に重要です。これは、この特定のリカバリUSBを作成するために非常に重要であるddオプションを使用できる最後のバージョンです。このリポジトリのすべてのバージョンが見つかりますRUFUS REPOSITORYこれらのファイルが使用できなくなった場合に備えて、このドキュメントにはフルバージョンとポータブルバージョンのインストーラも含まれています。



USBからのアプライアンスのブート

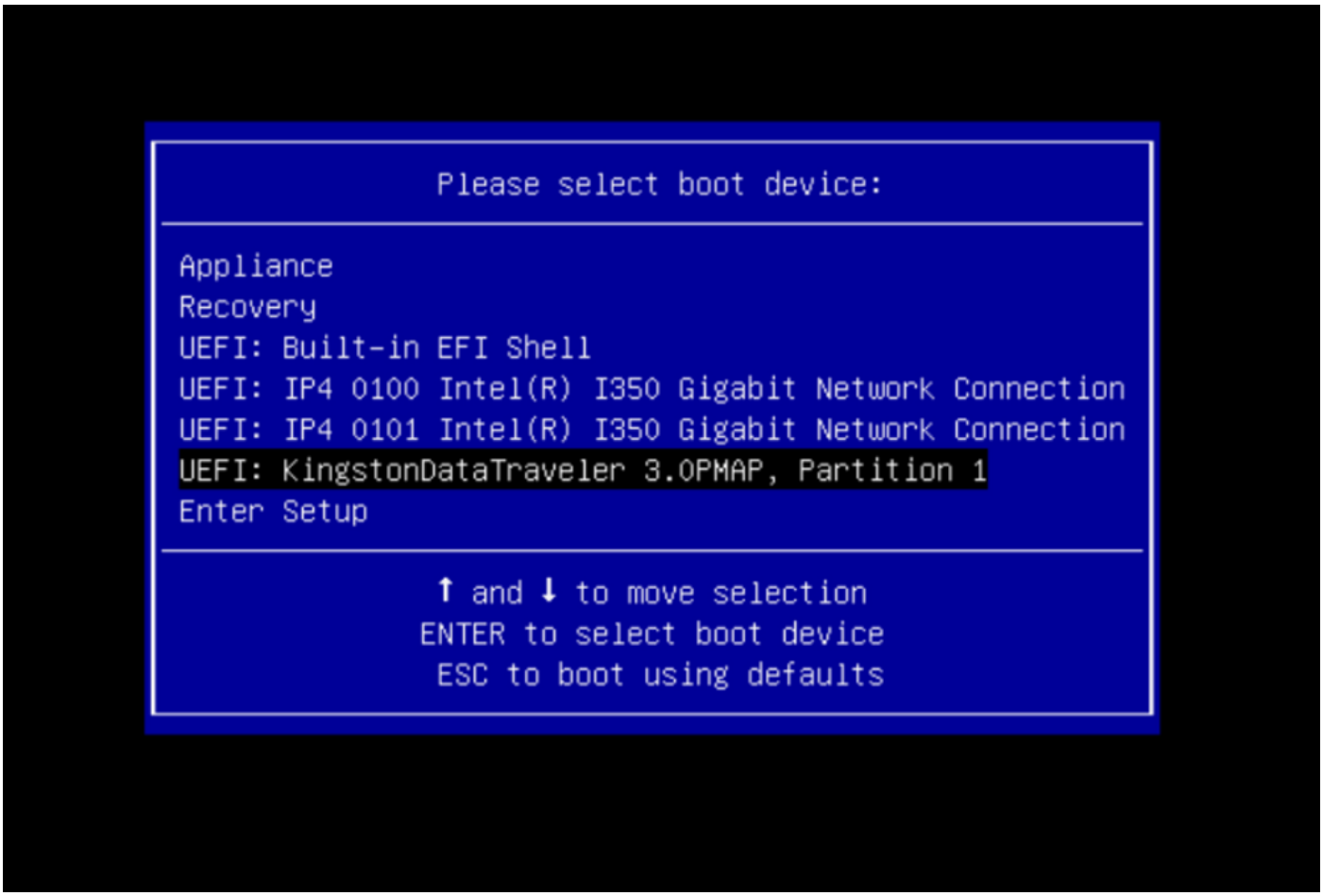
USBドライブを挿入し、アプライアンスの電源を入れるか再起動します。シスコのブート画面で、F6キーを押してブートメニューに入ります。君は急がなければならない。この選択は数秒で完了します。これを見逃した場合は、リブートして再試行する必要があります。

図1 - F6を押してブートメニューに入る



更新プログラムが格納されているUSBドライブに移動し、Enterキーを押して次の項目を選択します。

図2 -[Update USB]（USBのアップデート）の選択



アップデートメディアは、アップグレードパス内の次のリリースを判別し、そのリリースのコンテンツをアプライアンスにコピーします。アプライアンスでは、アップグレードが即座に実行されるか、リブートして通常の動作モードに戻り、OpAdminに切り替えて手動でアップグレードを開始できます。

ISOブートプロセスが完了したら、Secure Malware Analyticsアプライアンスを再起動して動作モードに戻します。

続行する前に、ポータルUIにログインし、アップグレードが安全かどうかなど、警告がないか確認します。

再起動中に自動的に適用されなかった場合は、OpAdminインタフェースに移動して更新を適用します。 OpAdmin > Operations > Update Appliance注：更新プロセスには、USBメディアから行われる更新の一部として、追加の再起動が含まれます。たとえば、更新プログラムをインストールした後、インストールページの[再起動]ボタンを使用する必要があります。

必要に応じて、USBの各バージョンに対して手順を繰り返します。

正しい/devデバイスの検索方法

USBがまだエンドポイントに接続されていない状態で、コマンド「lsblk | grep -iE 'disk|part'を実

行します。

```
xsilenc3x@Alien15:~/testarea/usb$ lsblk | grep -iE 'disk|part'
sda           8:0    0 931.5G  0 disk
├─sda1        8:1    0   128M  0 part
└─sda2        8:2    0 931.4G  0 part /media/DATA
nvme0n1     259:0    0 238.5G  0 disk
├─nvme0n1p1 259:1    0   650M  0 part
├─nvme0n1p2 259:2    0   128M  0 part
├─nvme0n1p3 259:3    0 114.1G  0 part
├─nvme0n1p4 259:4    0   525M  0 part /boot
├─nvme0n1p5 259:5    0   7.6G  0 part [SWAP]
├─nvme0n1p6 259:6    0  38.2G  0 part /
├─nvme0n1p7 259:7    0  62.7G  0 part /home
├─nvme0n1p8 259:8    0  13.1G  0 part
└─nvme0n1p9 259:9    0   1.1G  0 part
xsilenc3x@Alien15:~/testarea/usb$
```

USBスティックが接続された後。

```
xsilenc3x@Alien15:~/testarea/usb$ lsblk | grep -iE 'disk|part'
.sda           8:0    0 931.5G  0 disk
├─sda1        8:1    0   128M  0 part
└─sda2        8:2    0 931.4G  0 part /media/DATA
sdb           8:16   1   3.7G  0 disk
└─sdb1        8:17   1   3.7G  0 part /media/xsilenc3x/ARCH_201902  <--------- not observed when the USB was not
nvme0n1     259:0    0 238.5G  0 disk
├─nvme0n1p1 259:1    0   650M  0 part
├─nvme0n1p2 259:2    0   128M  0 part
├─nvme0n1p3 259:3    0 114.1G  0 part
├─nvme0n1p4 259:4    0   525M  0 part /boot
├─nvme0n1p5 259:5    0   7.6G  0 part [SWAP]
├─nvme0n1p6 259:6    0  38.2G  0 part /
├─nvme0n1p7 259:7    0  62.7G  0 part /home
├─nvme0n1p8 259:8    0  13.1G  0 part
└─nvme0n1p9 259:9    0   1.1G  0 part
xsilenc3x@Alien15:~/testarea/usb$
```

これにより、/devのUSBデバイスが「/dev/sdb」であることが確認されます。

USBスティックが接続された後に確認する他の方法：

dmesgコマンドを発行すると、いくつかの情報が得られます。USBを接続したら、dmesgコマンドを実行します。 | grep -iE 'usb|attached'を実行します。

```
xsilenc3x@Alien15:~/testarea/usb$ dmesg | grep -iE 'usb|attached'
[842717.663757] usb 1-1.1: new high-speed USB device number 13 using xhci_hcd
[842717.864505] usb 1-1.1: New USB device found, idVendor=0781, idProduct=5567
[842717.864510] usb 1-1.1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
```

```
[842717.864514] usb 1-1.1: Product: Cruzer Blade
[842717.864517] usb 1-1.1: Manufacturer: SanDisk
[842717.864519] usb 1-1.1: SerialNumber: 4C530202420924105393
[842717.865608] usb-storage 1-1.1:1.0: USB Mass Storage device detected
[842717.866074] scsi host1: usb-storage 1-1.1:1.0
[842718.898700] sd 1:0:0:0: Attached scsi generic sg1 type 0
[842718.922265] sd 1:0:0:0: [sdb] Attached SCSI removable disk  <-------
xsilenc3x@Alien15:~/testarea/usb$
```

コマンドfidskは、サイズに関する情報を提供します。この情報は、sudo fdisk -l /dev/sdbで確認できます。

```
xsilenc3x@Alien15:~/testarea/usb$ sudo fdisk -l /dev/sdb
Disk /dev/sdb: 3.7 GiB, 4004511744 bytes, 7821312 sectors  <-------
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x63374e06

Device     Boot Start    End Sectors  Size Id Type
/dev/sdb1  *        0 675839  675840  330M  0 Empty
/dev/sdb2         116   8307    8192    4M ef EFI (FAT-12/16/32)
xsilenc3x@Alien15:~/testarea/usb$
```

注:「dd」コマンドを実行する前に、USBをアンマウントすることを忘れないでください。

例のUSBデバイスがマウントされていることを確認します。

```
xsilenc3x@Alien15:~/testarea/usb$ sudo mount -l | grep -i sdb
/dev/sdb1 on /media/xsilenc3x/ARCH_201902 type vfat (rw,nosuid,nodev,relatime,uid=1000,gid=1000,fmask=0
```

USBデバイスをアンマウントするには、sudo umount /dev/sdb1を使用します。

```
xsilenc3x@Alien15:~/testarea/usb$ sudo umount /dev/sdb1
```

デバイスが「マウント済み」として認識されないことを再確認します。

```
xsilenc3x@Alien15:~/testarea/usb$ sudo mount -l | grep -i sdb
```

status=progressオプション

ddコマンドのoflag=syncおよびstatus=progressオプション。

多数のデータ・ブロックを書き込む場合、「status=progress」オプションは現在の書き込み操作の情報を提供します。これは、「dd」コマンドがページキャッシュに書き込み中かどうかを確認するのに便利です。このコマンドを使用すると、すべての書き込み操作の進行状況と完了した時間を秒単位で表示できます。

使用しない場合、「dd」は進行状況に関する情報を提供せず、「dd」が戻る前に書込み操作の結果のみが提供されます：

```
[rootuser@centos8-01 tga-airgap]$ dd if=/dev/zero of=testfile.txt bs=1M count=8192
8192+0 records in
8192+0 records out
8589934592 bytes (8.6 GB, 8.0 GiB) copied, 5.03493 s, 1.7 GB/s
[rootuser@centos8-01 tga-airgap]$
```

書き込み操作に関するリアルタイム情報は、使用されると1秒ごとに更新されます。

```
[rootuser@centos8-01 tga-airgap]$ dd if=/dev/zero of=testfile.txt bs=1M count=8192 status=progress
8575254528 bytes (8.6 GB, 8.0 GiB) copied, 8 s, 1.1 GB/s <----------------
8192+0 records in
8192+0 records out
8589934592 bytes (8.6 GB, 8.0 GiB) copied, 8.03387 s, 1.1 GB/s
[rootuser@centos8-01 tga-airgap]
```

注:TGAオフラインアップグレードプロセスの公式ドキュメントでは、通知されるコマンドは次のとおりです。dd if=airgap-update.iso of=/dev/<MY_USB> bs=64M

いくつかのテストの後、次の例が確認されました。

ファイルが作成されると、「dd」を含む10 MBの空き領域が/dev/zeroデバイスを使用して作成されます。

1M x 10 = 10M(10240 kB +ダーティファイルページキャッシュ内の以前のシステムデータ= 10304 kB —>これは、「dd」の最後のダーティページキャッシュで認識される内容です)。

```
[rootuser@centos8-2 testarea]$ cat /proc/meminfo | grep -iE 'dirty' && dd if=/dev/zero of=testfile.txt
count=10 status=progress && cat /proc/meminfo | grep -iE 'dirty' && date +%s
Dirty:                92 kB
10+0 records in
10+0 records out
```

```
10485760 bytes (10 MB, 10 MiB) copied, 0.0138655 s, 756 MB/s
Dirty:             10304 kB <----- dirty page cache after "dd" returned | data still to be written to t
1633260775 <---- epoch time
[rootuser@centos8-2 testarea]$ cat /proc/meminfo | grep -iE 'dirty' && date +%s
Dirty:             10372 kB
1633260778
[rootuser@centos8-2 testarea]$ cat /proc/meminfo | grep -iE 'dirty' && date +%s
Dirty:             10380 kB
1633260779
[rootuser@centos8-2 testarea]$ cat /proc/meminfo | grep -iE 'dirty' && date +%s
Dirty:             10404 kB
1633260781
[rootuser@centos8-2 testarea]$ cat /proc/meminfo | grep -iE 'dirty' && date +%s
Dirty:             10412 kB
1633260782
[rootuser@centos8-2 testarea]$ cat /proc/meminfo | grep -iE 'dirty' && date +%s
Dirty:             10424 kB
1633260783
[rootuser@centos8-2 testarea]$ cat /proc/meminfo | grep -iE 'dirty' && date +%s
Dirty:             10436 kB
1633260785
[rootuser@centos8-2 testarea]$ cat /proc/meminfo | grep -iE 'dirty' && date +%s
Dirty:                 0 kB  <--- data in the dirty page cache flushed = written to the block device
1633260786 <---- epoch time
[rootuser@centos8-2 testarea]$
```

1633260786 - 1633260775 = 11 seconds

注:「dd」コマンドが返された後、ブロックデバイスへの書き込み操作が完了せず、返された後11秒後に認識されました。
TGA ISOを使用してブート可能なUSBを作成する際にこの「dd」コマンドを使用し、その11秒前にエンドポイントからUSBを取り外していた場合、ブート可能なUSBに破損したISOが含まれていたことになります。

説明：

ブロックデバイスは、ハードウェアデバイスへのバッファアクセスを提供します。これにより、ハードウェアデバイスを使用する際にアプリケーションに抽象化レイヤが提供されます。

ブロック・デバイスを使用すると、アプリケーションはサイズの異なるデータ・ブロックを使用して読み取り/書き込みを行うことができます。このread()/writes()はページ・キャッシュ（バッファ）に適用され、ブロック・デバイスに直接適用されることはありません。
カーネル（読み取り/書き込みを行うアプリケーションではない）は、バッファ（ページキャッシュ）からブロックデバイスへのデータの移動を管理します。

したがって

が指示されていない場合、アプリケーション（この場合は「dd」）はバッファのフラッシュを制御できません。

オプション"oflag=sync"は、出力ブロック（"dd"が提供）がページキャッシュに置かれた後に（カーネルによる）同期物理書き込みを強制する。

oflag=syncは、このオプションを使わないときと比べて、「dd」の性能を下げます。しかし、このオプションが有効な場合は、「dd」からのwrite()呼び出しごとにブロックデバイスへの物理的な書き込みが行われます。

テスト：「dd」コマンドの「oflag=sync」オプションを使用して、「dd」コマンドの戻り値でダーティ・ページ・キャッシュ・データによるすべての書込み操作が完了したことを確認します。

```
[rootuser@centos8-2 testarea]$ cat /proc/meminfo | grep -iE 'dirty' && dd if=/dev/zero of=testfile.txt b
count=10 oflag=sync status=progress && cat /proc/meminfo | grep -iE 'dirty' && date +%s
Dirty:                60 kB
10+0 records in
10+0 records out
10485760 bytes (10 MB, 10 MiB) copied, 0.0841956 s, 125 MB/s
Dirty:                68 kB <---- No data remaining in the dirty page cache after "dd" returned
1633260819
[rootuser@centos8-2 testarea]$ cat /proc/meminfo | grep -iE 'dirty' && date +%s
Dirty:                36 kB
1633260821
[rootuser@centos8-2 testarea]$
```

ダーティ・ページ・キャッシュへの書き込み操作でデータが残っていない。
書き込み操作は、「dd」コマンドが返される前（または同時）に適用されました（前のテストの11秒後ではありません）。
「dd」コマンドが返された後、書き込み操作に関連するダーティ・ページ・キャッシュ内にデータは存在しませんでした=ブート可能なUSB作成に問題はありません（ISOチェックサムが正しい場合）。

注：この種のケースを扱う際には、「dd」コマンドのこのフラグ(oflag=sync)を考慮してください。

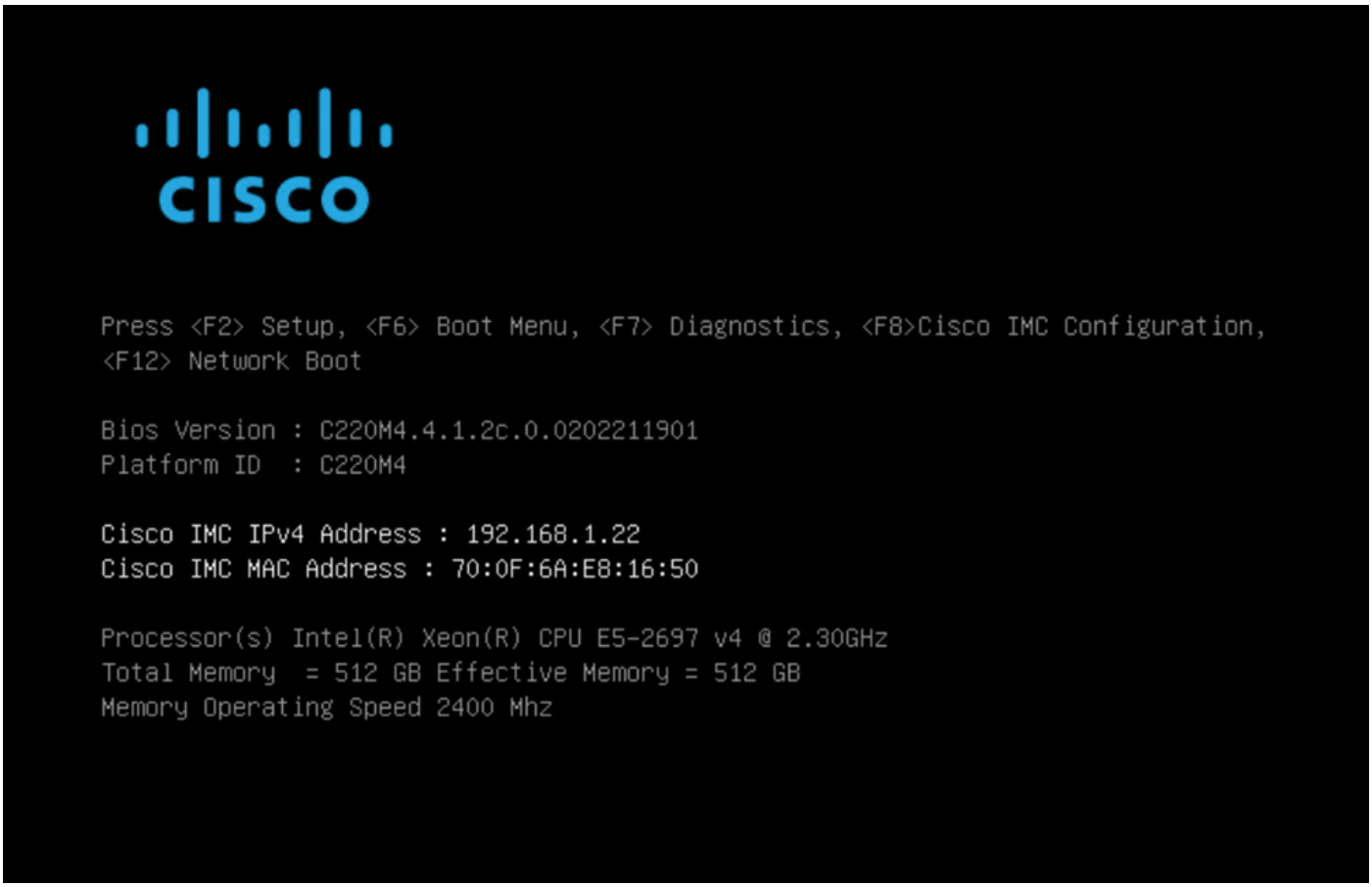
オフラインアップグレード用のHDDドライブのブートシーケンス

要件：

HDDが利用可能なツールを使用して「DD」オプションを使用してフォーマットされ、メディアがドライブに後でコピーされることを確認する必要があります。このフォーマットを使用しないと、このメディアを読み取ることができません。

「DD」フォーマットを使用してメディアをHDD/USBにロードしたら、それをTGAアプライアンスに接続してデバイスを再起動する必要があります。

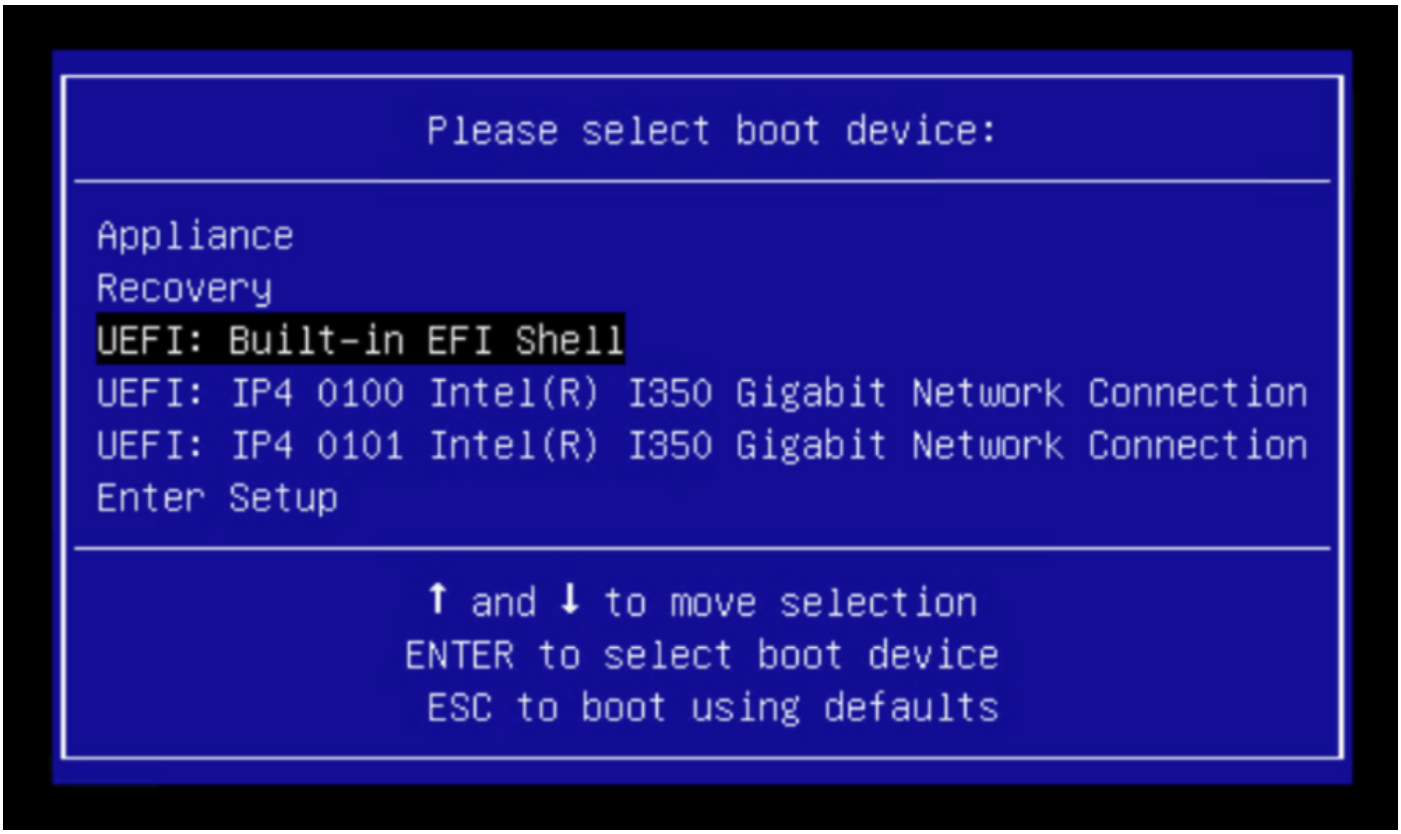
これはデフォルトのブートメニュー選択画面です。「F6」を押してデバイスを起動し、ブートメディアを選択する必要があります



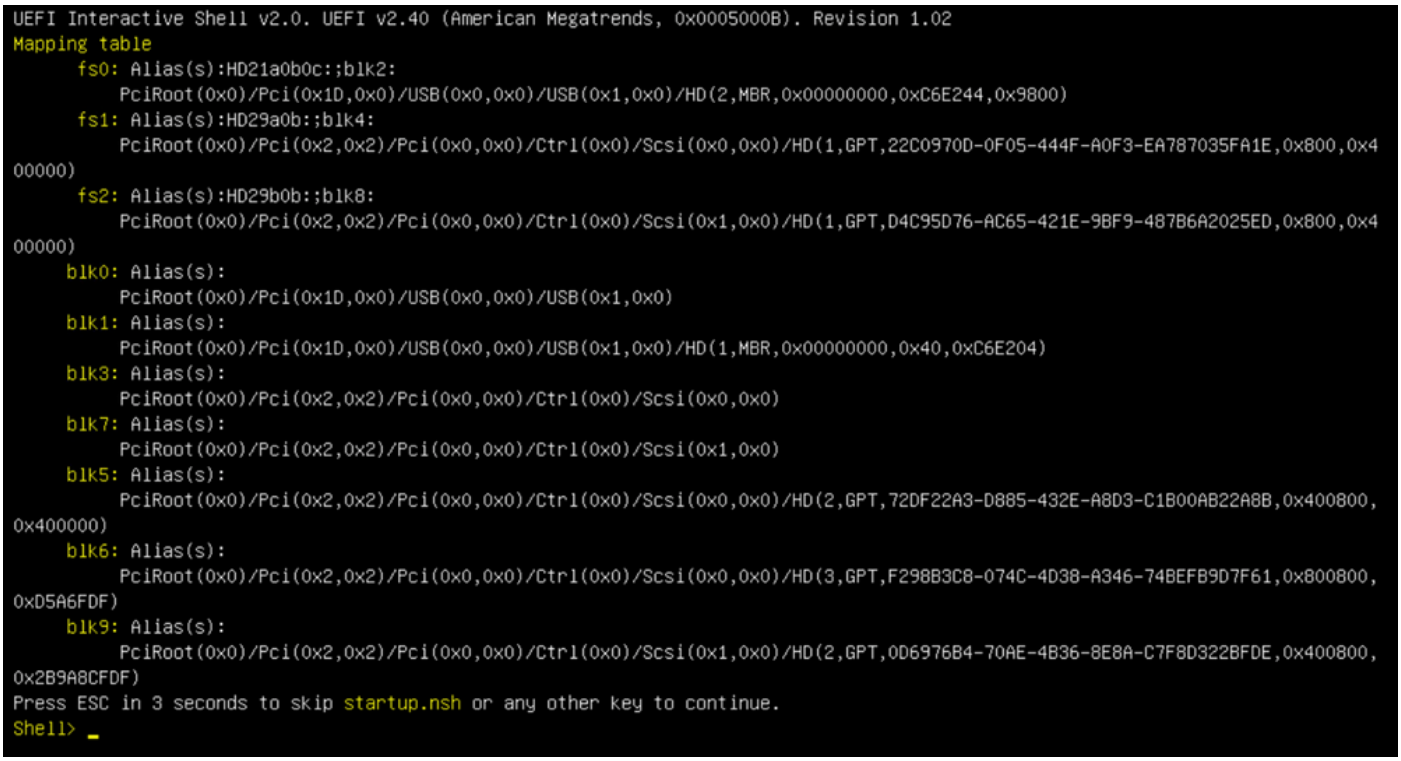
デバイスが入力を認識すると、デバイスはブート選択メニューに入ることを求められます。

これは、異なるTGAモデル間で異なる可能性があるプロンプトです。理想的には、このメニュー

自体からブートメディア (ファイルシステムのアップグレード) を使用して起動するオプションが表示されますが、表示されない場合は「EFIシェル」にログインする必要があります。



「startup.sh」スクリプトが終了してEFIシェルに移動する前に、「ESC」を押す必要があります。EFIシェルにログインすると、この場合に検出されるパーティションが3つのファイルシステム(fs0:、fs1:、fs2)であることがわかります。



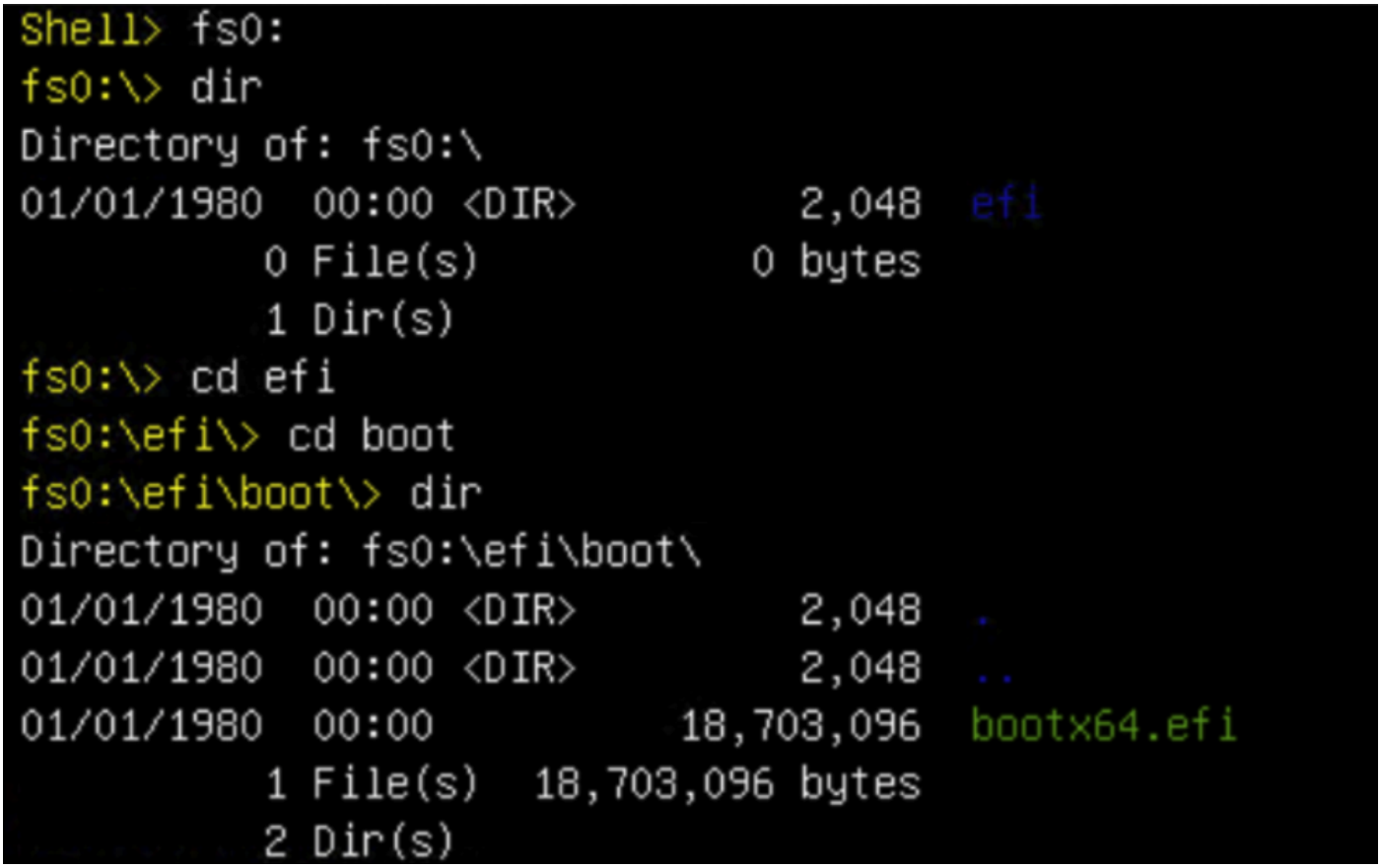
重要

正しいファイルシステムの識別：

- 上記のスクリーンショットでは、「fs0:」がパスに「USB」を持つ唯一のメディアであることがわかります。したがって、このファイルシステムにはブートメディア（アップグレードファイルシステム）が含まれていると確信できます。

ファイル・システムが見つからない場合：

- fs0:とfs1:だけが使用可能で、fs2:が存在しない場合は、ブートメディア（アップグレードファイルシステム）がddモードで書かれ、正常に接続されていることを確認します。
- ブートメディア（アップグレードファイルシステム）の番号は常にリカバリメディアよりも小さくし、常に隣り合っている必要があります。USB接続ドライブが変更される可能性が高い最後の最初にあるかどうか(つまり、USB接続ドライブがfs0:で前面に位置するか、fs2:で背面に位置するか)を特定する必要があります
- 次のスクリーンショットのこの例では、「\efi\boot」パーティションの下にあり、命名規則が「bootx64.efi」であるため、正しい「.efi」ファイルです。



ブートメディア（アップグレードファイルシステム）でデバイスをブートするには、「bootx64.efi」ファイルを実行する必要があります。

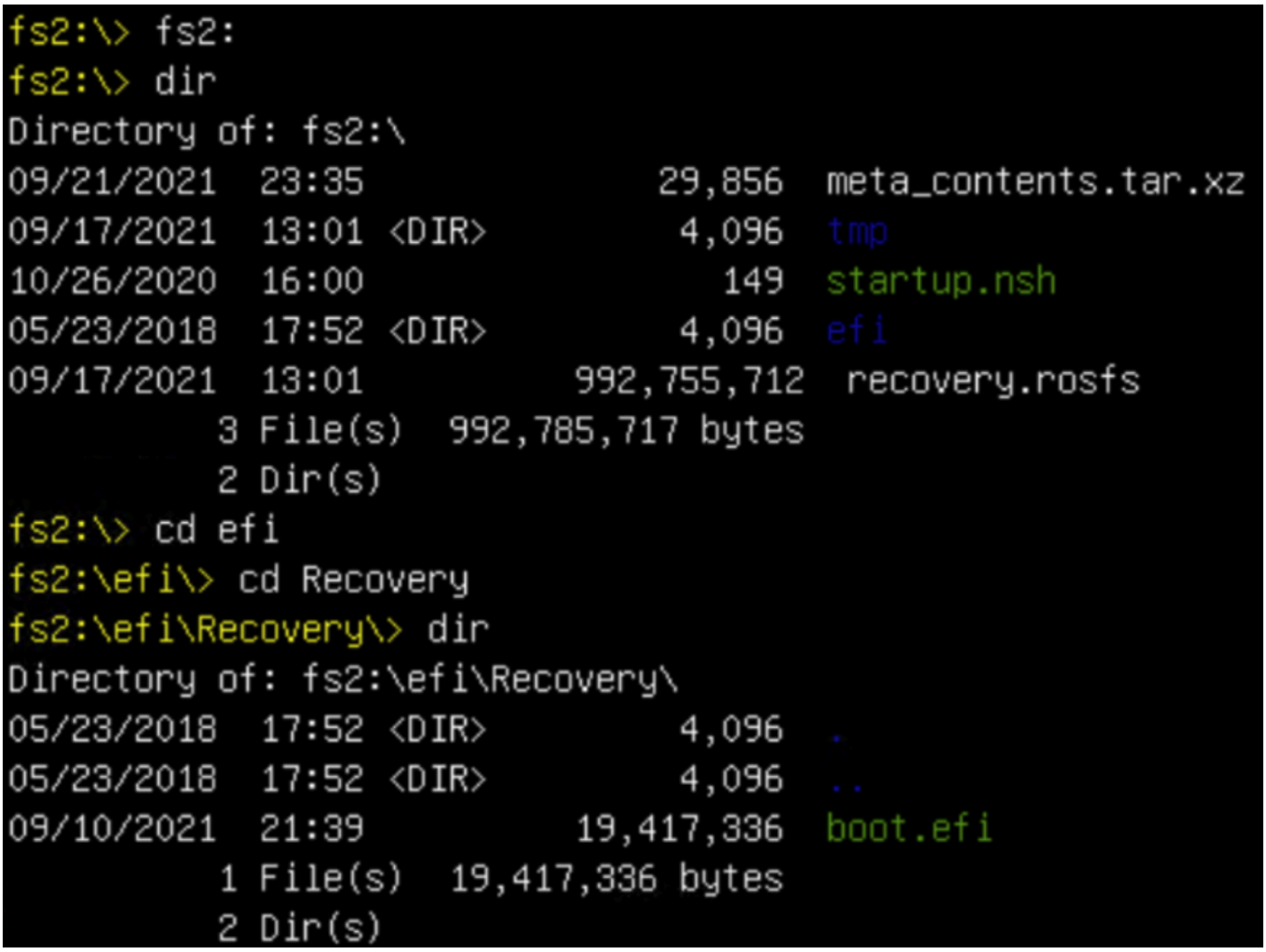
```
fs0:\efi\boot\bootx64.efi
```

参考のために、他のファイルシステムの内容も表示しました。

fs1：これはメインのブートファイルシステムです。

```
fs1:\> fs1:
fs1:\> dir
Directory of: fs1:\
01/01/1980  00:00          43,985,838  initramfs-appliance.img
01/01/1980  00:00                 287  initramfs-appliance.img.sig
01/01/1980  00:00           5,490,560  vmlinuz-appliance
01/01/1980  00:00                 287  vmlinuz-appliance.sig
01/01/1980  00:00                   4  .gitignore
01/01/1980  00:00 <DIR>         4,096  efi
01/01/1980  00:00                 149  startup.nsh
01/01/1980  00:00           6,199,680  vmlinuz-linux
          7 File(s)  55,676,805 bytes
          1 Dir(s)
fs1:\> cd efi
fs1:\efi\> dir
Directory of: fs1:\efi\
05/23/2018  17:52 <DIR>         4,096  .
05/23/2018  17:52 <DIR>             0  ..
01/01/1980  00:00 <DIR>         4,096  Appliance
          0 File(s)              0 bytes
          3 Dir(s)
fs1:\efi\> cd Appliance
fs1:\efi\Appliance\> dir
Directory of: fs1:\efi\Appliance\
05/23/2018  17:52 <DIR>         4,096  .
05/23/2018  17:52 <DIR>         4,096  ..
01/01/1980  00:00       r  18,131,752  boot.efi
01/01/1980  00:00                 287  boot.efi.sig
          2 File(s)  18,132,039 bytes
          2 Dir(s)
```

fs2：これはリカバリイメージのブートファイルシステムです。



その他の手順：

マウントされたブートメディアを含む正しいファイルシステムを確認します。これを行うには、異なるファイルシステムを参照し、「.efi」ブートファイルを確認します

✎ 注：実際のブートメディア（アップグレードファイルシステム）の順序は、この場合「fs0:」であり、他のデバイスによっても異なります。
名前とパスは異なる場合がありますが、現在のすべてのイメージでは同じでなければなりません。

正しいブートメディアを見つけるのに役立つチェックリスト（ファイルシステムのアップグレード）：

- ファイルシステムのルートに「vmlinuz-appliance」が含まれている場合、それはブートメディア（アップグレードファイルシステム）ではありません。
- ファイルシステムのルートに「meta_contents.tar.xz」が含まれている場合、それはブートメディア（アップグレードファイルシステム）ではありません。
- ファイルシステムに「efi\boot\bootx64.efi」が含まれていない場合は、ブートメディアではありません（ファイルシステムのアップグレード）。

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。