

更新安全惡意軟體分析裝置空隙模式

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[背景資訊](#)

[限制](#)

[需求](#)

[開始之前](#)

[更新離線\(Airgapped\)安全惡意軟體分析裝置](#)

[命名慣例](#)

[限制](#)

[Linux/MAC - ISO下載](#)

[需求](#)

[採用元件](#)

[設定](#)

[使用Desync命令下載ISO](#)

[Windows - ISO下載](#)

[使用Desync命令下載ISO](#)

[驗證](#)

[從USB啟動裝置](#)

[如何查詢正確的/dev裝置](#)

[狀態=進度選項](#)

[用於離線升級的硬碟機開機順序](#)

[需求：](#)

簡介

本文檔介紹更新安全惡意軟體分析裝置Air-Gap模式的步驟。

必要條件

需求

思科建議您瞭解以下主題：

- 在Windows和Unix/Linux環境中透過指令行輸入的基本知識
- 惡意軟體分析裝置的知識
- 思科整合管理控制器(IMC)知識

採用元件

本文中的資訊係根據以下軟體和硬體版本：

- Windows 10和CentOS-8
- 魯弗斯2.17
- C220 M4

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

背景資訊

大多數安全惡意軟體分析裝置都連線到網際網路，因此使用線上更新流程。但是，在某些情況下，安全惡意軟體分析裝置僅在內部網路內維護，即「氣隙」。我們不建議保持電器氣隙，因為這樣做會降低其效率；但是，這一權衡對於支援附加安全或法規要求可能是必要的。

對於未連線到Internet運行安全惡意軟體分析裝置的使用者，我們提供本文檔中描述的離線更新過程。Secure Malware Analytics Support在請求時提供更新媒體，請參閱以下內容瞭解詳細資訊。

媒體：Airgap（離線）更新媒體由Secure Malware Analytics Support以ISO格式提供，只要有足夠的大小，就可以複製到USB媒體或HDD（硬碟）。

大小：大小取決於更新介質支援的版本，但在源版本和目的版本之間引入新VM時，大小通常為幾十吉位元組。在當前版本中，它可能大約為30 GB，因為取消同步工具有助於以增量方式更新與VM相關的更改。

升級啟動週期：每次啟動airgap更新媒體時，它都會確定要升級的下一個版本，並將與該下一個版本相關的內容複製到裝置上。如果給定版本沒有裝置運行時必須運行的任何先決條件檢查，則該版本也可以啟動軟體套件安裝。如果發行版本包括此類檢查或對更新過程中可能增加此類檢查的部分進行覆蓋，則更新實際上不會應用，直到使用者登入到OpAdmin並使用OpAdmin > Operations > Update Appliance呼叫更新為止。

預安裝掛接：根據是否存在用於該特定升級的任何預安裝掛接，它會立即運行升級或將裝置重新引導回其常規操作模式，以允許使用者進入常規管理介面並開始手動升級。

視需要重複：因此每個此類媒體啟動週期僅會升級（或準備升級）一個步驟，到達最終目標版本；使用者必須啟動任意次數，才能升級至想要的目標版本。

限制

空隙更新不支援CIMC介質。

由於使用的第三方元件的許可限制，在UCS M3硬體達到EOL（壽命終止）後，1.x版本的升級介質不再可用。因此，在EOL之前更換或升級UCS M3裝置至關重要。

需求

遷移：如果所涵蓋版本的發行版本註釋包含必須在安裝下一個版本之前進行遷移的情況，則使用者必須在重新引導之前執行以下步驟，以避免其裝置處於不可用狀態。

 注意：第一個2.1.x版本比2.1.4版本新，它運行多個資料庫遷移。在這些遷移完成之前繼續操作是不安全的。有關詳細資訊，請參閱[Threat Grid裝置2.1.5遷移說明](#)。

如果從2.1.3之前的版本開始，airgap升級介質會使用從單個許可證衍生的加密金鑰，因此需要基於每個裝置進行自定義。（唯一對使用者可見的影響是，對於支援2.1.3之前原始版本的媒體而言，安全惡意軟體分析需要事先在這些裝置上安裝許可證，而媒體在構建該媒體的清單中以外的任何裝置上都將無法工作。）

如果從2.1.3版開始或更早版本開始，則airgap介質是通用的，不需要客戶資訊。

開始之前

- 備份。在繼續進行更新之前，必須考慮備份裝置。
- 檢視要更新的版本的發行版本註釋，以驗證在計畫更新到較新版本之前是否需要進行任何後台遷移
- 驗證裝置的當前版本：OpAdmin > Operations > Update Appliance
- 檢視生成編號/版本查詢表中的安全惡意軟體分析裝置版本歷史記錄，該表在所有[Threat Grid裝置文檔](#)中均提供：發行說明、遷移說明、設定和配置指南以及管理員指南。

更新離線(Airgapped)安全惡意軟體分析裝置

首先檢查此頁上可用的Air Gapped版本：[裝置版本查詢表](#)

1. 打開TAC支援請求以獲取離線更新媒體。此請求應包括裝置序列號以及裝置內部版本號。
2. TAC支援根據您的安裝提供更新的ISO。
3. 將ISO影像燒錄至可開機USB。請注意，USB是唯一支援離線更新的裝置/方法。

命名慣例

這是更新的檔名，例如：TGA Airgap Update 2.13.2-2.14.0。

這意味著此介質可用於運行最低版本2.13.2的裝置並將裝置升級到版本2.14.0。

限制

- 空隙更新不支援CIMC介質。
- 由於使用的第三方元件的許可限制，在UCS M3硬體達到EOL（壽命終止）後，1.x版本的升級介質不再可用。因此，在EOL之前更換或升級UCS M3裝置至關重要。

Linux/MAC - ISO下載

需求

思科建議您瞭解以下主題：

- 可上網下載ISO並建立可開機USB安裝磁碟機的Linux機器。
- Airgap下載說明由安全惡意軟體分析支援提供。
- GO程式語言。 [下載](#)
- .caibx索引檔案 (包含在TAC支援部門提供的zip檔案中)。
- 取消同步工具 (包含在安全惡意軟體分析支援提供的zip檔案中)。

採用元件

本文檔中的資訊基於CentOS Linux版本7.6.1810 (核心)。

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除 (預設) 的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

設定

安裝GO程式語言

```
# 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
```

安裝後運行以下三個命令，如果desync命令未失敗

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

您可以透過以下方式驗證GO版本：

```
# go version
```

使用Desync命令下載ISO

步驟 1.複製Secure Malware Analytics Support提供的Zip檔案的內容，包括電腦上同一目錄中的desync.linux和.caibx檔案。

步驟 2.變更至儲存檔案的目錄：


範例：

```
# cd MyDirectory/TG
```

步驟 3. 運行pwd命令以確保您位於目錄內。

```
# pwd
```

步驟 4. 進入包括desync.linux 命令和.caibx 檔案的目錄後，運行您選擇的命令以開始下載過程。

 注意：以下是不同ISO版本的示例，請參閱安全惡意軟體分析支援提供的說明中的.caibx檔案。

對於版本2.1.3到2.4.3.2 ISO：

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


對於版本2.4.3.2到2.5 ISO：

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

對於2.5到2.7.2ag ISO版本：

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

下載開始後，會顯示進度列。

 注意：環境中的下載速度和升級介質的大小會影響構成ISO的時間。
請確保將下載檔案的MD5與支援提供的捆綁包中的可用檔案進行比較，以驗證下載的ISO的完整性。


下載完成後，ISO即會在同一目錄中建立。

將USB外掛到此電腦，並運行dd命令以建立可啟動USB驅動器。

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

其中<MY_USB>是USB隨身碟的名稱 (請保留尖括弧)。

插入USB驅動器並打開或重新啟動裝置。在Cisco boot up螢幕上，按F6進入Boot Menu。

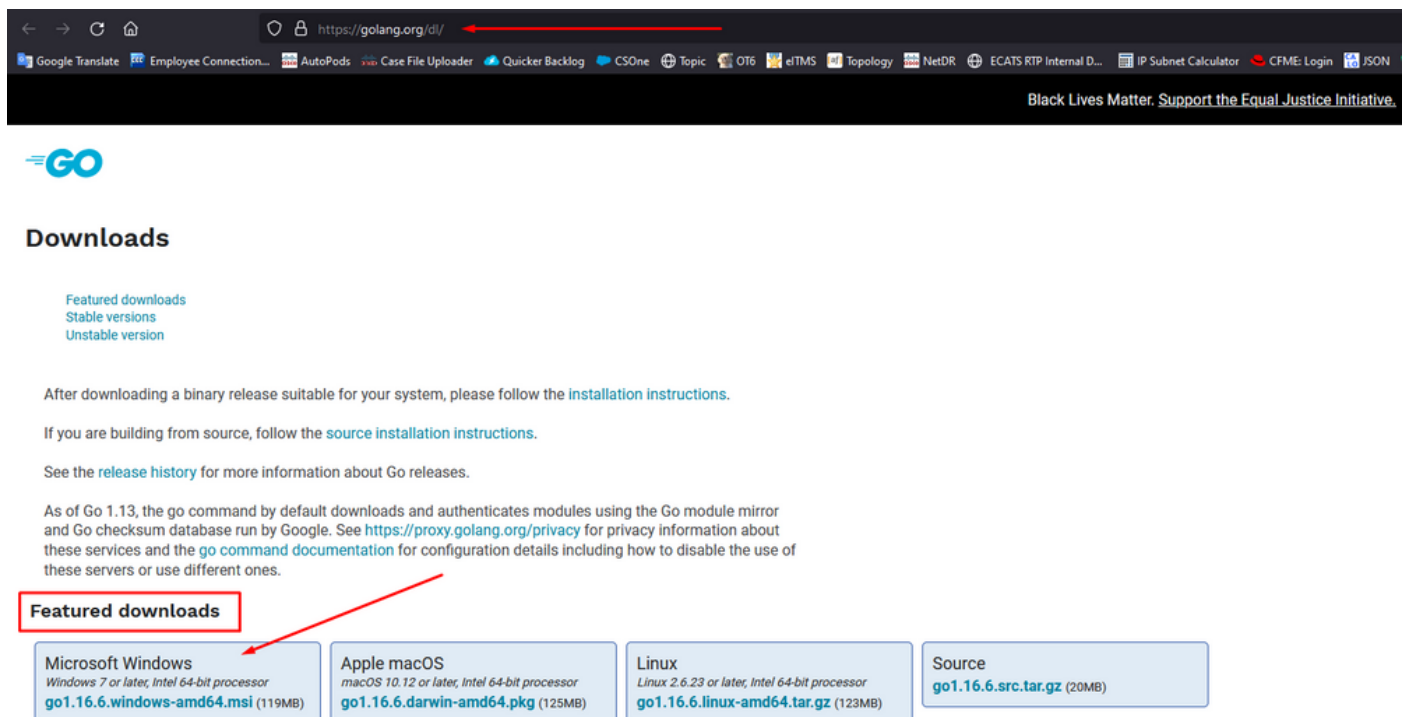
 提示：

在辦公時間或非高峰時間運行下載，因為它可能會影響頻寬。
要停止工具，請關閉終端或按Ctrl+c/Ctrl+z。
若要繼續，請執行相同的命令以繼續下載。

Windows - ISO下載

安裝GO程式語言

#1：下載所需的GO程式語言。從<https://golang.org/dl/>安裝 (以我的為例) 我選擇了特色版本。重新啟動CMD並使用



Featured downloads
Stable versions
Unstable version

After downloading a binary release suitable for your system, please follow the [installation instructions](#).

If you are building from source, follow the [source installation instructions](#).

See the [release history](#) for more information about Go releases.

As of Go 1.13, the go command by default downloads and authenticates modules using the Go module mirror and Go checksum database run by Google. See <https://proxy.golang.org/privacy> for privacy information about these services and the [go command documentation](#) for configuration details including how to disable the use of these servers or use different ones.

Featured downloads

Microsoft Windows <i>Windows 7 or later, Intel 64-bit processor</i> go1.16.6.windows-amd64.msi (119MB)	Apple macOS <i>macOS 10.12 or later, Intel 64-bit processor</i> go1.16.6.darwin-amd64.pkg (125MB)	Linux <i>Linux 2.6.23 or later, Intel 64-bit processor</i> go1.16.6.linux-amd64.tar.gz (123MB)	Source go1.16.6.src.tar.gz (20MB)
---	--	---	---

關閉並重新打開CMD run命令以驗證：

```
go version
```

```
C:\Users\rvalenta>go version  
go version go1.16.6 windows/amd64
```

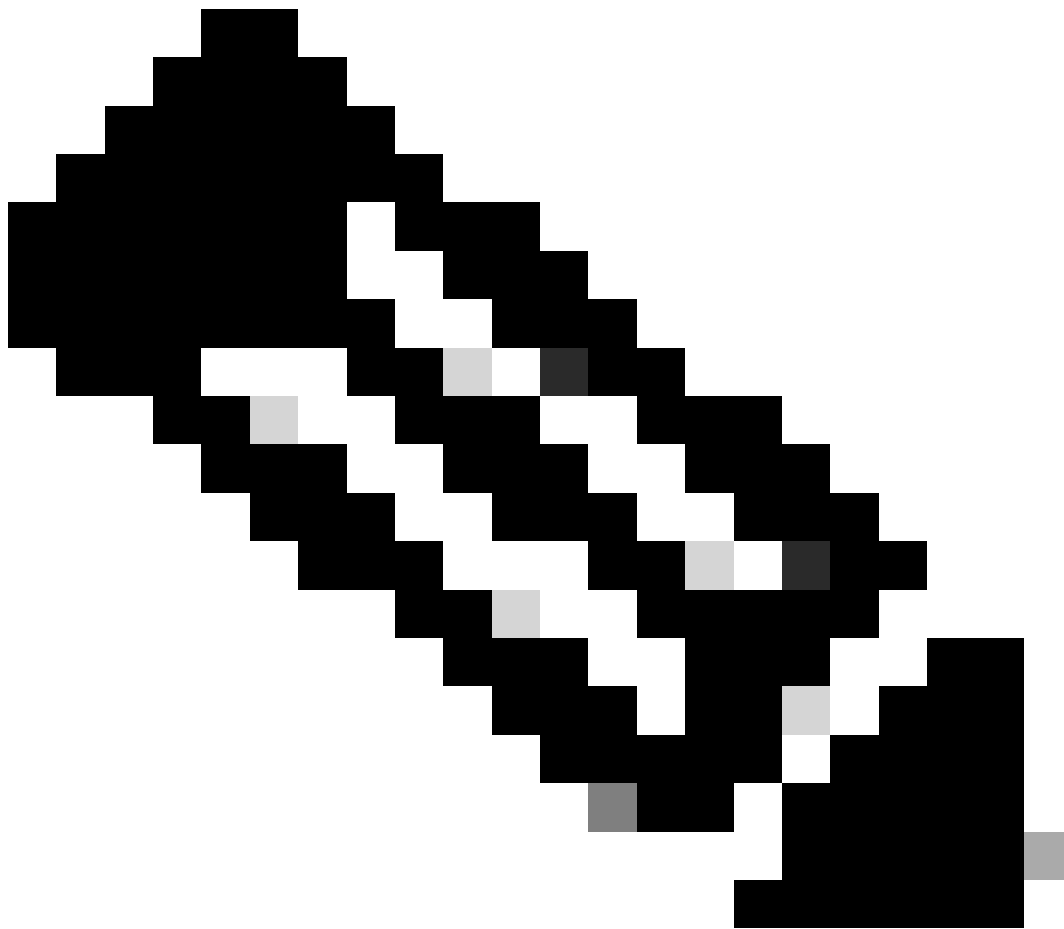
使用Desync命令下載ISO

#2：安裝DESYNC工具。執行完命令後，您可以注意到一堆下載提示。大約在2-3分鐘後，應完成下載。

```
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
```



注意：如果git命令不起作用，則可以從此處下載並安裝Git：<https://git-scm.com/download/win>。

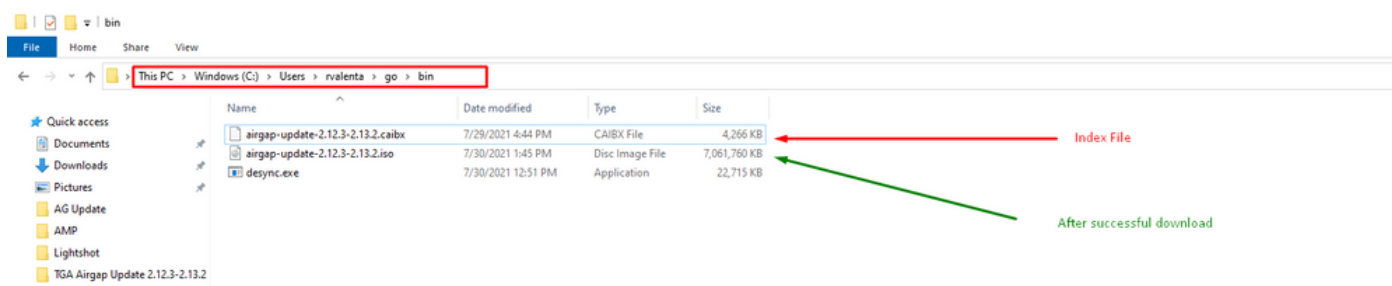
然後，逐一運行以下兩個命令：

```
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-eeec23a3978ad
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-runeewidth v0.0.9
go: downloading golang.org/x/sys v0.0.0-20201201145000-ef89a241ccb3
```

#3：導覽至執行—> bin位置。在我的案例中，它是C:\Users\rvalenta\go\bin，並在此複製/貼上TAC提供的.caibx索引檔案。



驗證

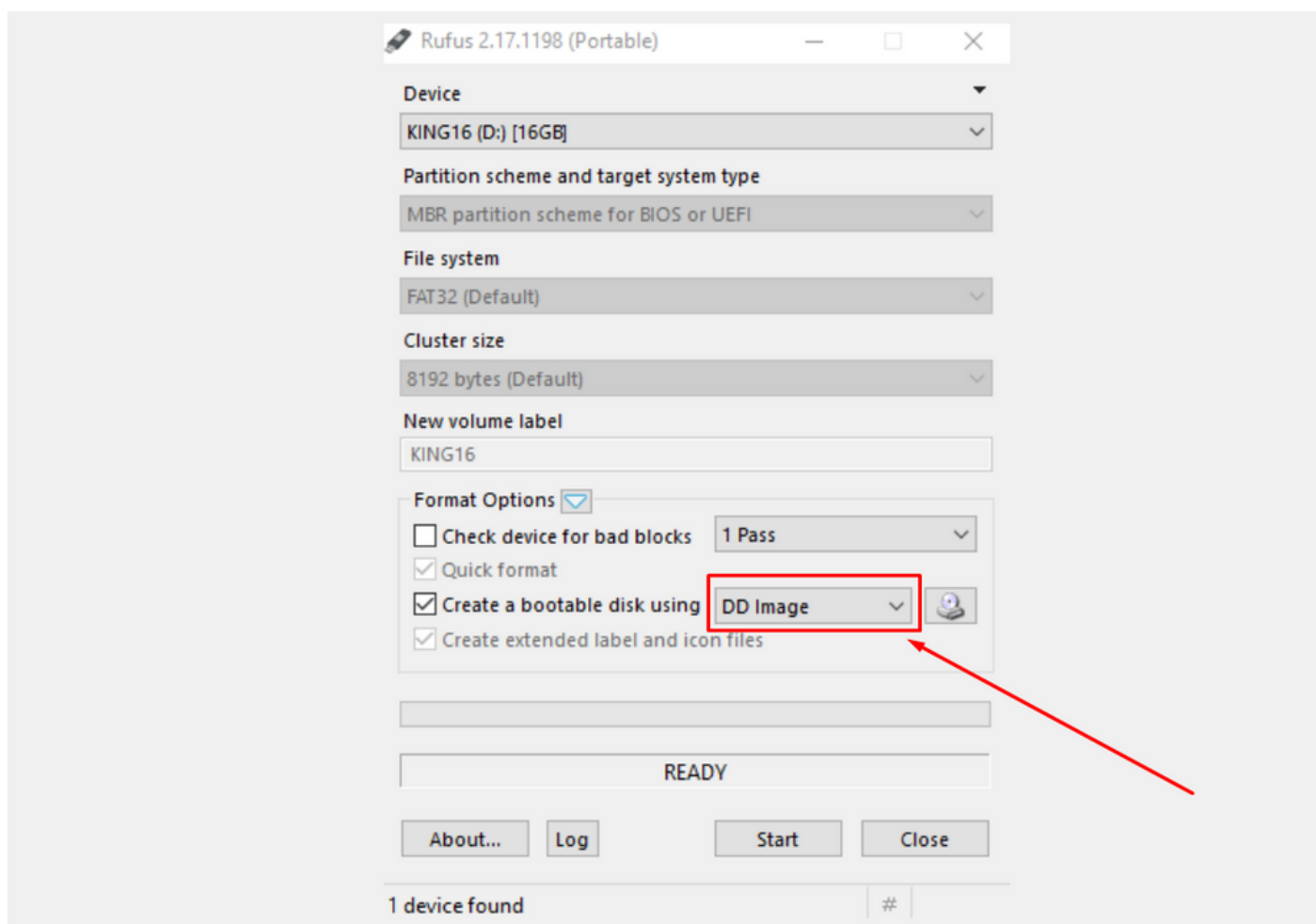
#4：返回到CMD提示符，導航到資料夾go\bin並運行下載命令。您應該會立即看到下載繼續。等待下載完成。現在，您應將整個.ISO檔案放置在先前複製的.caibx索引檔案的相同位置

```
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>
```

然後使用RUFUS建立可引導USB。使用版本2.17非常重要。這是可以使用dd選項的最後一個版本，該選項對於建立此特定恢復USB非常重要。您可以找到此儲存庫[RUFUS REPOSITORY](https://github.com/pbatard/rufus)的所有版本。如果這些檔案不再可用，我也會在本文檔中包含完整版本和可移植版本的安裝程式。



從USB啟動裝置

插入USB驅動器並打開或重新啟動裝置。在Cisco boot up畫面中，選取「F6」以進入Boot Menu。你一定要快！您只有幾秒鐘的時間可以完成此選擇。如果找不到，您必須重新開機，然後再試一次。

圖1 -按F6進入引導選單



```
Press <F2> Setup, <F6> Boot Menu, <F7> Diagnostics, <F8>Cisco IMC Configuration,  
<F12> Network Boot
```

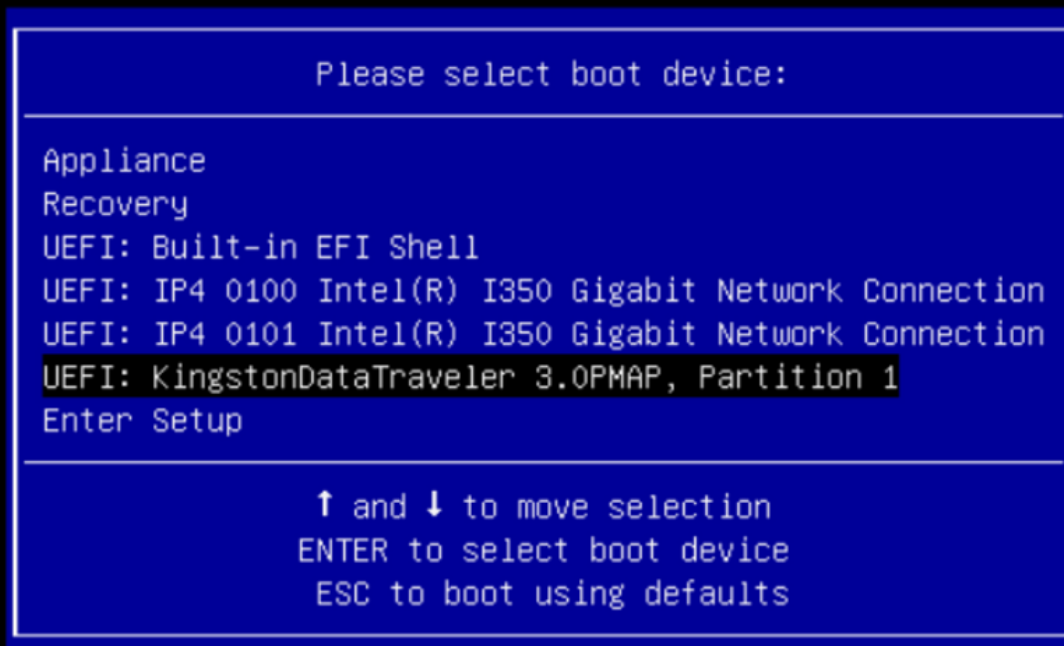
```
Bios Version : C220M4.2.0.13d.0.0812161113  
Platform ID : C220M4
```

```
Cisco IMC IPv4 Address : 10.77.1.71  
Cisco IMC MAC Address : CC:46:D6:FC:B5:1C
```

```
Processor(s) Intel(R) Xeon(R) CPU E5-2697 v4 @ 2.30GHz  
Total Memory = 512 GB Effective Memory = 512 GB  
Memory Operating Speed 2400 Mhz  
Entering boot selection menu...
```

導航到包含更新的USB驅動器，然後按Enter鍵選擇：

圖2 -選擇更新USB



更新媒體會決定升級路徑中的下一個版本，並將該版本的內容複製到裝置上。裝置可以立即運行升級，也可以重新引導至常規操作模式，以便進入OpAdmin並手動啟動升級。

一旦ISO啟動過程完成，請將Secure Malware Analytics裝置重新引導回操作模式。

登入門戶UI並檢查是否存在任何警告以說明升級是否安全等，然後繼續。

導航到OpAdmin介面並應用更新（如果在重新啟動過程中未自動應用）：OpAdmin > Operations > Update Appliance注意：更新過程包括其他重新啟動作為更新的一部分，該更新由USB介質提供。例如，安裝更新後，必須使用安裝頁面上的「重新開機」按鈕。

根據USB上的每個版本重複上述步驟。

如何查詢正確的/dev裝置

USB仍然未連線到終端時，運行命令「lsblk | grep -iE 'disk|part'」。

```
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$

```

連線USB線之後。

```

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$

```

這可以確認/dev中的USB裝置是「/dev/sdb」。

連線USB介面後，透過其他方法確認：

命令dmesg提供一些資訊。連線USB之後，運行命令dmesg | grep -iE 'usb|已連線'。

```

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$

```

命令fdisk提供有關大小的資訊，可用於確認：sudo fdisk -l /dev/sdb。

```
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$
```

 註：請記得在執行「dd」命令之前解除安裝USB。

確認示例中的USB裝置已裝入。

```
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,mask=0
```

要解除安裝USB裝置，請使用sudo umount /dev/sdb1。

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

重新檢查裝置是否被視為「已裝載」。

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

狀態=進度選項

oflag=sync和status=progress選項。


寫入大量資料區塊時，「status=progress」選項會提供目前寫入作業的相關資訊。這可用於確認「dd」命令當前是否正在寫入頁快取；它可以用於顯示所有寫入操作的進度和完整時間（以秒為單位）。

如果未使用，則「dd」不提供有關進度的資訊，在「dd」返回之前僅提供寫入操作的結果：

```
[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]$
```

使用時，每秒更新寫入操作的即時資訊。

```
[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]
```

 注意：在TGA離線升級過程的官方文檔中，所通知的命令為：`dd if=airgap-update.iso of=/dev/<MY_USB> bs=64M`

經過一些測試後，可觀察到以下示例。

使用裝置/dev/zero建立一個10MB大小且帶有「dd」的檔案後。

1M x 10 = 10M (10240 kB + 髒檔案頁快取中的上一系統資料= 10304 kB —>這是在「dd」末尾的髒頁快取中觀察到的內容)。

```
[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


```

---

 **注意：**返回「dd」命令後，對塊裝置的寫入操作未完成，在返回11秒後感覺到。如果這是使用TGA ISO建立可開機USB時的「dd」指令，並且我在這11秒之前已從端點移除USB =我在可開機USB中會有ISO損毀。

---

說明：

塊裝置提供對硬體裝置的緩衝訪問。這在使用硬體裝置時為應用程式提供了一層抽象。

區塊裝置允許應用程式以不同大小的資料區塊進行讀取/寫入；此讀取()/寫入()會套用至頁面快取（緩衝區），而不是直接套用至區塊裝置。

核心（而不是執行讀/寫操作的應用程式）管理資料從緩衝區（頁快取）到塊裝置的移動。

因此：

如果沒有指示，應用程式（在本例中為「dd」）無法控制緩衝區的刷新。

選項「oflag=sync」會在每個輸出區塊（由「dd」提供）置於頁面快取中之後，強制進行同步實體寫入（由核心執行）。

oflag=sync會降低與不使用選項相比的「dd」效能；但是，如果啟用，它將確保在每個「dd」的write()呼叫之後對塊裝置進行物理寫入。

測試：使用「dd」命令的「oflag=sync」選項，確認在「dd」命令返回後，已完成了所有包含頁面快取資料的寫入操作：

```

[rootuser@centos8-2 testarea]$ cat /proc/meminfo | grep -iE 'dirty' && dd if=/dev/zero of=testfile.txt
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]$


```

髒頁快取中的寫入操作未保留任何資料。

寫入操作是在「dd」命令返回之前（或同一時刻）應用的（而不是在上次測試之後的11秒）。

現在，我確信在返回「dd」命令後，髒頁快取中沒有與寫入操作相關的資料=建立可引導USB時沒有問題（如果ISO校驗和正確）。

---

 注意：處理此類情況時，請考慮使用「dd」命令的此標誌(oflag=sync)。

---

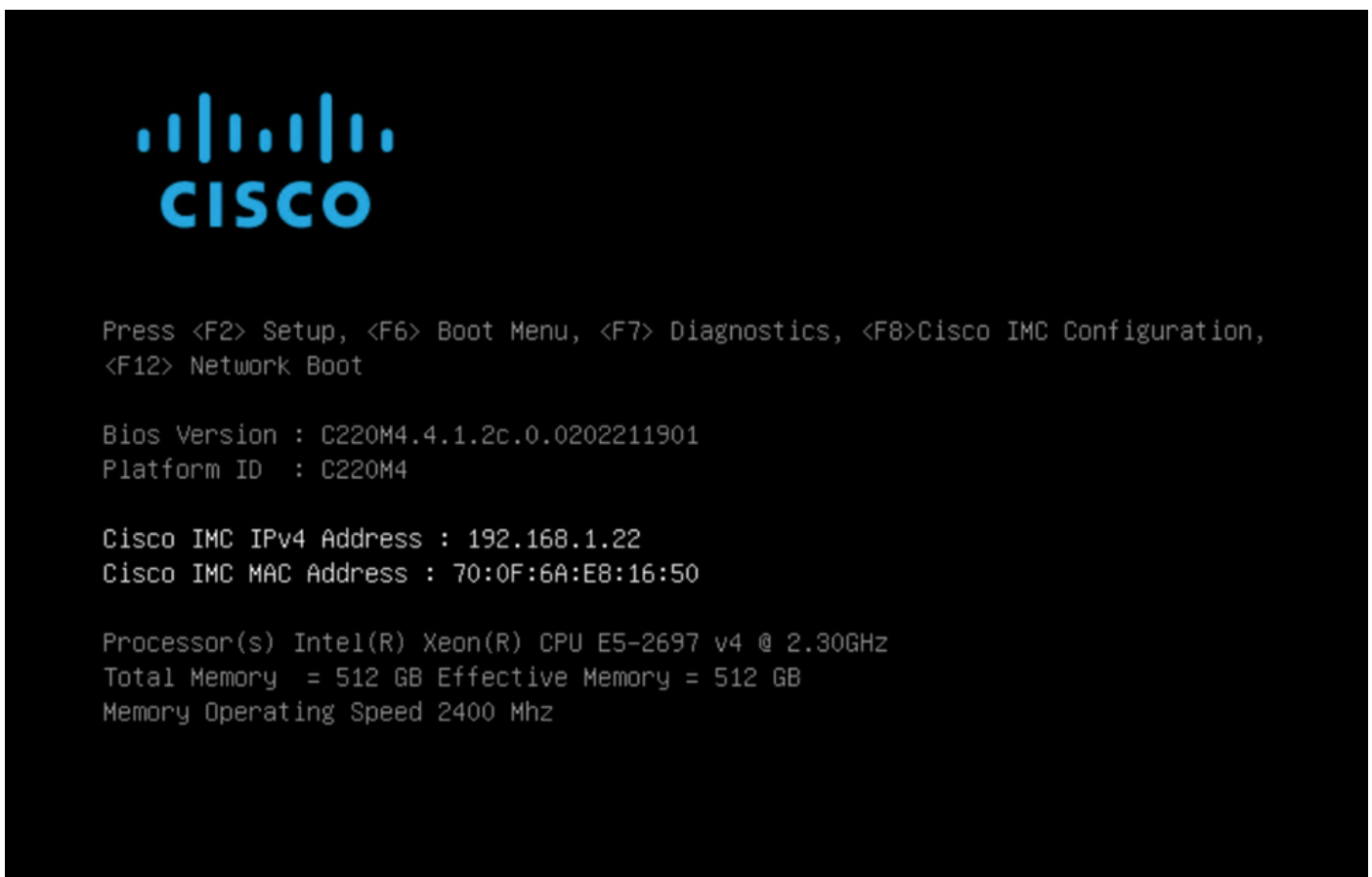
## 用於離線升級的硬碟機開機順序

需求：

我們需要確保使用「DD」選項使用任何可用工具格式化硬碟，之後應將介質複製到驅動器。如果不使用此格式，則無法讀取此媒體。

一旦使用「DD」格式將介質載入到HDD/USB上，我們就需要將其連線到TGA裝置並重新啟動裝置。

這是預設的「Boot Menu（引導選單）」選擇螢幕。我們需要按「F6」來引導裝置以選擇引導介質



一旦裝置辨識了我們的輸入，它將提示裝置進入引導選擇選單。





Press <F2> Setup, <F6> Boot Menu, <F7> Diagnostics, <F8>Cisco IMC Configuration,  
<F12> Network Boot

Bios Version : C220M4.4.1.2c.0.0202211901  
Platform ID : C220M4

Cisco IMC IPv4 Address : 192.168.1.22  
Cisco IMC MAC Address : 70:0F:6A:E8:16:50

Processor(s) Intel(R) Xeon(R) CPU E5-2697 v4 @ 2.30GHz  
Total Memory = 512 GB Effective Memory = 512 GB  
Memory Operating Speed 2400 Mhz  
Entering boot selection menu...

這是不同TGA型號之間可能不同的提示。理想情況下，我們可以從此選單本身看到使用引導介質（升級檔案系統）進行引導的選項，但如果看不到該選項，則需要登入到「EFI Shell」。

Please select boot device:

Appliance

Recovery

**UEFI: Built-in EFI Shell**

UEFI: IP4 0100 Intel(R) I350 Gigabit Network Connection

UEFI: IP4 0101 Intel(R) I350 Gigabit Network Connection

Enter Setup

↑ and ↓ to move selection  
ENTER to select boot device  
ESC to boot using defaults

您必須在「startup.sh」程式檔完成前按ESC鍵才能移入EFI Shell。登入EFI Shell後，我們會注意到在此情況下檢測到的分割槽為3個檔案系統：fs0：、fs1：、fs2。

```
UEFI Interactive Shell v2.0. UEFI v2.40 (American Megatrends, 0x0005000B). Revision 1.02
Mapping table
fs0: Alias(s):HD21a0b0c:;blk2:
 PciRoot(0x0)/Pci(0x1D,0x0)/USB(0x0,0x0)/USB(0x1,0x0)/HD(2,MBR,0x00000000,0xC6E244,0x9800)
fs1: Alias(s):HD29a0b:;blk4:
 PciRoot(0x0)/Pci(0x2,0x2)/Pci(0x0,0x0)/Ctrl(0x0)/Scsi(0x0,0x0)/HD(1,GPT,22C0970D-0F05-444F-A0F3-EA787035FA1E,0x800,0x4
00000)
fs2: Alias(s):HD29b0b:;blk8:
 PciRoot(0x0)/Pci(0x2,0x2)/Pci(0x0,0x0)/Ctrl(0x0)/Scsi(0x1,0x0)/HD(1,GPT,D4C95D76-AC65-421E-9BF9-487B6A2025ED,0x800,0x4
00000)
blk0: Alias(s):
 PciRoot(0x0)/Pci(0x1D,0x0)/USB(0x0,0x0)/USB(0x1,0x0)
blk1: Alias(s):
 PciRoot(0x0)/Pci(0x1D,0x0)/USB(0x0,0x0)/USB(0x1,0x0)/HD(1,MBR,0x00000000,0x40,0xC6E204)
blk3: Alias(s):
 PciRoot(0x0)/Pci(0x2,0x2)/Pci(0x0,0x0)/Ctrl(0x0)/Scsi(0x0,0x0)
blk7: Alias(s):
 PciRoot(0x0)/Pci(0x2,0x2)/Pci(0x0,0x0)/Ctrl(0x0)/Scsi(0x1,0x0)
blk5: Alias(s):
 PciRoot(0x0)/Pci(0x2,0x2)/Pci(0x0,0x0)/Ctrl(0x0)/Scsi(0x0,0x0)/HD(2,GPT,72DF22A3-D885-432E-A8D3-C1B00AB22A8B,0x400800,
0x400000)
blk6: Alias(s):
 PciRoot(0x0)/Pci(0x2,0x2)/Pci(0x0,0x0)/Ctrl(0x0)/Scsi(0x0,0x0)/HD(3,GPT,F298B3C8-074C-4D38-A346-74BEFB9D7F61,0x800800,
0xD5A6FDF)
blk9: Alias(s):
 PciRoot(0x0)/Pci(0x2,0x2)/Pci(0x0,0x0)/Ctrl(0x0)/Scsi(0x1,0x0)/HD(2,GPT,0D6976B4-70AE-4B36-8E8A-C7F8D322BFDE,0x400800,
0x2B9A8CFDF)
Press ESC in 3 seconds to skip startup.nsh or any other key to continue.
Shell> _
```

## 重要

辨識正確的檔案系統：

- 根據上述螢幕截圖，您將能夠看到「fs0：」是路徑中唯一帶有「USB」的介質，因此我們可以確信此檔案系統將包含引導介質（升級檔案系統）。

如果缺少檔案系統：

- 如果只有fs0：和fs1：可用，並且沒有fs2：，請驗證引導介質（升級檔案系統）是否以dd模式寫入並成功連線。
- 引導介質（升級檔案系統）應始終比恢復介質少一些，並且它們應始終緊挨著；需要確定的是，連線USB的驅動器是否位於末端開始的位置（因此，它位於fs0：的前面位置還是fs2：的後面位置）
- 在本例中，在下面的螢幕截圖中，是正確的「.efi」檔案，因為它位於「\efi\boot」分割槽下，並且具有「bootx64.efi」命名約定

```
Shell> fs0:
fs0:\> dir
Directory of: fs0:\
01/01/1980 00:00 <DIR> 2,048 efi
 0 File(s) 0 bytes
 1 Dir(s)
fs0:\> cd efi
fs0:\efi\> cd boot
fs0:\efi\boot\> dir
Directory of: fs0:\efi\boot\
01/01/1980 00:00 <DIR> 2,048 .
01/01/1980 00:00 <DIR> 2,048 ..
01/01/1980 00:00 18,703,096 bootx64.efi
 1 File(s) 18,703,096 bytes
 2 Dir(s)
```

若要在開機媒體（升級檔案系統）中啟動裝置，我們必須執行「bootx64.efi」檔案：

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

為了供您參考，我們已在下面顯示其他檔案系統的內容：

fs1：這是主引導檔案系統。

```

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 : 這是恢復映像引導檔案系統。

```
fs2:\> fs2:
fs2:\> dir
Directory of: fs2:\
09/21/2021 23:35 29,856 meta_contents.tar.xz
09/17/2021 13:01 <DIR> 4,096 tmp
10/26/2020 16:00 149 startup.nsh
05/23/2018 17:52 <DIR> 4,096 efi
09/17/2021 13:01 992,755,712 recovery.rosfs
 3 File(s) 992,785,717 bytes
 2 Dir(s)
fs2:\> cd efi
fs2:\efi\> cd Recovery
fs2:\efi\Recovery\> dir
Directory of: fs2:\efi\Recovery\
05/23/2018 17:52 <DIR> 4,096 .
05/23/2018 17:52 <DIR> 4,096 ..
09/10/2021 21:39 19,417,336 boot.efi
 1 File(s) 19,417,336 bytes
 2 Dir(s)
```

其他說明：

檢驗包含已裝載引導介質的正確檔案系統。我們可以透過瀏覽不同的檔案系統並驗證「.efi」引導檔案來執行此操作

---

 註：實際引導介質（升級檔案系統）的順序（在本例中為「fs0：」）可能因其他裝置而異。名稱和路徑可能不同，但在所有現代影像中，這應該是相同的。

---

有助於找到正確引導介質（升級檔案系統）的清單：

- 如果檔案系統的根包含「vmlinuz-appliance」，則它不是引導介質（升級檔案系統）。
- 如果檔案系統的根包含「meta\_contents.tar.xz」，則它不是引導介質（升級檔案系統）。
- 如果檔案系統不包含「efi\boot\bootx64.efi」，則它不是引導介質（升級檔案系統）。

## 關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。