更新安全恶意软件分析设备Air-Gap模式

目录

简介

本指南概述以气隙模式更新安全恶意软件分析设备的步骤。

注意：将设备以气隙模式维护会降低其效率。在继续操作之前，请考虑在安全性和功能之间进行权衡。

先决条件

Cisco 建议您了解以下主题：

- 在Windows和Unix/Linux环境中通过命令行输入的基本知识
- 恶意软件分析设备知识
- 思科集成管理控制器(IMC)知识

使用的组件

思科建议熟悉以下主题：

- 基于Windows 10和Linux的操作系统（例如：CentOS、RedHat）
- RUFUS 2.17
- C220 M4、M510和M520 M5、M610和M620 M6（设备型号）

本文档中的信息基于受控实验环境中采用默认配置的设备。如果您的网络处于活动状态，请小心谨慎并充分了解所有命令的潜在影响，然后再继续。

背景信息

大多数安全恶意软件分析设备都连接到互联网并使用在线更新流程。但是，某些设备严格在内部网络内维护（气隙）。思科不推荐这种方法，因为它会降低效率。本指南为必须维护气隙设备的用户提供离线更新流程。

对于离线安全恶意软件分析更新，思科会根据请求提供更新媒体。按照本文档中介绍的脱机更新流程进行操作。

媒体：Airgap（离线）更新媒体由安全恶意软件分析支持根据请求提供。它是ISO文件，可以复制到USB驱动器或HDD（具有足够的大小）。

大小:更新介质的大小根据支持的版本而有所不同，并会随着新虚拟机的引入而显著增加。 对于当前版本， 大小约为30 GB，包括取消同步工具， 为虚拟机相关更改启用增量更新。

升级启动周期：每次启动Airgap更新媒体时，它都会确定要升级到的下一个版本，并将与该下一个版本相关的内容复制到设备上。如果给定版本没有任何在设备运行时必须运行的前提条件检查，则该版本也可以启动软件包安装。如果版本包括此类检查或对更新过程中可能添加此类检查的部分进行覆盖，则更新实际上不会应用，直到用户登录到OpAdmin并使用OpAdmin > Operations > Update Appliance调用更新。

安装前挂钩：根据是否存在用于该特定升级的任何安装前挂钩，它或者立即运行升级，或者将设备重新启动回其常规操作模式，以允许用户进入常规管理界面并手动启动升级。

根据需要重复：每次这样的介质启动周期仅升级（或准备升级）到最终目标版本的一个步骤；用户必须根据需要多次启动，才能升级到所需目标版本。

限制

气隙更新不支持CIMC媒体。

由于使用的第三方组件的许可限制，在UCS M3硬件达到寿命终止(EOL)后，1.x版本的升级介质不再可用。因此，在EOL之前更换或升级UCS M3设备至关重要。

要求

迁移：如果所涵盖版本的版本说明包括必须在安装下一个版本之前进行迁移的情况，则用户必须在重新启动之前执行以下步骤，以避免其设备处于不可用状态。

注意：第一个版本比2.1.4版本新，特别是运行多个数据库迁移。在这些迁移完成之前继续操作是不安全的。有关详细信息，请参阅Threat Grid设备2.1.5迁移说明。

如果从2.1.3之前的版本开始，airgap升级媒体使用从单个许可证衍生的加密密钥，因此需要基于每个设备自定义。（唯一对用户可见的影响是，通过构建媒体来支持2.1.3之前的源版本，安全恶意软件分析需要事先在这些设备上安装许可证，而媒体将无法在其构建列表中未列出的任何设备上运行。）

如果从版本2.1.3开始或更晚，则airgap介质是通用的，不需要客户信息。

开始使用前

- 备份。在继续进行更新之前，必须考虑备份设备。
- 在计划更新到新版本之前，请查看该版本的发行版本注释进行更新，以验证是否需要任何后台迁移
- 验证设备的当前版本：OpAdmin > Operations > Update Appliance
- 查看内部版本号/版本查找表中的安全恶意软件分析设备版本历史记录，所有Threat Grid设备文档均提供此信息：版本说明、迁移说明、设置和配置指南以及管理员指南。

更新脱机(Airgapped)安全恶意软件分析设备

首先检查此页上可用的气隙版本：设备版本查找表

1. 打开TAC支持请求以获取离线更新媒体。此请求应包括设备序列号以及设备内部版本号。

2. TAC支持根据您的安装提供更新的ISO。

3. 将ISO映像刻录到可引导USB。请注意，USB是唯一受支持的离线更新设备/方法。

命名规则

这是更新的文件名，例如：TGA Airgap Update 2.16.2-2.17.2。

这意味着此媒体可用于运行最低版本2.16.2的设备并将设备升级到版本2.17.2。

限制

- 气隙更新不支持CIMC媒体。
- 由于使用的第三方组件的许可限制，在UCS M3硬件达到寿命终止(EOL)后，1.x版本的升级介质不再可用。因此，在EOL之前更换或升级UCS M3设备至关重要。

Linux/MAC - ISO下载

要求

Cisco 建议您了解以下主题：

- 可访问互联网的Linux计算机，用于下载ISO并创建可引导USB安装驱动器。
- Airgap下载说明由安全恶意软件分析支持提供。
- GO编程语言。下载
- .caibx索引文件（包含在TAC支持提供的zip文件中）。
- Desync工具（包含在安全恶意软件分析支持提供的zip文件中）。

使用的组件

本文档中的信息基于基于Linux的操作系统（例如：CentOS、RedHat）。

本文档中的信息基于受控实验环境中采用默认配置的设备。如果您的网络处于活动状态，请小心谨慎并充分了解所有命令的潜在影响，然后再继续。

配置

安装GO编程语言

```
# wget https://go.dev/dl/go1.23.1.linux-amd64.tar.gz
# tar -xzf go1.23.1.linux-amd64.tar.gz
# mv go /usr/local
```

安装后运行这三个命令（如果不运行，则desync命令会失败）

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

您可以通过以下方式验证GO版本：

```
# go version
```

使用Desync命令下载ISO

步骤1:将Secure Malware Analytics Support提供的Zip文件的内容，包括desync.linux和.caibx文件复制到计算机上本地的相同目录中。

第二步：切换到存储文件的目录：

示例：

```
# cd MyDirectory/TG
```

第三步：运行pwd命令以确保您位于目录内部。

```
# pwd
```

第四步：进入包含desync.linux 命令和.caibx文件的目录后，运行您选择的命令开始下载过程。

注意：以下是不同ISO版本的示例，请参阅安全恶意软件分析支持提供的说明中的.caibx文件。

对于版本2.16.2到2.17.2 ISO：

```
# desync extract -k -s s3+https://s3.amazonaws.com/sma-appliance-airgap-update airgap-update-2.16.2ag-2
```

对于版本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
```

对于版本2.5到2.7.2ag ISO：

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

下载开始后，会显示进度条。

注意：您的环境中的下载速度和升级介质的大小可能会影响合成ISO的时间。
请确保将下载文件的MD5与支持部门提供的捆绑包的MD5进行比较，以验证下载的ISO的完整性。

下载完成后，会在同一目录下创建ISO。

将USB插件到计算机并运行dd命令以创建可引导USB驱动器。

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

其中，<MY_USB>是USB闪存盘的名称（卸下尖括号）。

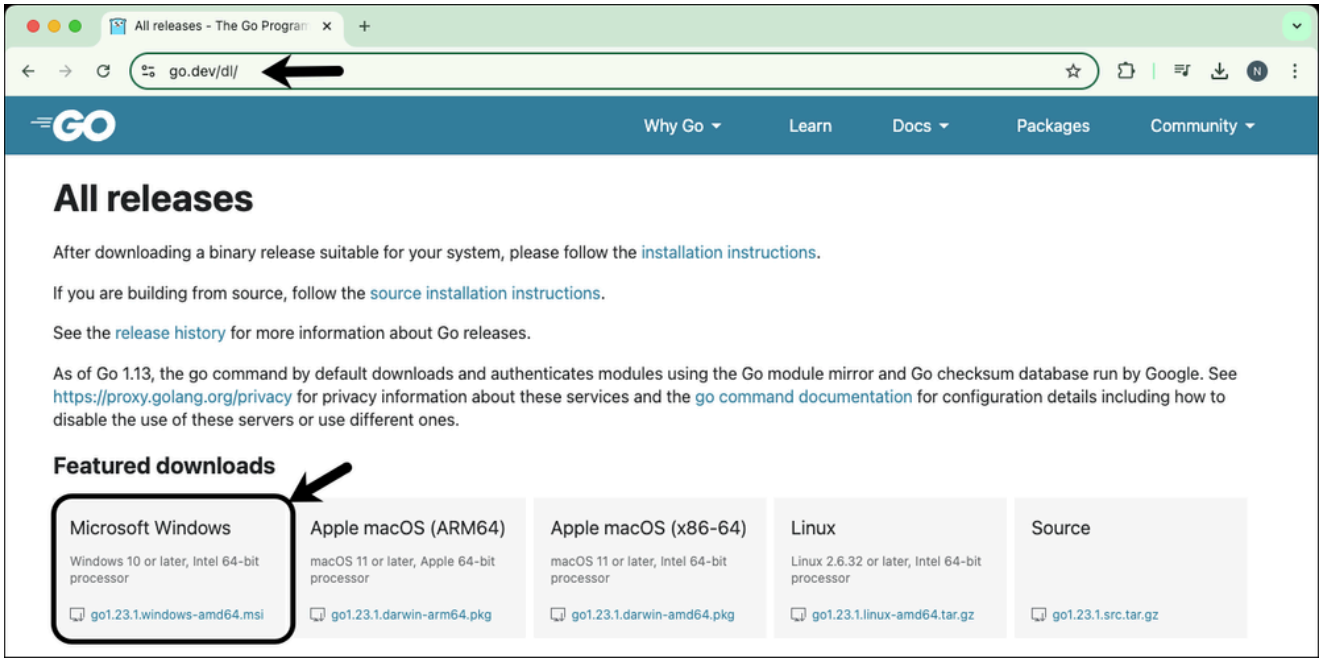
插入USB驱动器并打开或重新启动设备。在Cisco启动屏幕上，按F6以输入Boot Menu。

提示：
在下班后或非高峰时段运行下载，因为这可能会影响带宽。
要停止工具，请关闭终端或按Ctrl+c/Ctrl+z。
要继续，请运行同一命令以继续下载。

Windows - ISO下载

安装GO编程语言

1. 下载所需的GO编程语言。安装位置 https://golang.org/dl/ 就我而言，我选择特色版本。重新启动CMD并使用



关闭并重新打开CMD run命令以验证：

```
go version
```

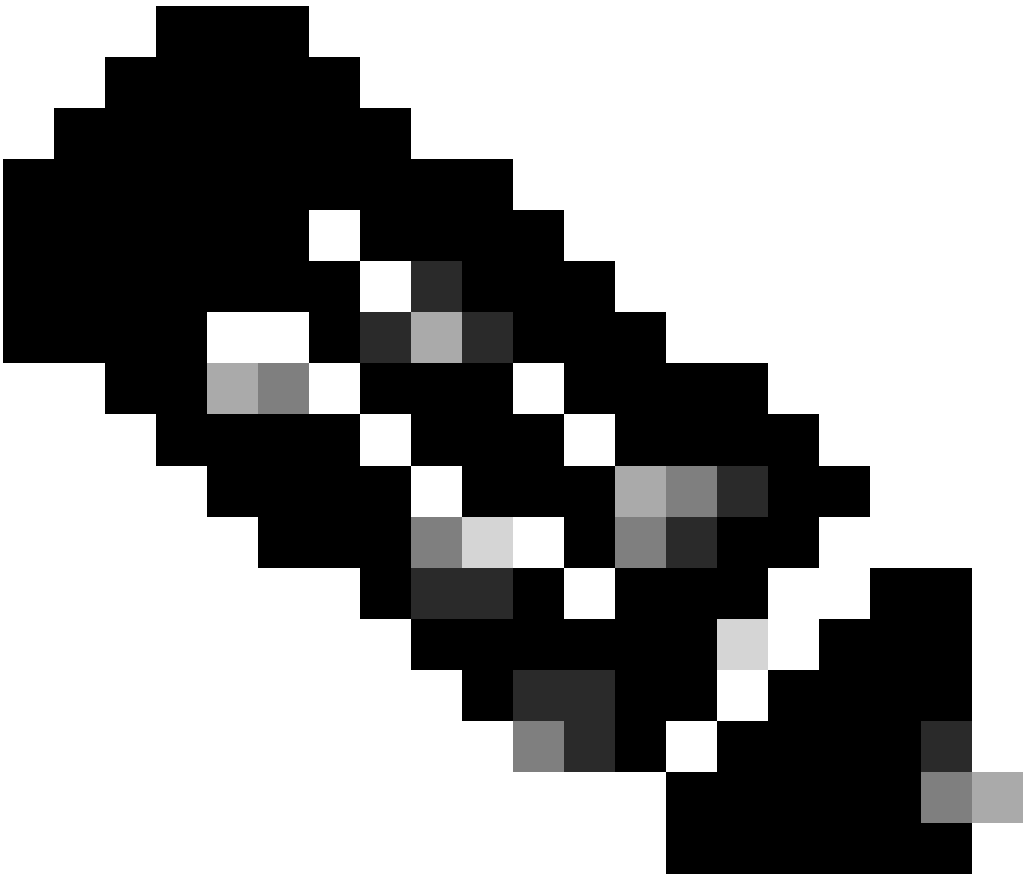
使用Desync命令下载ISO

2. 安装 DESYNC 工具。执行完该命令后，您会注意到大量下载提示。大约在2-3分钟后，应完成下载。

```
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 co
```

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



> 注意：如果git命令不起作用，则可以从此处下载并安装Git：https://git-scm.com/download/win。

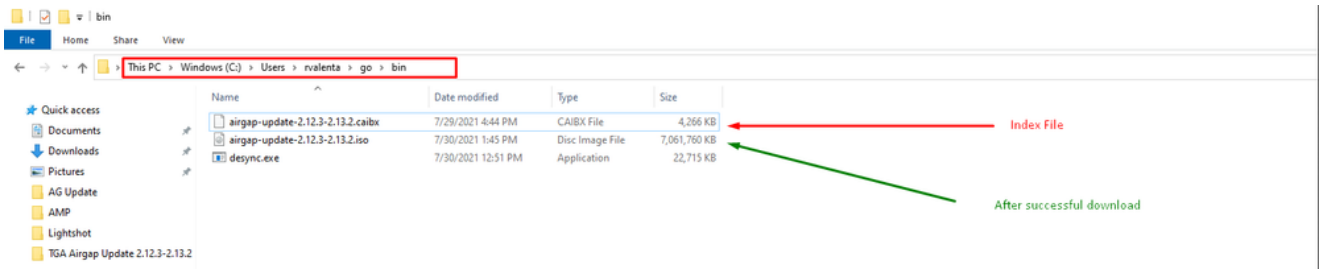
然后，逐一运行以下两个命令：

```
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-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. 导航至 go — >bin 位置.例如 C:\Users\<用户名>\go\bin 并复制/粘贴TAC提供的 .caibx 索引文件。

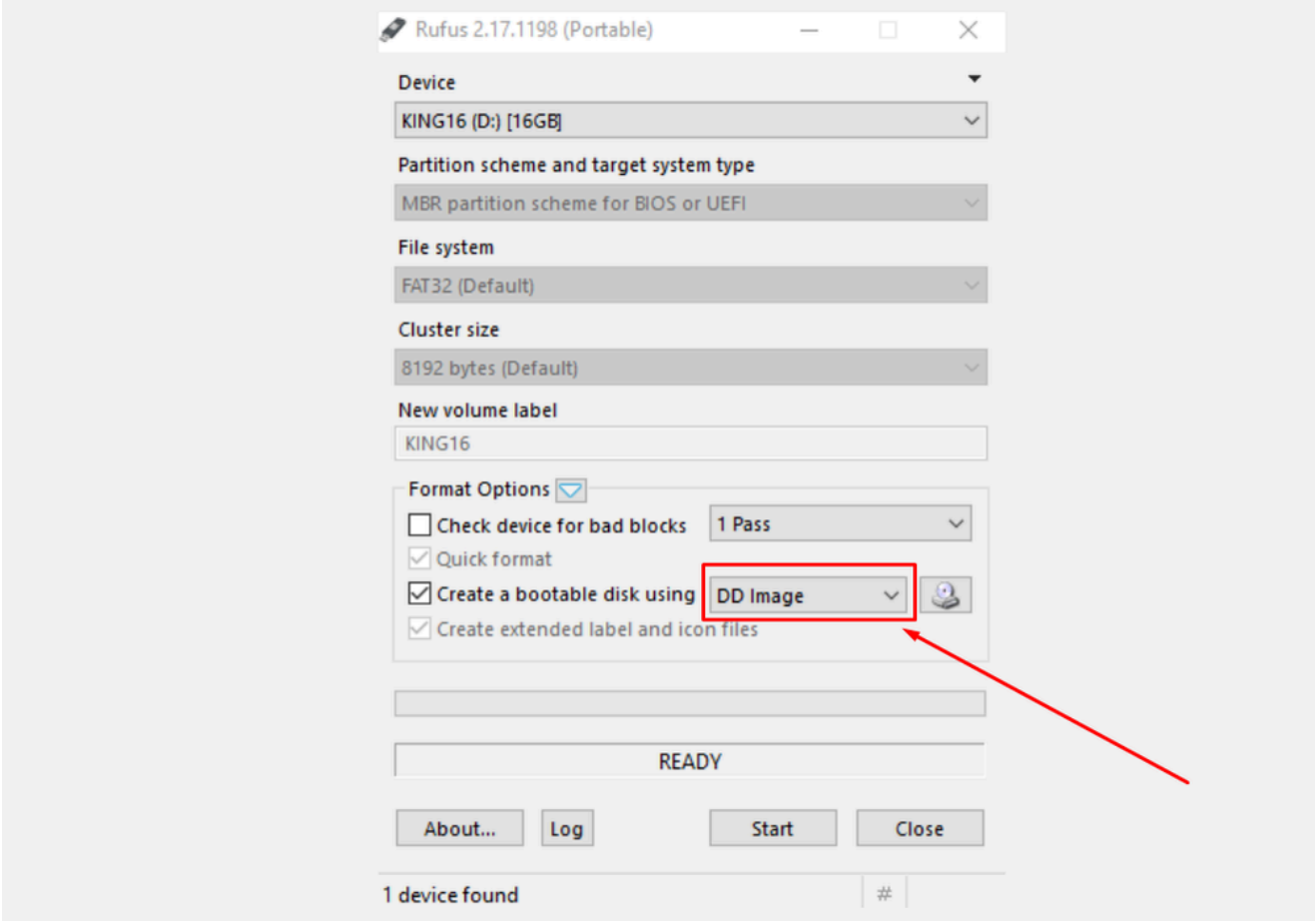


4. （验证）返回到CMD提示符，导航到文件夹go\bin并运行download命令。您应该会立即看到下载继续。等待下载完成。现在，您应该拥有整个 .ISO 与之前复制的文件位于同一位置 .caibx 索引文件

```
\$HOME/go/bin/desync extract -k -s s3+https://s3.amazonaws.com/sma-appliance-airgap-update airgap-
```

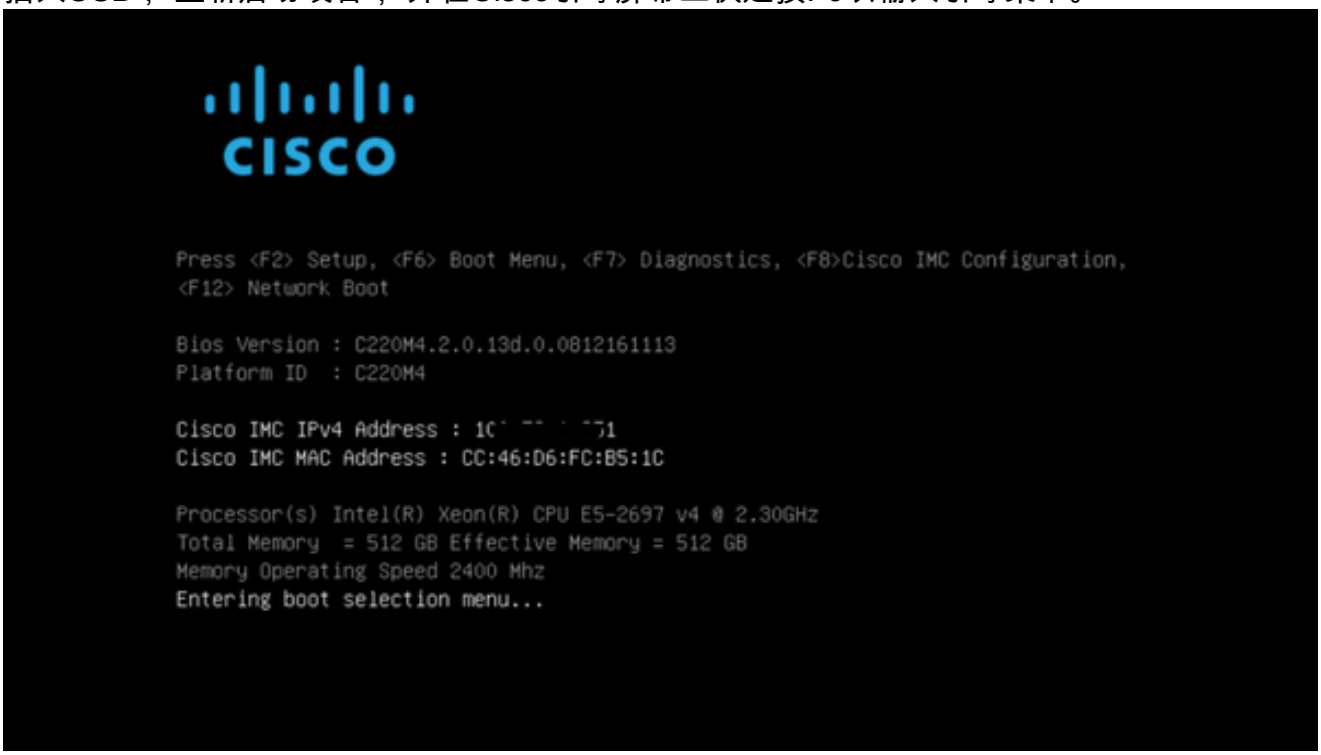
```
C:\Users\rvalenta>cd go

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>
```

要创建此特定恢复USB，使用Rufus版本2.17至关重要，因为它允许您使用基本的dd选项。您可以在此存储库中找到所有RUFUS版本。

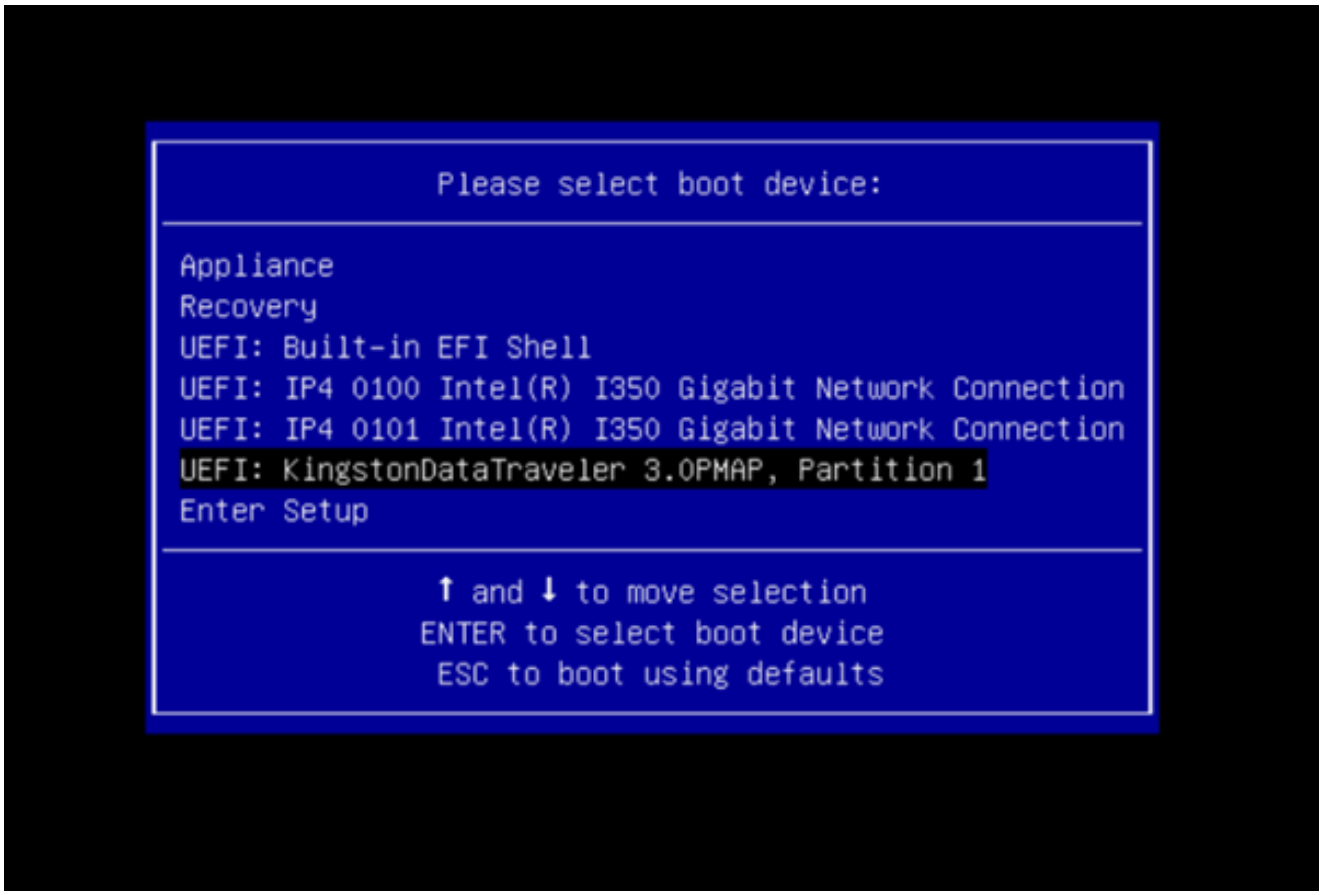


从USB启动设备

1. 插入USB， 重新启动设备， 并在Cisco引导屏幕上快速按F6以输入引导菜单。



2. 导航到包含更新的USB驱动器，然后按Enter选择。



更新媒体确定升级路径中的下一个版本，并将该版本的内容复制到设备上。设备会立即运行升级，或重新启动回其常规操作模式，以允许您进入OpAdmin并手动启动该升级。

完成ISO引导过程后，请将Secure Malware Analytics设备重新引导至操作模式。

登录门户UI并检查是否有任何警告，提示升级是否安全等，然后继续。

3. 导航到OpAdmin界面并应用更新（如果更新在重新启动期间未自动应用）：OpAdmin >操作>更新设备注意：更新过程包括其他重新引导作为更新的一部分，更新由USB介质产生。例如，安装更新后，需要使用安装页面上的Reboot按钮。
根据需要为USB上的每个版本重复上述步骤。

如何查找正确的/dev设备

USB未连接到终端时，运行命令“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$
```

在连接USB接口之后。

```
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$
```

这确认/dev中的USB设备为“/dev/sdb”。

其他确认方法，在连接USB接口后：

命令dmesg提供一些信息。连接USB之后，运行命令dmesg | grep -iE 'usb| 已连接'。

```
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fidsk提供有关大小的信息，可用于确认：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$
```

 注意：请记得在执行“dd”命令之前卸载USB。

确认示例中的USB设备已安装。

```
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
```

要卸载USB设备，请使用sudo umount /dev/sdb1。

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

重新检查设备是否被视为“已装载”。

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

status=progress选项

oflag=sync和status=progress选项。

写入大量数据块时，“status=progress”选项提供关于当前写入操作的信息。这在确认“dd”命令当前是否正在写入页缓存时非常有用；它可用于显示所有写入操作的进度和完整时间（以秒为单位）。

如果未使用，“dd”不提供进度信息，只在“dd”返回之前提供写入操作的结果：

```
[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]$
```

使用时，每秒更新关于写入操作的实时信息。

```
[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脱机升级过程的正式文档中，通知的命令是：dd if=airgap-update.iso of=/dev/<MY_USB> bs=64M

经过一些测试后，可看到以下示例。

使用设备/dev/zero创建一个10MB大小的文件(其中dd表示)。

1M x 10 = 10M(10240 kB +脏文件页面缓存中的上一个系统数据= 10304 kB —>这是“dd”末尾的脏页面缓存中感知到的数据)。

```
[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”命令后，对块设备的写入操作未完成，在返回11秒后会感觉到。
如果这是使用TGA ISO创建可引导USB时的“dd”命令，并且我在这11秒之前已从终端删除了USB =我在可引导USB中可能有损坏的ISO。

说明：

块设备提供对硬件设备的缓冲访问。这在使用硬件设备时为应用提供了一层抽象。

块设备允许应用程序按不同大小的数据块进行读/写；此read()/write()应用于页面缓存（缓冲区），而不是直接应用于块设备。
内核（而不是执行读/写操作的应用程序）管理数据从缓冲区（页面缓存）到块设备的移动。

因此：

如果未指示，应用程序（在本例中为“dd”）无法控制缓冲区的刷新。

选项“oflag=sync”强制在每个输出块（由“dd”提供）放入页面缓存后，进行同步物理写入（由内核执行）。
oflag=sync会降低未使用该选项时的“dd”性能；但是，如果启用该选项，它将确保在每个“dd”的write()调用之后对块设备进行物理写入。

测试：使用“dd”命令的“oflag=sync”选项，确认在“dd”命令返回时，已完成了包含脏页缓存数据的所有写入操作：

```
[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]$
```

脏页缓存中的写入操作没有保留任何数据。
写操作在“dd”命令返回之前（或同一时刻）应用（而不是在上次测试后11秒）。
现在，我确定返回“dd”命令后，脏页缓存中没有与写入操作相关的数据=创建可引导USB时没有问题（如果ISO校验和正确）。

注意：处理此类问题时请考虑使用“dd”命令的此标志(oflag=sync)。

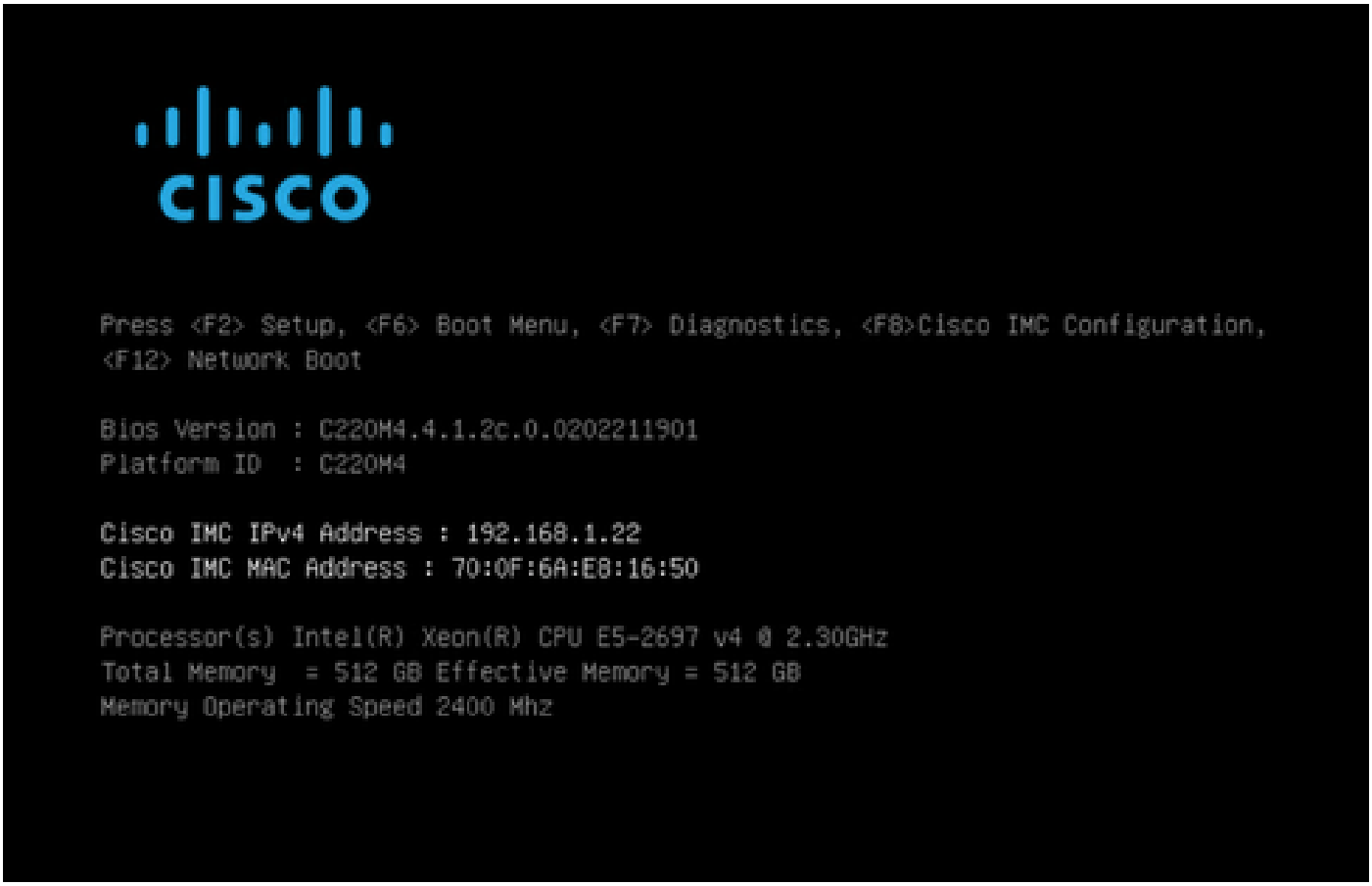
用于脱机升级的HDD驱动器的启动顺序

要求：

我们需要确保使用“DD”选项使用任何可用工具格式化硬盘，并且之后应将介质复制到驱动器。如果我们不使用此格式，我们就无法读取此介质。

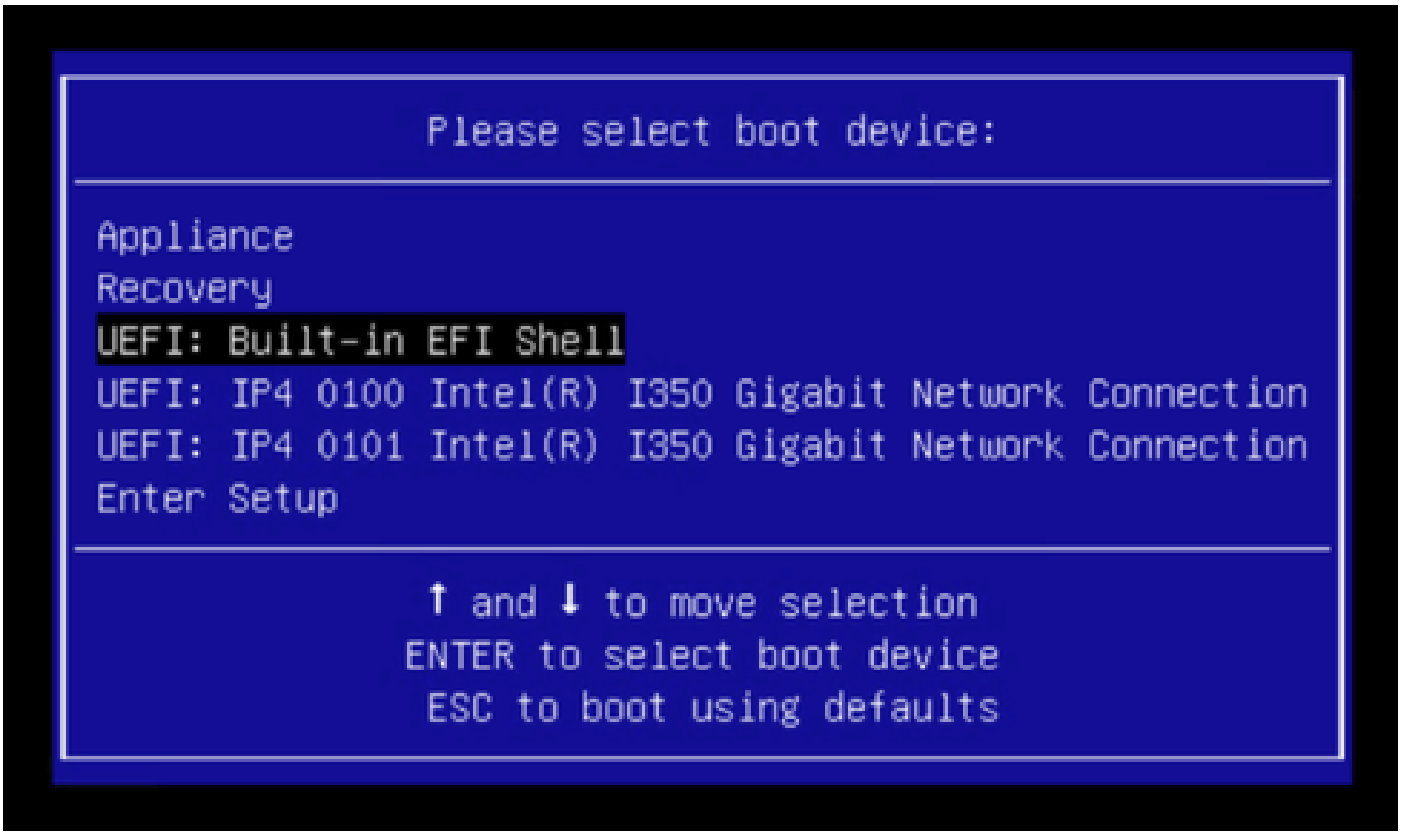
一旦使用“DD”格式将介质加载到HDD/USB上，我们需要将其连接到TGA设备并重新启动设备。

这是默认引导菜单选择屏幕。我们需要按“F6”启动设备以选择引导介质

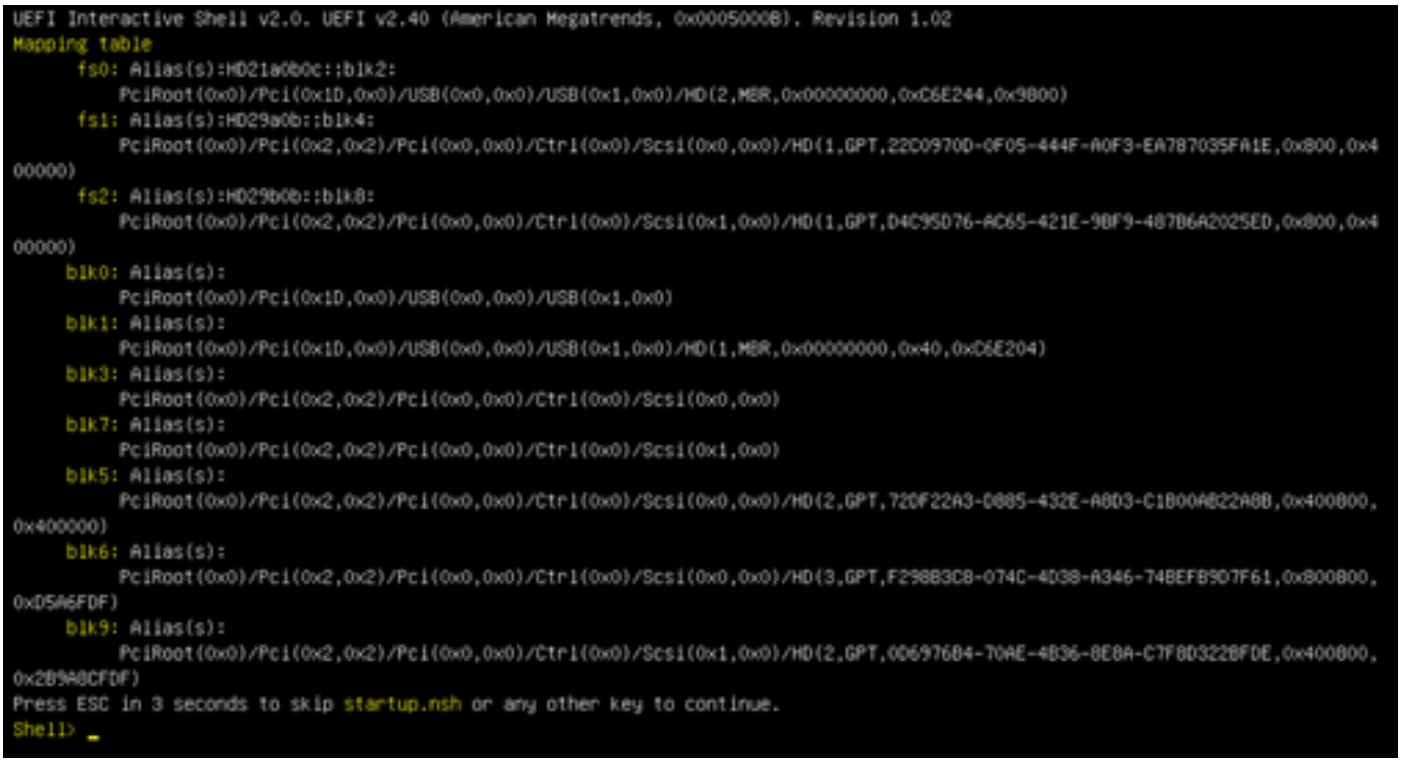


设备识别我们的输入后，会提示设备进入引导选择菜单。

这是不同TGA型号之间可能不同的提示。理想情况下，我们可以在此菜单中看到使用引导介质（升级文件系统）进行引导的选项，但如果看不到该选项，则需要登录到“EFI Shell”。



在“startup.sh”脚本完成之前，您必须按“ESC”才能进入EFI Shell。登录到EFI Shell后，我们会注意到在此情况下检测到的分区是3个文件系统：fs0：、fs1：、fs2。



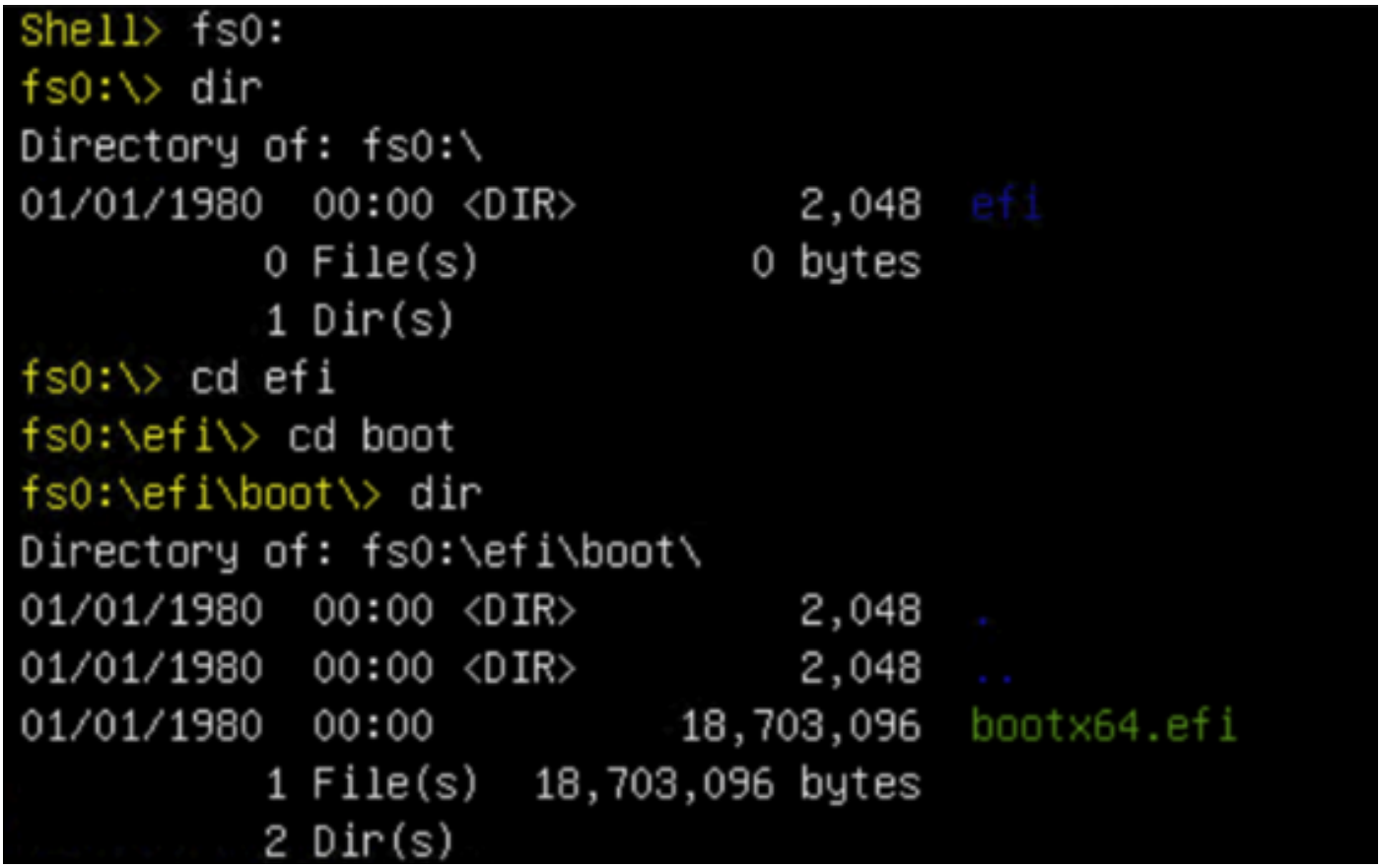
重要信息

识别正确的文件系统：

- 根据上述屏幕截图，您可以看到“fs0：”是在其路径中唯一具有“USB”的介质，因此我们可以确信此文件系统将包含引导介质（升级文件系统）。

如果缺少文件系统：

- 如果只有fs0：和fs1：可用，并且没有fs2：，请验证引导介质（升级文件系统）是否以dd模式写入并成功连接。
- 引导介质（升级文件系统）的数字应始终低于恢复介质，并且它们应始终紧挨着；需要确定通过USB连接的驱动器是否位于末端开始处(因此，它是fs0：的前位置还是fs2：的后位置)
- 在本例中，在下面的屏幕截图中，正确的是“.efi”文件，因为它位于“\efi\boot”分区下，且命名约定为“bootx64.efi”

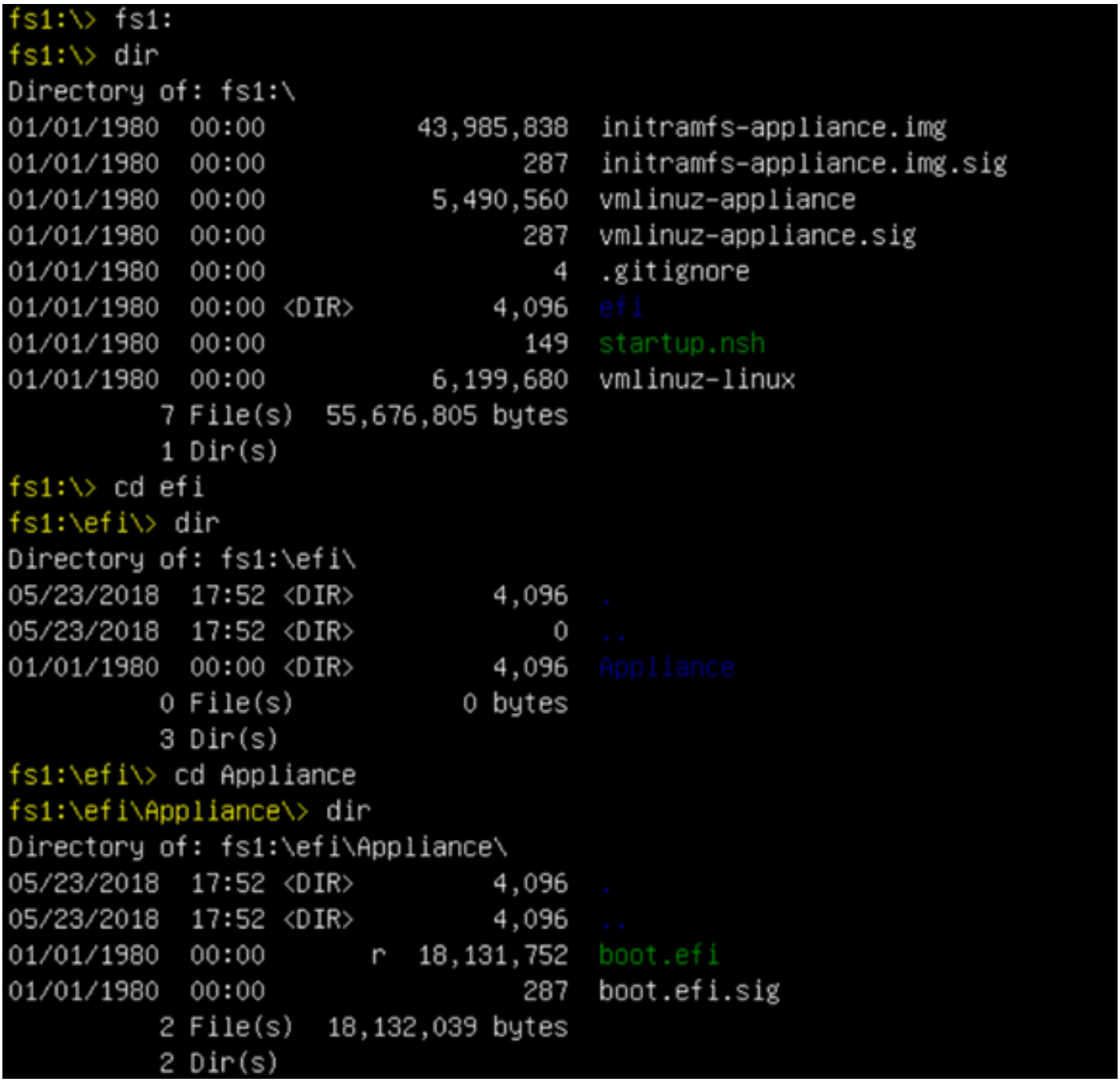


要在引导介质（升级文件系统）中引导设备，必须执行“bootx64.efi”文件：

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

下面还显示了其他文件系统的内容，供您参考：

fs1：这是主引导文件系统。



fs2：这是恢复映像启动文件系统。

```
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)
```

其他说明：

验证包含已装载的引导介质的正确文件系统。我们可以通过浏览不同的文件系统并验证“.efi”引导文件来完成此操作

✎ 注意：实际引导介质（升级文件系统）的顺序，在本例中为“fs0：”，也可能因其他设备而异。名称和路径可能不同，但在所有现代图像中，这应该是相同的。

有助于找到正确引导介质（升级文件系统）的核对表：

- 如果文件系统的根包含“vmlinuz-appliance”，则它不是引导介质（升级文件系统）。
- 如果文件系统的根目录包含“meta_contents.tar.xz”，则它不是引导介质（升级文件系统）。
- 如果文件系统不包含“efi\boot\bootx64.efi”，则它不是引导介质（升级文件系统）。

SMA现场安装程序2.19.2
对于损坏和/或无法修复的SMA设备，请使用现场安装程序重新安装SMA软件。请注意，此特殊软件包仅用于恢复目的。使用它进行升级可能会导致不可逆转的数据丢失。

恢复

在TGA卡住时进行恢复，一旦GATE提供此特殊映像，我们需要使用已知软件的特定版本，即RUFUS。RUFUS广泛用于创建可引导USB。对于此特定映像，我们需要使用RUFUS版本2.17。使用2.17版本非常重要。这是可以使用dd选项的最后一个版本，该选项对于创建此特定恢复USB非常重要。您可以找到此存储库Rufus存储库的所有版本，以防这些文件不再可用。本文档中还包含完整版本和可移植版本的安装程序。

RUFUS_217.zip的密码

Spoiler（突出显示以便阅读）
C1sco！123
C1sco！123

离线ISO TGA Airgap更新2.x-2.12.3ag2 MUST_RESET的特殊说明[airgap-update-MUST_RESET-2.12.3ag2]

如果您正在使用TGA Airgap Update 2.x-2.12.3ag2 MUST_RESET升级早于2.11.x的设备，则必须使用RESET data [data-destroy]才能使升级正常工作。

此airgap升级介质是一次性的，允许从非常旧的2.x版本直接升级到2.12.3ag2；它经过专门测试，可与2.2.3和2.5一起作为起始版本使用：比以上版本更新的版本很可能工作；比以上版本（但比2.0更新）的版本可能很有效。

- 从2.11.x之前的版本升级需要重置数据才能正常运行。这是因为常规升级过程涉及数据迁移，而下一次要版本之后将不再包括这些迁移。出于同样的原因，在2.11.x之前的版本上创建的备份可能无法恢复到此介质安装的内部版本中，或者可能在恢复后导致错误行为。

-如果从使用“/sandcastle”而不是“/data”进行批量存储（也就是说，从2.7以前的版本）的版本升级，则安装此内部版本后可能会出现一次性临时引导故障。发生这种情况时，系统的状态如与本自述文件位于同一目录中，名为“airgap-update-MUST_RESET-2.12.3ag2-filesystem-rename-hang-screenshot.png”的文件所示。如果此屏幕已显示超过15秒且未做任何更改，则重新引导系统是安全的。

脱机ISO索引文件包

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。