## 概要

ネットワーク・オペレーティング・システムのコマンドライン・インターフェイス（CLI）の基本的な働きは、20 年以上変わっていません。スイッチやルーターとやり取りするための主なメカニズムとして作成されて以降、CLI は変化していない、謎めいた存在のままです。ひとたび CLI をマスターすれば、設定や監視からトラブルシューティングや管理まで、さまざまな作業を実行できます。しかし、今日の高パフォーマンス・ネットワークの、急速に変化する要件をサポートし続けることはできません。

アリスタの EOS の設計には拡張性があるので、素早くカスタマイズして運用要件を満たすことができます。「オフボックス」のスクリプトやツールを作成したり、選んだベンダーが対応してくれるのを何年も待つ代わりに、EOS を簡単にカスタマイズできます。

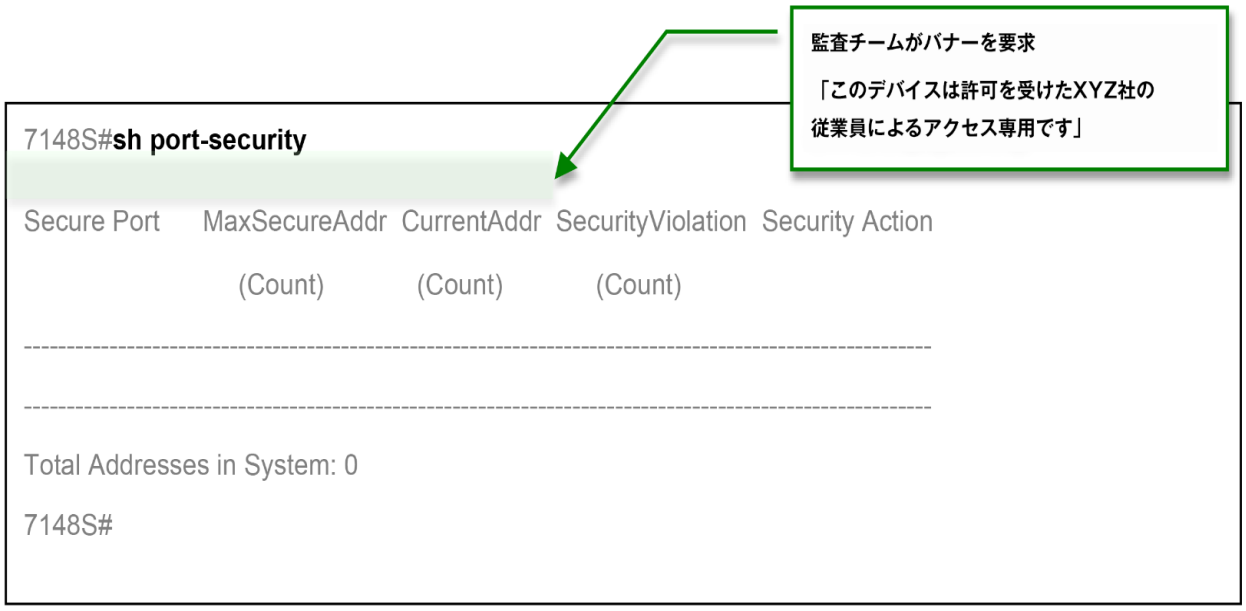
## はじめに

アリスタの EOS は、変化する運用ニーズを満たすためのカスタマイズが可能です。カスタマイズという言葉は、通常はネットワーク・オペレーシング・システムとは無縁です。以下のシナリオで、CLI の基本的な変更に必要なステップを順を追って説明します。

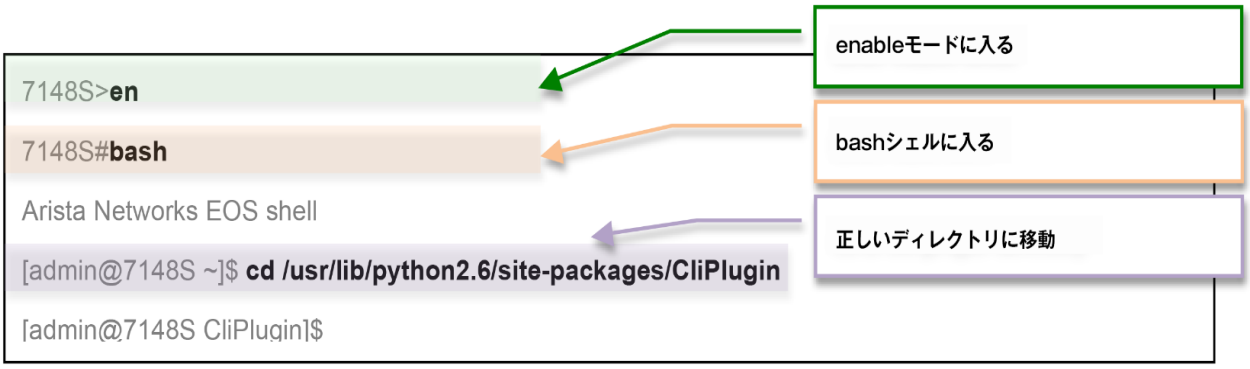
## シナリオ

XYZ 社の年次セキュリティ監査で、非常に望ましい機能として、監査人が ‘show port-security’ の出力に免責事項のバナーを付加する機能を挙げました。EOS を使用していれば、その対応も問題ありません。順を追って以下にステップを説明します。

まず、‘show port-security’ の出力と監査人の要望をご覧ください。



EOS のカスタマイズを実行するには、“bash”シェルに入り、CLI プラグイン・データを保持しているディレクトリに移動します。




**備考**: ここで、このディレクトリ内のファイルについて解説します。ファイルには、Arista EOS CLI を構成するコードが格納されています。‘ls –la *Cli*py’ を使用して出力を絞り込み、すべての関連ファイルのリストを取得することができます。通常、ファイル名は内容を説明するようなものになっています。/mnt/flash（または存在する場合、/mnt/usb1）に保存されていないファイルは、起動時に作成される RAM ディスクにあります。/mnt/flash 以外の場所にあるファイルに加えた変更はすべて、再起動時に失われます。変更を永続的にする方法については、後ほど説明します。

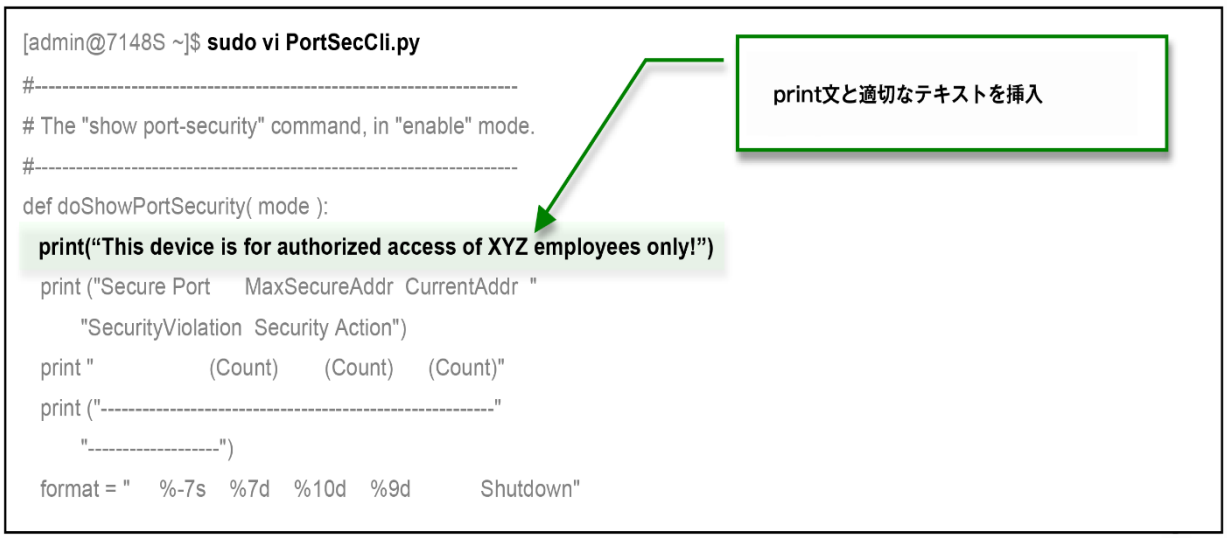
CLI は Python で書かれていて、提供する機能に合致した名前が付いています。ここでは、port security コマンドのコードを格納しているファイル、以下に赤色で強調表示されている PortSecCli.py を探します。

```
[admin@7148S CliPlugin]$ ls *Cli*py
AaaCli.py                   IntfCli.py              PimCli.py
AclCli.py                   IntfRangeCli.py         PortSecCli.py
AclCliRules.py              IntfSnmpCli.py          PowerCli.py
ActiveManagementIntfCli.py  IraIp6Cli.py           PowerDiagsCli.py
AgentCli.py                 IraIp6IntfCli.py        RadiusCli.py
AliasIntfCli.py             IraIpCli.py             RedSupCli.py
BackupIntfCli.py            IraIpIntfCli.py         ReloadCauseCli.py
BeaconLedCli.py             LagCli.py               ReloadCli.py
...                         ...                     ...
FileCli.py                  PeerIntfCli.py          VmTracerIntfCli.py
FocalPointCli.py            PfcCli.py               VrrpCli.py
FruCli.py                   PhyAelurosCli.py        WaitForWarmupCli.py
IgmpCli.py                  PhyCli.py
IgmpSnoopingCli.py          PhyConfigCli.py
```

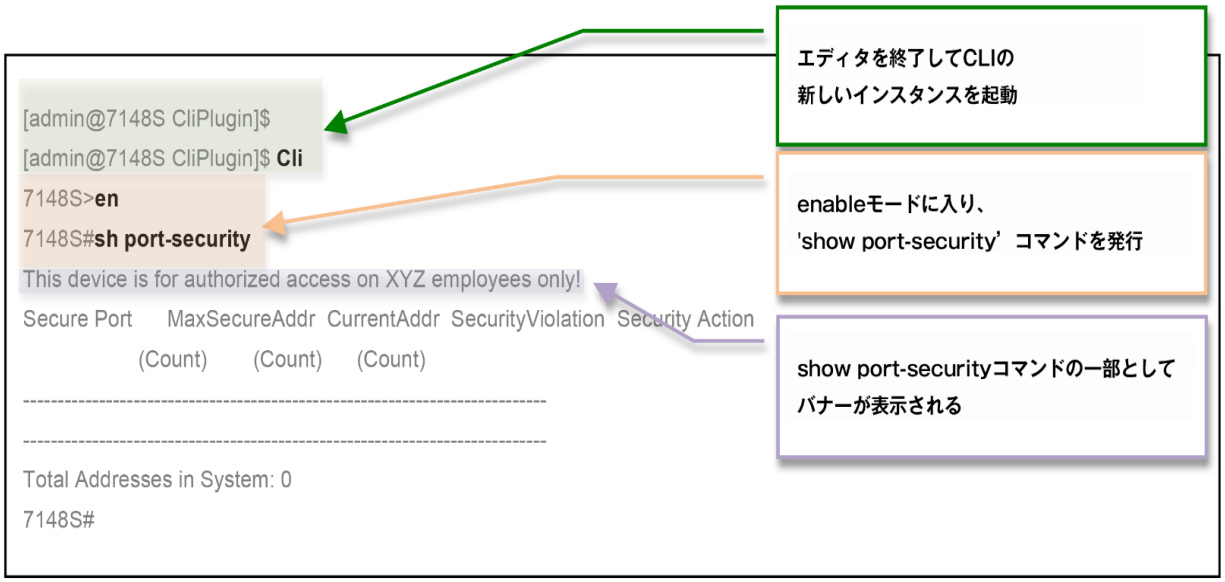
本書では'vi’ エディタを使用していますが、Emacs 風エディタである'zile'もインストールされています。

CliPlugin にあるファイルは root 以外のユーザーには読み取り専用なので、root としてファイルを開き、内容を確認して、‘show port-security’ コマンドのコードが見つかるまでスクロールします。


PortSecCli.py ファイルにコードを挿入します。



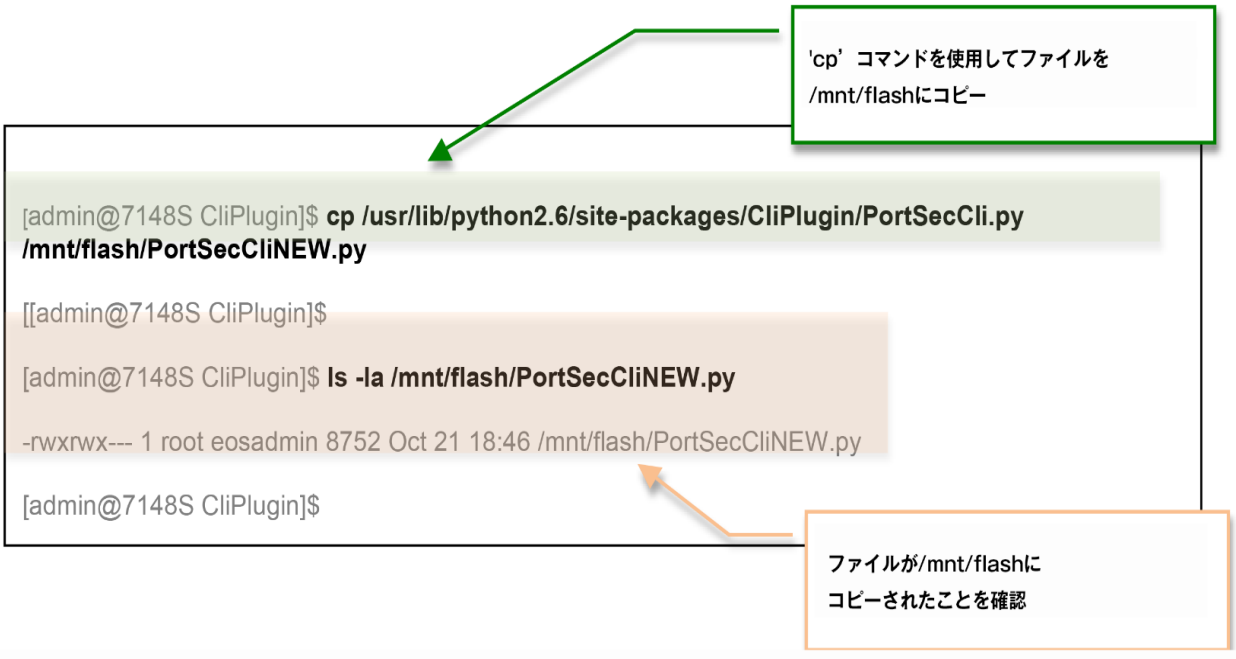
保存してエディタを終了し、新しい CLI セッションを開始して、show port-security コマンドを確認します。




EOS CLI への初めての変更が正常に完了しました。ご覧のように、重要な運用ニーズを満たすために変更を行うことができます。

## 最後の仕上げ

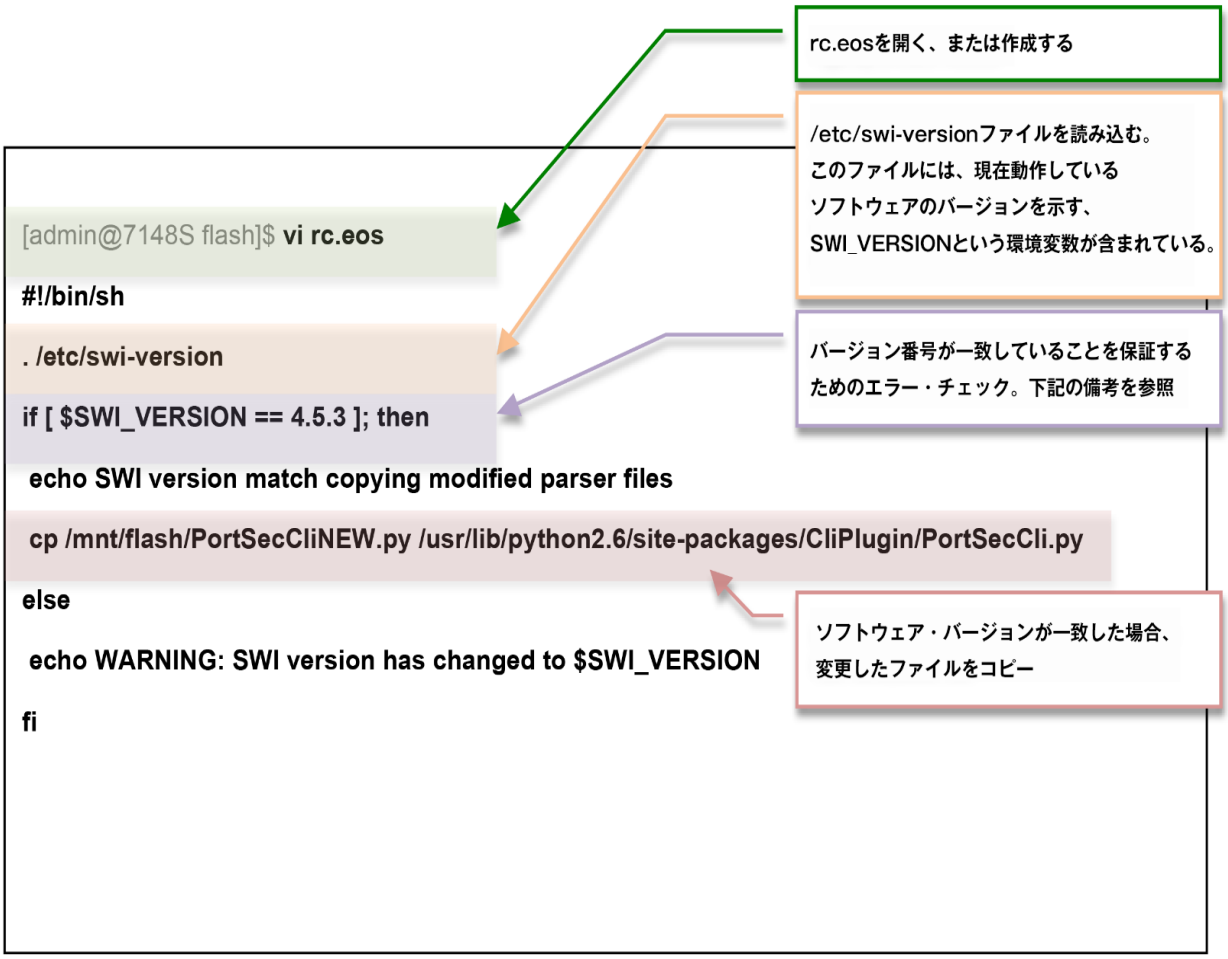
変更が完了したので、再起動しても変更が失われないようにする必要があります。/mnt/flash に保存されたファイル以外は、デバイスの起動時に作成され、マウントされる一時ファイル・システムにファイルとボリュームが格納されることを思い出してください。ファイルを/mnt/flash にコピーしないと、行った変更はスイッチの再起動時にすべて失われます。



ファイルは永続ストレージに保存されているので、起動プロセスの一環として、/usr/lib/python2.6/site-packages/CliPlugin に自動的にコピーされるようにするだけです。EOS の組み込み機能である rc.eos を利用すれば、この処理は簡単です。

**備考**:rc.eos は、起動プロセスの早い段階で、EOS の動作を変更するためのフックを提供します。

/mnt/flash の rc.eos ファイルを編集または作成し、bash スクリプトを使用して、変更した PortSecCli.py を /mnt/flash からもとの場所である/usr/lib/python2.6/site-packages/CliPlugin にコピーします。




CLI の変更は、変更が作成されたソフトウェア・バージョンに限定すべきです。限定しない場合、元になっている CLI が変わると、予期せぬ結果につながる可能性があります。アップグレードを実行する場合は、すべてのカスタマイズを確認し、新たに作成された CLI プラグイン内に作成しなおすことをお勧めします。

**備考**:アリスタは、お客様が新規またはカスタマイズしたエージェントやサービスをカスタマイズ、有効化、インストールすることを利点の１つとして許可しています。しかし、このようなカスタマイズに対するアリスタのサポートは「ベスト・エフォート」であることをご了承ください。システム・サポートの問題があり、カスタマイズが基本的なシステムまたはスイッチのフォワーディング動作に影響を与えていると考えられる場合は、カスタマイズを一時的に無効にするように依頼することがあります。