

index

- [管理および運用管理](#)
- [アプリケーションとサービス](#)
- [NDcPP 準拠証明書チェックを設定する](#)
- [DNSSEC セキュリティに関する推奨事項](#)
- [レガシー構成](#)
- [ロギングと監視](#)
- [LOM 設定](#)
- [システムおよびユーザーアカウント](#)
- [NetScaler® Gateway セキュリティに関する推奨事項](#)
- [ネットスケーラー® ウェブアプリファイアウォール セキュリティに関する推奨事項](#)

管理および運用管理

このセクションでは、NetScalerおよびNetScaler SDXのセキュリティを強化するために適用できる特定の構成変更の例を示します。NetScaler構成のベストプラクティスに関する詳細なガイダンスは、記事「NetScalerアプライアンスの一般的な実装に関する推奨設定とベストプラクティス」で確認できます。

詳細については、以下のトピックを参照してください。

- [システムおよびユーザーアカウント](#)
- [ログと監視](#)
- [LOM の構成](#)
- [アプリケーションとサービス](#)
- [DNSSECセキュリティに関する推奨事項](#)
- [レガシー構成](#)
- [NDCPP準拠証明書チェックの構成](#)

アプリケーションとサービス

NetScaler® を設定してアップグレードヘッダーを削除または渡す

バックエンドサーバーへの攻撃を防ぐため、新しいパラメータ `passProtocolUpgrade` が HTTP プロファイルに追加されました。このパラメータの状態に応じて、アップグレードヘッダーはバックエンドに送信されるリクエストで渡されるか、リクエストを送信する前に削除されます。

- `passProtocolUpgrade` パラメータが有効になっている場合、アップグレードヘッダーはバックエンドに渡されます。サーバーはアップグレードリクエストを受け入れ、その応答で通知します。
- このパラメータが無効になっている場合、アップグレードヘッダーは削除され、残りのリクエストはバックエンドに送信されます。

`passProtocolUpgrade` パラメータは以下のプロファイルに追加されます。

- `nshttp_default_profile` - enabled by default
- `nshttp_default_strict_validation` - disabled by default
- `nshttp_default_internal_apps` - デフォルトで無効に設定されています
- `nshttp_default_http_quic_profile` - enabled by default

`passProtocolUpgrade` パラメータを無効に設定することをお勧めします。

CLI を使用して `passProtocolUpgrade` パラメータを設定する:

コマンドプロンプトで、次のように入力します。

```
set ns httpProfile <name> [-passProtocolUpgrade ( ENABLED | DISABLED )]
```

GUI を使用して `passProtocolUpgrade` パラメータを設定する:

1. システム -> プロファイル -> HTTPプロファイル に移動します。
2. HTTPプロファイルを作成または編集します。
3. 「Pass Protocol Upgrade」チェックボックスをオンにします。

無効なHTTPリクエストを破棄するようにNetScalerを構成する

仮想サーバーを通過する無効なHTTPリクエストを防ぐため、NetScalerを厳密なHTTPリクエストのチェックと適用で構成することをお勧めします。そのためには、CLIで次のコマンドを使用し

て、組み込みのHTTPプロファイル `nshttp_default_strict_validation` を1つ以上の仮想サーバーにバインドします。

```
show ns httpProfile (Shows the available http profile (default+user
configured profiles))
```

```
set lb vserver <vserver name> -httpProfileName
nshttp_default_strict_validation
```

このオプションを使用するお客様は、本番環境にリリースする前に、ステージング環境で変更をテストすることをお勧めします。

HTTPデシンク攻撃に対する保護は、厳密なHTTP検証プロファイル (`nshttp_default_strict_validation`) でデフォルトで有効になっています。すべてのクライアント向けエンティティに厳密なプロファイルを使用してください。

HTTPリクエスト密輸攻撃とその軽減策の詳細については、サポート記事「[NetScaler - HTTP Request Smuggling Reference Guide](#)」を参照してください。

HTTPサービス拒否攻撃に対する保護を構成する

NetScalerファームウェアは、「スローリード」タイプの攻撃を含むHTTPサービス拒否攻撃に対する限定的な対策をサポートしています。これらの機能は、NetScalerシェルプロンプトから `nsapimgr` ユーティリティを使用して構成できます。

- スモールウィンドウしきい値 (デフォルト=1)
- 小ウィンドウアイドルタイムアウト (デフォルト=7秒)
- 小ウィンドウクリーンしきい値 (デフォルト=100)
- 小ウィンドウ保護 (デフォルト=有効)

デフォルト設定は、スローリード攻撃を含むHTTPサービス拒否攻撃を防ぐのに十分ですが、他の攻撃に対してはパラメータの調整が必要になる場合があります。

このような攻撃から保護するには、NetScalerシェルプロンプトから次の `nsapimgr` コマンドを使用して、`small_window_threshold` プロパティを上方調整します。

```
$ nsapimgr -ys small_window_threshold=<desired value>
```

注:

`small_window_threshold` の希望する値は、展開における受信トラフィックパターンに基づいて設定できます。許容範囲は0から 2^{32} です。

NetScalerシェルプロンプトから `nsconmsg -d stats` コマンドを使用して、次のカウンターを監視することで、HTTPサービス拒否攻撃に対する保護を確認できます。

- `nstcp_cur_zero_win_pcb`: このカウンターは、現在ウィンドウ値が低いPCBの数を追跡します。
- `nstcp_err_conndrop_at_pass`: このカウンターは、NetScalerがパケットを一方から他方へ通過させている間に、`nscfg_small_window_idletimeout`値を超過したことを検出したときにインクリメントされます。
- `nstcp_err_conndrop_at_retx`: このカウンターは、再送信中に経過する時間が `nscfg_small_window_idletimeout`値を超過したときにインクリメントされます。
- `nstcp_cur_pcb`: このカウンターは、KAプローブでプローブされたサージキュー内のPCBの数を追跡します。

このオプションを使用するお客様は、本番環境にリリースする前に、ステージング環境で変更をテストすることをお勧めします。

TCPスプーフィング攻撃から防御するようにNetScalerを構成する

バックエンドサーバーをTCPスプーフィング攻撃から保護するために、次のコマンドを使用できます。

```
set ns tcpProfile profile1 -rstWindowAttenuate ENABLED -spoofSynDrop
ENABLED
```

Done

```
set lb vserver lbvserver1 -tcpProfileName profile1
```

Done

このオプションを使用するお客様は、本番環境にリリースする前に、ステージング環境で変更をテストすることをお勧めします。

RFC 5961準拠を構成する

NetScalerリリース14.1-4.x以降、TCPプロファイルで `-rfc5961Compliance` オプションが利用可能です。このオプションにはスプーフィング保護が含まれており、`-rstWindowAttenuate` と `-spoofSynDrop` の両方をカバーします。各オプションを個別に設定することなく、これらすべての保護を得るには `-rfc5961Compliance` を有効にします。

詳細については、「[Sample TCP Configurations](#)」を参照してください。

特定のHTTPヘッダーを受け入れるようにNetScalerを構成する

NetScalerが特定のHTTPヘッダーのみを受け入れるように構成できます。これは、特定の定義済みHTTPヘッダーを持つネットワークトラフィックがバックエンドサーバーに渡されないように制限する書き換えアクションを追加することで実現できます。

次のグローバル書き換えアクションは、Host、Accept、testなどのヘッダーを持つネットワークトラフィックのみをサーバーに送信します。

```
add rewrite action act1 replace_all q/  
HTTP.REQ.FULL_HEADER.after_str("\r\n")/      q{TARGET.REGEX_SELECT(re/  
(iu)^(Host|Accept|test):.*\r\n/) ALT ""} -pattern q{re/(U).+:\r\n/}  
  
add rewrite policy pol1 HTTP.REQ.IS_VALID act1  
  
bind rewrite global pol1 100
```

HTTPヘッダーのヘッダーサイズを構成する

`maxHeaderLen` および `maxHeaderFieldLen` HTTPヘッダーサイズパラメーターには、ビジネスニーズに適した値を維持することをお勧めします。デフォルト値は24820に設定されており、これは大きなHTTPヘッダーを必要とするアプリケーションとの互換性を保証します。

```
set httpprofile http-prof -maxHeaderLen <header-length> -maxHeaderFieldLen
<header-field-length>
```

`maxHeaderLen` パラメーターはヘッダー全体のサイズを制御し、`maxHeaderFieldLen` は個々のヘッダーフィールドのサイズを制御します。これらのパラメーターは、バッファオーバーフローやヘッダーインジェクション攻撃などのヘッダーベースの悪用に対する攻撃対象領域を制限することで、セキュリティを強化するために利用できます。ヘッダーサイズは、仮想サーバーに構成されているHTTPプロファイル、およびサービスグループまたは仮想サーバーにバインドされているサービスに設定します。

これらのパラメーターをより低い値に構成することは可能ですが、そうすると、アプリケーションが構成された制限を超えるヘッダーを使用している場合に、正当なリクエストが破棄される可能性があります。したがって、管理者はこれらの値を下げる前に、アプリケーションの要件を慎重に評価する必要があります。

クローズノーティファイを設定する

close-notifyは、SSLデータ転送の終了を示す安全なメッセージです。RFC 5246に準拠して、クライアントとサーバーは、切断攻撃を回避するために接続が終了することを共有する必要があります。どちらの当事者も、close_notifyアラートを送信することで、終了メッセージの交換を開始できます。クローズアラートの後に受信されたデータは、他の致命的なアラートが送信されていない限り無視され、各当事者は接続の書き込み側を閉じる前にclose_notifyアラートを送信する必要があります。TLS終了イベントの監査イベントがキャプチャされるようにするには、スーパーユーザーまたはsysadminとしてCLIにログオンし、次のコマンドを実行します。

```
set ssl parameter -sendCloseNotify y
```

```
save ns config
```

管理GUI、NITRO API、およびRPC通信を保護する

NetScalerおよびNetScaler Gateway上の管理GUI、NITRO API、およびRPC通信を保護するには、内部HTTP/HTTPSサービス用にNetScalerで `-maxclient` を設定する必要があります。

```
set service nshttpd-gui-127.0.0.1-80 -maxclient $max_client{}
```

```
set service nshttpd-vpn-127.0.0.1-81 -maxclient $max_client{}
```

```
set service nshttps-127.0.0.1-443 -maxclient $max_client{}
```

`-maxclient` の値の設定、およびこの設定をサポートするNetScalerのバージョンに関する詳細については、[NetScaler GUI](#)、[NetScaler CLI](#)、または[NetScaler NITRO API](#)を使用して内部HTTP GUIサービスを構成する を参照してください。

NDcPP 準拠証明書チェックを設定する

NDcPP 準拠証明書チェックは、NetScaler がクライアント (バックエンド接続) として機能する場合に適用されます。証明書の検証中に、SSL 証明書に SAN が存在する場合は、コモン名を無視します。

コマンドプロンプトで、SSL パラメータの「ndcppComplianceCertCheck」属性を設定するには、次のコマンドを入力します。

```
set ssl parameter [-quantumSize <quantumSize>] [-crlMemorySizeMB
<positive_integer>] [-strictCAChecks (YES | NO)] [-sslTriggerTimeout
<positive_integer>] [-sendCloseNotify (YES | NO)] [-encryptTriggerPktCount
<positive_integer>] [-denySSLReneg <denySSLReneg>] [-insertionEncoding
(Unicode|UTF-8)] [-ocspCacheSize <positive_integer>][- pushFlag
<positive_integer>] [- dropReqWithNoHostHeader (YES | NO)] [-
pushEncTriggerTimeout <positive_integer>] [-ndcppComplianceCertCheck ( YES
| NO)] [-heterogeneousSSLHW (ENABLED | DISABLED )]
```

例:

```
set ssl parameter -quantumSize 8 -crlMemorySizeMB 256 -strictCAChecks no
-ssltriggerTimeout 100 -sendClosenotify no -encryptTriggerPktCount 45
-denySSLReneg NONSECURE -insertionEncoding unicode -ocspCacheSize 10
-pushFlag 3 -dropReqWithNoHostHeader YES -pushEncTriggerTimeout 100 ms
-ndcppComplianceCertCheck YES
```

DNSSEC セキュリティに関する推奨事項

DNSSEC を使用するお客様には、以下の推奨事項を適用することをお勧めします。

KSK/ZSK プライベートキーには RSA 1024 ビット以上を使用する

NIST は、DNS 管理者が 2015 年 10 月 1 日まで 1024 ビットの RSA/SHA-1 または RSA/SHA-256 ZSK を維持することを推奨しています。

DNSSEC キーの有効期限切れに対する SNMP アラームを有効にする

デフォルトでは、DNSSEC キーの有効期限切れに対する SNMP アラームは NetScaler で有効になっています。キーの有効期限切れ通知は、`dnskeyExpiry` と呼ばれる SNMP トラップを介して送信されます。`dnskeyExpiry` SNMP トラップとともに、3 つの MIB 変数 `dnskeyName` および `dnskeyUnitsOfExpiry` が送信されます。詳細については、「NetScaler SNMP OID リファレンス」を参照してください。

x.509 証明書の有効期限が切れる前に KSK/ZSK プライベートキーをロールオーバーする

NetScaler では、ゾーン署名キーとキー署名キーのロールオーバーを実行するために、事前公開および二重署名方式を使用できます。詳細については、NetScaler Docs の「**Domain Name System > Configuring DNSSEC**」トピックを参照してください。

DNSSEC ADNS サーバーを保護する

NetScaler が DNSSEC プロキシモードで構成されている場合、バックエンド ADNS サーバーからの応答をキャッシュし、キャッシュされた応答を DNS クライアントに転送します。

NetScaler が特定のゾーンの権威サーバーである場合、そのゾーン内のすべてのリソースレコードは NetScaler 上で構成されます。権威ゾーンに署名するには、そのゾーンのキー（ゾーン署名キーとキー署名キー）を作成し、そのキーを NetScaler に追加してから、ゾーンに署名する必要があります。

NetScaler を権威サーバーとして構成するには、次の手順を実行します。

1. ADNS サービスを追加します。

例えば:

```
add service s1 <ip address> adns 53`
```

2. DNS キーを作成します。

例えば、`com` ドメインの権威サーバーとして機能させるには、

```
create dns key -zoneName com -keytype ksk -algorithm rsASHA512 -keysize  
3000 -fileNamePrefix com.ksk.rsasha1.3000
```

```
create dns key -zoneName com -keytype zsk -algorithm rsASHA512 -keysize  
3000 -fileNamePrefix com.zsk.rsasha1.3000
```

注:

DNSキーは一度作成すると、/nsconfig/dns に保存されます。

3. DNSキーを追加します。

例えば、

```
add dns key com.zsk.3000 /nsconfig/dns/com.zsk.rsasha1.3000.key /  
nsconfig/dns/com.zsk.rsasha1.3000.private
```

```
add dns key com.ksk.3000 /nsconfig/dns/com.ksk.rsasha1.3000.key /  
nsconfig/dns/com.ksk.rsasha1.3000.private
```

4. `com` ゾーンのNSおよびSOAレコードを追加し、ゾーンに署名します。

```
add dns soaRec com -originServer n1.com -contact citrix
```

```
add dns nsrec com n1.com
```

```
add dns zone com -proxyMode no
```

```
add dns addRec n1.com 1.1.1.1
```

```
sign dns zone com
```

注: さらに、DNSグローバルパラメータでDNSEC拡張パラメータも有効にする必要があります。

NetScalerを権威ドメインネームサーバーとして構成する方法の詳細については、[製品ドキュメント](#)の「ドメインネームシステム > NetScalerをADNSサーバーとして構成する」トピックを参照してください。

レガシー構成

NetScaler®で暗号リダイレクトを無効にするように構成する

暗号リダイレクトを使用すると、SSLハンドシェイクが失敗したときに、正確で意味のあるエラーメッセージを配信するようにSSL仮想サーバーを構成できます。SSLハンドシェイクが失敗すると、NetScalerはユーザーを以前に構成されたURLにリダイレクトします。URLが構成されていない場合、NetScalerは内部で生成されたエラーページを表示します。

コマンドプロンプトで、次のコマンドを実行して暗号リダイレクトを構成し、構成を確認します。

```
set ssl vservice <vServerName> -cipherRedirect DISABLED

show ssl vservice <vServerName>
```

NetScalerでデフォルトのSSLプロファイルを仮想サーバーにバインドすると、その仮想サーバーのSSLパラメーターは、バインドされたプロファイルから継承されます。その結果、SSLプロファイルがこれらの設定を制御するため、仮想サーバーでSSLパラメーターを直接変更することはできません。

```
set ssl profile <ssl Profile name> -cipherredirect DISABLED
```

詳細については、[暗号リダイレクト](#)を参照してください。

NetScalerでSSL再ネゴシエーション設定を構成する

SSL再ネゴシエーションを無効にすることをお勧めします。

```
set ssl parameter -denySSLReneg ALL
```

プロファイルが有効になっている場合、このパラメーターをフロントエンドプロファイルとバックエンドプロファイルの両方で設定することもお勧めします。

```
set ssl profile <frontend ssl profile name> -denySSLReneg ALL
```

```
set ssl profile <backend ssl profile name> -denySSLReneg ALL
```

詳細については、-denySSLRenegパラメーターの構成方法と使用方法を参照してください。

ロギングと監視

ネットワークタイムプロトコルを構成する

NetScalerでネットワークタイムプロトコル (NTP) を有効にし、信頼できるネットワークタイムサーバーを使用するように構成することをお勧めします。NTPを有効にすると、ログエントリとシステムイベントに記録される時刻が正確になり、他のネットワークリソースと同期されます。

NTPを構成する際、ntp.confファイルを変更して、NTPサーバーが機密パケット内の情報を開示しないように制限する必要があります。

NetScalerでNTPを構成するには、次のコマンドを実行します。

```
add ntp server <IP_address> 10
```

```
enable ntp sync
```

追加する信頼できるNTPサーバーごとにntp.confファイルを変更します。すべてのサーバーエントリに対応するrestrictエントリが必要です。NetScalerシェルプロンプトから `find . -name ntp.conf` コマンドを実行すると、ntp.confファイルを見つけることができます。

SNMPを構成する

NetScalerはSNMPプロトコルのバージョン3をサポートしています。SNMPv3には、認証、アクセス制御、データ整合性チェックなどの管理およびセキュリティ機能が組み込まれています。詳細については、「[SNMPv3クエリ用にNetScalerを構成する](#)」を参照してください。

注:

SNMPv1およびSNMPv2の代わりにSNMPv3を使用することをお勧めします。

少なくとも1つのSNMPマネージャーを構成しない場合、NetScalerは、ネットワーク内のすべてのIPアドレスからのSNMPクエリを受け入れ、応答します。SNMPマネージャーを追加し、この動作を制限するには、次のコマンドを実行します。

```
add snmp manager <IP_address>
```

SNMPが不要な展開では、次のコマンドで機能を無効にする必要があります。

```
set ns ip <IP_Address> -snmp disabled
```

外部NetScalerログホストへのロギングを構成する

NetScaler Audit Serverは、カーネルおよびユーザーレベルのデーモン内のさまざまなモジュールによって収集されたすべての状態およびステータス情報をログに記録します。Audit Serverを使用すると、管理者はイベント履歴を時系列で確認できます。Audit Serverは、NetScalerからログを収集するSYSLOGサーバーに似ています。Audit Serverは、管理者資格情報を使用して、1つ以上のNetScalerアプライアンスからログを取得します。

- ローカル監査サーバーの構成

NetScalerでローカル監査サーバーへのログ記録を設定するには、次のコマンドを実行します。

```
> set audit nslogparams -serverip <hostname> -serverport <port>
```

- リモート監査サーバーの構成

リモートコンピューターの監査サーバーにログ記録を設定するには、そのコンピューターに監査サーバーをインストールします。以下は、監査サーバーのオプションの例です。

```
./audserver -help

usage : audserver -[cmds] [cmd arguments]

cmds cmd arguments: -f <filename> -d debug

-help - detail help

-start - cmd arguments,[starts audit server]

-stop - stop audit server
```

```
-verify - cmd arguments [verifies config file]

-addns - cmd arguments [add a netscaler to conf file]

-version - prints the version info
```

これらのオプションは、NetScaler ns.logファイルによって生成された監査メッセージのみをログに記録する機能を提供します。すべてのsyslogメッセージをログに記録するには、次の手順を実行します。

1. ローカルファシリティのnsconfig/syslog.confファイルからログファイル指定を削除します。
2. ログファイル指定を、次のエントリと同様に、リモートsyslogホストのログホスト名またはIPアドレスに置き換えます。

```
local0.* @10.100.3.53
```

```
local1.* @10.100.3.53
```

3. 上記のログ機能からのログエントリを受け入れるようにsyslogサーバーを設定します。詳細については、syslogサーバーのドキュメントを参照してください。
4. 標準のsyslogソフトウェアを使用するほとんどのUNIXベースのサーバーでは、メッセージとnsvpn.logファイル用のローカルファシリティ構成エントリをsyslog.conf構成ファイルに追加する必要があります。ファシリティ値は、NetScalerで構成された値に対応している必要があります。
5. UNIXベースのコンピューターのリモートsyslogサーバーは、デフォルトではリモートログをリッスンしません。したがって、リモートsyslogサーバーを起動するには、次のコマンドを実行します。

```
syslogd -m 0 -r
```

注記：

監査サーバーに展開されているsyslogバリエーションの同等のオプションを参照してください。

セキュアチャネルを介したメッセージのエクスポートを構成する

SSL (Secure Sockets Layer) を介してsyslogを安全にエクスポートすることで、機密性の高いログデータが暗号化され、認証された方法でネットワークを通過するようにできます。SSLを活用することで、syslogの送信を盗聴、改ざん、不正アクセスから保護できます。詳細については、「[SSLを介してsyslogを安全にエクスポートする](#)」を参照してください。

LOM 設定

LOM インターフェイスを保護するために、以下の対策を講じることをお勧めします。

- LOM ポートをインターネットに公開しないでください。
- LOM を SPI ファイアウォールの背後に展開します。
- LOM を、信頼できないネットワークトラフィックから論理的 (VLAN の分離) または物理的 (LAN の分離) に分離されたネットワークセグメントに展開します。
- LOM および NetScaler 管理ポートには、異なるユーザー名、パスワード、SSL 証明書、および SSL キーの値を設定します。
- LOM 管理インターフェイスへのアクセスに使用されるデバイスが、ネットワーク管理専用であり、他の管理デバイスポートと同じ物理 LAN または VLAN 内にある管理ネットワークセグメントに配置されていることを確認してください。
- LOM IP アドレスを簡単に識別および分離するために、LOM 管理インターフェイスおよび管理サーバー用に特別な IP アドレス (プライベートサブネット) を予約します。管理対象 NetScaler の LAN インターフェイスで予約済み IP サブネットを使用しないでください。DHCP によって割り当てられた動的 IP アドレスは、LAN セグメント外の MAC アドレスに基づいてファイアウォールアクセス制御リストを実装することが困難になるため、推奨されません。
- パスワードは、最低 8 文字で、英数字と特殊文字を組み合わせで設定します。パスワードは頻繁に変更してください。

システムおよびユーザーアカウント

スーパーユーザーアカウントのパスワードを変更する: 組み込みの管理者スーパーユーザー (`nsroot`) は削除できません。そのため、そのアカウントのデフォルトパスワードを安全なパスワードに変更してください。管理者ユーザーのデフォルトパスワードを変更するには、次の手順を実行します。

1. スーパーユーザーとしてログオンし、構成ユーティリティを開きます。
2. ナビゲーションペインで、ノードを展開します。

3. ノードを選択します。
4. ページで、**nsroot** ユーザーを選択します。
5. を選択します。
6. フィールドと フィールドに、必要なパスワードを入力します。
7. をクリックします。

代替スーパーユーザーアカウントを作成する: スーパーユーザーアカウントを作成するには、次のコマンドを実行します。

```
add system user <newuser> <password>

bind system user <newuser> superuser 0
```

デフォルトの **nsroot** スーパーユーザーアカウントの代わりに、このスーパーユーザーアカウントを使用してください。

NetScaler SDX の展開では、管理者は初期設定後に NetScaler SDX およびその GUI 管理コンソールのデフォルトの資格情報を変更する必要があります。デフォルトユーザーのパスワードを変更するには、次の手順を実行します。

1. スーパーユーザーとしてログオンし、構成ユーティリティを開きます。
2. ナビゲーションペインで、ノードを展開します。
3. ノードを選択します。
4. 「システムユーザー」ページで、デフォルトユーザーを選択します。
5. 「変更」を選択します。
6. 「パスワード」フィールドと「パスワードの確認」フィールドに、必要なパスワードを入力します。
7. 「OK」をクリックします。

システムユーザーの強力なパスワード:

NetScalerで作成されたシステムユーザーアカウントには、強力なパスワードを使用することをお勧めします。パスワードの複雑さの要件の例は次のとおりです。

- パスワードは8文字以上の長さである必要があります。

- パスワードには、辞書に載っている単語や、辞書に載っている単語の組み合わせを含めてはなりません。
- パスワードには、少なくとも1つの大文字、1つの小文字、1つの数字、および1つの特殊文字を含める必要があります。

強力なパスワードは、パスワードの最小長とパスワードの複雑さを強制する2つのパラメータを設定することで適用できます。

```
set system parameter -minpasswordlen <positive_integer> -  
-strongpassword ( ENABLED | DISABLED )
```

複数の管理者が要求される展開では、ユーザーを認証するためにRADIUS、TACACS+、LDAP(S)などの外部認証方法の使用を検討してください。詳細については、「[外部ユーザー認証](#)」を参照してください。

管理アクセス用のシステムユーザーアカウントのロック: NetScalerでは、システムユーザーを24時間ロックし、ユーザーへのアクセスを拒否できます。NetScalerは、システムユーザーと外部ユーザーの両方の構成をサポートしています。コマンドプロンプトで次のように入力します。

```
set aaa parameter -persistentLoginAttempts DISABLED
```

次に、ユーザーアカウントをロックするには、コマンドプロンプトで次のように入力します。

```
lock aaa user test
```

GUIを使用してこの機能を構成する方法については、「[ユーザーアカウントとパスワード管理](#)」を参照してください。

管理アクセス用にロックされたシステムユーザーアカウントのロックを解除する: システムユーザーと外部ユーザーは、認証、承認、監査ユーザーコマンドのロックを使用して24時間ロックできます。NetScalerでは、ロックされたシステムユーザーのロックを解除できます。コマンドプロンプトで、次のように入力します。

```
unlock aaa user test
```

 GUIを使用してこの機能を構成する方法については、「[ユーザーアカウントとパスワード管理](#)」を参照してください。

システムユーザーアカウントの管理アクセスを無効にする: NetScalerで外部認証が構成されており、管理者がシステムユーザーの管理アクセスへのログオンを拒否したい場合は、システムパラメータでlocalAuthオプションを無効にする必要があります。

注:

外部サーバーが構成されている必要があります。

コマンドプロンプトで、次のように入力します。

```
set system parameter localAuth <ENABLED|DISABLED>
```

例:

`set system parameter localAuth DISABLED` GUI を使用してこの機能を構成する方法については、「[ユーザーアカウントとパスワード管理](#)」を参照してください。

管理者ユーザーのパスワード変更を強制する: `nsroot` で保護された認証の場合、システムパラメータで `forcePasswordChange` オプションが有効になっていると、NetScalerはユーザーにデフォルトパスワードを新しいパスワードに変更するよう促します。デフォルトの資格情報で初めてログインする際に、CLIまたはGUIから `nsroot` パスワードを変更できます。コマンドプロンプトで、次のように入力します。

```
set system parameter -forcePasswordChange ( ENABLED | DISABLED )
```

この機能の構成例については、「[ユーザーアカウントとパスワード管理](#)」を参照してください。

SSHキーとパスワードなしでNetScalerにアクセスする: 多数のNetScalerアプライアンスを管理する必要がある展開では、SSHキーとパスワードなしを使用することを検討してください。この機能の構成方法については、「[SSHキーとパスワードなしでNetScalerアプライアンスにアクセスする](#)」を参照してください。

データ保護のためのシステムメインキーを作成する: Citrix ADC 12.1からCitrix 13.0–71.44では、LDAP認証に必要なサービスアカウントパスワードやローカルに保存された認証、承認、監査ユーザーアカウントなど、特定のセキュリティパラメータを保護するためにシステムメインキーを作成する必要があります。

注記:

Citrix 13.0 ビルド 76.31 以降では、アップグレードプロセスによってランダムなシステムメインキーがデフォルトで自動的に作成されます。組織のパスワードポリシーに従って、KEKを頻繁に更新してください。

システムメインキーを作成するには:

1. CLI を使用して、システム管理者としてログインします。
2. 次のコマンドを入力します。

```
create kek <passphrase>
```

注記:

- `create kek` コマンドの実行後、KEK はほとんどのパスワード暗号化に使用されます（ローカルユーザーパスワードは KEK で暗号化されません）。
- KEK ファイルを削除してはいけません。シェルアクセスがあり、誤ってキーフラグメントファイルを削除した場合、設定の損失、同期の失敗、ログオンの失敗につながる可能性があります。以下に注意してください。
 - ダウングレードする際は、常にインストールされているビルドと一致する古い設定ファイルを使用してください。そうしないと、ログオン、ソース設定、同期、フェイルオーバーが失敗する可能性があります。
 - キーフラグメントファイルのいずれかが失われたり破損したりした場合、機密データの暗号化/復号化が失敗し、その結果、設定の損失、同期の失敗、ログオンの失敗につながる可能性があります。
- パスフレーズは8文字以上である必要があります。

展開済みの NetScaler でキー暗号化キーを更新する:

NetScaler は、展開済みの ADC で KEK の更新をサポートしています。KEK を更新するには、次のコマンドを使用します。

```
update kek -level <basic | extended>
```

KEK 更新コマンドは NSIP インターフェイスでのみサポートされています。このコマンドは次の2つのオプションをサポートしています。

- **Basic:** 古いキーをバックアップし、キーを作成して応答します。ファイルの更新が失敗した場合、システムはエラーを報告し、元の状態に戻ります。
- **拡張:** 古いキーをバックアップし、キーを作成します。デフォルトおよび非デフォルトのパーティションにある `ns.conf` や `ns.conf.0` などの設定ファイルを更新します。更新中は、すべての設

定変更をブロックします。更新が完了すると、NetScalerが応答します。これらのファイル更新のいずれかが失敗した場合、システムはエラーを報告し、元の状態に戻ります。

以前は、NetScalerはノードごとのデフォルトのKEKのみをサポートしていました。KEKを更新するオプションはありませんでした。

セキュリティのベストプラクティスとして、KEKは組織のパスワードポリシーに従って頻繁に変更する必要があります。

アクセス制御リストを使用する:

デフォルトでは、GUIやSSHを含むすべてのプロトコルとポートがNetScalerでアクセス可能です。アクセス制御リスト (ACL) を使用すると、明示的に指定されたユーザーのみがポートとプロトコルにアクセスできるようにすることで、NetScalerを安全に管理できます。

NetScalerへのアクセスを制御するための推奨事項:

- NetScalerへのアクセスをGUIのみに制限するために、NetScaler Gatewayの使用を検討してください。GUI以外のアクセス方法を必要とする管理者向けには、NetScaler Gatewayを、ポート80、443、3010に対してデフォルトの「DENY」ACLで構成する必要がありますが、これらのポートにアクセスするための信頼できるIPアドレスに対しては明示的な「ALLOW」を設定する必要があります。

このポリシーは、次のNSCLIコマンドを使用して、信頼できるIPアドレスの範囲で使用するよう拡張できます。

```
add acl local_access allow -srcip 192.168.0.1-192.168.0.3 -destip  
192.168.0.1-192.168.0.3
```

```
apply acls
```

- SNMPを使用する場合は、ACLでSNMPトラフィックを明示的に許可します。以下にコマンドの例を示します。

```
add acl snmp1-ssh ALLOW -srcip 10.0.0.1-10.0.0.20 -destip  
192.168.0.2-192.168.0.3 -destport 161 -protocol udp
```

```
add acl snmp2-ssh ALLOW -srcip 172.16.0.1-172.16.0.20 -destip
```

```
192.168.0.2-192.168.0.3 -destport 161 -protocol udp
```

```
apply acls
```

前の例では、コマンドは、クエリが適切に定義されたコミュニティに対するものであっても、定義された2つのサブネットへのすべてのSNMPクエリにアクセスを提供します。

NSIPおよびSNIPアドレスで管理機能を有効にできます。有効になっている場合は、管理機能へのアクセスを保護するために、ACLを使用してNSIP、SNIPアドレスへのアクセスを提供します。管理者は、pingコマンドでアクセスできないようにNetScalerを設定することもできます。

- Open Shortest Path First (OSPF) とIPSECは、TCPまたはUDPベースのプロトコルではありません。したがって、NetScalerでこれらのプロトコルをサポートする必要がある場合は、ACLを使用してこれらのプロトコルを使用するトラフィックを明示的に許可します。プロトコル番号でOSPFとIPSECを指定するACLを定義するには、次のコマンドを実行します。

```
add acl allow_ospf allow -protocolnumber 89
```

```
add acl allow_ipsec allow -protocolnumber 50
```

- XML-API Webサービスを使用する場合は、APIインターフェイスを保護するために次のタスクを完了します。
- ACLを使用して、インターフェイスにアクセスするためのホストに権限を付与します。たとえば、10.0.0.1-20および172.16.0.1-20のIPアドレス範囲のホストがXML-APIインターフェイスにアクセスできるようにするには、次のコマンドを実行します。

```
add acl xml-api1 ALLOW -srcip 10.0.0.1-10.0.0.20 -destip  
192.168.0.2-192.168.0.3 -destport 80 -protocol tcp
```

```
add acl xml-api2 ALLOW -srcip 172.16.0.1-172.16.0.20 -destip  
192.168.0.2-192.168.0.3 -destport 80 -protocol tcp
```

```
apply acls
```

- 内部ポートにACLを適用するには、次のコマンドを使用します。

```
set l3param -implicitACLAllow DISABLED
```

注:

implicitACLAllow コマンドのデフォルト値は **有効** です。

- 内部ポートからACLを削除するには、次のコマンドを使用します。

```
set l3param -implicitACLAllow ENABLED
```

- クライアント側証明書を使用することで、追加のセキュリティを確保できます。クライアント側証明書とクライアント認証の詳細については、「[クライアント認証](#)」またはKnowledge Centerの記事CTX214874:「NetScaler Applianceでファームウェア10.1以降のクライアント証明書を作成して使用する方法」を参照してください。

管理者ユーザーにロールベースのアクセス制御を使用する:

NetScalerには、オペレーター、読み取り専用、ネットワーク、スーパーユーザーの4つのコマンドポリシーまたはロールが含まれています。コマンドポリシーを定義し、異なるロールに対して異なる管理アカウントを作成し、そのロールに必要なコマンドポリシーをアカウントに割り当てることもできます。以下は、読み取り専用ユーザーへの読み取り専用アクセスを制限するためのコマンド例です。

```
add system user readonlyuser
```

```
bind system user readonlyuser read-only 0
```

ユーザー、ユーザーグループ、またはコマンドポリシーの設定の詳細については、「[ユーザー、ユーザーグループ、およびコマンドポリシー](#)」を参照してください。

システムセッションタイムアウトの設定:

セッションタイムアウト間隔は、セッション（GUI、CLI、またはAPI）が使用されていないときにアクティブな状態を維持する期間を制限するために提供されます。NetScalerでは、システムセッションタイムアウトを次のレベルで設定できます。

- ユーザーレベルのタイムアウト。特定のユーザーに適用されます。

GUI: **System > User Administration > Users** に移動し、ユーザーを選択して、ユーザーのタイムアウト設定を編集します。CLI: コマンドプロンプトで、次のコマンドを入力します。

```
set system user <name> -timeout <secs>
```

- ユーザーグループレベルのタイムアウト。グループ内のすべてのユーザーに適用されます。

GUI: **System > User Administration > Groups** に移動し、グループを選択して、グループのタイムアウト設定を編集します。CLI: コマンドプロンプトで、次のコマンドを入力します。

```
set system group <groupName> -timeout <secs>
```

- グローバルシステムタイムアウト。タイムアウトが設定されていないすべてのユーザーおよびグループのユーザーに適用されます。

GUI: **System > Settings** に移動し、**Set global system parameters** をクリックして、任意のクライアントアイドルタイムアウト（秒）パラメーターを設定します。CLI: コマンドプロンプトで、次のコマンドを入力します。

```
set system parameter -timeout <secs>
```

ユーザーに指定されたタイムアウト値が最も優先度が高くなります。ユーザーにタイムアウトが設定されていない場合、メンバーグループに設定されたタイムアウトが考慮されます。グループにタイムアウトが指定されていない場合（またはユーザーがグループに属していない場合）、グローバルに設定されたタイムアウト値が考慮されます。いずれのレベルでもタイムアウトが設定されていない場合、デフォルト値の900秒がシステムセッションタイムアウトとして設定されます。

セッションタイムアウト値が管理者によって設定されたタイムアウト値を超えて設定できないように、タイムアウト値を制限することもできます。タイムアウト値は5分から1日の間に制限できます。タイムアウト値を制限するには：

- GUI：システム > 設定に移動し、**グローバルシステムパラメータの設定**をクリックして、「制限付きタイムアウト」フィールドを選択します。
- CLI：コマンドプロンプトで、次のコマンドを入力します。

```
set system parameter -restrictedtimeout <ENABLED/DISABLED>
```

ユーザーがrestrictedTimeoutパラメータを有効にした後、タイムアウト値がすでに1日より大きい値または5分より小さい値に設定されている場合、ユーザーはタイムアウト値を変更するよう通知されます。ユーザーがタイムアウト値を変更しない場合、デフォルトで、次の再起動時にタイムアウト値は900秒（15分）に再設定されます。

アクセスしている各インターフェースに対してタイムアウト期間を指定することもできます。ただし、特定のインターフェースに指定されたタイムアウト値は、そのインターフェースにアクセスしているユーザーに設定されたタイムアウト値に制限されます。例えば、ユーザー `publicadmin` のタイムアウト値が20分であるとします。この場合、インターフェースにアクセスする際、ユーザーは20分以内のタイムアウト値を指定する必要があります。

各インターフェースでタイムアウト期間を設定するには：

- CLI：次のコマンドを使用して、コマンドプロンプトでタイムアウト値を指定します。

```
set cli mode -timeout <secs>
```

- API：ログインペイロードでタイムアウト値を指定します。

NetScaler® Gateway セキュリティに関する推奨事項

'Default Deny' ポリシーを使用する

管理者は、グローバルレベルで「すべて拒否」ポリシーを使用してNetScaler Gatewayを構成することをお勧めします。これは、グループベースでリソースへのアクセスを選択的に有効にするための承認ポリシーの使用に加えて適用されます。

デフォルトでは、defaultAuthorizationAction パラメータはDENYに設定されています。この設定を確認し、各ユーザーに明示的なアクセスを許可してください。設定を確認するには、CLIで `show vpn parameter` コマンドを使用できます。グローバルレベルですべてのリソースを拒否するようにパラメータを設定するには、CLIから次のコマンドを実行します。

```
set vpn parameter -defaultAuthorizationAction DENY
```

サーバー間のTLS1.2通信を使用する

NetScaler Gatewayと、LDAPやWeb Interfaceサーバーなどの他のサービス間のリンクには、TLS1.2またはTLS 1.3を使用することをお勧めします。このプロトコルの古いバージョン（TLS 1.1、TLS 1.0、SSLv3以前）の使用は推奨されません。

イントラネットアプリケーションを使用する

NetScaler Gatewayプラグインによってインターセプトされ、ゲートウェイに送信されるネットワークを定義するには、**Intranet Applications** 機能を使用します。以下は、インターセプトを定義するためのコマンドのサンプルセットです。

```
add vpn intranetApplication intra1 ANY 10.217.0.0 -netmask 255.255.0.0
    -destPort 1-65535 -interception TRANSPARENT

bind vpn vserver v1 -intranetapp intra1
```

認証、承認、および監査のセキュリティに関する推奨事項

NetScalerまたはNetScaler GatewayアプライアンスがSAML SPまたはSAML IdP、あるいはその両方として構成されている場合、推奨される構成の詳細については、記事 [<https://support.citrix.com/article/CTX316577>](https://support.citrix.com/article/CTX316577) を参照してください。

SAML認証の詳細については、[SAML認証](#) を参照してください。

nFactor認証のためのNetScaler Gatewayログイン情報の暗号化を有効にする

nFactor認証を備えたNetScaler Gatewayアプライアンスは、認証プロセス中にクライアント（ブラウザまたはSSOアプリ）によって送信されるログイン要求フィールドを暗号化できます。暗号化されたログイン要求フィールドは、ユーザーの機密データが開示されるのを防ぐための追加のセキュリティ層を提供します。

CLIを使用してログイン暗号化を有効にするには、次のコマンドを実行します。

```
set aaa parameter [-loginEncryption (ENABLED | DISABLED)]
```

GUIを使用してログイン暗号化を有効にするには

1. **セキュリティ > AAA – アプリケーショントラフィック** に移動します。
2. 「認証設定」セクションの下にある「**認証AAA設定の変更**」をクリックします。
3. 「**AAAパラメータの構成**」ページで、「**ログイン暗号化**」の「**有効**」をクリックします。

ログイン暗号化の詳細については、「[nFactor認証用のNetScaler Gatewayログイン情報の暗号化](#)」を参照してください。

認証、承認、監査VPN仮想サーバーおよびNetScaler Gatewayを保護する

認証、承認、監査VPN仮想サーバーおよびNetScaler Gatewayを保護するために、以下の軽減策を推奨します。

1. NetScaler Gatewayで多要素認証が有効になっており、多要素認証の検証要素がLDAP要素より前であることを確認してください。詳細については、「[多要素認証の構成](#)」を参照してください。
2. 攻撃の一部はNetScaler GatewayのIPアドレスに対してトリガーされ、FQDNに対してではないように見えるため、目的のFQDNのみにリクエストを許可するレスポnderポリシーを作成します。

```
add responder policy IP_Block "HTTP.REQ.HOSTNAME.EQ(\"<enter gateway  
FQDN here>\").NOT" DROP
```

```
bind vpn vserver <vserver name> -policy IP_Block -priority 100 -type  
REQUEST
```

3. IPレピュテーションを有効にします。IPレピュテーションを有効にすると、既知の悪意のあるIPアドレスからのリクエストを破棄し、攻撃の影響を軽減するのに役立ちます。IPレピュテーションを有効にするには、次のコマンドを実行します。

```
enable feature reputation
```

```
add responder policy policy_brute_block_ip
```

```
"CLIENT.IP.SRC.IPREP_IS_MALICIOUS" DROP
```

```
bind vpn vserver <Gateway_vserver_name> -policy policy_brute_block_ip
-priority 50 -gotoPriorityExpression END -type AAA_REQUEST
```

IPレピュテーションの設定方法の詳細については、「[IPレピュテーション](#)」を参照してください。

4. NetScalerでreCAPTCHAを有効にします。reCAPTCHAの設定方法については、「[nFactor認証のreCAPTCHA構成](#)」を参照してください。

スプリットDNS値をREMOTEとして構成する

DNS解決を `REMOTE` として構成することをお勧めします。

以下のいずれかの方法でDNS解決を構成できます。

1. `set vpn parameter splitDns REMOTE`
2. `set vpn sessionAction <action-name> splitDns REMOTE`

`sessionAction` で `splitDns` を設定することはオプションです。もし `sessionAction` で `splitDns` が提供されない場合、ゲートウェイは `vpn parameter` 設定から `splitDns` の値を取得します。

もし `sessionAction` で `splitDns` が設定されている場合、`sessionAction` 設定は `vpn parameter` 設定よりも優先されます。

デフォルトでは、CLIは `vpn parameter splitDns` を `BOTH` に設定します。この値を `REMOTE` として設定することをお勧めします。

ネットスケーラー® ウェブアプリファイアウォール セキュリティに関する推奨事項

- RFC準拠チェックの場合、WAFプロファイルのデフォルトの `rfcprofile` として「`APPPFW_RFC_BLOCK`」を維持することをお勧めします。
- WAFは `Samesite` クッキー属性値の挿入をサポートしており、クッキーは「`Strict`」または「`Lax`」の値を選択することで、同一サイトまたはクロスサイトのコンテキストに制限できます。

NetScalerを2アームモードで展開する

2アームモードのインストールでは、NetScalerはNetScalerが保護するユーザーとWebサーバーの間に物理的に配置されます。接続はNetScalerを通過する必要があります。この配置により、NetScalerを迂回する経路が見つかる可能性が最小限に抑えられます。

「デフォルト拒否」ポリシーを使用する

管理者は、NetScaler Web App Firewallポリシーに一致しないすべてのリクエストをブロックするために、グローバルレベルで「すべて拒否」ポリシーを設定することをお勧めします。以下は、グローバルレベルで「すべて拒否」ポリシーを設定するためのコマンドのサンプルセットです。

```
add appfw profile default_deny_profile -defaults advanced

add appfw policy default_deny_policy true default_deny_profile

bind appfw global default_deny_policy <PRIORITY>
```

注:

PRIORITY設定は、デフォルトポリシーが最後に評価されるようにする必要があります（リクエストが他の設定済みポリシーと一致しない場合にのみ）。

NetScalerソフトウェアには、appfw_blockなどのデフォルトプロファイルが含まれており、これらが設定されると、NetScaler Web App Firewallポリシーに一致しないリクエストをブロックします。デフォルトプロファイルを設定するには、次のコマンドを実行します。

```
set appfw settings -defaultProfile appfw_block
```

ネットスケーラー ウェブアプリファイアウォール – 複数のセキュリティ層を構築する

以下のガイドラインは、環境とサポートされているアプリケーションに応じて、複数のセキュリティ層を構築するのに役立ちます。

各層で`sessionCookieName`パラメーターに異なる値を設定します。

```
set appfw settings -sessionCookieName "citrix_ns_id_1"
```

セキュリティの第一層

セキュリティの第一層を構築するには、以下を実行します。

- バッファオーバーフロー、SQLインジェクション、クロスサイトスクリプティングを有効にします。
- アプリケーションがアクセスする必要があるURLを特定し、強制的なブラウジングから保護する必要がある場合、開始URLが必要です。
- アプリケーションがフォームフィールドに入力を期待している場合は、フィールド形式チェックを有効にします。

多くの企業が同一オリジンポリシーに違反するJavaScript強化ウェブコンテンツの大量のインストールベースを持っているため、クロスサイトスクリプティングチェックは誤検知を生成する可能性があります。そのようなサイトでHTMLクロスサイトスクリプティングチェックを有効にする場合、チェックが正当なアクティビティをブロックしないように、適切な例外を生成する必要があります。

最初の層を展開し、誤検知を探し、例外をデプロイしてから、次の層に進みます。段階的な実装は、AppFwの展開管理に役立ちます。

セキュリティの第2層

セキュリティの第2層を構築するには、以下を実行します。

バッファオーバーフロー、SQLインジェクション、クロスサイトスクリプティングに加えて、プロファイルでシグネチャを有効にします。1300以上のシグネチャがあります。すべてのシグネチャルールを有効にするのではなく、アプリケーションの保護に適用できるシグネチャのみを有効にしてください。

第2層を展開し、誤検知を探し、例外をデプロイしてから、次の層に進みます。段階的な実装は、NetScaler Web App Firewallの展開管理に役立ちます。

セキュリティの第3層

セキュリティの第3層を構築するには、以下を実行します。

- アプリケーションのニーズに基づいて、CSRFタグ付け、Cookieの一貫性、フォームフィールドの一貫性などの高度なプロファイルセキュリティチェックを、それらを必要とするアプリケーションの部分で有効にします。
- 高度なセキュリティチェックはより多くの処理を必要とし、パフォーマンスに影響を与える可能性があります。アプリケーションが高度なセキュリティを必要としない限り、基本的なプロファイルから開始し、アプリケーションの要件に応じてセキュリティを強化することをお勧めします。

基本的なNetScaler Web App Firewallプロファイルで無効になっているセキュリティチェックはすべて、HTTP応答内のオブジェクトに対して動作します。したがって、これらのセキュリティチェックはより多くのリソースを消費します。NetScaler Web App Firewallが応答側の保護を実行する場合、個々のクライアントに送信された情報を記憶する必要があります。たとえば、フォームがNetScaler Web App Firewallによって保護されている場合、応答で送信されたフォームフィールド情報はメモリに保持されます。クライアントが次の後続のリクエストでフォームを送信すると、情報がWebサーバーに送信される前に不整合がないかチェックされます。この概念はセッション化と呼ばれます。開始URL内のURLエンクロージャ、Cookieの一貫性、フォームフィールドの一貫性、CSRFフォームタグ付けなどのセキュリティチェックはすべてセッション化を意味します。これらのセキュリティチェックによって使用されるCPUおよびメモリリソースの量は、NetScaler Web App Firewallを介して送信されるリクエストの数に比例して増加します。例：

- フォームフィールドの一貫性チェックを有効にする：このチェックは、Webフォームがクライアントによって不適切に改変されていないかを確認するために必要です。フォームで重要な情報を提供およびホストするアプリケーションには、このチェックが必要です。
- CSRFフォームタグ付けチェック：このチェックはフォーム用です。クロスサイトリクエストフォージェリ（CSRF）フォームタグ付けチェックは、保護されたWebサイトからユーザーに送信される各Webフォームに一意で予測不可能なFormIDをタグ付けし、その後、ユーザーから返されたWebフォームを調べて、提供されたFormIDが正しいことを確認します。このチェックは、クロスサイトリクエストフォージェリ攻撃から保護します。アプリケーションにWebベースのフォームがある場合、このチェックを有効にする必要があります。このチェックは、Webフォームを詳細に分析する他の特定のセキュリティチェックと比較して、CPU処理能力を比較的ほとんど必要としません。したがって、保護されたWebサイトまたはNetScaler Web

App Firewall自体のパフォーマンスを著しく低下させることなく、大量の攻撃を処理できます。

ネットスケーラー ウェブアプリファイアウォール ワークフロー ステップ

NetScaler Web App Firewallのワークフローで次の手順を実行します。

1. セキュリティプロファイルを構成します。
2. 既知のすべての脅威（ネガティブモデル）に署名を適用します。
3. このセキュリティプロファイルをアクティブ化する必要がある、正しいトラフィックフローを検出できるトラフィックポリシーを構成します。

本番トラフィックがシステムを通過する準備ができました。フローの最初のレベルが完了しました。さらに、学習インフラストラクチャを構成します。多くの場合、顧客は本番トラフィックで学習を行いたいと考えているため、署名を適用することでリスクを回避できます。次の手順を実行します。

1. 学習インフラストラクチャを構成します。
2. 保護のために学習したルールを展開します。
3. 本番稼働前に、適用された署名とともに学習データを検証します。