セキュアなWebアプライアンスのベストプラクティスの使用

内容

はじめに

このドキュメントでは、CiscoセキュアWebアプライアンス(SWA)の設定方法に関するベストプラクティスについて説明します。

背景説明

このガイドは、ベストプラクティス構成のリファレンスとして使用することを目的としており、サポートされているネットワーク環境、ポリシー構成、モニタリング、トラブルシューティングなど、SWAの導入のさまざまな側面に対応しています。ここで説明するベストプラクティスは、すべての管理者、アーキテクト、オペレータが理解する必要がありますが、単なるガイドラインであり、そのように扱う必要があります。各ネットワークには固有の要件と課題があります。

SWAはセキュリティデバイスとして、いくつかの独自の方法でネットワークと通信します。これは、Webトラフィックの送信元と宛先の両方になります。WebサーバとWebクライアントとして同時に動作します。少なくとも、サーバ側のIPアドレススプーフィングとman-in-the-middle（中間者）技術を使用して、HTTPSトランザクションを検査します。また、クライアントのIPアドレスをスプーフィングして、導入をさらに複雑にし、サポートするネットワーク設定に追加の要件を課すこともあります。このガイドでは、関連するネットワークデバイスの設定に関連する最も一般的な問題について説明します。

SWAポリシー設定は、セキュリティの有効性と適用だけでなく、アプライアンスのパフォーマンスにも影響を与えます。このガイドでは、設定の複雑さがシステムリソースに与える影響について説明します。このコンテキストで複雑さを定義し、ポリシー設計でそれを最小限に抑える方法を説明します。また、特定の機能と、セキュリティ、スケーラビリティ、および有効性を向上するためにこれらの機能をどのように設定する必要があるかについても注意が払われています。

このドキュメントの「モニタリングとアラート」セクションでは、アプライアンスを監視する最も効果的な方法について説明し、パフォーマンスと可用性、およびシステムリソースの使用状況のモニタリングについても説明します。また、基本的なトラブルシューティングに役立つ情報も提供します。

ネットワーク環境

ICMP

RFC 1191で定義されているように、このメカニズムによって、任意のパスに沿ったパケットの最大サイズが決定されます。IPv4の場合、デバイスはパケットのIPヘッダー内のDon't Fragment（DF；フラグメントなし）ビットを設定することによって、パス上のすべてのパケットの最大伝送ユニット(MTU)を決定できます。パスに沿ったリンク上で、デバイスがパケットをフラグメント化せずに転送できない場合、Internet Control Message Protocol(ICMP)の「Fragmentation Needed」（タイプ3、コード4）メッセージが送信元に送り返されます。次に、クライアントはより小さいパケットを再送信します。これは、フルパスのMTUが検出されるまで続きます。IPv6はフラグメンテーションをサポートせず、Packet Too Big（タイプ2）ICMPv6メッセージを使用して、特定のリンクを介してパケットを収めることができないことを示します。

パケットのフラグメンテーションのプロセスはTCPフローのパフォーマンスに重大な影響を与える可能性があるため、SWAはPath MTU Discoveryを使用します。SWAがネットワークを通過するパスのMTUを判別できるように、関連するネットワークデバイスで前述のICMPメッセージを有効にする必要があります。SWAでこの動作を無効にするには、pathmtudiscoveryコマンドラインインターフェイス(CLI)コマンドを使用します。これを実行すると、デフォルトのMTUが576バイト（RFC 879による）に低下し、パフォーマンスに大きな影響を与えます。管理者は追加の手順として、etherconfig CLIコマンドから手動でSWAのMTUを設定する必要があります。

Web Cache Communication Protocol(WCCP)の場合、Webトラフィックは、インターネットへのクライアントパスに沿って、別のネットワークデバイスからSWAにリダイレクトされます。この場合、ICMPなどの他のプロトコルはSWAにリダイレクトされません。SWAがネットワーク上のルータからのICMP Fragmentation Neededメッセージをトリガーする可能性がありますが、メッセージはSWAに配信されません。これがネットワークで発生する可能性がある場合は、Path MTU Discovery(PMTUD)を無効にする必要があります。前述したように、この設定では、etherconfig CLIコマンドを使用してSWAのMTUを手動で設定する追加の手順が必要です。

ファイアウォール

デフォルト設定では、SWAは接続をプロキシするときにクライアントIPアドレスをスプーフィングしません。つまり、すべての発信Webトラフィックの送信元はSWAのIPアドレスになります。これに対応するには、ネットワークアドレス変換(NAT)デバイスに十分な大きさの外部アドレスとポートのプールがあることを確認する必要があります。この目的には、特定のアドレスを割り当てることをお勧めします。

一部のファイアウォールでは、サービス拒否(DoS)保護や、単一のクライアントIPアドレスから大量の同時接続が発信されたときにトリガーされる他のセキュリティ機能が採用されています。クライアントのIPスプーフィングが有効になっていない場合は、SWAのIPアドレスをこれらの保護から除外する必要があります。

Unicast Reverse Path Forwarding

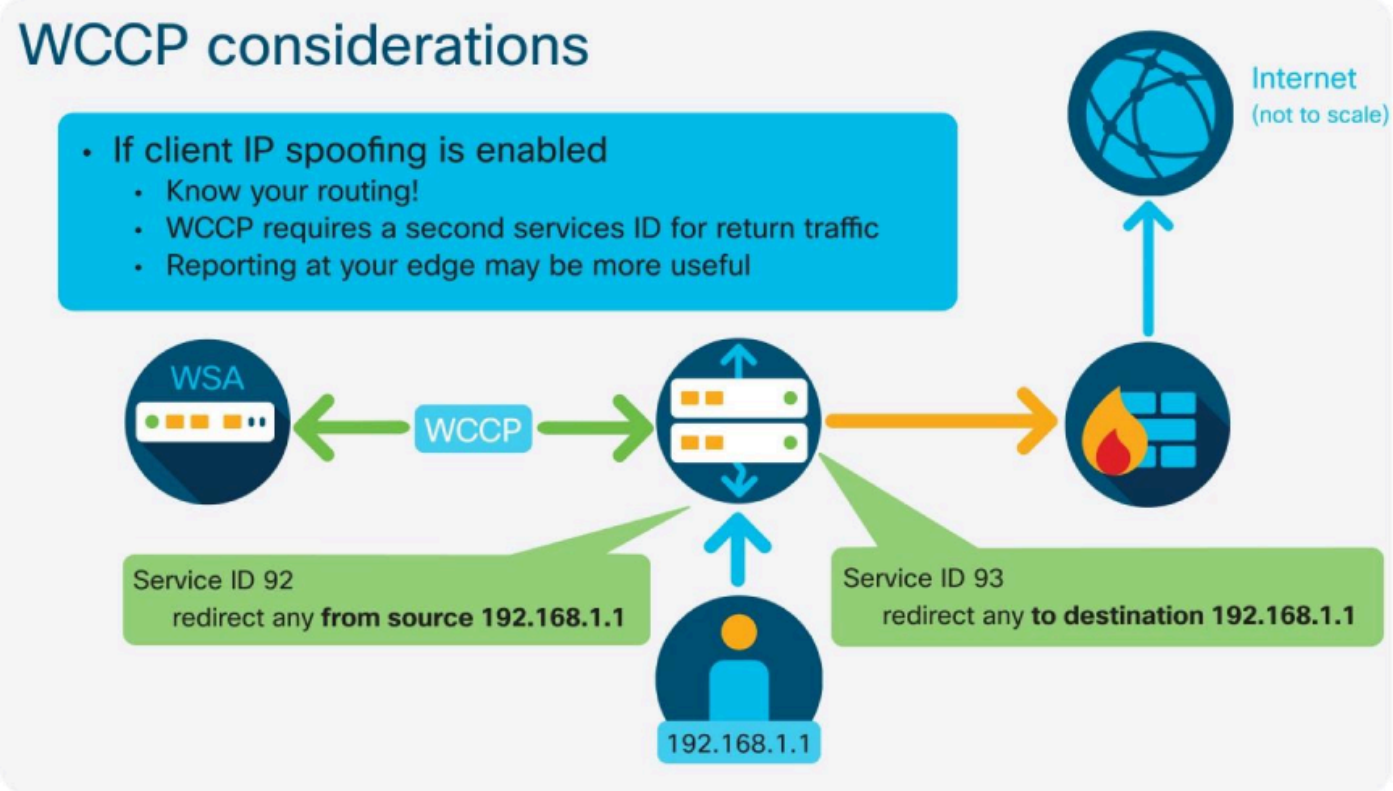
SWAは、クライアントと通信するときにサーバのIPアドレスをスプーフィングします。オプションで、アップストリームサーバと通信するときにクライアントのIPアドレスをスプーフィングするように設定できます。Unicast Reverse Path Forwarding(uRPF)などの保護をスイッチ上で有効にして、着信パケットが期待される入力ポートに確実に一致するようにします。これらの保護は、パケットの送信元インターフェイスをルーティングテーブルと照合して、パケットが期待どおりのポートに到着したかどうかを確認します。必要に応じて、SWAをこれらの保護から除外する必要があります。

WCCPによるIPスプーフィング

IPスプーフィング機能がSWAで有効になっている場合、アウトバウンド要求はアプライアンスから送信され、元のクライアント要求の送信元アドレスが使用されます。これには、要求を発信したクライアントではなくSWAアウトバウンドインターフェイスに戻りパケットがルーティングさ

れるように、関連するネットワークインフラストラクチャを追加設定する必要があります。

WCCPがネットワークデバイス (ルータ、スイッチ、またはファイアウォール) に実装されている場合、アクセスコントロールリスト(ACL)に基づいてトラフィックを照合するサービスID(SID)が定義されます。その後、サービスIDがインターフェイスに適用され、リダイレクション用のトラフィックの照合に使用されます。IPスプーフィングが有効な場合は、リターントラフィックもSWAにリダイレクトされるようにするために、2つ目のサービスID(SID)を作成する必要があります。



SWAネットワークの設定

インターフェイス

SWAには、5つの使用可能なネットワークインターフェイス (M1、P1、P2、T1、およびT2) があります。これらはそれぞれ、可能な限り特定の目的に利用する必要があります。各ポートは、それぞれの理由で使用すると便利です。M1インターフェイスは専用の管理ネットワークに接続する必要があり、管理サービスの露出を制限するためにスプリットルーティングを有効にする必要があります。P1はクライアント要求トラフィックに制限できますが、P2は明示的なプロキシ要求を受け入れることができません。これにより、各インターフェイスのトラフィック量が減少し、ネットワーク設計のセグメント化が向上します。

T1およびT2ポートは、レイヤ4トラフィックモニタ(L4TM)機能で使用できます。この機能は、ミラー化されたレイヤ2ポートを監視し、既知の悪意のあるIPアドレスとドメイン名のブロックされたリストに基づいてトラフィックをブロックする機能を追加します。これは、トラフィックの送

信元と宛先のIPアドレスを調べることによって行われ、ブロックされたリストが一致する場合は、TCPリセットパケット、またはポート到達不能メッセージを送信します。任意のプロトコルで送信されるトラフィックは、この機能でブロックできます。

L4TM機能が有効になっていない場合でも、T1およびT2ポートがミラーポートに接続されていると、トランスペアレントバイパスを強化できます。WCCPの場合、SWAは着信パケットの送信元と宛先のIPアドレスだけを認識し、その情報に基づいてパケットをプロキシするか、またはバイパスするかを決定する必要があります。SWAは、レコードの存続可能時間(TTL)に関係なく、バイパス設定リストのすべてのエントリを30分ごとに解決します。ただし、L4TM機能が有効になっている場合、SWAはスヌーピングされたDNSクエリを使用して、これらのレコードをより頻繁に更新できます。これにより、クライアントがSWAとは異なるアドレスを解決した場合に、誤検出のリスクが軽減されます。

管理ネットワークルーティング

専用管理ネットワークがインターネットにアクセスできない場合、各サービスはデータルーティングテーブルを使用するように設定できます。これはネットワークトポロジに合わせて調整できますが、一般に、すべてのシステムサービスに管理ネットワークを使用し、クライアントトラフィックにデータネットワークを使用することをお勧めします。AsyncOSバージョン11.0では、ルーティングを設定できるサービスは次のとおりです。

- 外部URLフィード
- 高度なマルウェア防御(AMP)ファイルレピュテーションと分析
- アップデートとアップグレード
- DNS
- Active Directory

管理トラフィックの出力フィルタリングを追加するには、次のサービスで使用するスタティックアドレスを設定します。

- 外部URLフィード:
  1.カスタムはホスティングされる場所によって異なります
  2. AMPファイルレピュテーションおよび分析
  3. cloud-sa.amp.cisco.com（北米）
  4. cloud-sa.eu.amp.cisco.com（ヨーロッパ）
  5. cloud-sa.apjc.amp.cisco.com（アジア太平洋）

- 更新とアップグレード:
  1. downloads-static.ironport.com
  2. updates-static.ironport.com

ターロステレメトリ

Cisco Talosグループは、新たな脅威を特定することで知られています。ターロスに送信されるすべてのデータは匿名化され、米国のデータセンターに保存されます。SensorBaseに参加すると、Web脅威の分類と特定が強化され、SWAや他のシスコのセキュリティソリューションからの保護

が強化されます。

DNS

ドメインネームサーバー(DNS)セキュリティのベストプラクティスでは、すべてのネットワークで2つのDNSリゾルバーをホストする必要があることを提案しています。1つはローカルドメイン内からの権限のあるレコード用で、もう1つはインターネットドメインの再帰的な解決用です。これに対応するため、SWAでは特定のドメインに対してDNSサーバを設定できます。ローカルクエリと再帰クエリの両方で使用できるDNSサーバーが1つだけの場合は、すべてのSWAクエリで使用するときに追加される負荷を考慮してください。ローカルドメインには内部リゾルバを使用し、外部ドメインにはルートインターネットリゾルバを使用するのが、より適切なオプションです。これは、管理者のリスクプロファイルと許容範囲によって異なります。

デフォルトでは、SWAはレコードのTTLにかかわらず、30分以上DNSレコードをキャッシュします。コンテンツ配信ネットワーク(CDN)を多用する最近のWebサイトでは、IPアドレスが頻繁に変更されるため、TTLレコードが低くなっています。これにより、クライアントは特定のサーバに対して1つのIPアドレスをキャッシュし、SWAは同じサーバに対して異なるアドレスをキャッシュする可能性があります。これに対処するために、次のCLIコマンドを使用して、SWAのデフォルトTTLを5分に下げることができます。

```
SWA_CLI> dnsconfig
...
Choose the operation you want to perform:
- NEW - Add a new server.
- EDIT - Edit a server.
- DELETE - Remove a server.
- SETUP - Configure general settings.
- SEARCH - Configure DNS domain search list.
[]> SETUP
...
Enter the minimum TTL in seconds for DNS cache.
...
```

セカンダリDNSサーバは、プライマリが使用できない場合に備えて設定する必要があります。すべてのサーバが同じ優先順位で設定されている場合、サーバのIPはランダムに選択されます。設定されたサーバの数に応じて、サーバのタイムアウトは異なります。テーブルは、最大6台のDNSサーバに対するクエリのタイムアウトです。

DNSサーバの数	クエリのタイムアウト（順序）
1	60
2	5 45
3	5、10、45
4	1、3、11、45
5	1、3、11、45、1

6	1、3、11、45、1、1

また、CLIでのみ使用できる高度なDNSオプションもあります。CLIでこれらのオプションを使用するには、advancedproxyconfig > DNSコマンドを使用します。

次のいずれかのオプションを選択します。

- 0：常に順番にDNS応答を使用する
- 1：クライアントが指定したアドレスを使用し、次にDNS
- 2—DNSの使用に制限がある
- 3 – 非常に限定的なDNSの使用

オプション1および2では、Webレピュテーションが有効な場合はDNSが使用されます。

オプション2および3では、アップストリームプロキシがない場合、または設定されたアップストリームプロキシで障害が発生した場合に、明示的なプロキシ要求にDNSが使用されます。

すべてのオプションについて、ポリシーメンバーシップで宛先IPアドレスが使用される場合はDNSが使用されます。

これらのオプションは、クライアントの要求を評価する際に、SWAが接続先のIPアドレスをどのように決定するかを制御します。要求を受信すると、SWAは宛先IPアドレスとホスト名を確認します。SWAは、TCP接続の元の宛先IPアドレスを信頼するか、独自のDNS解決を実行して解決されたアドレスを使用するかを決定する必要があります。デフォルトは「0 = Always use DNS answers in order」です。これは、SWAがクライアントにIPアドレスを提供することを信頼しないことを意味します。

- オプション1:SWAは接続のためにクライアント指定のIPアドレスを試行しますが、失敗した場合は解決済みのアドレスにフォールバックします。解決されたアドレスは、ポリシー評価(Webカテゴリ、Webレピュテーションなど)に使用されます。
- オプション2:SWAは接続にクライアント指定のアドレスのみを使用し、フォールバックしません。解決されたアドレスは、ポリシー評価（Webカテゴリ、Webレピュテーションなど）に使用されます。
- オプション3:SWAは接続にクライアント指定のアドレスのみを使用し、フォールバックしません。クライアントが指定したIPアドレスは、ポリシー評価（Webカテゴリ、Webレピュテーションなど）に使用されます。

選択するオプションは、特定のホスト名の解決済みアドレスを決定する際に管理者がクライアントに設定する必要がある信頼度によって異なります。クライアントがダウンストリームプロキシの場合は、不要なDNSルックアップによる遅延の増加を回避するために、オプション3を選択します。

ロード バランシング

WCCPでは、最大8台のアプライアンスを使用する場合に、トラフィックの透過的なロードバランシングが可能です。これにより、ハッシュまたはマスクに基づいてトラフィックフローのバラ

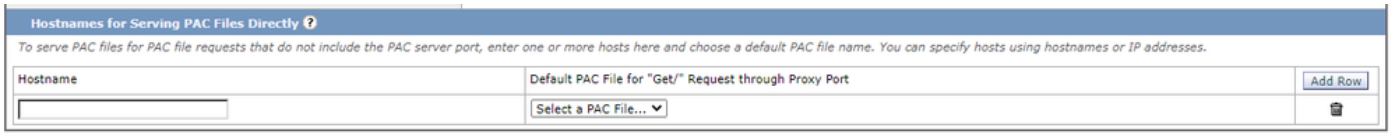
ンスを取ることができ、ネットワーク内にアプライアンスモデルが混在する場合に重み付けを行うことができ、ダウンタイムなしでデバイスをサービスプールに追加したりサービスプールから削除したりできます。8つのSWAで処理できる量を超えるニーズが発生した場合は、専用のロードバランサを使用することをお勧めします。

WCCP設定の具体的なベストプラクティスは、使用するプラットフォームによって異なります。Cisco Catalyst®スイッチのベストプラクティスは、Cisco Catalyst Instant Accessソリューションのホワイトペーパーに記載されています。

WCCPをCisco適応型セキュリティアプライアンス(ASA)と併用する場合には制限があります。つまり、クライアントのIPスプーフィングはサポートされていません。また、クライアントとSWAは同じインターフェイスの背後に配置する必要があります。このため、レイヤ4スイッチまたはルータを使用してトラフィックをリダイレクトする方が柔軟です。ASAプラットフォームでのWCCPの設定については、『ASAでのWCCP：概念、制限事項、および設定』を参照してください。

明示的な導入の場合、プロキシ自動設定(PAC)ファイルが最も広く導入されている方法ですが、このドキュメントでは説明できない多くの欠点とセキュリティの影響があります。PACファイルを展開する場合、ロケーションの構成にグループポリシーオブジェクト(GPO)を使用することをお勧めします。PACファイルは攻撃者の一般的なターゲットであり、誤って構成すると容易に悪用される可能性があるWebプロキシ自動検出プロトコル(WPAD)に依存しません。SWAは複数のPACファイルをホストし、ブラウザのキャッシュ内でそれらの有効期限を制御できます。

PACファイルは、設定可能なTCPポート番号（デフォルトでは9001）からSWAに直接要求できます。ポートが指定されていない場合、要求はアウトバウンドWeb要求であるかのように、プロキシプロセス自体に送信できます。この場合、要求に存在するHTTPホストヘッダーに基づいて特定のPACファイルを提供できます。



ハイアベイラビリティ環境で使用する場合は、Kerberosを別の方法で設定する必要があります。SWAはキータブファイルをサポートしているため、複数のホスト名をサービスプリンシパル名(SPN)に関連付けることができます。詳細は、『ハイアベイラビリティ導入環境でのKerberos認証用のWindows Active Directoryでのサービスアカウントの作成』を参照してください。

アクティブ認証

Kerberosは、NT LAN Manager(NTLM)Security Support Provider(NTLMSSP)よりも安全で、幅広くサポートされている認証プロトコルです。Apple OS Xオペレーティングシステムは NTLMSSPをサポートしていませんが、ドメインが参加していればKerberosを使用して認証できます。基本認証は使用しないでください。基本認証はHTTPヘッダー内で暗号化されずにクレデンシャルを送信し、ネットワーク上の攻撃者から簡単に傍受される可能性があります。基本認証を使用する必要がある場合は、クレデンシャルが暗号化されたトンネル経由で送信されるように

、クレデンシャルの暗号化を有効にする必要があります。

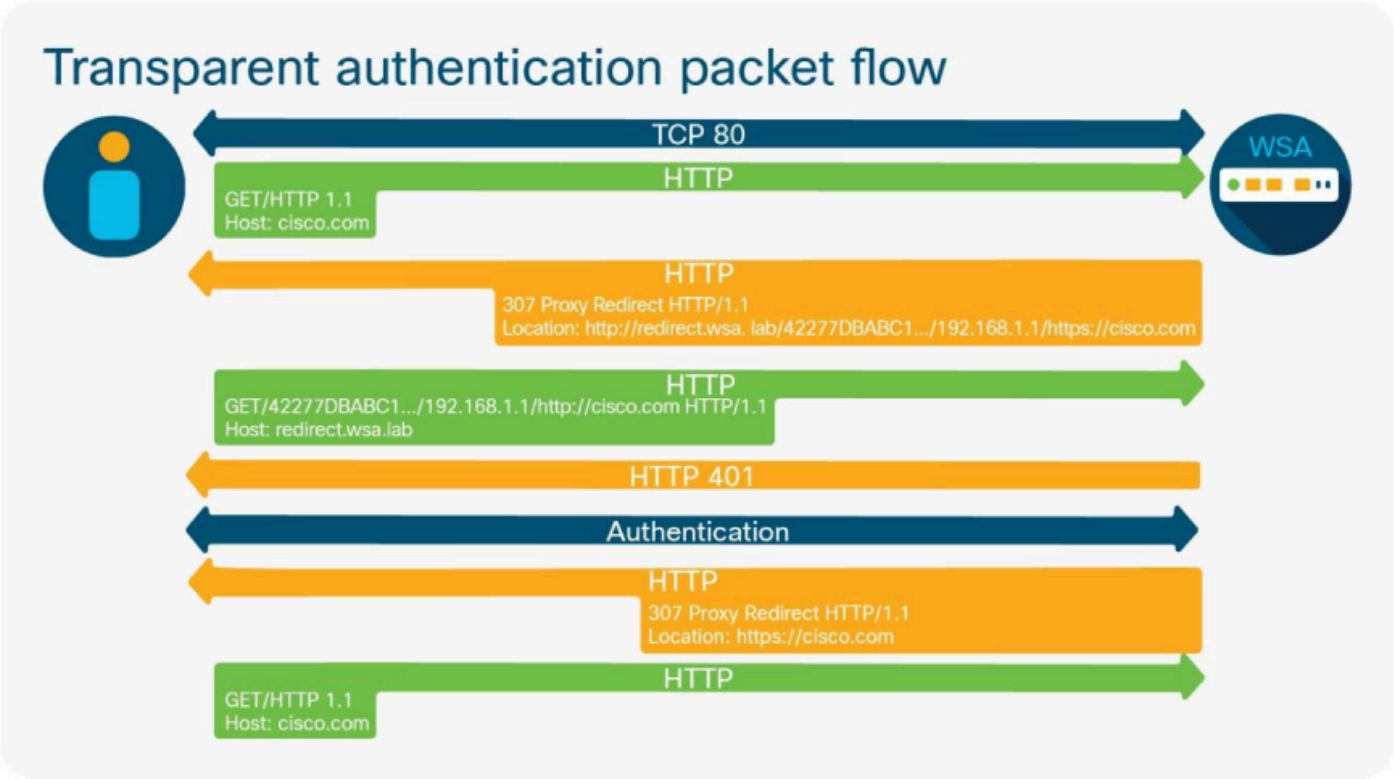
アベイラビリティを確保するために複数のドメインコントローラを設定に追加する必要がありますが、このトラフィックに固有のロードバランシングはありません。SWAは、設定されたすべてのドメインコントローラにTCP SYNパケットを送信し、最初に応答したドメインコントローラが認証に使用されます。

認証設定ページで設定されたリダイレクトホスト名によって、認証を完了するための透過クライアントの送信先が決まります。Windowsクライアントが統合認証を完了してシングルサインオン(SSO)を実現するには、インターネットオプションコントロールパネルの信頼済みサイトゾーンにリダイレクトホスト名がある必要があります。Kerberosプロトコルでは、リソースを指定するために完全修飾ドメイン名(FQDN)を使用する必要があります。つまり、Kerberosが意図的な認証メカニズムである場合は、「ショートネーム」（または「NETBIOS」名）は使用できません。FQDNは、手動で信頼済みサイトに追加する必要があります（グループポリシーなど）。さらに、Internet Optionsコントロールパネルで、ユーザ名とパスワードを使用した自動ログインを設定する必要があります。

ブラウザがネットワークプロキシによる認証を完了するには、Firefoxでも追加設定が必要です。これらの設定はabout:configページで設定できます。Kerberosが正常に完了するためには、network.negotiate-auth.trusted-urisオプションにリダイレクトホスト名を追加する必要があります。NTLMSSPの場合、network.automatic-ntlm-auth.trusted-urisオプションに追加する必要があります。

認証サロゲートは、認証が完了した後、設定された期間、認証されたユーザを記憶するために使用されます。発生するアクティブ認証イベントの数を制限するために、可能な限りIPサロゲートを使用する必要があります。クライアントのアクティブな認証は、特にKerberosが使用されている場合、リソースを大量に消費するタスクです。サロゲートタイムアウトはデフォルトで3600秒（1時間）で、これを短縮できますが、推奨値の最小値は900秒（15分）です。

次の図に、「redirect.WSA.lab」がリダイレクトホスト名として使用される方法を示します。



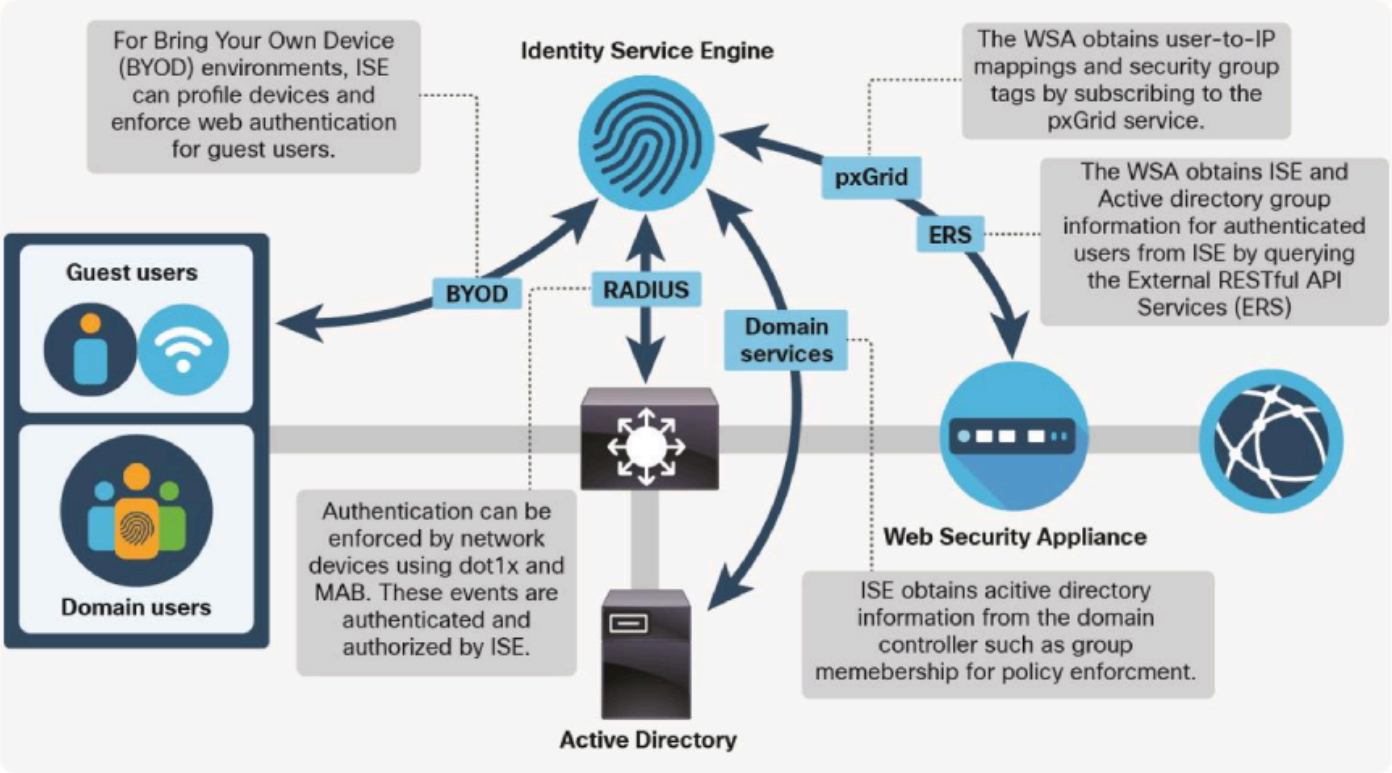
パッシブ認証

SWAは、他のシスコセキュリティプラットフォームを利用して、プロキシユーザを受動的に特定できます。ユーザをパッシブに識別することで、直接認証の課題やSWAからのActive Directory通信が不要になり、アプライアンスでの遅延やリソースの使用が軽減されます。現在利用可能なパッシブ認証のメカニズムは、Context Directory Agent(CDA)、Identity Services Engine(ISE)、およびIdentity Services Connector Passive Identity Connector(ISE-PIC)を介するものです。

ISEは機能豊富な製品で、管理者が認証サービスを一元化し、広範なネットワークアクセスコントロールを活用するのに役立ちます。ISEは（Dot1x認証またはWeb認証リダイレクトを通じて）ユーザ認証イベントについて学習すると、認証に関与するユーザとデバイスに関する情報を含むセッションデータベースを作成します。SWAはPlatform Exchange Grid(pxGrid)を介してISEに接続し、プロキシ接続に関連付けられたユーザ名、IPアドレス、セキュリティグループタグ(SGT)を取得します。AsyncOSバージョン11.7以降、SWAはISE上で外部Restfulサービス(ERS)に照会してグループ情報を取得することもできます。

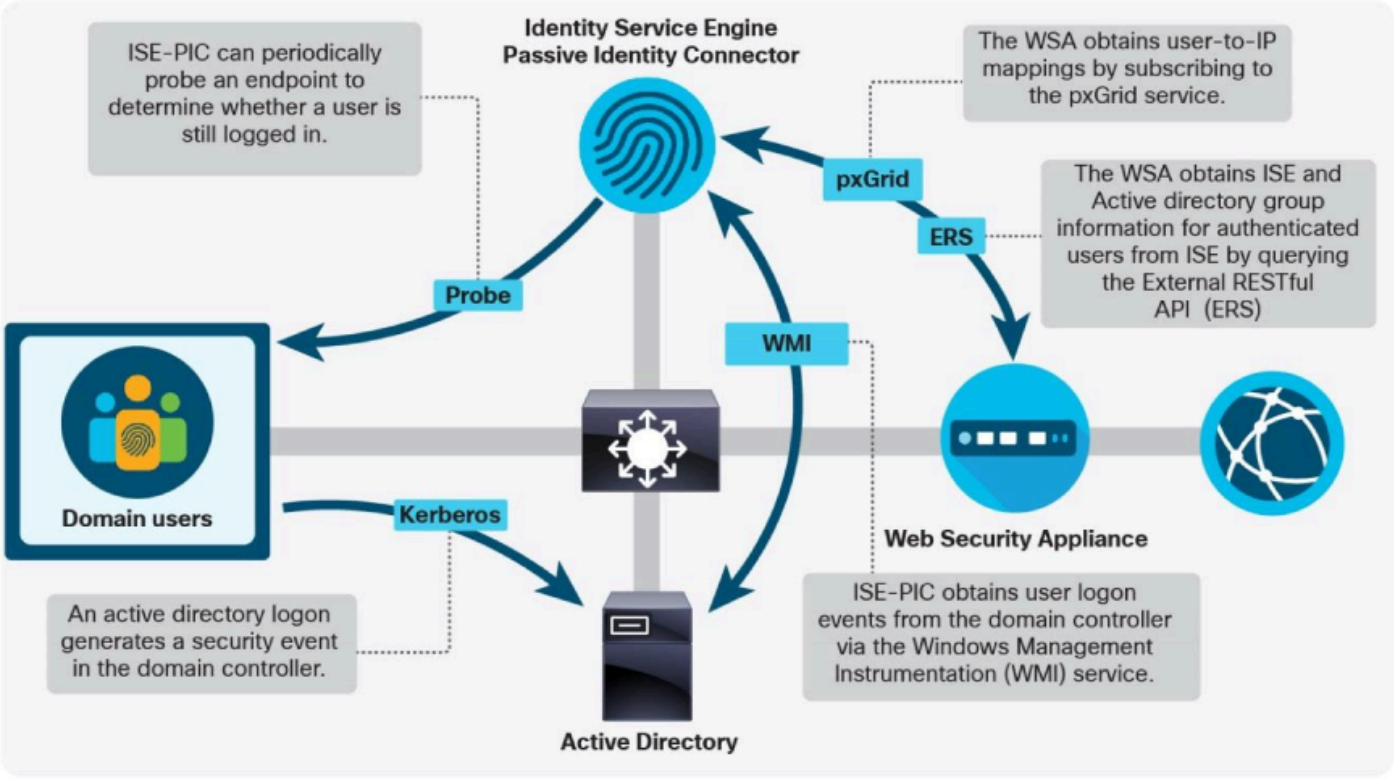
推奨されるバージョンはISE 3.1とSWA 14.0.2-X以降です。SWAのISE互換性マトリクスの詳細については、『セキュアWebアプライアンスのISE互換性マトリクス』を参照してください。

完全な統合手順の詳細については、『Webセキュリティアプライアンスエンドユーザガイド』を参照してください。



シスコは、Cisco Context Directory Agent(CDA)ソフトウェアのサポート終了を発表しました。『Cisco Context Directory Agent(CDA)』を参照してください。

CDAパッチ6では、Microsoft Server 2016と互換性があります。ただし、管理者はCDA環境をISE-PICに移行することを積極的に推奨します。どちらのソリューションもWMIを使用してWindowsセキュリティイベントログをサブスクライブし、ユーザからIPへのマッピング（「セッション」と呼ばれる）を生成します。CDAの場合、SWAはRADIUSを使用してこれらのマッピングを照会します。ISE-PICの場合、完全なISE導入と同じpxGridおよびERS接続が使用されます。ISE-PIC機能は、完全なISEインストールと、スタンドアロンの仮想アプライアンスで使用できます。

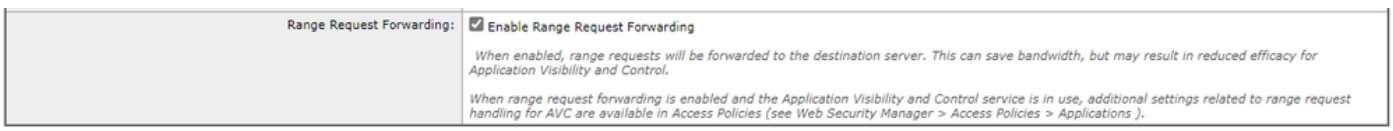


サービスの設定

Webプロキシ

帯域幅を節約し、パフォーマンスを向上させるために、Webプロキシ設定でキャッシングを有効にする必要があります。SWAはデフォルトではHTTPSトランザクションをキャッシュしないため、これはHTTPSトラフィックの割合が増加するにつれて重要性が低下します。プロキシが明示的なクライアントだけにサービスを提供するように展開されている場合、プロキシサービスを宛先としないトラフィックを拒否するには、転送モードを指定する必要があります。このようにして、アプライアンスの攻撃対象領域が縮小され、優れたセキュリティ原則が実践されます。不要な場合はオフにします。

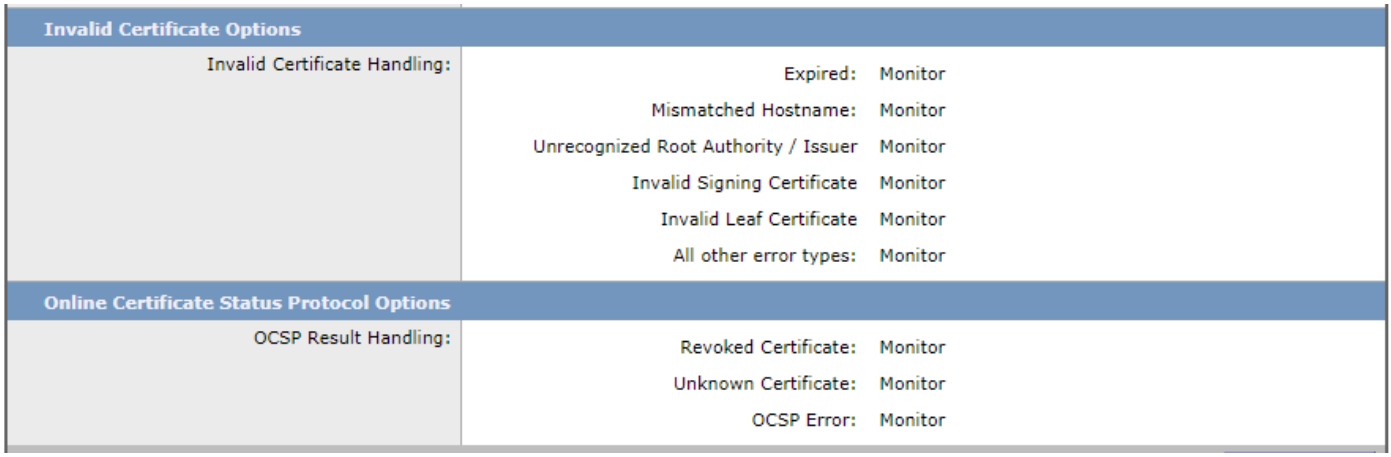
範囲要求ヘッダーは、ダウンロードするファイルのバイト範囲を指定するためにHTTP要求で使用されます。オペレーティングシステムやアプリケーションの更新デーモンが、ファイルの小さな部分を一度に転送する場合によく使用されます。デフォルトでは、SWAはウイルス対策(AV)スキャン、ファイルレピュテーションと分析、およびApplication Visibility Control(AVC)の目的でファイル全体を取得できるように、これらのヘッダーを削除します。プロキシ設定で範囲要求ヘッダーの転送をグローバルに有効にすると、管理者は、これらのヘッダーを転送または削除する個別のアクセスポリシーを作成できます。この設定の詳細については、「アクセスポリシー」の項で説明します。



HTTPSプロキシ

セキュリティのベストプラクティスでは、秘密鍵が使用されるアプライアンス上で生成される必要があり、他の場所には転送されないことが推奨されています。HTTPSプロキシウィザード(HTTPS-proxy-wizard)では、Transport Layer Security(TLS)接続の復号化に使用するキーペアと証明書を作成できます。その後、証明書署名要求(CSR)をダウンロードし、社内の認証局(CA)によって署名することができます。Active Directory(AD)環境では、ADに統合されたCAはドメインのすべてのメンバーによって自動的に信頼され、証明書を展開するために追加の手順を必要としないため、この方法が最適です。

HTTPSプロキシのセキュリティ機能の1つは、サーバ証明書を検証することです。ベストプラクティスでは、無効な証明書では接続をドロップする必要があることを推奨しています。EUNの復号化を有効にすると、SWAはブロックの理由を説明するブロックページを表示できます。これを有効にしないと、ブロックされたHTTPSサイトでブラウザエラーが発生します。これにより、ヘルプデスクチケットが増加し、SWAが接続をブロックしたという認識ではなく、何かが壊れているというユーザの推測が生じます。すべての無効な証明書オプションは、少なくともDecryptに設定する必要があります。証明書の問題によってサイトの読み込みが妨げられる場合、これらのオプションのいずれかをモニタのままにしても、有用なエラーメッセージは記録されません。



同様に、Online Certificate Services Protocol(OCSP)チェックは有効のままにしておく必要があり、どのオプションにもモニタを使用しないでください。関連するエラーメッセージをログに記録するには、失効した証明書を削除し、その他すべての証明書を少なくともDecryptに設定する必要があります。Authority Information Access Chasing(AIA)は、クライアントが証明書の署名者を収集できる手段と、追加の証明書をフェッチできるURLです。たとえば、サーバから受信した証明書チェーンが不完全である（中間証明書またはルート証明書がない）場合、SWAはAIAフィールドをチェックし、それを使用して不足している証明書を取得し、信頼性を確認できます。この設定は、CLIで次のコマンドからのみ使用できます。

```
SWA_CLI> advancedproxyconfig

Choose a parameter group:
- AUTHENTICATION - Authentication related parameters
- CACHING - Proxy Caching related parameters
- DNS - DNS related parameters
- EUN - EUN related parameters
- NATIVEFTP - Native FTP related parameters
- FTPOVERHTTP - FTP Over HTTP related parameters
- HTTPS - HTTPS related parameters
```

```
- SCANNING - Scanning related parameters
- PROXYCONN - Proxy connection header related parameters
- CUSTOMHEADERS - Manage custom request headers for specific domains
- MISCELLANEOUS - Miscellaneous proxy related parameters
- SOCKS - SOCKS Proxy parameters
- CONTENT-ENCODING - Block content-encoding types
- SCANNERS - Scanner related parameters
[]> HTTPS
...
Do you want to enable automatic discovery and download of missing Intermediate Certificates?
[Y]>
...
```

注：この設定はデフォルトで有効になっていますが、最新のサーバの多くはこのメカニズムに依存してクライアントに完全な信頼チェーンを提供しているため、無効にしないでください。

レイヤ4トラフィックモニタ(L4TM)

L4TMは、SWAの範囲を拡張して、プロキシを通過しない悪意のあるトラフィックを含め、すべてのTCPポートとUDPポートのトラフィックを含めるための非常に効果的な方法です。T1およびT2ポートは、ネットワークタップまたはスイッチモニタセッションに接続されます。これにより、SWAはクライアントからのすべてのトラフィックを受動的に監視できます。悪意のあるIPアドレス宛てのトラフィックが見つかった場合、SWAはサーバIPアドレスをスプーフィングしながらRSTを送信することでTCPセッションを終了できます。UDPトラフィックの場合は、Port Unreachableメッセージを送信できます。モニタセッションを設定する際には、SWAの管理インターフェイス宛てのトラフィックを除外して、この機能がデバイスへのアクセスを妨害しないように設定するのが最善です。

L4TMは、悪意のあるトラフィックを監視するだけでなく、バイパス設定リストを更新するためにDNSクエリーをスヌーピングします。このリストは、Webサーバへの直接ルーティングのために特定の要求をWCCPルータに返すために、WCCP展開で使用されます。バイパス設定リストに一致するパケットは、プロキシによって処理されません。リストには、IPアドレスまたはサーバ名を含めることができます。SWAは、レコードのTTLにかかわらず、バイパス設定リストのすべてのエントリを30分ごとに解決します。ただし、L4TM機能が有効になっている場合、SWAはスヌーピングされたDNSクエリを使用して、これらのレコードをより頻繁に更新できます。これにより、クライアントがSWAとは異なるアドレスを解決した場合に、誤検出のリスクが軽減されます。

ポリシー設定

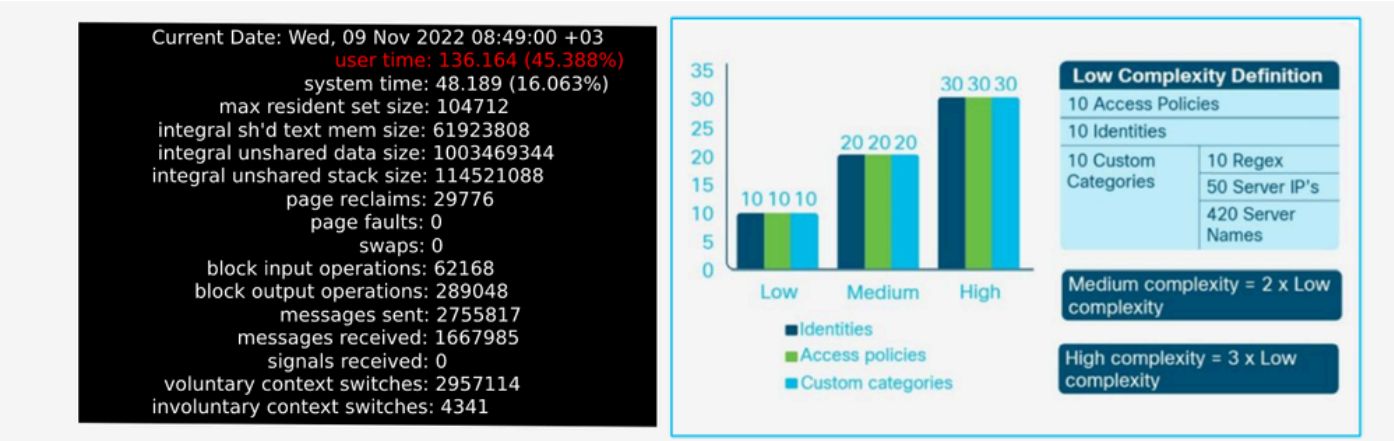
正しいポリシー設定は、SWAのパフォーマンスとスケーラビリティの中核を成します。これは、クライアントの保護と企業の要件の適用においてポリシー自体が効果的であるだけでなく、設定されるポリシーがリソースの使用率やSWA全体の健全性とパフォーマンスに直接影響を与えることも理由です。ポリシーが複雑すぎたり、不適切に設計されていると、アプライアンスが不安定になったり、応答が遅くなったりすることがあります。

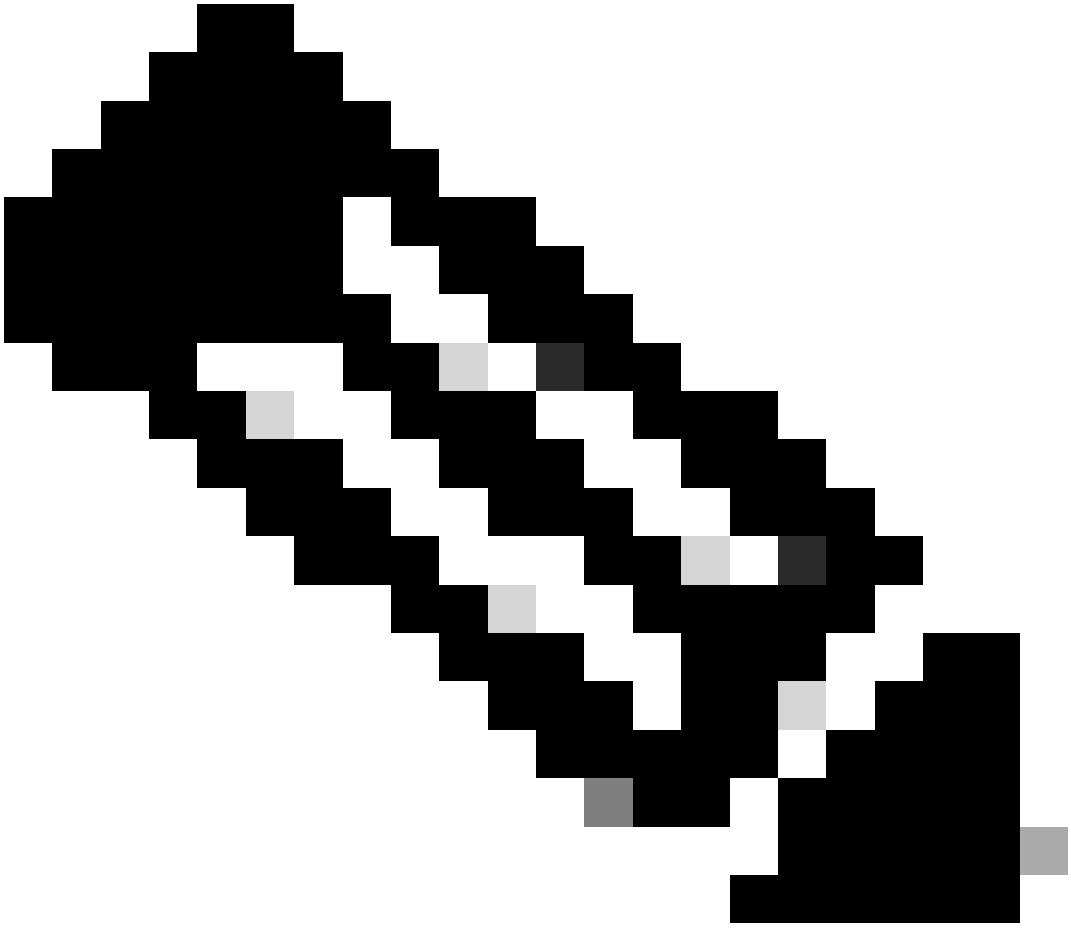
複雑度

SWAポリシーの構築には、さまざまなポリシー要素が使用されます。設定から生成されたXMLファイルは、多数のバックエンド設定ファイルとアクセスルールを作成するために使用されます。設定が複雑になればなるほど、プロキシプロセスが各トランザクションのさまざまなルールセットの評価に費やす時間は長くなります。SWAのベンチマークとサイジングでは、設定の複雑さの3つの階層を表すポリシー要素の基本セットが作成されます。10個のアイデンティティプロファイル、復号化ポリシー、アクセスポリシー、および10個の正規表現エントリ、50個のサーバIPアドレス、420個のサーバホスト名を含む10個のカスタムカテゴリは、低複雑度の設定と見なされます。これらの各数値に2と3を掛けると、それぞれ中複雑度と高複雑度の設定になります。

設定が複雑すぎると、通常、最初の症状としてWebインターフェイスやCLIでの応答が遅くなります。最初は、ユーザに大きな影響を与えることはできません。ただし、設定が複雑であるほど、プロキシプロセスがユーザモードで費やす時間は長くなります。このため、このモードで費やされた時間のパーセンテージのチェックは、SWAの速度低下の原因として非常に複雑な設定を診断する場合に便利な方法です。

CPU時間（秒単位）は、5分ごとにtrack_statsログに記録されます。つまり、ユーザ時間のパーセンテージは、（ユーザ時間+システム時間）/300として計算できます。ユーザ時間が270に近づくと、プロセスはユーザモードで大量のCPUサイクルを消費します。これはほぼ常に、設定が複雑すぎて効率的に解析できないことが原因です。





注:SWAの同時クライアント接続数の上限は、現時点で60,000、同時サーバ接続数では60,000です。

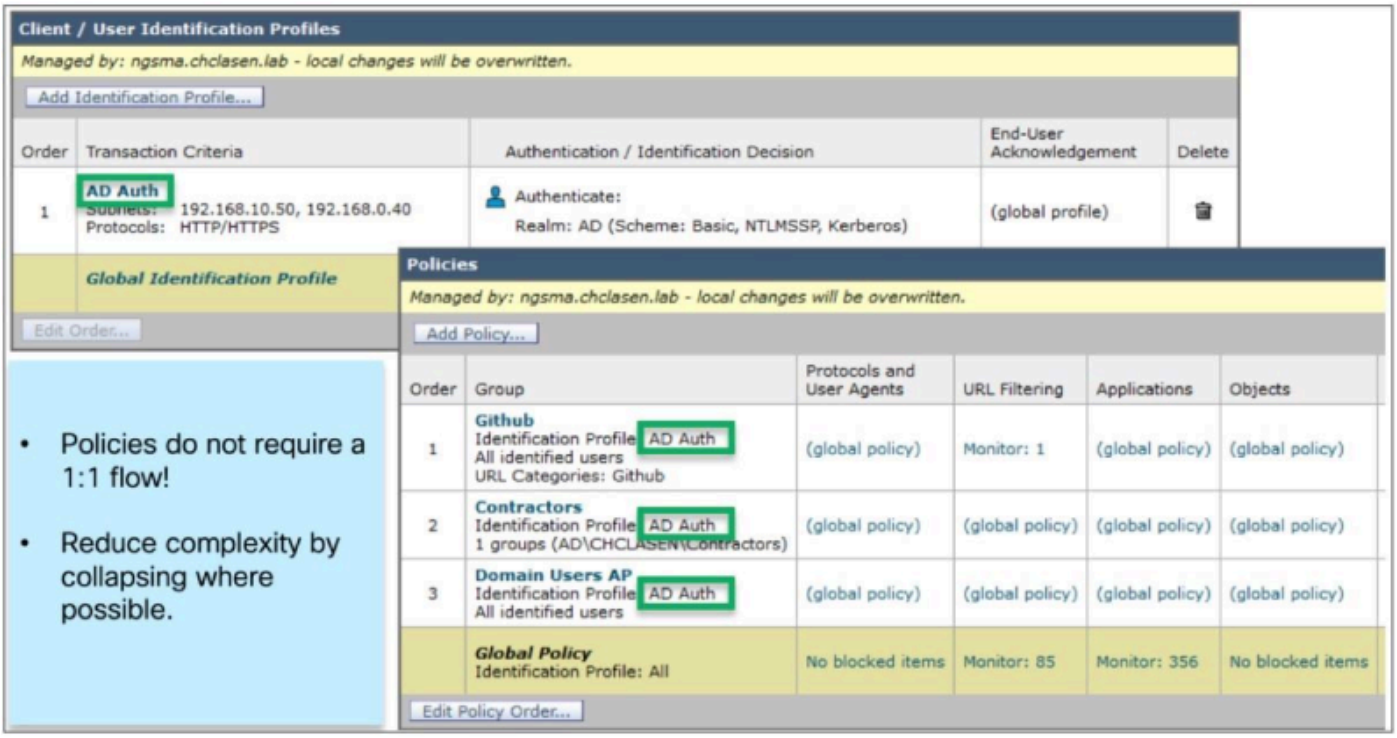
識別プロファイル

識別(ID)プロファイルは、新しい要求が受信されたときに評価される最初のポリシー要素です。IDプロファイルの最初のセクションで設定されたすべての情報は、論理ANDで評価されます。つまり、要求がプロファイルに一致するには、すべての条件が一致する必要があります。ポリシーを作成する場合は、絶対に必要な条件を満たす必要があります。個々のホストアドレスを含むプロファイルはほとんど必要なく、無秩序な設定につながる可能性があります。HTTPヘッダー、カスタムカテゴリリスト、またはサブネットにあるユーザエージェント文字列を利用する方が、一般にプロファイルの範囲を制限するより優れた方法です。

一般に、認証が必要なポリシーは下部で設定され、その上に例外が追加されます。認証を必要としないポリシーをオーダーする場合は、最も使用するポリシーを可能な限り上位に配置する必要があります。失敗した認証に依存してアクセスを制限しないでください。ネットワーク上のクライアントがプロキシに対して認証できないことが判明している場合は、認証を免除し、アクセス

ポリシーでブロックする必要があります。認証できないクライアントは、認証されていない要求をSWAに繰り返し送信します。これにより、リソースが使用され、CPUとメモリの過剰使用が引き起こされる可能性があります。

管理者は、一意のIDプロファイルと、対応する復号化ポリシーおよびアクセスポリシーが必要であると誤解することがよくあります。これは、ポリシー設定に対する非効率的な戦略です。1つのIDプロファイルを複数の復号化ポリシーおよびアクセスポリシーに関連付けられるように、可能な限りポリシーを「折りたたむ」必要があります。これは、トラフィックがポリシーに一致するためには、特定のポリシー内のすべての条件が一致する必要があるためです。認証ポリシーをより一般的にし、その結果得られるポリシーをより詳細にすることで、全体としてポリシーの数を減らすことができます。



復号化ポリシー

IDプロファイルと同様に、復号化ポリシーで設定された基準も、ISEからの情報が使用される場合の1つの重要な例外を除いて、論理ANDとして評価されます。設定されている要素（ADグループ、ユーザ、またはSGT）に応じて、ポリシー照合がどのように機能するかを次に示します。

- ADグループとユーザ：以前の動作に変更はありません。ユーザがグループのメンバーであるか、ポリシーでユーザが指定されている場合は、ポリシーが照合されます。
- SGTおよびADグループおよびユーザ：ユーザがSGTに関連付けられておりADグループのメンバーである場合、またはポリシーでユーザが指定されている場合、ポリシーは一致します。
- SGTおよびユーザ：ユーザがSGTに関連付けられているか、ポリシーでユーザが指定されている場合、ポリシーは照合されます。

SWAによって実行されるすべてのサービスの中で、パフォーマンスの観点から見ると、HTTPSトラフィックの評価が最も重要です。復号化されたトラフィックの割合は、アプライアンスのサイジング方法に直接影響します。管理者は、Webトラフィックの少なくとも75 %をHTTPSと見な

すことができます。

最初のインストールの後、復号化されたトラフィックの割合を決定して、将来の成長に対する期待が正確に設定されるようにする必要があります。導入後は、四半期に1回、この数値を確認する必要があります。SWAによって復号化されたHTTPSトラフィックの割合は、追加のログ管理ソフトウェアがなくても、access_logsのコピーを使用して簡単に確認できます。この数値は、単純なBashコマンドまたはPowerShellコマンドを使用して取得できます。各環境について説明する手順を次に示します。

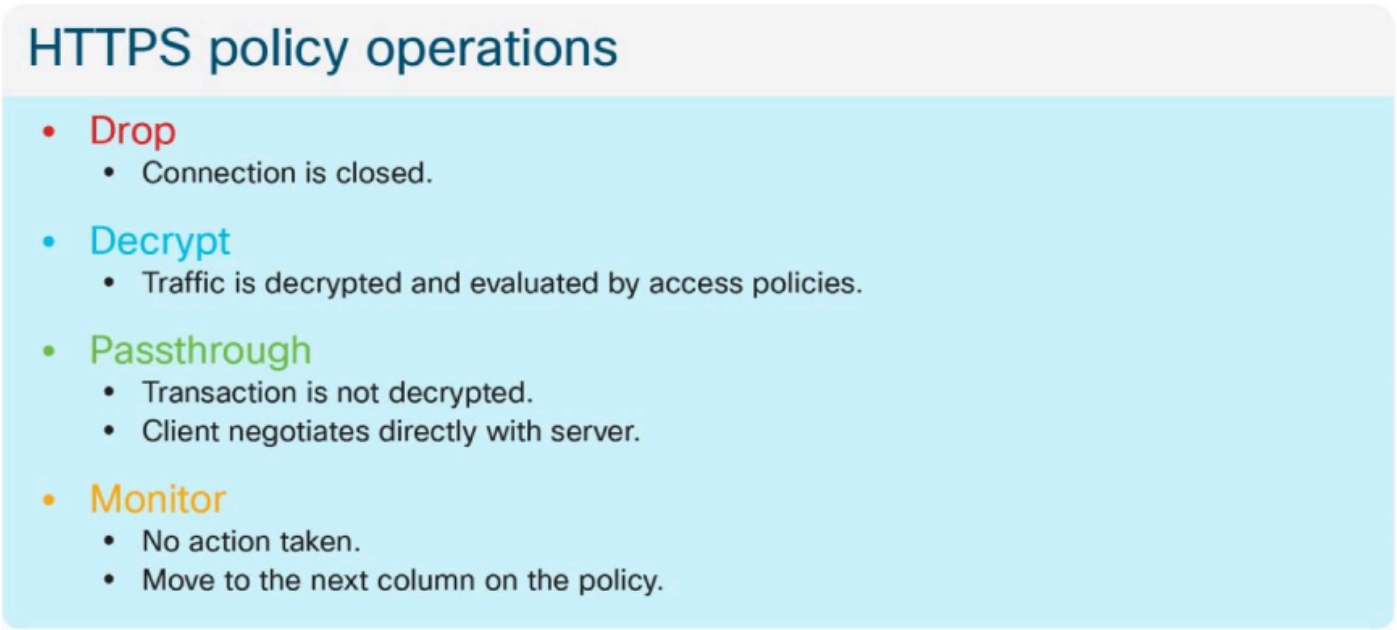
1. Linuxコマンド：

```
cat aclog.current | grep -Ev "\/407|\/401" | awk 'BEGIN { total=0; decrypt=0; ssl=0;} {total++; if($0 ~
```

2. Powershellコマンド：

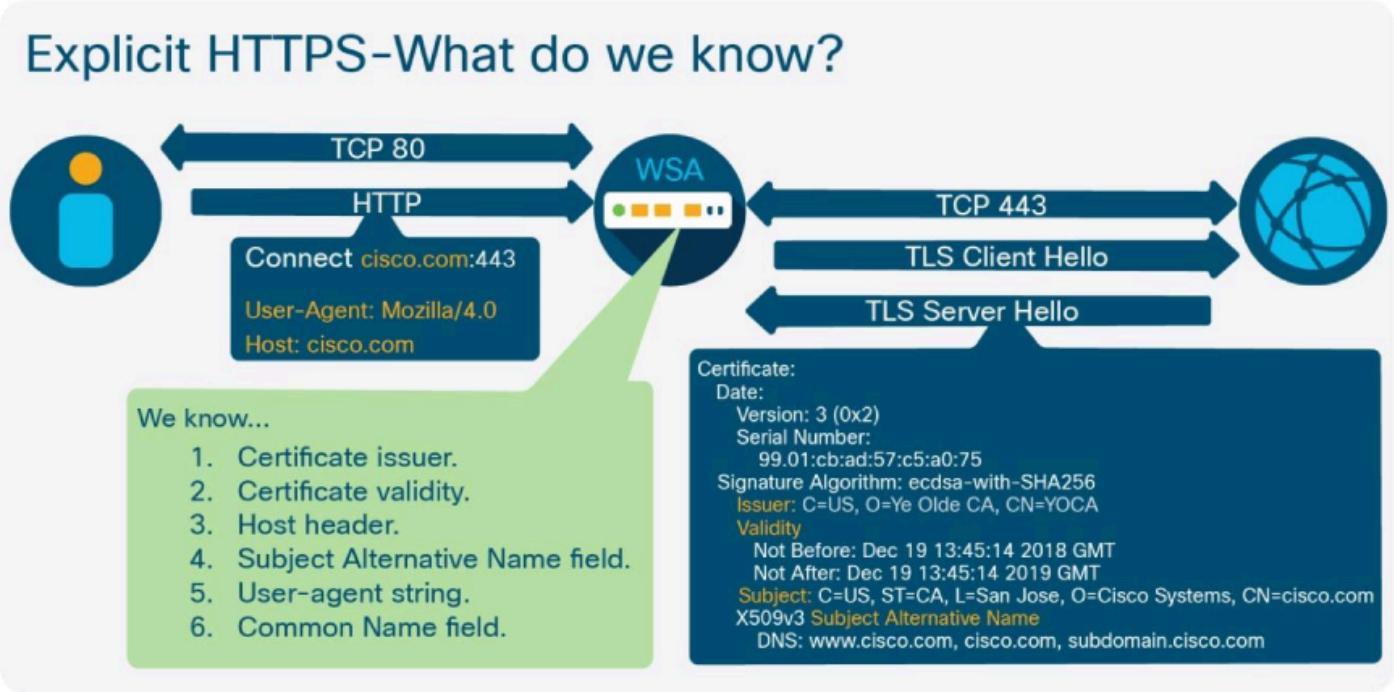
```
$lines = Get-Content -Path "aclog.current" | Where-Object { $_ -notmatch "/407|/401" }; $total = 0; $de
```

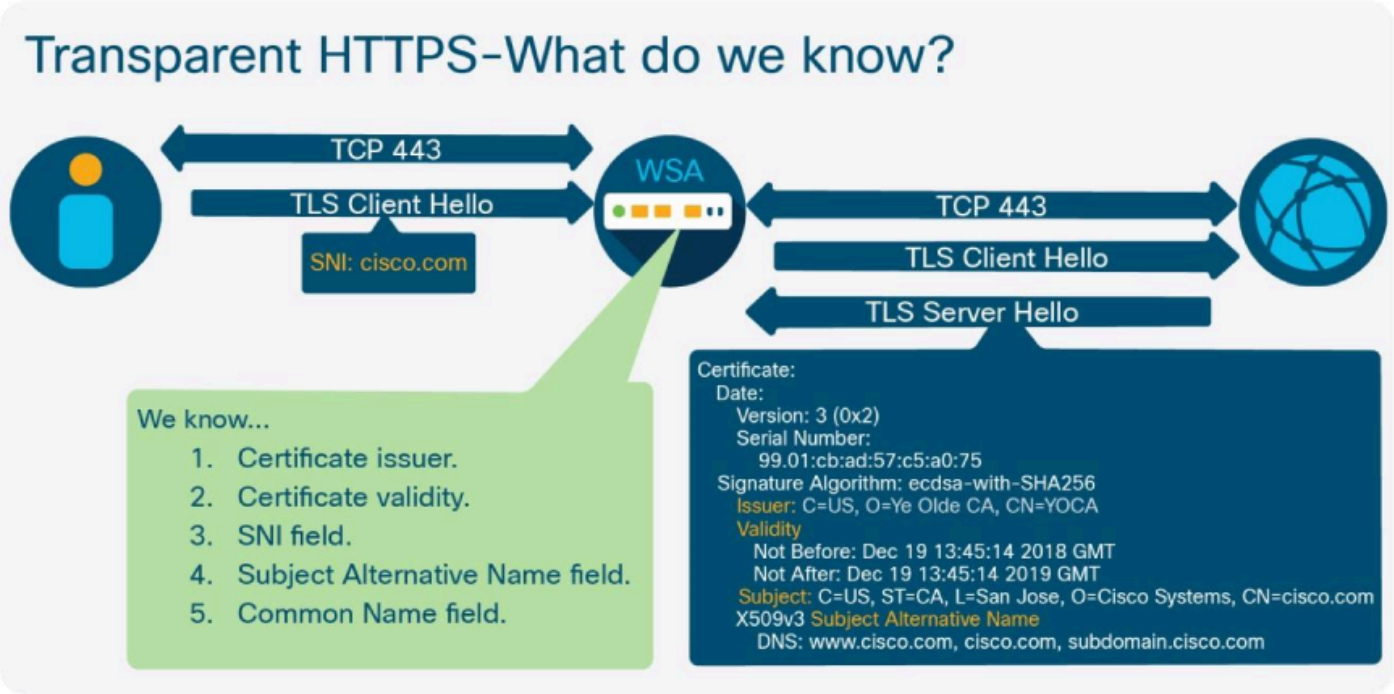
復号化ポリシーを設計する際は、ポリシーに記載されているさまざまなアクションによって、アプライアンスがどのようにHTTPS接続を評価するかを理解することが重要です。パススルーアクションは、SWAがすべてのパケットを復号化せずに、クライアントとサーバがTLSセッションの各終端を終端できるようにする必要がある場合に使用されます。サイトがパススルーに設定されている場合でも、SWAはサーバとの1つのTLSハンドシェイクを完了する必要があります。これは、SWAが証明書の有効性に基づいて接続をブロックすることを選択し、証明書を取得するためにサーバとのTLS接続を開始する必要があるためです。証明書が有効な場合、SWAは接続を閉じ、クライアントがサーバと直接セッションのセットアップを続行できるようにします。



SWAがTLSハンドシェイクを実行しない唯一のケースは、サーバ名またはIPアドレスがカスタムカテゴリに存在し、それがパススルーに設定されていて、サーバ名がHTTP CONNECTまたはTLS Client Helloのいずれかで使用できる場合です。明示的なシナリオでは、クライアントはTLSセッションを開始する前に（ホストヘッダー内で）プロキシにサーバのホスト名を提供するため、このフィールドはカスタムカテゴリと照合されます。透過的な導入では、SWAはTLS Client HelloメッセージのServer Name Indication(SNI)フィールドをチェックし、カスタムカテゴリと照合して評価します。ホストヘッダーまたはSNIがない場合、SWAはサーバとのハンドシェイクを続行し、証明書のサブジェクト代替名(SAN)フィールドと共通名(CN)フィールドをその順序で確認する必要があります。

ポリシー設計におけるこの動作の意味は、SWAがサーバとのTLSハンドシェイクを完了する必要があるWebカテゴリやレピュテーションスコアに依存せずに、既知の内部で信頼されるサーバを判別してカスタムカテゴリリストからのパススルーに設定することで、TLSハンドシェイクの数を減らすことができるということです。ただし、証明書の有効性チェックの妨げにもなる点に注意してください。





新しいサイトがWebに出現する速度を考えると、SWAで使用されるWebレピュテーションおよび分類データベースによって分類されていないサイトが多数ある可能性があります。これは、サイトが悪意のある可能性が必ずしも高いことを示しているわけではありません。さらに、これらすべてのサイトは、AVスキャン、AMPファイルのレピュテーションと分析、および設定されているオブジェクトのブロックまたはスキャンの対象となります。これらの理由から、カテゴリ化されていないサイトを削除することはお勧めできません。AVエンジンによって復号化およびスキャンされ、AVC、AMP、アクセスポリシーなどによって評価されるように設定するのが最善です。カテゴリ化されていないサイトの詳細については、「アクセスポリシー」セクションを参照してください。

アクセスポリシー

IDプロファイルと同様に、復号化ポリシーで設定された基準は、ISEからの情報が使用される際に、1つの重要な例外を除いて論理的なANDとしても評価されます。次に、設定されている要素（ADグループ、ユーザ、またはSGT）に基づくポリシー照合の動作について説明します。

- ADグループとユーザ：以前の動作に変更はありません。ユーザがグループのメンバーであるか、ポリシーでユーザが指定されている場合は、ポリシーが照合されます。
- SGTおよびADグループおよびユーザ：ユーザがSGTに関連付けられておりADグループのメンバーである場合、またはポリシーでユーザが指定されている場合、ポリシーは一致します。
- SGTおよびユーザ：ユーザがSGTに関連付けられているか、ポリシーでユーザが指定されている場合、ポリシーは照合されます。

HTTPトラフィックは、認証後すぐにアクセスポリシーに対して評価されます。HTTPSトラフィックは、認証された後、および一致する復号化ポリシーに従って復号化アクションが適用されているかどうかを評価されます。復号化された要求には、2つのaccess_logエントリがあります。最初のログエントリは、初期TLS接続（復号化）に適用されるアクションを示し、2番目のログエントリは、復号化されたHTTP要求に対してアクセスポリシーによって適用されるアクションを示

します。

「Webプロキシ」セクションで説明されているように、範囲要求ヘッダーは、ファイルの特定のバイト範囲を要求するために使用され、一般的にOSおよびアプリケーション更新サービスによって使用されます。SWAでは、ファイル全体がないとマルウェアスキャンの実行やAVC機能の利用が不可能になるため、デフォルトでこれらのヘッダーが発信要求から削除されます。ネットワーク上の多くのホストが更新を取得するために小さなバイト範囲を頻繁に要求している場合、これによってSWAがファイル全体を同時に数回ダウンロードする可能性があります。これにより、使用可能なインターネット帯域幅が急速に枯渇し、サービスが停止する可能性があります。この障害の最も一般的な原因は、Microsoft WindowsのアップデートとAdobeソフトウェアのアップデートデーモンです。

これを軽減する最善のソリューションは、このトラフィックをSWAに完全に誘導することです。これは透過的に展開される環境では必ずしも実現できません。このような場合は、トラフィック専用のアクセスポリシーを作成し、それらのポリシーに対して範囲要求ヘッダー転送を有効にするのが次善のオプションです。これらの要求に対してはAVスキャンとAVCが不可能であることを考慮する必要があります。そのため、ポリシーは目的のトラフィックだけを対象として慎重に設計する必要があります。多くの場合、これを実現する最善の方法は、要求ヘッダーにあるユーザエージェント文字列を照合することです。一般的な更新デーモンのユーザエージェント文字列は、オンラインで見つけることも、管理者が要求をキャプチャして確認することもできます。Microsoft WindowsおよびAdobeソフトウェアアップデートを含むほとんどのアップデートサービスは、HTTPSを使用していません。

「復号化ポリシー」セクションで説明したように、未分類のサイトを復号化ポリシーで廃棄することは推奨されません。同じ理由から、アクセスポリシーでこれらをブロックすることはお勧めしません。動的コンテンツ分析(DCA)エンジンは、特定のサイトのコンテンツを、他のヒューリスティックデータと共に使用して、分類されたサイトを作成できます。分類されていないサイトは、URLデータベース検索によって分類されていないとマークされます。この機能を有効にすると、SWA内の未分類の判定の数が減少します。

アクセスポリシーのオブジェクトスキャン設定では、複数のタイプのアーカイブファイルを検査できます。ネットワークがアプリケーションアップデートの一部としてアーカイブファイルを定期的にダウンロードする場合、アーカイブファイルの検査を有効にすると、CPU使用率が大幅に増加する可能性があります。すべてのアーカイブファイルを検査する場合は、このトラフィックを事前に識別し、除外する必要があります。このトラフィックを識別するために考えられる方法を調査する最初の場所は、ユーザエージェント文字列です。これは、維持が面倒になる可能性のあるIP allowed-listを回避するのに役立ちます。

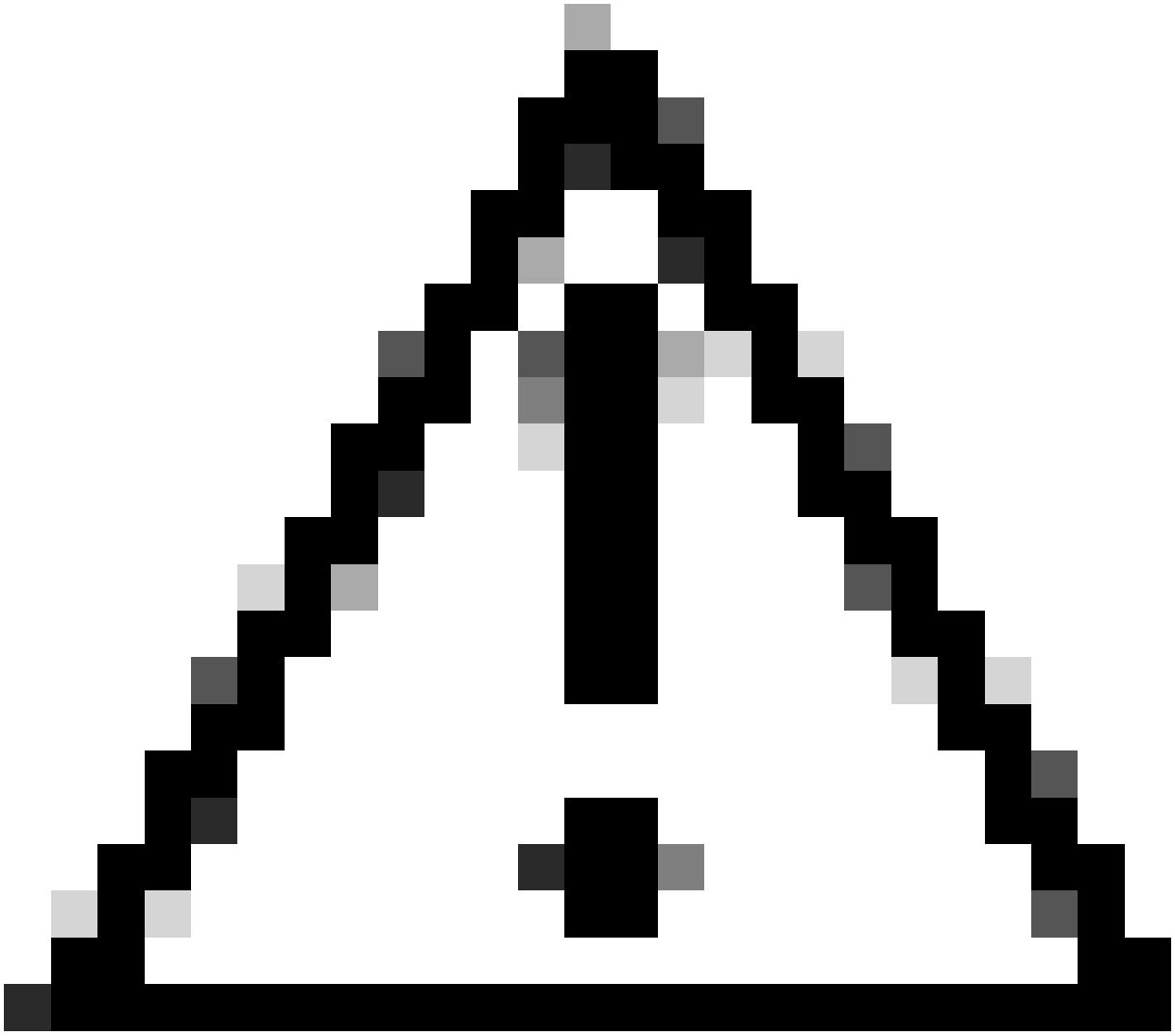
カスタムおよび外部URLカテゴリ

カスタムカテゴリリストは、IPアドレスまたはホスト名でサーバを識別するために使用されます。正規表現(regex)を使用して、サーバ名を照合するパターンを指定できます。サーバ名の照合に正規表現パターンを使用する方が、部分文字列の照合を使用するよりもリソースの負荷がはるか

に大きいため、正規表現パターンは絶対に必要な場合にのみ使用する必要があります。正規表現を使用せずにサブドメインを照合するために、ドメイン名の先頭に「。」を追加できます。たとえば、「.cisco.com」は「www.cisco.com」とも一致します。
「複雑度」セクションで説明したように、低複雑度は10のカスタムカテゴリリストとして定義され、中複雑度は20、高複雑度は30として定義されます。特にリストが正規表現パターンを使用する場合やエントリが多い場合は、この数値を20未満にしておくことをお勧めします。各タイプのエントリ数についての詳細は、「アクセスポリシー」セクションを参照してください。

外部URLフィードは、静的なカスタムカテゴリリストよりもはるかに柔軟です。外部URLフィードを利用すると、管理者が手動で管理する必要がなくなるため、セキュリティに直接影響する可能性があります。この機能は、SWA管理者によって管理されていないリストを取得するために使用できるため、ダウンロードされたアドレスに個別の例外を追加する機能は、AsyncOSバージョン11.8で追加されました。

Office 365 APIは、この一般的に展開されるサービスのポリシー決定に特に役立ち、個々のアプリケーション（PowerPoint、Skype、Wordなど）に利用できます。Microsoftでは、パフォーマンスを最適化するために、すべてのOffice 365トラフィックのプロキシをバイパスすることをお勧めします。Microsoftのドキュメントには次のように記載されています。



注意:「SSLのBreak and Inspectが最大の遅延を引き起こしているのに対し、プロキシ認証やレピュテーションルックアップなどの他のサービスは、パフォーマンスの低下やユーザエクスペリエンスの低下を引き起こす可能性があります。さらに、これらの境界ネットワークデバイスには、すべてのネットワーク接続要求を処理するのに十分な容量が必要です。直接のOffice 365ネットワーク要求にプロキシまたはインスペクションデバイスをバイパスすることをお勧めします。」 - https://learn.microsoft.com/en-us/microsoft-365/enterprise/managing-office-365-endpoints?view=o365-worldwide

---

トランスペアレントプロキシ環境でこのガイダンスを使用することは困難な場合があります。AsyncOSバージョン11.8以降では、Office365 APIから取得した動的なカテゴリリストを使用してバイパス設定リストにデータを入力できます。このリストは、直接ルーティングのために透過的にリダイレクトされたトラフィックをWCCPデバイスに送信するために使用されます。

すべてのOffice 365トラフィックをバイパスすると、このトラフィックの基本的なセキュリティ制御とレポート機能を必要とする管理者にとって、盲点となります。Office 365トラフィックがSWAによってバイパスされない場合は、発生する可能性のある特定の技術的課題を理解すること

が重要です。その1つが、アプリケーションに必要な接続数です。サイジングは、Office 365アプリケーションに必要な追加の永続的なTCP接続に対応するように適切に調整する必要があります。これにより、ユーザあたりの永続的なTCPセッションの総数が10 ~ 15増加する可能性があります。

HTTPSプロキシによって実行される復号化と再暗号化のアクションによって、接続に少しの遅延が生じます。Office 365アプリケーションは遅延の影響を非常に受けやすく、WAN接続の遅さや地理的に分散しているなどの要因によって遅延が悪化すると、ユーザエクスペリエンスが低下する可能性があります。

一部のOffice 365アプリケーションは独自のTLSパラメータを使用するため、HTTPSプロキシはアプリケーションサーバーとのハンドシェイクを完了できません。これは、証明書の検証またはホスト名の取得に必要です。これを、TLS Client HelloメッセージでServer Name Indication(SNI)フィールドを送信しないSkype for Businessなどのアプリケーションと組み合わせると、このトラフィックを完全にバイパスする必要が生じます。AsyncOS 11.8では、宛先IPアドレスのみに基づいてトラフィックをバイパスする機能が導入されており、このシナリオに対処するための証明書チェックは用意されていません。

モニタとアラート

CLIモニタ

SWA CLIには、重要なプロセスをリアルタイムで監視するコマンドがあります。最も便利なのは、proxプロセスに関連する統計情報を表示するコマンドです。status detailコマンドは、リソースの使用状況やパフォーマンスに関する測定基準 (稼働時間、使用帯域幅、応答遅延、接続数など) の要約に適しています。このコマンドの出力例を次に示します。

```
SWA_CLI>  status detail

Status as of:                  Fri Nov 11 14:06:52 2022 +03
Up since:                      Fri Apr 08 10:15:00 2022 +03 (217d 3h 51m 52s)
System Resource Utilization:
  CPU                                3.3%
  RAM                                6.2%
  Reporting/Logging Disk            45.6%
Transactions per Second:
  Average in last minute               55
  Maximum in last hour                201
  Average in last hour                 65
  Maximum since proxy restart       1031
  Average since proxy restart          51
Bandwidth (Mbps):
  Average in last minute            4.676
  Maximum in last hour            327.258
  Average in last hour             10.845
  Maximum since proxy restart    1581.297
  Average since proxy restart      11.167
Response Time (ms):
  Average in last minute               635
```

```
  Maximum in last hour                  376209
  Average in last hour                     605
  Maximum since proxy restart          2602943
  Average since proxy restart             701
Cache Hit Rate:
  Average in last minute                    0
  Maximum in last hour                      2
  Average in last hour                      0
  Maximum since proxy restart              15
  Average since proxy restart               0
Connections:
  Idle client connections                 186
  Idle server connections                 184
  Total client connections               3499
  Total server connections               3632
SSLJobs:
  In queue Avg in last minute               4
  Average in last minute                45214
  SSLInfo Average in last min              94
Network Events:
  Average in last minute                  0.0
  Maximum in last minute                   35
  Network events in last min           124502
```

rateコマンドを使用すると、proxプロセスが使用するCPUのパーセンテージ、Requests Per Second（RPS；要求/秒）数、およびキャッシュ統計情報に関するリアルタイム情報が表示されます。このコマンドは、中断されるまでポーリングを続け、新しい出力を表示します。このコマンドの出力例を次に示します。

```
SWA_CLI>  rate

Press Ctrl-C to stop.
  %proxy  reqs                          client    server    %bw  disk  disk
     CPU  /sec   hits blocks misses     kb/sec    kb/sec  saved   wrs   rds
  5.00    51      1    147    370       2283      2268    0.6    48    37
  4.00    36      0    128    237      21695     21687    0.0    47    38
  4.00    48      2    179    307       8168      8154    0.2    65    33
  5.00    53      0    161    372       2894      2880    0.5    48    32
  6.00    52      0    198    328      15110     15100    0.1    63    33
  6.00    77      0    415    363       4695      4684    0.2    48    34
  7.00    85      1    417    433       5270      5251    0.4    49    35
  7.00    67      1    443    228       2242      2232    0.5    85    44
```

tcpservicesコマンドは、選択されたプロセスのリスニングポートに関する情報を表示します。各プロセスの説明と、アドレスおよびポートの組み合わせも表示されます。

```
SWA_CLI> tcpservices

System Processes (Note: All processes may not always be present)
  ftpd.main    - The FTP daemon
  ginetd       - The INET daemon
  interface    - The interface controller for inter-process communication
```

```
  ipfw         - The IP firewall
  slapd        - The Standalone LDAP daemon
  sntpd        - The SNTP daemon
  sshd         - The SSH daemon
  syslogd      - The system logging daemon
  winbindd     - The Samba Name Service Switch daemon

Feature Processes
  coeuslogd    - Main WSA controller
  gui          - GUI process
  hermes       - Mail server for sending alerts, etc.
  java         - Processes for storing and querying Web Tracking data
  musd         - AnyConnect Secure Mobility server
  pacd         - PAC file hosting daemon
  prox         - WSA proxy
  trafmon      - L4 Traffic Monitor
  uds          - User Discovery System (Transparent Auth)
  wccpd        - WCCP daemon

COMMAND      USER         TYPE NODE   NAME
connector    root         IPv4 TCP    127.0.0.1:8823
java         root         IPv6 TCP    [::127.0.0.1]:18081
hybridd      root         IPv4 TCP    127.0.0.1:8833
gui          root         IPv4 TCP    172.16.40.80:8443
ginetd       root         IPv4 TCP    172.16.40.80:ssh
nginx        root         IPv6 TCP    *:4431
nginx        root         IPv4 TCP    127.0.0.1:8843
nginx        nobody       IPv6 TCP    *:4431
nginx        nobody       IPv4 TCP    127.0.0.1:8843
nginx        nobody       IPv6 TCP    *:4431
nginx        nobody       IPv4 TCP    127.0.0.1:8843
api_serve    root         IPv4 TCP    172.16.40.80:6080
api_serve    root         IPv4 TCP    127.0.0.1:60001
api_serve    root         IPv4 TCP    172.16.40.80:6443
chimera      root         IPv4 TCP    127.0.0.1:6380
nectar       root         IPv4 TCP    127.0.0.1:6382
redis-ser    root         IPv4 TCP    127.0.0.1:6383
redis-ser    root         IPv4 TCP    127.0.0.1:6379
prox         root         IPv4 TCP    127.0.0.1:http
prox         root         IPv6 TCP    [::1]:http
prox         root         IPv4 TCP    172.16.11.69:http
prox         root         IPv4 TCP    172.16.11.68:http
prox         root         IPv4 TCP    172.16.11.252:http
prox         root         IPv4 TCP    127.0.0.1:3128
prox         root         IPv6 TCP    [::1]:3128
prox         root         IPv4 TCP    172.16.11.69:3128
prox         root         IPv4 TCP    172.16.11.68:3128
prox         root         IPv4 TCP    172.16.11.252:3128
prox         root         IPv4 TCP    127.0.0.1:https
prox         root         IPv6 TCP    [::1]:https
prox         root         IPv4 TCP    172.16.11.69:https
prox         root         IPv4 TCP    172.16.11.68:https
prox         root         IPv4 TCP    172.16.11.252:https
prox         root         IPv4 TCP    127.0.0.1:http
prox         root         IPv6 TCP    [::1]:http
prox         root         IPv4 TCP    172.16.11.69:http
prox         root         IPv4 TCP    172.16.11.68:http
prox         root         IPv4 TCP    172.16.11.252:http
prox         root         IPv4 TCP    127.0.0.1:3128
prox         root         IPv6 TCP    [::1]:3128
prox         root         IPv4 TCP    172.16.11.69:3128
prox         root         IPv4 TCP    172.16.11.68:3128
```

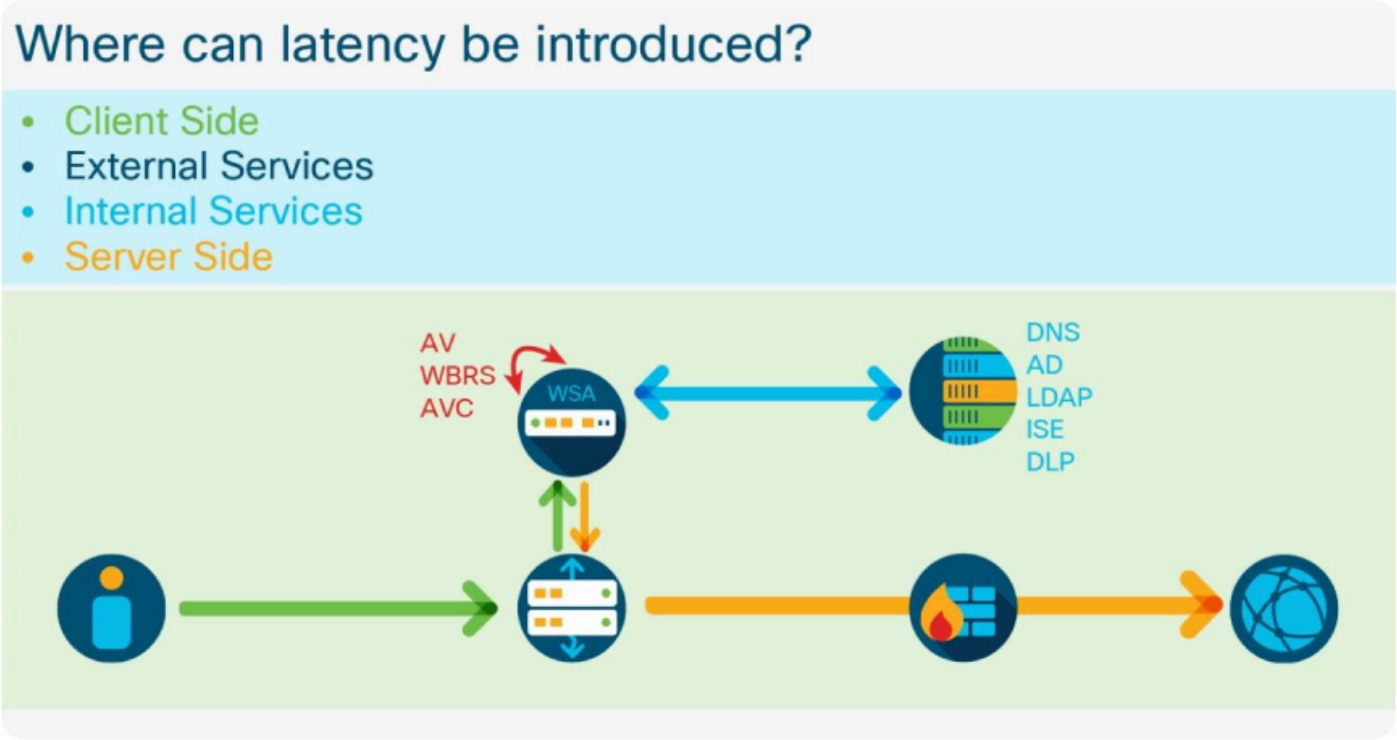
```
prox         root         IPv4 TCP    172.16.11.252:3128
prox         root         IPv4 TCP    127.0.0.1:https
prox         root         IPv6 TCP    [::1]:https
prox         root         IPv4 TCP    172.16.11.69:https
prox         root         IPv4 TCP    172.16.11.68:https
prox         root         IPv4 TCP    172.16.11.252:https
prox         root         IPv4 TCP    127.0.0.1:25255
prox         root         IPv4 TCP    127.0.0.1:socks
prox         root         IPv6 TCP    [::1]:socks
prox         root         IPv4 TCP    172.16.11.69:socks
prox         root         IPv4 TCP    172.16.11.68:socks
prox         root         IPv4 TCP    172.16.11.252:socks
prox         root         IPv4 TCP    127.0.0.1:ftp-proxy
prox         root         IPv6 TCP    [::1]:ftp-proxy
prox         root         IPv4 TCP    172.16.11.69:ftp-proxy
prox         root         IPv4 TCP    172.16.11.68:ftp-proxy
prox         root         IPv4 TCP    172.16.11.252:ftp-proxy
prox         root         IPv4 TCP    127.0.0.1:http
prox         root         IPv6 TCP    [::1]:http
prox         root         IPv4 TCP    172.16.11.69:http
prox         root         IPv4 TCP    172.16.11.68:http
prox         root         IPv4 TCP    172.16.11.252:http
prox         root         IPv4 TCP    127.0.0.1:3128
prox         root         IPv6 TCP    [::1]:3128
prox         root         IPv4 TCP    172.16.11.69:3128
prox         root         IPv4 TCP    172.16.11.68:3128
prox         root         IPv4 TCP    172.16.11.252:3128
prox         root         IPv4 TCP    127.0.0.1:https
prox         root         IPv6 TCP    [::1]:https
prox         root         IPv4 TCP    172.16.11.69:https
prox         root         IPv4 TCP    172.16.11.68:https
prox         root         IPv4 TCP    172.16.11.252:https
prox         root         IPv4 TCP    127.0.0.1:25256
prox         root         IPv4 TCP    127.0.0.1:http
prox         root         IPv6 TCP    [::1]:http
prox         root         IPv4 TCP    172.16.11.69:http
prox         root         IPv4 TCP    172.16.11.68:http
prox         root         IPv4 TCP    172.16.11.252:http
prox         root         IPv4 TCP    127.0.0.1:3128
prox         root         IPv6 TCP    [::1]:3128
prox         root         IPv4 TCP    172.16.11.69:3128
prox         root         IPv4 TCP    172.16.11.68:3128
prox         root         IPv4 TCP    172.16.11.252:3128
prox         root         IPv4 TCP    127.0.0.1:https
prox         root         IPv6 TCP    [::1]:https
prox         root         IPv4 TCP    172.21.11.69:https
prox         root         IPv4 TCP    172.21.11.68:https
prox         root         IPv4 TCP    172.21.11.252:https
prox         root         IPv4 TCP    127.0.0.1:25257
smart_age    root         IPv6 TCP    [::127.0.0.1]:65501
smart_age    root         IPv6 TCP    [::127.0.0.1]:28073
interface    root         IPv4 TCP    127.0.0.1:domain
stunnel      root         IPv4 TCP    127.0.0.1:32137
```

Logging

Webトラフィックは非常に動的で、変化に富んでいます。プロキシの導入が完了したら、アプライアンスを通過するトラフィックの量と構成を定期的に再評価することが重要です。復号化され

たトラフィックの割合を定期的（四半期に1回）にチェックして、サイズが初期インストールの予想および仕様と一致していることを確認する必要があります。これは、Advanced Web Security Reporting(AWSR)などのログ管理製品、またはアクセスログを持つ単純なBashまたはPowerShell(SSH)コマンドを使用して実行できます。また、RPSの数も定期的に再評価して、可用性が高くロードバランスされた構成で、アプライアンスがトラフィックのスパイクと可能なフェールオーバーに対処するために十分なオーバーヘッドがあることを確認する必要があります。

track_statsログは5分ごとに追加され、メモリ内のproxプロセスとそのオブジェクトに直接関連する出力の複数のセクションが含まれます。パフォーマンスモニタリングで最も役立つのは、さまざまな要求プロセスの平均遅延を示すセクションです。このセクションには、DNSルックアップ時間、AVエンジンスキャン時間、その他多くの有用なフィールドが含まれます。このログは、GUIやCLIからは設定できず、Secure Copy Protocol(SCP)またはFile Transfer Protocol(FTP)からのみアクセスできます。これは、パフォーマンスのトラブルシューティング時に最も重要なログであるため、頻繁にポーリングする必要があります。



Client side latency

```
Client Time       1.0 ms     15575
Client Time       1.6 ms     185
Client Time       2.5 ms     855
Client Time       4.0 ms     573
Client Time       6.3 ms     180
Client Time      10.0 ms     264
Client Time      15.8 ms     580
Client Time      25.1 ms     924
Client Time      39.8 ms     1330
Client Time      63.1 ms     4936
Client Time     100.0 ms     5278
Client Time     158.5 ms     10
Client Time     251.2 ms     13
Client Time     398.1 ms     0
Client Time     631.0 ms     0
Client Time    1000.0 ms     0
Client Time    1584.9 ms     0
Client Time    2511.9 ms     0
Client Time    3981.1 ms     0
Client Time    6309.6 ms     30328
```

- **"Client Time"** in **track_stats** log.
- The amount of time in milliseconds that the client was waiting for a response.
- May indicate an upstream issues-keep investigating!
- Access logs can show this in custom field **% : 1>**

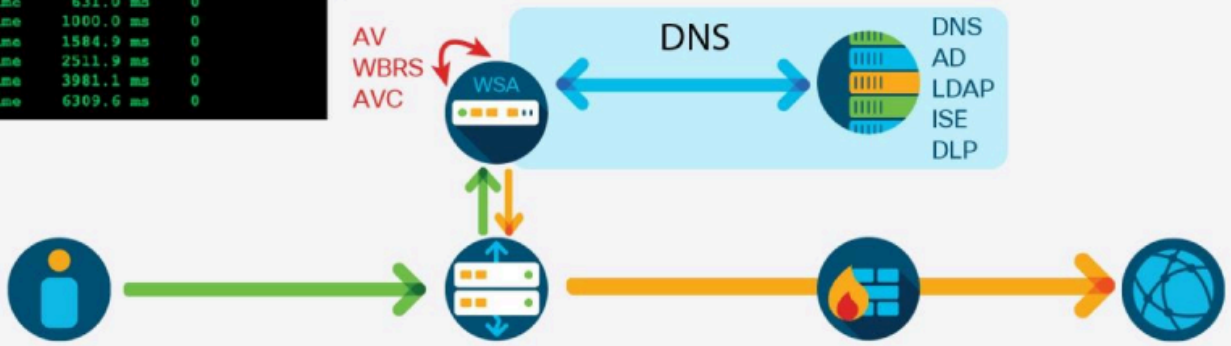
%:1>	x-p2c-first-byte-time	Wait-time for first byte written to client

DNS latency

```
DNS Time       1.0 ms     51
DNS Time       1.6 ms     347
DNS Time       2.5 ms     152
DNS Time       4.0 ms     71
DNS Time       6.3 ms     98
DNS Time      10.0 ms     7
DNS Time      15.8 ms     11
DNS Time      25.1 ms     13
DNS Time      39.8 ms     2
DNS Time      63.1 ms     3
DNS Time     100.0 ms     7
DNS Time     158.5 ms     16
DNS Time     251.2 ms     4
DNS Time     398.1 ms     1
DNS Time     631.0 ms     0
DNS Time    1000.0 ms     0
DNS Time    1584.9 ms     0
DNS Time    2511.9 ms     0
DNS Time    3981.1 ms     0
DNS Time    6309.6 ms     0
```

- The amount of time in milliseconds that the WSA waited for a DNS response.
- Calls for investigation for your DNS resolvers (or path to them).
- **access logs** can show this in custom field **% : >d**

%:>d	x-p2p-dns-svc-time	Time taken by the Web Proxy DNS Process to send a DNS result to ehe Web proxy.

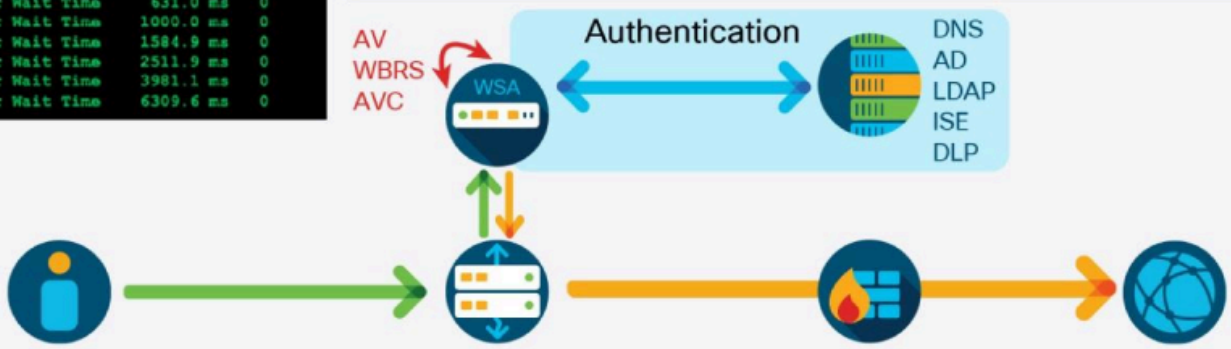


Authentication latency

```
Server Wait Time       1.0 ms   0
Server Wait Time       1.6 ms   0
Server Wait Time       2.5 ms   0
Server Wait Time       4.0 ms   0
Server Wait Time       6.3 ms   0
Server Wait Time      10.0 ms   0
Server Wait Time      15.8 ms   0
Server Wait Time      25.1 ms   0
Server Wait Time      39.8 ms   0
Server Wait Time      63.1 ms   0
Server Wait Time     100.0 ms   0
Server Wait Time     158.5 ms   1
Server Wait Time     251.2 ms   1
Server Wait Time     398.1 ms   0
Server Wait Time     631.0 ms   0
Server Wait Time    1000.0 ms   0
Server Wait Time    1584.9 ms   0
Server Wait Time    2511.9 ms   0
Server Wait Time    3981.1 ms   0
Server Wait Time    6309.6 ms   0
```

- There are two metrics: **"Auth Helper Wait Time"** and **"Auth Helper Service Wait Time."**
- Use the first to get pure auth time without the request time added.
- **access logs** can show this in custom field **% : >a**

%:<a	x-p2p-auth-wait-time	Wait-time to receive the response from the Web Proxy authentication process, after the Web Proxy sent the request.

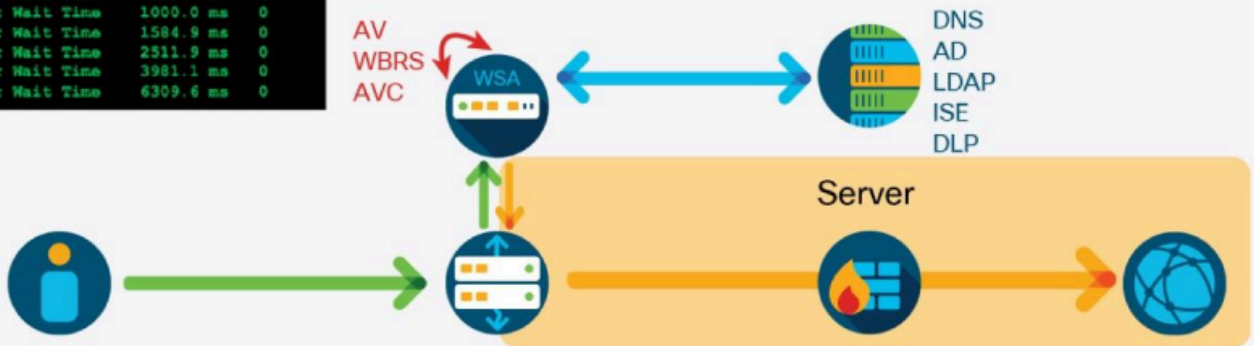


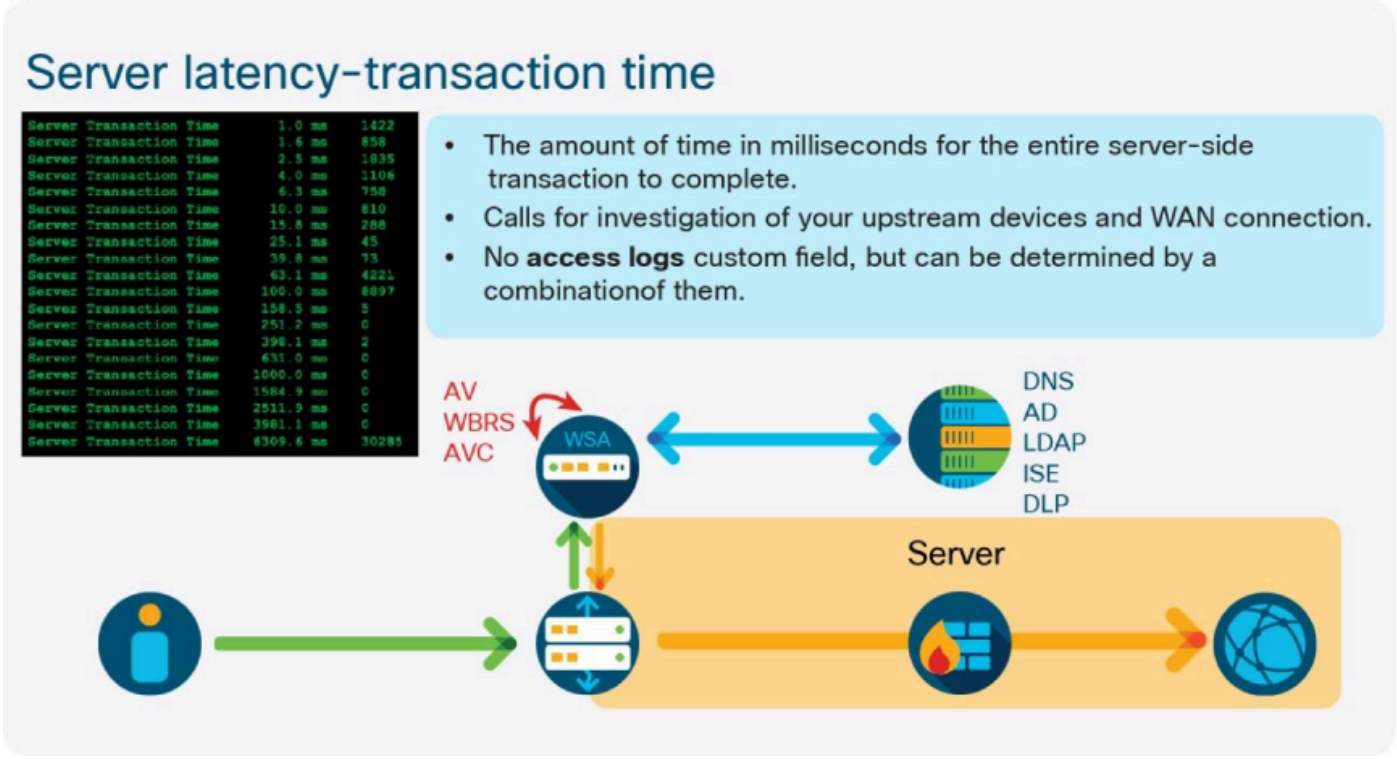
Server latency-wait time

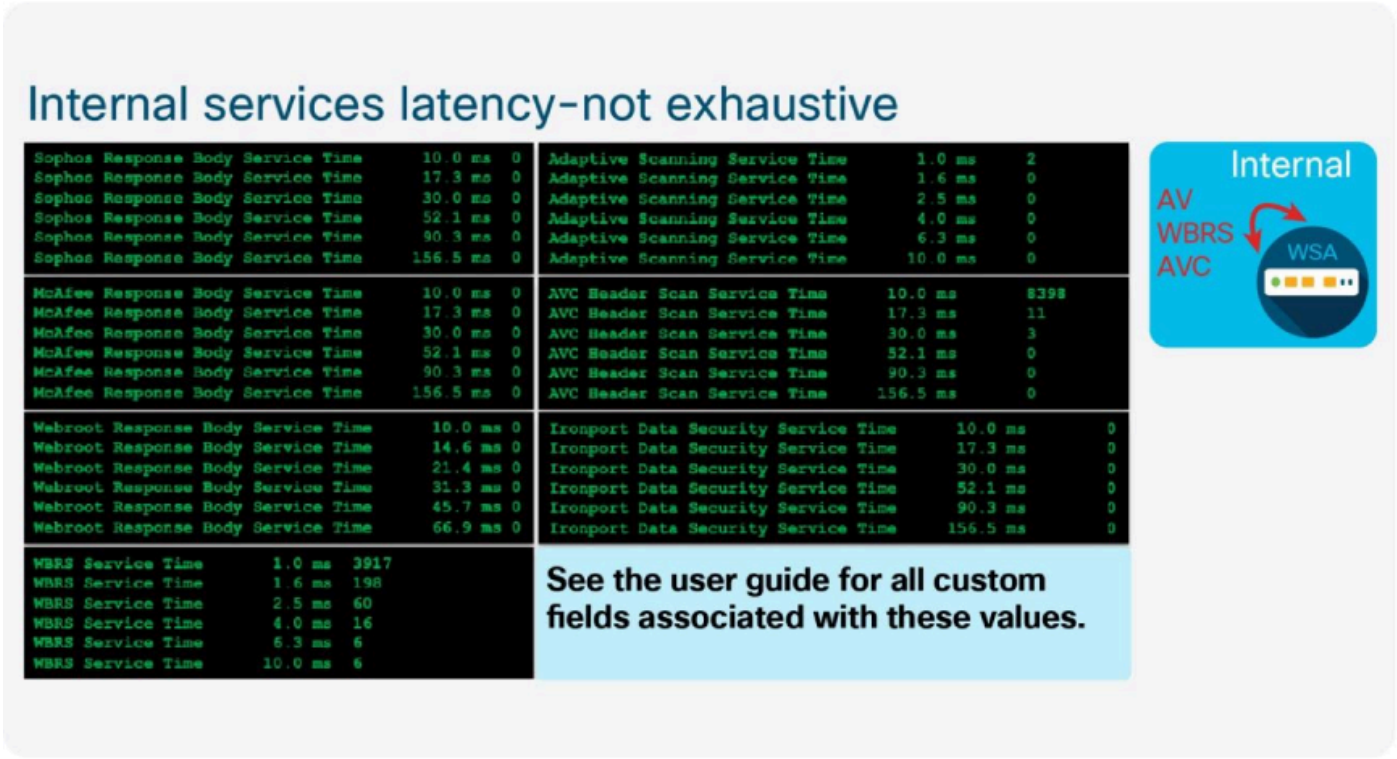
```
Server Wait Time       1.0 ms   0
Server Wait Time       1.6 ms   0
Server Wait Time       2.5 ms   0
Server Wait Time       4.0 ms   0
Server Wait Time       6.3 ms   0
Server Wait Time      10.0 ms   0
Server Wait Time      15.8 ms   0
Server Wait Time      25.1 ms   0
Server Wait Time      39.8 ms   0
Server Wait Time      63.1 ms   0
Server Wait Time     100.0 ms   0
Server Wait Time     158.5 ms   1
Server Wait Time     251.2 ms   1
Server Wait Time     398.1 ms   0
Server Wait Time     631.0 ms   0
Server Wait Time    1000.0 ms   0
Server Wait Time    1584.9 ms   0
Server Wait Time    2511.9 ms   0
Server Wait Time    3981.1 ms   0
Server Wait Time    6309.6 ms   0
```

- The amount of time in milliseconds that the WSA waited for the first byte of the server response.
- Calls for investigation of your upstream devices and WAN connection.
- **access logs** can show this in custom field **% : >1**

%:>1	x-s2p-first-byte-time	Wait-time for first response byte from server







個々のSHDログ行は60秒ごとに書き込まれ、遅延、RPS、クライアント側とサーバ側の接続の合計など、パフォーマンスの監視に重要な多くのフィールドが含まれます。SHDログ行の例を次に示します。

```
Fri Nov 11 14:16:42 2022 Info: Status: CPULd 2.4 DskUtil 45.7 RAMUtil 6.7 Reqs 62 Band 11383 Latency 61
Fri Nov 11 14:17:42 2022 Info: Status: CPULd 2.6 DskUtil 45.7 RAMUtil 6.7 Reqs 55 Band 10532 Latency 77
Fri Nov 11 14:18:43 2022 Info: Status: CPULd 1.9 DskUtil 45.7 RAMUtil 6.6 Reqs 48 Band 7285 Latency 579
Fri Nov 11 14:19:43 2022 Info: Status: CPULd 2.3 DskUtil 45.7 RAMUtil 6.6 Reqs 52 Band 34294 Latency 79
Fri Nov 11 14:20:43 2022 Info: Status: CPULd 2.4 DskUtil 45.7 RAMUtil 6.7 Reqs 55 Band 8696 Latency 691
```

```
Fri Nov 11 14:21:43 2022 Info: Status: CPULd 2.3 DskUtil 45.7 RAMUtil 6.7 Reqs 49 Band 7064 Latency 140
Fri Nov 11 14:22:43 2022 Info: Status: CPULd 1.9 DskUtil 45.7 RAMUtil 6.8 Reqs 41 Band 5444 Latency 788
Fri Nov 11 14:23:43 2022 Info: Status: CPULd 2.2 DskUtil 45.7 RAMUtil 6.8 Reqs 48 Band 6793 Latency 820
Fri Nov 11 14:24:44 2022 Info: Status: CPULd 2.3 DskUtil 45.7 RAMUtil 6.7 Reqs 44 Band 8735 Latency 673
Fri Nov 11 14:25:44 2022 Info: Status: CPULd 2.4 DskUtil 45.7 RAMUtil 6.7 Reqs 53 Band 8338 Latency 731
```

access_logsには、個々のリクエストの遅延情報を示すカスタムフィールドを追加できます。これらのフィールドには、サーバ応答、DNS解決、およびAVスキャナ遅延が含まれます。トラブルシューティングに使用する貴重な情報を収集するには、フィールドをログに追加する必要があります。使用するカスタムフィールドの推奨文字列を次に示します。

```
[ Request Details: ID = %I, User Agent = %u, AD Group Memberships = ( %m ) %g ] [ Tx Wait Times (in ms)
```

```
, Response Header = %:h>, Client Body = %:b> ] [ Rx Wait Times (in ms): 1st request byte = %:1<,
```

```
a; DNS response = %:
```

```
d, WBRS response = %:
```

```
r, AVC response = %:A>, AVC total = %:A<, DCA response = %:C>, DCA total = %:C<, McAfee respon
```

```
s, AMP response = %:e>, AMP total = %:e<; Latency = %x; %L ][Client Port = %F, Server IP = %k
```

これらの値から導き出されるパフォーマンス情報は次のとおりです。

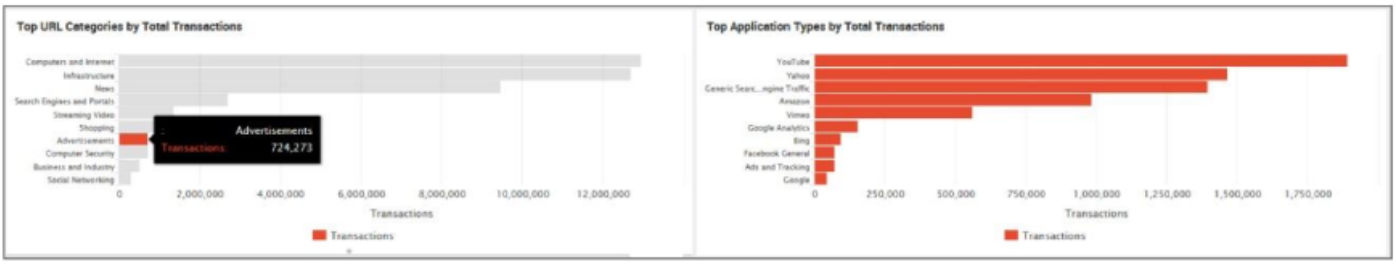
カスタムフィールド	説明
%:<a	Webプロキシが要求を送信した後、Webプロキシ認証プロセスから応答を受信するまでの待機時間。
%:<b	ヘッダーの後に要求本文をサーバーに書き込むまでの待機時間です。
%:<d	Webプロキシが要求を送信した後、WebプロキシDNSプロセスから応答を受信するまでの待機時間。
%:<h	最初のバイトの後に要求ヘッダーをサーバに書き込むまでの待機時間。
%:<r	Webプロキシが要求を送信した後、Webレピュテーションフィルタからの応答を受信するまでの待機時間。
%:<s	Webプロキシが要求を送信した後、Webプロキシのスパイウェア対策プロセスから判定を受け取るまでの待ち時間。
%:>	サーバからの最初の応答バイトの待機時間。
%:>a	Webプロキシ認証プロセスからの応答を受信するまでの待機時間には、Webプロキシが要求を送信するために必要な時間が含まれます。
%:>b	ヘッダー受信後の応答本文の完了までの待機時間です。
%:>c	Webプロキシがディスクキャッシュから応答を読み取るのに必要な時間。
%:>d	WebプロキシDNSプロセスからの応答を受信するまでの待機時間。Webプロキシが要求を送信するために必要な時間が含まれます。
%:>h	最初の応答バイトの後のサーバヘッダーの待機時間。
%:>r	Webレピュテーションフィルタから判定を受信するまでの待機時間には、Webプロキシが要求を送信するのに必要な時間が含まれます。
%:>秒	Webプロキシのスパイウェア対策プロセスから判定を受信するまでの待機時間には、Webプロキシが要求を送信するのに必要な時間が含まれます。
%:1<	新しいクライアント接続からの最初の要求バイトの待機時間。
%:1>	クライアントに書き込まれた最初のバイトの待機時間。
%:b<	クライアント本体の完了までの待機時間です。
%:b>	本体が完全にクライアントに書き込まれるまで待つ時間。
%:e>	Webプロキシが要求を送信した後、AMPスキャンエンジンから応答を受信するまでの待機時間。
%:e<	AMPスキャンエンジンから判定を受信するまでの待機時間には、Webプロキシが要求を送信するために必要な時間が含まれます。
%:h<	最初のバイトの後に完全なクライアントヘッダーを待つ時間。

%:h>	ヘッダー全体がクライアントに書き込まれるまでの待機時間。
%:m<	McAfeeスキャンエンジンから判定を受信するまでの待機時間には、Webプロキシが要求を送信するために必要な時間が含まれます。
%:m>	Webプロキシが要求を送信した後、McAfeeスキャンエンジンから応答を受信するまでの待機時間。
%F	クライアント送信元ポート。
%p	Webサーバポート。
%k	データソースIPアドレス（Webサーバ IPアドレス）。
%:w<	Webrootスキャンエンジンから判定を受信するまでの待機時間には、Webプロキシが要求を送信するために必要な時間が含まれます。
%:w>	Webプロキシが要求を送信した後、Webrootスキャンエンジンから応答を受信するまでの待機時間。

SWAライセンスモデルでは、物理アプライアンスのライセンスを仮想アプライアンスに再利用できます。この機能を活用して、ラボ環境で使用するテスト用SWAvアプライアンスを導入できます。新しい機能と構成をこの方法で試験的に導入することで、ライセンス条項に違反することなく、また同時に安定性と信頼性を確保できます。

高度なWebセキュリティレポート(AWSR)

SWAからのレポートデータを最大限に活用するには、AWSRを活用する必要があります。特に多くのSWAが展開されている環境では、このソリューションは、セキュリティ管理アプライアンス(SMA)での中央集中型レポート機能を利用する場合に比べて何倍もスケーラブルです。また、データに非常に深みのあるカスタムのレポート属性を提供し、データをカスタマイズできます。レポートは、あらゆる組織のニーズに合わせてグループ化およびカスタマイズできます。AWSRのサイジングでは、シスコアドバンスドサービスグループを活用する必要があります。



電子メールアラート

SWAに組み込まれているEメールアラートシステムは、ベースラインのアラートシステムとして活用するのが最適です。すべての情報イベントが有効になっているとノイズが非常に多くなる可能性があるため、管理者のニーズに合わせて適切に調整する必要があります。すべてをアラートし、スパムとして無視するよりも、アラートを制限してアクティブにモニタする方が重要です。

アラート設定	コンフィギュレーション
アラート送信時に使用する送信元アドレス	自動生成
重複したアラートを送信するまでの初期待機時間（秒）	300 seconds
重複したアラートを送信するまでの最大待機時	3600 seconds

間（秒）	

可用性の監視

Webプロキシの可用性を監視するには、2つの方法があります。

1. 1つ目はレイヤ3(L3)モニタリングです。アプライアンスのIPアドレスがネットワーク上で到達可能かどうかをテストします。これをテストする最も簡単な方法は、ICMPエコー(ping)要求を一定の間隔でアドレスに送信し、応答パケットを確認することです。TTLや遅延などの応答の属性を解析して、ネットワーク層の状態を判別できます。
2. デバイスがpingに応答できる一方で、プロキシプロセスが応答しない、または断続的になる可能性があります。このため、レイヤ7(L7)モニタを使用することをお勧めします。このモニタは明示的なプロキシ要求をアプライアンスに送信し、200 OK HTTP応答コードを待ちます。このテストでは、ネットワークインターフェイスの到達可能性だけでなく、プロキシサービスの応答性、および外部リソースが要求された場合のアップストリームサービスの実行可能性もテストされます。このタイプのモニタリングは通常、明示的なHTTP HEAD要求の形式を取り、プロキシにリソースへの接続を要求します。HEADメソッドは、返されるヘッダーを要求するときにクライアントがGET要求を送信する必要がありますが、応答ヘッダーだけが含まれ、データは含まれません。
    - L7モニタリングツールまたはスクリプトを使用する場合、トラフィックが認証から免除されていることを確認することが重要です。そうしないと、認証が定期的に失敗し、リソースが消費されます。モニタリングツールでカスタムユーザエージェント文字列を使用する場合は、トラフィックを識別するために使用する必要があります。トラフィックは認証から除外されますが、アクセスポリシーを通じて不必要なインターネットアクセスから制限される可能性があります。

これらの方法を1つ以上使用する場合、管理者はプロキシ・レスポンスに関する許容可能なメトリックのベースラインを確立し、それを使用してアラートのしきい値を設定する必要があります。このようなチェックの応答を収集し、しきい値とアラートの設定方法を決定する前に時間を割く必要があります。

SNMPモニタリング

簡易ネットワーク管理プロトコル(SNMP)は、アプライアンスの状態を監視するための主要な方法です。この機能を使用して、アプライアンスからアラートを受信したり（トラップ）、さまざまなオブジェクト識別子(OID)をポーリングして情報を収集したりできます。SWAでは、ハードウェアからリソースの使用、個々のプロセス情報、および要求の統計情報に至るまで、あらゆることをカバーする多くのOIDを使用できます。

ハードウェアとパフォーマンスの両方に関連する理由から、監視する必要がある特定のMachine Information Base（MIB；マシン情報ベース）(MIB))が多数あります。MIBの全リストは、https://www.cisco.com/web/ironport/tools/web/asyncosweb-mib.txtから入手できます。

次に示すのは、監視が推奨されるMIBのリストであり、完全なリストではありません。

ハードウェアOID	[名前(Name)]
1.3.6.1.4.1.15497.1.1.1.18.1.3	raidID

1.3.6.1.4.1.15497.1.1.1.18.1.2	raidステータス
1.3.6.1.4.1.15497.1.1.1.18.1.4	raidLastError
1.3.6.1.4.1.15497.1.1.1.10	ファンテーブル
1.3.6.1.4.1.15497.1.1.1.9.1.2	摂氏

次に、status detail CLIコマンドの出力に直接マッピングされたOIDを示します。

OID	[名前(Name)]	Status Detailフィールド
System Resources		
1.3.6.1.4.1.15497.1.1.1.2.0	perCentCPUtilization	CPU
1.3.6.1.4.1.15497.1.1.1.1.0	メモリ使用率	RAM
1秒あたりのトランザクション		
1.3.6.1.4.1.15497.1.2.3.7.1.1.0	cacheThruputNow	最後の1分間の1秒あたりの平均トランザクション数。
1.3.6.1.4.1.15497.1.2.3.7.1.2.0	cacheThruput1hrPeak	過去1時間の1秒あたりの最大トランザクション数。
1.3.6.1.4.1.15497.1.2.3.7.1.3.0	cacheThruput1hrMean（平均）	過去1時間の1秒あたりの平均トランザクション数。
1.3.6.1.4.1.15497.1.2.3.7.1.8.0	cacheThruputLifePeak	プロキシ再起動以降の1秒あたりの最大トランザクション数。
1.3.6.1.4.1.15497.1.2.3.7.1.9.0	cacheThruPutLifeMean	プロキシ再起動以降の1秒あたりの平均トランザクション数。
帯域幅		
1.3.6.1.4.1.15497.1.2.3.7.4.1.0	cacheBwidthTotalNow	過去1分間の平均帯域幅。
1.3.6.1.4.1.15497.1.2.3.7.4.2.0	cacheBwidthTotal1hrPeak	過去1時間の最大帯域幅。
1.3.6.1.4.1.15497.1.2.3.7.4.3.0	cacheBwidthTotal1hrMean	過去1時間の平均帯域幅。
1.3.6.1.4.1.15497.1.2.3.7.4.8.0	cacheBwidthTotalLifePeak	プロキシ再起動後の最大帯域幅。
1.3.6.1.4.1.15497.1.2.3.7.4.9.0	cacheBwidthTotalLifeMean	プロキシ再起動以降の平均帯域幅。
応答所要時間		
1.3.6.1.4.1.15497.1.2.3.7.9.1.0	cacheHitsNow	過去1分間の平均キャッシュヒット率。
1.3.6.1.4.1.15497.1.2.3.7.9.2.0	キャッシュヒット1時間ピーク	過去1時間の最大キャッシュヒット率。
1.3.6.1.4.1.15497.1.2.3.7.9.3.0	cacheHits1hrMean	過去1時間の平均キャッシュヒット率。
1.3.6.1.4.1.15497.1.2.3.7.9.8.0	cacheHitsLifePeak	プロキシ再起動後の最大キャッシュヒット率。

1.3.6.1.4.1.15497.1.2.3.7.9.9.0	cacheHitsLifeMean	プロキシ再起動後の平均キャッシュヒット率。
キャッシュヒット率		
1.3.6.1.4.1.15497.1.2.3.7.5.1.0	cacheHitsNow	過去1分間の平均キャッシュヒット率。
1.3.6.1.4.1.15497.1.2.3.7.5.2.0	キャッシュヒット1時間ピーク	過去1時間の最大キャッシュヒット率。
1.3.6.1.4.1.15497.1.2.3.7.5.3.0	cacheHits1hrMean	過去1時間の平均キャッシュヒット率。
1.3.6.1.4.1.15497.1.2.3.7.5.8.0	cacheHitsLifePeak	プロキシ再起動後の最大キャッシュヒット率。
1.3.6.1.4.1.15497.1.2.3.7.5.9.0	cacheHitsLifeMean	プロキシ再起動後の平均キャッシュヒット率。
接続		
1.3.6.1.4.1.15497.1.2.3.2.7.0	cacheClientIdleConns	アイドル状態のクライアント接続。
1.3.6.1.4.1.15497.1.2.3.3.7.0	cacheServerIdleConns	アイドル状態のサーバ接続
1.3.6.1.4.1.15497.1.2.3.2.8.0	cacheClientTotal接続	クライアント接続の合計。
1.3.6.1.4.1.15497.1.2.3.3.8.0	cacheServerTotalConns	サーバー接続の合計。

結論

このガイドでは、SWAの設定、導入、およびモニタリングの最も重要な側面について説明します。参考資料として、SWAを最も効果的に使用したい方に有益な情報を提供することを目的としています。ここで説明するベストプラクティスは、セキュリティツールとしてのデバイスの安定性、スケーラビリティ、および有効性にとって重要です。また、関連するリソースが進むにつれて、ネットワーク環境や製品の機能セットの変更を反映するために頻繁に更新する必要があります。

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。