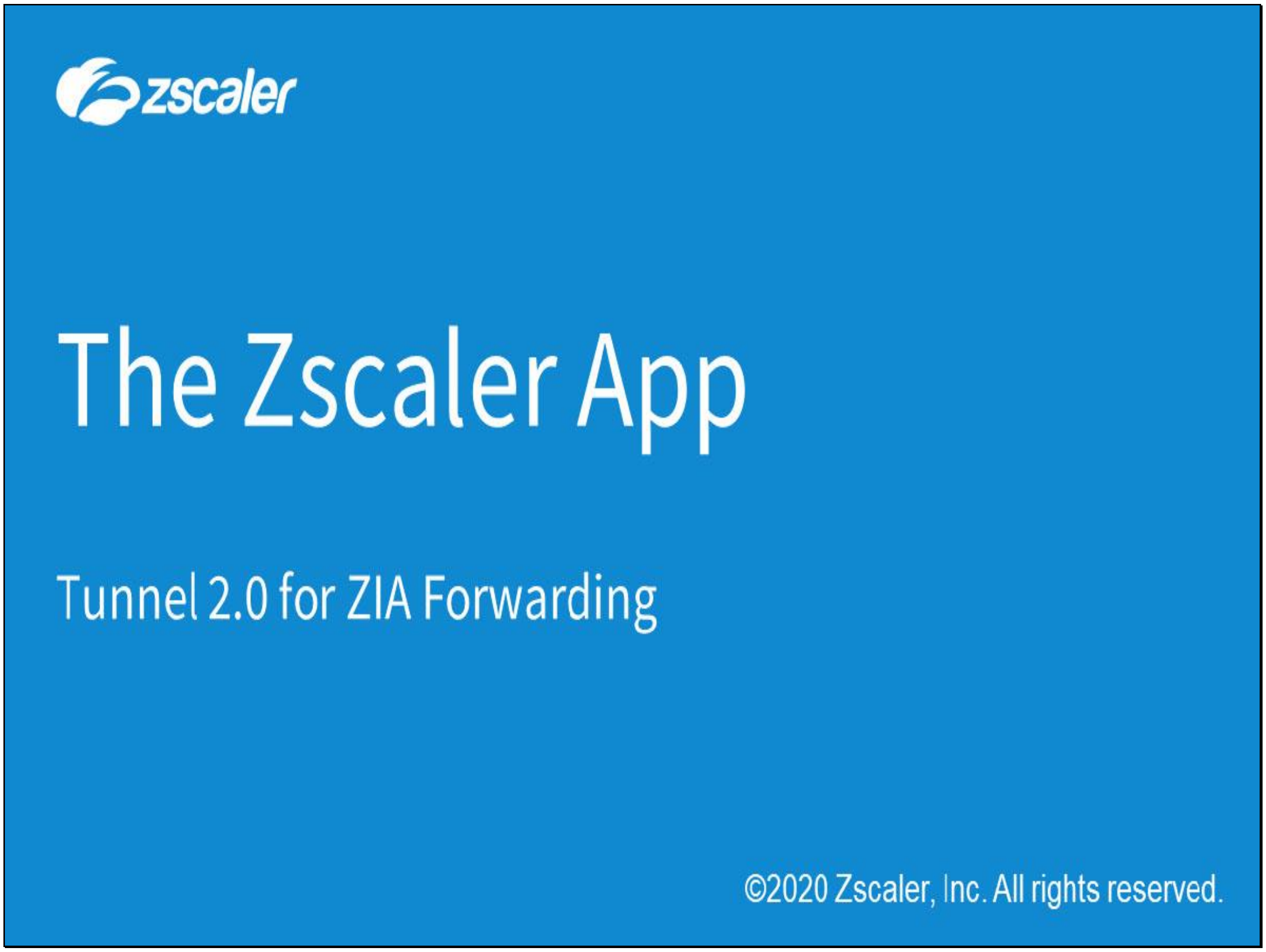
Slide 1 – ZAPP: Tunnel 2.0 for ZIA Forwarding



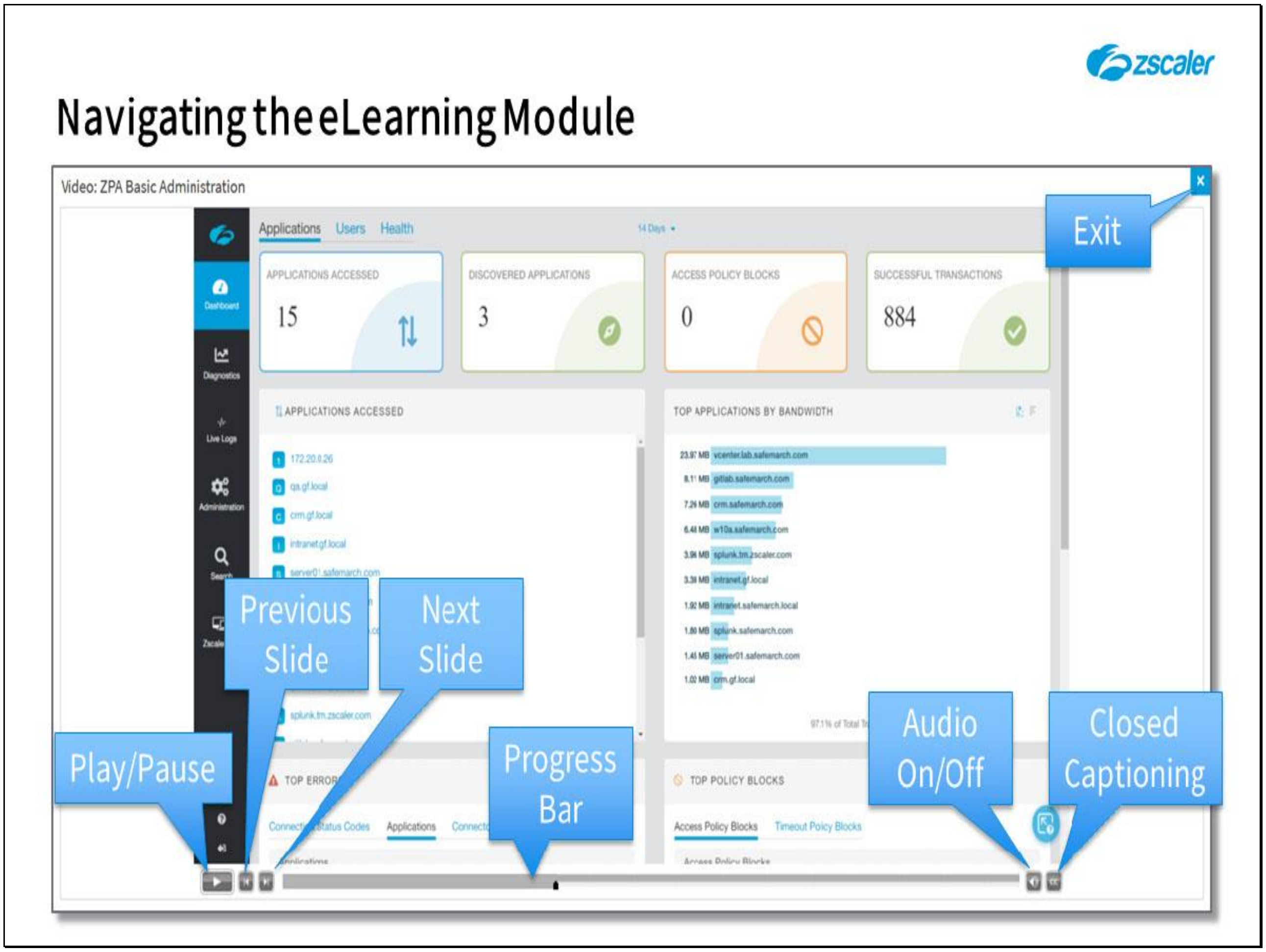
# The Zscaler App

## Tunnel 2.0 for ZIA Forwarding

Slide notes

Welcome to this training module for a detailed look at the Zscaler App **Tunnel 2.0** forwarding method for sending traffic into the ZIA cloud service.

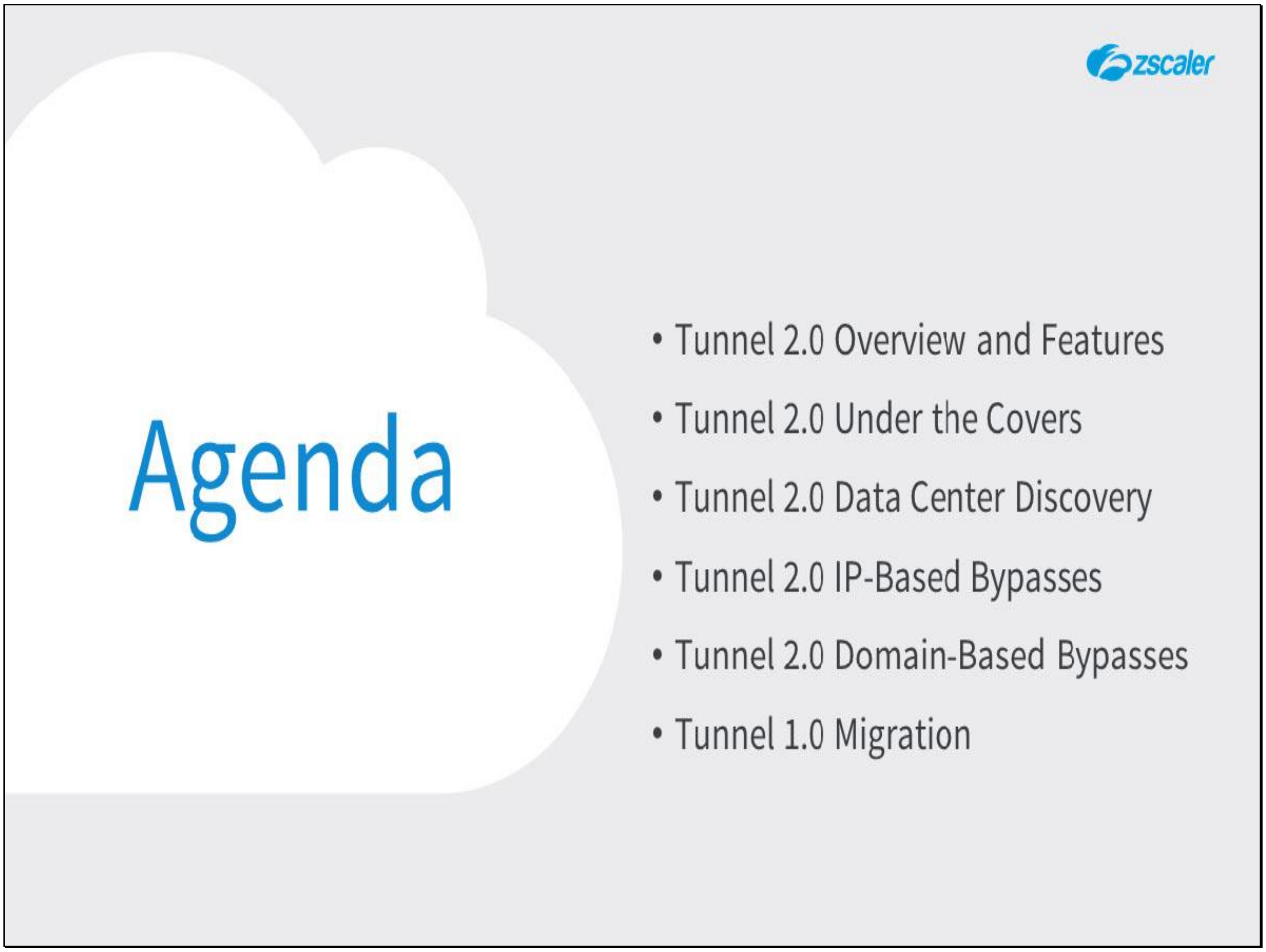
## Slide 2 - Navigating the eLearning Module



### Slide notes

Here is a quick guide to navigating this module. There are various controls for playback including **Play** and **Pause**, **Previous** and **Next** slide. You can also **Mute** the audio or enable **Closed Captioning** which will cause a transcript of the module to be displayed on the screen. Finally, you can click the **X** button at the top to exit.

## Slide 3 - Agenda



zscaler

Agenda

- Tunnel 2.0 Overview and Features
- Tunnel 2.0 Under the Covers
- Tunnel 2.0 Data Center Discovery
- Tunnel 2.0 IP-Based Bypasses
- Tunnel 2.0 Domain-Based Bypasses
- Tunnel 1.0 Migration

### Slide notes

In this module, we will cover the following topics:

- An overview of the **Tunnel 2.0** forwarding method and its key features;
- A detailed look 'under the covers' at how the **Tunnel 2.0** method works;
- A look at how Zscaler App finds **Tunnel 2.0** compliant data centers to connect to;
- A look at how to configure IP-Layer **Tunnel 2.0** bypasses;
- Also at how to bypass Domains in a **Tunnel 2.0** environment;
- And finally, at some of the things you need to consider when migrating from **Tunnel 1.0** to the **Tunnel 2.0** method.

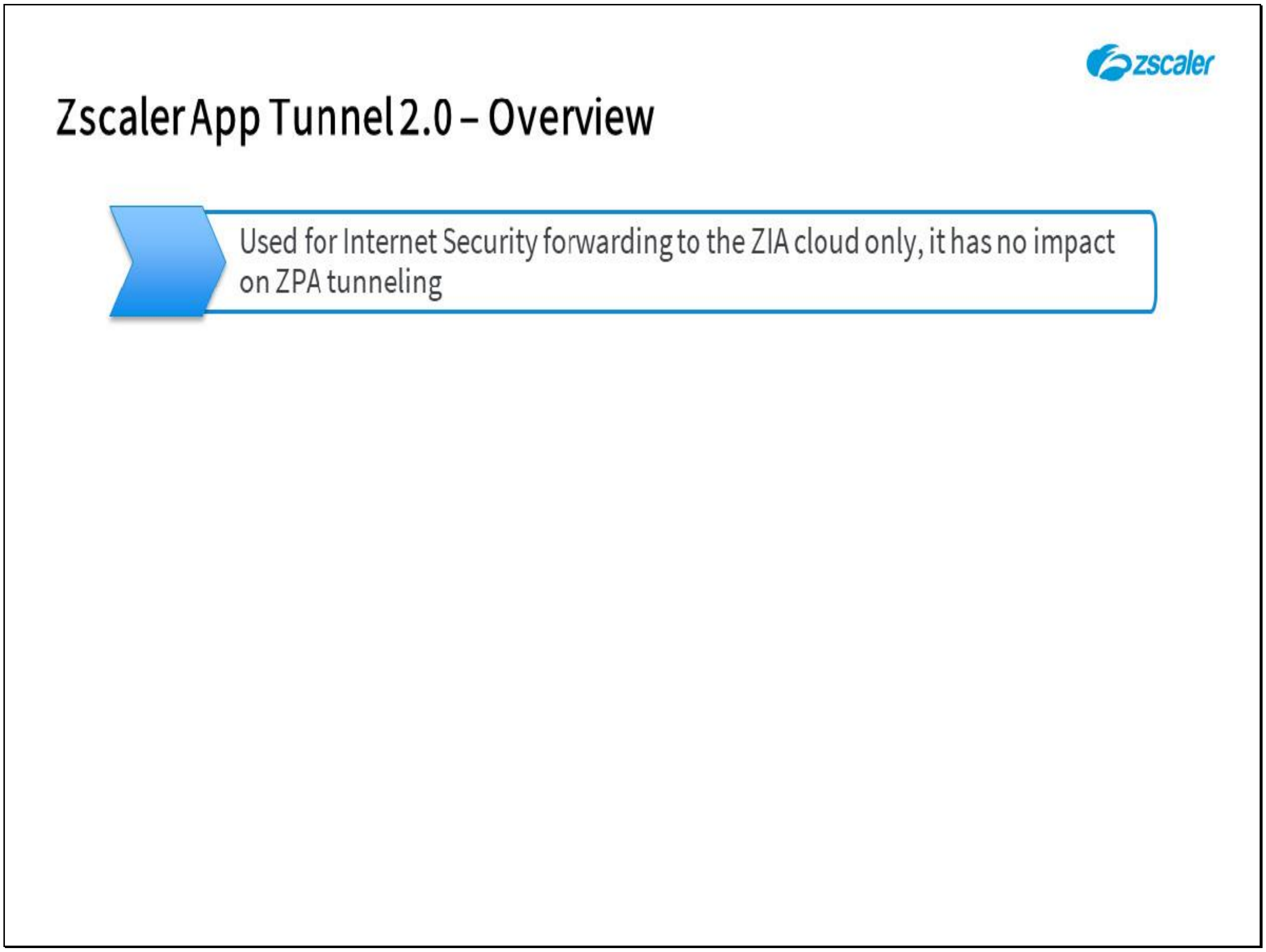
### Slide 4 - Tunnel 2.0 Overview and Features

Tunnel 2.0 Overview and Features

### Slide notes

The first topic that we will cover is an overview of the **Tunnel 2.0** forwarding method and its key features.

Slide 5 - Zscaler App Tunnel 2.0 – Overview



zscaler

## Zscaler App Tunnel 2.0 – Overview

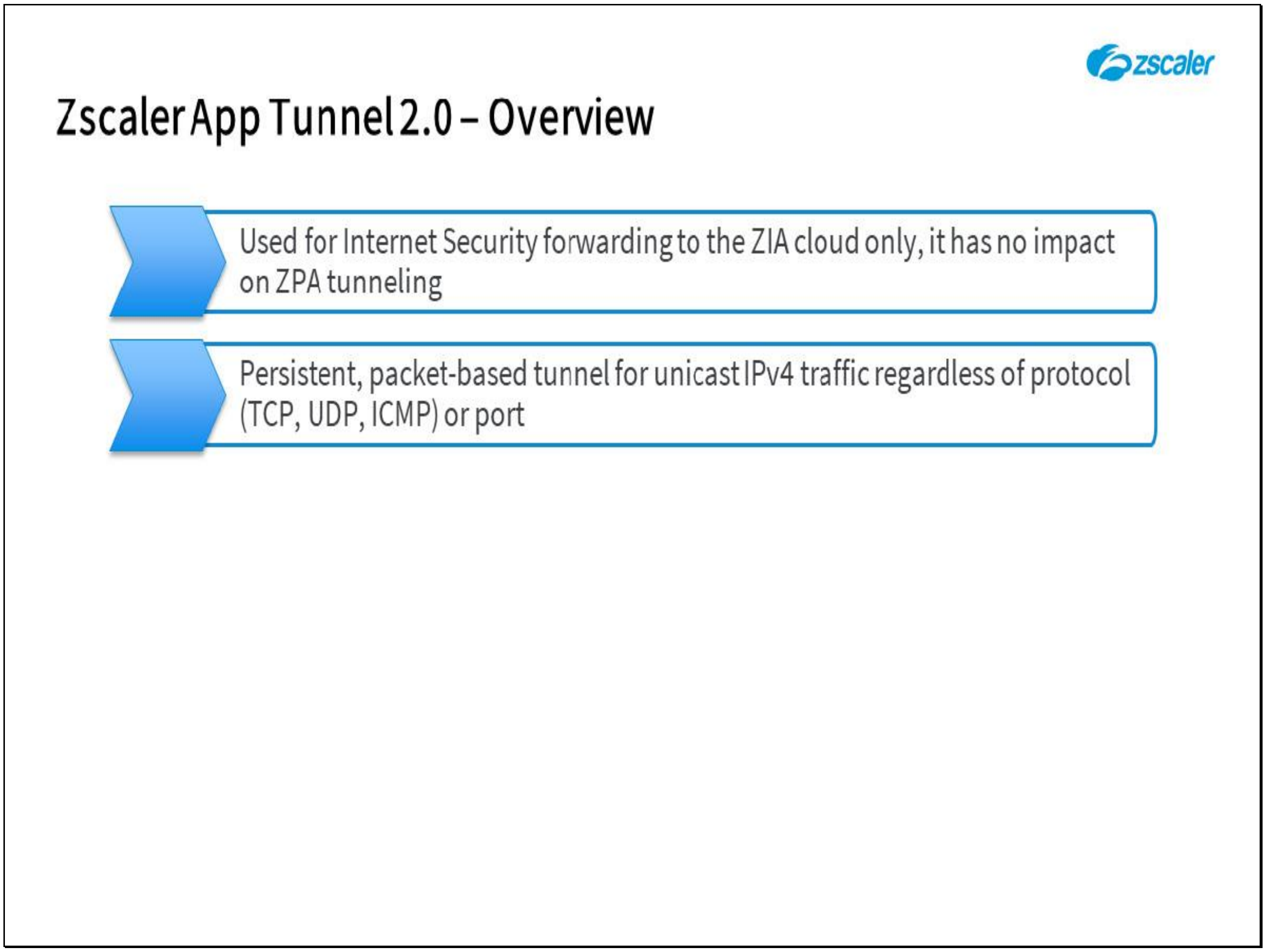
Used for Internet Security forwarding to the ZIA cloud only, it has no impact on ZPA tunneling

### Slide notes

Some of the key features of the **Tunnel 2.0** forwarding method include:

- It is a Zscaler App forwarding method for data to be sent into the Zscaler Internet Access (ZIA) cloud for scanning and policy control to provide improved Internet Security. It has no impact on and can co-exist with ZPA data forwarding to private applications.

Slide 6 - Zscaler App Tunnel 2.0 – Overview



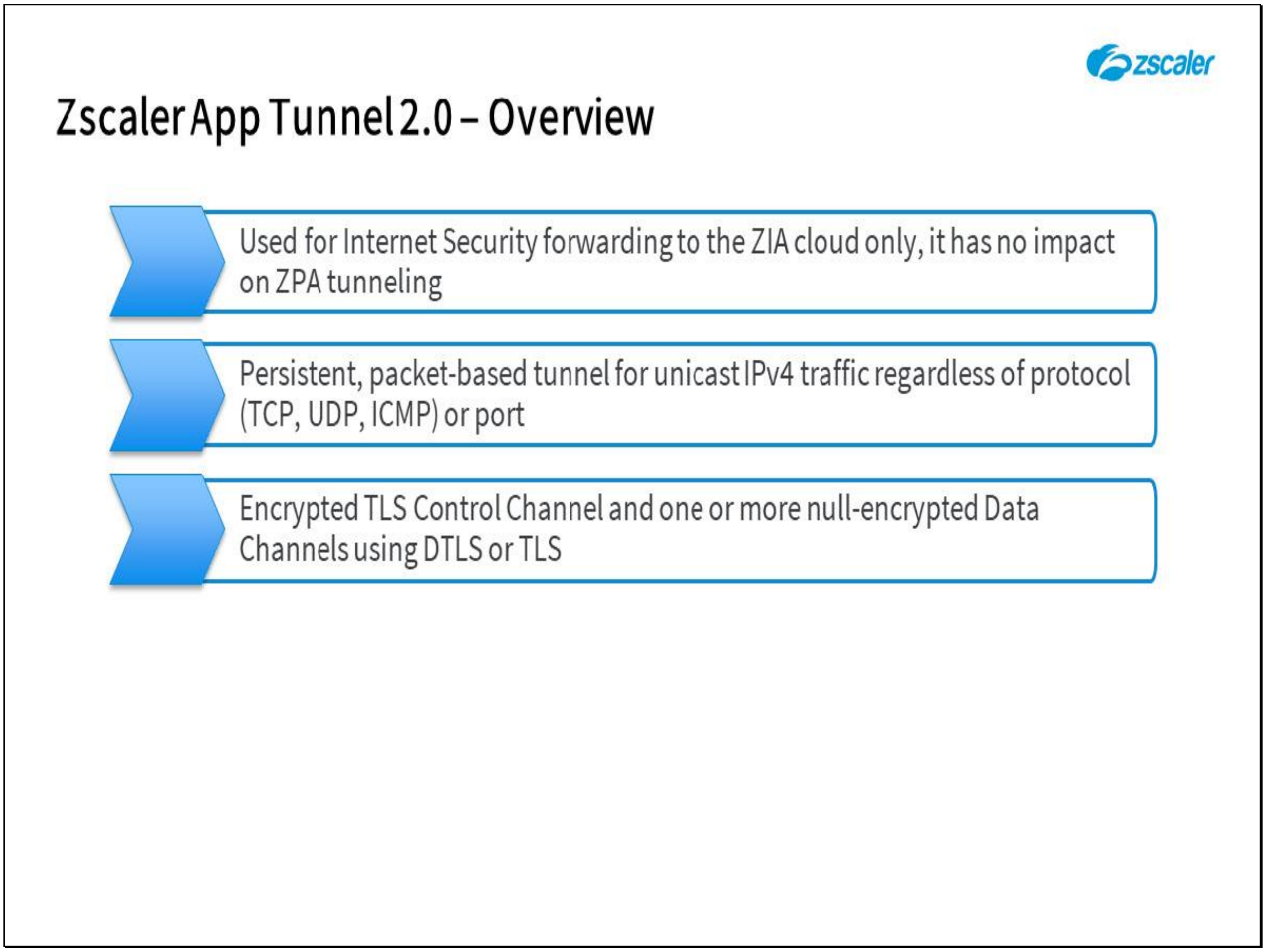
## Zscaler App Tunnel 2.0 – Overview

- Used for Internet Security forwarding to the ZIA cloud only, it has no impact on ZPA tunneling
- Persistent, packet-based tunnel for unicast IPv4 traffic regardless of protocol (TCP, UDP, ICMP) or port

Slide notes

- In contrast to the **Tunnel 1.0** method, which forwards either **TCP** port **80**/**443** traffic in **Tunnel** mode or all **HTTP(S)** traffic (regardless of port) to the loopback port in **Tunnel With Local Proxy** mode, **Tunnel 2.0** uses persistent, packet-based tunnels for **all unicast IPv4 traffic** regardless of protocol (**TCP**, **UDP**, **ICMP**) or port.

Slide 7 - Zscaler App Tunnel 2.0 – Overview



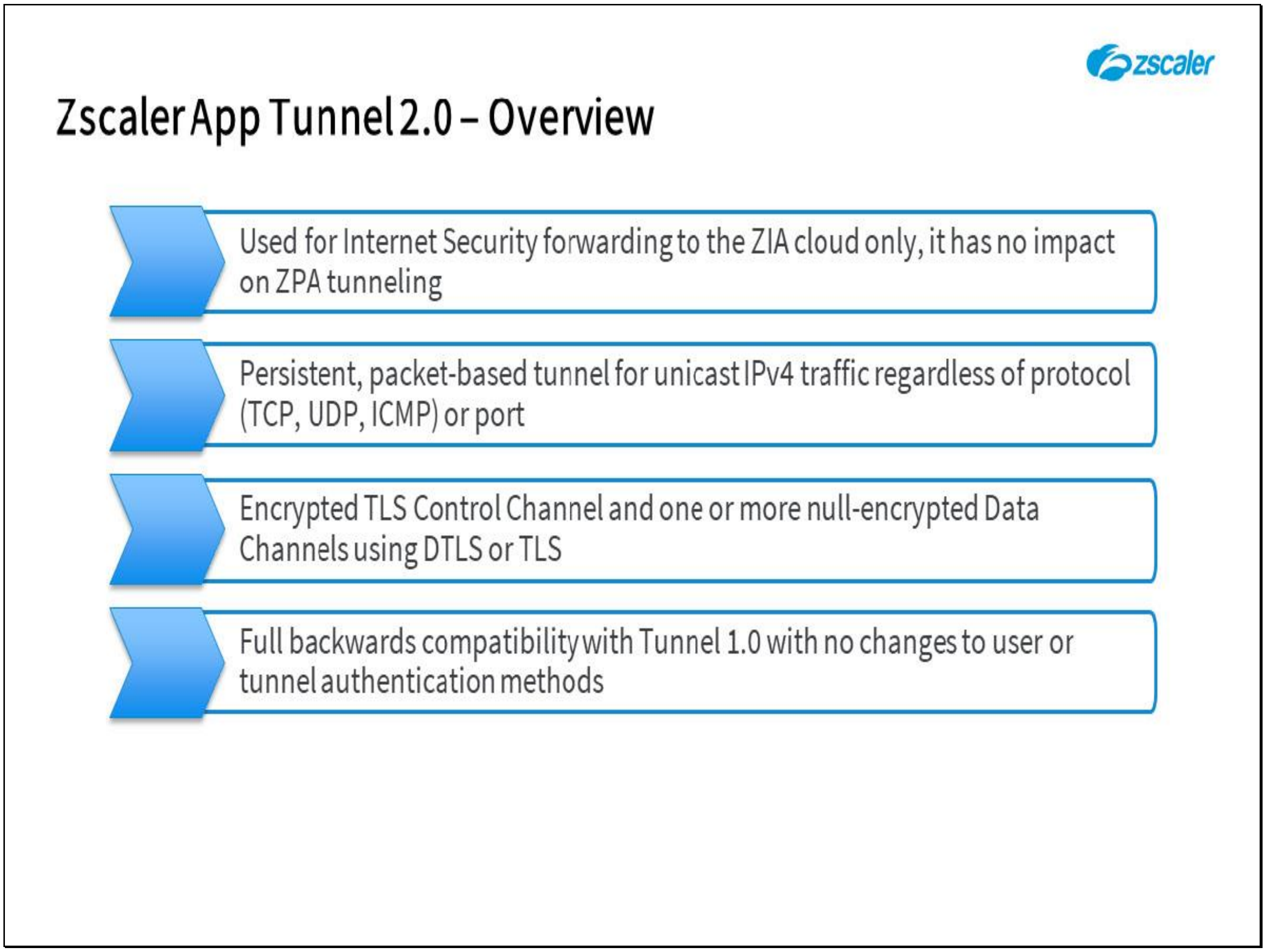
## Zscaler App Tunnel 2.0 – Overview

- Used for Internet Security forwarding to the ZIA cloud only, it has no impact on ZPA tunneling
- Persistent, packet-based tunnel for unicast IPv4 traffic regardless of protocol (TCP, UDP, ICMP) or port
- Encrypted TLS Control Channel and one or more null-encrypted Data Channels using DTLS or TLS

Slide notes

- Once enabled and configured, **Tunnel 2.0** will establish an encrypted **TLS control channel** to the closest (or to a specified) ZEN, for user authentication and device fingerprinting. Data is sent on one or more null-encrypted **DTLS** or **TLS** data channels.

Slide 8 - Zscaler App Tunnel 2.0 – Overview



zscaler

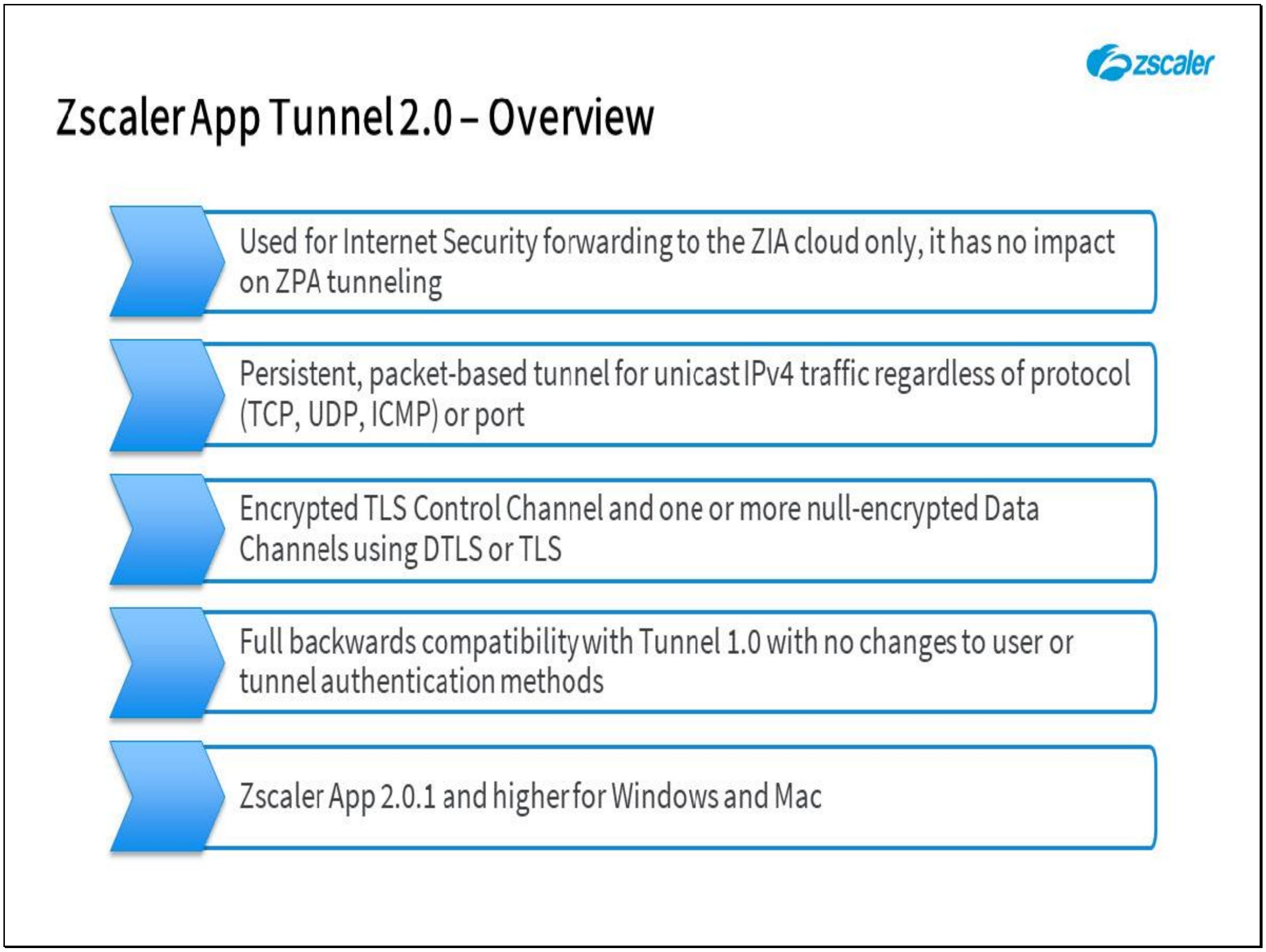
## Zscaler App Tunnel 2.0 – Overview

- Used for Internet Security forwarding to the ZIA cloud only, it has no impact on ZPA tunneling
- Persistent, packet-based tunnel for unicast IPv4 traffic regardless of protocol (TCP, UDP, ICMP) or port
- Encrypted TLS Control Channel and one or more null-encrypted Data Channels using DTLS or TLS
- Full backwards compatibility with Tunnel 1.0 with no changes to user or tunnel authentication methods

Slide notes

- Zscaler App with **Tunnel 2.0** is fully backwardly compatible with the **Tunnel 1.0** method, and it will actually fallback to **Tunnel 1.0** if it is unable to establish or maintain **Tunnel 2.0** connections.

Slide 9 - Zscaler App Tunnel 2.0 – Overview



zscaler

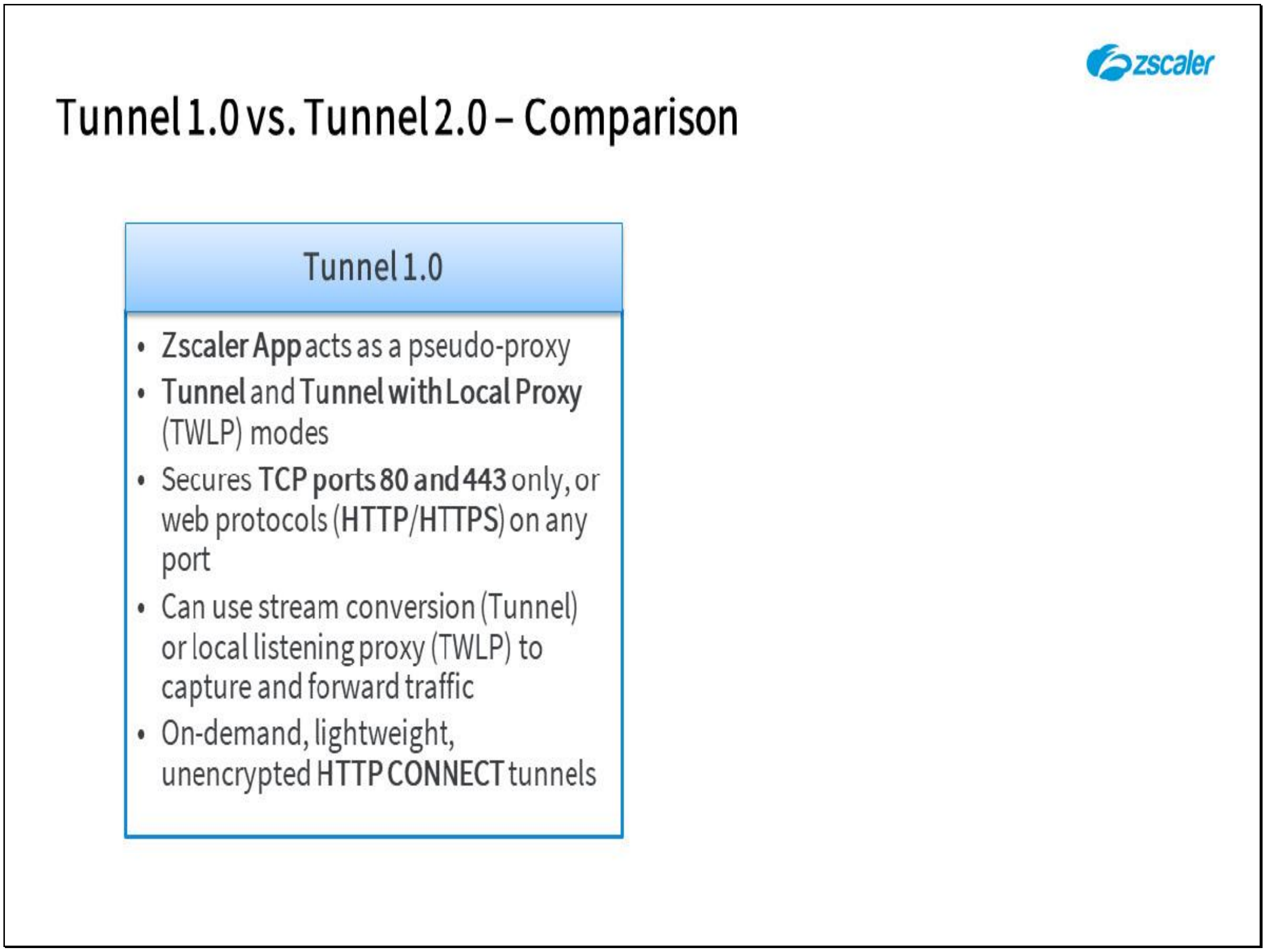
# Zscaler App Tunnel 2.0 – Overview

- Used for Internet Security forwarding to the ZIA cloud only, it has no impact on ZPA tunneling
- Persistent, packet-based tunnel for unicast IPv4 traffic regardless of protocol (TCP, UDP, ICMP) or port
- Encrypted TLS Control Channel and one or more null-encrypted Data Channels using DTLS or TLS
- Full backwards compatibility with Tunnel 1.0 with no changes to user or tunnel authentication methods
- Zscaler App 2.0.1 and higher for Windows and Mac

Slide notes

- **Tunnel 2.0** requires **Zscaler App v2.0.1** or later. It is currently only available for PC platforms (Windows and macOS) and requires the **Packet Filter Based** driver on Windows, plus a **TUN** driver on the macOS platforms (which is installed by Z App).

Slide 10 - Tunnel 1.0 vs. Tunnel 2.0 – Comparison



zscaler

# Tunnel 1.0 vs. Tunnel 2.0 – Comparison

**Tunnel 1.0**

- **Zscaler App** acts as a pseudo-proxy
- **Tunnel** and **Tunnel with Local Proxy** (TWLP) modes
- Secures **TCP ports 80 and 443** only, or web protocols (**HTTP/HTTPS**) on any port
- Can use stream conversion (Tunnel) or local listening proxy (TWLP) to capture and forward traffic
- On-demand, lightweight, unencrypted **HTTP CONNECT** tunnels

## Slide notes

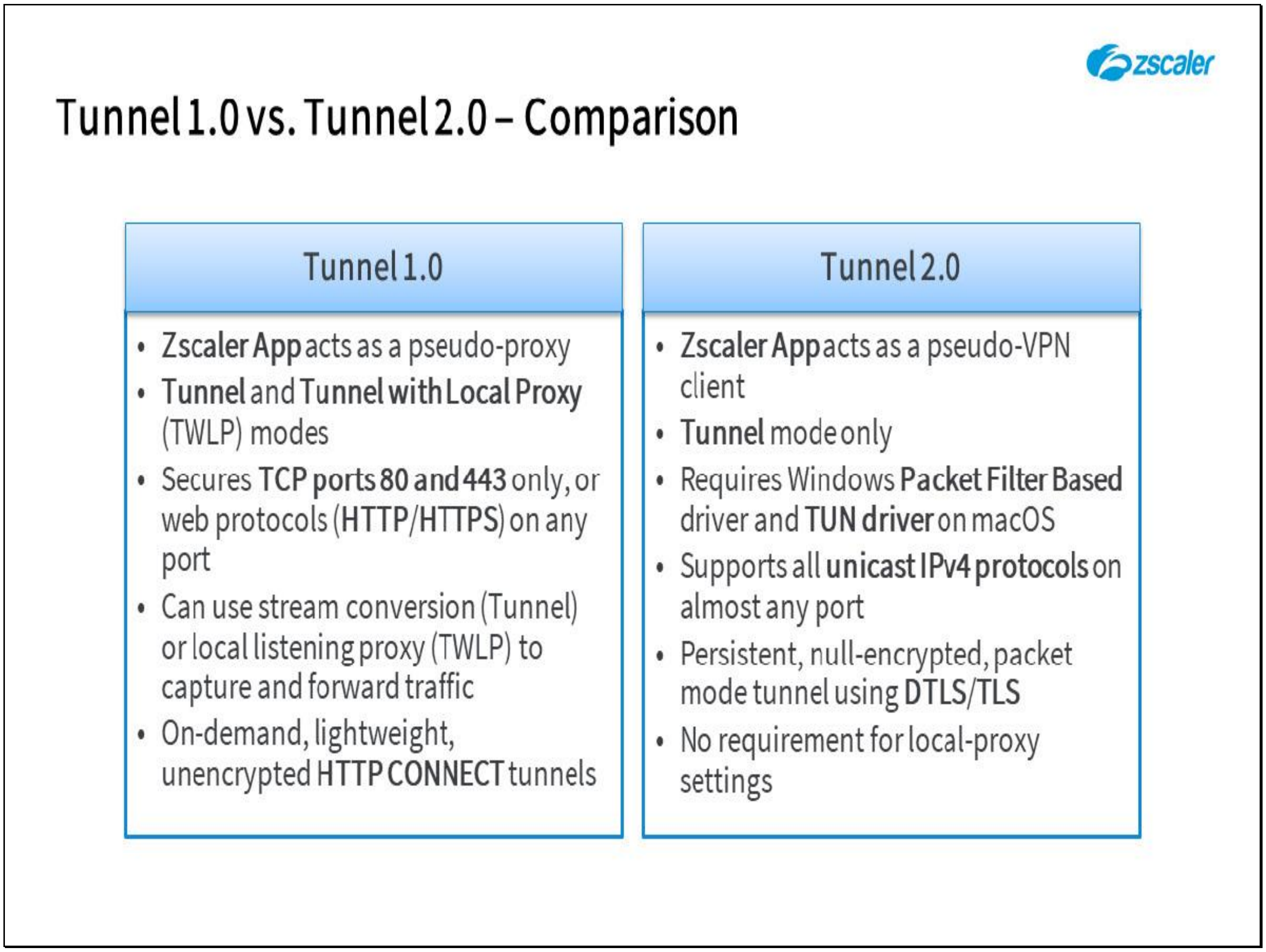
Here is a high-level comparison of the **Tunnel 1.0** forwarding method(s) and **Tunnel 2.0**.

With **Tunnel 1.0**, the Zscaler App acts as a **pseudo-proxy device** for sending traffic into the tunnels to the ZIA service. This method actually supports two forwarding modes, **Tunnel** and **Tunnel with Local Proxy** (**TWLP**):

- The **Tunnel** option forwards **TCP** ports **80**/**443** to Zscaler App for conversion to a Byte stream and forwarding to the ZIA cloud service.
- The **TWLP** option sends web protocols (so **HTTP/HTTPS**) on any port into Z App for tunneling, based on a listening loopback proxy port applied to the system in the **Forwarding Profile** PAC file.

The tunnels used for either type of **Tunnel 1.0** connection are on-demand, lightweight and unencrypted **HTTP CONNECT** tunnels, to either the closest healthy ZENs or to the ZENs specified in the **App Profile** PAC file.

Slide 11 - Tunnel 1.0 vs. Tunnel 2.0 – Comparison



zscaler

## Tunnel 1.0 vs. Tunnel 2.0 – Comparison

### Tunnel 1.0

- **Zscaler App** acts as a pseudo-proxy
- **Tunnel** and **Tunnel with Local Proxy** (TWLP) modes
- Secures **TCP ports 80 and 443** only, or web protocols (**HTTP/HTTPS**) on any port
- Can use stream conversion (Tunnel) or local listening proxy (TWLP) to capture and forward traffic
- On-demand, lightweight, unencrypted **HTTP CONNECT** tunnels

### Tunnel 2.0

- **Zscaler App** acts as a pseudo-VPN client
- **Tunnel** mode only
- Requires Windows **Packet Filter Based** driver and **TUN driver** on macOS
- Supports all **unicast IPv4 protocols** on almost any port
- Persistent, null-encrypted, packet mode tunnel using **DTLS/TLS**
- No requirement for local-proxy settings

Slide notes

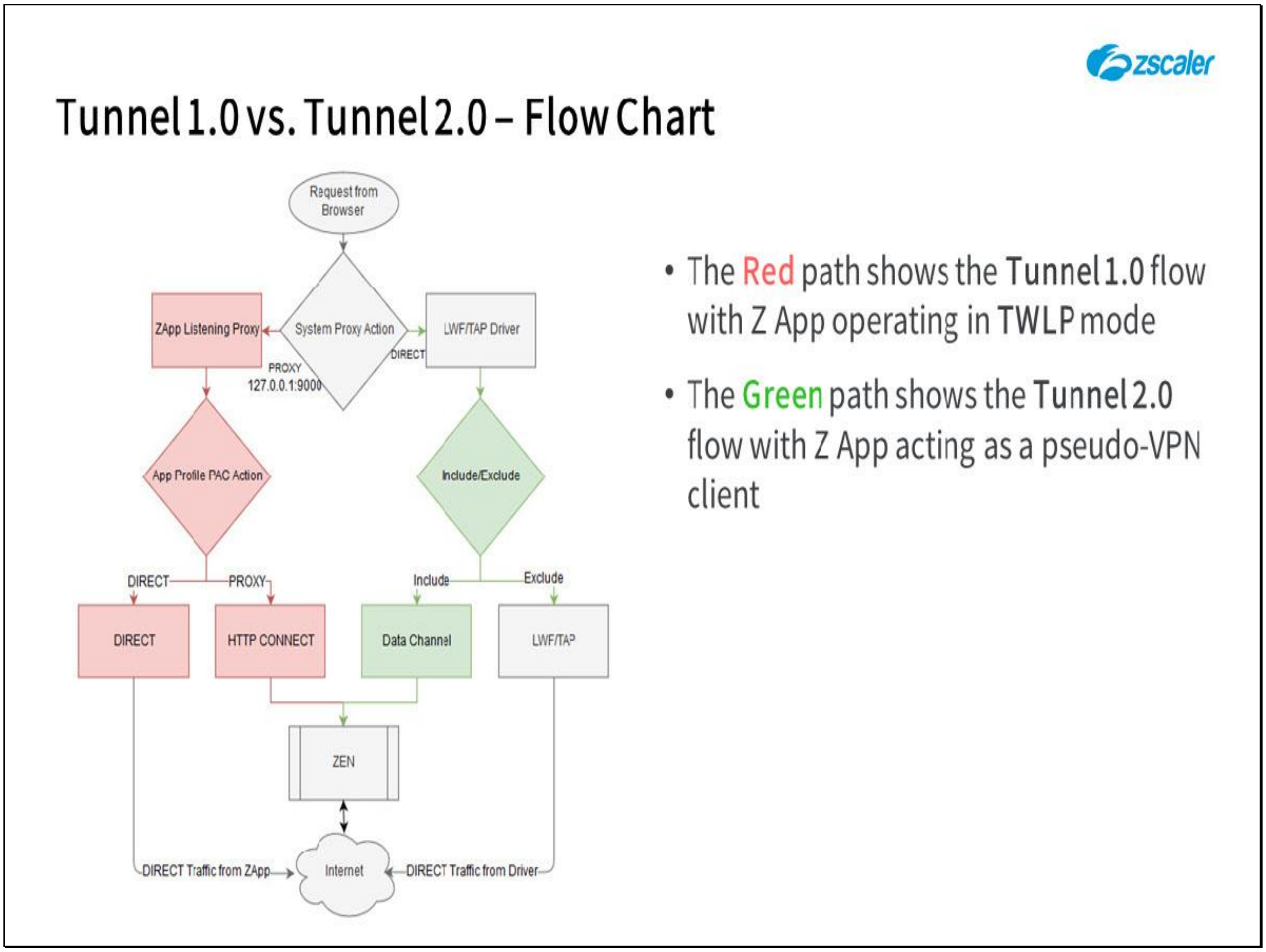
With **Tunnel 2.0** by contrast, Zscaler App acts as a **pseudo-VPN client**. Z App does no transformations, it simply sends packets down a persistent tunnel to the ZIA cloud infrastructure.

**Tunnel** mode only is supported, there is no equivalent to the **TWLP** mode. For the situations where the **TWLP** method is required, you would need to configure Z App to fall back to the **Tunnel 1.0** method.

**Tunnel 2.0** requires the **Packet Filter Based** driver on Windows devices and macOS devices need to have the Z App installed **TUN** driver. The traffic forwarded into the tunnels by default is **all unicast IPv4 traffic** regardless of protocol or port, this allows the ZIA service to provide full policy, cloud firewall and DLP controls over any data sent, in addition to the basic Internet Security service.

The tunnels used to transfer data are persistent, null-encrypted, packet mode tunnels using **DTLS** where possible and falling back to **TLS** where necessary. The **DTLS** and **TLS** tunnels are authenticated and, while they do not provide encryption, they do provide for certificate validation and data integrity. As **Tunnel 2.0** operates at the IP layer (L3), it has no inherent requirement for or understanding of local system proxy settings.

Slide 12 - Tunnel 1.0 vs. Tunnel 2.0 – Flow Chart



zscaler

Tunnel 1.0 vs. Tunnel 2.0 – Flow Chart

- The Red path shows the **Tunnel 1.0** flow with Z App operating in **TWLP** mode
- The Green path shows the **Tunnel 2.0** flow with Z App acting as a pseudo-VPN client

### Slide notes

Here is a flow diagram comparing **Tunnel 1.0** forwarding (in **TWLP** mode) and **Tunnel 2.0**.

The Red path shows the **Tunnel 1.0** flow with Z App in **TWLP** mode:

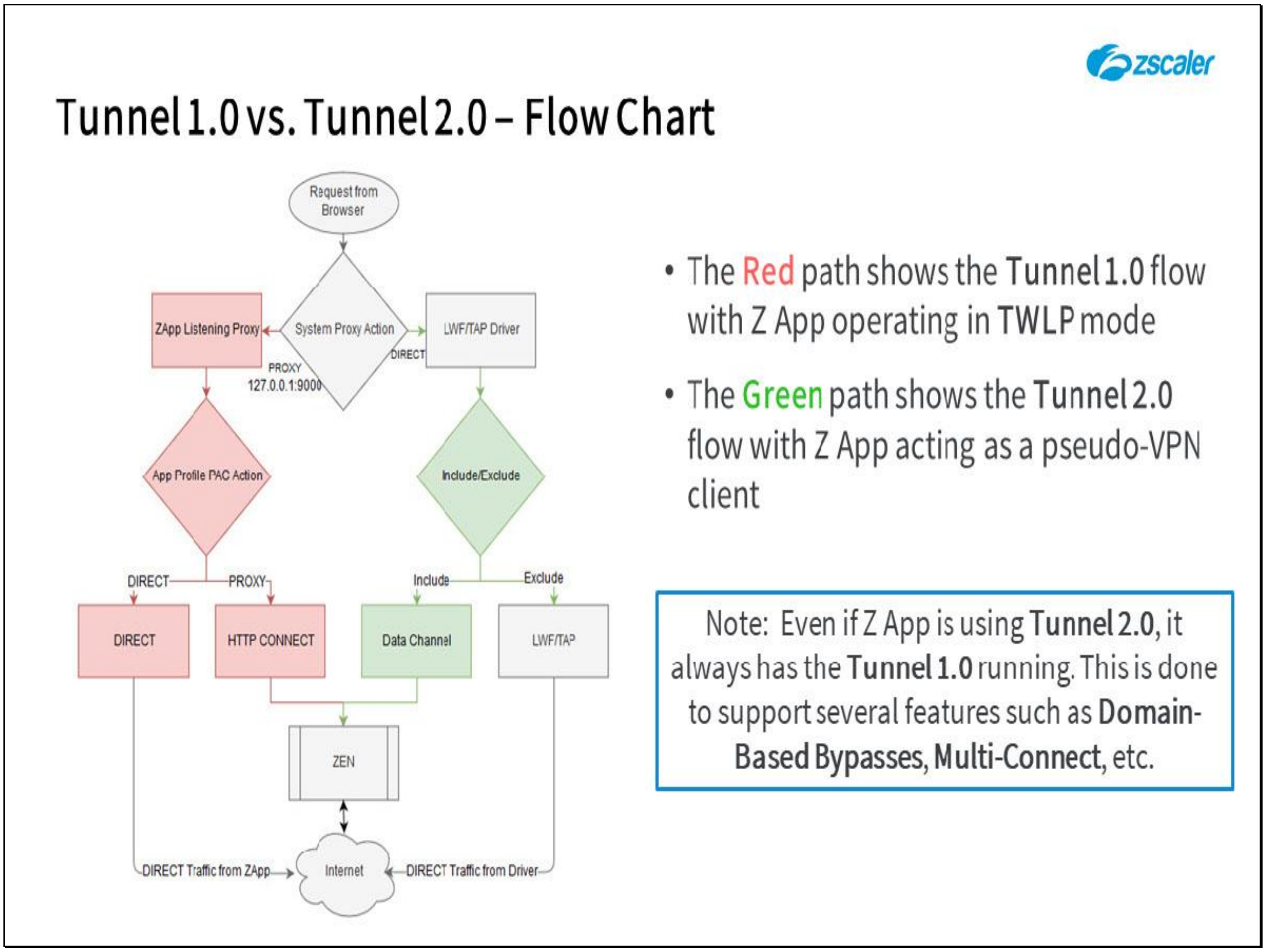
- The system proxy is configured by a **Forwarding Profile** PAC file to send traffic to Zscaler App using a loopback proxy definition on port **9000** (**127.0.0.1:9000** by default);
- The **App Profile** PAC file rules are applied to any traffic received by the App;
- Traffic for destinations specified as **DIRECT** is bypassed by the App and sent straight out to the Internet;
- Traffic for destinations configured to be proxied are forwarded to Zscaler in **lightweight HTTP CONNECT** tunnels for scanning and policy controls.

By contrast, the Green path shows the **Tunnel 2.0** flow with Z App acting as a pseudo-VPN client:

- Any traffic sent **DIRECT** by the system proxy is forwarded into the filter driver for processing;
- The Tunnel 2.0 **Include**/**Exclude** rules then control what traffic is sent into the App for tunneling, a default **0.0.0.0 Inclusion** rule sends all IP traffic into the App, with RFC1918 and Multicast address **Exclusions**.

Note that the **Tunnel 2.0 Include**/**Exclude** rules operate at the IP-layer (L3) and have no visibility into higher layer or application specific addressing (**Domains**, **FQDNs**, or **URLs**). As a result, the **Tunnel 1.0** loopback proxy configuration must be used to configure and manage **Domain-based** exceptions or bypasses for **Domains** or **FQDNs**.

## Slide 13 - Tunnel 1.0 vs. Tunnel 2.0 – Flow Chart



# Tunnel 1.0 vs. Tunnel 2.0 – Flow Chart

- The Red path shows the **Tunnel 1.0** flow with Z App operating in **TWLP** mode
- The Green path shows the **Tunnel 2.0** flow with Z App acting as a pseudo-VPN client

Note: Even if Z App is using **Tunnel 2.0**, it always has the **Tunnel 1.0** running. This is done to support several features such as **Domain-Based Bypasses**, **Multi-Connect**, etc.

### Slide notes

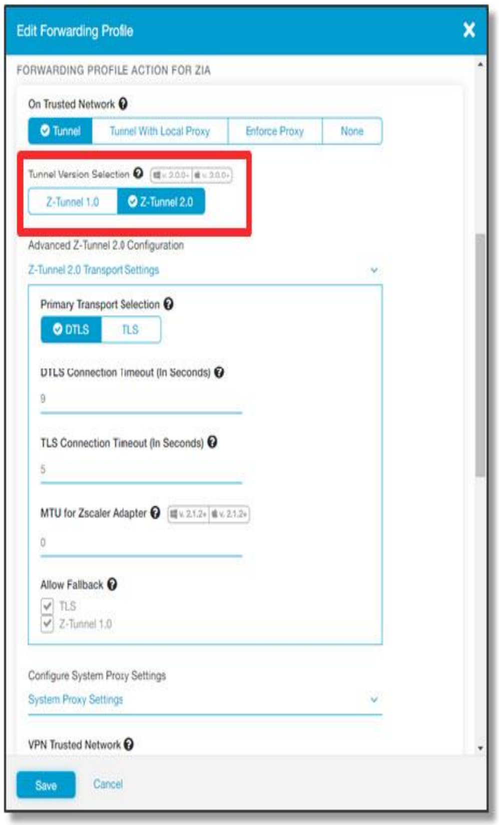
Note that even if Z App is set to use **Tunnel 2.0**, it always has **Tunnel 1.0** running in the background. In fact, under certain circumstances, it may well fall back to the **Tunnel 1.0** method (i.e. **TCP** port **80**/**443** only goes into Z App). This is done for backwards compatibility and to support features such as; **Domain-Based Bypasses**, **Multi-Connect**, etc.

Also note that the fall back to **Tunnel 1.0** can be disabled in the **Forwarding Profile** if necessary.

Slide 14 - Tunnel 2.0 Configuration

# Tunnel 2.0 Configuration

- Tunnel version selected in the **Forwarding Profile**
  - Configure Tunnel 2.0 functionality
  - Assign to specific groups in the **App Profile**



Edit Forwarding Profile

FORWARDING PROFILE ACTION FOR ZIA

On Trusted Network: ● Tunnel | Tunnel With Local Proxy | Enforce Proxy | None

Tunnel Version Selection (v. 2.0.0+ | v. 2.0.0+): Z-Tunnel 1.0 | ● Z-Tunnel 2.0

Advanced Z-Tunnel 2.0 Configuration

Z-Tunnel 2.0 Transport Settings

Primary Transport Selection: ● DTLS | TLS

DTLS Connection Timeout (In Seconds): 9

TLS Connection Timeout (In Seconds): 5

MTU for Zscaler Adapter (v. 2.1.2+ | v. 2.1.2+): 0

Allow Fallback
☑ TLS
☑ Z-Tunnel 1.0

Configure System Proxy Settings

System Proxy Settings

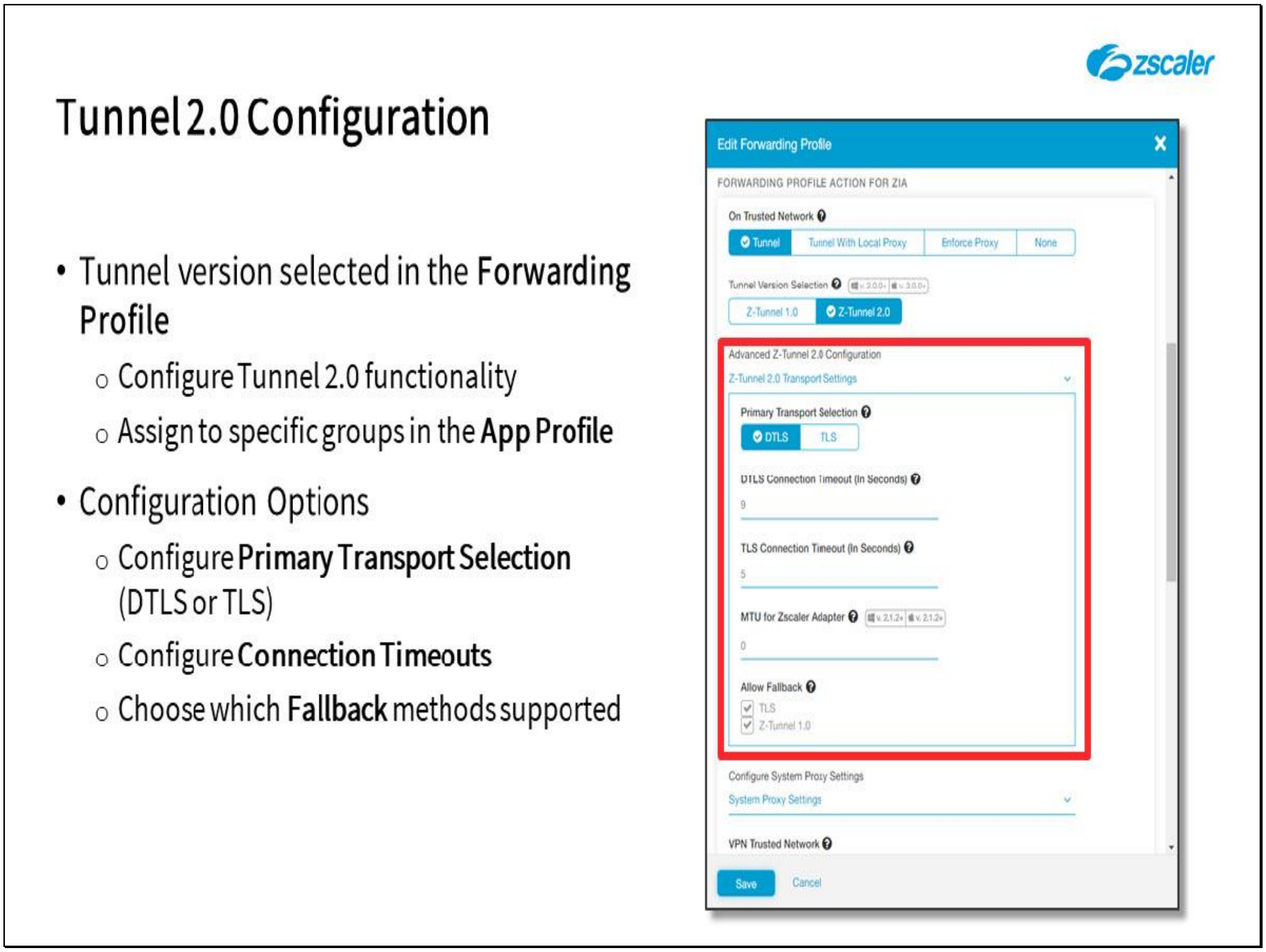
VPN Trusted Network

## Slide notes

To use **Tunnel 2.0**, in the **Forwarding Profile**, under **FORWARDING PROFILE ACTION FOR ZIA**, for each of the available scenarios (**On Trusted Network**, **Off Trusted Network**, **VPN Trusted Network**) select **Tunnel**, then **Z-Tunnel 2.0**.

Remember, the **Forwarding Profile** is assigned to the required user groups in the **App Profile** configuration.

## Slide 15 - Tunnel 2.0 Configuration



# Tunnel 2.0 Configuration

- Tunnel version selected in the **Forwarding Profile**
  - Configure Tunnel 2.0 functionality
  - Assign to specific groups in the **App Profile**
- Configuration Options
  - Configure **Primary Transport Selection** (DTLS or TLS)
  - Configure **Connection Timeouts**
  - Choose which **Fallback** methods supported

Edit Forwarding Profile

FORWARDING PROFILE ACTION FOR ZIA

On Trusted Network: [✓ Tunnel] | Tunnel With Local Proxy | Enforce Proxy | None

Tunnel Version Selection (v. 2.0.0+ | v. 2.0.0+): Z-Tunnel 1.0 | [✓ Z-Tunnel 2.0]

Advanced Z-Tunnel 2.0 Configuration

Z-Tunnel 2.0 Transport Settings

Primary Transport Selection: [✓ DTLS] | TLS

DTLS Connection Timeout (In Seconds): 9

TLS Connection Timeout (In Seconds): 5

MTU for Zscaler Adapter (v. 2.1.2+ | v. 2.1.2+): 0

Allow Fallback

☑ TLS

☑ Z-Tunnel 1.0

Configure System Proxy Settings

System Proxy Settings

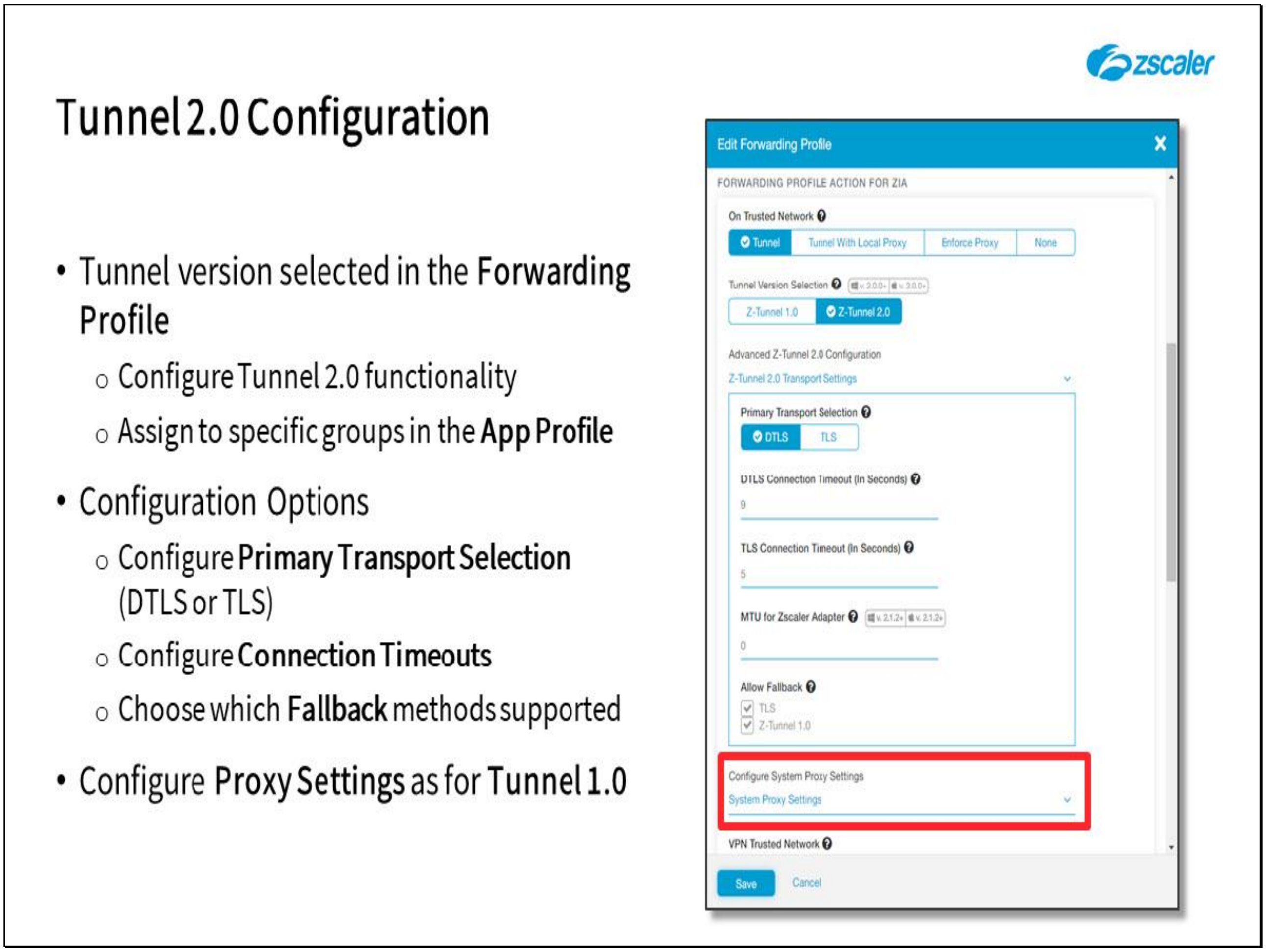
VPN Trusted Network

## Slide notes

In the **Z-Tunnel 2.0 Transport Settings**, you have the option to specify the **Primary Transport Selection**, whether **DTLS** or **TLS**, the default setting being **DTLS**. You can also adjust **Connection Timeouts** for **DTLS** and **TLS** independently, or the **MTU for Zscaler Adapter**, if the built-in defaults do not work in your environment. You can also control the allowed protocols for the fallback process, options being to enable fallback to **TLS** and/or **Z-Tunnel 1.0**.

Note that if you allow the fallback to the **Tunnel 1.0** method, any traffic sent on such a connection is subject only to the basic Internet Security service. There is no option to support traffic other than **TCP** on ports **80**/**443**, so most of the benefits of **Tunnel 2.0** are lost.

## Slide 16 - Tunnel 2.0 Configuration



# Tunnel 2.0 Configuration

- Tunnel version selected in the **Forwarding Profile**
  - Configure Tunnel 2.0 functionality
  - Assign to specific groups in the **App Profile**
- Configuration Options
  - Configure **Primary Transport Selection** (DTLS or TLS)
  - Configure **Connection Timeouts**
  - Choose which **Fallback** methods supported
- Configure **Proxy Settings** as for **Tunnel 1.0**

Edit Forwarding Profile

FORWARDING PROFILE ACTION FOR ZIA

On Trusted Network: Tunnel | Tunnel With Local Proxy | Enforce Proxy | None

Tunnel Version Selection (v. 2.0.0+ | v. 2.0.0+): Z-Tunnel 1.0 | Z-Tunnel 2.0

Advanced Z-Tunnel 2.0 Configuration

Z-Tunnel 2.0 Transport Settings

Primary Transport Selection: DTLS | TLS

DTLS Connection Timeout (In Seconds): 9

TLS Connection Timeout (In Seconds): 5

MTU for Zscaler Adapter (v. 2.1.2+ | v. 2.1.2+): 0

Allow Fallback

☑ TLS

☑ Z-Tunnel 1.0

Configure System Proxy Settings

System Proxy Settings

VPN Trusted Network

## Slide notes

The **System Proxy Settings** available are exactly the same as for the **Tunnel 1.0** method. You may need to provide the URL for a custom PAC file here, to support the required **Domain-based** forwarding mechanism (we'll discuss this requirement in a later section of this module).

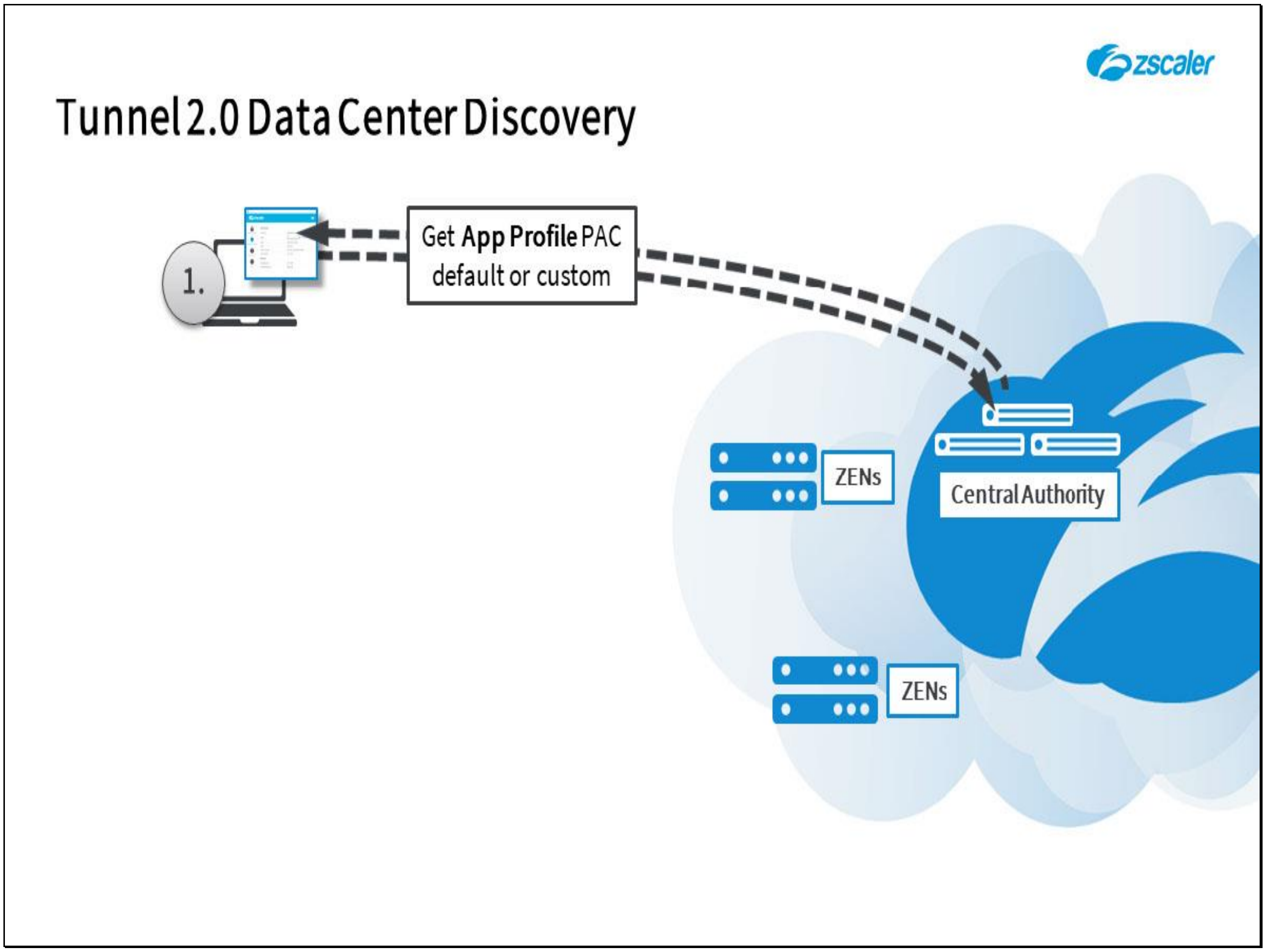
## Slide 17 - Tunnel 2.0 Data Center Discovery

Tunnel 2.0 Data Center Discovery

### Slide notes

The next topic that we will cover is a look at how Zscaler App discovers the ZENs and their capabilities.

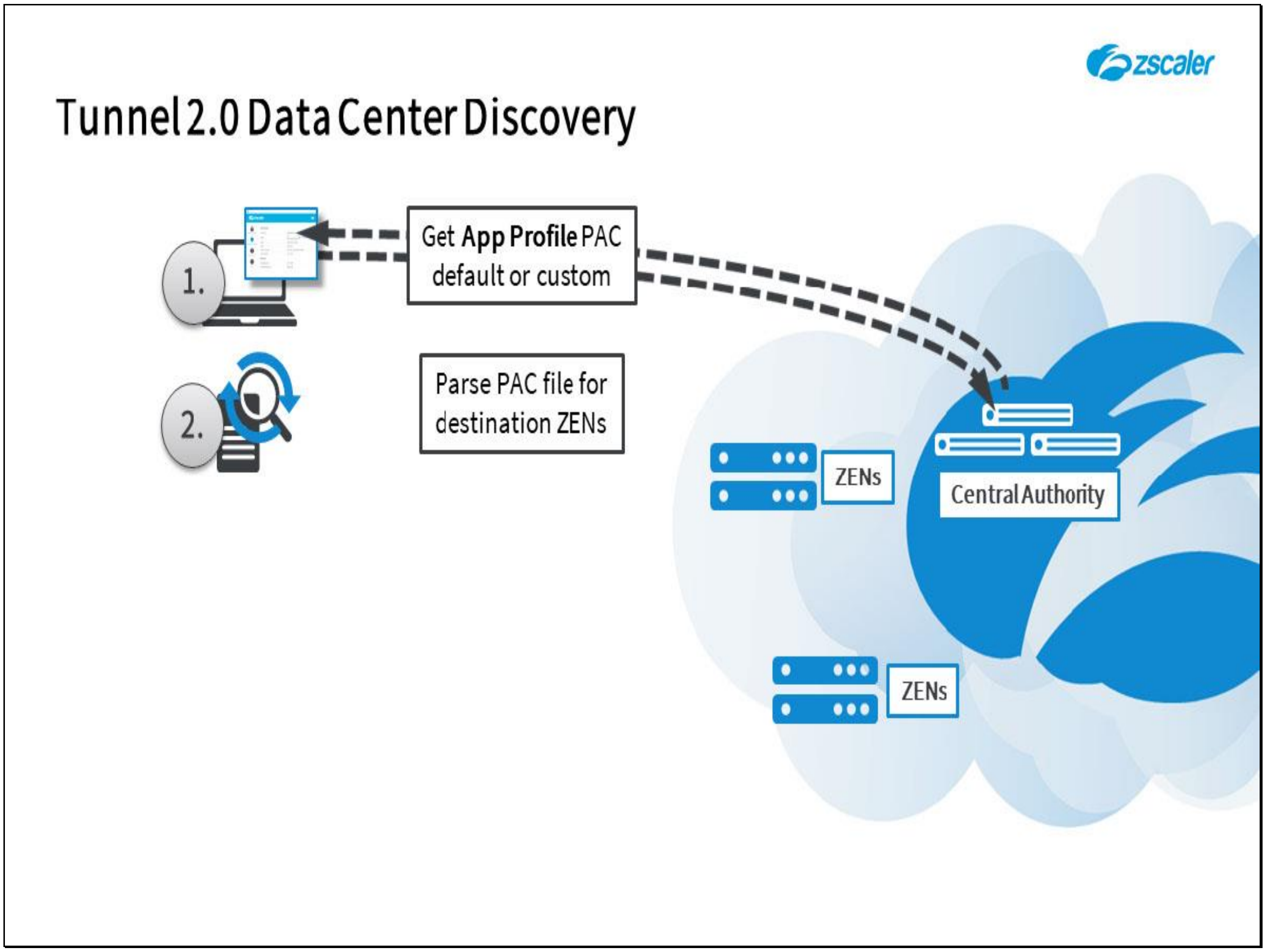
## Slide 18 - Tunnel 2.0 Data Center Discovery



### Slide notes

The first step in the process is for Z App to retrieve the relevant **App Profile** PAC file from the appropriate Zscaler cloud Central Authority (CA), whether that is the default PAC file or a specified custom file. If the App is unable to download a custom file, the default file for the cloud is used.

## Slide 19 - Tunnel 2.0 Data Center Discovery

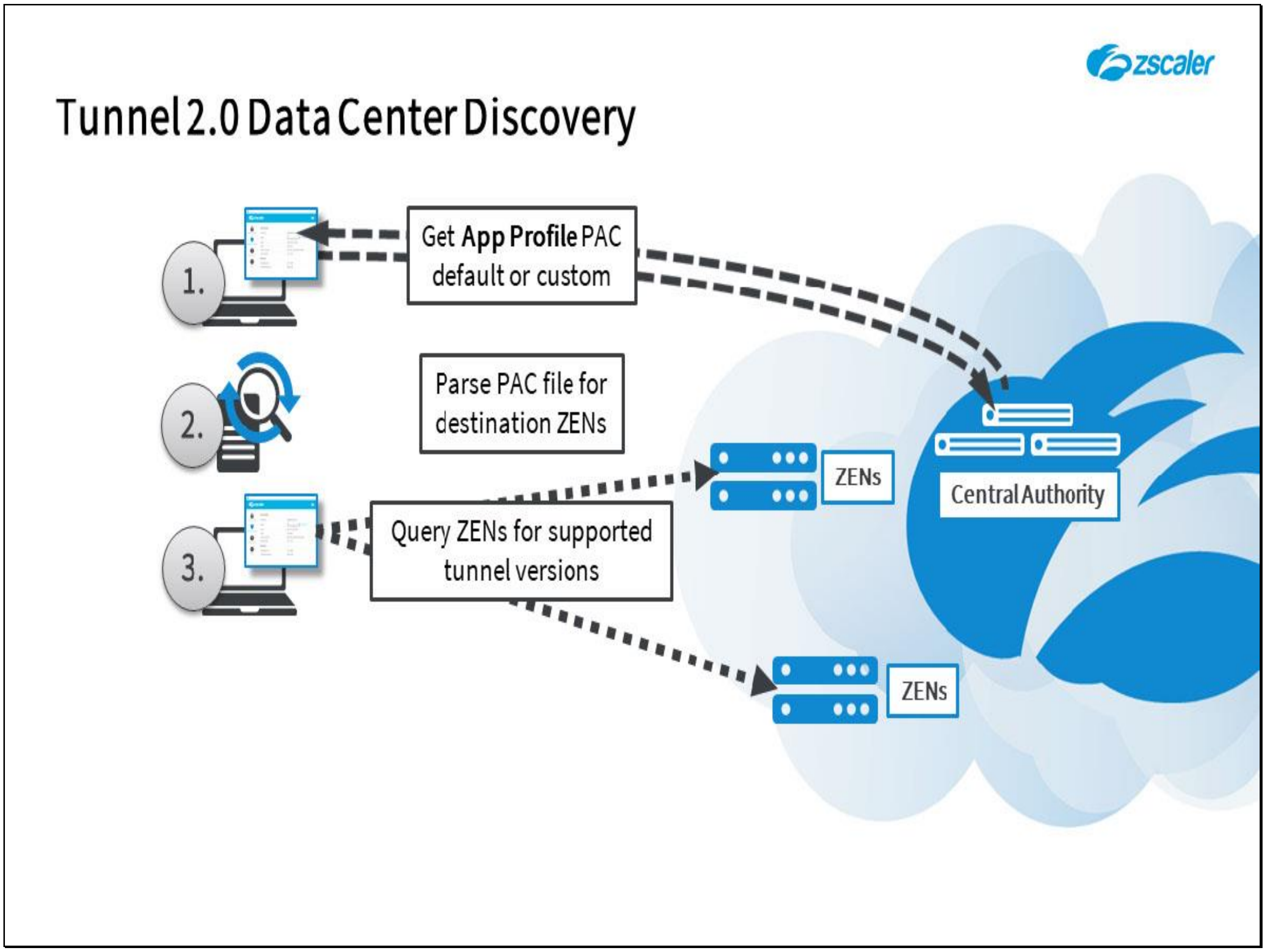

zscaler
Tunnel 2.0 Data Center Discovery
1.
Get App Profile PAC default or custom
2.
Parse PAC file for destination ZENs
ZENs
Central Authority
ZENs

### Slide notes

The App then validates the PAC file syntax, looking for the proxy **Return** statements for the primary and secondary ZENs:

- If the proxy statements are empty, the PAC file is marked as invalid;
- Otherwise the PAC file is considered valid and the proxy destinations resolved.

Slide 20 - Tunnel 2.0 Data Center Discovery



zscaler

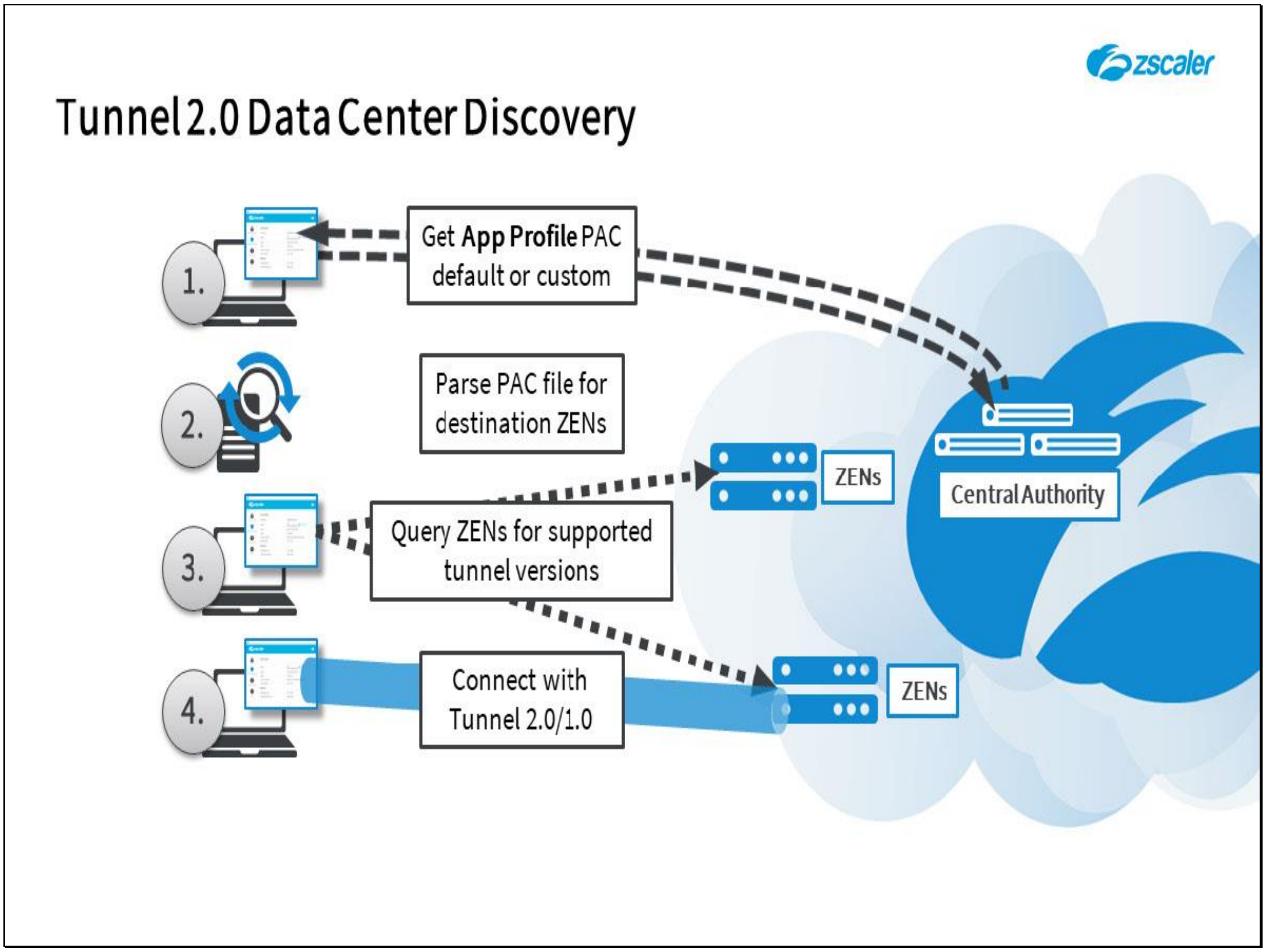
Tunnel 2.0 Data Center Discovery

1.
Get **App Profile** PAC default or custom
2.
Parse PAC file for destination ZENs
3.
Query ZENs for supported tunnel versions
ZENs
Central Authority
ZENs

## Slide notes

The App will then query the IPs for the ZENs specified in the PAC file, for both the primary and secondary connections, looking for the supported tunnel versions. The first ZEN from the list that supports **Tunnel 2.0** will be selected as the initial tunnel destination.

## Slide 21 - Tunnel 2.0 Data Center Discovery

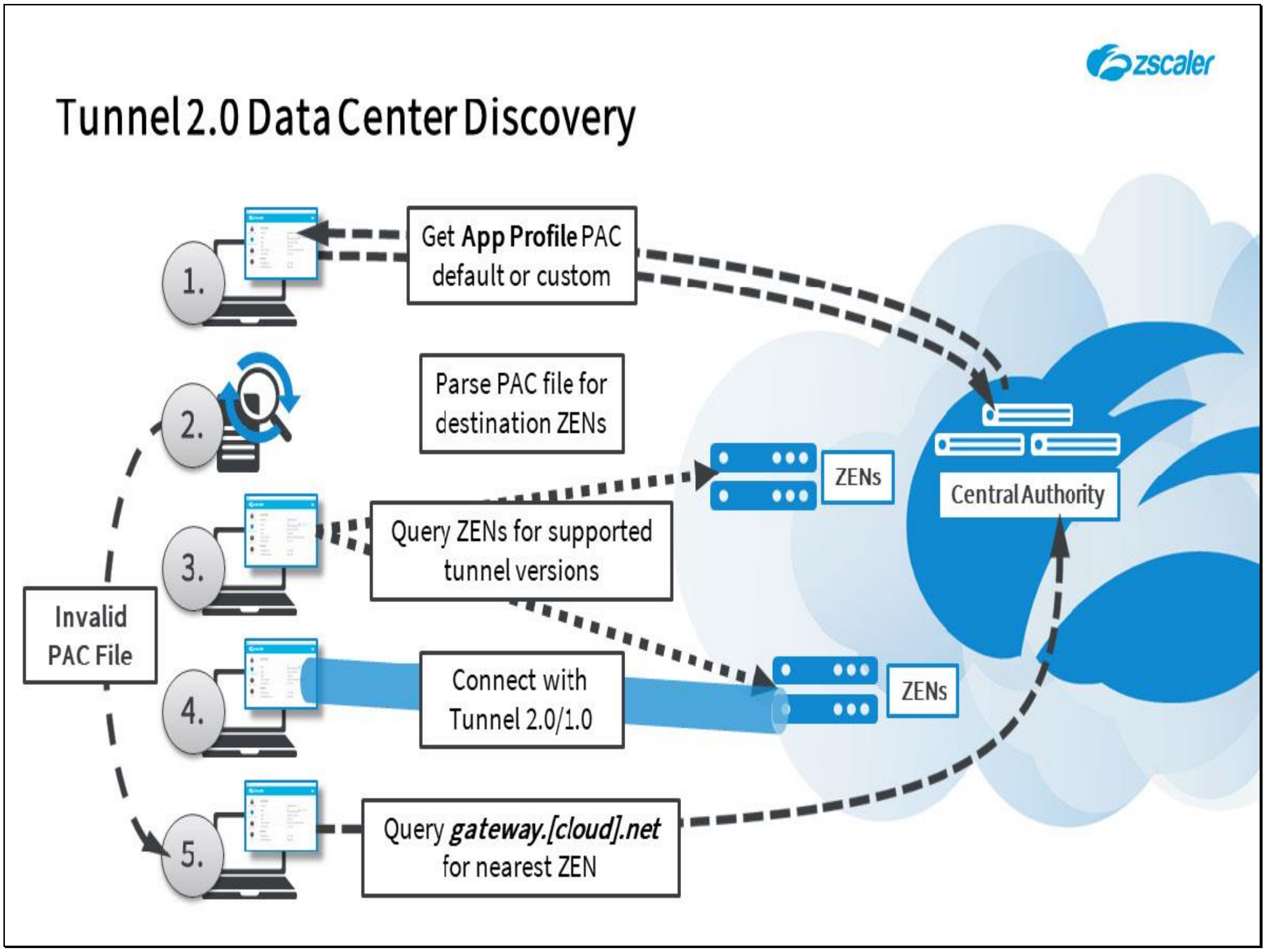


zscaler
Tunnel 2.0 Data Center Discovery
1.
Get App Profile PAC default or custom
2.
Parse PAC file for destination ZENs
3.
Query ZENs for supported tunnel versions
4.
Connect with Tunnel 2.0/1.0
ZENs
ZENs
Central Authority

## Slide notes

If the selected ZEN supports it, then a **Tunnel 2.0** connection will be attempted to it. If **Tunnel 2.0** setup fails Z App will try a **Tunnel 1.0** connection to the same ZENs.

Slide 22 - Tunnel 2.0 Data Center Discovery



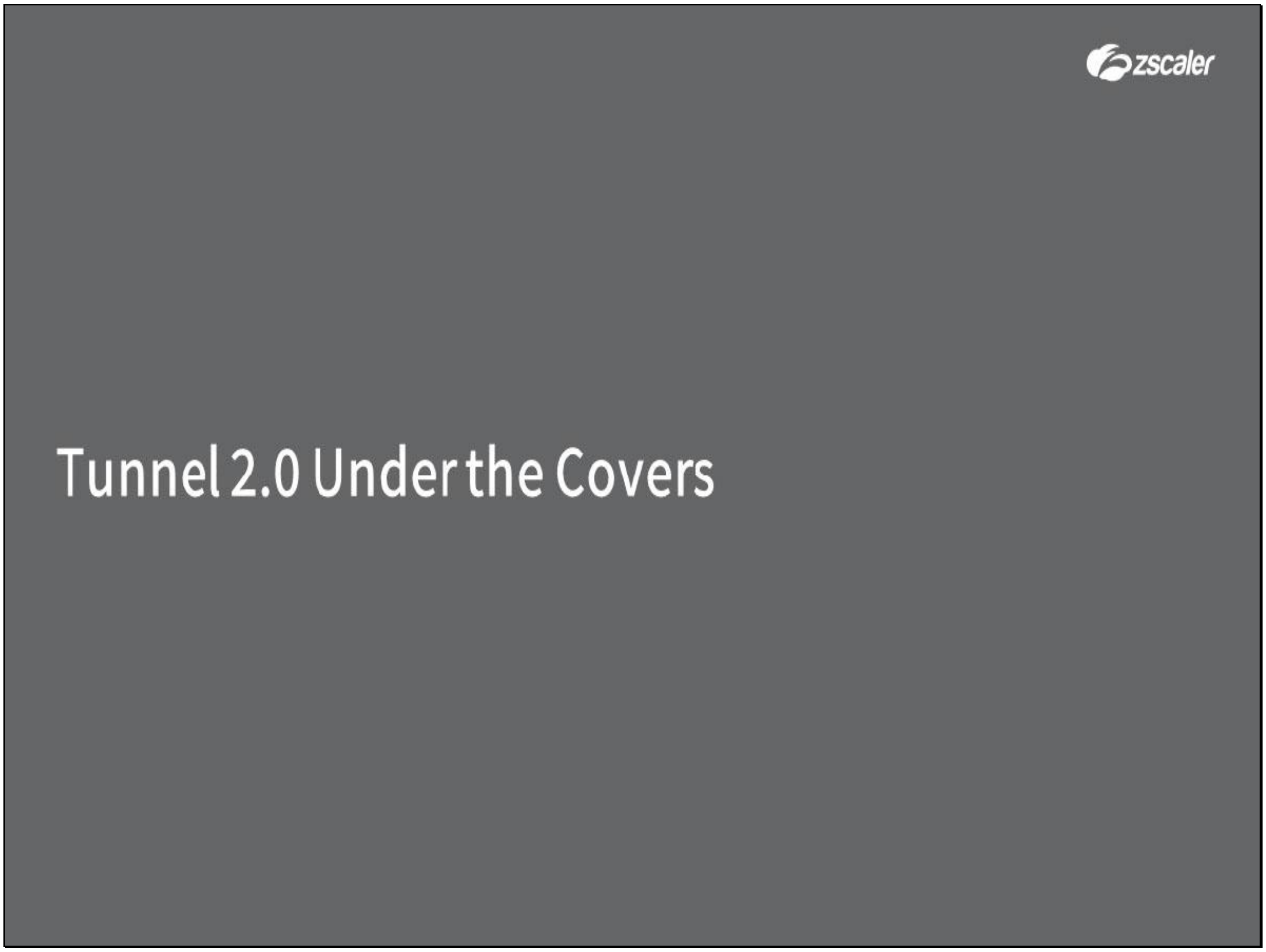
Tunnel 2.0 Data Center Discovery

zscaler
1.
Get App Profile PAC default or custom
2.
Parse PAC file for destination ZENs
3.
Query ZENs for supported tunnel versions
ZENs
Central Authority
Invalid PAC File
4.
Connect with Tunnel 2.0/1.0
ZENs
5.
Query gateway.[cloud].net for nearest ZEN

Slide notes

If **Tunnel 1.0** setup also fails, or if the PAC file was invalid in the first place, then Z App will fall back to query **gateway.[cloud].net** for the next closest ZEN.

Slide 23 - Tunnel 2.0 Under the Covers



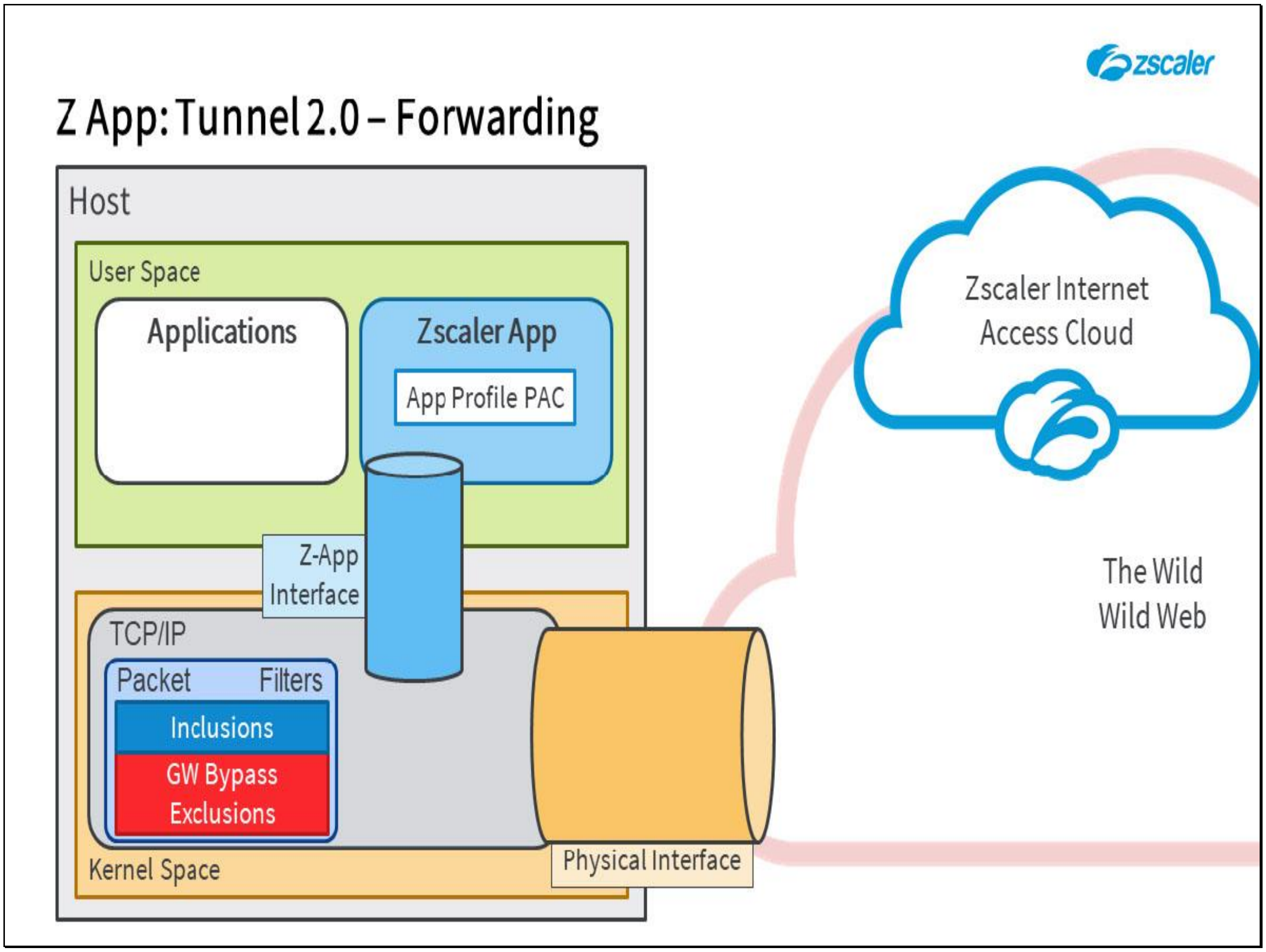
zscaler

Tunnel 2.0 Under the Covers

Slide notes

The next topic that we will cover is a detailed look at the **Tunnel 2.0** forwarding method.

Slide 24 - Z App: Tunnel 2.0 – Forwarding



## Slide notes

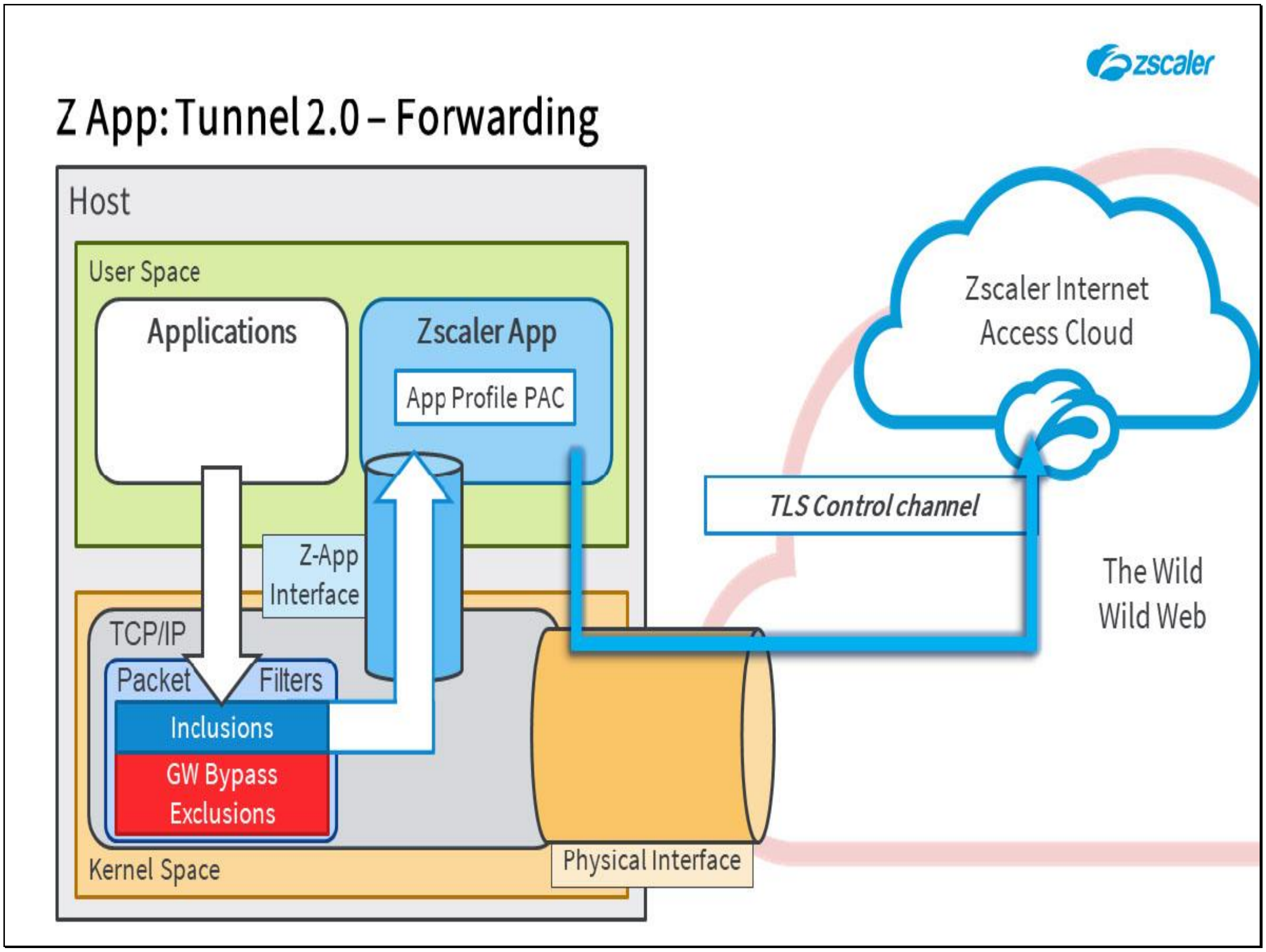
When configured to use the **Tunnel 2.0** forwarding method, Zscaler App acts like a pseudo-VPN client and will tunnel just about **all unicast IPv4 traffic** to the ZIA service, regardless of protocol (**TCP**, **UDP**, **ICMP**) or port. This allows a wider range of the available ZIA policies and filters to be applied to this traffic for better protection of the end user's device.

For maximum efficiency **Tunnel 2.0** defaults to use **DTLS** tunnels (**UDP** on port **443**) which are validated and protected from tampering, although they are not currently encrypted. If greater reliability is required, **Tunnel 2.0** will fall back to use null-encrypted **TLS** tunnels (**TCP** on port **443**) and if necessary, it will fall all the way back to the **Tunnel 1.0** method. Tunnels are persistent and packet-based, so no stream conversion is required.

Let's step through the workings of **Tunnel 2.0**:

- As before, it is the packet filters defined on the system by the Zscaler App (through the **VPN GW Bypass** configuration or **Include**/**Exclude** rules) that control the forwarding of traffic, whether it is to be tunneled to the ZIA service or forwarded direct.

Slide 25 - Z App: Tunnel 2.0 – Forwarding



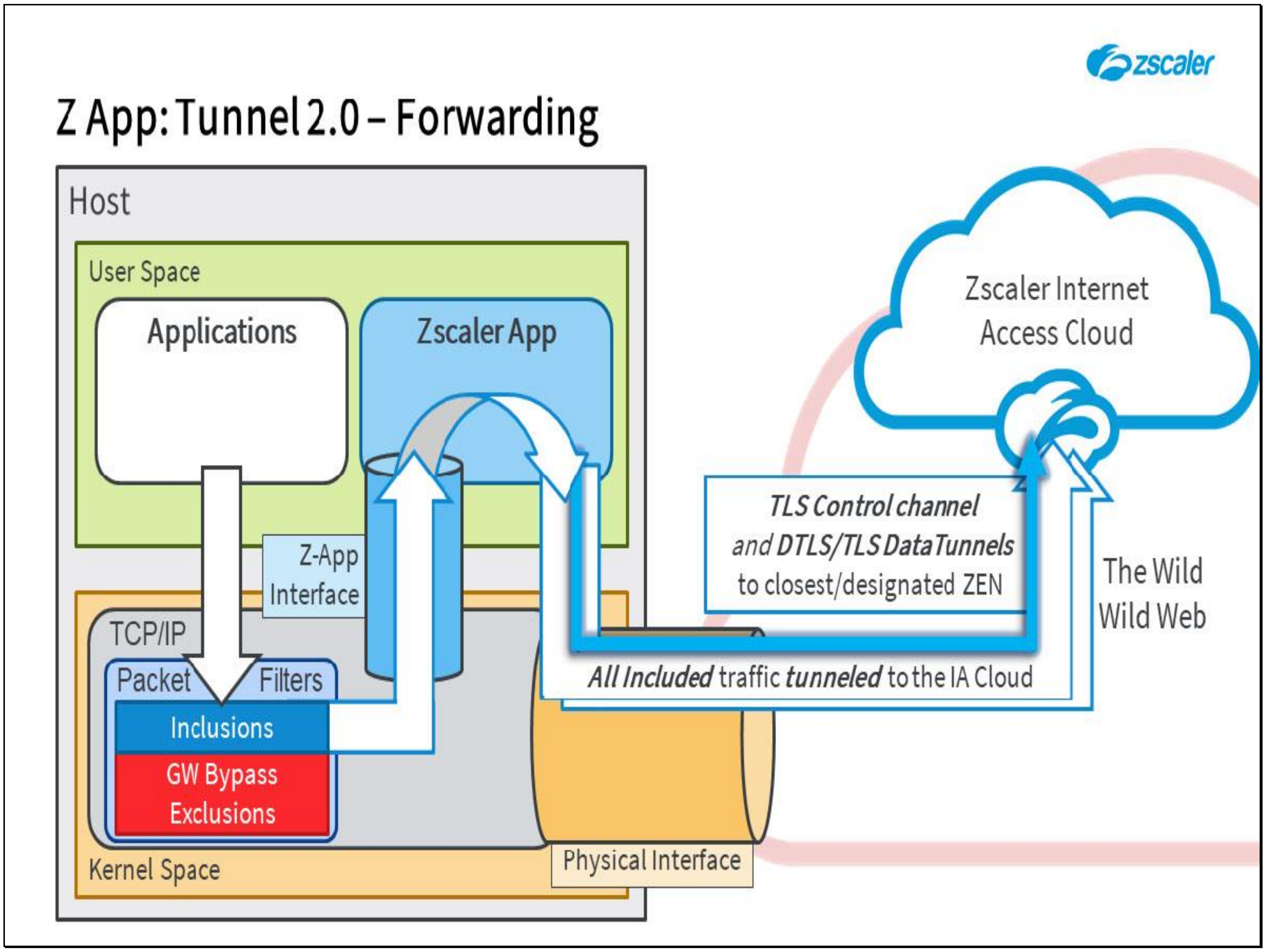
Z App: Tunnel 2.0 – Forwarding

Slide notes

- Before transferring any data however, the Zscaler App must first establish a persistent **TLS control channel** to the nearest (or a specified) **Tunnel 2.0** compliant ZEN. The control channel is used for device and/or end user authentication, which establishes a **session-id** for the subsequent data tunnels, plus it allows the App to detect what transfer modes are supported by the ZEN.

  As a by-product, this control channel also provides a bi-directional communications channel, that allows the updating of Zscaler App Portal policies to Zscaler App on the device in real-time (rather that the App having to poll for them every 60 minutes).

Slide 26 - Z App: Tunnel 2.0 – Forwarding



Z App: Tunnel 2.0 – Forwarding

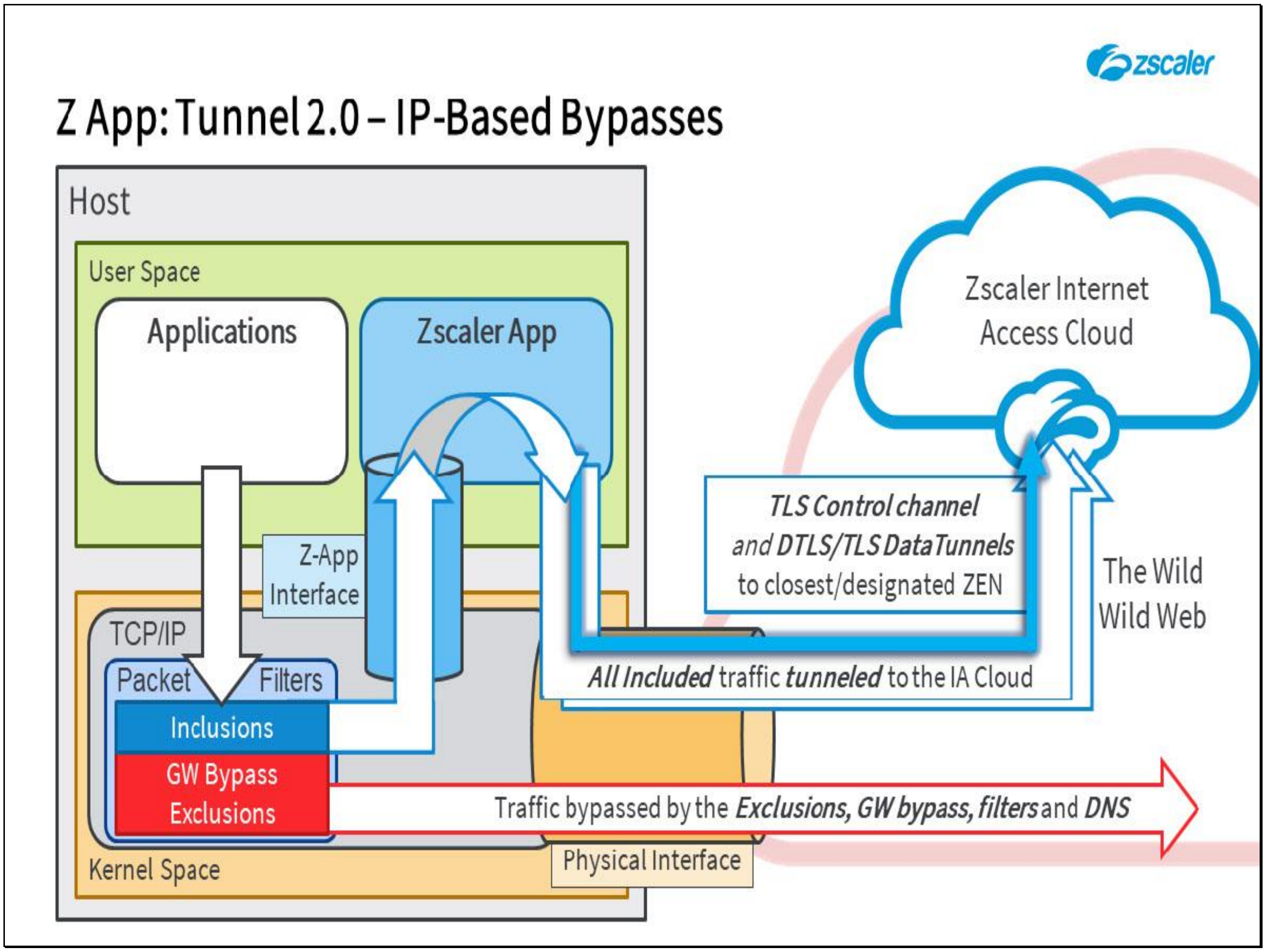
Slide notes

- Packets to be forwarded to the ZIA service will be sent on one or more **Tunnel 2.0 data channels**, which are authenticated using the **Proxy Digest** method and the **session-id** established on the control channel. By default they will first attempt to connect using null-encrypted **DTLS** tunnels over **UDP** on destination port **443**. Using **DTLS** allows for better throughput, plus it allows for server validation and integrity checking to prevent MitM attacks.

  Should it become necessary, for better transmission reliability, **Tunnel 2.0** will fall back to using null-encrypted **TLS** tunnels instead. Under some circumstances, it may also fallback to the **Tunnel 1.0** method. The initial protocol to use for tunnels (**DTLS** or **TLS**) can be specified and the fallback methods supported are configurable.

  You can add destination IPs/Subnets that you specifically want to **include** for **Tunnel 2.0** forwarding in the **Destination Inclusions** option in the **App Profile**. This configuration is only applicable to the **Tunnel 2.0** method and allows you to add any applicable IPv4 address or IPv4 subnet and mask; wildcards are supported. For Windows devices you can also add protocol (UDP/TCP) and port. By default we automatically add the '**quad zero**' IPv4 default route to the inclusion configuration, meaning all IPv4 traffic on any port will be processed by Zscaler App.

Slide 27 - Z App: Tunnel 2.0 – IP-Based Bypasses

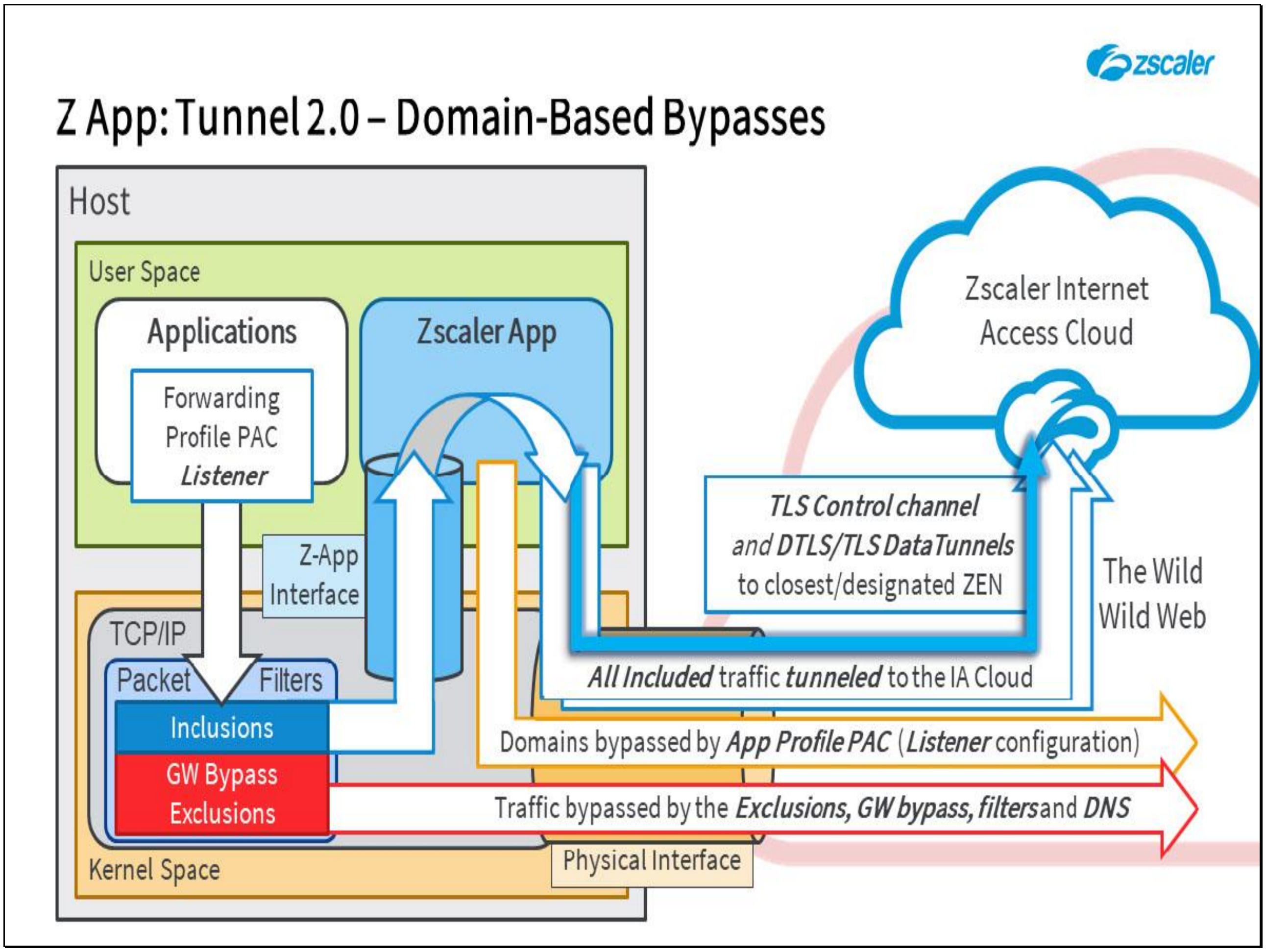

zscaler
Z App: Tunnel 2.0 – IP-Based Bypasses
Host
User Space
Applications
Zscaler App
Z-App Interface
TCP/IP
Packet Filters
Inclusions
GW Bypass Exclusions
Kernel Space
Physical Interface
Zscaler Internet Access Cloud
TLS Control channel and DTLS/TLS Data Tunnels to closest/designated ZEN
The Wild Wild Web
All Included traffic tunneled to the IA Cloud
Traffic bypassed by the Exclusions, GW bypass, filters and DNS

Slide notes

- As before, some categories of traffic may be forwarded direct to the physical interface without the Zscaler App ever processing it. This includes; DNS traffic, anything added to the **Hostname/IP Address Bypass for VPN Gateway** field of the **App Profile**, plus any destination IPs/Subnets that you specifically want to **exclude** from **Tunnel 2.0** forwarding.

  As with the **Inclusions**, the **Destination Exclusions** option is only applicable to the **Tunnel 2.0** method and is configured in the applied **App Profile**. As before, you can add any applicable IPv4 address or IPv4 subnet and mask, with wildcards if necessary. For Windows devices you can also add protocol (UDP/TCP) and port. By default we automatically add **Exclusions** for the RFC1918 private networks and the IPv4 multicast address space.

Slide 28 - Z App: Tunnel 2.0 – Domain-Based Bypasses



Z App: Tunnel 2.0 – Domain-Based Bypasses

*TLS Control channel and DTLS/TLS Data Tunnels* to closest/designated ZEN

***All Included*** traffic ***tunneled*** to the IA Cloud

Domains bypassed by ***App Profile PAC*** (***Listener*** configuration)

Traffic bypassed by the ***Exclusions, GW bypass, filters*** and ***DNS***

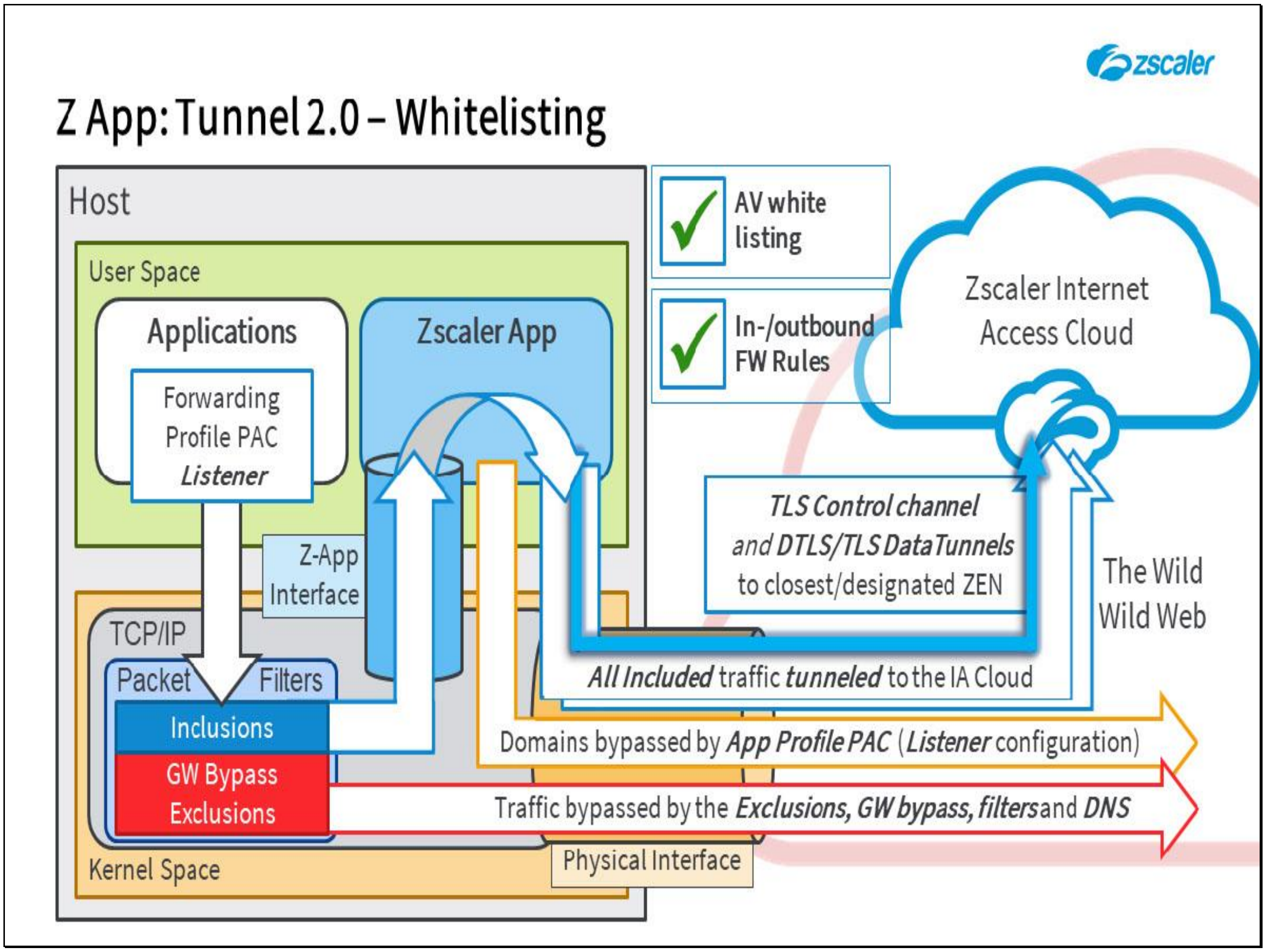
## Slide notes

- Finally, it is possible to add domain-based bypasses, however with **Tunnel 2.0** this must be done in a 2-step process:

  1. First, using the **Forwarding Profile** PAC file, you must signal to that App that there is a domain-based bypass using the **${ZAPP_TUNNEL2_BYPASS}** macro as a '**Listener**' configuration;
  2. Then, you must actually specify the bypass for the domain or host in the **App Profile** PAC file.

  This method for bypassing domain-based destinations is significantly different to the **Tunnel 1.0** bypass mechanisms. It will require careful planning to migrate any existing **Tunnel 1.0** bypasses into the **Tunnel 2.0** model when moving to the 2.0 method.

Slide 29 - Z App: Tunnel 2.0 – Whitelisting

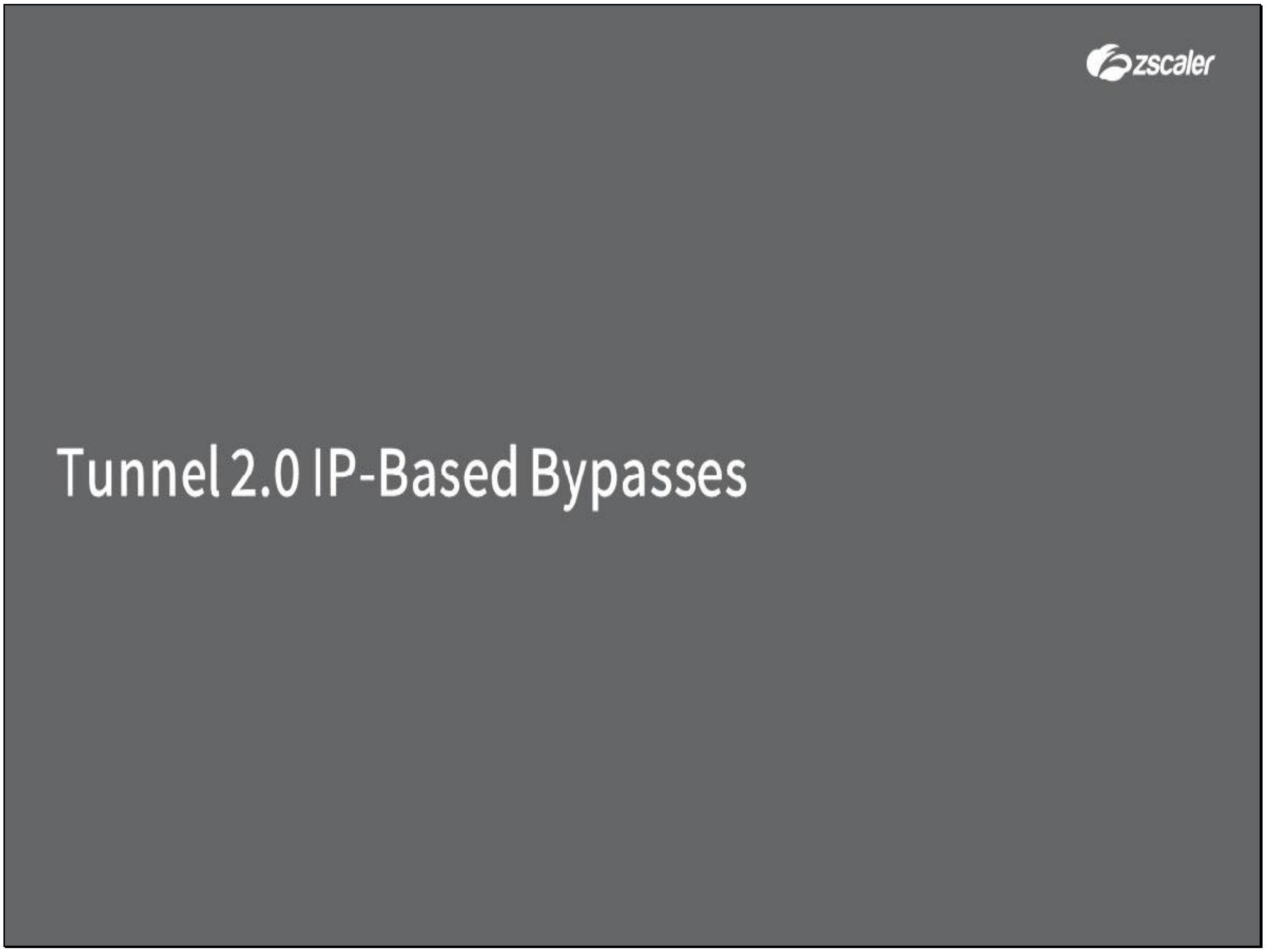


zscaler
Z App: Tunnel 2.0 – Whitelisting
Host
User Space
Applications
Forwarding Profile PAC Listener
Zscaler App
Z-App Interface
AV white listing
In-/outbound FW Rules
Zscaler Internet Access Cloud
TLS Control channel and DTLS/TLS Data Tunnels to closest/designated ZEN
The Wild Wild Web
TCP/IP
Packet Filters
Inclusions
GW Bypass Exclusions
All Included traffic tunneled to the IA Cloud
Domains bypassed by App Profile PAC (Listener configuration)
Traffic bypassed by the Exclusions, GW bypass, filters and DNS
Physical Interface
Kernel Space

Slide notes

- As before, the App needs to be white listed by any AV software installed on the host machine, plus inbound and outbound Firewall rules are required for the **zsatunnel.exe** process.

Slide 30 - Tunnel 2.0 IP-Based Bypasses

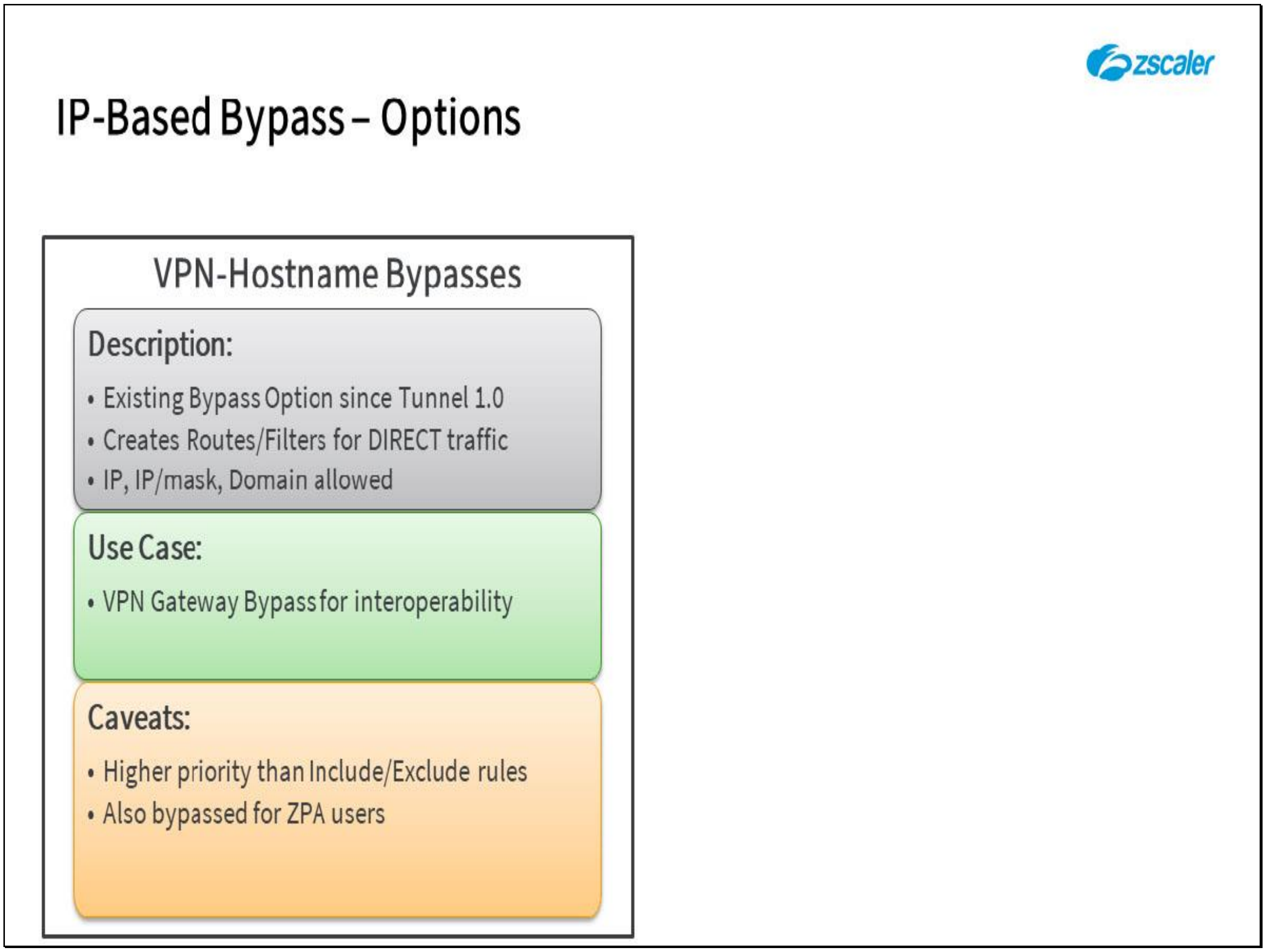


Tunnel 2.0 IP-Based Bypasses

Slide notes

The next topic that we will cover is how to manage IP-Layer (L3) bypasses when using Z App **Tunnel 2.0**.

Slide 31 - IP-Based Bypass – Options



# IP-Based Bypass – Options

## VPN-Hostname Bypasses

**Description:**

- Existing Bypass Option since Tunnel 1.0
- Creates Routes/Filters for DIRECT traffic
- IP, IP/mask, Domain allowed

**Use Case:**

- VPN Gateway Bypass for interoperability

**Caveats:**

- Higher priority than Include/Exclude rules
- Also bypassed for ZPA users

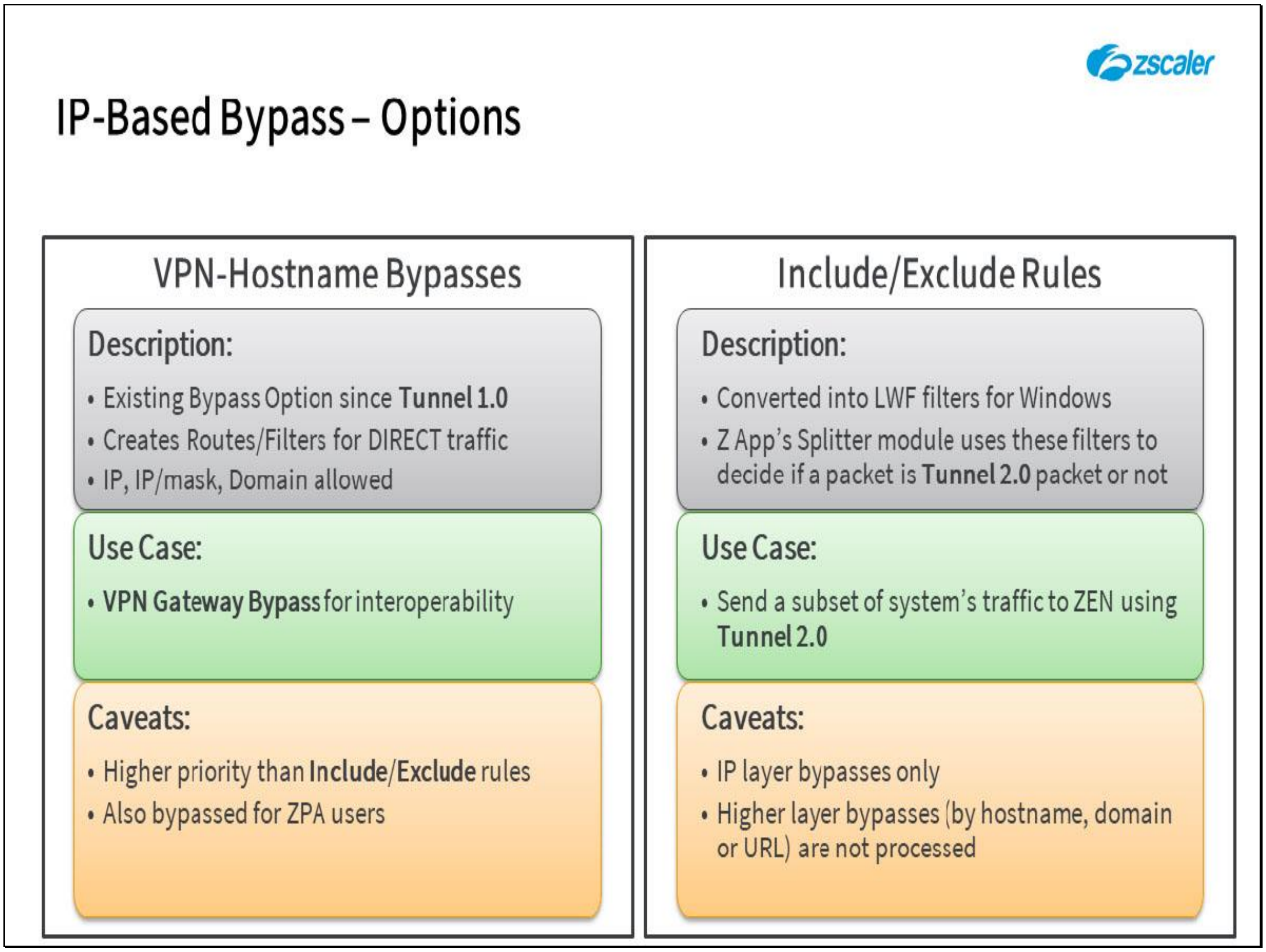
Slide notes

IP-based (L3) bypasses for **Tunnel 2.0** are pretty straightforward, there are two options both configured in the **App Profile**:

- The first is the familiar **Hostname/IP Address Bypass for VPN Gateway** configuration in the App Profile. This has not changed at all compared to the **Tunnel 1.0** method and the same configuration options are still possible (IP addresses, IP Subnets, or FQDNs, up to 6,144 characters). The primary (and we recommend the only) use for this option is to specify the address(es) for any corporate VPN gateway device that should be the destination for traffic protected by a legacy VPN solution.

  Note that this configuration has the highest priority, Z App will look at these IPs/hosts and make a forwarding decision before processing the **Include**/**Exclude** rules. In addition, note that these bypasses also apply for ZPA traffic.

Slide 32 - IP-Based Bypass – Options



zscaler

## IP-Based Bypass – Options

### VPN-Hostname Bypasses

**Description:**

- Existing Bypass Option since **Tunnel 1.0**
- Creates Routes/Filters for DIRECT traffic
- IP, IP/mask, Domain allowed

**Use Case:**

- **VPN Gateway Bypass** for interoperability

**Caveats:**

- Higher priority than **Include/Exclude** rules
- Also bypassed for ZPA users

### Include/Exclude Rules

**Description:**

- Converted into LWF filters for Windows
- Z App's Splitter module uses these filters to decide if a packet is **Tunnel 2.0** packet or not

**Use Case:**

- Send a subset of system's traffic to ZEN using **Tunnel 2.0**

**Caveats:**

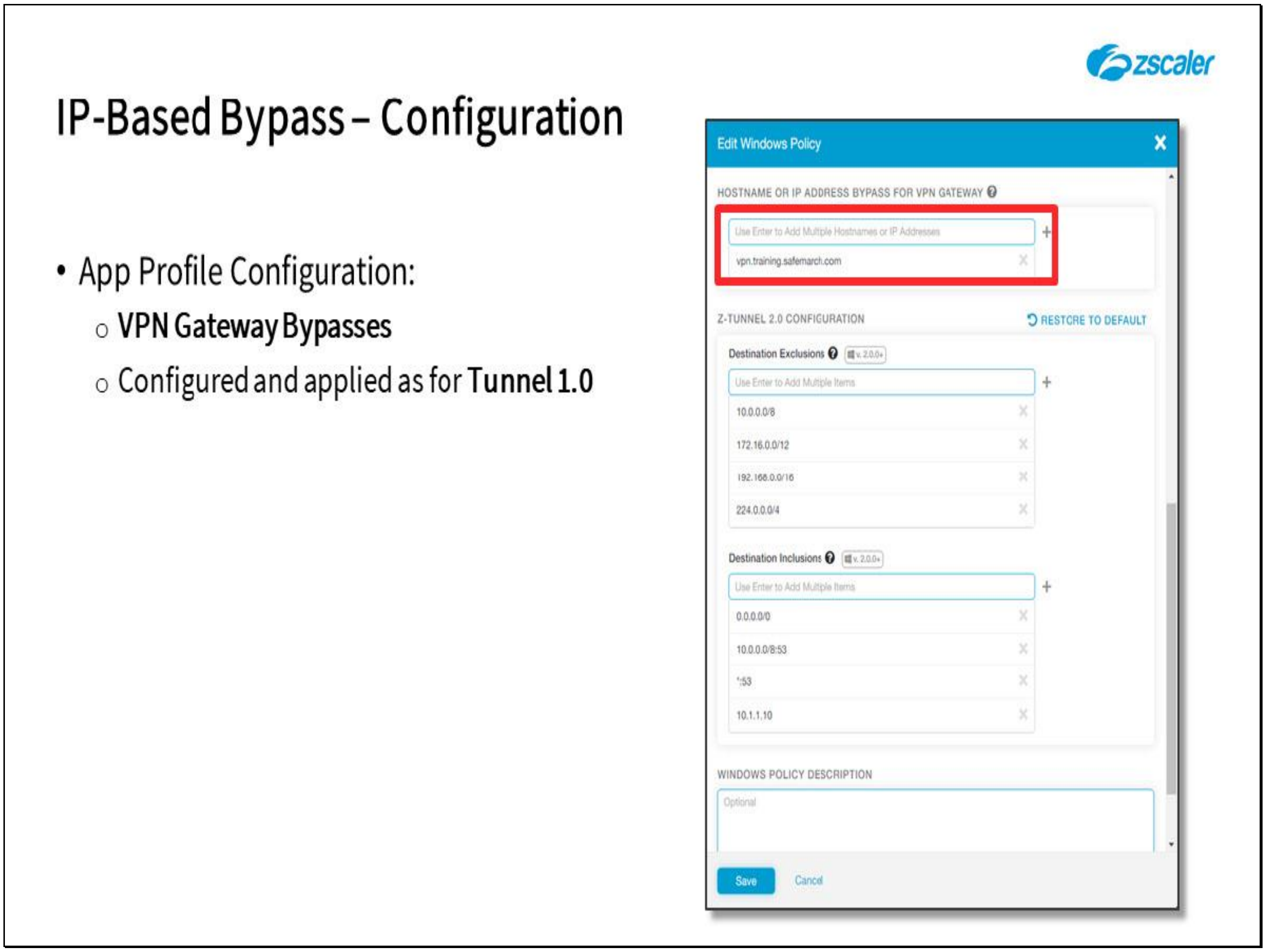
- IP layer bypasses only
- Higher layer bypasses (by hostname, domain or URL) are not processed

Slide notes

- The second option is the ability to define an **Include**/**Exclude** configuration. This is new in **Tunnel 2.0** and it allows you to steer traffic towards the ZIA cloud service in the tunnels, or to send it direct based on the destination IP addresses. All platforms support the configuration of **IP addresses** with or without **subnet masks**, the Windows platform also allows the definition of **Protocol** and **Port**.

  It is important to understand that these are L3 bypasses only, the **Include**/**Exclude** configuration has no ability to recognize, and therefore does not process traffic based on destination **FQDNs**, **Domains** or **URLs**.

## Slide 33 - IP-Based Bypass – Configuration



# IP-Based Bypass – Configuration

- App Profile Configuration:
  - VPN Gateway Bypasses
  - Configured and applied as for **Tunnel 1.0**

zscaler
Edit Windows Policy
Use Enter to Add Multiple Hostnames or IP Addresses
RESTORE TO DEFAULT
Use Enter to Add Multiple Items
v. 2.0.0+
Optional
Save
Cancel

HOSTNAME OR IP ADDRESS BYPASS FOR VPN GATEWAY

vpn.training.safemarch.com

Z-TUNNEL 2.0 CONFIGURATION

Destination Exclusions

10.0.0.0/8
172.16.0.0/12
192.168.0.0/16
224.0.0.0/4

Destination Inclusions

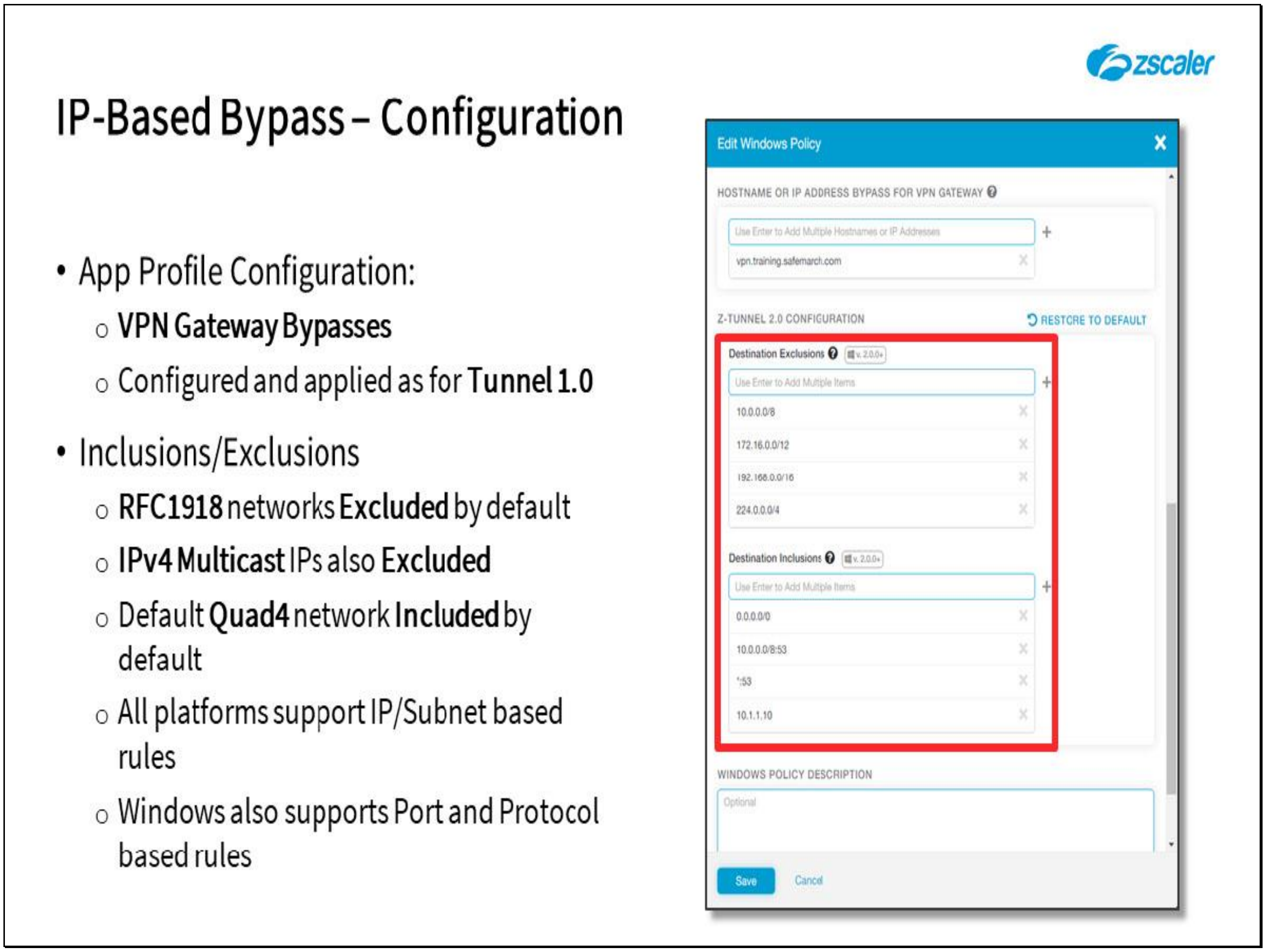
0.0.0.0/0
10.0.0.0/8:53
*:53
10.1.1.10

WINDOWS POLICY DESCRIPTION

### Slide notes

To configure the **Hostname/IP Address Bypass for VPN Gateway** option, open the relevant **App Profile** and scroll down to the applicable section. Nothing is added here by default, you will need to specify any relevant destination addresses. Remember, this configuration takes priority over the **Tunnel 2.0 Inclusion**/**Exclusion** configuration.

Slide 34 - IP-Based Bypass – Configuration



# IP-Based Bypass – Configuration

- App Profile Configuration:
  - **VPN Gateway Bypasses**
  - Configured and applied as for **Tunnel 1.0**
- Inclusions/Exclusions
  - **RFC1918** networks **Excluded** by default
  - **IPv4 Multicast** IPs also **Excluded**
  - Default **Quad4** network **Included** by default
  - All platforms support IP/Subnet based rules
  - Windows also supports Port and Protocol based rules

Edit Windows Policy

HOSTNAME OR IP ADDRESS BYPASS FOR VPN GATEWAY

vpn.training.safemarch.com

Z-TUNNEL 2.0 CONFIGURATION

Destination Exclusions (v. 2.0.0+)

10.0.0.0/8
172.16.0.0/12
192.168.0.0/16
224.0.0.0/4

Destination Inclusions (v. 2.0.0+)

0.0.0.0/0
10.0.0.0/8:53
*:53
10.1.1.10

WINDOWS POLICY DESCRIPTION

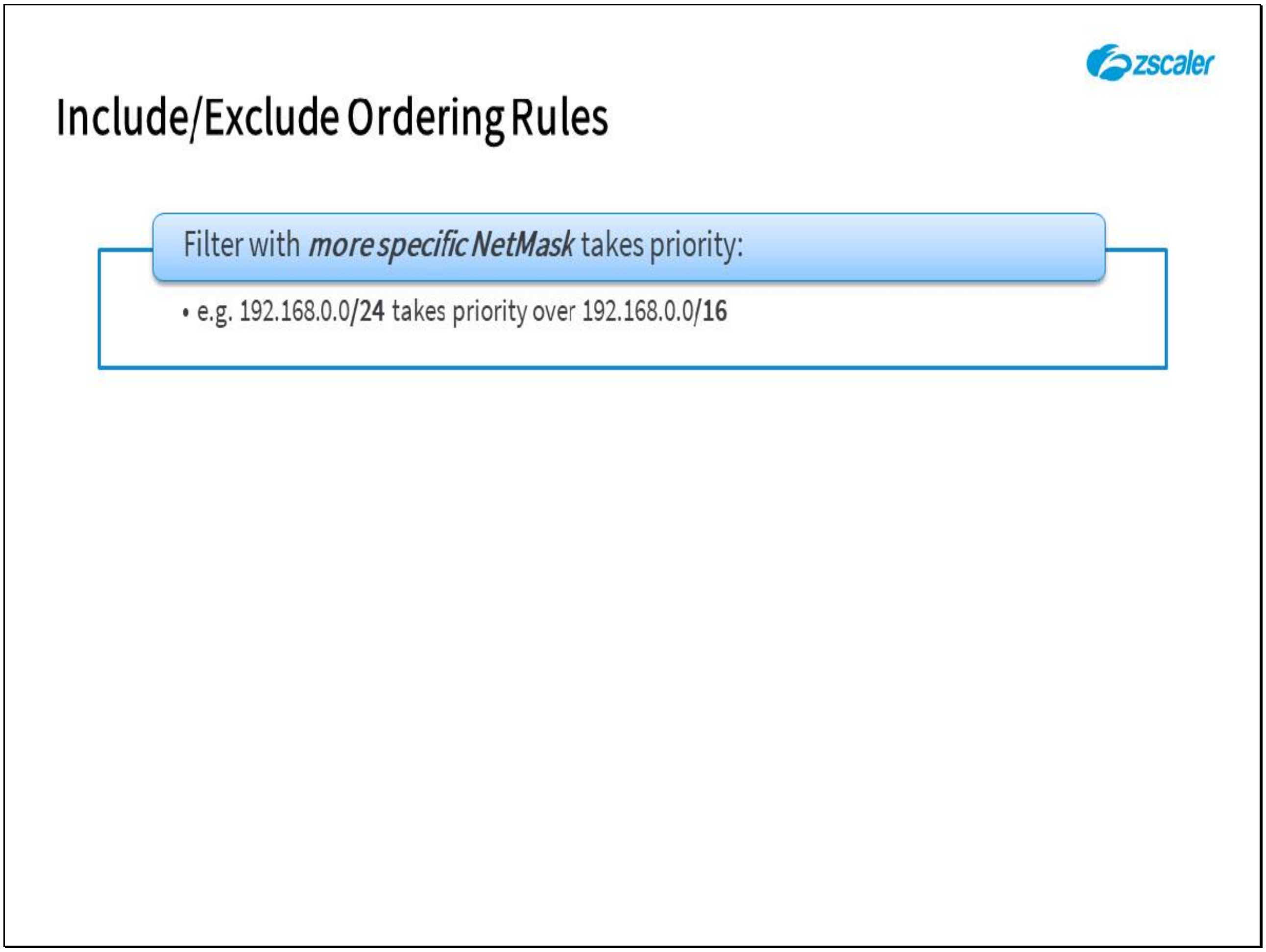
zscaler
×
Use Enter to Add Multiple Hostnames or IP Addresses
RESTORE TO DEFAULT
Use Enter to Add Multiple Items
Use Enter to Add Multiple Items
Optional
Save
Cancel

Slide notes

Also in the **App Profile**, the next section down is the **Z-TUNNEL 2.0 CONFIGURATION** section, with options to add **Destination Exclusions** and **Destination Inclusions**. By default we automatically add the '**quad zero**' IPv4 default route to the inclusion configuration, meaning all IP traffic on any port will be processed by Zscaler App. However, also by default we automatically add the **RFC1918 private networks** and the **IPv4 Multicast** address space as **Exclusions**, meaning that traffic for these networks will be sent direct (i.e. will NOT be tunneled).

As previously mentioned, all platforms support the configuration of **IP Addresses** with or without **Subnet Masks**, the Windows platform also allows the definition of **Protocol** and **Port**.

Slide 35 - Include/Exclude Ordering Rules



## Include/Exclude Ordering Rules

Filter with ***more specific NetMask*** takes priority:

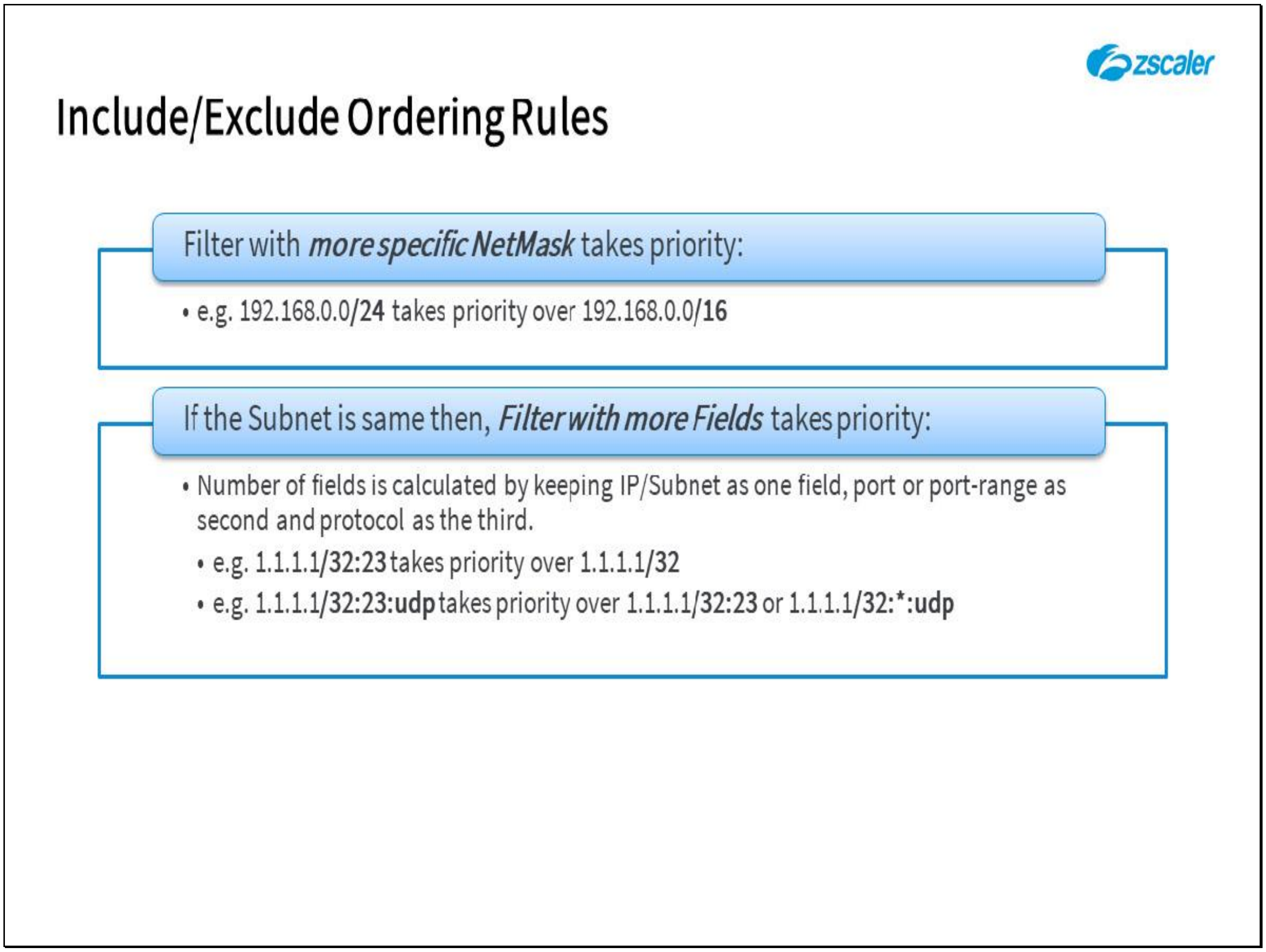
- e.g. 192.168.0.0/**24** takes priority over 192.168.0.0/**16**

Slide notes

Looking at how Z App processes the **Include**/**Exclude** configuration: The first criterion looked at is the **Netmask**, with a **more specific** mask taking priority.

So if there is an **Include** rule for the **192.168.0.0** network with a mask of **/24**, this will trump an **Exclude** rule with the same network defined, but with a **/16** mask. Meaning that traffic for **192.168.0.0/24** destination will be included;

Slide 36 - Include/Exclude Ordering Rules



## Include/Exclude Ordering Rules

Filter with ***more specific NetMask*** takes priority:

- e.g. 192.168.0.0/**24** takes priority over 192.168.0.0/**16**

If the Subnet is same then, ***Filter with more Fields*** takes priority:

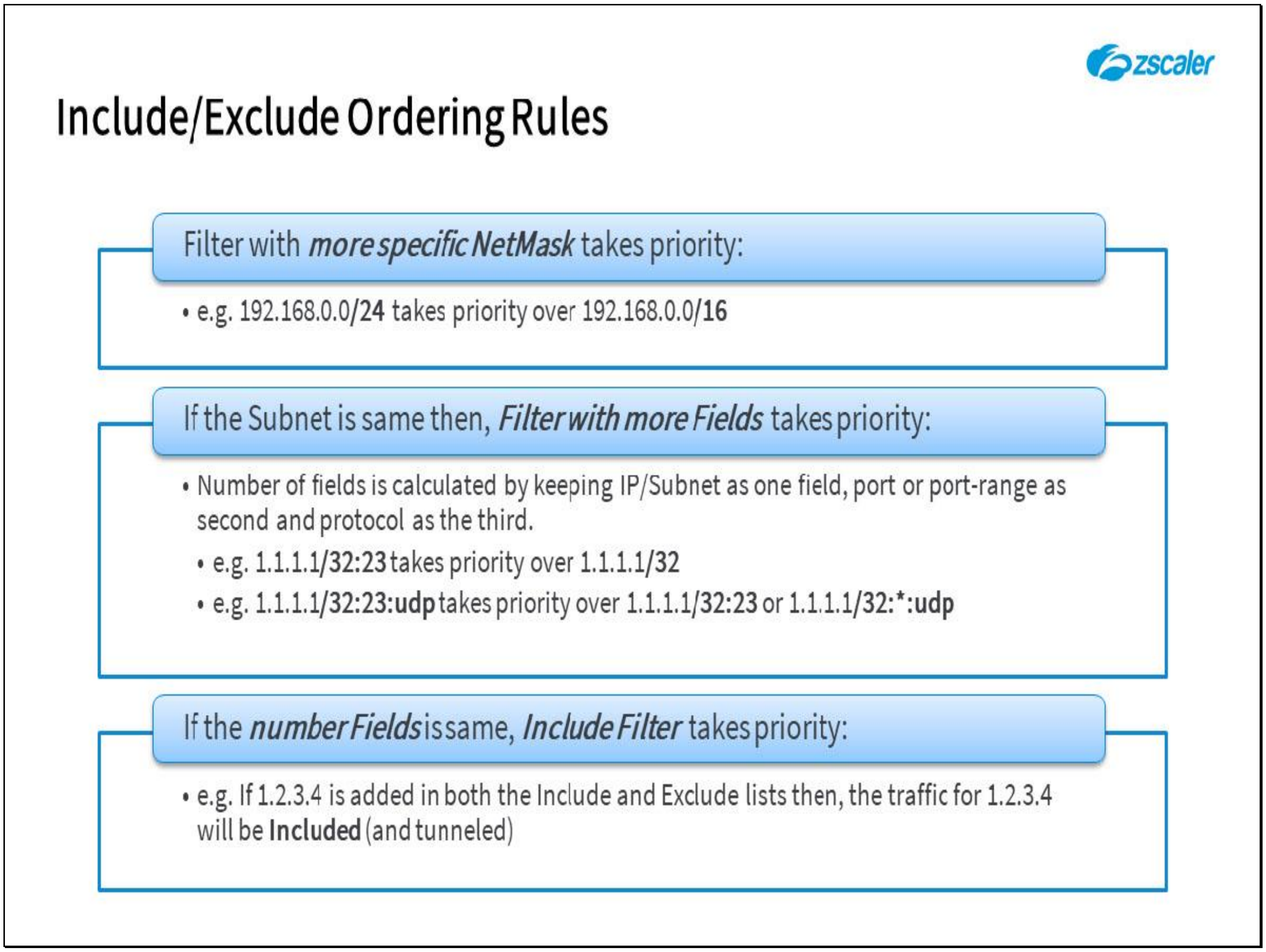
- Number of fields is calculated by keeping IP/Subnet as one field, port or port-range as second and protocol as the third.
  - e.g. **1.1.1.1/32:23** takes priority over **1.1.1.1/32**
  - e.g. **1.1.1.1/32:23:udp** takes priority over **1.1.1.1/32:23** or **1.1.1.1/32:*:udp**

Slide notes

The next criterion Z App examines is the **number of fields** in the rules, and a **more specific** rule (meaning more fields) will win.

So an **Include** rule with **Network** / **Netmask** / **Port** / **Protocol** will be applied over a rule that only defines **Network** / **Netmask** / **Port**;

Slide 37 - Include/Exclude Ordering Rules



# Include/Exclude Ordering Rules

Filter with ***more specific NetMask*** takes priority:

- e.g. 192.168.0.0/**24** takes priority over 192.168.0.0/**16**

If the Subnet is same then, ***Filter with more Fields*** takes priority:

- Number of fields is calculated by keeping IP/Subnet as one field, port or port-range as second and protocol as the third.
  - e.g. **1.1.1.1/32:23** takes priority over **1.1.1.1/32**
  - e.g. **1.1.1.1/32:23:udp** takes priority over **1.1.1.1/32:23** or **1.1.1.1/32:*:udp**

If the ***number Fields*** is same, ***Include Filter*** takes priority:

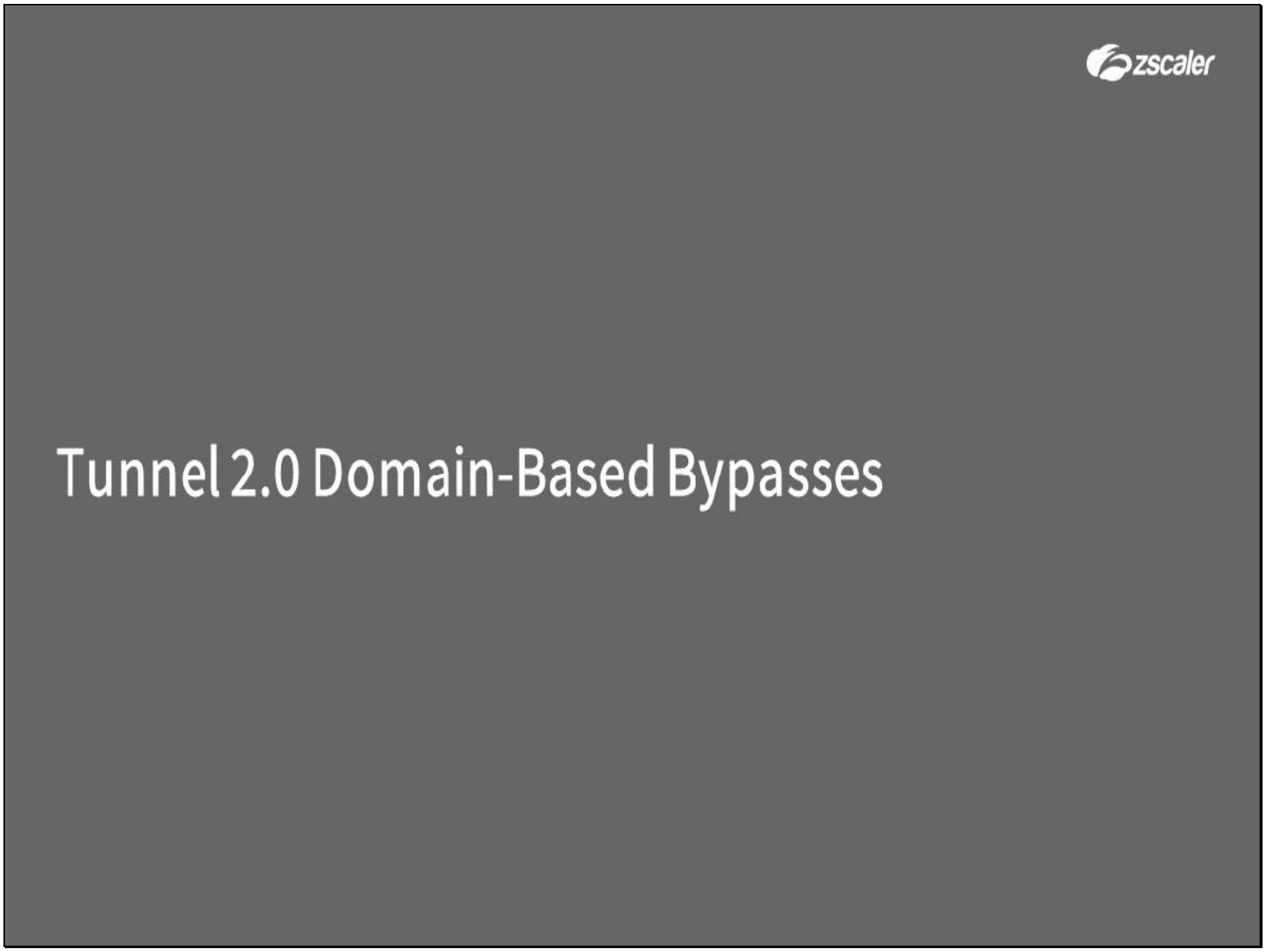
- e.g. If 1.2.3.4 is added in both the Include and Exclude lists then, the traffic for 1.2.3.4 will be **Included** (and tunneled)

Slide notes

Finally, if the number of fields is the same, an **Include** rule will always win.

So if the rule **1.2.3.4/24:19:TCP** is added as BOTH an **Include** and **Exclude** rule, this traffic will be **included** (and sent into a tunnel).

Slide 38 - Tunnel 2.0 Domain-Based Bypasses



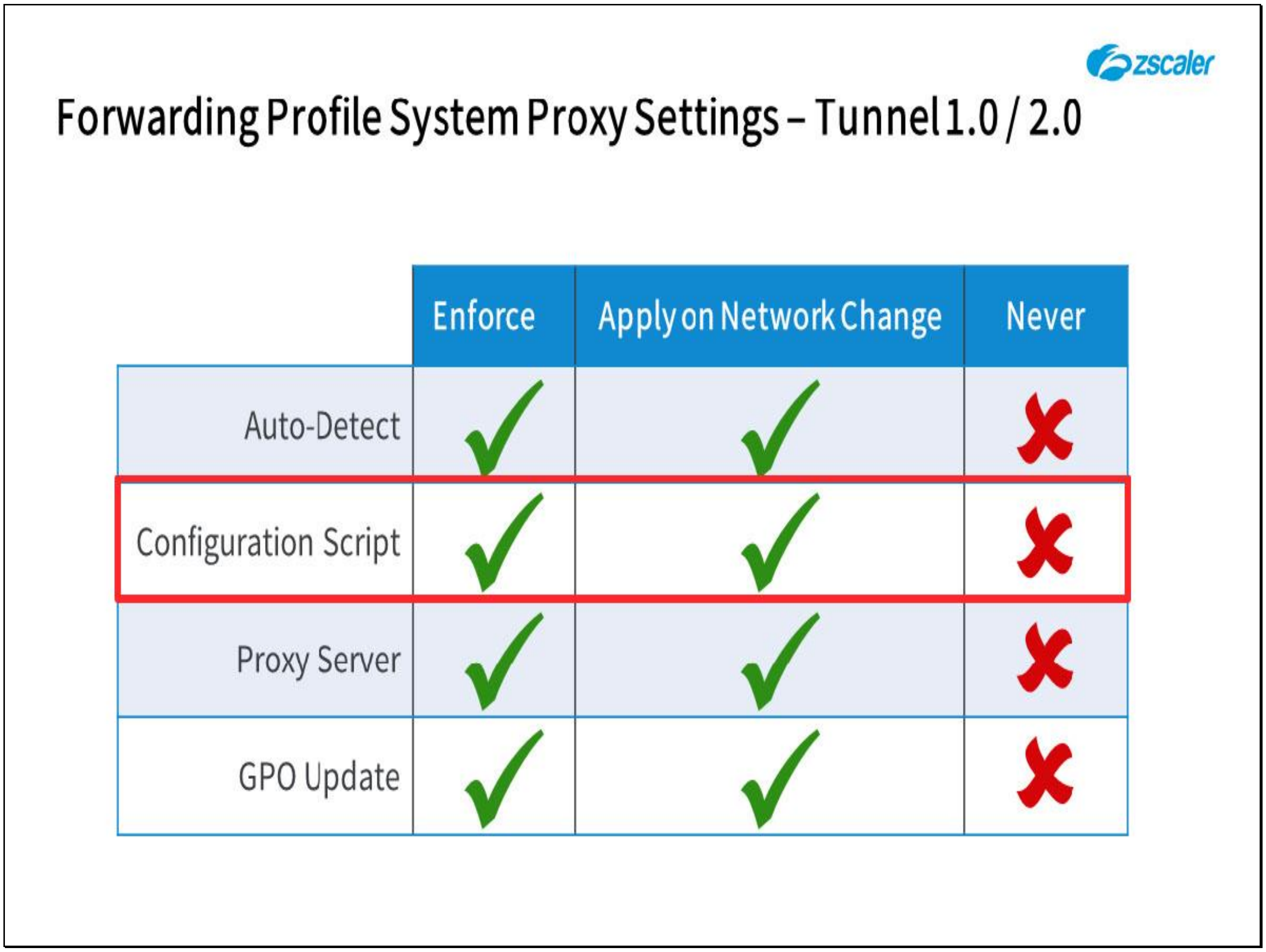
zscaler

Tunnel 2.0 Domain-Based Bypasses

Slide notes

The next topic that we will cover is a look at how to define **Domain-based** bypasses for a **Tunnel 2.0** system.

Slide 39 - Tunnel 1.0 / 2.0



zscaler

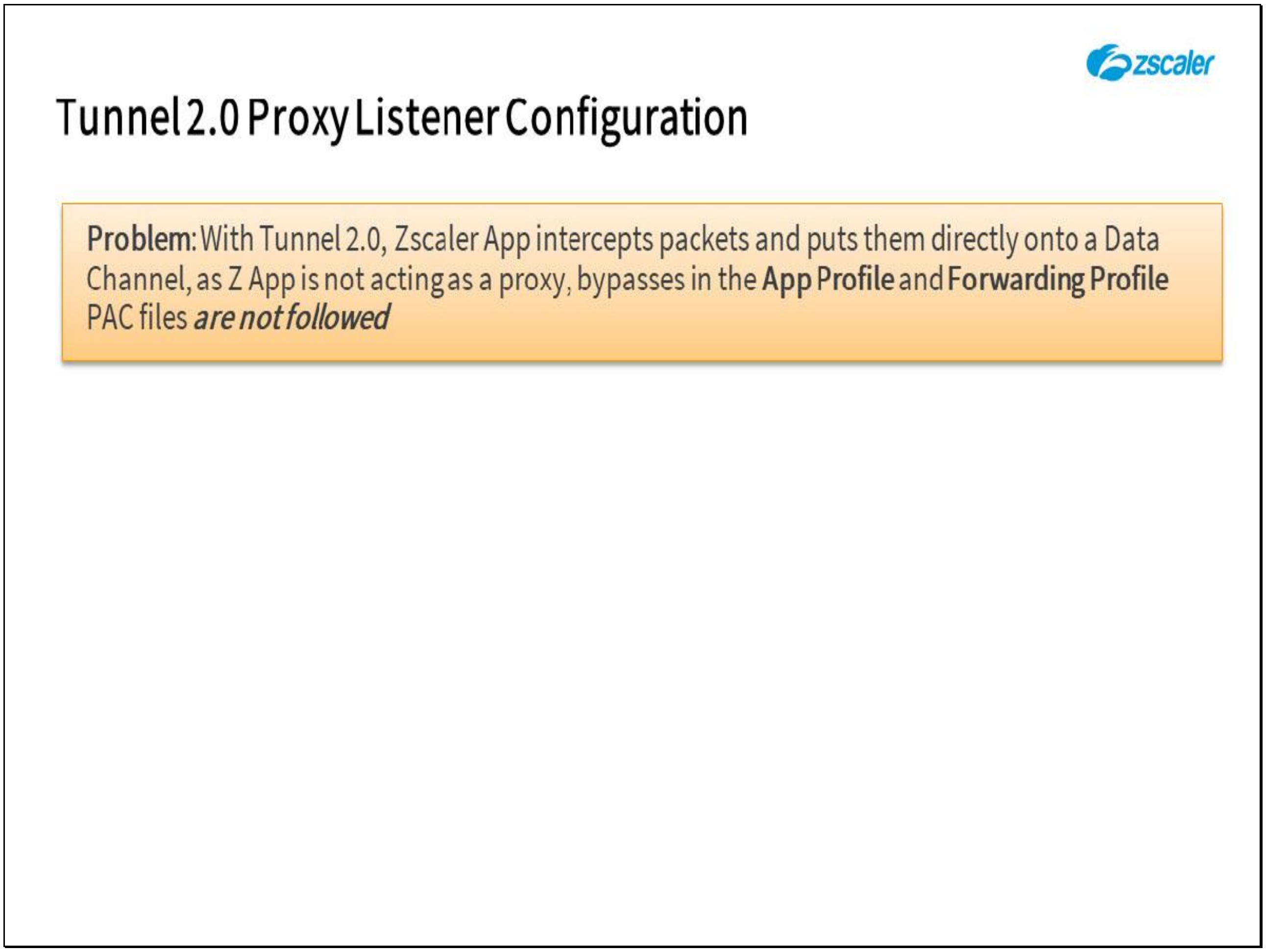
## Forwarding Profile System Proxy Settings – Tunnel 1.0 / 2.0

| | Enforce | Apply on Network Change | Never |
|---|---|---|---|
| Auto-Detect | ✓ | ✓ | ✗ |
| Configuration Script | ✓ | ✓ | ✗ |
| Proxy Server | ✓ | ✓ | ✗ |
| GPO Update | ✓ | ✓ | ✗ |

Slide notes

Just as a reminder, these are the **Forwarding Profile** system proxy configuration options for both **Tunnel 1.0** and **Tunnel 2.0**. The most significant of these is probably the **Configuration Script** option, where you can specify the URL for a custom PAC file. This is the **Forwarding Profile** PAC file, which may contain bypass destinations when used with **Tunnel 1.0**.

Slide 40 - Tunnel 2.0 Proxy Listener Configuration



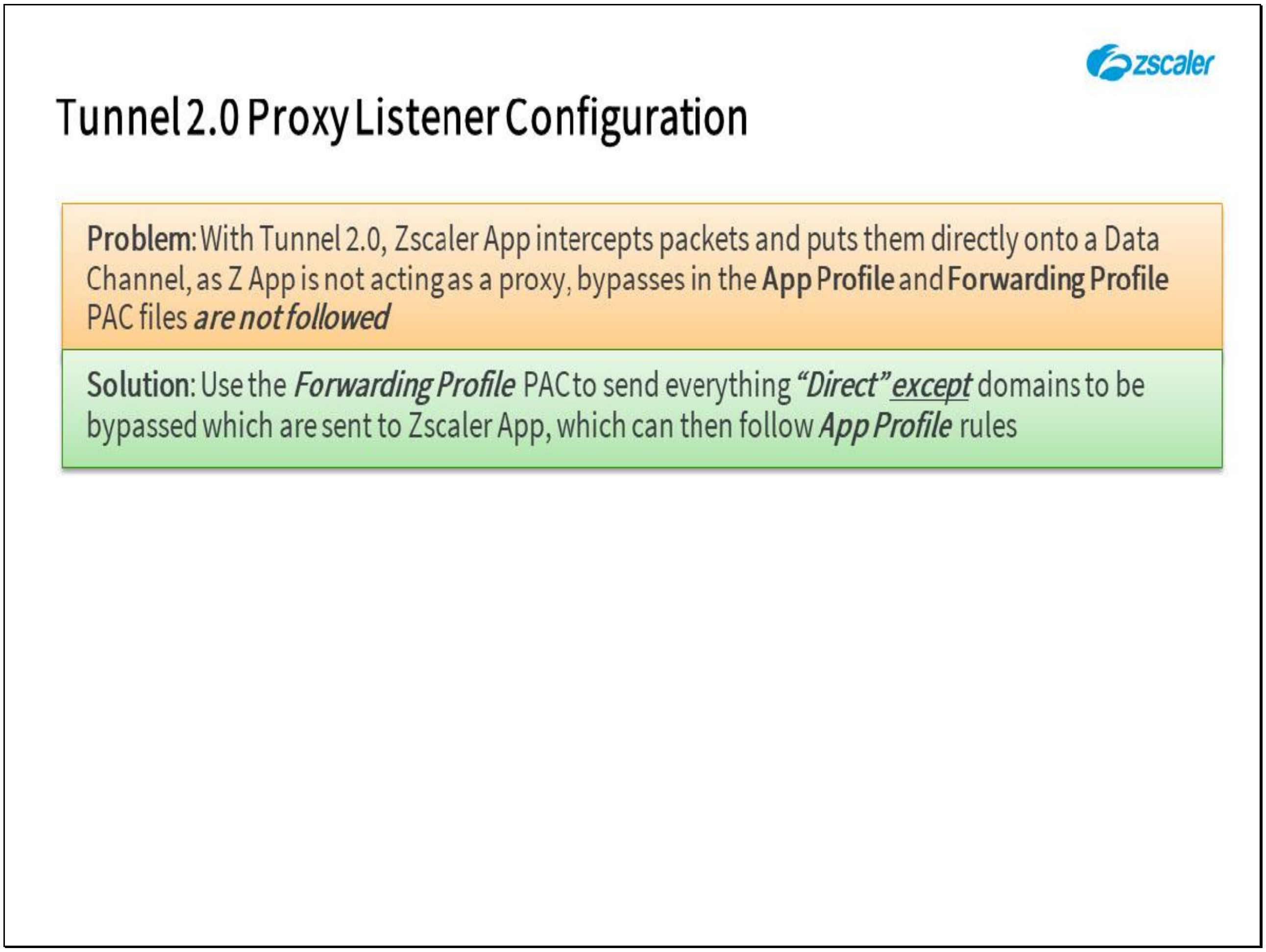
# Tunnel 2.0 Proxy Listener Configuration

**Problem:** With Tunnel 2.0, Zscaler App intercepts packets and puts them directly onto a Data Channel, as Z App is not acting as a proxy, bypasses in the **App Profile** and **Forwarding Profile** PAC files ***are not followed***

Slide notes

IP-based bypasses for **Tunnel 2.0** are pretty straight forward, however, when it comes to bypasses specified by **FQDN**, **Domain** or **URL** there is a problem. As **Tunnel 2.0** intercepts packets and puts them directly onto a data channel, Z App is not acting as a proxy and therefore has no visibility into domain-based addresses or URLs. As a consequence, any destinations specified in legacy **Tunnel 1.0** PAC files (whether from the **Forwarding Profile** or **App Profile** PAC files) will not be detected and hence will not be followed.

Slide 41 - Tunnel 2.0 Proxy Listener Configuration



## Tunnel 2.0 Proxy Listener Configuration

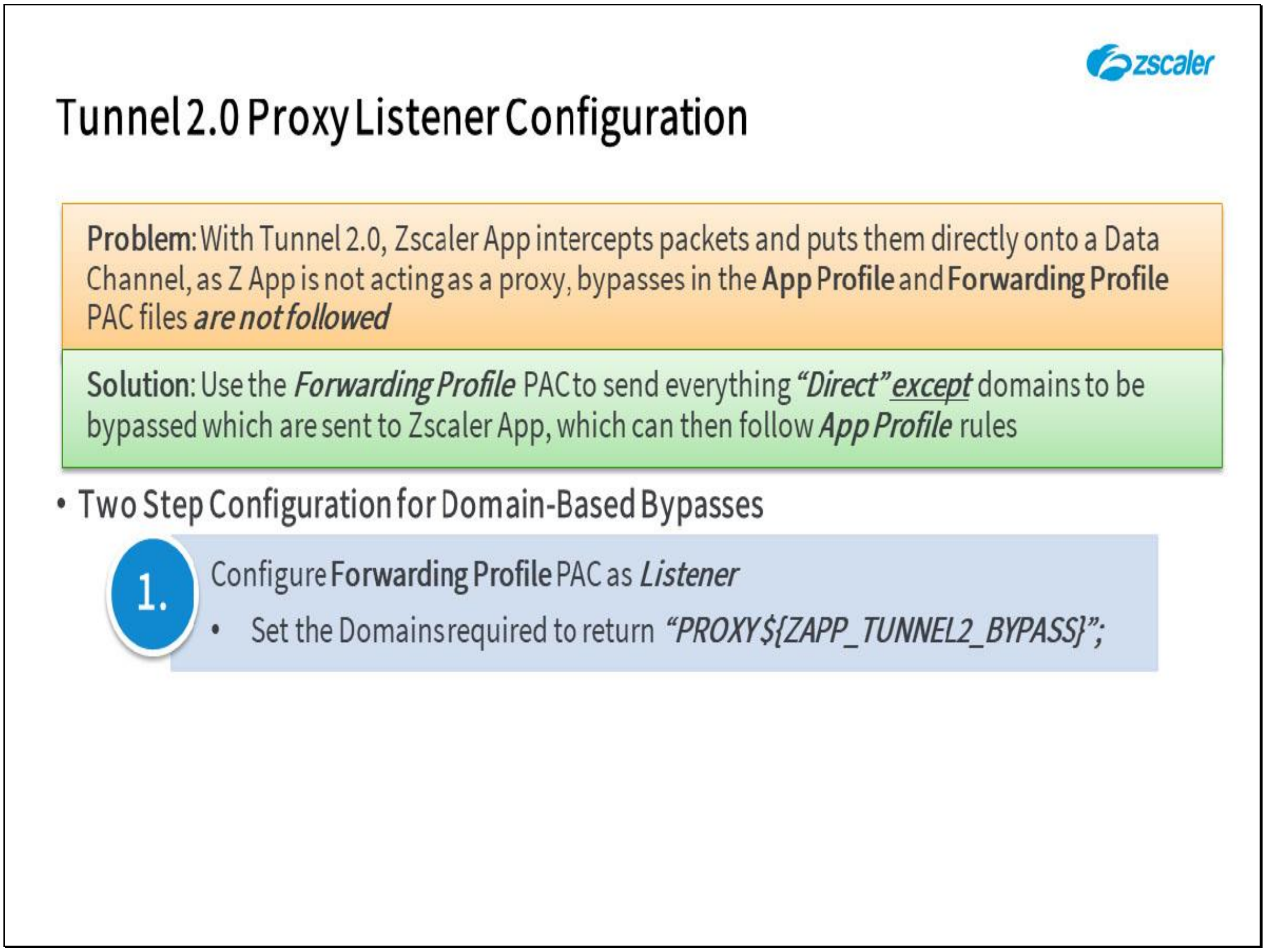
**Problem**: With Tunnel 2.0, Zscaler App intercepts packets and puts them directly onto a Data Channel, as Z App is not acting as a proxy, bypasses in the **App Profile** and **Forwarding Profile** PAC files ***are not followed***

**Solution**: Use the ***Forwarding Profile*** PAC to send everything ***"Direct" except*** domains to be bypassed which are sent to Zscaler App, which can then follow ***App Profile*** rules

Slide notes

The solution to this problem is to use the **proxy loopback configuration** in the **Forwarding Profile** PAC file, which the App DOES listen on. **FQDNs**, **Domains**, or even **URLs** to be bypassed can be sent to the loopback listening proxy, at which point they are explicitly proxied into Z App. The App is consequently able to see the **FQDNs**/**Domains** and is able to match them to proxy statements added to the **App Profile** PAC file.

Slide 42 - Tunnel 2.0 Proxy Listener Configuration



## Tunnel 2.0 Proxy Listener Configuration

**Problem:** With Tunnel 2.0, Zscaler App intercepts packets and puts them directly onto a Data Channel, as Z App is not acting as a proxy, bypasses in the **App Profile** and **Forwarding Profile** PAC files ***are not followed***

**Solution:** Use the ***Forwarding Profile*** PAC to send everything ***"Direct" except*** domains to be bypassed which are sent to Zscaler App, which can then follow ***App Profile*** rules

- Two Step Configuration for Domain-Based Bypasses

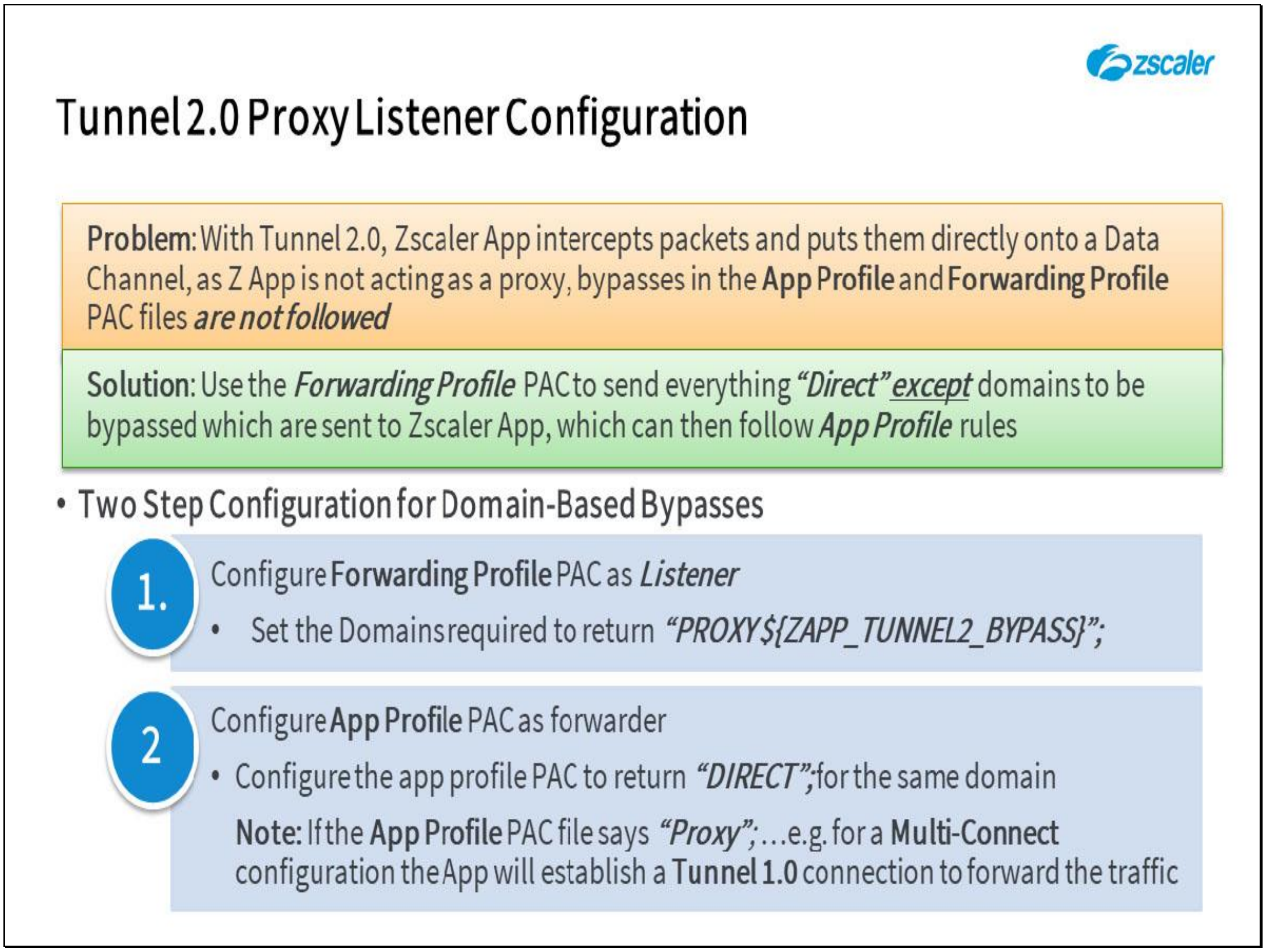
1. Configure **Forwarding Profile** PAC as *Listener*
   - Set the Domains required to return *"PROXY ${ZAPP_TUNNEL2_BYPASS}";*

Slide notes

As a result, the support for **Domain-based** bypasses is a two-step configuration:

1. **Step 1** - In the **Forwarding Profile** PAC file, send any **FQDNs**, **Domains** or **URLs** that you wish the App to bypass to the loopback listening proxy, preferably using the Zscaler macro **${ZAPP_TUNNEL2_BYPASS}**. Everything else should end up going **DIRECT**.

Slide 43 - Tunnel 2.0 Proxy Listener Configuration



## Tunnel 2.0 Proxy Listener Configuration

**Problem:** With Tunnel 2.0, Zscaler App intercepts packets and puts them directly onto a Data Channel, as Z App is not acting as a proxy, bypasses in the **App Profile** and **Forwarding Profile** PAC files ***are not followed***

**Solution:** Use the ***Forwarding Profile*** PAC to send everything ***"Direct"*** ***except*** domains to be bypassed which are sent to Zscaler App, which can then follow ***App Profile*** rules

- Two Step Configuration for Domain-Based Bypasses

1. Configure **Forwarding Profile** PAC as ***Listener***
   - Set the Domains required to return *"PROXY ${ZAPP_TUNNEL2_BYPASS}";*

2. Configure **App Profile** PAC as forwarder
   - Configure the app profile PAC to return *"DIRECT";* for the same domain

   **Note:** If the **App Profile** PAC file says *"Proxy";*…e.g. for a **Multi-Connect** configuration the App will establish a **Tunnel 1.0** connection to forward the traffic

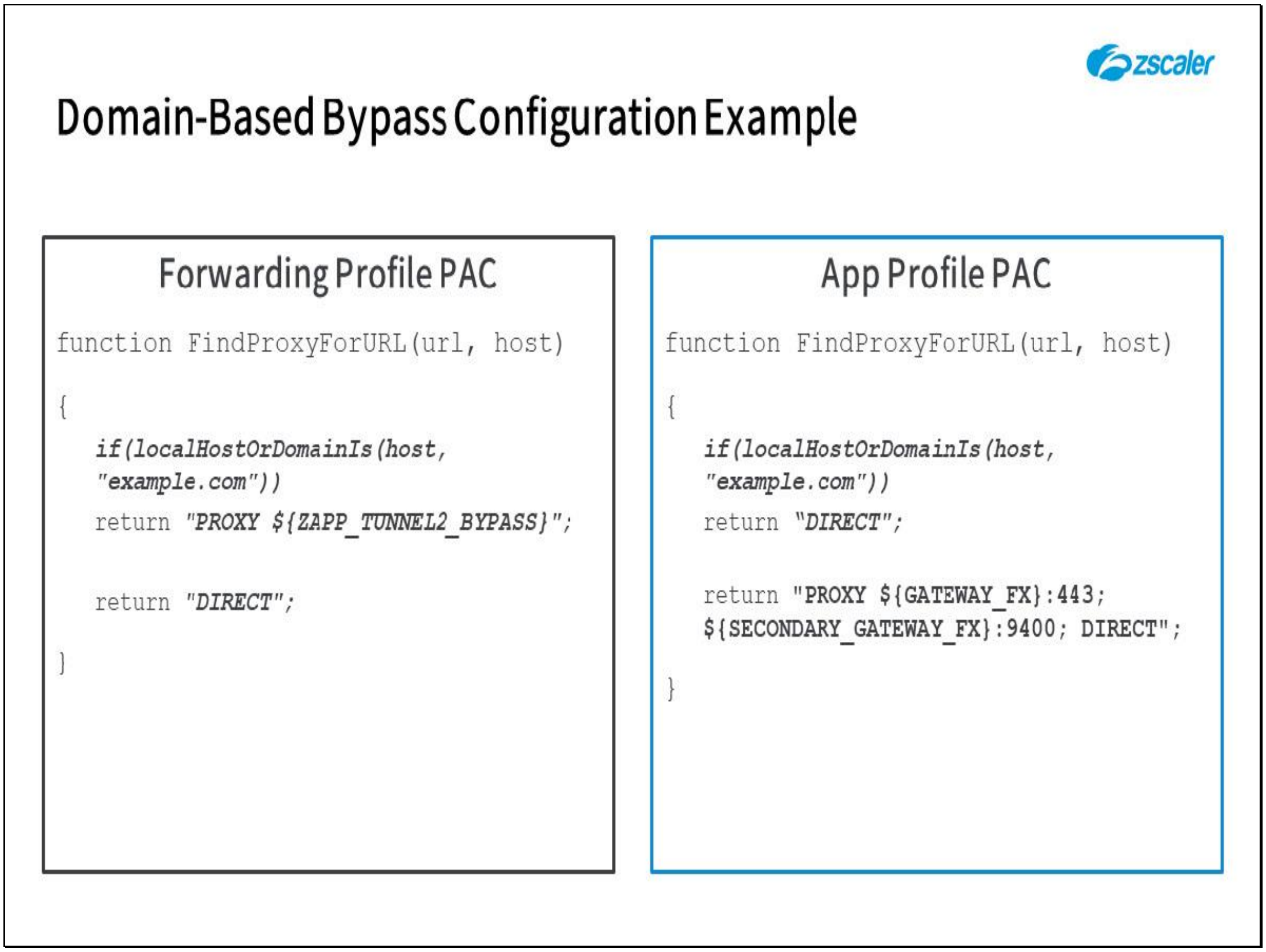
Slide notes

2. **Step 2** - Bypass the same **FQDNs** or **Domains** in the **App Profile** PAC file by sending them **DIRECT**.

   Note that the **App Profile** PAC file does not support **URL** based rules, however you can substitute a **FQDN** rule for the web server to be bypassed.

   Also note, that if the **App Profile** PAC file includes a **PROXY** statement for a destination (e.g. as part of a **Multi-Connect** configuration), tunnels to those destinations will use the **Tunnel 1.0** method (lightweight **HTTP CONNECT** tunnels).

Slide 44 - Domain-Based Bypass Configuration



# Domain-Based Bypass Configuration Example

## Forwarding Profile PAC

```
function FindProxyForURL(url, host)

{

   if(localHostOrDomainIs(host,
   "example.com"))
   return "PROXY ${ZAPP_TUNNEL2_BYPASS}";


   return "DIRECT";

}
```

## App Profile PAC

```
function FindProxyForURL(url, host)

{

   if(localHostOrDomainIs(host,
   "example.com"))
   return "DIRECT";


   return "PROXY ${GATEWAY_FX}:443;
   ${SECONDARY_GATEWAY_FX}:9400; DIRECT";

}
```

Slide notes

To illustrate this 2-step process, here is a simple example of the PAC file configurations required for both the **Forwarding Profile** and **App Profile** PAC files to bypass the **example.com** domain.

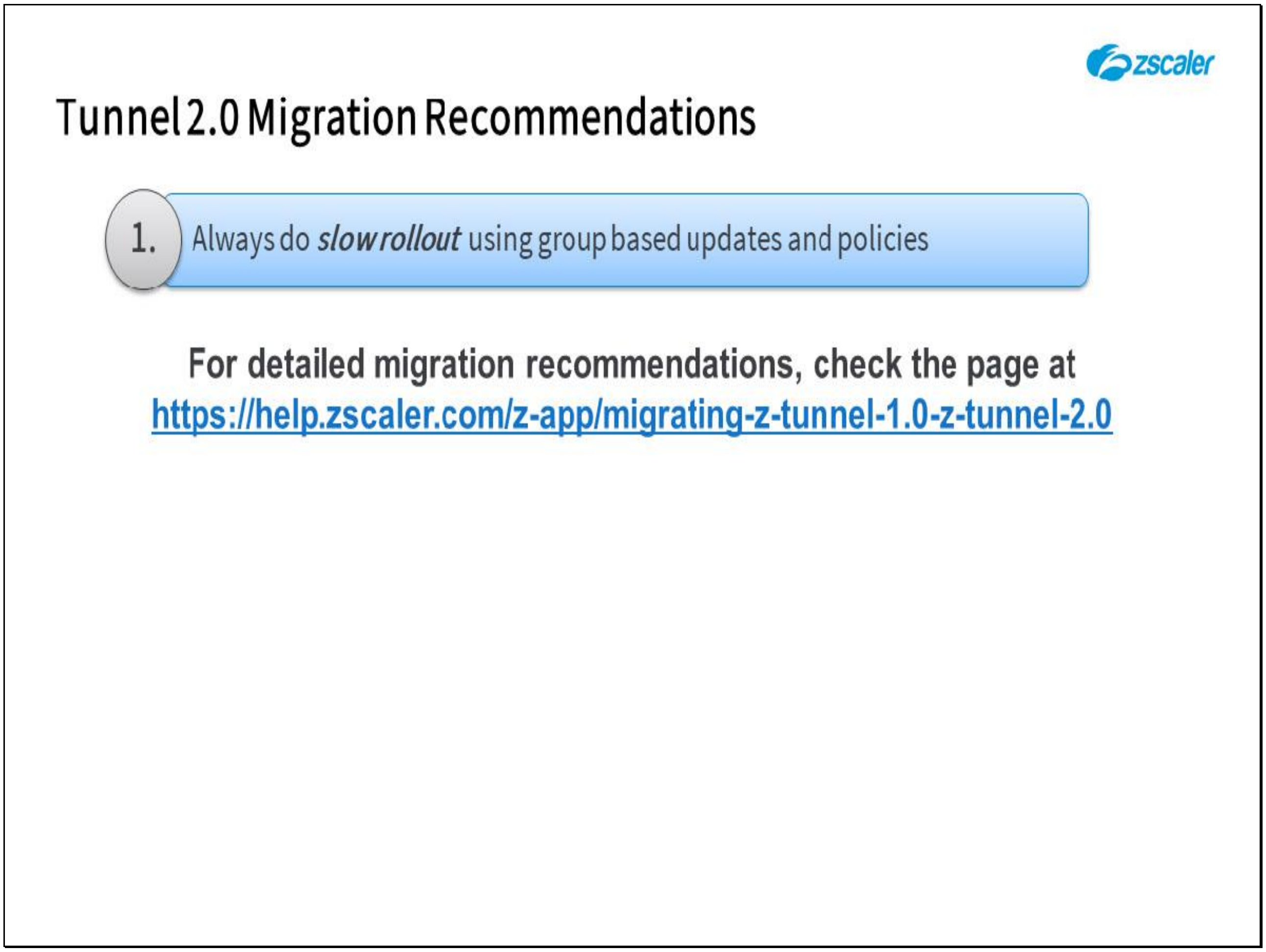
## Slide 45 - Tunnel 1.0 Migration

Tunnel 1.0 Migration

### Slide notes

The final topic that we will cover is a look at the process of migrating users from the **Tunnel 1.0** method to **Tunnel 2.0**.

Slide 46 - Tunnel 2.0 Migration Recommendations



Tunnel 2.0 Migration Recommendations

1. Always do ***slow rollout*** using group based updates and policies

**For detailed migration recommendations, check the page at https://help.zscaler.com/z-app/migrating-z-tunnel-1.0-z-tunnel-2.0**

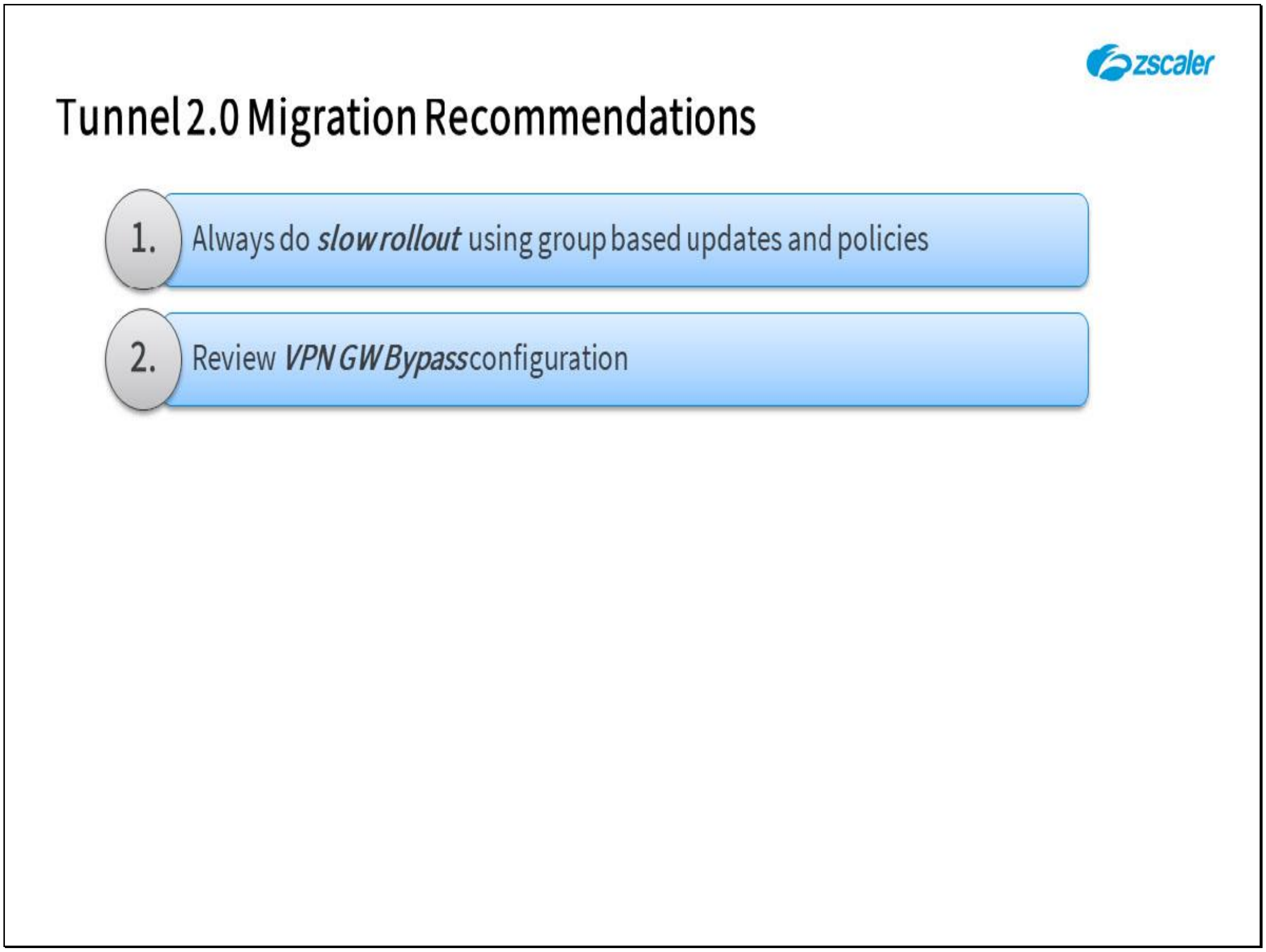
Slide notes

In a 'green field' **Tunnel 2.0** deployment, where you go straight to the **Tunnel 2.0** mechanisms for adding **Inclusions**, **Exclusions** and destination bypasses, it is no great problem to specify the configurations that you need. However if you have to migrate users from a legacy **Tunnel 1.0** deployment into the new **Tunnel 2.0** methods, things can start to get complicated!

Check the support page at https://help.zscaler.com/z-app/migrating-z-tunnel-1.0-z-tunnel-2.0 for our recommendations on managing the migration from **Tunnel 1.0** to **Tunnel 2.0**.

- First and foremost, we recommend that you trial the upgrade process on a control group, to capture any unintended consequences in a controlled environment. Create a user group in the ZIA Hosted Database and assign a new **App Profile** to this group for **Tunnel 2.0** settings. Create a new **Forwarding Profile** with **Tunnel 2.0** enabled, select it in the new **App Profile** and ensure that the group members are upgraded to at least Zscaler App **v2.0.1**.

Slide 47 - Tunnel 2.0 Migration Recommendations



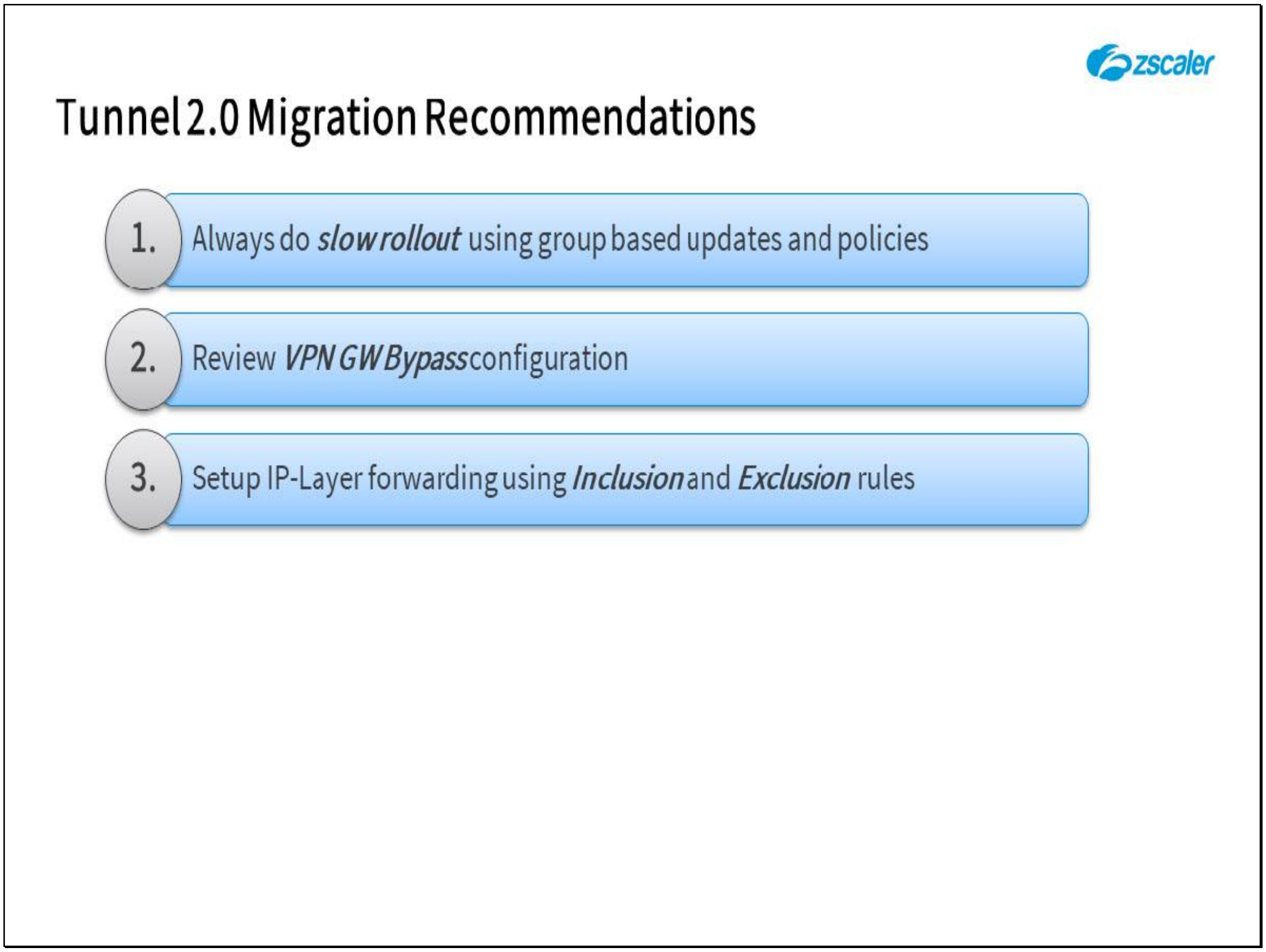
# Tunnel 2.0 Migration Recommendations

1. Always do ***slow rollout*** using group based updates and policies
2. Review ***VPN GW Bypass*** configuration

Slide notes

- Review any existing **VPN Gateway Bypasses** in your original **App Profile** and if any are still required, transfer them to the new **Tunnel 2.0 App Profile** that you created. Other **App Profile** settings can be left at defaults for the time being.

Slide 48 - Tunnel 2.0 Migration Recommendations



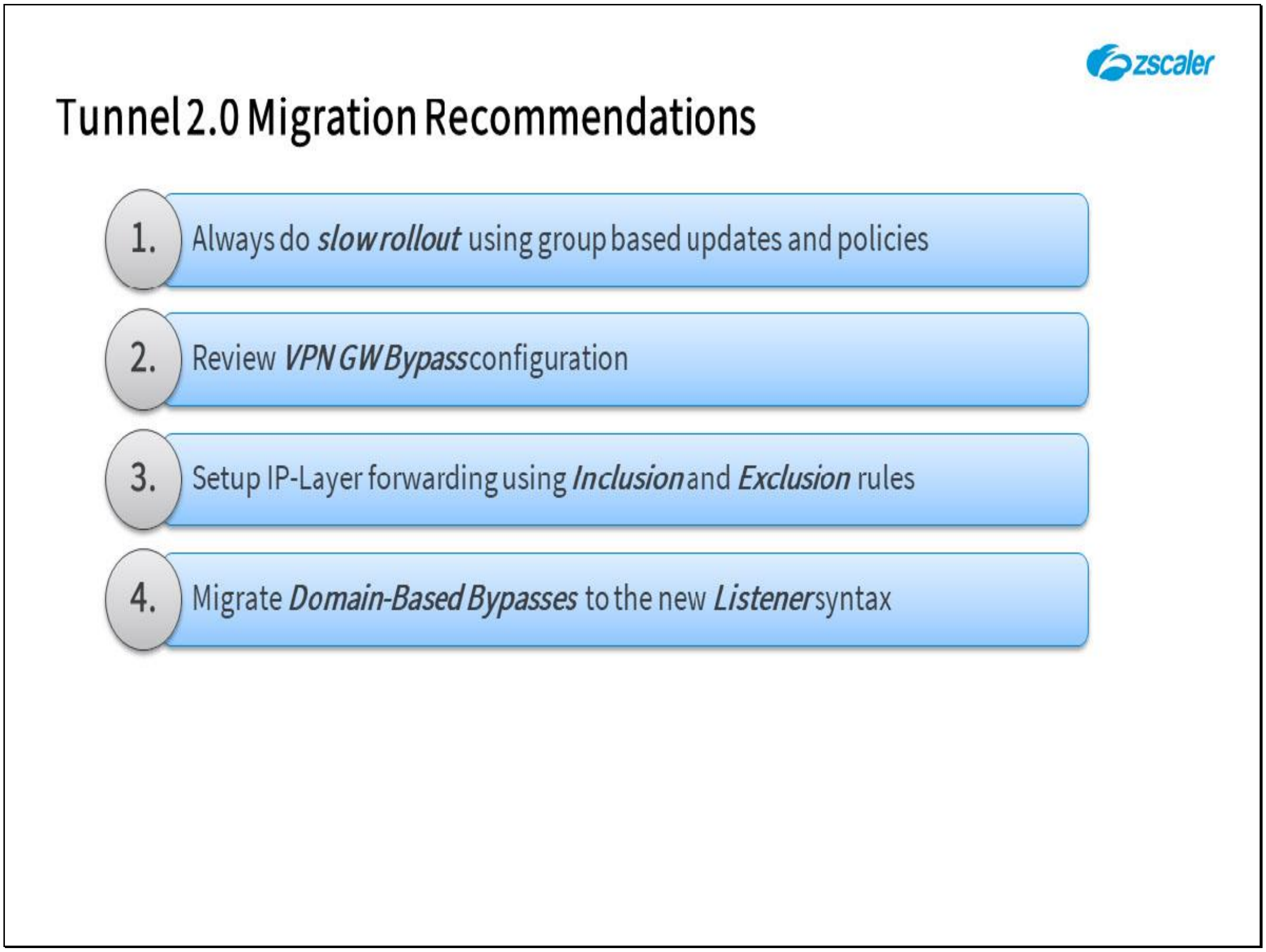
## Tunnel 2.0 Migration Recommendations

1. Always do ***slow rollout*** using group based updates and policies
2. Review *VPN GW Bypass* configuration
3. Setup IP-Layer forwarding using *Inclusion* and *Exclusion* rules

Slide notes

- You need to add your **IP-based Inclusions** and **Exclusions** to the **Tunnel 2.0 App Profile** configuration. You may need to refer back to the network address and range bypasses in your original **App Profile** PAC file.

Slide 49 - Tunnel 2.0 Migration Recommendations



## Tunnel 2.0 Migration Recommendations

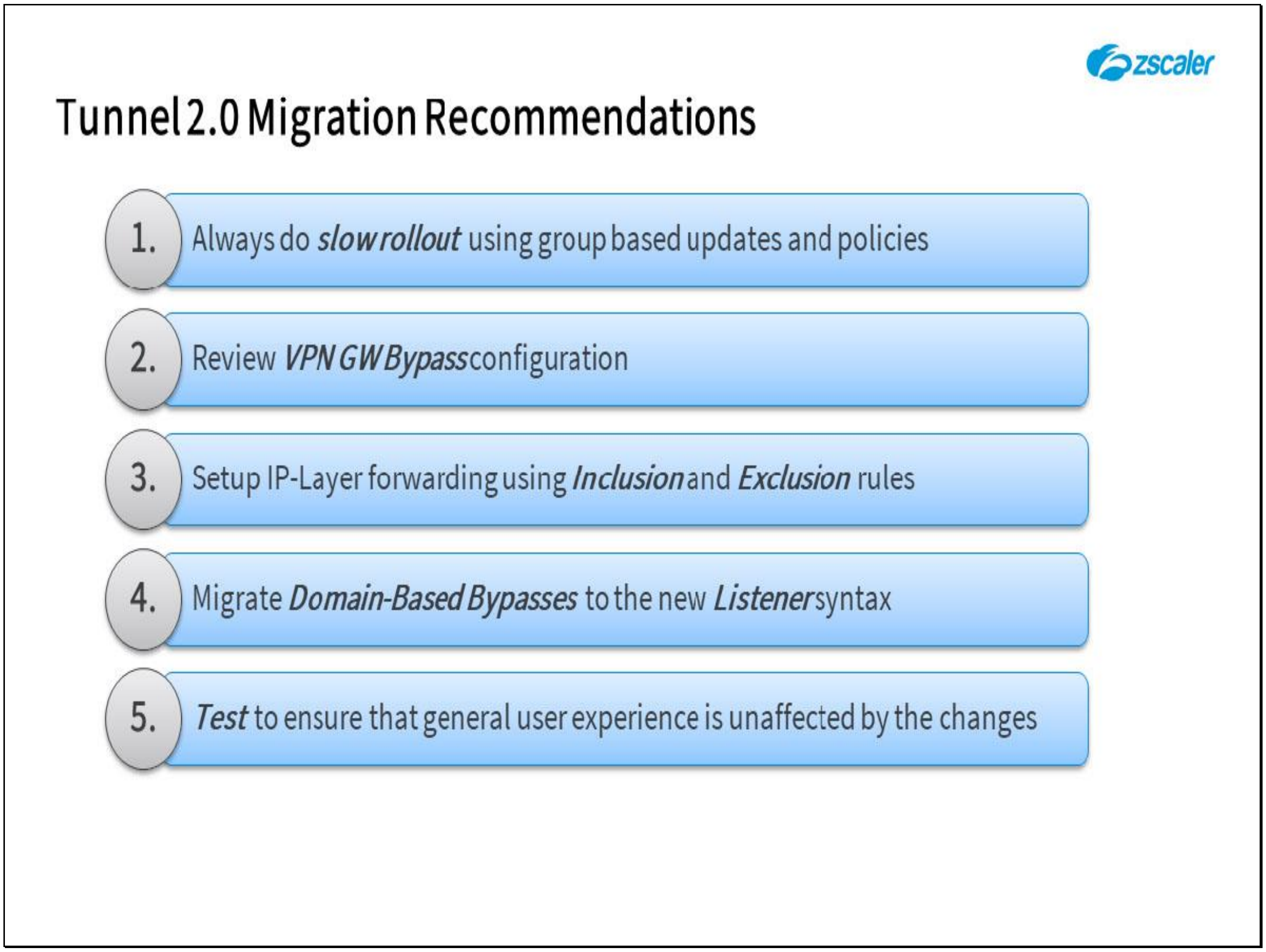
1. Always do ***slow rollout*** using group based updates and policies
2. Review *VPN GW Bypass* configuration
3. Setup IP-Layer forwarding using *Inclusion* and *Exclusion* rules
4. Migrate *Domain-Based Bypasses* to the new *Listener* syntax

Slide notes

- This next one is the difficult bit; you will need to add any **Domain-based** bypasses or **URLs** from your original **App Profile** and/or **Forwarding Profile** PAC files to your **Tunnel 2.0** configuration.

  Remember, this is a 2-step operation and requires a loopback configuration in the **Forwarding Profile** PAC file (using the **${ZAPP_TUNNEL2_BYPASS}** macro), with a **DIRECT** statement for the same hosts/domains in the **App Profile** PAC file.

Slide 50 - Tunnel 2.0 Migration Recommendations



zscaler

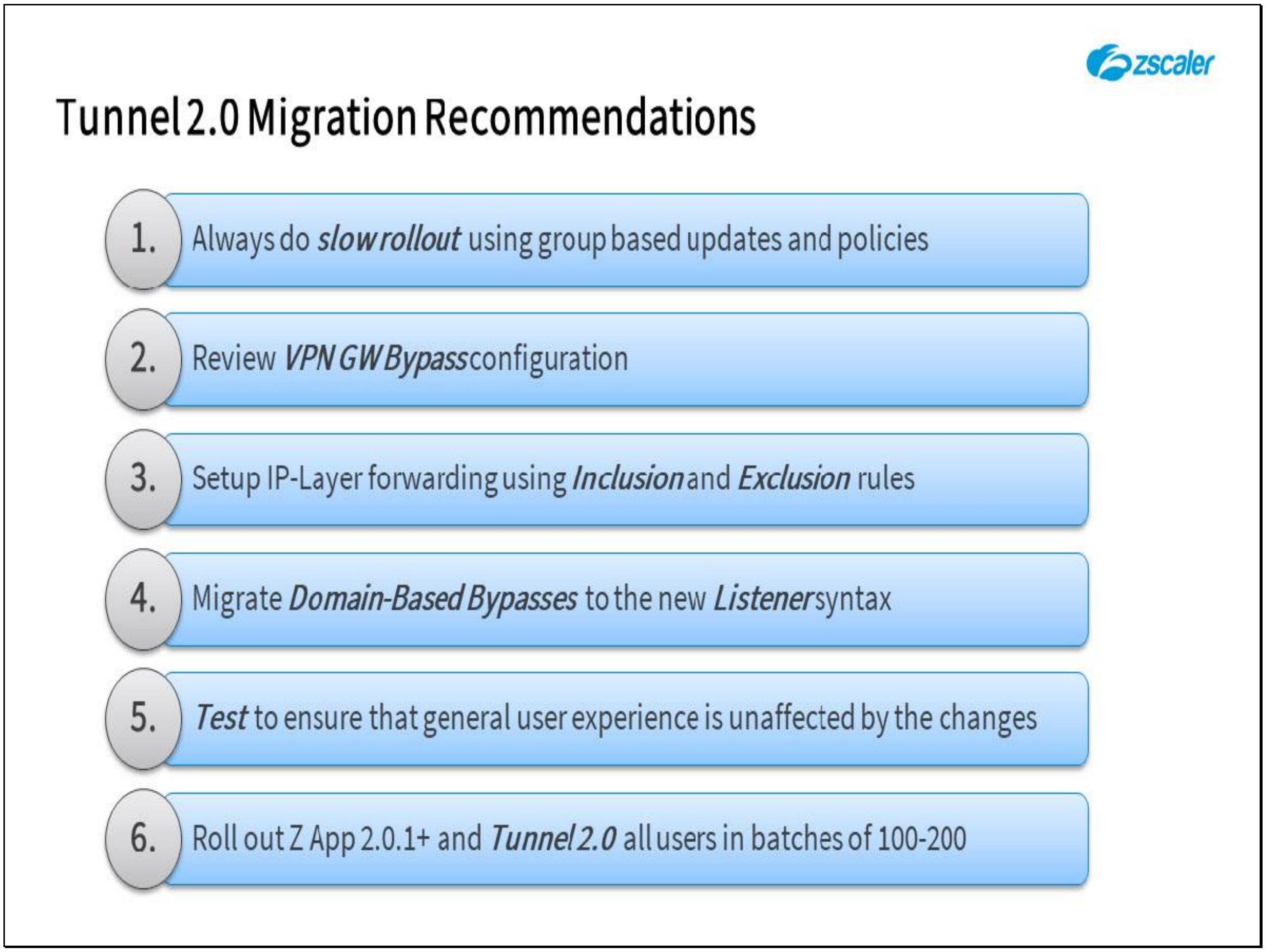
# Tunnel 2.0 Migration Recommendations

1. Always do ***slow rollout*** using group based updates and policies
2. Review *VPN GW Bypass* configuration
3. Setup IP-Layer forwarding using *Inclusion* and *Exclusion* rules
4. Migrate *Domain-Based Bypasses* to the new *Listener* syntax
5. *Test* to ensure that general user experience is unaffected by the changes

Slide notes

- The next step is to test thoroughly, to ensure that the general user experience is unaffected by these changes. As part of this process, you might like to:
    - Identify the top business applications that your organization uses;
    - Test access to these applications with the test user group;
    - Get user feedback on any issues they experience and adjust the configurations as necessary.

Slide 51 - Tunnel 2.0 Migration Recommendations



zscaler

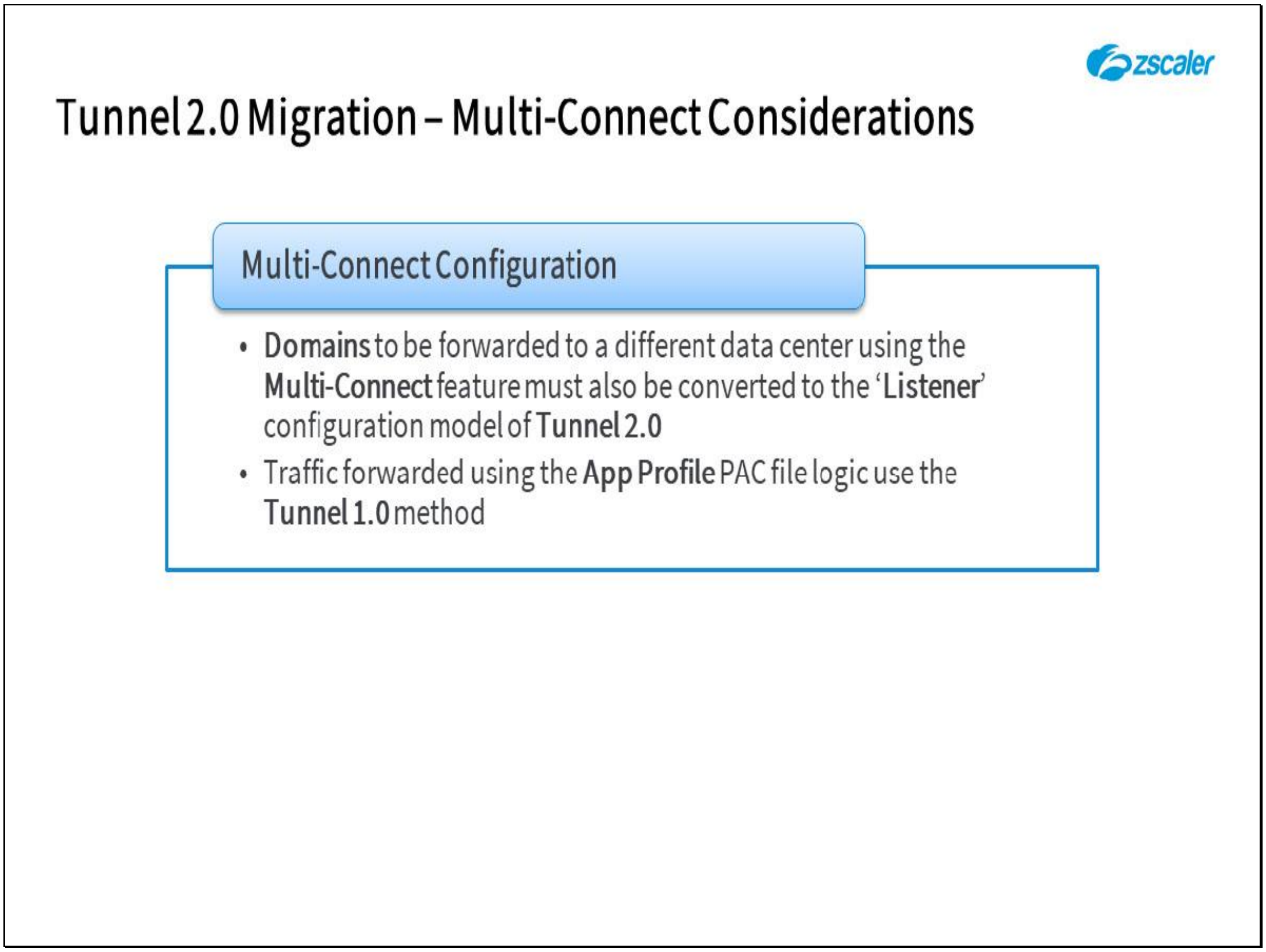
## Tunnel 2.0 Migration Recommendations

1. Always do ***slow rollout*** using group based updates and policies
2. Review ***VPN GW Bypass*** configuration
3. Setup IP-Layer forwarding using ***Inclusion*** and ***Exclusion*** rules
4. Migrate ***Domain-Based Bypasses*** to the new ***Listener*** syntax
5. ***Test*** to ensure that general user experience is unaffected by the changes
6. Roll out Z App 2.0.1+ and ***Tunnel 2.0*** all users in batches of 100-200

Slide notes

- Finally, you can start a more general roll out of Zscaler App 2.0.1 and **Tunnel 2.0** functionality to the rest of your users, preferably in batches of 100-200 users.

Slide 52 - Review and Revision of Bypass Configurations Required



## Tunnel 2.0 Migration – Multi-Connect Considerations

### Multi-Connect Configuration

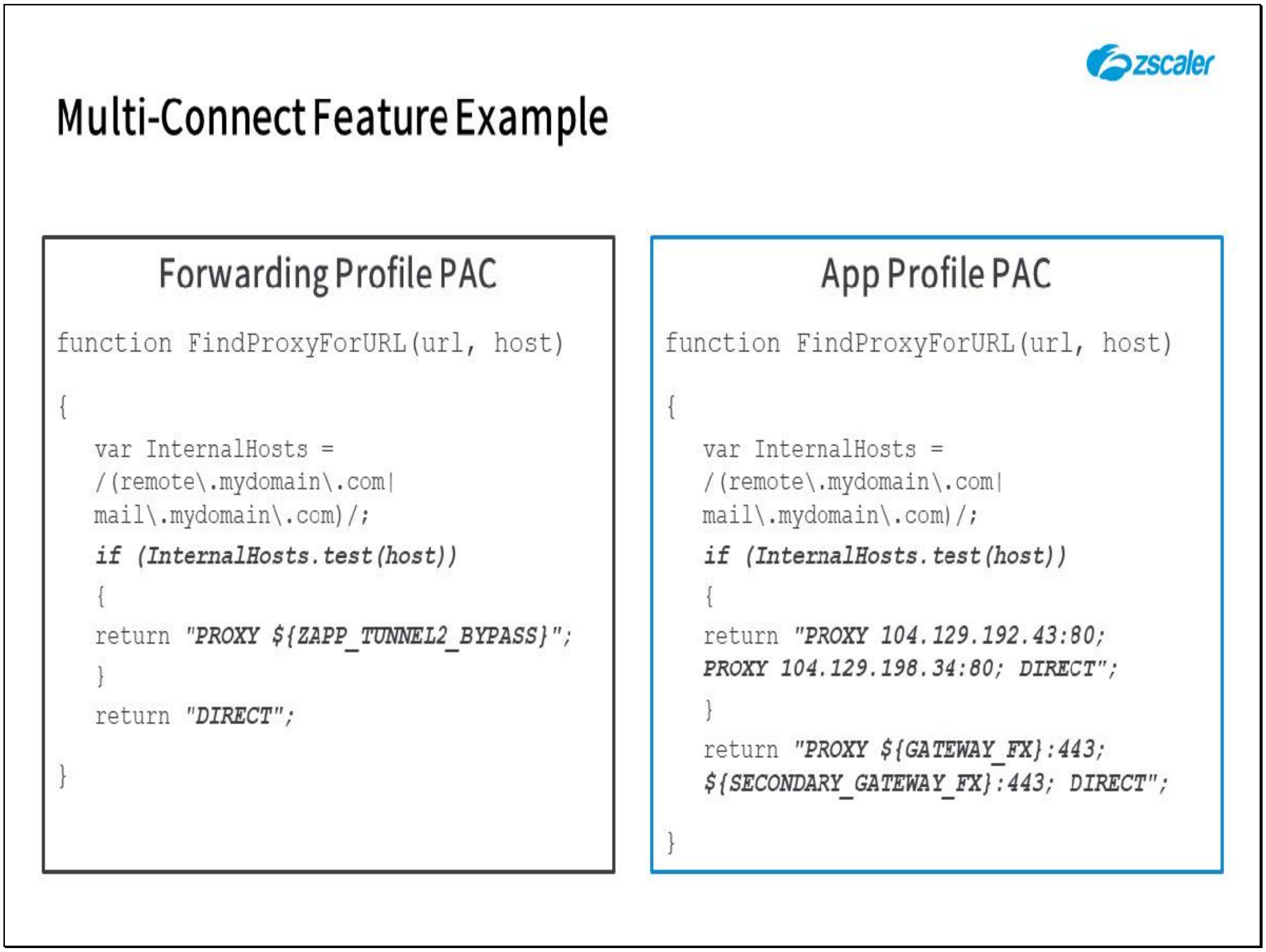
- **Domains** to be forwarded to a different data center using the **Multi-Connect** feature must also be converted to the '**Listener**' configuration model of **Tunnel 2.0**
- Traffic forwarded using the **App Profile** PAC file logic use the **Tunnel 1.0** method

### Slide notes

One final note about migrating any existing **Multi-Connect** configuration to the **Tunnel 2.0** model. This requires a similar configuration to the **Domain-based** bypasses, the domains to be forwarded to a different data center using the **Multi-Connect** feature must also be converted to the '**Listener**' configuration of a **Tunnel 2.0 Forwarding Profile**, with a **PROXY** statement to the required destinations in the **App Profile** PAC file.

Note that, traffic forwarded using the **App Profile** PAC file logic uses the **Tunnel 1.0** method, so the ZIA service will only be able to provide basic Internet Security scanning for those connections.

Slide 53 - Multi-Connect Feature



# Multi-Connect Feature Example

## Forwarding Profile PAC

```
function FindProxyForURL(url, host)

{
   var InternalHosts =
   /(remote\.mydomain\.com|
   mail\.mydomain\.com)/;
   if (InternalHosts.test(host))
   {
   return "PROXY ${ZAPP_TUNNEL2_BYPASS}";
   }
   return "DIRECT";

}
```

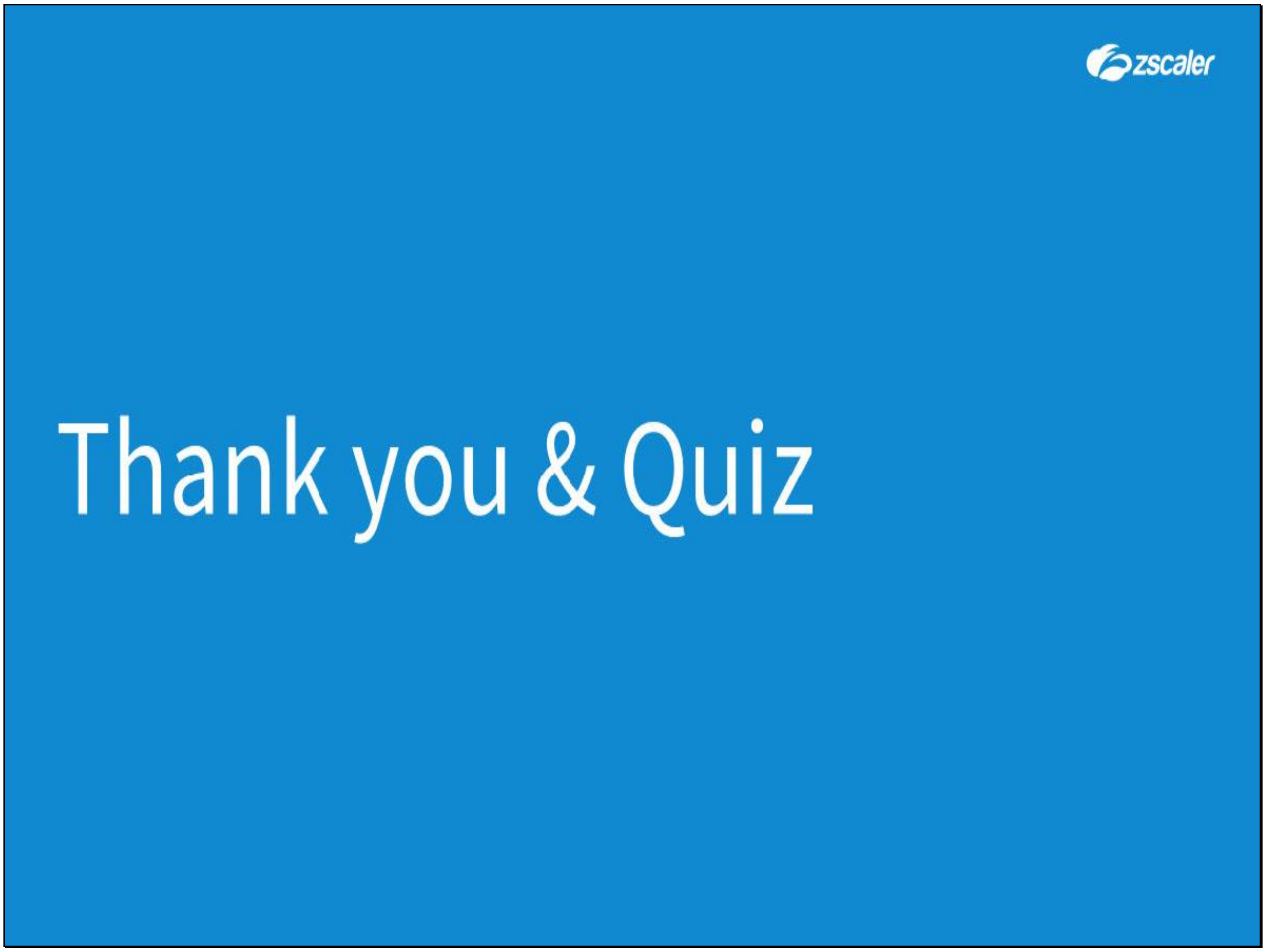
## App Profile PAC

```
function FindProxyForURL(url, host)

{
   var InternalHosts =
   /(remote\.mydomain\.com|
   mail\.mydomain\.com)/;
   if (InternalHosts.test(host))
   {
   return "PROXY 104.129.192.43:80;
   PROXY 104.129.198.34:80; DIRECT";
   }
   return "PROXY ${GATEWAY_FX}:443;
   ${SECONDARY_GATEWAY_FX}:443; DIRECT";

}
```

## Slide notes

To illustrate the 2-step configuration process for a **Multi-Connect** setup, here is a simple example of the PAC file configurations required for both the **Forwarding Profile** and **App Profile** PAC files to tunnel data for specific hosts to the ZEN on **104.129.198.34**.

Slide 54 - Thank you & Quiz



zscaler

Thank you & Quiz

Slide notes

Thank you for following this Zscaler training module, we hope this module has been useful to you and thank you for your time.

Click the X at top right to close this interface, then launch the quiz to test your knowledge of the material presented during this module.  You may retake the quiz as many times as necessary in order to pass.