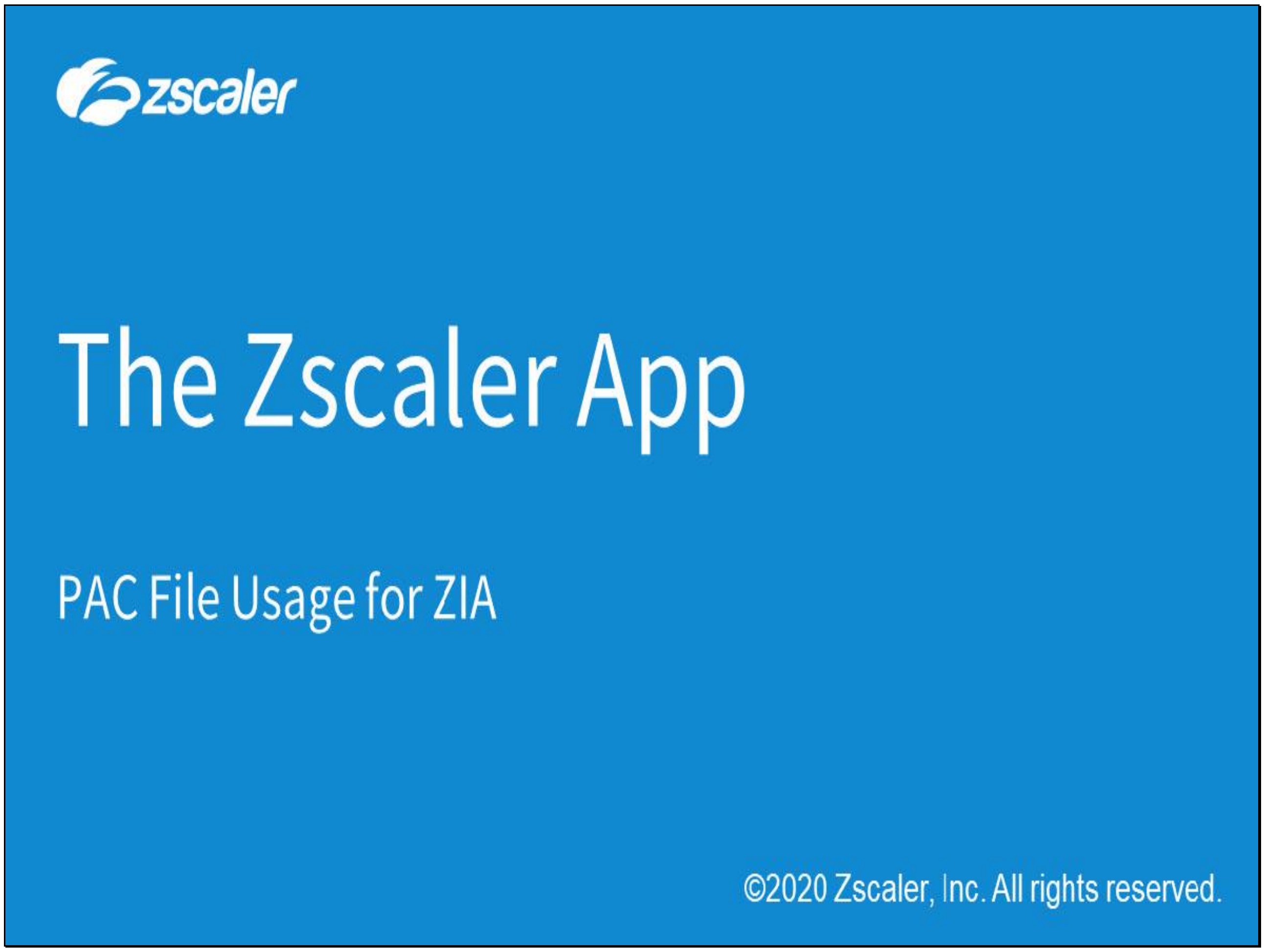
Slide 1 - The Zscaler App: PAC File Usage for ZIA



The Zscaler App

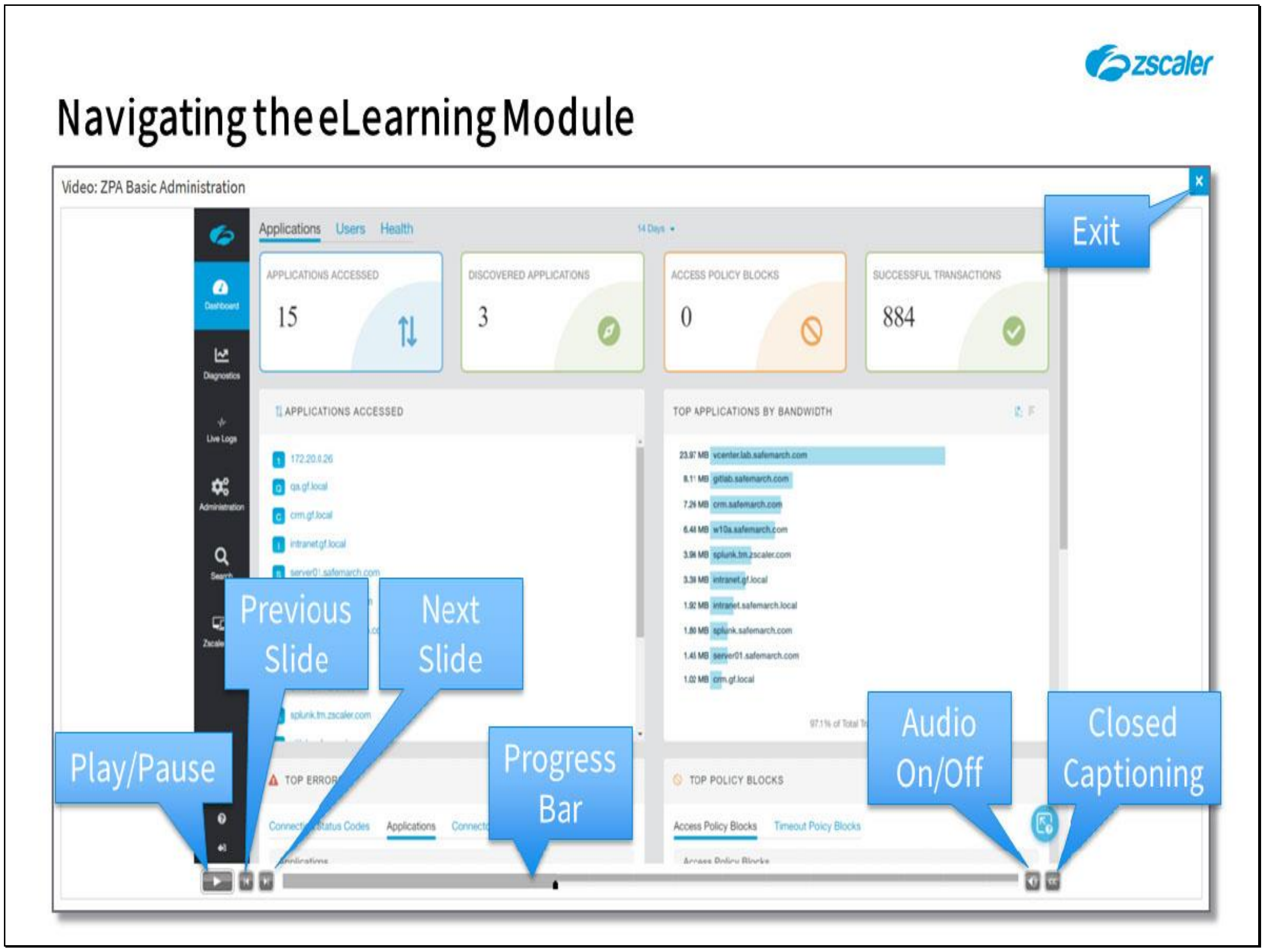
PAC File Usage for ZIA

©2020 Zscaler, Inc. All rights reserved.

Slide notes

Welcome to this training module for a look at how the Zscaler App uses PAC files to control connectivity to the ZIA service.

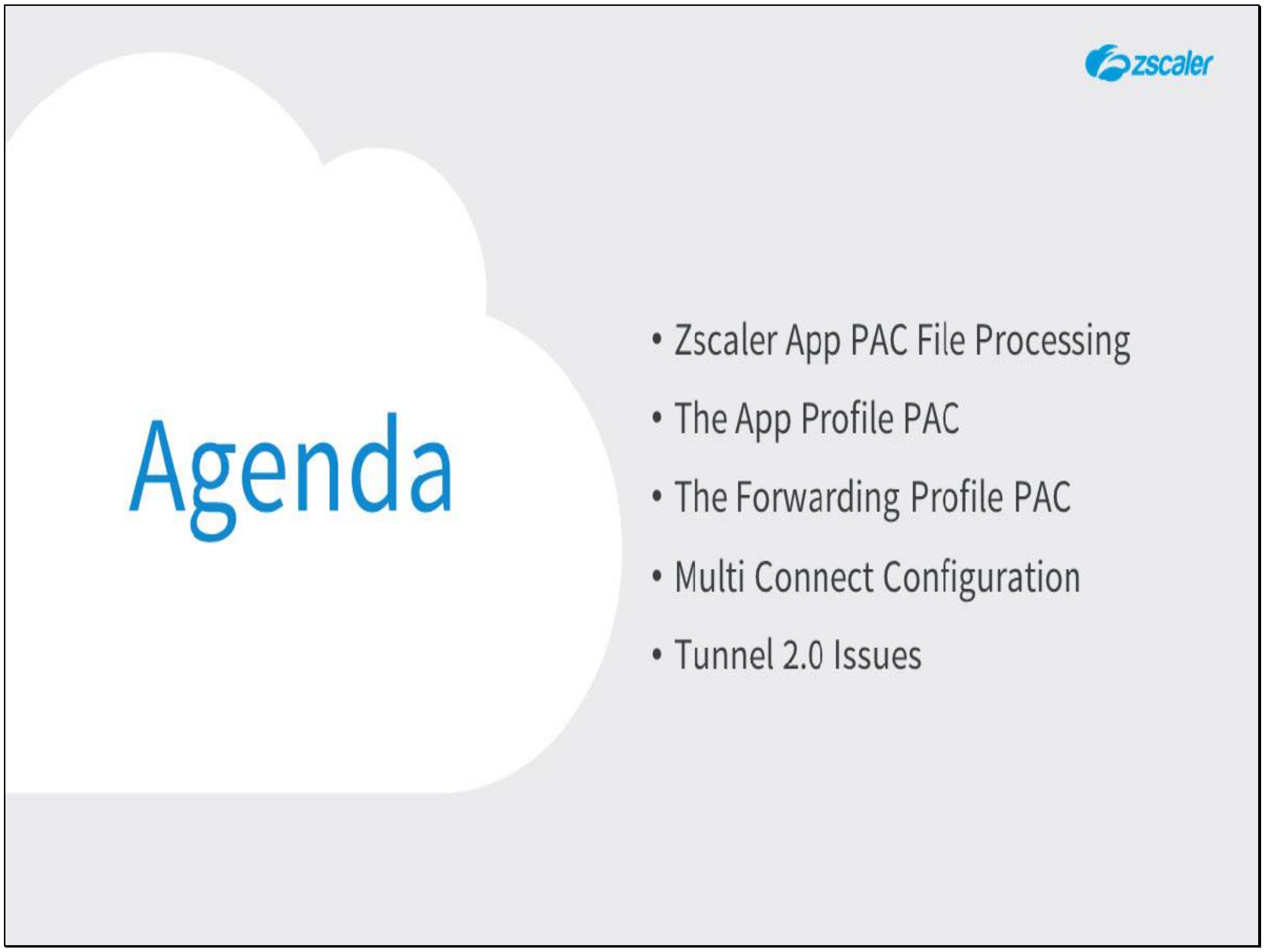
Slide 2 - Navigating the eLearning Module



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

- Zscaler App PAC File Processing
- The App Profile PAC
- The Forwarding Profile PAC
- Multi Connect Configuration
- Tunnel 2.0 Issues

Slide notes

In this module, we will cover the following topics:

- A look at how Zscaler App processes PAC files;
- A detailed look at the **App Profile** PAC file and it's typical contents;
- A look at the **Forwarding Profile** PAC file and its use in the various forwarding modes;
- A look at the Zscaler App **Multi-Connect** option;
- And a look at how the PAC file usage is modified with the **Tunnel 2.0** forwarding method.

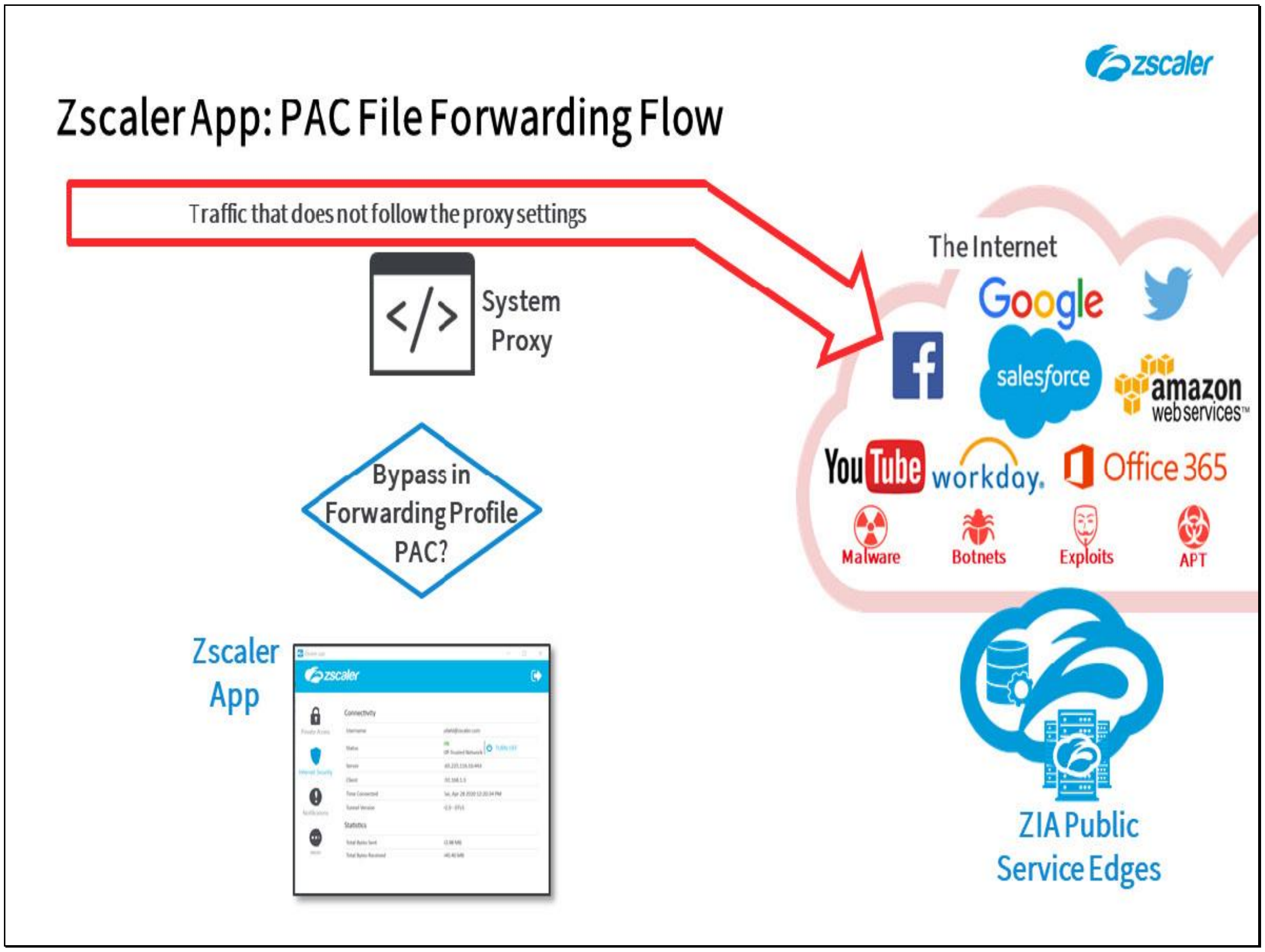
Slide 4 - PAC File Processing

Zscaler App PAC File Processing

Slide notes

The first topic that we will cover is a look at how Zscaler App processes PAC files.

Slide 5 - Zscaler App: PAC File Forwarding Flow

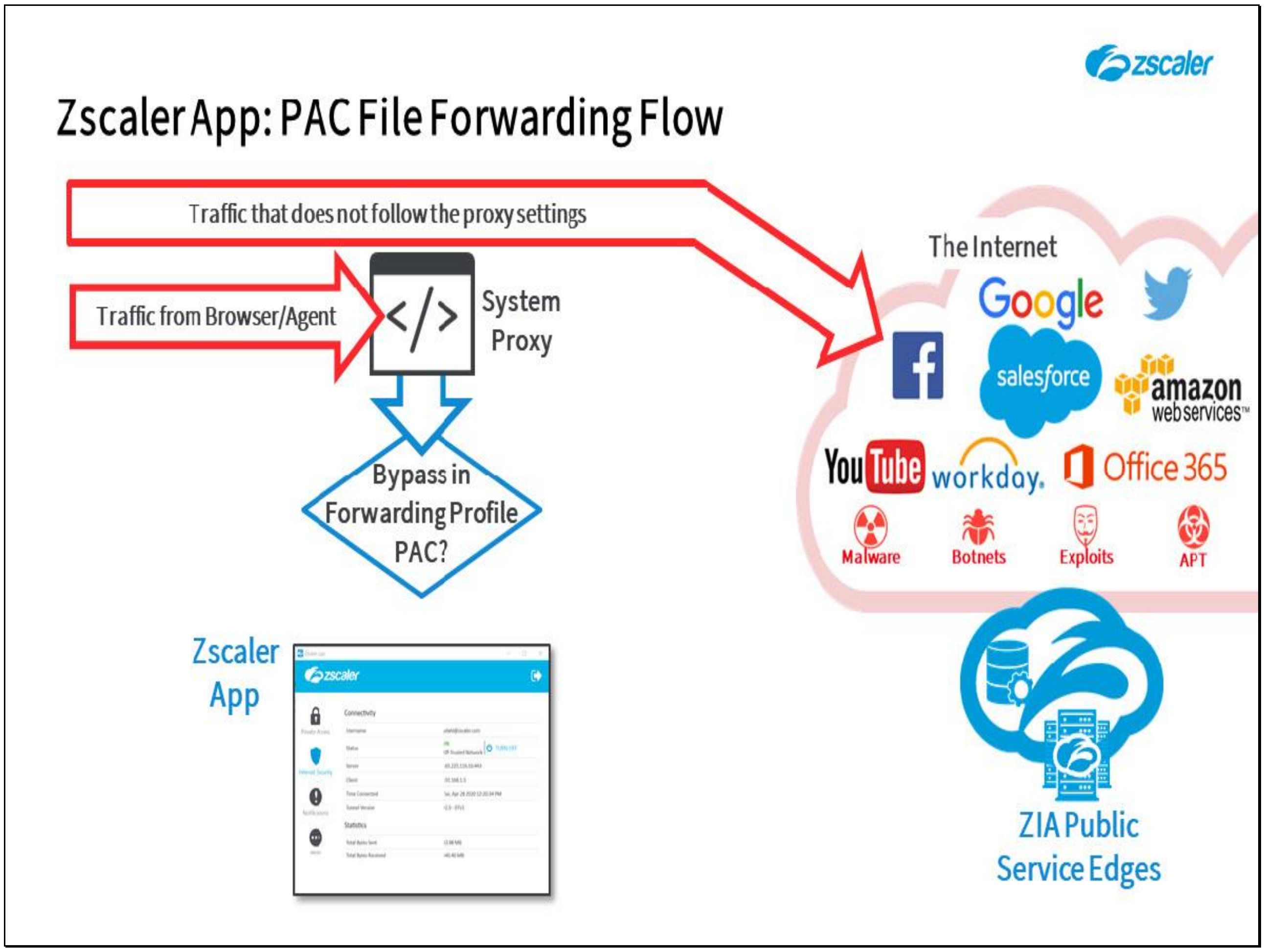


Slide notes

One thing it is important to understand is that the system proxy settings ONLY apply to traffic that follow the system proxy, ...which basically means **HTTP** or **HTTPS**. Any other traffic will normally just flow direct to the Internet.

The only exception here being when the **Tunnel 2.0** method is being used, in which case non-proxy traffic can be ‘included’ and sent by Zscaler App to the ZIA service.

Slide 6 - Zscaler App: PAC File Forwarding Flow

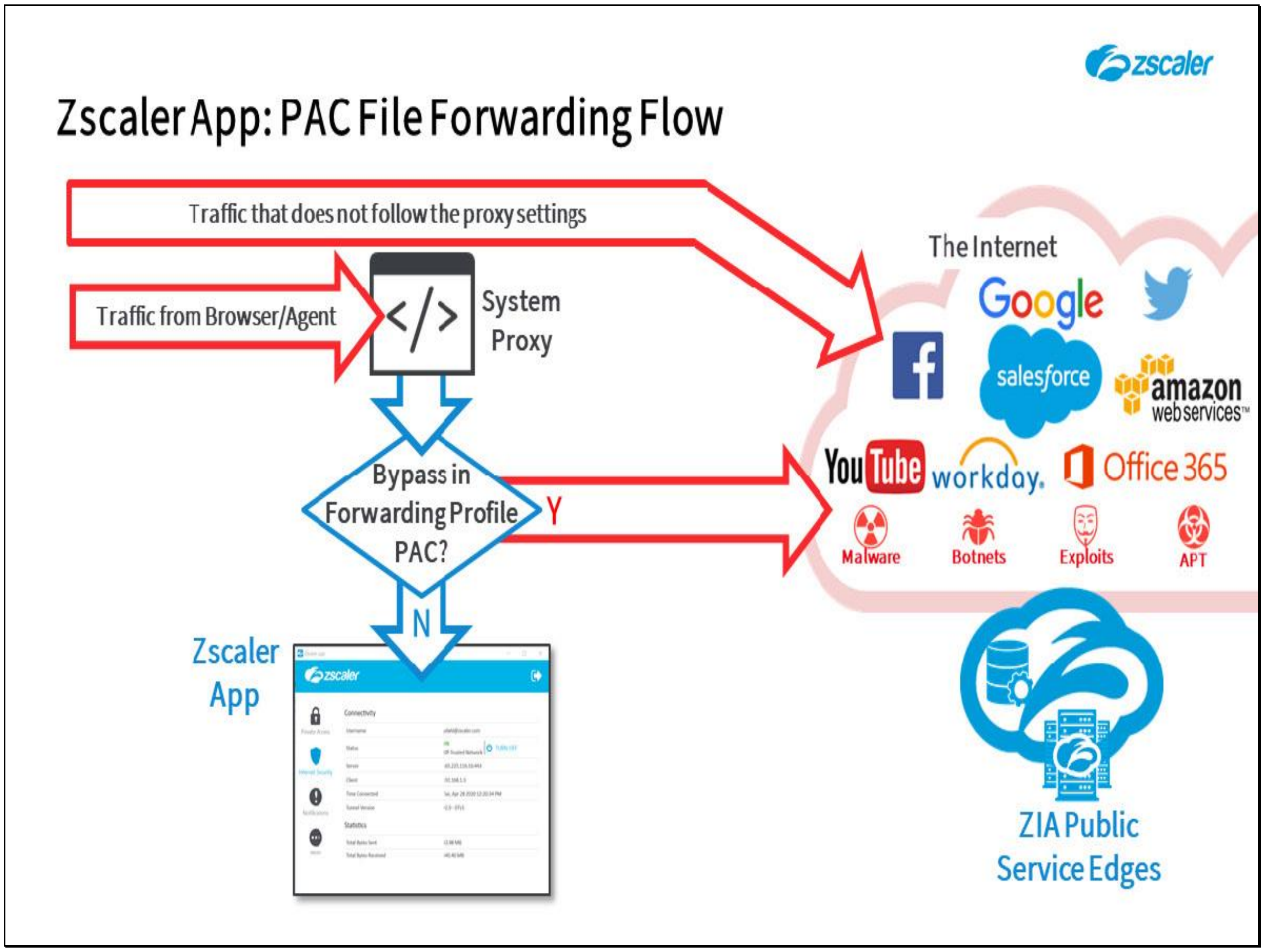


Slide notes

When an end user on the device generates traffic that follows the system proxy settings, that traffic is of course, first processed by the system proxy. An option with Z App is to replace any existing system proxy settings with a PAC file applied in a **Forwarding Profile**.

Any custom PAC file applied here may contain additional forwarding statements (bypasses) over and above the default statements, the options available being dependent on the forwarding mode specified in the applied **Forwarding Profile**.

Slide 7 - Zscaler App: PAC File Forwarding Flow



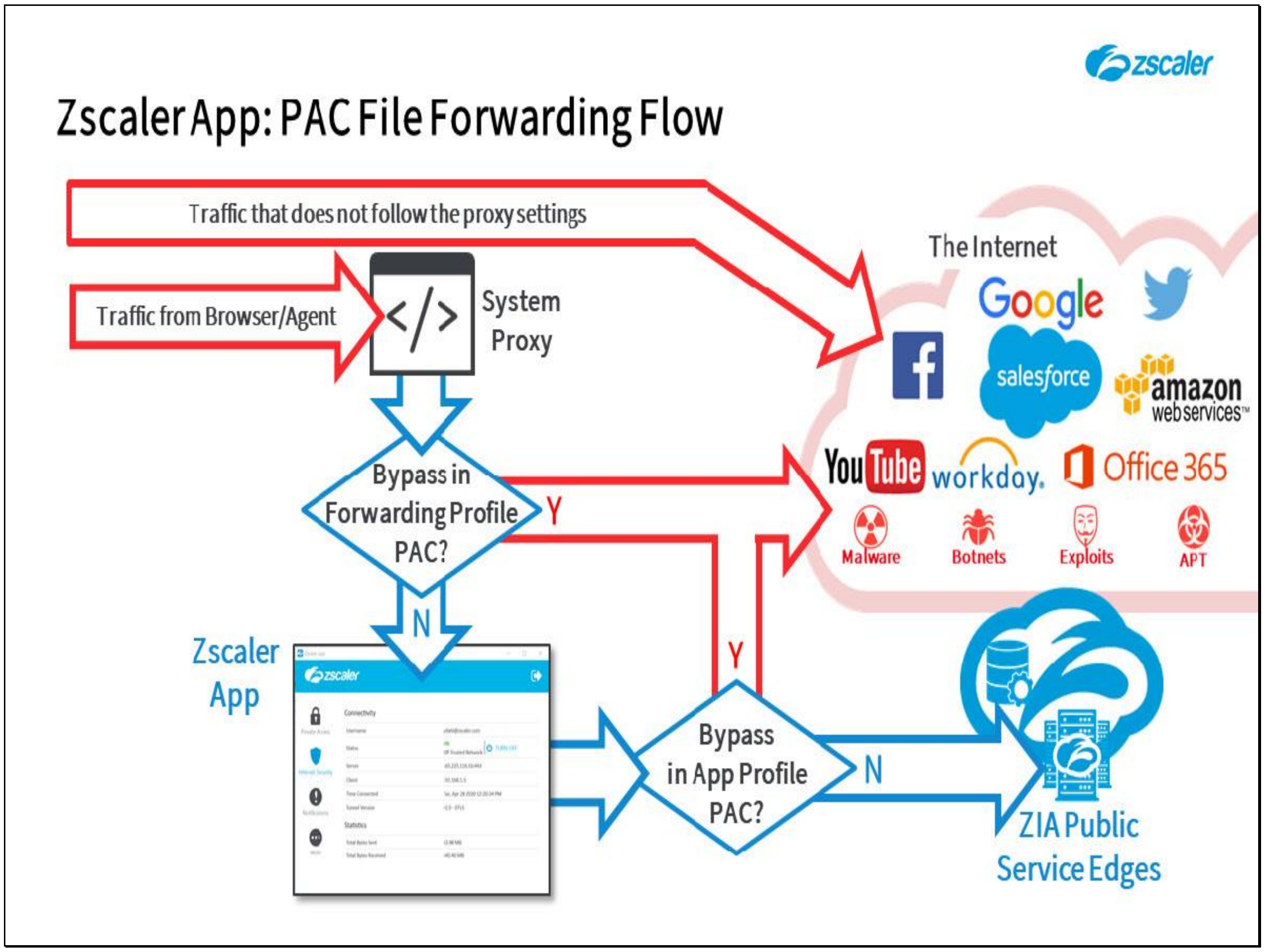
Slide notes

If the destination for the traffic generated matches a bypass rule in the **Forwarding Profile** PAC file, that traffic will be forwarded directly through the system's physical interface; and if this is the case, then this traffic is never even processed by Z App.

Other proxy traffic is forwarded into Zscaler App. The method used to channel this traffic into the App depends on the forwarding mode applied in the **Forwarding Profile**:

- For **Tunnel 1.0** mode, all **TCP** port **80**/**443** traffic (that is not bypassed) flows to Z App;
- For **TWLP** mode, **HTTP**/**HTTPS** traffic is explicitly proxied into the App.
- For **Tunnel 2.0** mode, all 'included' traffic regardless of protocol or port is forwarded to Z App, although it is possible to configure domain-based bypasses in the **App Profile** applied (more on that later).

Slide 8 - Zscaler App: PAC File Forwarding Flow

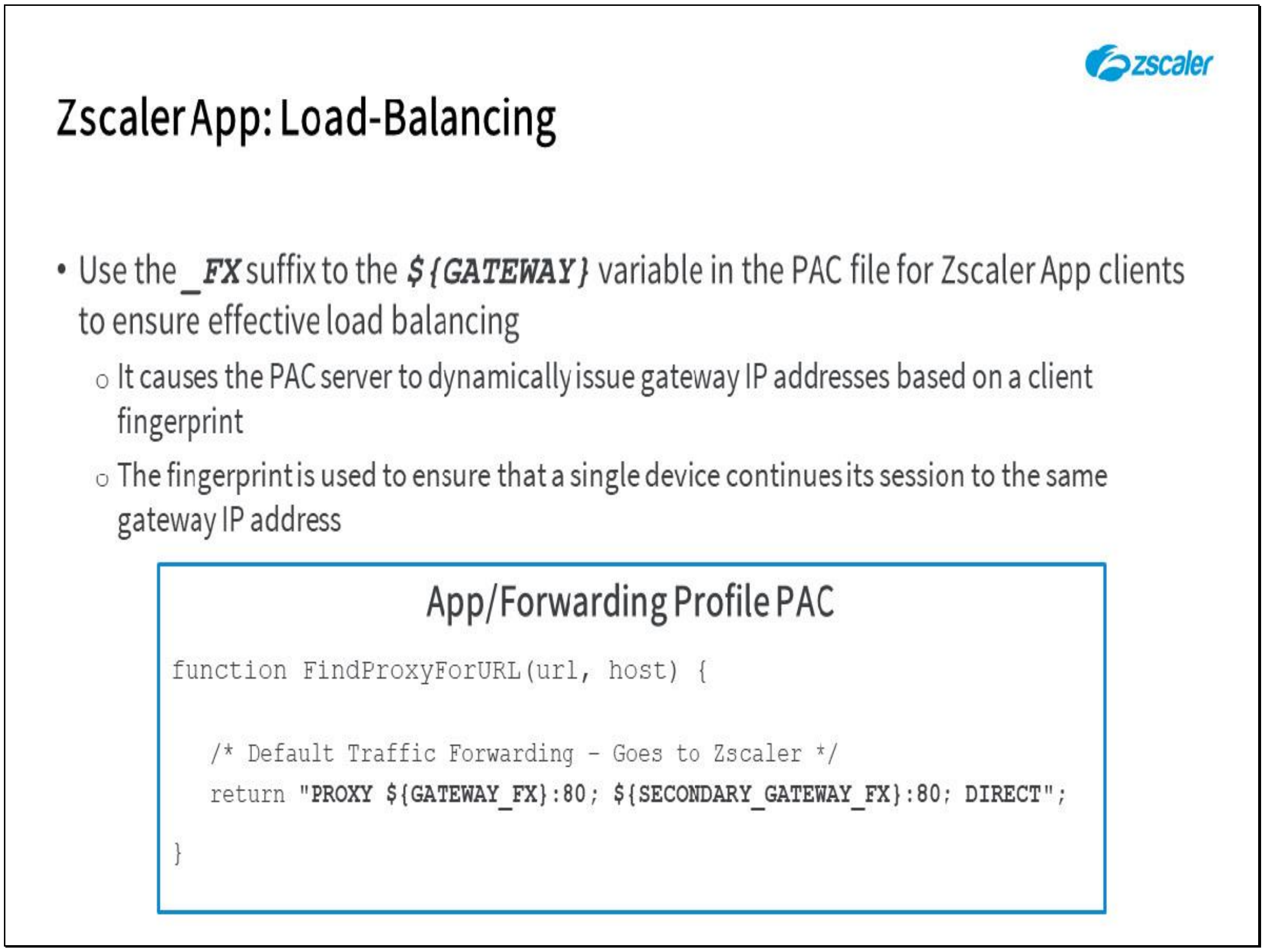

zscaler
Zscaler App: PAC File Forwarding Flow
Traffic that does not follow the proxy settings
Traffic from Browser/Agent
System Proxy
The Internet
Google
salesforce
amazon webservices
YouTube
workday.
Office 365
Malware
Botnets
Exploits
APT
Bypass in Forwarding Profile PAC?
Y
N
Zscaler App
Y
Bypass in App Profile PAC?
N
ZIA Public Service Edges

Slide notes

Zscaler App will then process, and forward traffic based on the configuration in the **App Profile** PAC file. Any bypasses here are sent direct, everything else will be sent to the nearest ZIA Public Service Edge node for scanning and the application of policy.

It is possible to override the default behavior and send traffic to a specific Service Edge node (**Public**, **Private** or **Virtual**), or even have branching logic in the **App Profile** PAC file to send traffic to different nodes based on certain conditions. This is the **Multi-Connect** feature, which we will talk about in a bit more detail later in this module.

Slide 9 - Zscaler App: Load-Balancing



Zscaler App: Load-Balancing

- Use the ***_FX*** suffix to the ***${GATEWAY}*** variable in the PAC file for Zscaler App clients to ensure effective load balancing
  - It causes the PAC server to dynamically issue gateway IP addresses based on a client fingerprint
  - The fingerprint is used to ensure that a single device continues its session to the same gateway IP address

App/Forwarding Profile PAC

```
function FindProxyForURL(url, host) {

   /* Default Traffic Forwarding - Goes to Zscaler */
   return "PROXY ${GATEWAY_FX}:80; ${SECONDARY_GATEWAY_FX}:80; DIRECT";

}
```

Slide notes

Under normal circumstances, where end users connect to the ZIA cloud from a known location on a GRE or IPSec tunnel, Zscaler is able to dynamically load-balance them across the Service Edge nodes as they are all NAT'd from a single egress IP address to the ZIA service. With Zscaler App however this method is not available, as every user has their own unique egress IP address. To resolve this, it is strongly recommended that you use the **_FX** suffix on any **${GATEWAY}** or **${SECONDARY_GATEWAY}** specifications in your Z App PAC files.

The **_FX** suffix provides load balancing for multiple Virtual IPs depending on the HTTP headers (**useragent**, **x-forwarded-for**, and **z-client**). This variable is used only for Zscaler App clients because the **z-client** ID is different for each user. Any forwarding statements in the **App Profile** PAC files that you apply to Zscaler App users should include this suffix. Note that this method can also be used for **Subcloud** destinations.

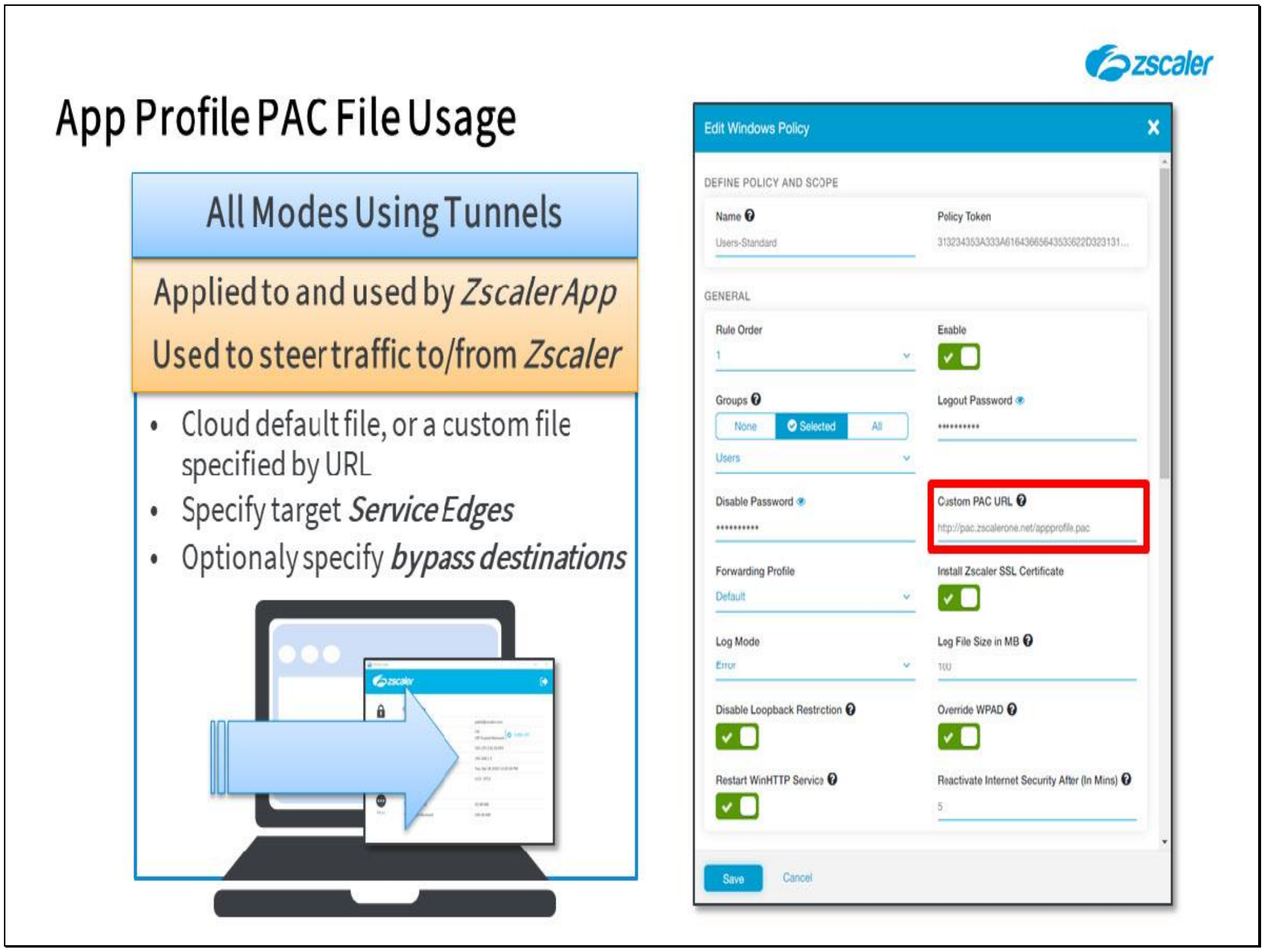
Slide 10 - The App Profile PAC

The App Profile PAC

Slide notes

The next topic that we will cover is a detailed look at the **App Profile** PAC file and at some of the default contents of it.

Slide 11 - App Profile PAC File Usage



App Profile PAC File Usage

All Modes Using Tunnels

Applied to and used by *Zscaler App*
Used to steer traffic to/from *Zscaler*

- Cloud default file, or a custom file specified by URL
- Specify target *Service Edges*
- Optionaly specify *bypass destinations*

Edit Windows Policy

DEFINE POLICY AND SCOPE

Name	Users-Standard
Policy Token	313234353A333A616436656435353622D323131...

GENERAL

Rule Order	1
Enable	✓
Groups	None [✓ Selected] All
	Users
Logout Password	••••••••••
Disable Password	••••••••••
Custom PAC URL	http://pac.zscalerone.net/appprofile.pac
Forwarding Profile	Default
Install Zscaler SSL Certificate	✓
Log Mode	Error
Log File Size in MB	100
Disable Loopback Restriction	✓
Override WPAD	✓
Restart WinHTTP Service	✓
Reactivate Internet Security After (In Mins)	5

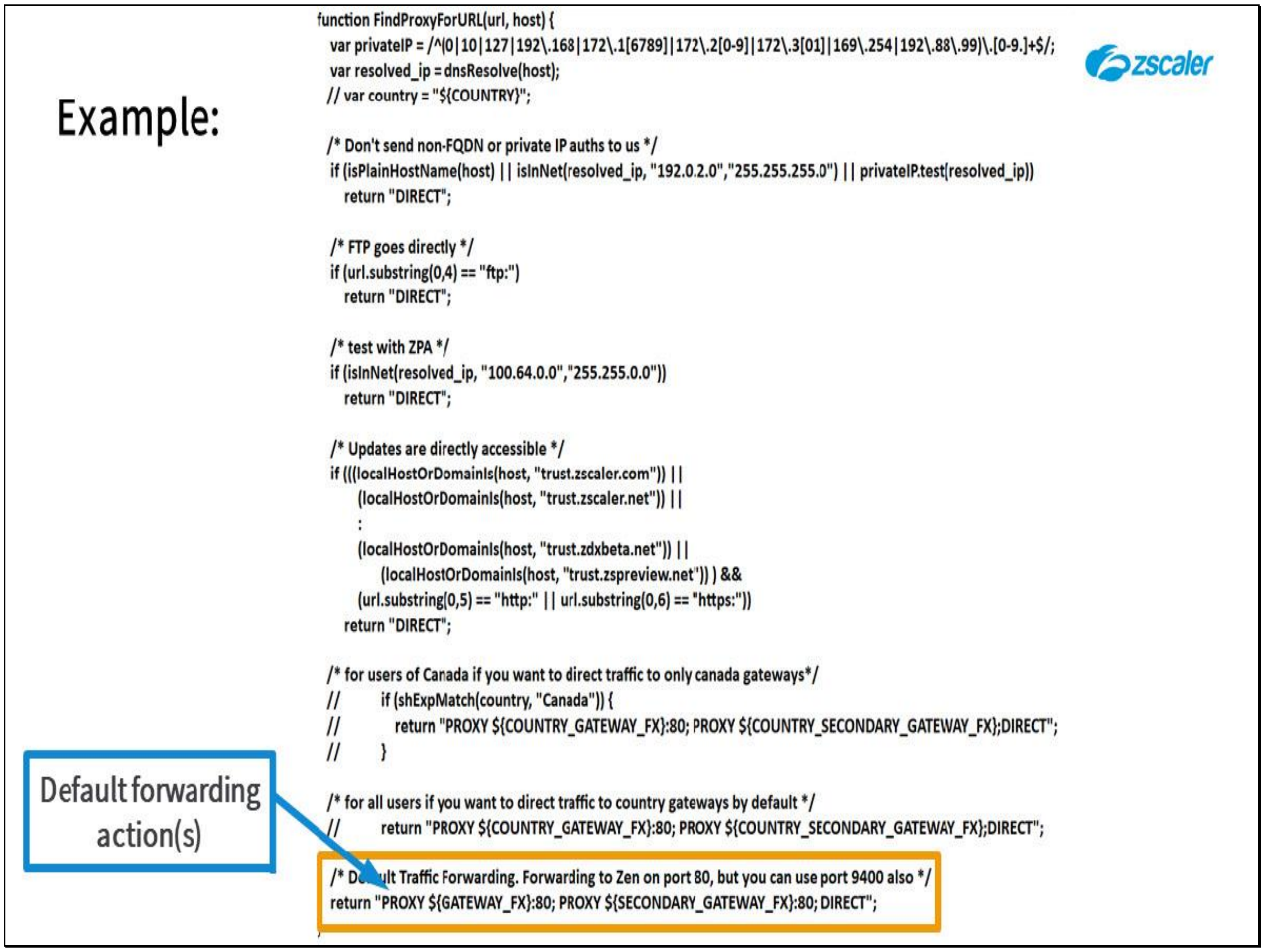
Slide notes

The PAC file applied by the **App Profile** by default is the **default PAC file** for the Cloud that you are connected to; although if you wish you can override this and provide the URL for a custom PAC file.

This file is only applied to and used by the App as a pseudo-configuration file, the host system never sees this file. Its primary use is to specify the target **Service Edges** for **Tunnels** (either the closest **Public** Service Edge nodes or a specified **Public**, **Private** or **Virtual** node). It can also be used to specify destinations that are to bypass tunnels in the **Tunnel** or **TWLP** forwarding modes.

Note that any bypass destinations in this file are bypassed by Zscaler App itself, meaning that this traffic is still processed by the App. Also note that this file is not used at all in the **Enforce PAC** mode. This PAC file is refreshed every **15 minutes** (as are all PAC files applied to Z App).

Slide 12 - Example:



Example:

```
function FindProxyForURL(url, host) {
  var privateIP = /^(0|10|127|192\.168|172\.1[6789]|172\.2[0-9]|172\.3[01]|169\.254|192\.88\.99)\.[0-9.]+$/;
  var resolved_ip = dnsResolve(host);
  // var country = "${COUNTRY}";

  /* Don't send non-FQDN or private IP auths to us */
  if (isPlainHostName(host) || isInNet(resolved_ip, "192.0.2.0","255.255.255.0") || privateIP.test(resolved_ip))
    return "DIRECT";

  /* FTP goes directly */
  if (url.substring(0,4) == "ftp:")
    return "DIRECT";

  /* test with ZPA */
  if (isInNet(resolved_ip, "100.64.0.0","255.255.0.0"))
    return "DIRECT";

  /* Updates are directly accessible */
  if (((localHostOrDomainIs(host, "trust.zscaler.com")) ||
      (localHostOrDomainIs(host, "trust.zscaler.net")) ||
      :
      (localHostOrDomainIs(host, "trust.zdxbeta.net")) ||
         (localHostOrDomainIs(host, "trust.zspreview.net")) ) &&
      (url.substring(0,5) == "http:" || url.substring(0,6) == "https:"))
    return "DIRECT";

  /* for users of Canada if you want to direct traffic to only canada gateways*/
  //      if (shExpMatch(country, "Canada")) {
  //         return "PROXY ${COUNTRY_GATEWAY_FX}:80; PROXY ${COUNTRY_SECONDARY_GATEWAY_FX};DIRECT";
  //      }

  /* for all users if you want to direct traffic to country gateways by default */
  //      return "PROXY ${COUNTRY_GATEWAY_FX}:80; PROXY ${COUNTRY_SECONDARY_GATEWAY_FX};DIRECT";

  /* Default Traffic Forwarding. Forwarding to Zen on port 80, but you can use port 9400 also */
  return "PROXY ${GATEWAY_FX}:80; PROXY ${SECONDARY_GATEWAY_FX}:80; DIRECT";
}
```

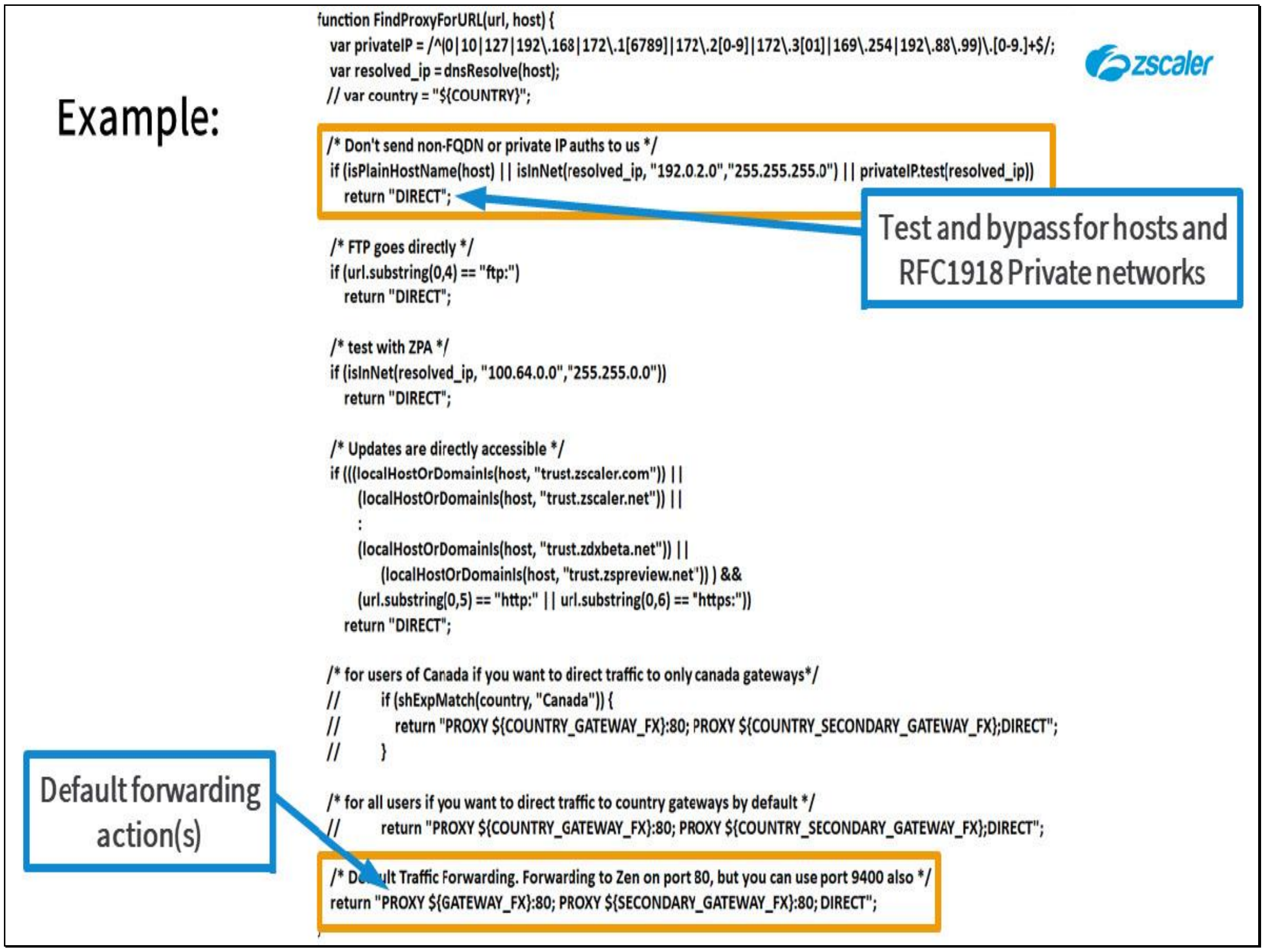
Slide notes

If you do not specify a custom PAC file URL in the **App Profile**, the PAC file applied is the **default PAC file** for the cloud that you are connected to. In the ZIA Admin Portal, navigate to the page at **Administration > Resources > Hosted PAC Files** and review the **Service Default** file named **proxy.pac**; an example is shown here.

The default forwarding statements are at the bottom of the file using both primary and secondary gateway definitions (both with the **_FX** suffix), and a fail-open **DIRECT** statement.

Note, if you wish to create your own custom **App Profile** PAC file, this file would be a good starting point.

Slide 13 - Example:



Example:

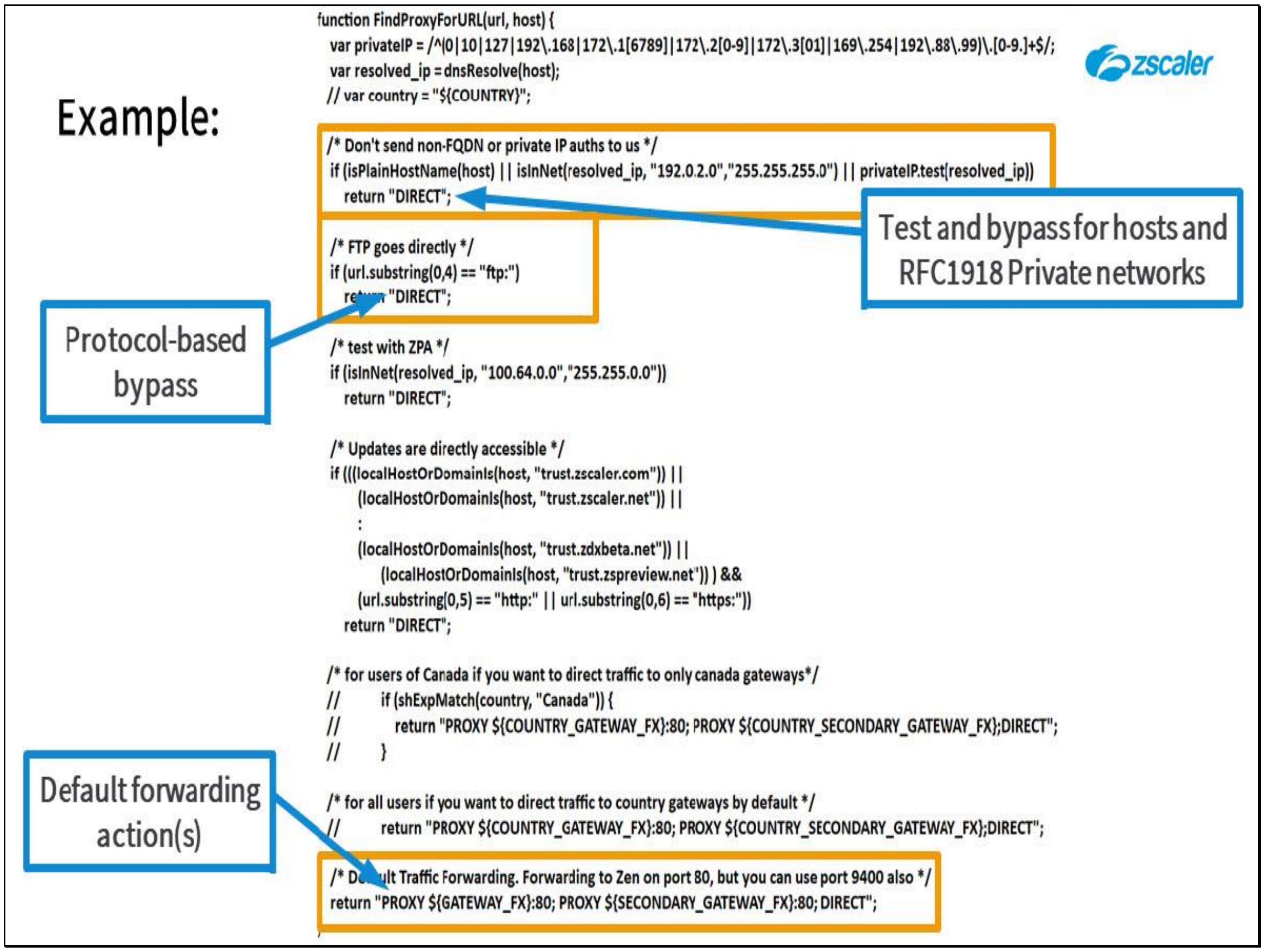
```
function FindProxyForURL(url, host) {
  var privateIP = /^(0|10|127|192\.168|172\.1[6789]|172\.2[0-9]|172\.3[01]|169\.254|192\.88\.99)\.[0-9.]+$/;
  var resolved_ip = dnsResolve(host);
 // var country = "${COUNTRY}";

 /* Don't send non-FQDN or private IP auths to us */
  if (isPlainHostName(host) || isInNet(resolved_ip, "192.0.2.0","255.255.255.0") || privateIP.test(resolved_ip))
    return "DIRECT";

  /* FTP goes directly */
  if (url.substring(0,4) == "ftp:")
    return "DIRECT";

  /* test with ZPA */
  if (isInNet(resolved_ip, "100.64.0.0","255.255.0.0"))
    return "DIRECT";

  /* Updates are directly accessible */
  if (((localHostOrDomainIs(host, "trust.zscaler.com")) ||
      (localHostOrDomainIs(host, "trust.zscaler.net")) ||
      :
      (localHostOrDomainIs(host, "trust.zdxbeta.net")) ||
         (localHostOrDomainIs(host, "trust.zspreview.net")) ) &&
      (url.substring(0,5) == "http:" || url.substring(0,6) == "https:"))
    return "DIRECT";

 /* for users of Canada if you want to direct traffic to only canada gateways*/
 //        if (shExpMatch(country, "Canada")) {
 //           return "PROXY ${COUNTRY_GATEWAY_FX}:80; PROXY ${COUNTRY_SECONDARY_GATEWAY_FX};DIRECT";
 //        }

 /* for all users if you want to direct traffic to country gateways by default */
 //        return "PROXY ${COUNTRY_GATEWAY_FX}:80; PROXY ${COUNTRY_SECONDARY_GATEWAY_FX};DIRECT";

 /* Default Traffic Forwarding. Forwarding to Zen on port 80, but you can use port 9400 also */
 return "PROXY ${GATEWAY_FX}:80; PROXY ${SECONDARY_GATEWAY_FX}:80; DIRECT";
```

zscaler
Test and bypass for hosts and RFC1918 Private networks
Default forwarding action(s)

Slide notes

The default file also contains: A test and bypass for **plain host names** (host names with no associated domain) and **RFC1918 private networks**;

Slide 14 - Example:



Example:

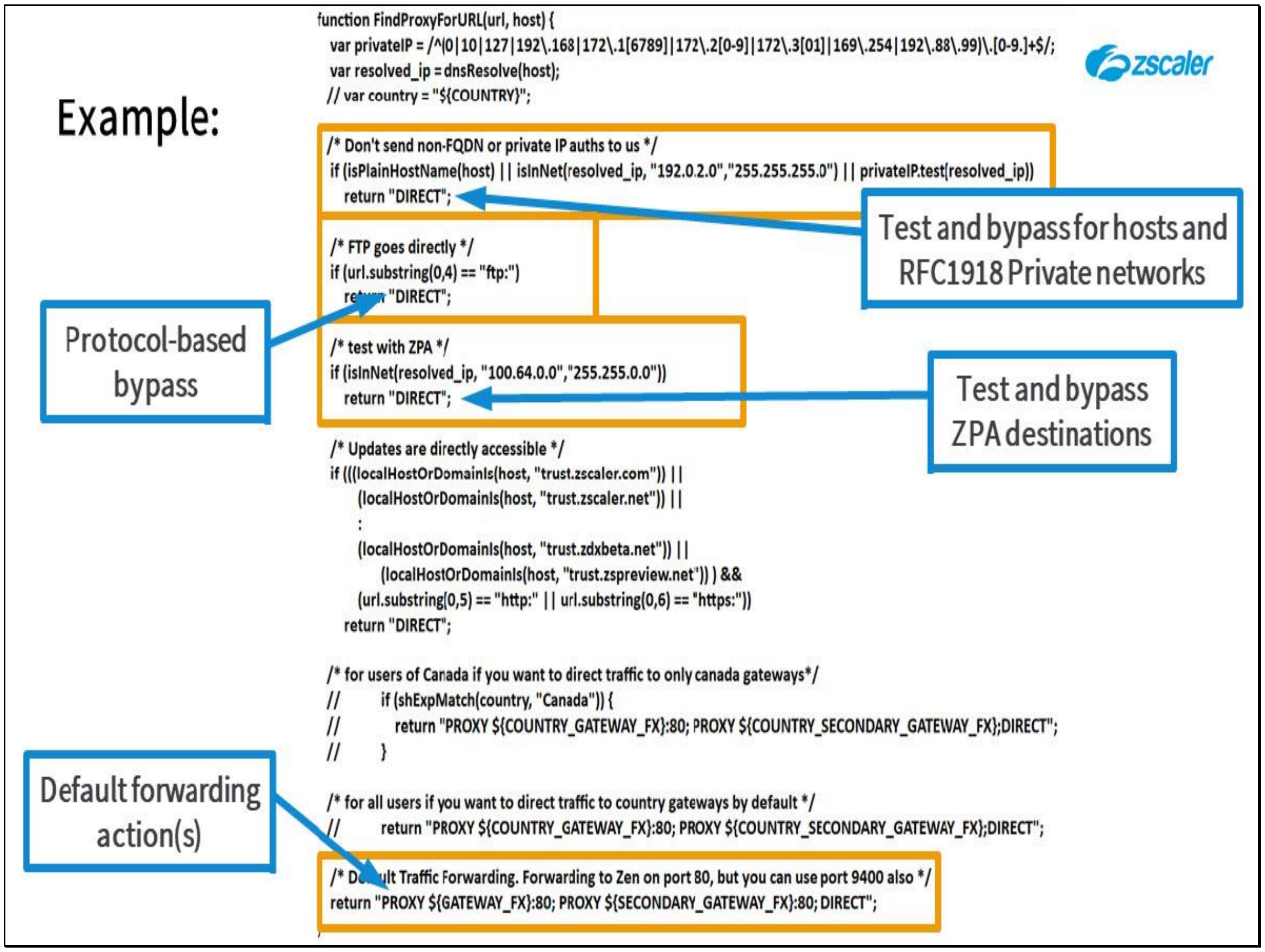
```
function FindProxyForURL(url, host) {
  var privateIP = /^(0|10|127|192\.168|172\.1[6789]|172\.2[0-9]|172\.3[01]|169\.254|192\.88\.99)\.[0-9.]+$/;
  var resolved_ip = dnsResolve(host);
  // var country = "${COUNTRY}";

  /* Don't send non-FQDN or private IP auths to us */
  if (isPlainHostName(host) || isInNet(resolved_ip, "192.0.2.0","255.255.255.0") || privateIP.test(resolved_ip))
    return "DIRECT";

  /* FTP goes directly */
  if (url.substring(0,4) == "ftp:")
    return "DIRECT";

  /* test with ZPA */
  if (isInNet(resolved_ip, "100.64.0.0","255.255.0.0"))
    return "DIRECT";

  /* Updates are directly accessible */
  if (((localHostOrDomainIs(host, "trust.zscaler.com")) ||
      (localHostOrDomainIs(host, "trust.zscaler.net")) ||
      :
      (localHostOrDomainIs(host, "trust.zdxbeta.net")) ||
          (localHostOrDomainIs(host, "trust.zspreview.net")) ) &&
      (url.substring(0,5) == "http:" || url.substring(0,6) == "https:"))
    return "DIRECT";

  /* for users of Canada if you want to direct traffic to only canada gateways*/
  //       if (shExpMatch(country, "Canada")) {
  //          return "PROXY ${COUNTRY_GATEWAY_FX}:80; PROXY ${COUNTRY_SECONDARY_GATEWAY_FX};DIRECT";
  //       }

  /* for all users if you want to direct traffic to country gateways by default */
  //       return "PROXY ${COUNTRY_GATEWAY_FX}:80; PROXY ${COUNTRY_SECONDARY_GATEWAY_FX};DIRECT";

  /* Default Traffic Forwarding. Forwarding to Zen on port 80, but you can use port 9400 also */
  return "PROXY ${GATEWAY_FX}:80; PROXY ${SECONDARY_GATEWAY_FX}:80; DIRECT";
```

Test and bypass for hosts and RFC1918 Private networks
Protocol-based bypass
Default forwarding action(s)

Slide notes

A test and bypass for the **FTP** protocol;

Slide 15 - Example:



Example:

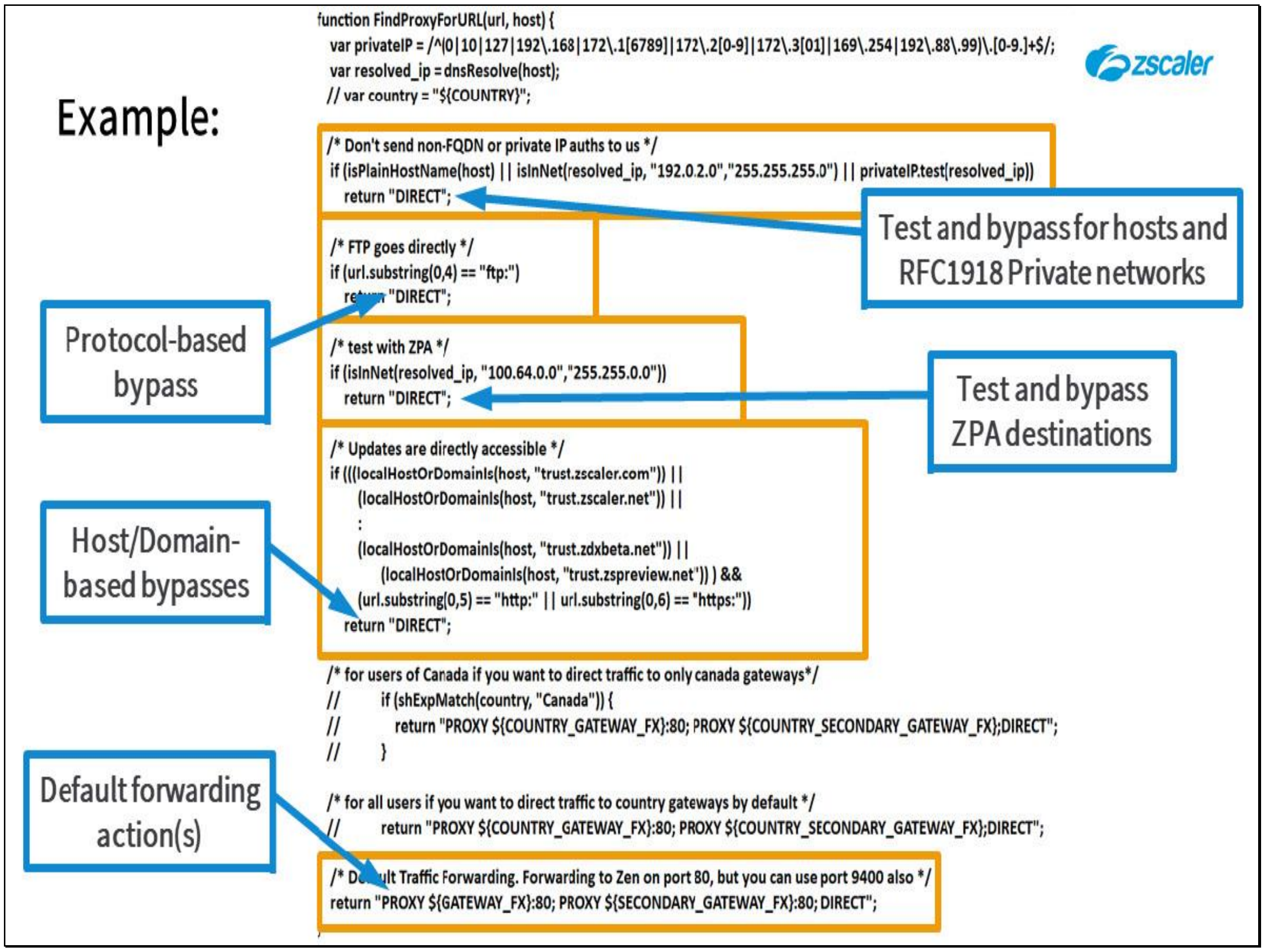
```
function FindProxyForURL(url, host) {
  var privateIP = /^(0|10|127|192\.168|172\.1[6789]|172\.2[0-9]|172\.3[01]|169\.254|192\.88\.99)\.[0-9.]+$/;
  var resolved_ip = dnsResolve(host);
 // var country = "${COUNTRY}";

 /* Don't send non-FQDN or private IP auths to us */
  if (isPlainHostName(host) || isInNet(resolved_ip, "192.0.2.0","255.255.255.0") || privateIP.test(resolved_ip))
    return "DIRECT";

  /* FTP goes directly */
  if (url.substring(0,4) == "ftp:")
    return "DIRECT";

  /* test with ZPA */
  if (isInNet(resolved_ip, "100.64.0.0","255.255.0.0"))
    return "DIRECT";

  /* Updates are directly accessible */
  if (((localHostOrDomainIs(host, "trust.zscaler.com")) ||
      (localHostOrDomainIs(host, "trust.zscaler.net")) ||
      :
      (localHostOrDomainIs(host, "trust.zdxbeta.net")) ||
         (localHostOrDomainIs(host, "trust.zspreview.net")) ) &&
      (url.substring(0,5) == "http:" || url.substring(0,6) == "https:"))
    return "DIRECT";

 /* for users of Canada if you want to direct traffic to only canada gateways*/
 //       if (shExpMatch(country, "Canada")) {
 //          return "PROXY ${COUNTRY_GATEWAY_FX}:80; PROXY ${COUNTRY_SECONDARY_GATEWAY_FX};DIRECT";
 //       }

 /* for all users if you want to direct traffic to country gateways by default */
 //       return "PROXY ${COUNTRY_GATEWAY_FX}:80; PROXY ${COUNTRY_SECONDARY_GATEWAY_FX};DIRECT";

 /* Default Traffic Forwarding. Forwarding to Zen on port 80, but you can use port 9400 also */
 return "PROXY ${GATEWAY_FX}:80; PROXY ${SECONDARY_GATEWAY_FX}:80; DIRECT";
```

zscaler
Test and bypass for hosts and RFC1918 Private networks
Protocol-based bypass
Test and bypass ZPA destinations
Default forwarding action(s)

Slide notes

A test and bypass for the ZPA subnet (**100.64.0.0/16**) – note that if ZPA runs out of addresses on this subnet, it will start allocating addresses on the **100.65.0.0/16** subnet, meaning that this subnet would also need to be added as a bypass;

Slide 16 - Example:


zscaler
Test and bypass for hosts and RFC1918 Private networks
Protocol-based bypass
Test and bypass ZPA destinations
Host/Domain-based bypasses
Default forwarding action(s)

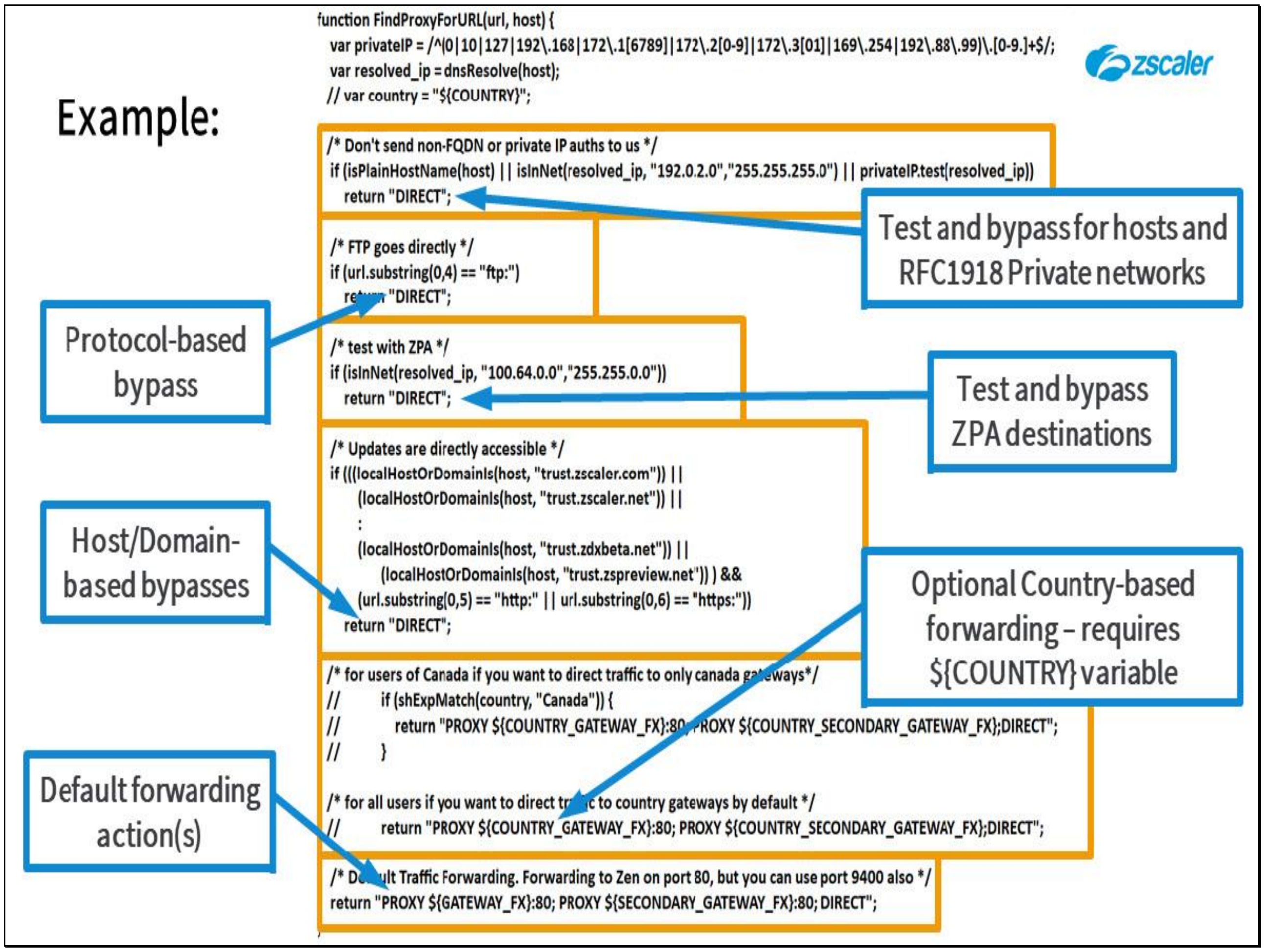
Example:

```
function FindProxyForURL(url, host) {
  var privateIP = /^(0|10|127|192\.168|172\.1[6789]|172\.2[0-9]|172\.3[01]|169\.254|192\.88\.99)\.[0-9.]+$/;
  var resolved_ip = dnsResolve(host);
 // var country = "${COUNTRY}";

/* Don't send non-FQDN or private IP auths to us */
 if (isPlainHostName(host) || isInNet(resolved_ip, "192.0.2.0","255.255.255.0") || privateIP.test(resolved_ip))
   return "DIRECT";

 /* FTP goes directly */
 if (url.substring(0,4) == "ftp:")
   return "DIRECT";

 /* test with ZPA */
 if (isInNet(resolved_ip, "100.64.0.0","255.255.0.0"))
   return "DIRECT";

 /* Updates are directly accessible */
 if (((localHostOrDomainIs(host, "trust.zscaler.com")) ||
      (localHostOrDomainIs(host, "trust.zscaler.net")) ||
      :
      (localHostOrDomainIs(host, "trust.zdxbeta.net")) ||
         (localHostOrDomainIs(host, "trust.zspreview.net")) ) &&
      (url.substring(0,5) == "http:" || url.substring(0,6) == "https:"))
   return "DIRECT";

/* for users of Canada if you want to direct traffic to only canada gateways*/
//       if (shExpMatch(country, "Canada")) {
//          return "PROXY ${COUNTRY_GATEWAY_FX}:80; PROXY ${COUNTRY_SECONDARY_GATEWAY_FX};DIRECT";
//       }

/* for all users if you want to direct traffic to country gateways by default */
//       return "PROXY ${COUNTRY_GATEWAY_FX}:80; PROXY ${COUNTRY_SECONDARY_GATEWAY_FX};DIRECT";

/* Default Traffic Forwarding. Forwarding to Zen on port 80, but you can use port 9400 also */
return "PROXY ${GATEWAY_FX}:80; PROXY ${SECONDARY_GATEWAY_FX}:80; DIRECT";
```

Slide notes

Tests and bypasses for all the Zscaler **Trust pages** (I have removed some from the list here for the sake of clarity);

Slide 17 - Example:



Example:

```
function FindProxyForURL(url, host) {
  var privateIP = /^(0|10|127|192\.168|172\.1[6789]|172\.2[0-9]|172\.3[01]|169\.254|192\.88\.99)\.[0-9.]+$/;
  var resolved_ip = dnsResolve(host);
 // var country = "${COUNTRY}";

 /* Don't send non-FQDN or private IP auths to us */
  if (isPlainHostName(host) || isInNet(resolved_ip, "192.0.2.0","255.255.255.0") || privateIP.test(resolved_ip))
    return "DIRECT";

  /* FTP goes directly */
  if (url.substring(0,4) == "ftp:")
    return "DIRECT";

  /* test with ZPA */
  if (isInNet(resolved_ip, "100.64.0.0","255.255.0.0"))
    return "DIRECT";

  /* Updates are directly accessible */
  if (((localHostOrDomainIs(host, "trust.zscaler.com")) ||
       (localHostOrDomainIs(host, "trust.zscaler.net")) ||
       :
       (localHostOrDomainIs(host, "trust.zdxbeta.net")) ||
          (localHostOrDomainIs(host, "trust.zspreview.net")) ) &&
       (url.substring(0,5) == "http:" || url.substring(0,6) == "https:"))
    return "DIRECT";

 /* for users of Canada if you want to direct traffic to only canada gateways*/
 //        if (shExpMatch(country, "Canada")) {
 //           return "PROXY ${COUNTRY_GATEWAY_FX}:80; PROXY ${COUNTRY_SECONDARY_GATEWAY_FX};DIRECT";
 //        }

 /* for all users if you want to direct traffic to country gateways by default */
 //        return "PROXY ${COUNTRY_GATEWAY_FX}:80; PROXY ${COUNTRY_SECONDARY_GATEWAY_FX};DIRECT";

  /* Default Traffic Forwarding. Forwarding to Zen on port 80, but you can use port 9400 also */
  return "PROXY ${GATEWAY_FX}:80; PROXY ${SECONDARY_GATEWAY_FX}:80; DIRECT";
```

Test and bypass for hosts and RFC1918 Private networks
Protocol-based bypass
Test and bypass ZPA destinations
Host/Domain-based bypasses
Optional Country-based forwarding – requires ${COUNTRY} variable
Default forwarding action(s)

Slide notes

Plus there are some commented out examples of how to use the **${COUNTRY}** variable to direct traffic to the closest Service Edge **within the country** that the end user's egress IP address resolves to through geo-location.

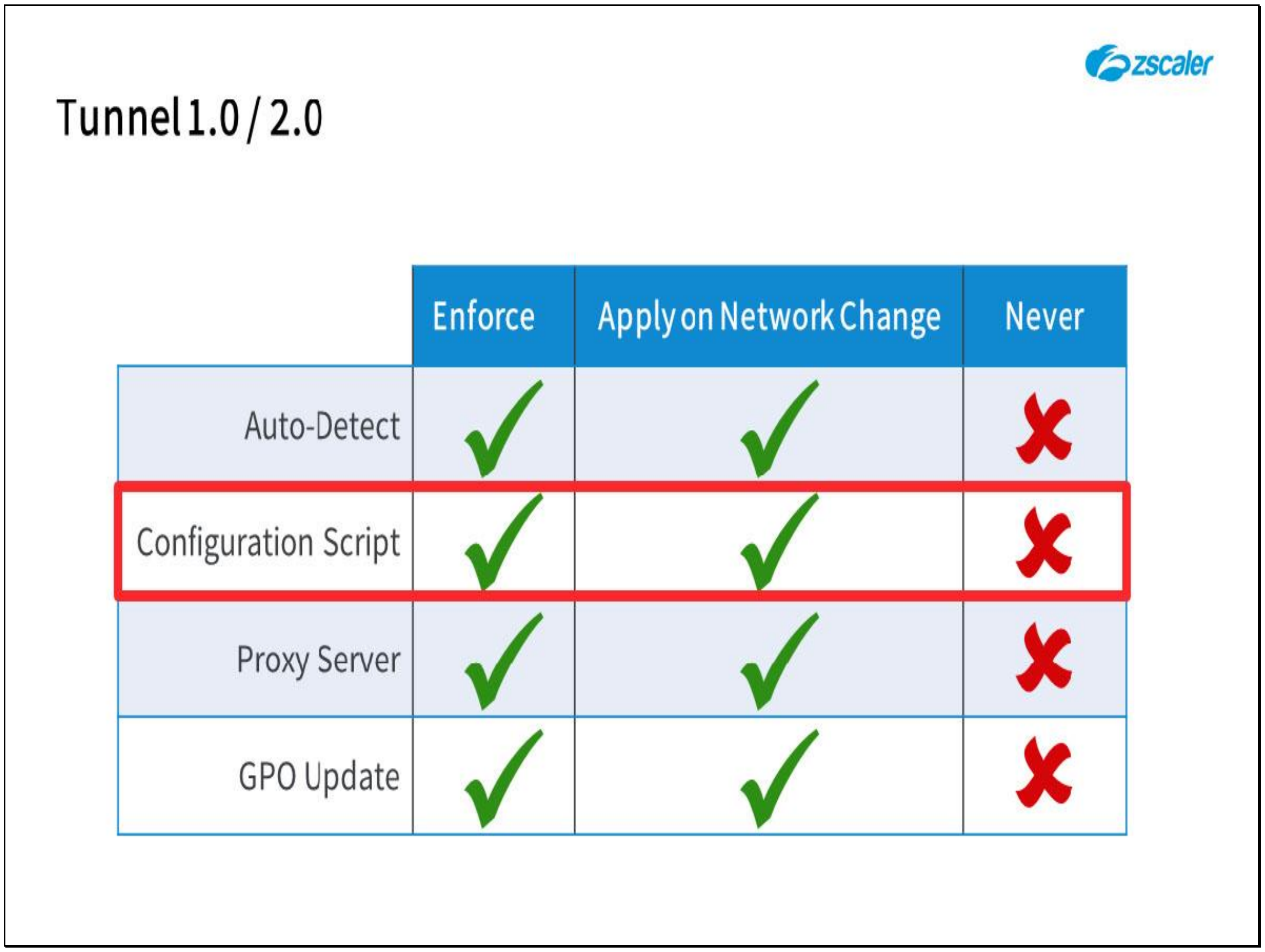
Slide 18 - The Forwarding Profile PAC

The Forwarding Profile PAC

Slide notes

The next topic that we will cover is a look at the **Forwarding Profile** PAC file and at how it may be used depending on the forwarding mode specified.

Slide 19 - Tunnel 1.0 / 2.0



Tunnel 1.0 / 2.0

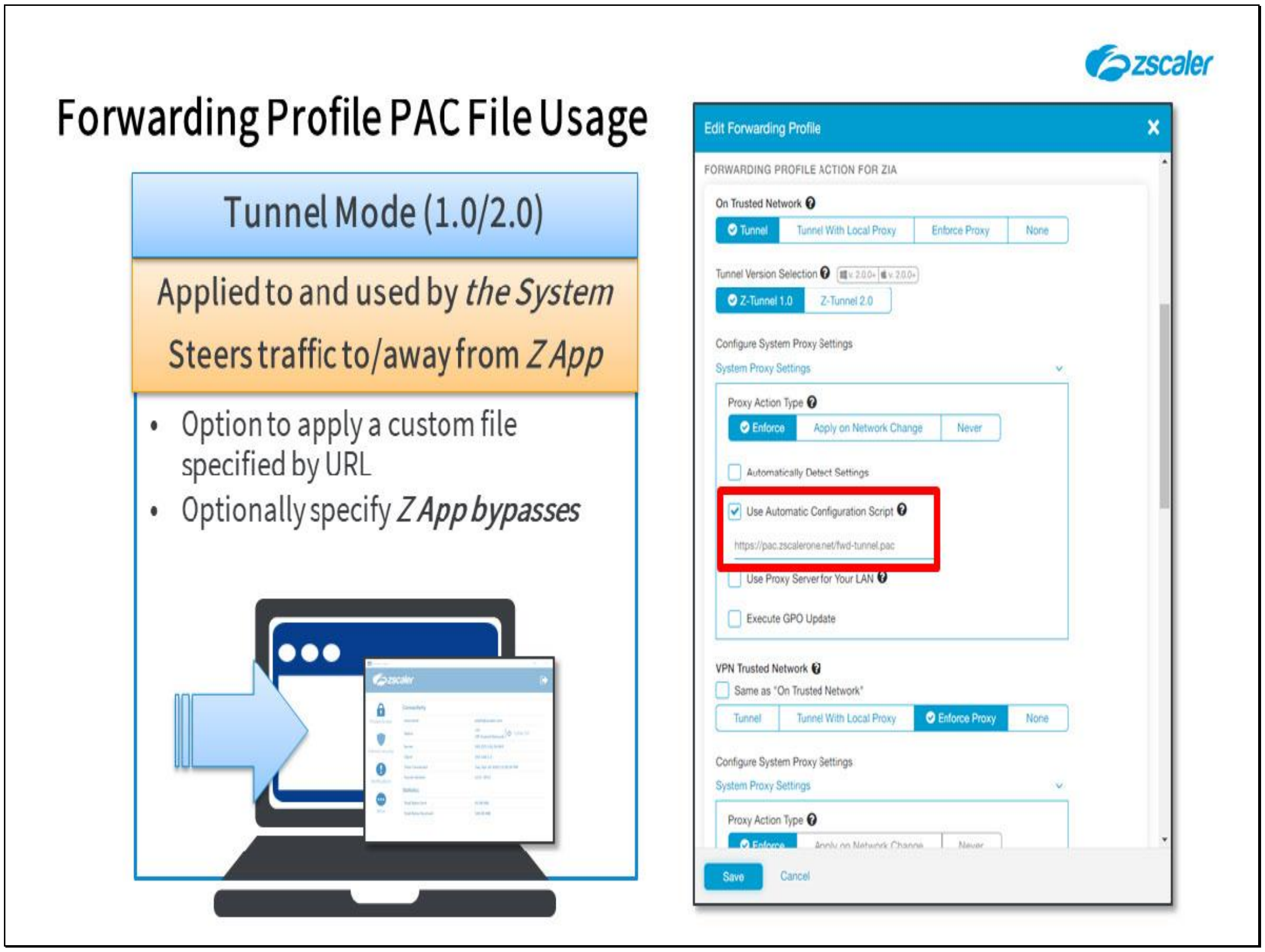
	Enforce	Apply on Network Change	Never
Auto-Detect	✓	✓	✗
Configuration Script	✓	✓	✗
Proxy Server	✓	✓	✗
GPO Update	✓	✓	✗

Slide notes

In the **Forwarding Profile**, when selecting a forwarding mode for **On**, **Off** or **VPN Trusted Network**, you have options for configuring the **System Proxy Settings** (we discussed these at some length in the **Under the Covers** module).

Shown here are the options for the **Tunnel** forwarding mode (**1.0** or **2.0**) and the **Forwarding Profile** PAC file option is the **Configuration Script** option highlighted here in red. We would recommend that, if you have a PAC file to be applied in the **Forwarding Profile**, that you configure ONLY this option, to avoid one of the other methods over-writing the PAC file that you apply.

Slide 20 - Forwarding Profile PAC File Usage



Forwarding Profile PAC File Usage

Tunnel Mode (1.0/2.0)

Applied to and used by *the System*

Steers traffic to/away from *Z App*

- Option to apply a custom file specified by URL
- Optionally specify *Z App bypasses*

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

Configure System Proxy Settings

System Proxy Settings

Proxy Action Type: [✓ Enforce] Apply on Network Change | Never

☐ Automatically Detect Settings

☑ Use Automatic Configuration Script

https://pac.zscalerone.net/fwd-tunnel.pac

☐ Use Proxy Server for Your LAN

☐ Execute GPO Update

VPN Trusted Network

☐ Same as "On Trusted Network"

Tunnel | Tunnel With Local Proxy | [✓ Enforce Proxy] None

Configure System Proxy Settings

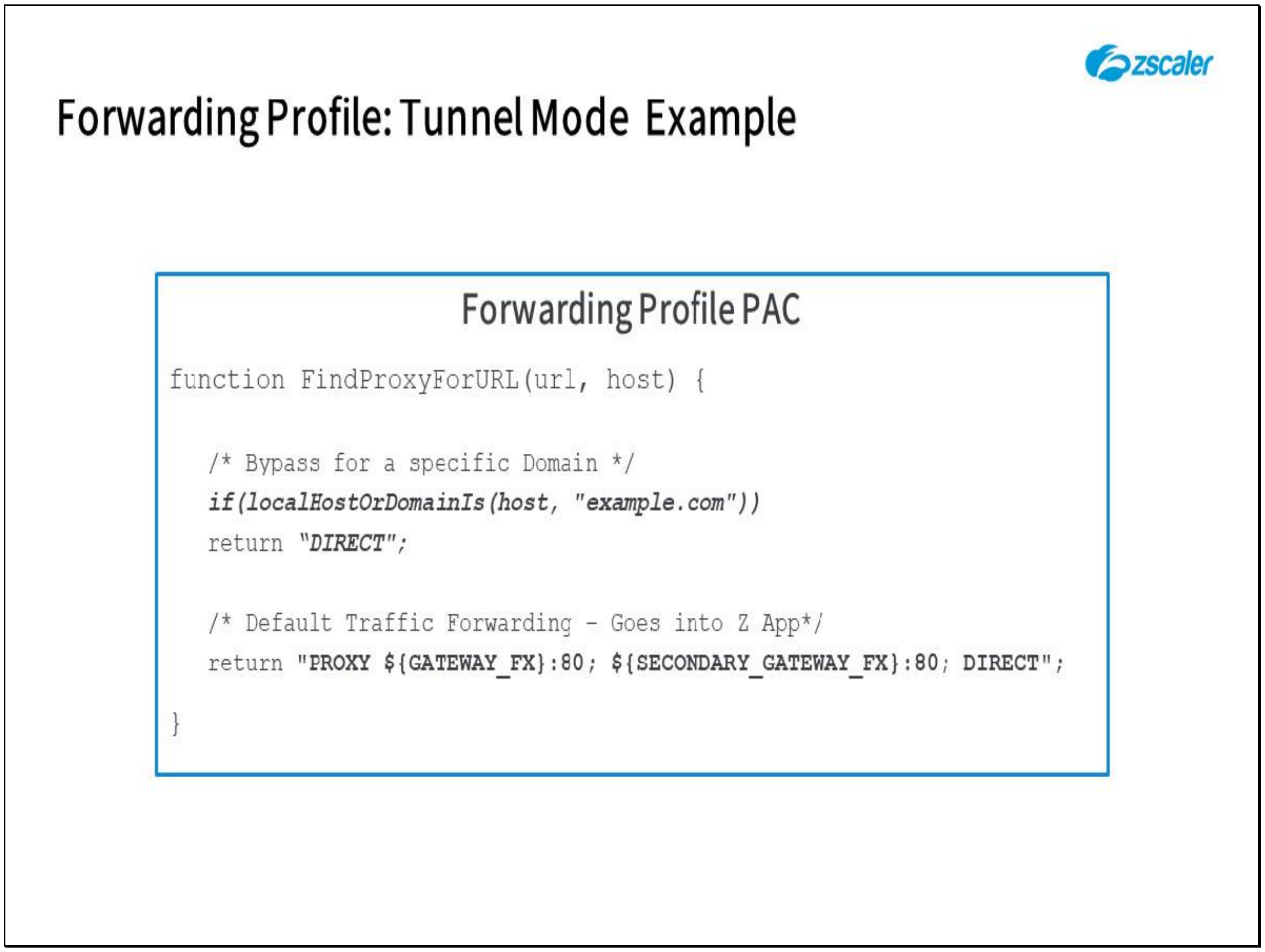
System Proxy Settings

Proxy Action Type: [✓ Enforce] Apply on Network Change | Never

Slide notes

In **Tunnel** mode (**1.0** or **2.0**) no file is applied by default, but you may provide the URL for a PAC file to be applied. The specified file is applied to the system to replace any configured proxy definitions. This file may contain destinations to bypass Zscaler App completely.

Slide 21 - Forwarding Profile: Tunnel Mode Example



Forwarding Profile: Tunnel Mode Example

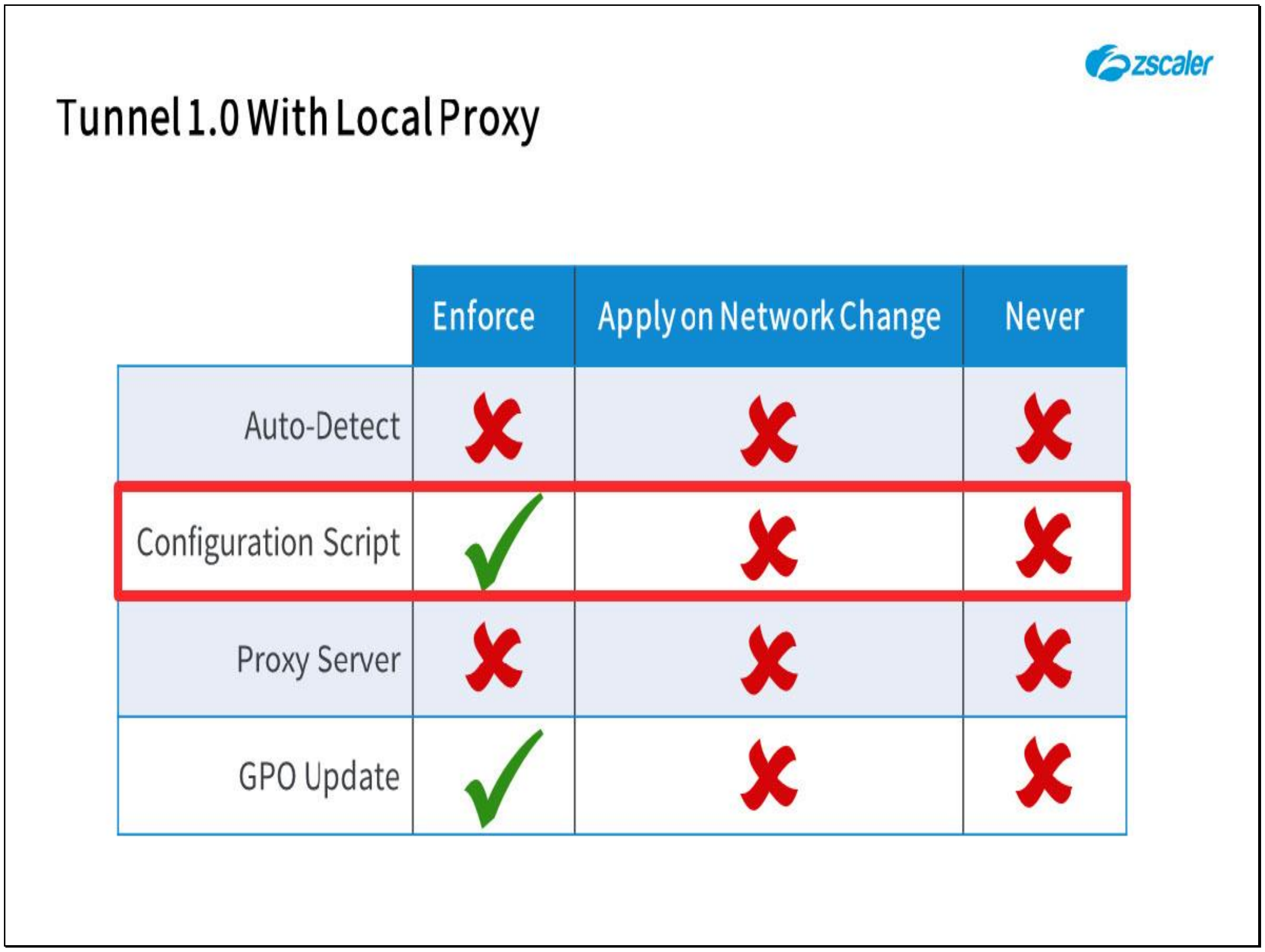
Forwarding Profile PAC

```
function FindProxyForURL(url, host) {

   /* Bypass for a specific Domain */
   if(localHostOrDomainIs(host, "example.com"))
   return "DIRECT";

   /* Default Traffic Forwarding - Goes into Z App*/
   return "PROXY ${GATEWAY_FX}:80; ${SECONDARY_GATEWAY_FX}:80; DIRECT";

}
```

Slide notes

Here is a very simple example of a **Forwarding Profile** PAC file for a **Tunnel** mode configuration, that includes a test and bypass for a specific domain. The default forwarding statements are as normal, although remember that with **Tunnel 1.0** mode **TCP** port **80**/**443** traffic is implicitly forwarded into Zscaler App and with **Tunnel 2.0** mode all **unicast IPv4 traffic** (with some 'exclusions') is 'included' by default (and sent into the App).

Slide 22 - Tunnel 1.0 With Local Proxy



Tunnel 1.0 With Local Proxy

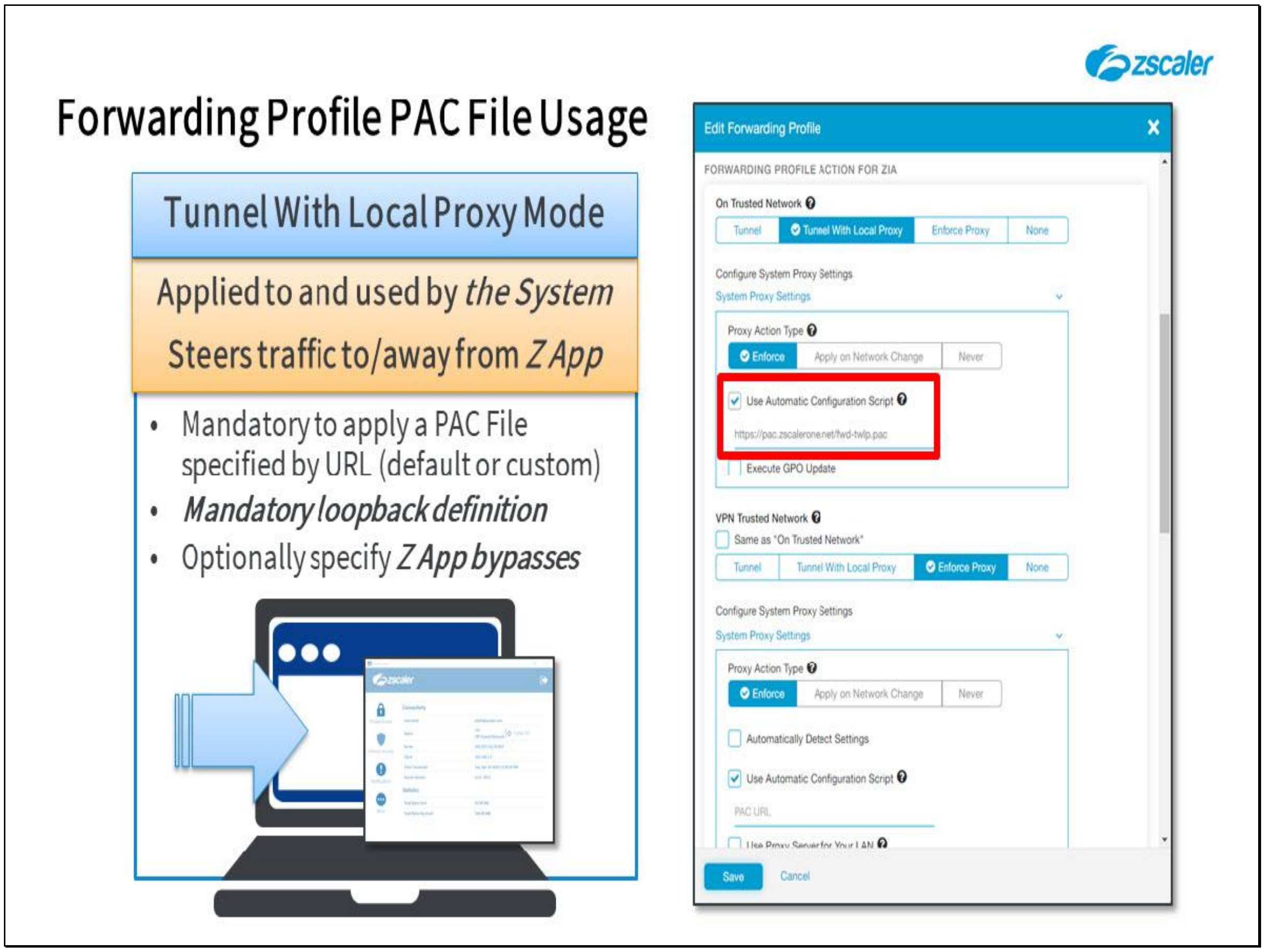
	Enforce	Apply on Network Change	Never
Auto-Detect	✗	✗	✗
Configuration Script	✓	✗	✗
Proxy Server	✗	✗	✗
GPO Update	✓	✗	✗

Slide notes

Here are the options for the **Tunnel (1.0) With Local Proxy** (**TWLP**) forwarding mode, once again, the **Forwarding Profile** PAC file option is the option highlighted in red.

In this mode the **Configuration Script** option cannot be disabled, if no custom PAC file URL is specified a default file is used with the loopback proxy macro **${ZAPP_LOCAL_PROXY}** as the default destination, to explicitly proxy **HTTP** and **HTTPS** traffic into the App. Any custom PAC file you apply here must also contain this configuration!

Slide 23 - Forwarding Profile PAC File Usage



Forwarding Profile PAC File Usage

Tunnel With Local Proxy Mode

Applied to and used by *the System*

Steers traffic to/away from *Z App*

- Mandatory to apply a PAC File specified by URL (default or custom)
- ***Mandatory loopback definition***
- Optionally specify ***Z App bypasses***

Edit Forwarding Profile

FORWARDING PROFILE ACTION FOR ZIA

On Trusted Network

Tunnel [✓ Tunnel With Local Proxy] Enforce Proxy None

Configure System Proxy Settings

System Proxy Settings

Proxy Action Type

[✓ Enforce] Apply on Network Change Never

☑ Use Automatic Configuration Script

https://pac.zscalerone.net/fwd-twlp.pac

☐ Execute GPO Update

VPN Trusted Network

☐ Same as "On Trusted Network"

Tunnel Tunnel With Local Proxy [✓ Enforce Proxy] None

Configure System Proxy Settings

System Proxy Settings

Proxy Action Type

[✓ Enforce] Apply on Network Change Never

☐ Automatically Detect Settings

☑ Use Automatic Configuration Script

PAC URL

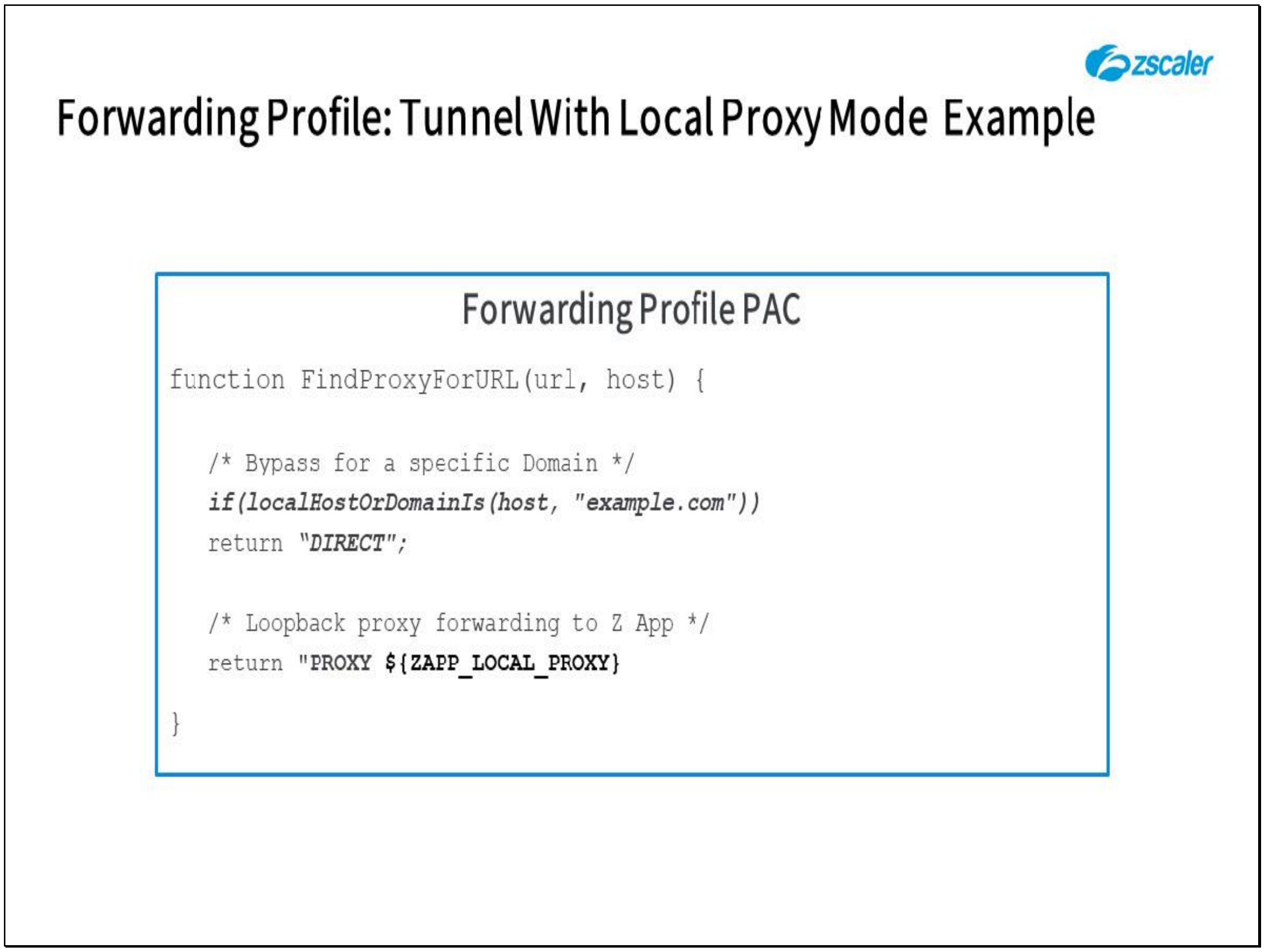
☐ Use Proxy Server for Your LAN

Slide notes

In a **Forwarding Profile** in the **TWLP** mode, the PAC file applied here is applied to the system to replace any configured proxy and is used to send traffic into the App. It may also contain destinations that are to bypass the App completely; traffic for any of the bypasses defined in this file are never seen or processed by Z App.

As before, this PAC file is refreshed every **15 minutes**.

Slide 24 - Forwarding Profile: Tunnel With Local Proxy Mode Example



Forwarding Profile: Tunnel With Local Proxy Mode Example

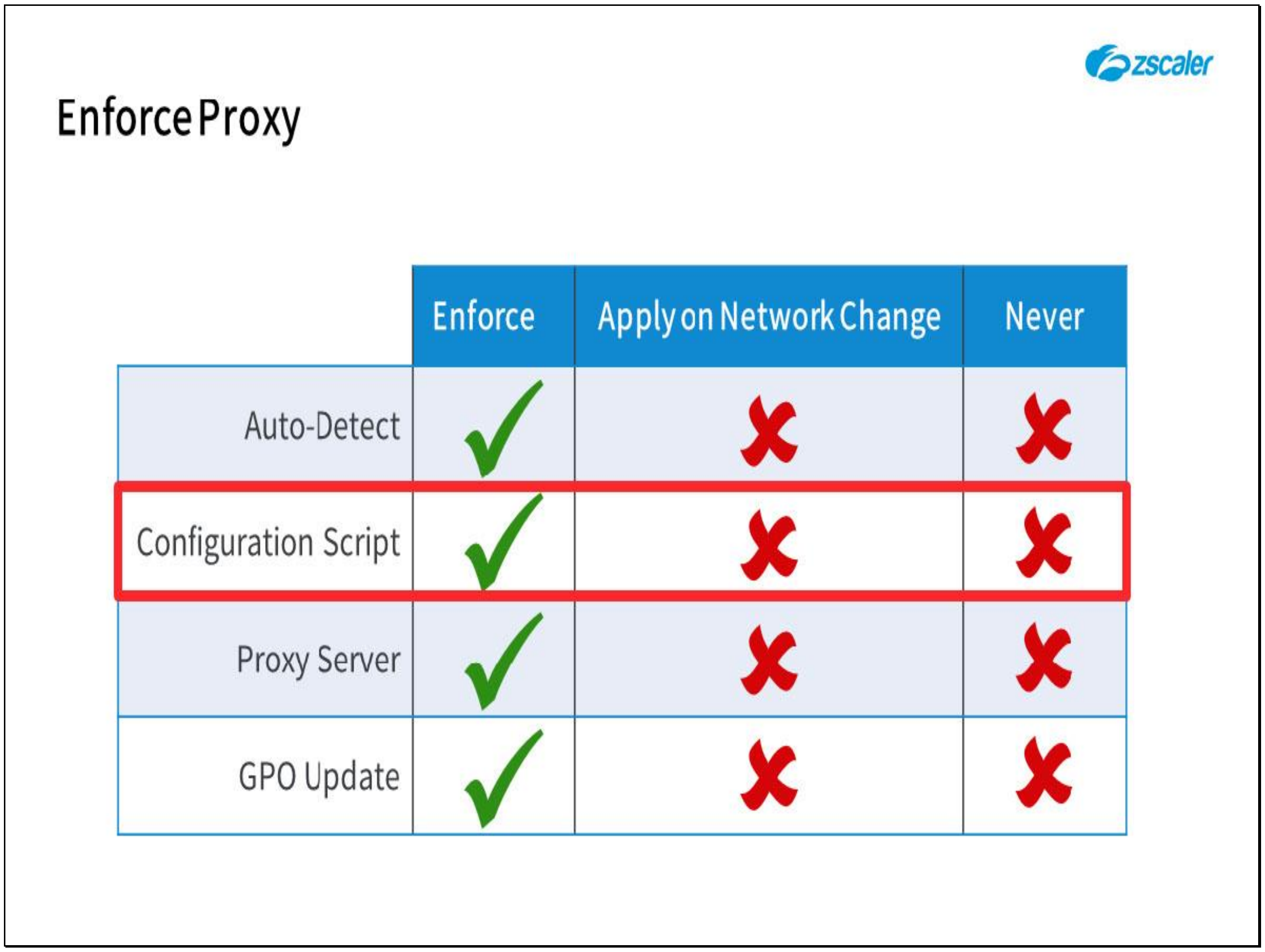
Forwarding Profile PAC

```
function FindProxyForURL(url, host) {

   /* Bypass for a specific Domain */
   if(localHostOrDomainIs(host, "example.com"))
   return "DIRECT";

   /* Loopback proxy forwarding to Z App */
   return "PROXY ${ZAPP_LOCAL_PROXY}

}
```

Slide notes

This is a very simple example of a **Forwarding Profile** PAC file for a **Tunnel (1.0) With Local Proxy** mode configuration, that also includes a test and bypass for a specific domain. The default forwarding statement MUST be the **${ZAPP_LOCAL_PROXY}** statement, to explicitly proxy **HTTP** and **HTTPS** into Zscaler App.

Slide 25 - Enforce Proxy



Enforce Proxy

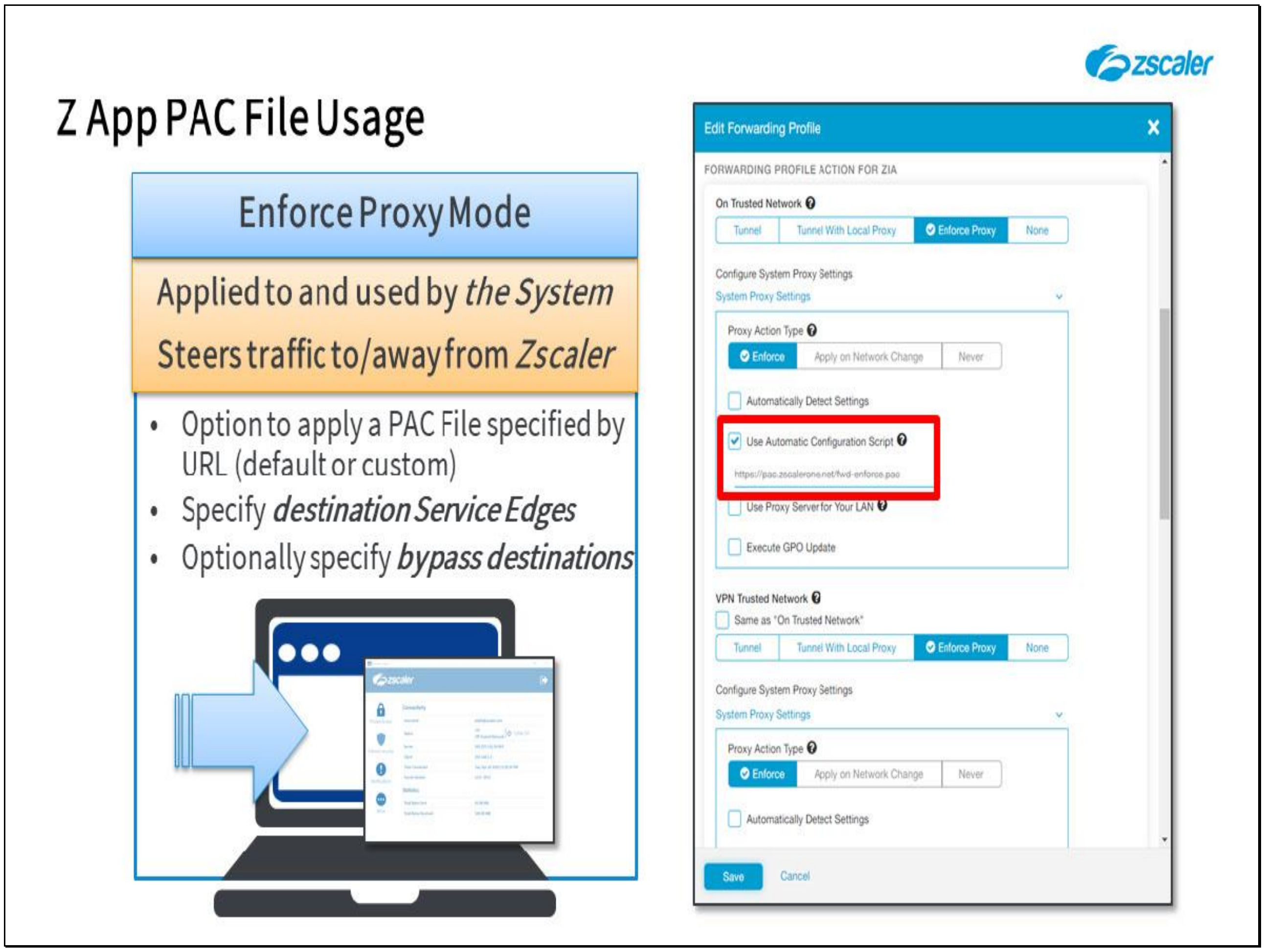
	Enforce	Apply on Network Change	Never
Auto-Detect	✓	✗	✗
Configuration Script	✓	✗	✗
Proxy Server	✓	✗	✗
GPO Update	✓	✗	✗

Slide notes

Here are the options for the **Enforce Proxy** forwarding mode, as before, the **Forwarding Profile** PAC file option is the one highlighted in red.

In this mode the **Configuration Script** option can be used to deliver a custom PAC file to replace the system proxy configuration. Traffic will be sent direct or proxied to the ZIA Service Edge nodes specified in the file. Note, there is no default file, you must create a custom file for this mode.

Slide 26 - Z App PAC File Usage



Z App PAC File Usage

Enforce Proxy Mode

Applied to and used by *the System*

Steers traffic to/away from *Zscaler*

- Option to apply a PAC File specified by URL (default or custom)
- Specify ***destination Service Edges***
- Optionally specify ***bypass destinations***

Edit Forwarding Profile

FORWARDING PROFILE ACTION FOR ZIA

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

Configure System Proxy Settings

System Proxy Settings

Proxy Action Type: [✓ Enforce] | Apply on Network Change | Never

- [ ] Automatically Detect Settings
- [x] Use Automatic Configuration Script
  https://pac.zscalerone.net/fwd-enforce.pac
- [ ] Use Proxy Server for Your LAN
- [ ] Execute GPO Update

VPN Trusted Network

- [ ] Same as "On Trusted Network"

Tunnel | Tunnel With Local Proxy | [✓ Enforce Proxy] | None

Configure System Proxy Settings

System Proxy Settings

Proxy Action Type: [✓ Enforce] | Apply on Network Change | Never

- [ ] Automatically Detect Settings

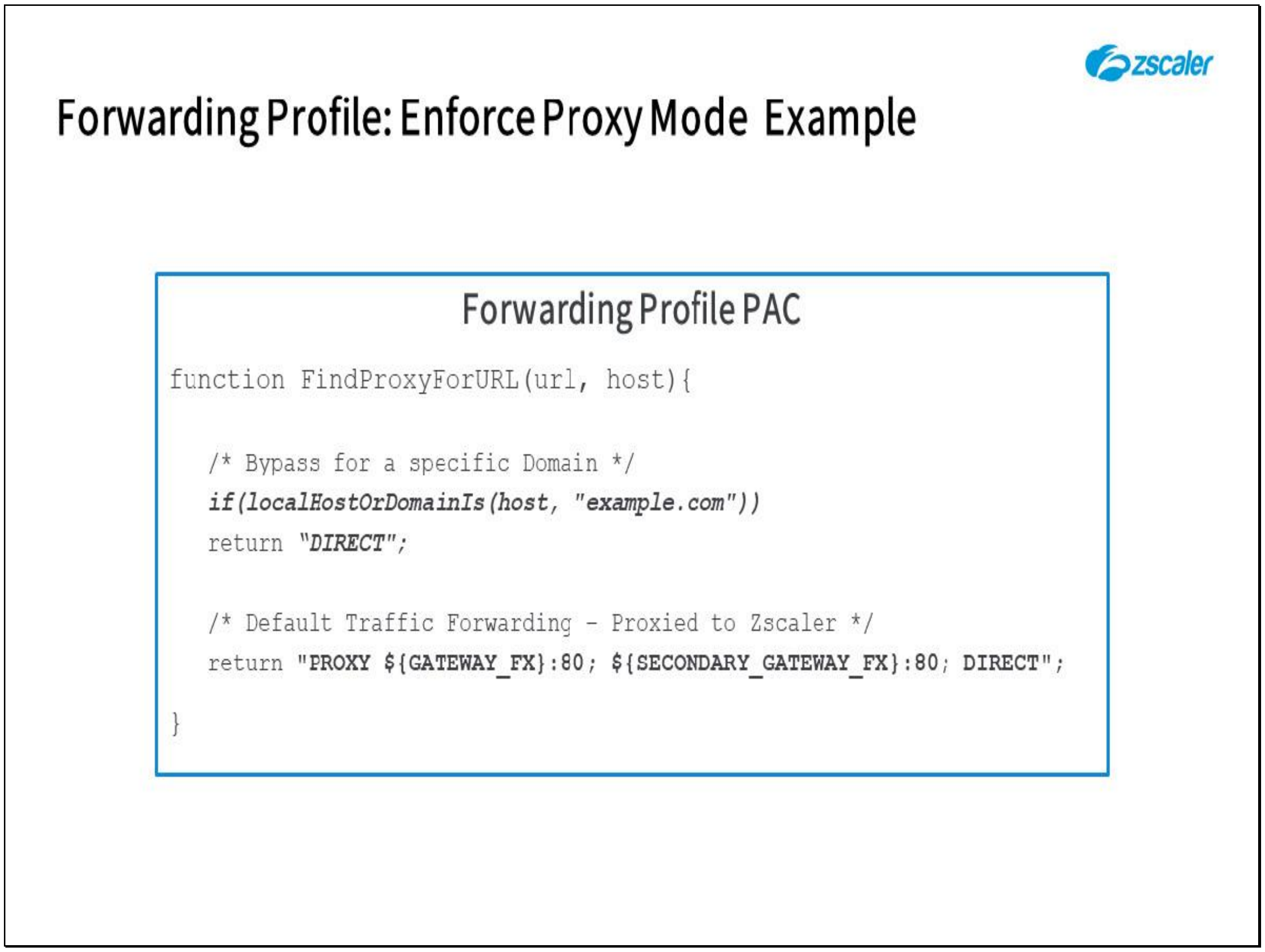
Save Cancel

Slide notes

In a **Forwarding Profile** in the **Enforce Proxy** mode, the specified PAC file is applied to the system to replace any configured proxy definitions. This file is used as a conventional PAC file to identify traffic to proxy, or to be sent direct to the destination.

The only part that Zscaler App plays in this mode is to enforce the configured settings, it does not take any part in the actual forwarding of traffic.

Slide 27 - Forwarding Profile: Enforce Proxy Mode Example



Forwarding Profile: Enforce Proxy Mode Example

Forwarding Profile PAC

```
function FindProxyForURL(url, host){

   /* Bypass for a specific Domain */
   if(localHostOrDomainIs(host, "example.com"))
   return "DIRECT";

   /* Default Traffic Forwarding - Proxied to Zscaler */
   return "PROXY ${GATEWAY_FX}:80; ${SECONDARY_GATEWAY_FX}:80; DIRECT";

}
```

Slide notes

This is a very simple example of a **Forwarding Profile** PAC file for an **Enforce Proxy** mode configuration, that includes a test and bypass for a specific domain. The default forwarding statements are at the bottom of the file using both primary and secondary gateway definitions (both with the **_FX** suffix), and a fail-open **DIRECT** statement.

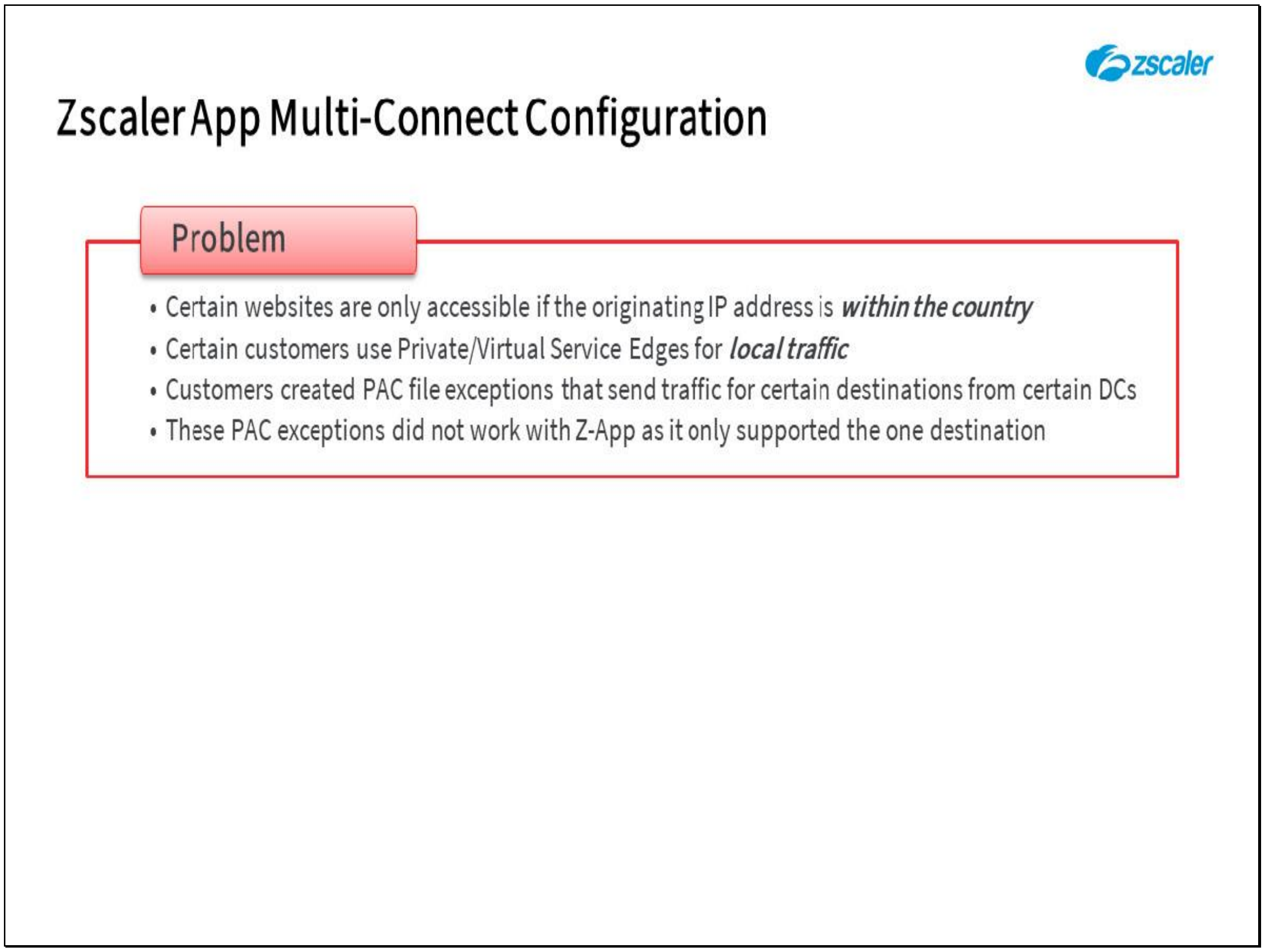
Slide 28 - Multi-Connect Configuration

Multi-Connect Configuration

Slide notes

The next topic that we will cover is a look at the **Multi-Connect** feature.

Slide 29 - Zscaler App Multi-Connect Configuration



Zscaler App Multi-Connect Configuration

Problem

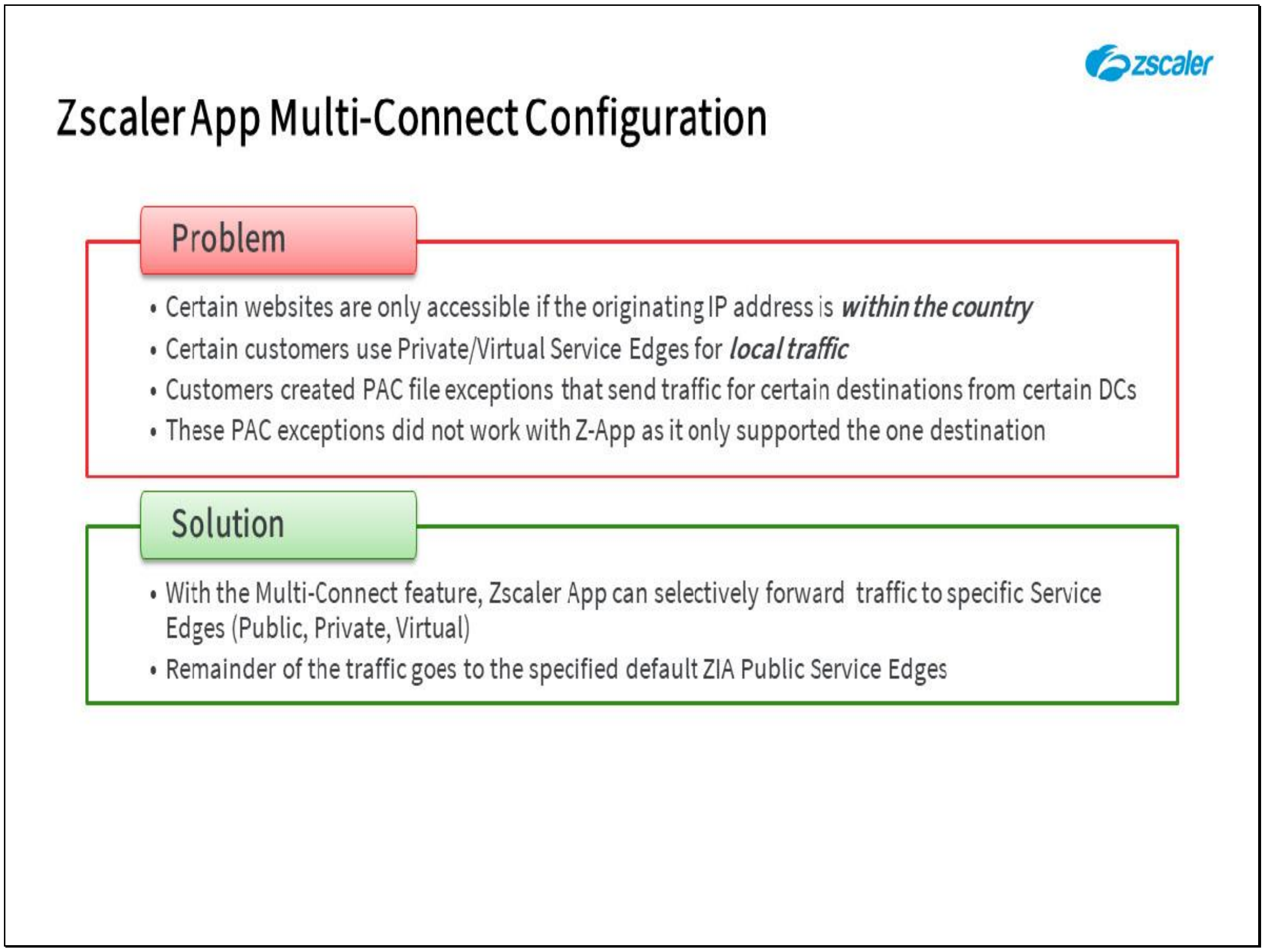
- Certain websites are only accessible if the originating IP address is ***within the country***
- Certain customers use Private/Virtual Service Edges for ***local traffic***
- Customers created PAC file exceptions that send traffic for certain destinations from certain DCs
- These PAC exceptions did not work with Z-App as it only supported the one destination

Slide notes

There are certain websites that are only accessible if the originating IP address is within the country in question. For example, the Indian Income Tax filing website only allows connections if the request has an IP address that is within India. In addition, certain customers use Private or Virtual Service Edge nodes for local traffic, e.g. they may have a Riverbed connection to O365 through a local Private Service Edge. This lets customers leverage that connection for better performance for O365 traffic but send all other traffic to the geographically closest Service Edge as normal.

In order to tackle this situation, customers have implemented PAC file exceptions that send traffic for certain destinations from certain DCs, but these PAC exceptions did not work with Z App as previously the App only supported the one return statement.

Slide 30 - Zscaler App Multi-Connect Configuration



Zscaler App Multi-Connect Configuration

Problem

- Certain websites are only accessible if the originating IP address is ***within the country***
- Certain customers use Private/Virtual Service Edges for ***local traffic***
- Customers created PAC file exceptions that send traffic for certain destinations from certain DCs
- These PAC exceptions did not work with Z-App as it only supported the one destination

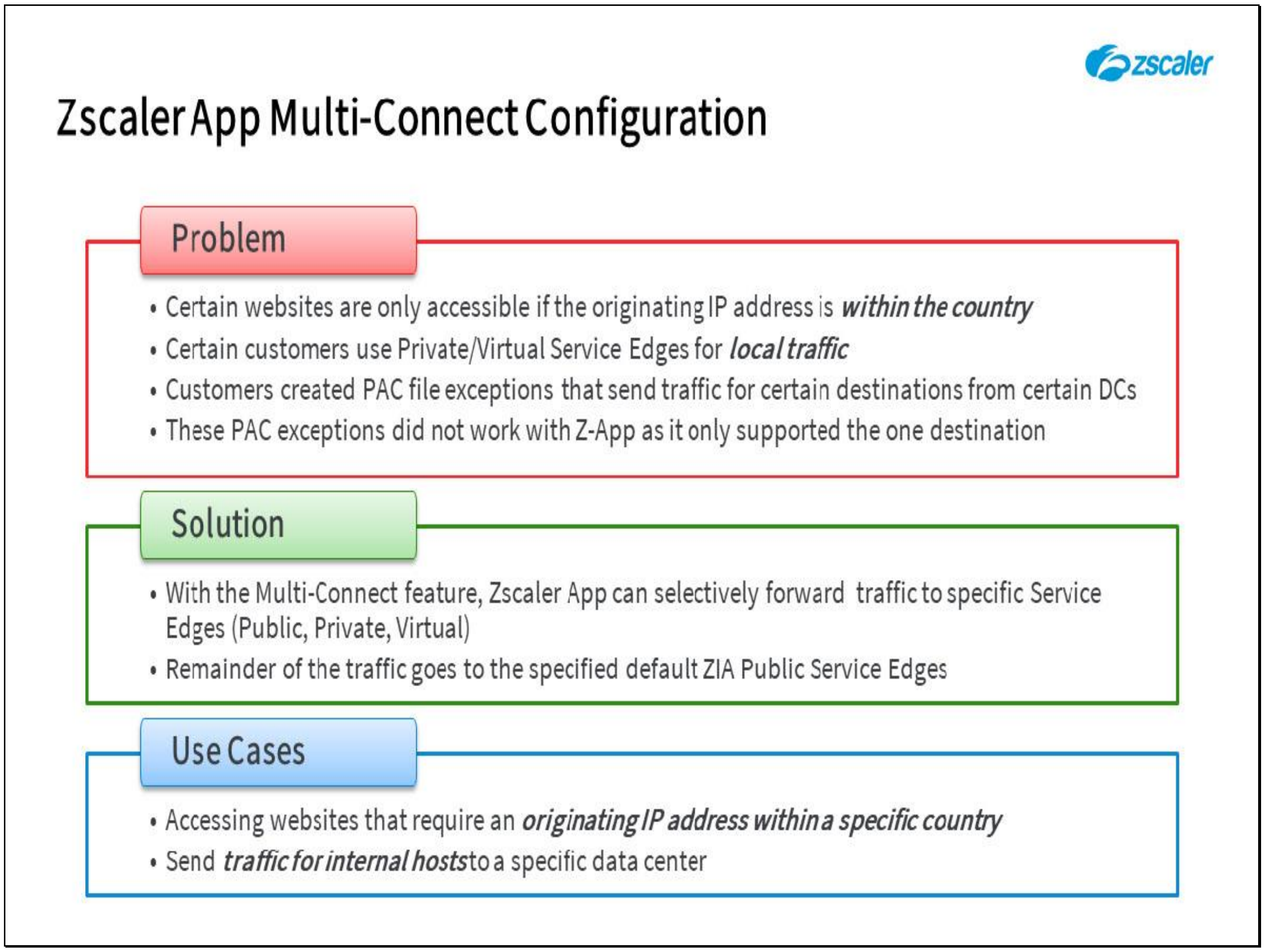
Solution

- With the Multi-Connect feature, Zscaler App can selectively forward traffic to specific Service Edges (Public, Private, Virtual)
- Remainder of the traffic goes to the specified default ZIA Public Service Edges

Slide notes

With the **Multi-Connect** feature, Zscaler App introduced the ability to forward selected traffic to specific Service Edge nodes (**Public**, **Private** or **Virtual**) and the rest of traffic to a default node.

Slide 31 - Zscaler App Multi-Connect Configuration



zscaler

Zscaler App Multi-Connect Configuration

Problem

- Certain websites are only accessible if the originating IP address is ***within the country***
- Certain customers use Private/Virtual Service Edges for ***local traffic***
- Customers created PAC file exceptions that send traffic for certain destinations from certain DCs
- These PAC exceptions did not work with Z-App as it only supported the one destination

Solution

- With the Multi-Connect feature, Zscaler App can selectively forward traffic to specific Service Edges (Public, Private, Virtual)
- Remainder of the traffic goes to the specified default ZIA Public Service Edges

Use Cases

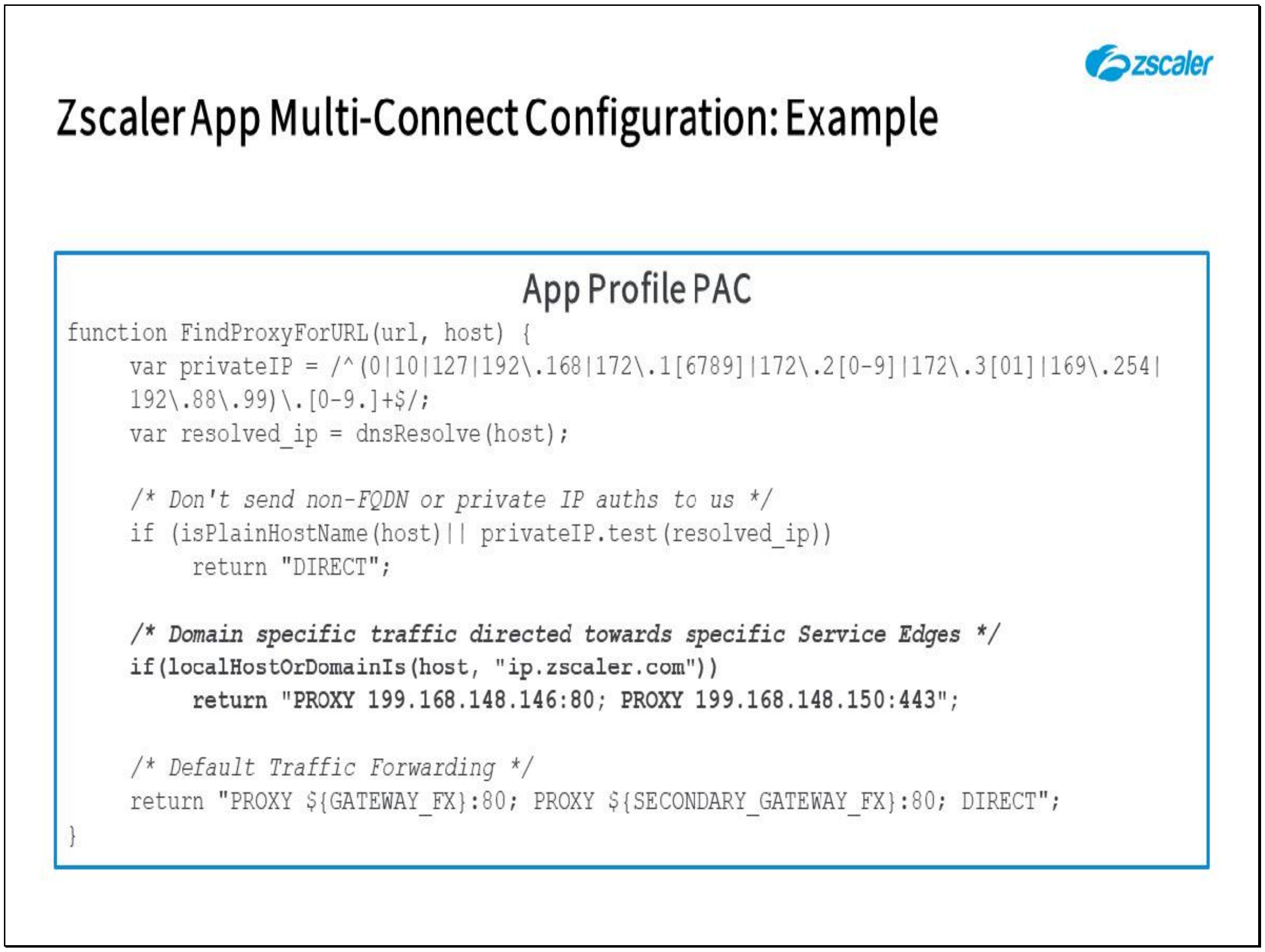
- Accessing websites that require an ***originating IP address within a specific country***
- Send ***traffic for internal hosts*** to a specific data center

Slide notes

This gives you the option to configure multiple Service Edge gateway nodes based on the destination the user is trying to access. For example:

- If you want traffic for **internal hosts** to go to a specific data center, but all other traffic to go to the geographically closest data center, you can define two return statements with the respective Service Edge gateway addresses;
- Another example would be where your end users must access a web service that requires the originating IP address to be **within a specific country**.

Slide 32 - Zscaler App Multi-Connect Configuration: Example



Zscaler App Multi-Connect Configuration: Example

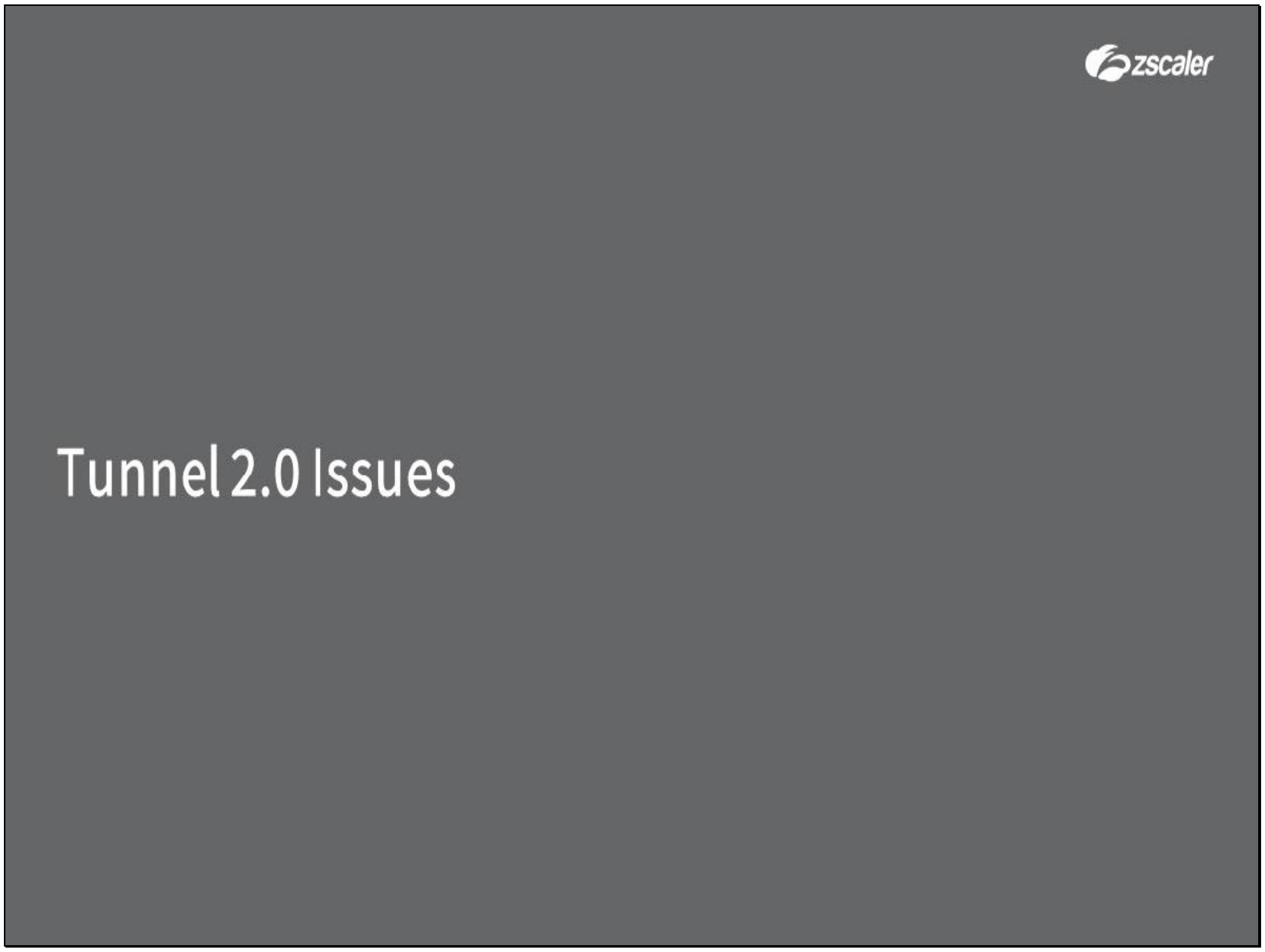
App Profile PAC

```
function FindProxyForURL(url, host) {
    var privateIP = /^(0|10|127|192\.168|172\.1[6789]|172\.2[0-9]|172\.3[01]|169\.254|
    192\.88\.99)\.[0-9.]+$/;
    var resolved_ip = dnsResolve(host);

    /* Don't send non-FQDN or private IP auths to us */
    if (isPlainHostName(host)|| privateIP.test(resolved_ip))
        return "DIRECT";

    /* Domain specific traffic directed towards specific Service Edges */
    if(localHostOrDomainIs(host, "ip.zscaler.com"))
        return "PROXY 199.168.148.146:80; PROXY 199.168.148.150:443";

    /* Default Traffic Forwarding */
    return "PROXY ${GATEWAY_FX}:80; PROXY ${SECONDARY_GATEWAY_FX}:80; DIRECT";
}
```

Slide notes

The example shown here illustrates how to configure PAC file entries to address the first use case mentioned (traffic for **internal hosts**), with all other traffic going to the default gateways as normal.

Slide 33 - Tunnel 2.0 Issues

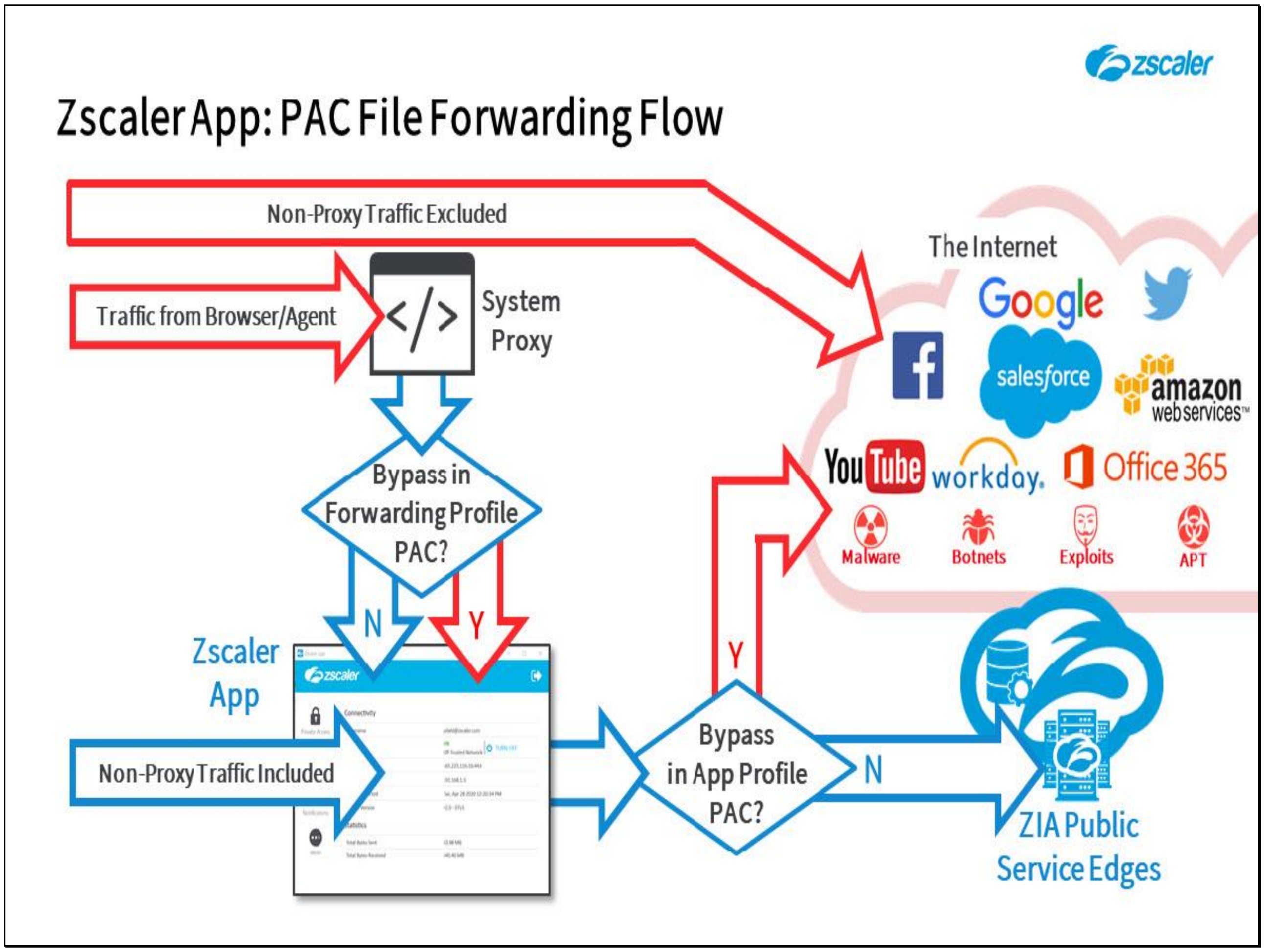


Tunnel 2.0 Issues

Slide notes

The final topic that we will cover is a look at how the PAC file behavior is modified with the **Tunnel 2.0** forwarding method.

Slide 34 - Zscaler App: PAC File Forwarding Flow


zscaler
Zscaler App: PAC File Forwarding Flow
Non-Proxy Traffic Excluded
The Internet
Traffic from Browser/Agent
System Proxy
Google
salesforce
amazon web services
YouTube
workday.
Office 365
Malware
Botnets
Exploits
APT
Bypass in Forwarding Profile PAC?
N
Y
Zscaler App
Y
Non-Proxy Traffic Included
Bypass in App Profile PAC?
N
ZIA Public Service Edges

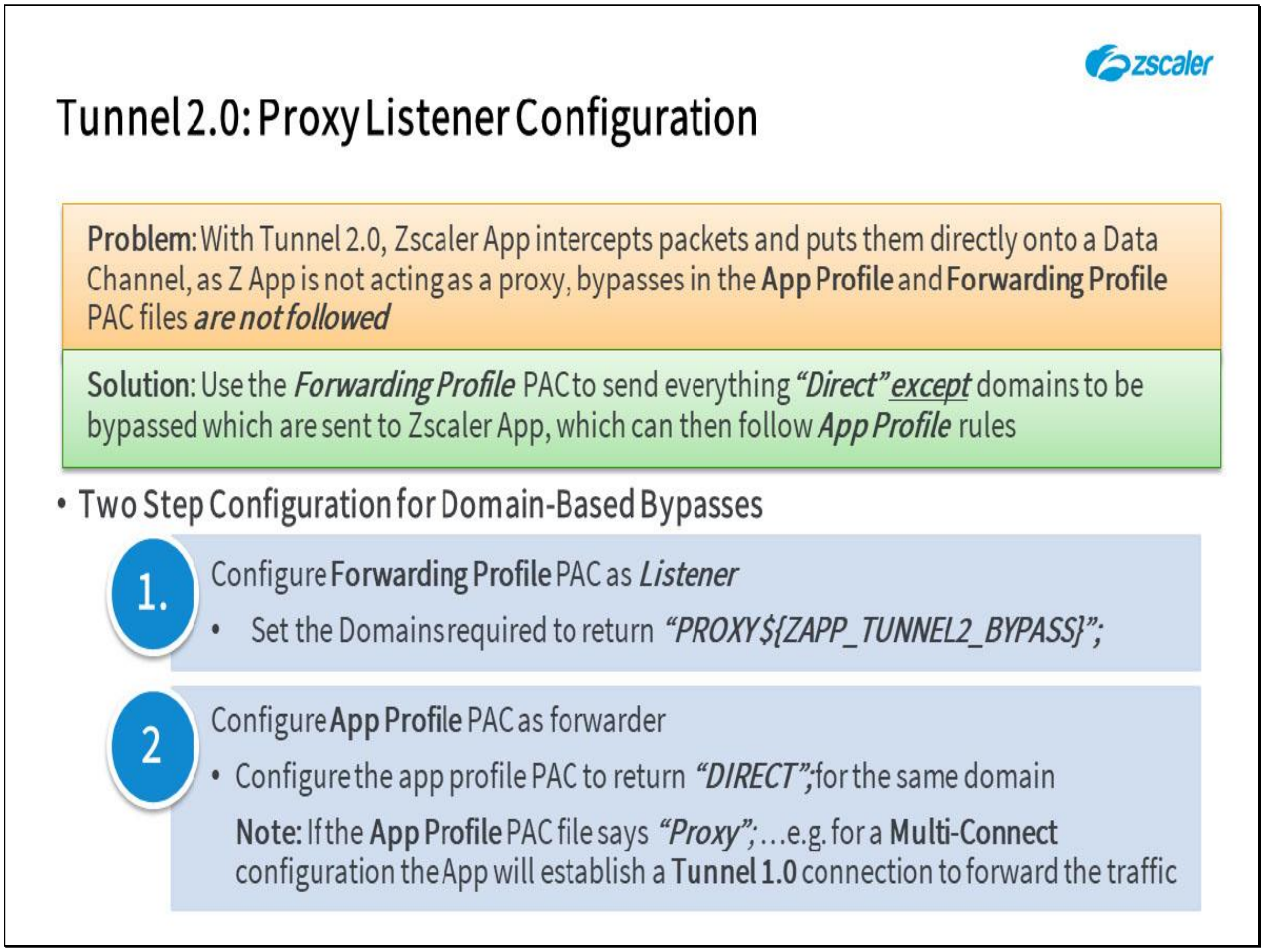
Slide notes

As we saw in the **Tunnel 2.0** module, the **Tunnel 2.0** forwarding method modifies the way PAC files are used and how you must define domain-based Zscaler App bypasses. With this method, by default, all unicast IPv4 traffic (with some IP-based **Exceptions**) is sent into Zscaler App.

You still have the option to apply a **Forwarding Profile** PAC file, however anything that you want to bypass must now be proxied explicitly into the App using the **${ZAPP_TUNNEL2_BYPASS}** macro. This means that all traffic processed by this PAC file will actually be sent into the App.

The bypass configuration (for the same domains proxied in the **Forwarding Profile** PAC) must be done in the **App Profile** PAC file.

Slide 35 - 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

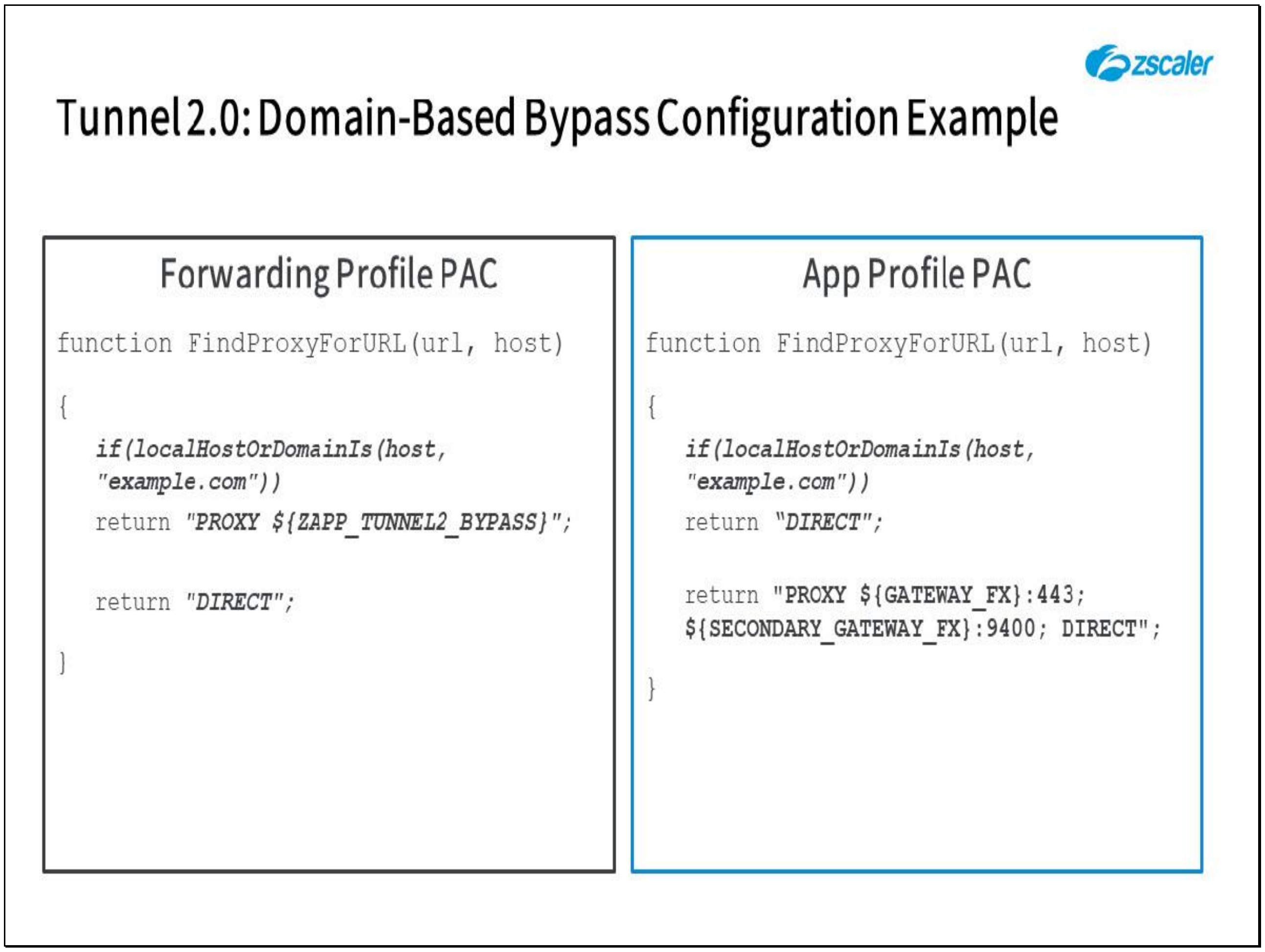
   Note: If the **App Profile** PAC file says *"Proxy";*…e.g. for a **Multi-Connect** configuration the App will establish a **Tunnel 1.0** connection to forward the traffic

Slide notes

Remember, in the **Tunnel 2.0** forwarding mode, Z App behaves as a pseudo-VPN client and 'includes' or 'excludes' traffic at the IP Layer, which means it has no native way to recognize domain-based addresses or URLs for special treatment. The resolution to this is a two-step configuration process to:

1. Explicitly proxy traffic for specific FQDNs, domains or URLs into the Zscaler App using the **${ZAPP_TUNNEL2_BYPASS}** macro in the **Forwarding Profile** PAC file, and…
2. Bypass the same FQDNs or domains in the **App Profile** PAC file.

Slide 36 - Tunnel 2.0: Domain-Based Bypass Configuration Example



Tunnel 2.0: Domain-Based Bypass Configuration Example

Forwarding Profile PAC

```
function FindProxyForURL(url, host)

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

   return "DIRECT";

}
```

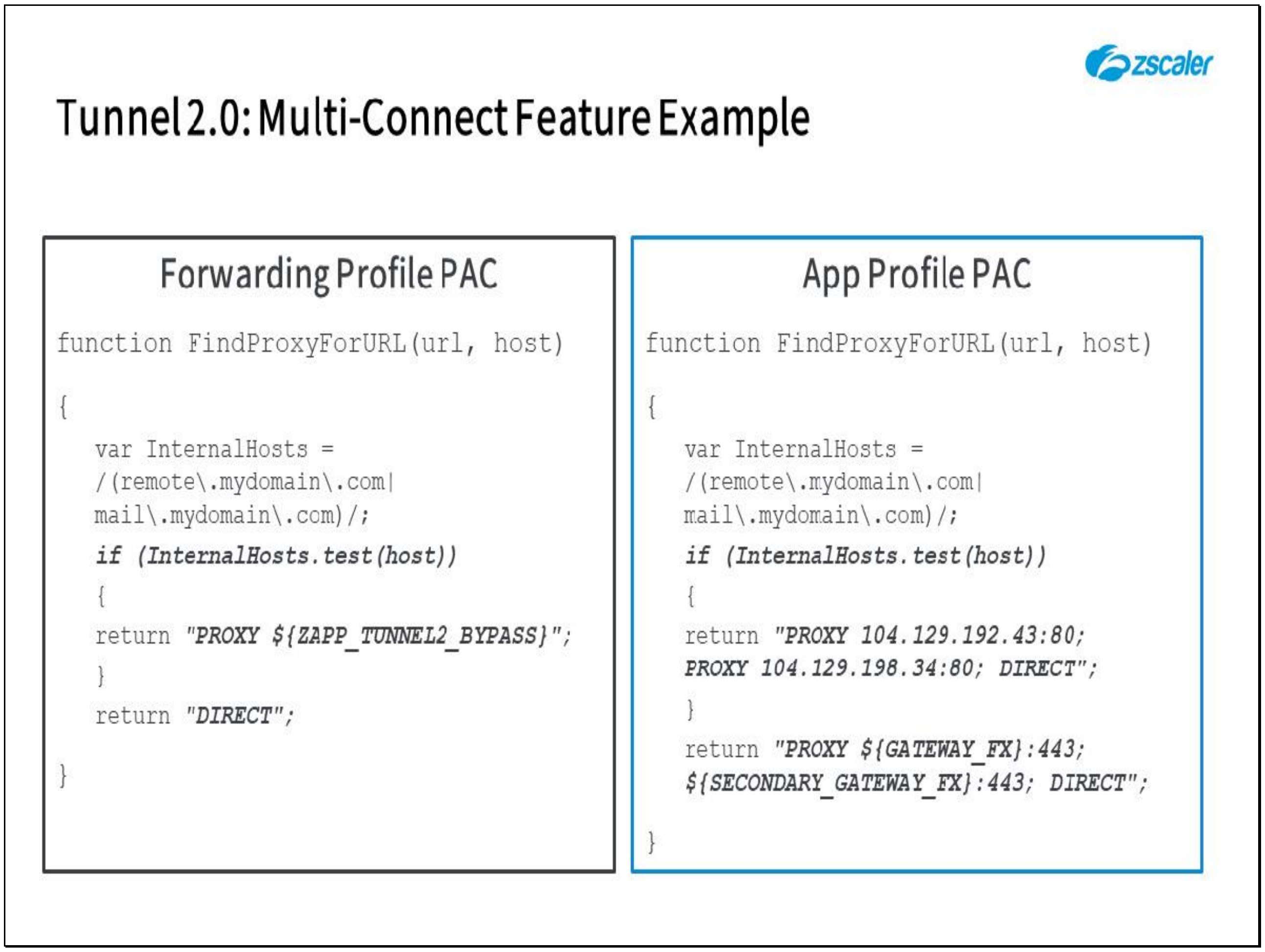
App Profile PAC

```
function FindProxyForURL(url, host)

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

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

}
```

Slide notes

Here is an illustration of this process, with an explicit loopback proxy specification for the **example.com** domain in the **Forwarding Profile** PAC file and a bypass for the same domain in the **App Profile** PAC file.

Slide 37 - Tunnel 2.0: Multi-Connect Feature Example



Tunnel 2.0: 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";

}
```

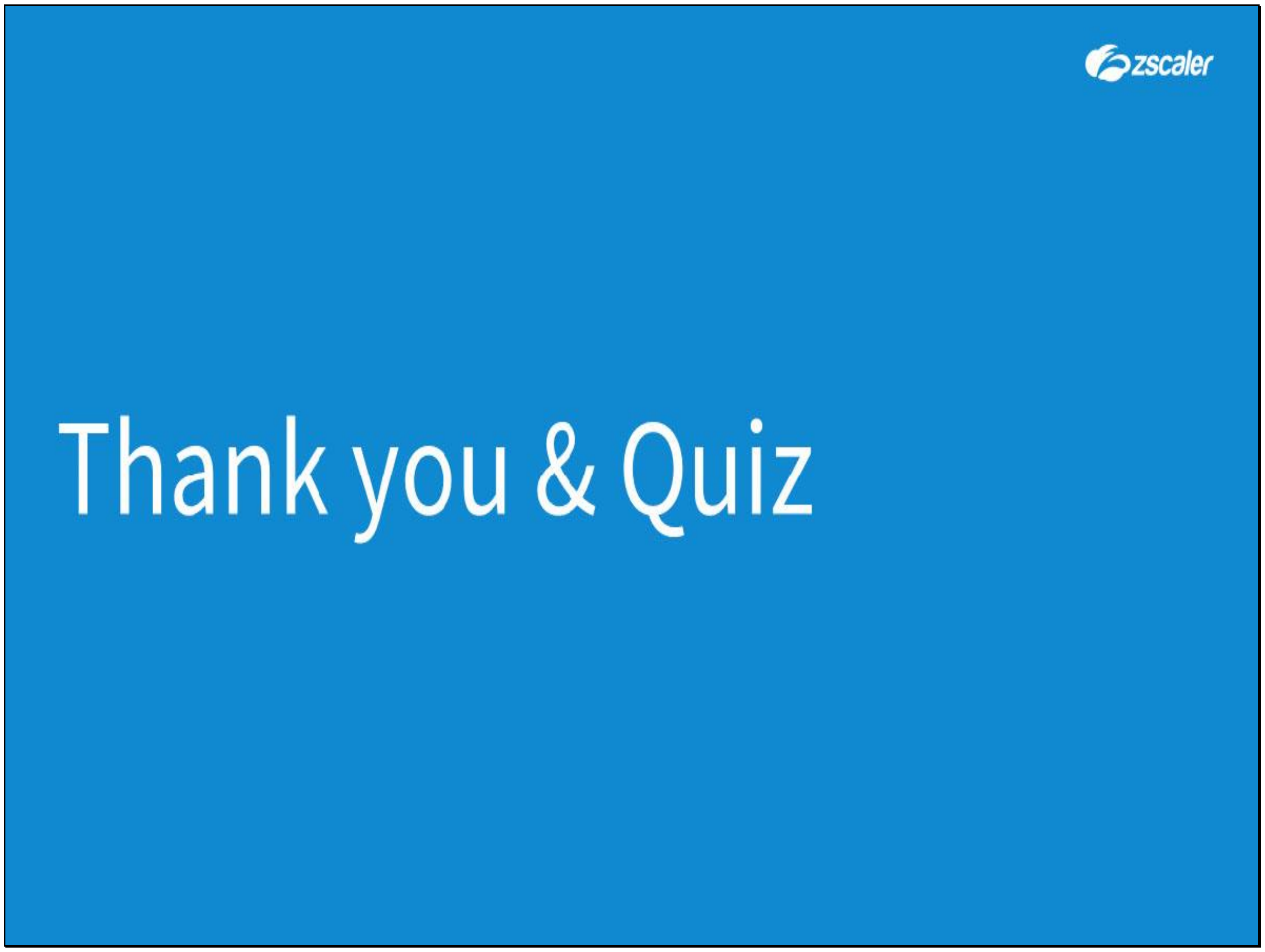
App Profile PAC

```
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

The same process must be used to support the **Multi-Connect** feature in a **Tunnel 2.0** environment. You will need an explicit loopback proxy specification for traffic to be forwarded to some other destination (in this case internal hosts) in the **Forwarding Profile** PAC file; and a specific forwarding statement for that same traffic in the **App Profile** PAC file. Remember though, that any traffic forwarded in this way uses the **Tunnel 1.0** forwarding method (**HTTP CONNECT** tunnels).

Slide 38 - Thank you & Quiz



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.