


Slide 1 - Traffic Forwarding



Traffic Forwarding

PAC Files

©2018 Zscaler, Inc. All rights reserved.

Slide notes

Welcome to this ZCCP Internet Access training module on the use of Proxy Auto-Configuration (PAC) files.


Slide 2 - Navigating the eLearning Module

Navigating the eLearning Module

The screenshot shows the Zscaler Cloud Portal dashboard. At the top right is the Zscaler logo. Below it is a navigation bar with links: Dashboard, Analytics, Policy, and Administration. The main content area is titled 'Web Overview' and contains several charts and tables. Overlaid on the bottom of the dashboard are several blue callout boxes with white text, each pointing to a specific control on the video player interface. These controls include: 'Previous Slide', 'Next Slide', 'Play/Pause', 'Fast Forward', 'Progress Bar', 'Audio On/Off', 'Closed Captioning', and 'Exit' (located at the top right of the video player frame).

Slide notes

Here is a quick guide to navigating this module. There are various controls for playback including play and pause, previous, next slide and fast forward. 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

Agenda

- What are PAC Files?
- Where to Host PAC Files?
- PAC File Syntax
- PAC File Use Cases
- Anatomy of a PAC File
- PAC File Best Practices

Slide notes

In this module, we will: look at what PAC files are and what they do; look at where best to host your PAC files; we will look at the syntax for PAC files; have a look at some PAC file use cases with the Zscaler infrastructure; examine the anatomy of a simple PAC file; and finally, we will look at some Zscaler best practices for PAC files.


Slide 4 - What are PAC Files?



Slide notes

In the first section, we have a look at what PAC files are, and at what they can do for you.

Slide 5 - What Are PAC Files?



What Are PAC Files?

Proxy Auto-Config File


- Used to automatically define the proxy server to use for a given URL
- The first file fetched by the Browser

Slide notes

A Proxy Auto-Configuration (PAC) file contains a set of rules coded in JavaScript which allows a Browser to determine whether to send Web traffic direct to the Internet, or that it be sent to a proxy server, to be forwarded to the Internet. A PAC file can control how a Browser handles HTTP, HTTPS, and FTP traffic, and is the first file fetched by the Browser.

Normally a PAC file is given the file extension **.pac**, and although this is not a requirement, many applications will not honor, or parse the file if it does not have the **.pac** extension. The MIME type for the file should be set to **application/x-ns-proxy-autoconfig**.

Slide 6 - What Are PAC Files?



What Are PAC Files?

- Proxy Auto-Config File**
 - Used to automatically define the proxy server to use for a given URL
 - The first file fetched by the Browser
- Javascript Functions**
 - Javascript file containing the function **FindProxyForURL(url, host)** together with access method specifications
 - Direct**: contact the target URL directly without going through a proxy
 - Proxy**: contact the target URL through the specified proxy
 - The PAC file may contain **If** statements and functions to manage traffic to different destination hosts
 - A PAC file may contain **domain-**, **host-**, or **time-based** statements

Slide notes


A PAC file is a JavaScript file containing the function **FindProxyForURL(url, host)** together with access method specifications:

- A **Direct** statement indicates that the target URL can be accessed directly without going through a proxy;
- A **Proxy** statement specifies that the URL must be reached through the specified proxy.

Multiple proxies may be specified separated by a semicolon (;), the left-most setting will be tried first, until the Browser fails to establish a connection to the proxy (based on a TCP timeout), when the next value will be tried, and so on. The Browser will automatically retry a previously unresponsive proxy after an interval (typically 30 minutes).

The PAC file may contain conditional **If** statements, and other functions to manage the traffic for different destination hosts, and may contain domain-, host-, or time-based conditions.

Slide 7 - What Are PAC Files?



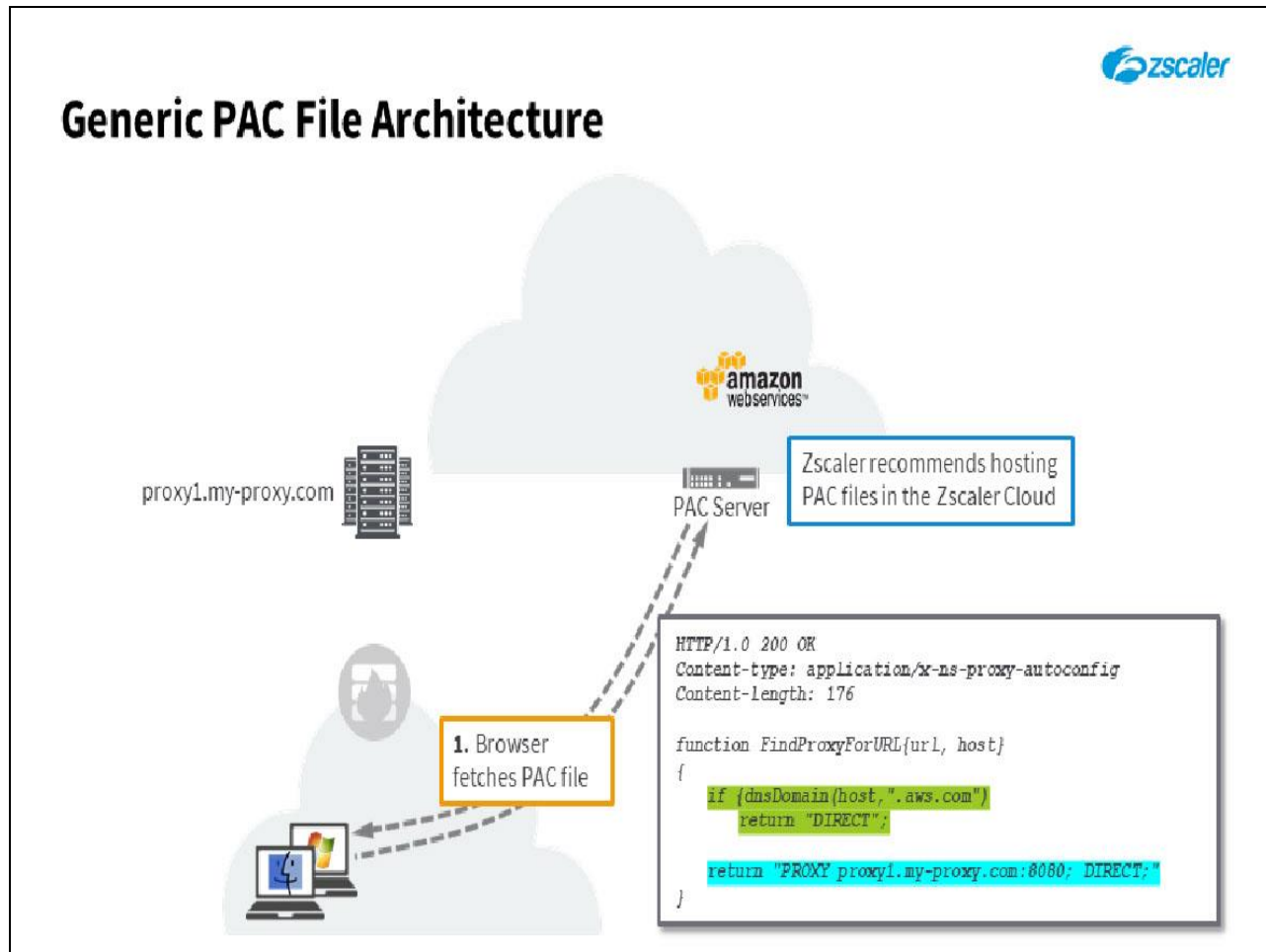
What Are PAC Files?

- Proxy Auto-Config File**
 - Used to automatically define the proxy server to use for a given URL
 - The first file fetched by the Browser
- Javascript Functions**
 - Javascript file containing the function **FindProxyForURL(url, host)** together with access method specifications
 - **Direct**: contact the target URL directly without going through a proxy
 - **Proxy**: contact the target URL through the specified proxy
 - The PAC file may contain **If** statements and functions to manage traffic to different destination hosts
 - A PAC file may contain **domain-, host-, or time-based** statements
- Deployment**
 - The PAC file URL can be statically defined, or found using Web Proxy Auto-discovery Protocol
 - If the PAC file cannot be found, loaded, or read the Browser fails over to a direct connection

Slide notes

The PAC file should be the first content fetched by the Browser, and a client machine must be told where to find it. The PAC file URL can be statically defined on the Browser (which is the recommended method) or found using the Web Proxy Auto-discovery Protocol (WPAD). If the PAC file cannot be found, cannot be loaded, or properly interpreted, the Browser should fail-over to a direct connection for all traffic (in other words it should fail open).

Slide 8 - Generic PAC File Architecture

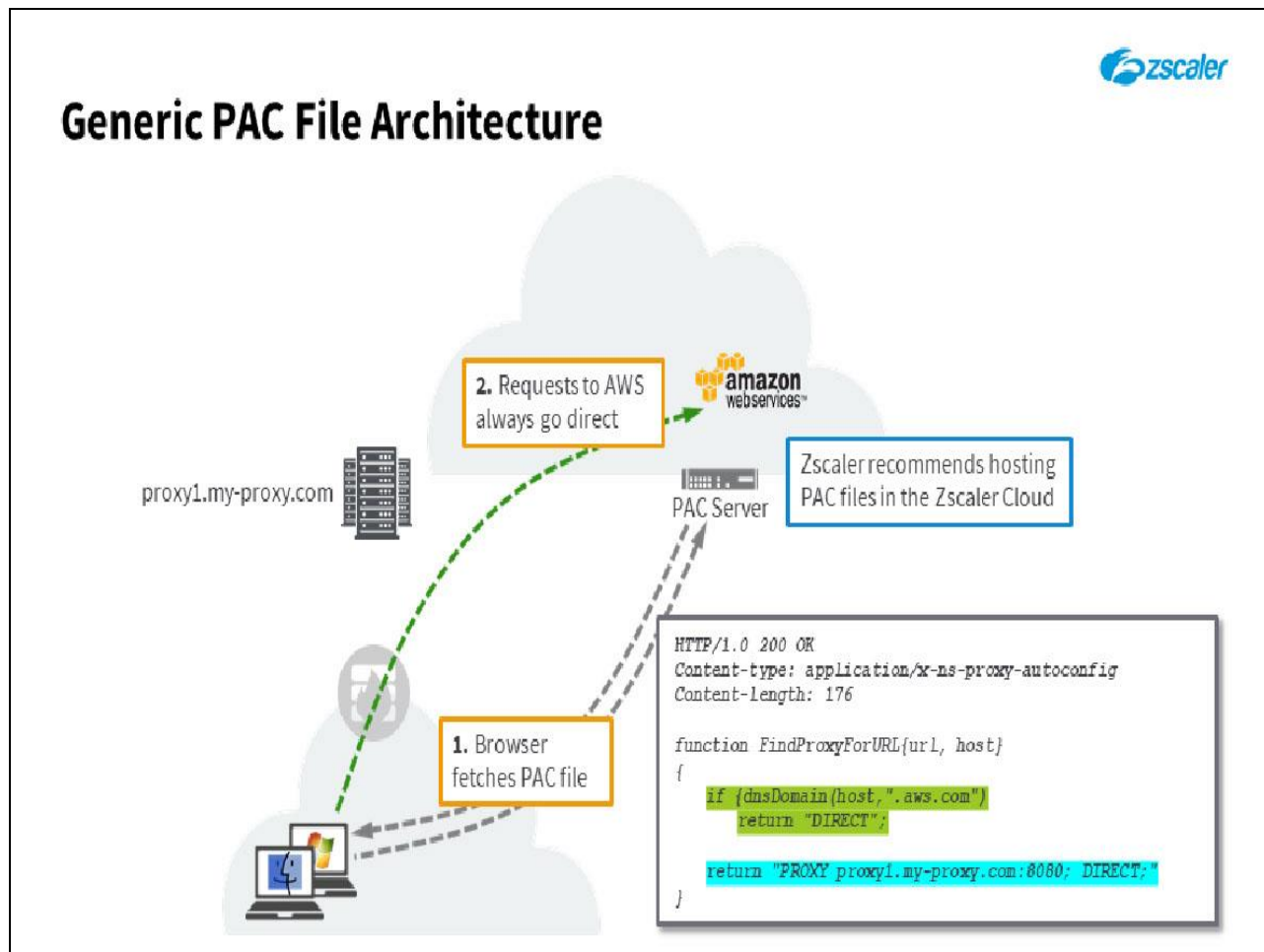


Slide notes

Here is an overview of how PAC files work in general, in this case we are going to specify that Browser calls to the Amazon Web Services domain should go direct, while all other calls should go through the Proxy at **proxy1.myproxy.com** on port **8080**.

1. The first step is for the Browser to fetch the PAC file, as it has been directed to do in the Browser settings. The URL for the PAC file may be statically defined (which is recommended), pushed to the Browser using a Microsoft AD Group Policy Object (GPO), or discovered using an auto-discovery protocol. The file itself may be hosted on an internally managed server, or on the Internet. We recommend that you host your PAC files on our infrastructure, for reasons that will become clear later.

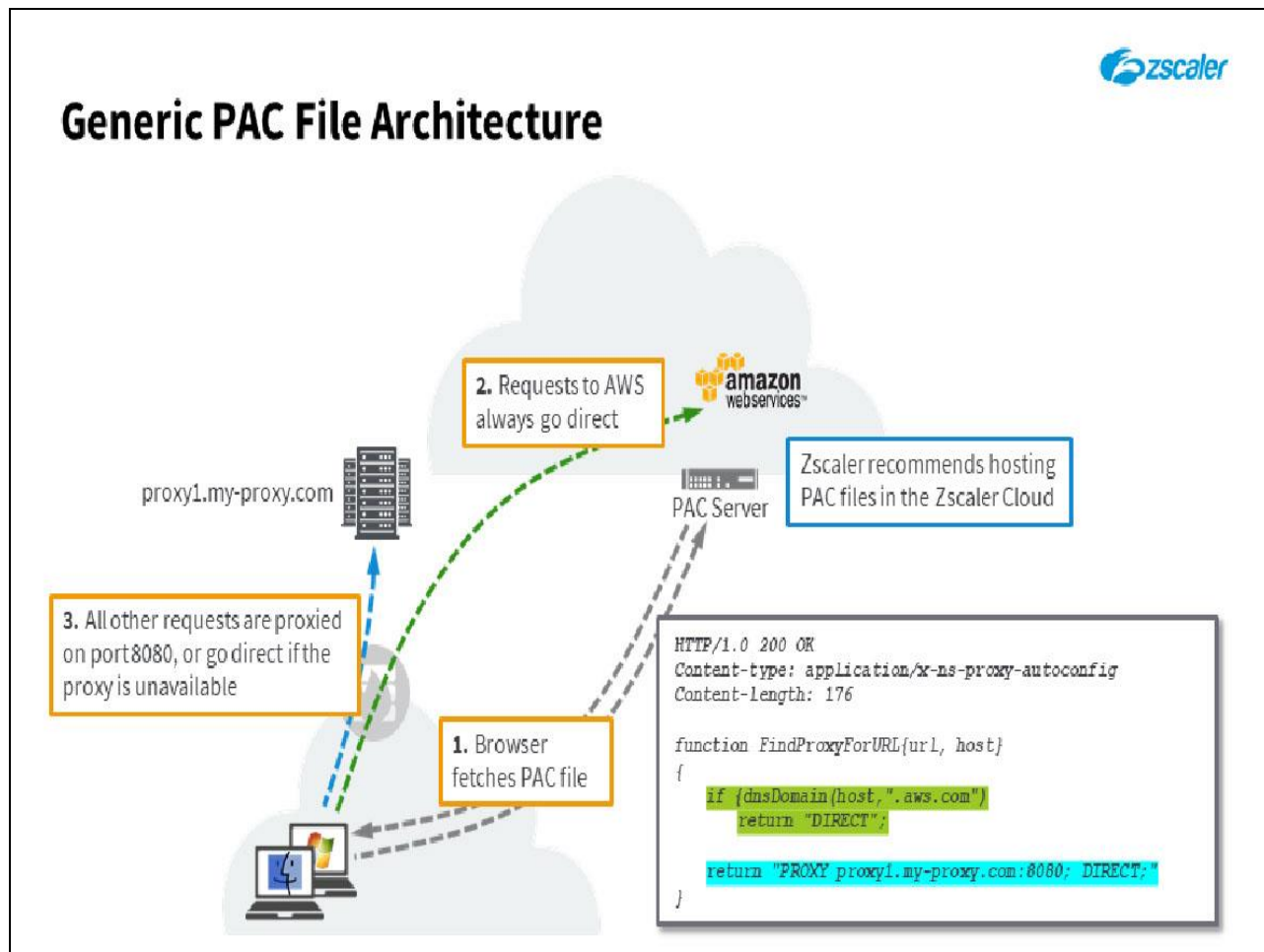
Slide 9 - Generic PAC File Architecture



Slide notes

2. Having retrieved the file, the Browser can parse it to understand the JavaScript statements that it contains. In this case it sees that if the URL requested is on the domain **.aws.com**, then it should access the site directly.


Slide 10 - Generic PAC File Architecture



Slide notes


- For any other Website, the PAC file directs that they must be accessed through the Proxy at **proxy1.mypoxy.com**, using the destination port **8080**.

Slide 11 - Browser PAC File Processing



Browser PAC File Processing

Browser tries to fetch PAC file

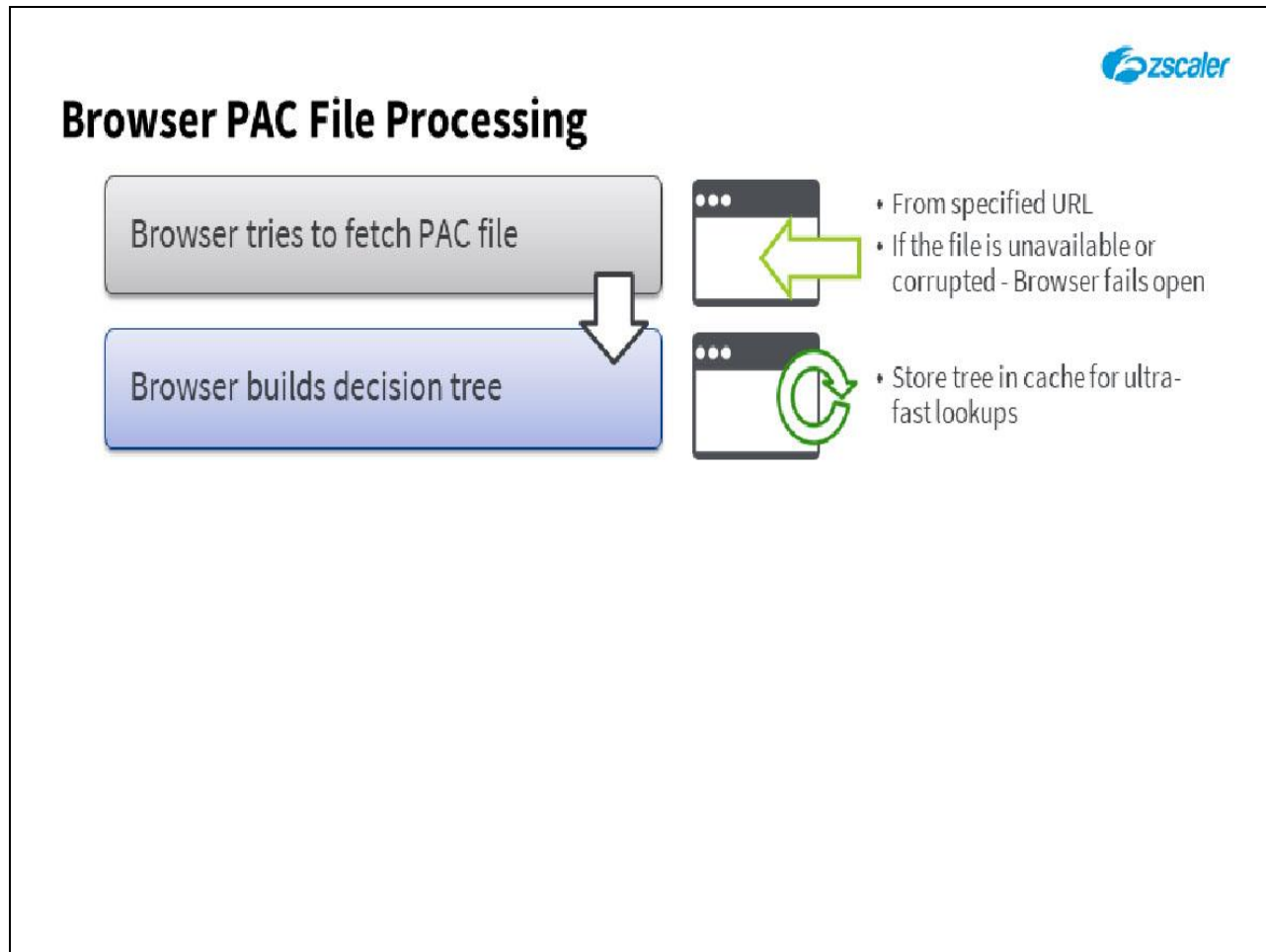


- From specified URL
- If the file is unavailable or corrupted - Browser fails open

Slide notes

A Browser must be configured to retrieve a PAC file, and this may be done using either a static URL definition (which may be provisioned centrally using an AD GPO), or some form of auto-discovery method such as WPAD. If the Browser is unable to retrieve the file, or cannot interpret the retrieved file, the Browser will 'fail open' and send all traffic direct to the requested URLs.

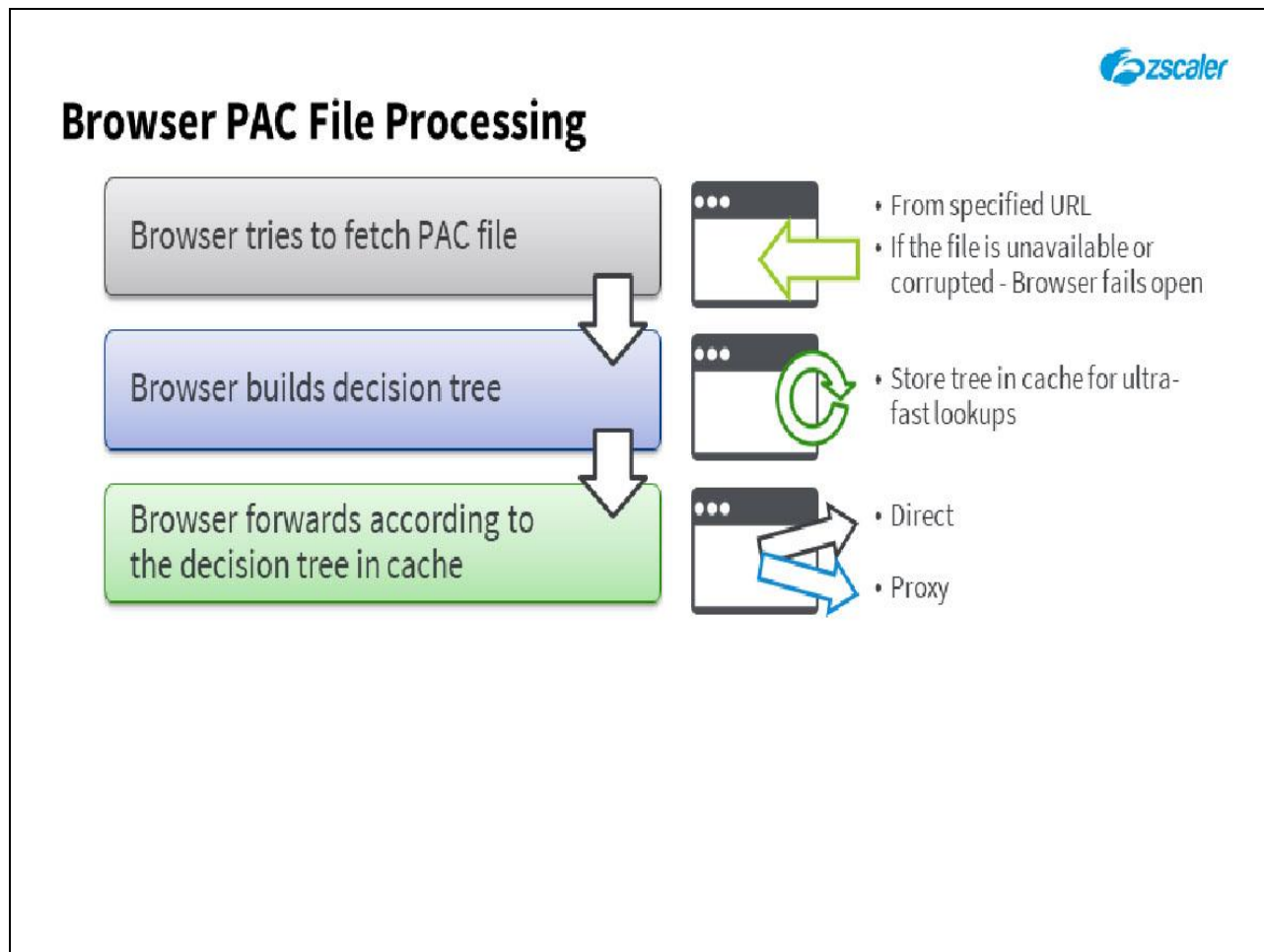
Slide 12 - Browser PAC File Processing



Slide notes

Having retrieved the PAC file, the Browser will parse it to build a decision tree that it then stores in cache. This allows the Browser to make ultra-fast forwarding decisions, without having to keep referring to the PAC file.

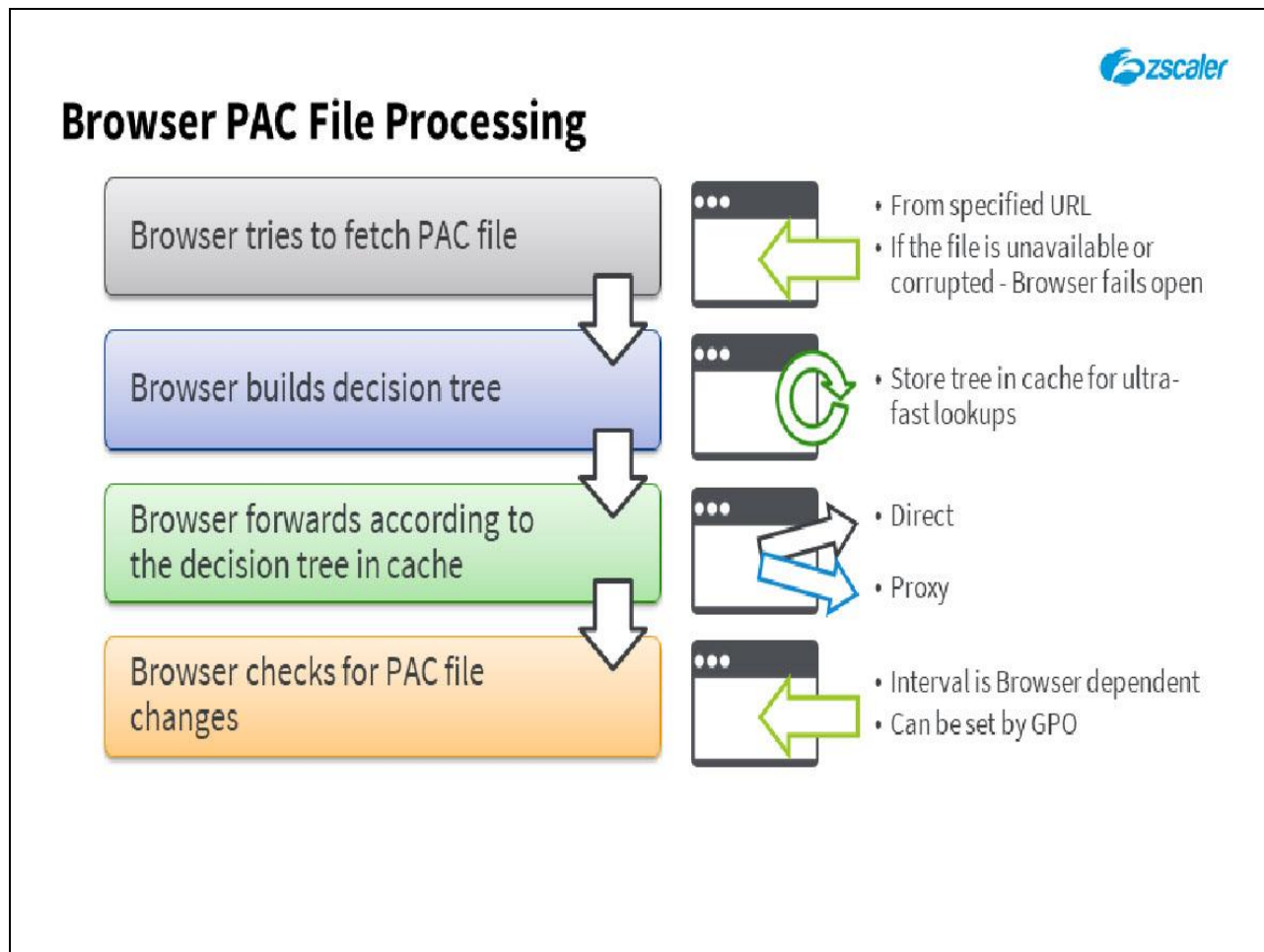
Slide 13 - Browser PAC File Processing



Slide notes

The Browser then forwards traffic based on the logic from the PAC file, some traffic may need to go direct, although most traffic will probably need to go via the specified proxy, or proxies.

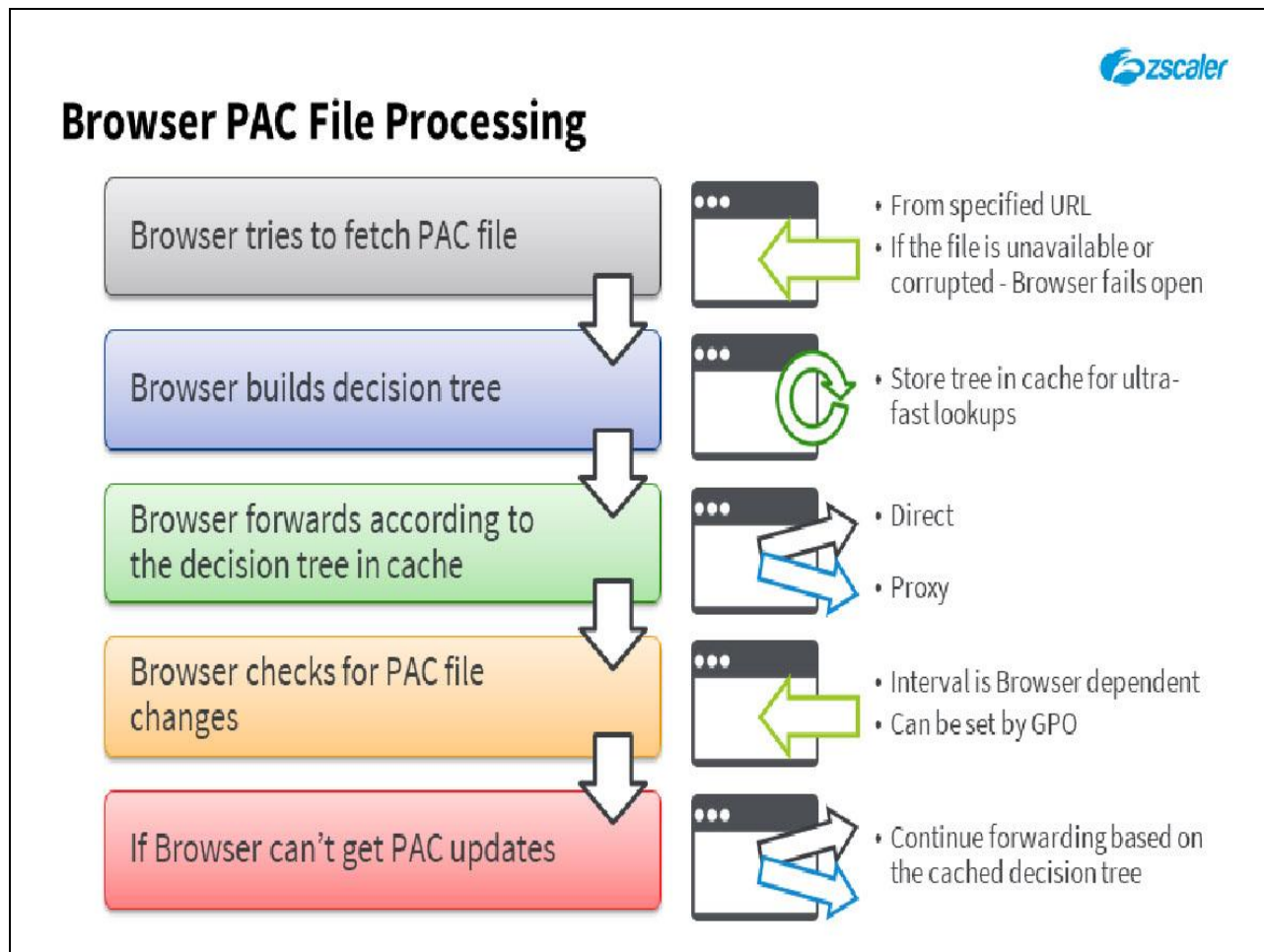
Slide 14 - Browser PAC File Processing



Slide notes

The Browser will check periodically for PAC file updates, although this interval is Browser dependent. If you manage the client Browser PAC file URL using an AD GPO, you can specify the interval for the Browser to check for updates.


Slide 15 - Browser PAC File Processing



Slide notes

If the Browser is unable to retrieve the PAC file on an update interval, or the PAC file retrieved is corrupt somehow, then the Browser continues to forward traffic based on the last known good PAC file logic.

Slide 16 - Zscaler and PAC Files



Zscaler and PAC Files

PAC File Hosting


- PAC files can be hosted by Zscaler (recommended), or by the customer
- A **Recommended PAC** file is provided, custom PAC files can be created from it
- The PAC file must be reachable from wherever the client may request it

Slide notes

The Zscaler service hosts a **Recommended PAC** file that uses geo-location technology to ensure that clients forward traffic to the nearest Zscaler Enforcement Node (ZEN). You can also create your own custom PAC files on the Zscaler service to take advantage of this geo-location capability (this is our recommended deployment model). PAC files hosted by Zscaler are stored and managed securely to guard against tampering. The maximum file size for Zscaler hosted PAC files is 256KB.

The PAC file may also be hosted by the customer, on a workstation, on an internal Web server, or on a server outside the corporate network, although the customer would then be responsible for securing and managing the file, and for ensuring that it is reachable by users under all circumstances.

Slide 17 - Zscaler and PAC Files



Zscaler and PAC Files

PAC File Hosting


- PAC files can be hosted by Zscaler (recommended), or by the customer
- A **Recommended PAC** file is provided, custom PAC files can be created from it
- The PAC file must be reachable from wherever the client may request it

The File name portion is case sensitive! ...if it does not match, the default PAC file is returned

Slide notes

Note that, for PAC files hosted on Zscaler, the filename of the called PAC file is **case sensitive!** ...if a request does not match exactly we will return the Cloud's default PAC file.

Slide 18 - Zscaler and PAC Files



Zscaler and PAC Files

PAC File Hosting

- PAC files can be hosted by Zscaler (recommended), or by the customer
- A **Recommended PAC** file is provided, custom PAC files can be created from it
- The PAC file must be reachable from wherever the client may request it


Zscaler Macros

- To select and return the local ZEN, based on geo-location rules
- Other macros are available for special purposes

Slide notes

Zscaler macros are available for use within the PAC files hosted by us, to automatically select and return the address of a ZEN local to the user, based on the geo-location capability of the service. Additional macros are available for use in special purposes, for example to dynamically specify a loopback configuration for the Zscaler App set to use the **Tunnel with Local Proxy** forwarding mode; or to identify the IP of the egress gateway of your locations.

Slide 19 - Zscaler and PAC Files



Zscaler and PAC Files

PAC File Hosting

- PAC files can be hosted by Zscaler (recommended), or by the customer
- A **Recommended PAC** file is provided, custom PAC files can be created from it
- The PAC file must be reachable from wherever the client may request it

Zscaler Macros

- To select and return the local ZEN, based on geo-location rules
- Other macros are available for special purposes

Special Ports

- **80** : the standard port for HTTP • **443** : alternative port if 80 is not available
- **9400** : for use when proxy chaining • **9480** : bypass authentication
- **9443** : can be used for Road Warriors to allow inspection of all HTTPS traffic
- **Dedicated Proxy Ports** : customer specific port for road warriors


Slide notes

ZENs accept web requests on ports **80**, **443**, **9400**, **9480** and **9443**, or on a dedicated port specific to an individual customer.

Port **80** is the standard port used by almost all Web servers, although port **443** can be used instead if port **80** is for some reason not available. Port **9400** can be used if another host between the end user and ZEN (another proxy solution) attempts to redirect the user's traffic before it can reach the ZEN. Port **9480** can be used to bypass authentication requirements from known locations. Port **9443** can be used for road warriors who need the service to proxy and inspect HTTPS transactions.


Unique Dedicated Proxy Ports are assigned to customers on subscription to that service and allows SSL inspection for Road Warriors.

Slide 20 - PAC Files and User Authentication



PAC Files and User Authentication

- PAC file users connecting from an *unknown* location
 - Authentication to the Zscaler service is required
 - Users must be provisioned to Zscaler
 - Authentication for the organization must be properly configured
 - User must close the Browser and re-open it after authenticating through a captive portal

**Slide notes**

If PAC file users connect to the service from an unknown location (meaning from somewhere other than one of the defined GRE or IPsec tunneled locations), they will have to log in to the service before the service will start to protect their web traffic.

This means that the users must already be provisioned to Zscaler using any of the supported methods (SAML auto-provisioning, manual creation in the Hosted DB, CSV upload to the Hosted DB, LDAP synchronization, Zscaler Authentication Bridge (ZAB)).

An appropriate authentication method would need to be configured, although we will default to Hosted DB password-based authentication. The authentication methods supported are: SAML, LDAP, Kerberos, Hosted DB passwords, ZAB, one-time link, and one-time token.

Note that, if a user is connecting from a Hotspot such as Starbucks or MacDonald's, and must log into a captive portal before connecting to The Internet, then for full protection the user must close the Browser and open it again in order to reload the PAC file (the Browser would only retry to fetch the PAC file when there is a PAC URL timeout, which may not be for a number of hours).


Slide 21 - Where to Host PAC Files?



Slide notes

In the next section, we will discuss where you should host your PAC files.

Slide 22 - PAC File Hosting



PAC File Hosting

General Considerations

- PAC files must be securely hosted and protected to prevent manipulation
- PAC files must be reachable by the users (on Corporate network, or Internet)
- PAC file recovery over SSL is recommended to avoid hijacking and MitM
- Users must be provided the correct URL for the appropriate file


Slide notes

A PAC file is a JavaScript file, and as such is open to abuse by threat actors, who are looking for a pathway to run malicious code on a host computer. They must therefore be hosted and managed securely to avoid them being hijacked, spoofed, or otherwise manipulated; although note that they must also be available to any client device that requests them.

Users must be able to reach the PAC files regardless of where they are connecting from, so are best stored on some cloud service with Firewall rules as necessary to allow their retrieval from within the corporate network. It is recommended that users retrieve the PAC file over SSL secured by trusted PKI, to ensure that the files are encrypted while in transit, and to guard against Man-in-the-Middle attacks that might lead to PAC file hijacking.

The Browser's installed on the User's workstations must of course, be configured to retrieve the correct PAC file to ensure that their Web traffic is properly managed and protected.

Slide 23 - PAC File Hosting



PAC File Hosting

General Considerations

- PAC files must be securely hosted and protected to prevent manipulation
- PAC files must be reachable by the users (on Corporate network, or Internet)
- PAC file recovery over SSL is recommended to avoid hijacking and MitM
- Users must be provided the correct URL for the appropriate file


Hosted Internally

- On a Corporate server
- Must be reachable by all authorized users Worldwide

Slide notes

PAC files may of course be hosted, managed, and made available by the corporate IT department, although they must fully understand the potential vulnerabilities, plus they must ensure the continuous availability of the files to their users wherever they may be connecting from.

Slide 24 - PAC File Hosting



PAC File Hosting

General Considerations

- PAC files must be securely hosted and protected to prevent manipulation
- PAC files must be reachable by the users (on Corporate network, or Internet)
- PAC file recovery over SSL is recommended to avoid hijacking and MitM
- Users must be provided the correct URL for the appropriate file

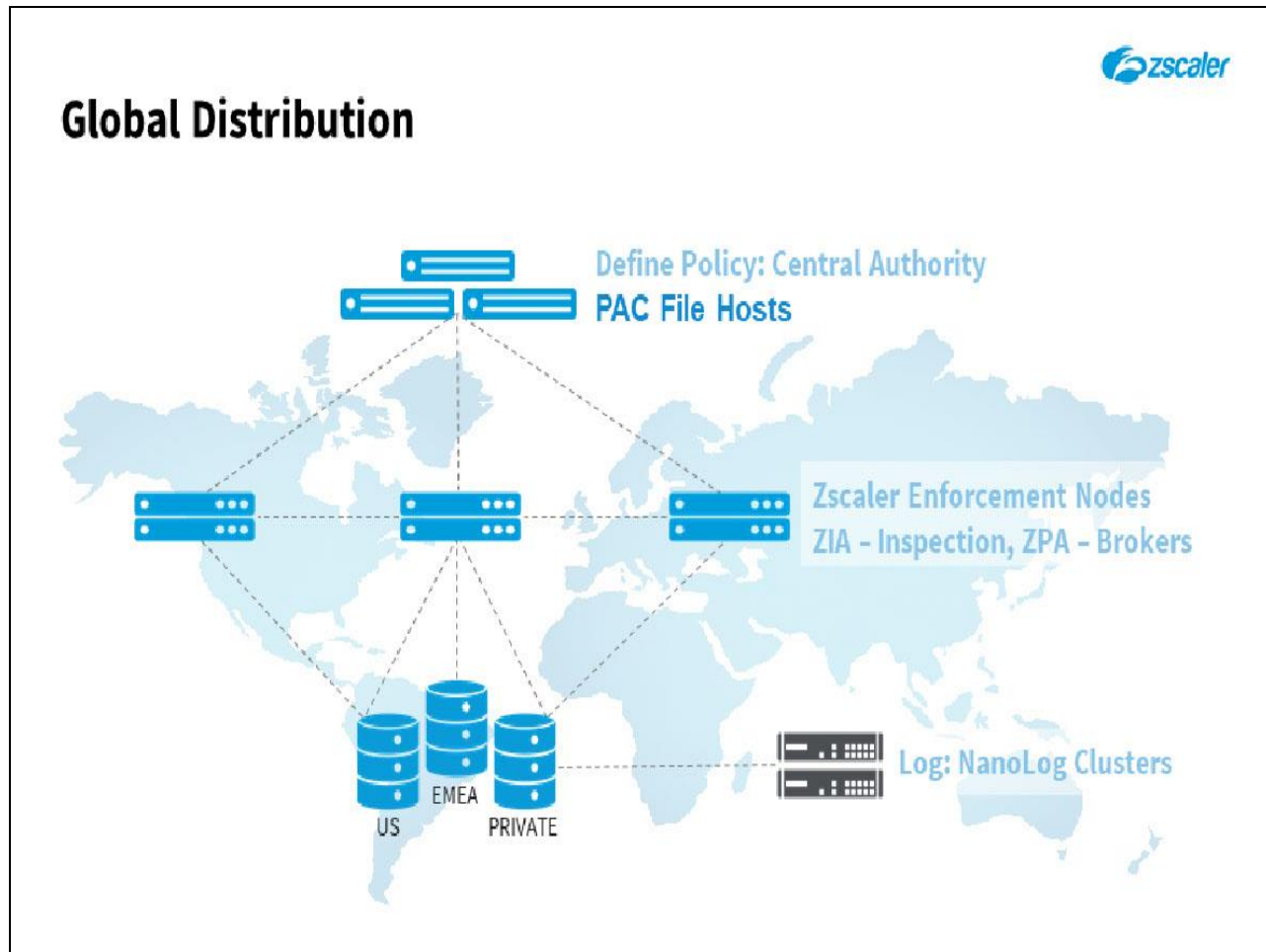
Hosted Internally	Hosted by Zscaler
<ul style="list-style-type: none">• On a Corporate server• Must be reachable by all authorized users Worldwide	<ul style="list-style-type: none">• Recommended• Secure PAC file hosting• Geo-location logic to identify the closest ZEN to the user• Zscaler Macros can be used• Optional URL obfuscation

Slide notes

Our recommendation is to host your PAC files on the Zscaler service, which brings with it a number of advantages: We ensure the secure storage and management of the files; PAC files hosted by us can use the Zscaler geo-location capability to allow the re-direction of user traffic to the closest ZEN proxy; plus, the Zscaler macros are available for enhanced PAC file logic.

Another advantage of hosting PAC files on Zscaler, is that you have the option to obfuscate the corporate domain part of the PAC file URL. Although note that this then makes the URLs unique, and it will change if you edit the file. This means that you will need to re-distribute the new obfuscated PAC file URL once you have finished editing it.


Slide 25 - Global Distribution



Slide notes

When you create a PAC file at the Zscaler Admin Portal, the file is hosted at the Central Authority (CA) for the Cloud. The file is replicated to the redundant CA locations to ensure continuous availability. The PAC file gateway definitions can employ Zscaler macros to allow geo-location of the user, and re-direction to the closest ZEN, to ensure optimum forwarding performance for your users.

Slide 26 - PAC Files in No Default Route Environments



PAC Files in No Default Route Environments

- In a no default route environment there is no default route known to most hosts that leads to the Internet, and thus this can be considered a closed network

Prerequisites

- GRE/IPSec tunnel to Zscaler
- PAC file(s) with return statement to the global ZEN IPs
- Explicit routing (PBR) from the internal network to these IPs through GRE/IPSec tunnels

Slide notes

In a no default route environment, no default route is defined that leads to the Internet, and thus this can be considered a closed network. In an environment like this certain hosts in the network must be explicitly configured to reach resources on the Internet. In today's world of cloud-based mission critical applications, mobile users, dynamic threats, and targeted attacks, the usefulness of a no default route environment is diminished. Regardless, Zscaler's cloud service may need to be deployed in a no default route environment to provide a greater level of protection for end-users connecting to the Internet.

Some prerequisites to enabling Zscaler as a proxy in a no default route environment are:

- Forwarding to Zscaler must be done using **GRE or IPSec tunnels** from fixed locations;
- The PAC files delivered to users must specify access to Zscaler using the **global ZEN IP addresses** (available from the Help Portal);
- The internal network must be configured to route traffic to these IP addresses down the tunnels (usually using policy-based routing).

Slide 27 - PAC Files in No Default Route Environments



PAC Files in No Default Route Environments


- In a no default route environment there is no default route known to most hosts that leads to the Internet, and thus this can be considered a closed network

Prerequisites	Local PAC File Server
<ul style="list-style-type: none">• GRE/IPSec tunnel to Zscaler• PAC file(s) with return statement to the global ZEN IPs• Explicit routing (PBR) from the internal network to these IPs through GRE/IPSec tunnels	<ul style="list-style-type: none">• Internal DNS pointing to a local PAC file server• Build location awareness into the PAC file by resolving a local hostname

Slide notes

There are two options for the hosting of the PAC files in a no default route environment, the first option being to host the files on an internal server. The PAC files must return one of the global ZEN IP addresses as the gateway, then the local network will route traffic into a tunnel for forwarding to Zscaler. If necessary you can build a single, location-aware PAC file by resolving a local hostname, and routing as appropriate.


Slide 28 - PAC Files in No Default Route Environments



PAC Files in No Default Route Environments

- In a no default route environment there is no default route known to most hosts that leads to the Internet, and thus this can be considered a closed network

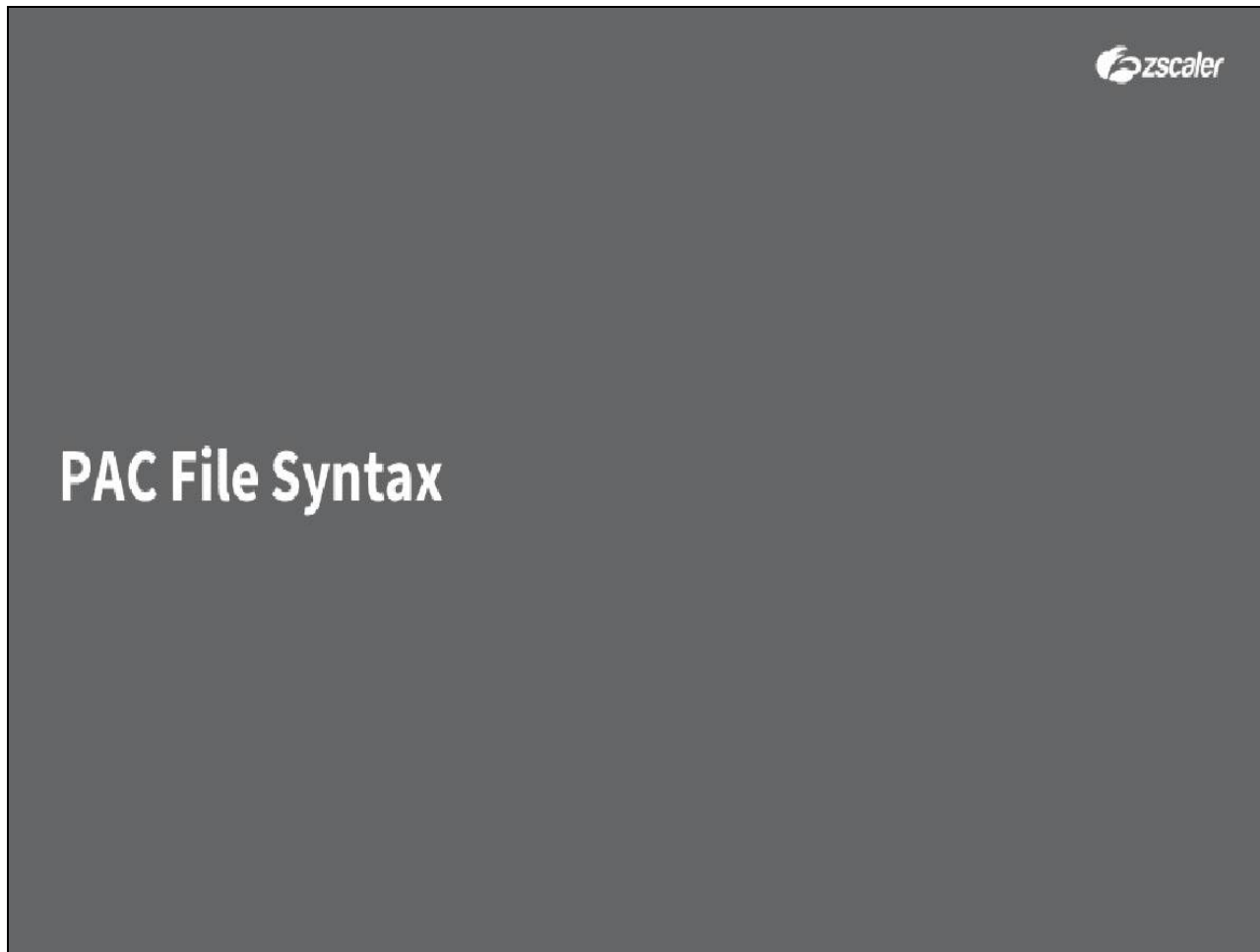
Prerequisites	Local PAC File Server	PAC Files on Zscaler
<ul style="list-style-type: none">• GRE/IPSec tunnel to Zscaler• PAC file(s) with return statement to the global ZEN IPs• Explicit routing (PBR) from the internal network to these IPs through GRE/IPSec tunnels	<ul style="list-style-type: none">• Internal DNS pointing to a local PAC file server• Build location awareness into the PAC file by resolving a local hostname	<ul style="list-style-type: none">• Locally resolve Zscaler hostnames to the global ZEN IPs• Build location awareness into the PAC file by resolving a local hostname



Slide notes

The recommended option is to host the PAC files on Zscaler and resolve the Zscaler hostnames locally to one of the global ZEN IPs. This means that users will actually retrieve the PAC files through the tunnels from Zscaler as well. If location-based intelligence is required within the PAC file, you can build this in by resolving a local hostname in the same way as for an internal PAC file server.

Slide 29 - PAC File Syntax



Slide notes

In the next section, we will describe some aspects of PAC file syntax.

Slide 30 - Standard Functions



Standard Functions

- Primary Function Call: *FindProxyForURL(url, host)*

- Useful Standard Functions

<i>isPlainHostName</i>	- Returns true if hostname contains no dots
<i>dnsDomainIs</i>	- Returns true if domain matches
<i>localHostOrDomainIs</i>	- Returns true if exact hostname matches
<i>isInNet</i>	- Returns true if hostname resolves to listed subnet
<i>dnsResolve</i>	- Resolves hostname into an IP address; often stored as a variable

- Other Functions

<i>shExpMatch</i>	<i>myIpAddress</i>	<i>isResolvable</i>	<i>dnsDomainLevels</i>
<i>weekdayRange</i>	<i>dateRange</i>	<i>timeRange</i>	<i>alert</i>

- Further reading: <http://www.findproxyforurl.com>

Slide notes

The Proxy auto-config file format was originally designed by Netscape in 1996 for the Netscape Navigator 2.0 and is a text file that defines at least one JavaScript function, **FindProxyForURL**, which has two arguments: **url** is the URL of the object requested; and **host** is the host-name derived from that URL (usually the FQDN of the host, although the port number is not included). Note that, for faster processing, some Browsers strip the full path and any query components from HTTPS URLs by default, although it is possible to override this behavior.

By convention, the PAC file is normally named **proxy.pac**, although this is not a hard requirement. To use it, a PAC file is published to a HTTP server, and client user agents are instructed to fetch it, either by entering the URL in the proxy connection settings of the Browser, or through the use of the WPAD protocol.

Some useful standard PAC file functions are:

- **isPlainHostName** which returns true if the destination hostname contains no dots, meaning that it is a simple host name, not a FQDN. Typically, such hosts are reachable locally, and do not need to be proxied;

- **dnsDomainIs** which returns true if the domain of the destination host matches that specified. This is the most commonly used, and most efficient function to exclude specific domains from being sent through Zscaler, e.g. bypassing internal domains from going through ZIA;
- **localHostOrDomainIs** which returns true if either the destination FQDN, or just the host name matches that specified;
- **isInNet** which returns true if the IP address for the destination host name resolves to the specified subnet. Note that this function is computationally quite intensive and requires DNS resolution of the destination host. This is best used together with a variable that is populated using the **dnsResolve** function to resolve the host IP just the one time;
- **dnsResolve** resolves the destination hostname (or a static FQDN) to an IP address, which is then often stored as a variable. Use this function sparingly to avoid excessive latency and ensure that the underlying DNS infrastructure is robust enough to support the use of this function.

Other functions available are: **shExpMatch**, **myIpAddress**, **isResolvable**, **dnsDomainLevels**, **weekdayRange**, **dateRange**, **timeRange**, and **alert**. See the Website at <http://www.findproxyforurl.com> for full details of these functions.

Slide 31 - Standard Functions



Standard Functions

- Primary Function Call: *FindProxyForURL(url, host)*
- Useful Standard Functions



To minimize latency,
avoid functions that
require a DNS lookup

<i>isPlainHostName</i>	- Returns true if hostname contains no dots
<i>dnsDomainIs</i>	- Returns true if domain matches
<i>localHostOrDomainIs</i>	- Returns true if exact hostname matches
<i>isInNet</i>	- Returns true if hostname resolves to listed subnet
<i>dnsResolve</i>	- Resolves hostname into an IP address; often stored as a variable

- Other Functions


<i>shExpMatch</i>	<i>myIpAddress</i>	<i>isResolvable</i>	<i>dnsDomainLevels</i>
<i>weekdayRange</i>	<i>dateRange</i>	<i>timeRange</i>	<i>alert</i>

- Further reading: <http://www.findproxyforurl.com>

Slide notes

Note that to minimize latency with the processing of the PAC file, it is recommended that you keep to a minimum the function calls that require a DNS lookup (such as **isInNet**, or **dnsResolve**).

Slide 32 - Zscaler Macros



Zscaler Macros

Identifying the ZEN through Zscaler Macros

- Dynamic, the recommended solution
`${GATEWAY}` and `${SECONDARY_GATEWAY}`
- Returns ZEN IP closest to client
`${GATEWAY_HOST}` and `${SECONDARY_GATEWAY_HOST}`
- Returns ZEN hostname closest to client (required for Kerberos implementations)
`${COUNTRY}` variable
- Can be used to identify the country the user is located in based on the egress IP address
`${COUNTRY_GATEWAY}` and `${COUNTRY_SECONDARY_GATEWAY}`
- Returns the closest ZEN within the client's country (identified through the `${COUNTRY}` variable)
- Fallback to `${GATEWAY}` behavior if no node in the country (or one only)

Slide notes

There are three main ways the ZEN to proxy user traffic to can be identified in a PAC file hosted by Zscaler:

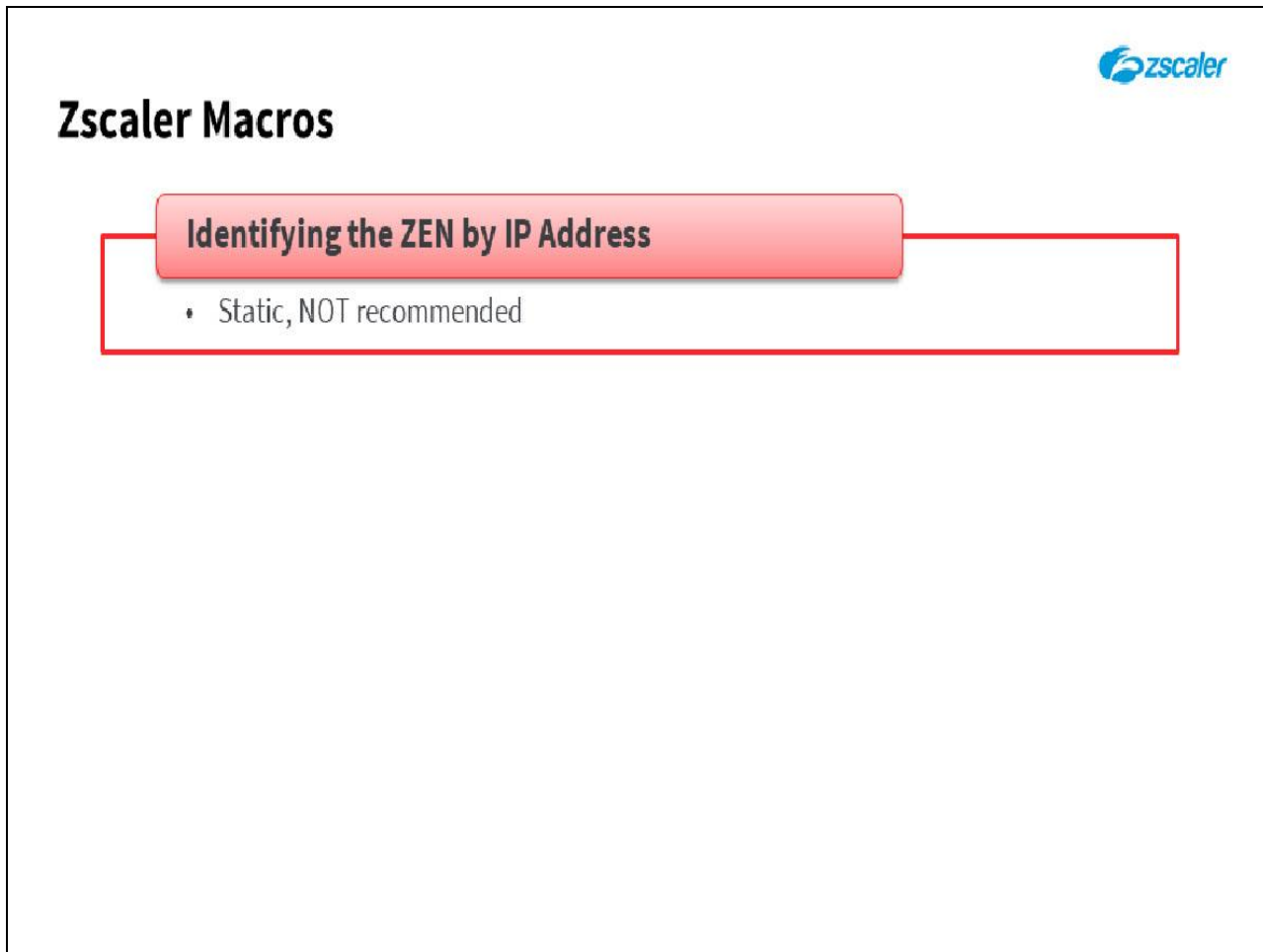
1. The first (and recommended) method is to host the files on Zscaler and make use of the Zscaler macros `${GATEWAY}` and `${SECONDARY_GATEWAY}` which return the IP address of the ZEN closest to the user.

Under special circumstances (for example when using Kerberos authentication), you must use the macros `${GATEWAY_HOST}` and `${SECONDARY_GATEWAY_HOST}` instead, which return the FQDNs of the ZENs rather than their IPs.

In addition, if you want to ensure that users connect through a ZEN within the borders of the country where they are based, you can first resolve their country to the `${COUNTRY}` variable (based on their egress IP address), then use the `${COUNTRY_GATEWAY}` and `${COUNTRY_SECONDARY_GATEWAY}` macros in the return statement, to return the closest ZEN within their country.

Note that if there is no node in the country a user connects from (or only the one), then the behavior for this macro falls back to that of the `${GATEWAY}` macro. There is no need for follow-on `${GATEWAY}` statements.

Slide 33 - Zscaler Macros



The slide features the Zscaler logo in the top right corner. The main title 'Zscaler Macros' is positioned on the left. A red-bordered box contains the text 'Identifying the ZEN by IP Address' in a bold, dark font. Below this, a bullet point indicates 'Static, NOT recommended'.

Zscaler Macros


Identifying the ZEN by IP Address

- Static, NOT recommended

Slide notes

2. Another option is to specify the IP address of a specific ZEN, although this is NOT recommended as this would be a static configuration, with no ability to use Zscaler's Geo-location technology.

Slide 34 - Zscaler Macros




Zscaler Macros

- Identifying the ZEN by IP Address**
 - Static, NOT recommended
- Identifying the ZEN by IP Host Name**
 - Dynamic using **Cloud** or **Sub-Cloud** name
`gateway.<cloud name>.net` and `secondary.gateway.<cloud name>.net`
 - Returns ZEN IP closest to DNS Server
 - Use if NOT hosting PAC file on Zscaler PAC server

Slide notes

3. Finally, you can specify the ZENs dynamically using the Cloud or Sub-Cloud name in the hostname, for example **gateway.zscalerone.net** and **secondary_gateway.zscalerone.net**. These hostnames will return the IP of the ZENs closest to the DNS server, and can be used if you are hosting the PAC files yourself.

Slide 35 - Zscaler Macros



Zscaler Macros

- Identifying the ZEN by IP Address**
 - Static, NOT recommended
- Identifying the ZEN by IP Host Name**
 - Dynamic using **Cloud** or **Sub-Cloud** name
`gateway.<cloud name>.net` and `secondary.gateway.<cloud name>.net`
 - Returns ZEN IP closest to DNS Server
 - Use if NOT hosting PAC file on Zscaler PAC server
- Other Zscaler Macros**
 - `${ZAPP_LOCAL_PROXY}` - for Zscaler App in **Tunnel with Local Proxy** mode
 - `${SRCIP}` - to identify egress gateway IP for use in the PAC file logic

Slide notes

There are additional Zscaler macros that are available for use under special circumstances, such as the `${ZAPP_LOCAL_PROXY}` option which must be used in the PAC file applied to the Zscaler App when it is set to the **Tunnel with Local Proxy** forwarding mode. Or the `${SRCIP}` option, that can be used to identify the egress gateway IP of the user traffic so that the Browser can be configured to forward traffic to a different port on the ZEN, depending on the user's location.

Slide 36 - PAC File Use Cases



Slide notes

In the next section, we look at some Zscaler use cases for PAC files.

Slide 37 - Zscaler PAC File Hosting

Zscaler PAC File Hosting



- Zscaler as PAC File Host
 - **Recommended** PAC file with option to create custom files
 - Zscaler securely hosts PAC files that are available across The Internet

No.	Description	Domain	Hosted URL
1	PAC for CM	pete.zscaler.com	http://pac.zscaler.com/pete.zscaler.com/ChangMai
2	Recommended PAC	zscalerwo.net	http://pac.zscalerwo.net/zscalerwo.net/recommended.pac
3	Service Default.	zscalerwo.net	http://pac.zscalerwo.net/zscalerwo.net/proxy.pac
4	Service Default.	zscalerwo.net	http://pac.zscalerwo.net/zscalerwo.net/kerberos.pac
5	Service Default.	zscalerwo.net	http://pac.zscalerwo.net/zscalerwo.net/mobile_proxy.pac

Slide notes

The first use case is to simply use the global Zscaler infrastructure to host, manage, and deliver your PAC files. In the Admin Portal under **Administration > Hosted PAC Files** you can view the **Recommended PAC** file for the Cloud your account is provisioned on.


Note that, here you have the ability to create your own PAC files that can be customized to your specific requirements and view the URL for the files (which you must provision to the Browsers of your end users). The starting point for any new PAC file that you create is the **Recommended PAC** file for the Cloud.

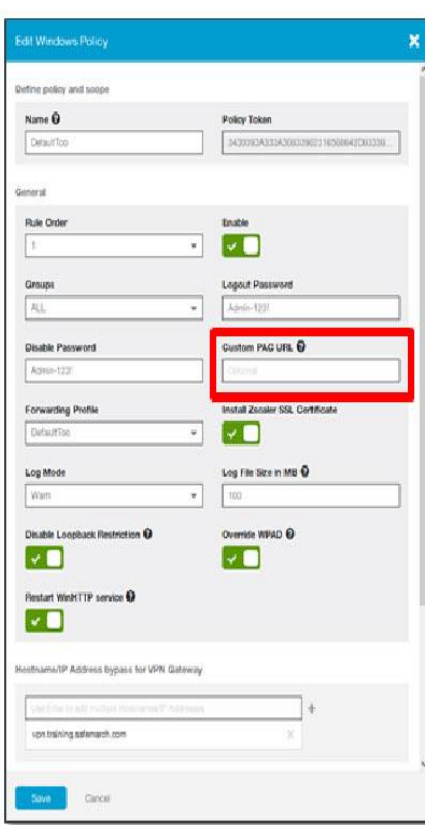
Slide 38 - Zscaler App PAC File Usage

Zscaler App PAC File Usage

App Profile

- Used only by the App
- Cloud default file, or custom file
- Specify target ZENs for the Z-Tunnels
- Optionally specify Z-Tunnel bypasses





Slide notes

For the Zscaler App, PAC file URLs can be defined in several places to manage how the App forwards traffic. Let's have a look at the places they can be defined, and what they actually control.

Firstly, the **Custom PAC URL** in the **App Profile**, which by default is set to use the **default PAC file for the Cloud** you are connected to. If you wish you can override this and provide the URL for a custom file. This file is only applied to, and used by the App as a configuration file, the host system does not see this file. It can be used to specify the target ZENs for the Z-Tunnels (if you do not want to use the closest public ZEN), or to specify destinations that are not to use the Z-Tunnels at all.

Note that as the files applied to the Zscaler App are used to configure App forwarding functionality, it is recommended that you rename them to use the extension **.cfg**, to distinguish them from your true PAC files.

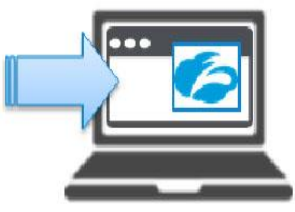
Slide 39 - Zscaler App PAC File Usage

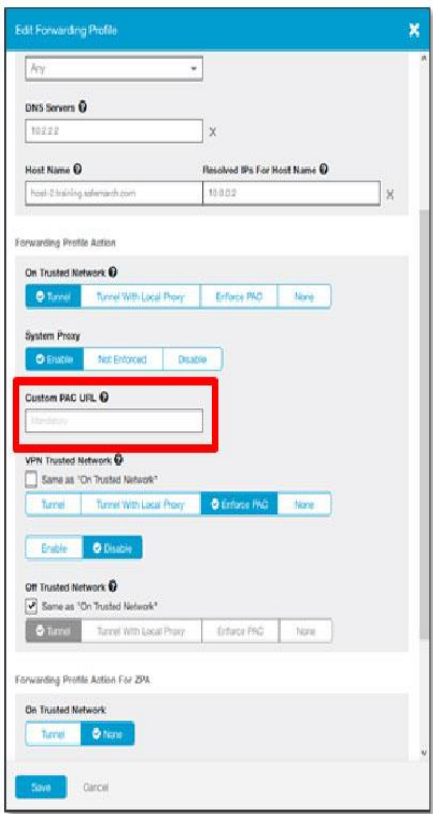
Zscaler App PAC File Usage

Forwarding Profile

Tunnel Mode

- If the **System Proxy** is enabled
- File is applied to the system
- Optionally specify Zscaler App bypasses





Slide notes

In a **Zscaler App Forwarding Profile** in **Tunnel** mode, if the **System Proxy** option is set to **Enable**, then the **Custom PAC URL** must be provided for the PAC file to apply. The specified file is applied to the system to replace any configured proxy definitions, and may contain destinations to be bypassed completely by the App.


Slide 40 - Zscaler App PAC File Usage

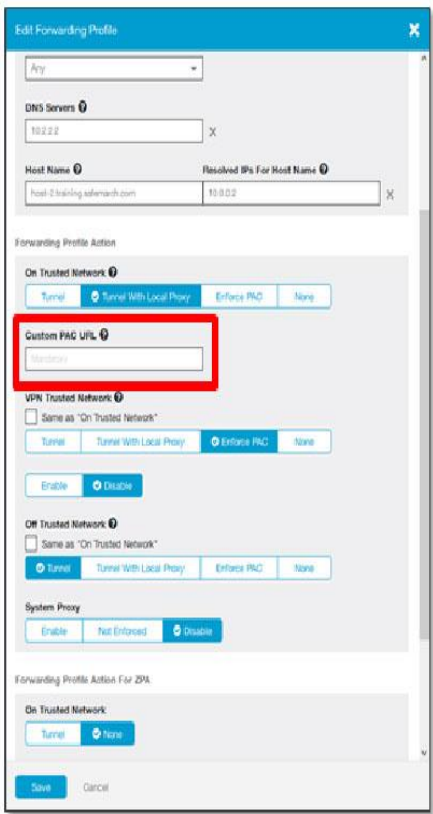
Zscaler App PAC File Usage

Forwarding Profile

Tunnel with Local Proxy Mode

- Applied to the system
- Mandatory loopback definition for gateway (macro function or static configuration)
- Optional Zscaler App bypasses





Slide notes

In a **Zscaler App Forwarding Profile**, in the **Tunnel with Local Proxy** mode, if you provide a **Custom PAC URL** for a PAC file, it **MUST** contain the loopback configuration; either the gateway macro function **`${ZAPP_LOCAL_PROXY}`** (recommended), or a static loopback address using ports **9000**, or **9001**. If no custom PAC file is specified, this configuration is applied automatically.

This file is applied to the system to replace any configured proxy, and is used to send traffic to the App. It may also contain destinations that are to be bypassed by the App completely.


Slide 41 - Zscaler App PAC File Usage

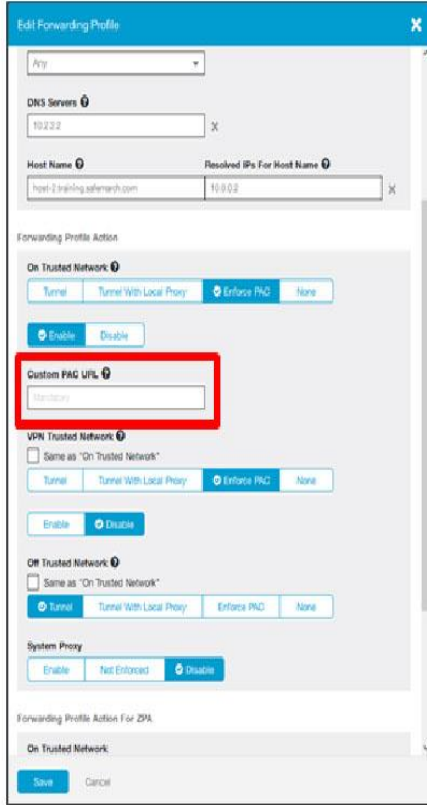
Zscaler App PAC File Usage

Forwarding Profile

Enable PAC Mode

- Applied to the system when enabled
- Regular system proxy file
- Identifies traffic to proxy, or to send direct





Slide notes

In a **Zscaler App Forwarding Profile**, in the **Enable PAC** mode, the file specified in the **Custom PAC URL** 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.

Slide 42 - ZIA and ZPA Customers



ZIA and ZPA Customers

- For customers that use the Zscaler App for both ZIA and ZPA, there is a change needed to the applied App Profile PAC file to ensure the services work together
 - A test, and bypass for the the RFC 6598 Carrier-grade range of 100.64.0.0/16 must be added

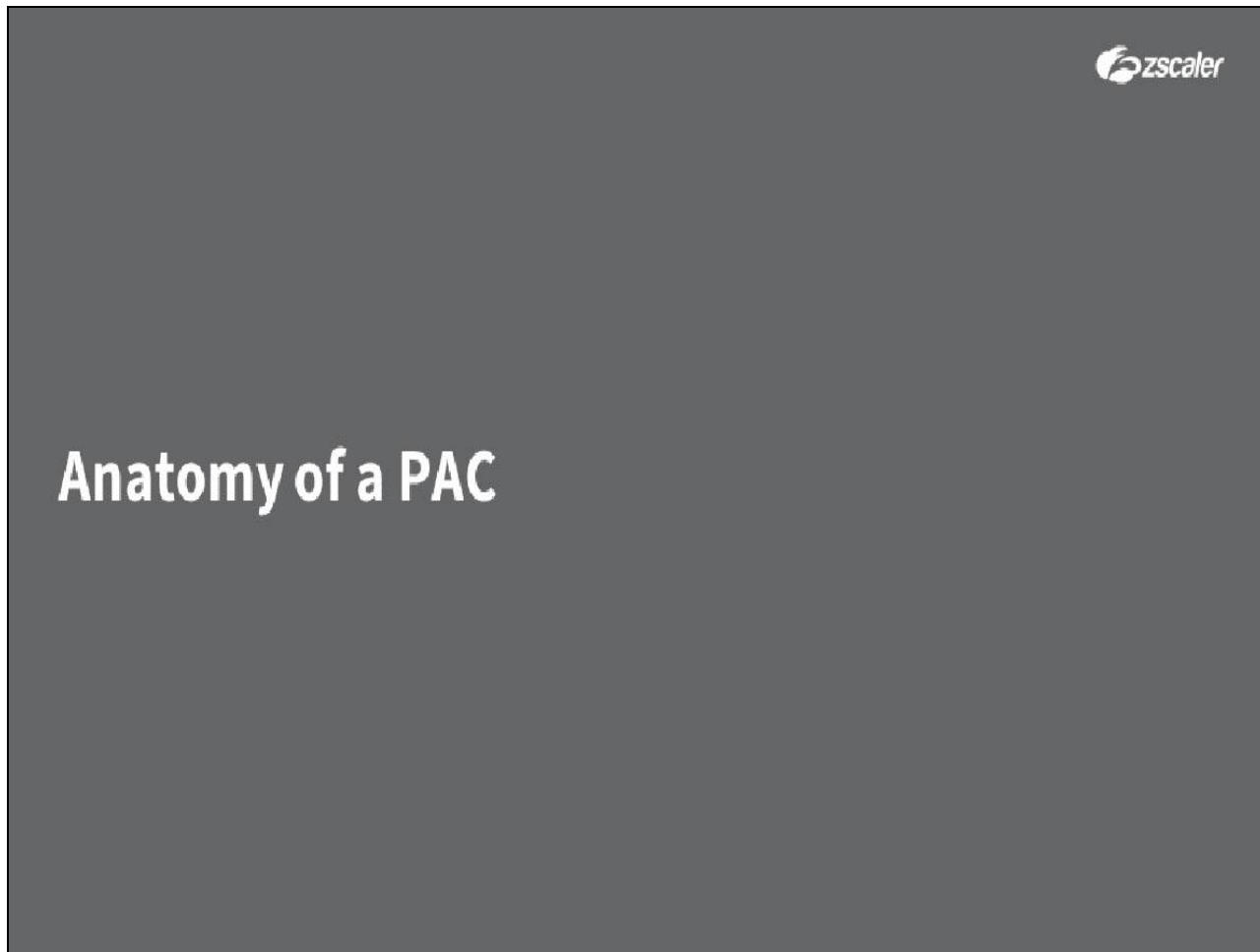
```
/* Test for ZPA*/  
if (isInNet(resolved_ip, "100.64.0.0", "255.255.0.0"))  
return "DIRECT";
```

- The service default PAC files already contains this clause

Slide notes

For customers that use the Zscaler App for both ZIA and ZPA, a change is required to the applied PAC file to make sure the services work together. A test and bypass for the RFC 6598 Carrier-grade range of **100.64.0.0/16** must be added to the PAC file applied in the **App Profile**, to ensure that traffic for private applications is not proxied, and is properly re-directed to the Zscaler App. The test for private access traffic, with the **DIRECT** return statement are shown here. Note that the service default PAC file already contains this clause.


Slide 43 - Anatomy of a PAC



Slide notes

In the next section, we will have a look at the **Recommended PAC** file.

Slide 44 - Zscaler Recommended PAC File



Zscaler Recommended PAC File

```
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);

    if (privateIP.test(host))
        return "DIRECT";
    if (isPlainHostName(host) || (host == "host.example.com") ||
        shExpMatch(host, "*.example.com"))
        return "DIRECT";
    if (shExpMatch(host, "internal.example.com"))
    {
        var resolved_ip = dnsResolve(host);
        if (privateIP.test(resolved_ip))
            return "DIRECT";
    }
}
```

Tests and bypasses for:
 RFC1918 Private IP
 space; specific hosts or
 domains; internal
 servers


Slide notes

Let's walk through the various sections of the **Recommended PAC** file to see how it works...

The first statement creates a variable that we will use later in the file. This is a list of all RFC 1918 internal IP addresses stored as the variable **privateIP**. Then comes a series of tests of the requested URL, to see if it matches that **privateIP** variable that we stored; or you may define specific hostnames or domains; or you can check to see if the resolved hostname is on a private network. In each case, there is no need (or no point) in proxying the traffic to these destinations, so these all return a **Direct** statement.

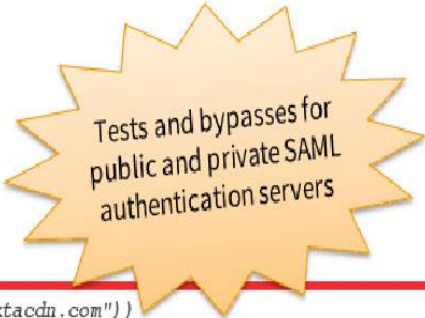
Note that we recommend avoiding the **dnsResolve** function wherever possible, to minimize latency as the Browser processes the PAC file, otherwise it must wait for an unpredictable time while the hostname is DNS resolved.

Slide 45 - Zscaler Recommended PAC File



Zscaler Recommended PAC File

```
if (shExpMatch(host, "internal.example.com"))
{
    var resolved_ip = dnsResolve(host);
    if (privateIP.test(resolved_ip))
        return "DIRECT";
}
```



Tests and bypasses for
public and private SAML
authentication servers


```
if (shExpMatch(host, "*.okta.com") || shExpMatch(host, "*.oktacdn.com"))
    return "DIRECT";
if (shExpMatch(host, "my_iwa_server.my_example_domain.com"))
    return "DIRECT";
```

```
if ((url.substring(0,5) != "http:") &&
    (url.substring(0,4) != "ftp:") &&
    (url.substring(0,6) != "https:"))
    return "DIRECT";
```

Slide notes

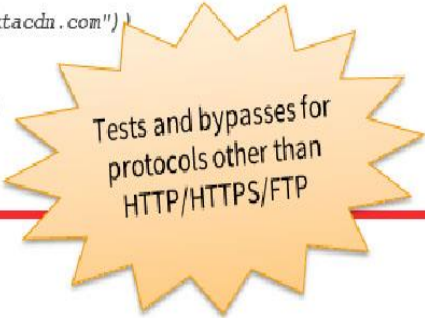
The next examples are tests for SAML IdP URLs, which should ideally go direct (even though these sites can be exempted in the Admin Portal).

Slide 46 - Zscaler Recommended PAC File



Zscaler Recommended PAC File

```
if (shExpMatch(host, "*.okta.com") || shExpMatch(host, "*.oktacdn.com"))  
    return "DIRECT";  
  
if (shExpMatch(host, "my_iwa_server.my_example_domain.com"))  
    return "DIRECT";  
  
if ((url.substring(0,5) != "http:") &&  
    (url.substring(0,4) != "ftp:") &&  
    (url.substring(0,6) != "https:"))  
    return "DIRECT";  
  
var trust = /^(trust|ips).(zscaler|zscalerone|zscalertwo|zscalerthree|zscalergov|zsccloud).(com|net)$/;  
if (trust.test(host))  
    return "DIRECT";  
  
if (isInNet(resolved_ip, "100.64.0.0", "255.255.0.0"))  
    return "DIRECT";
```




Tests and bypasses for
protocols other than
HTTP/HTTPS/FTP

Slide notes

Next are tests for protocols that are NOT HTTP, HTTPS, or FTP, and which should therefore go direct.

Slide 47 - Zscaler Recommended PAC File



Zscaler Recommended PAC File

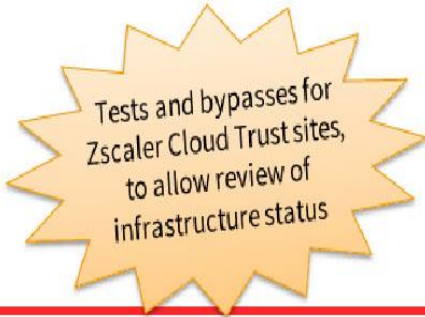
```
if (shExpMatch(host, "*.okta.com") || shExpMatch(host, "*.oktacdn.com"))
    return "DIRECT";

if (shExpMatch(host, "my_iwa_server.my_example_domain.com"))
    return "DIRECT";

if ((url.substring(0,5) != "http:") &&
    (url.substring(0,4) != "ftp:") &&
    (url.substring(0,6) != "https:"))
    return "DIRECT";

var trust = /^(trust|ips).(zscaler|zscalerone|zscalertwo|zscalerthree|zscalergov|zsccloud).(com|net)$/;
if (trust.test(host))
    return "DIRECT";

if (isInNet(resolved_ip, "100.64.0.0", "255.255.0.0"))
    return "DIRECT";
```




Tests and bypasses for
Zscaler Cloud Trust sites,
to allow review of
infrastructure status

```
var trust = /^(trust|ips).(zscaler|zscalerone|zscalertwo|zscalerthree|zscalergov|zsccloud).(com|net)$/;
if (trust.test(host))
    return "DIRECT";
```

Slide notes

The next test in the file is for the Zscaler Trust pages for each of our Clouds, which are bypassed to allow you to verify the status of the infrastructure you are connecting to in a troubleshooting situation.

Slide 48 - Zscaler Recommended PAC File




Zscaler Recommended PAC File

```
if ((url.substring(0,5) != "http:") &&
    (url.substring(0,4) != "ftp:") &&
    (url.substring(0,6) != "https:"))
    return "DIRECT";

var trust = /^(trust|ips), (zscaler|zscalerone|zscalertwo|zscal
if (trust.test(host))
    return "DIRECT";

if (isInNet(resolved_ip, "100.64.0.0", "255.255.0.0"))
    return "DIRECT";

return "PROXY ${GATEWAY}:9400; PROXY ${SECONDARY_GATEWAY}:9400; PROXY ${GATEWAY}:80; PROXY
${SECONDARY_GATEWAY}:80; DIRECT";
}
```




Test and bypass for ZPA destinations

Slide notes

Lastly is a test and bypass for the **100.64.0.0** subnet used for private applications that are to be accessed using the ZPA service.

Slide 49 - Zscaler Recommended PAC File



Zscaler Recommended PAC File

```

if ((url.substring(0,5) != "http:") &&
    (url.substring(0,4) != "ftp:") &&
    (url.substring(0,6) != "https:"))
    return "DIRECT";

var trust = /^(trust|ips), (zscaler|zscalerone|zscalerthree|zscalercloud), (com|net)$/;
if (trust.test(host))
    return "DIRECT";

if (isInNet(resolved_ip, "100.64.0.0", "255.255.0.0"))
    return "DIRECT";

return "PROXY ${GATEWAY}:9400; PROXY ${SECONDARY_GATEWAY}:9400; PROXY ${GATEWAY}:80; PROXY
${SECONDARY_GATEWAY}:80; DIRECT";

```

Proxy Gateway and
suggested port
definitions

return "PROXY \${GATEWAY}:9400; PROXY \${SECONDARY_GATEWAY}:9400; PROXY \${GATEWAY}:80; PROXY
\${SECONDARY_GATEWAY}:80; DIRECT";

Slide notes

Finally, is the return statement with both primary, and secondary Gateway definitions. We suggest using port **9400** for your primary proxy definitions to allow administrators to configure traffic routing policies on their CPE; destination port **9400** can go into the GRE or IPsec tunnels to Zscaler, while any traffic that you need to bypass can go direct. A secondary set of gateway definitions using port **80** can be specified as a fallback configuration.


Note the use of the Zscaler macros **\${GATEWAY}** and **\${SECONDARY_GATEWAY}** to allow the geo-location of the client, and subsequent re-direction to a local ZEN.

Slide 50 - PAC File Best Practices



Slide notes

In the final section, we will describe some Zscaler best practices for the use of PAC files.

Slide 51 - Traffic to Consider Sending Direct

Traffic to Consider Sending Direct


- Protocols**
 - All protocols but HTTP/HTTPS are normally sent direct
 - FTP may be proxied if supported over HTTP
- Private Networks / IPs / Hosts**
 - Local traffic, that a Proxy can't reach across the Internet
- Specific Hosts / Domains**
 - Internal or external Servers or Domains that should not be proxied
 - SAML IdPs To ensure reachability prior to authentication
- Zscaler Trust pages**
 - To allow review of Cloud status regardless of Proxy status

Slide notes

You should consider carefully what traffic should NOT go to the proxy service and construct a PAC file to account for it. Typical proxy exclusions are:

- Any protocol other than HTTP/HTTPS, although FTP over HTTP may be proxied;
- Any traffic to private IP addresses, or internal domains that a Proxy can't reach across the Internet;
- There may be specific internal or external Servers or Domains that should not be proxied, for example the hosts/domain for any external SAML IdP that you use;
- The Zscaler Trust pages should go direct, to allow the review of Cloud status regardless of Proxy status.

Slide 52 - Zscaler PAC File Best Practices



Zscaler PAC File Best Practices


- Use PAC file URL specified statically on the Browser**
 - WPAD is vulnerable to DHCP and DNS exploits
 - Deploy using GPO
- Host PAC Files on Zscaler**
 - We provide secure PAC file hosting
 - Allows the use of Zscaler macros for ZEN Geo-location
- Use Zscaler macros `${GATEWAY}` and `${SECONDARY_GATEWAY}`**
 - Do NOT use ZEN IPs
 - Using Cloud/Sub-Cloud names requires DNS lookup
 - Use `${COUNTRY}` variable, `${COUNTRY_GATEWAY}` and `${COUNTRY_SECONDARY_GATEWAY}` for in country ZEN resolution

Slide notes

Some Zscaler best practices are listed here. Please check the Help Portal for detailed recommendations for creating and managing PAC files.

- Use a PAC file URL specified statically on the client device Browsers, as WPAD is vulnerable to DHCP and DNS exploits. You can deploy the URL to client devices silently using an Active Directory Group Policy Object (GPO).
- Use PAC Files hosted on the Zscaler Cloud, as we provide secure PAC file hosting and management, plus this allows the use of Zscaler macros for closest ZEN Geo-location.
- Use the Zscaler macros `${GATEWAY}` and `${SECONDARY_GATEWAY}` to dynamically identify the IPs of the ZENs closest to the user. Do NOT hard-code the actual ZEN IPs into the PAC file, as this would result in a static configuration that would be difficult to manage. The use of the Cloud/Sub-Cloud names for gateway identification requires a DNS lookup, which may result in delay for the user on their first-time connection to the network. If you need to resolve a ZEN within the country where you are based, use the `${COUNTRY}` variable, and the macros `${COUNTRY_GATEWAY}` and `${COUNTRY_SECONDARY_GATEWAY}`.

Slide 53 - Zscaler PAC File Best Practices




Zscaler PAC File Best Practices

- Do not include host bypasses that can be defined elsewhere**
 - SSL sites that are not to be inspected
- Verify that your JavaScript is error-free**
 - Use the Verify button in Admin Portal
- Divide and Conquer!**
 - Create PAC files targeted at specific; Locations, Groups, Departments
 - Distribute the appropriate PAC files based on where users are based, or the Groups/Departments they are members of
- Resolve Gateway IP for applications that need it**
 - Use the macro `${SRCIP}` to resolve the egress gateway IP

Slide notes

- Do not include host bypasses that can be defined elsewhere, in particular sites that are to be exempted from SSL inspection should be defined in the **SSL Inspection** Policy, rather than bypassed in the PAC file.
- Use the **Verify** button in Admin Portal to confirm that your JavaScript is free from syntax errors.
- Compartmentalize your PAC file deployment, by targeting the files by Location, User Group, or Department. Distribute the appropriate PAC file based on where a user is located, or the Groups/Departments that they are members of.
- Some applications control access based on the source IP of a request, use the Zscaler macro `${SRCIP}` to identify the egress gateway IP address, and build PAC file logic to use it.

Slide 54 - Zscaler PAC File Best Practices



Zscaler PAC File Best Practices

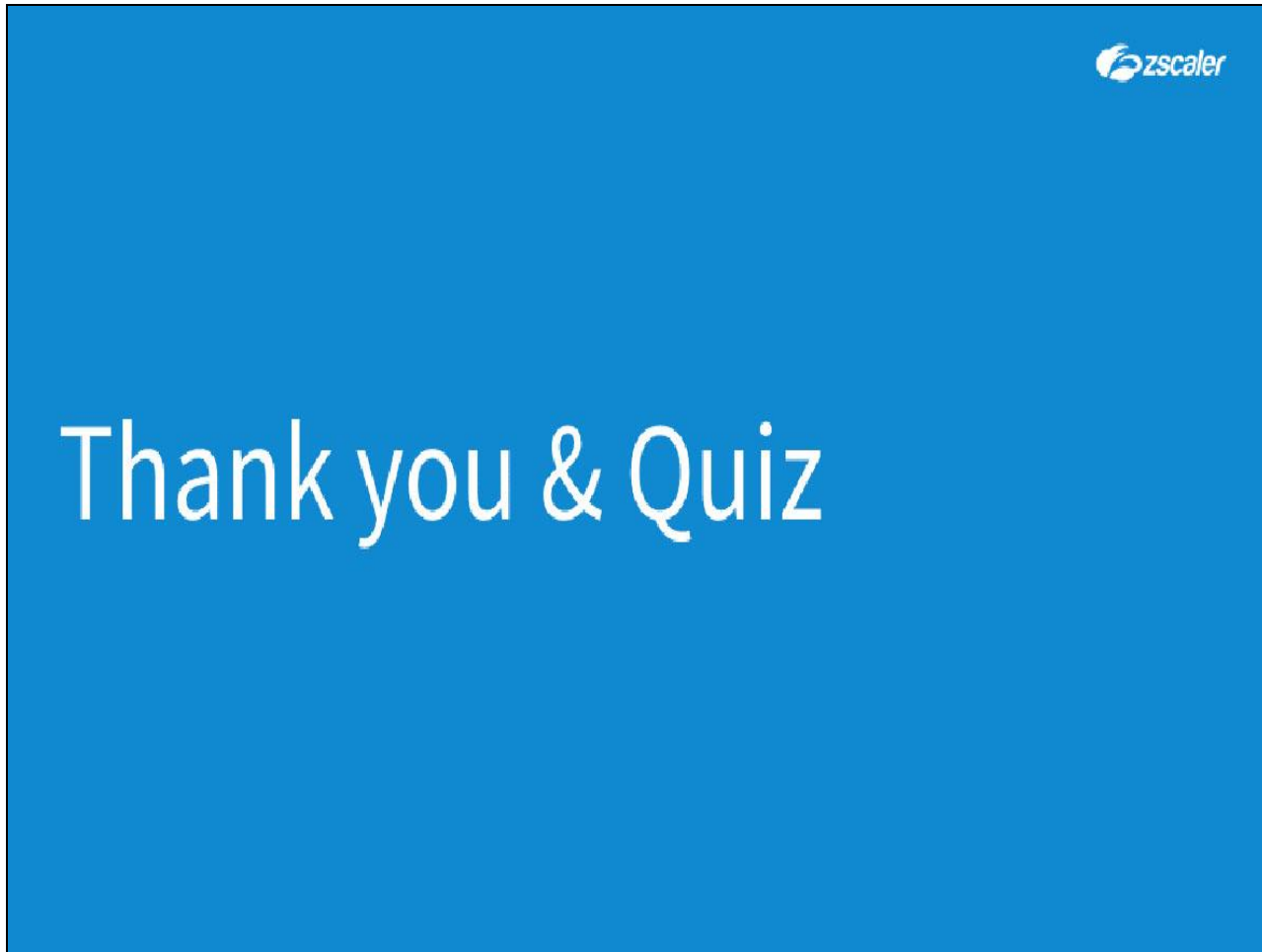
- Keep the PAC file compact and as efficient as possible
 - Place high probability checks at the top
 - Avoid excessive use of exclusion functions
 - Consolidate bypass criteria into fewer if() statements if possible
 - Beware the overuse of isResolvable(), dnsResolve(), and isInNet() to avoid potential DNS performance issues
 - Not need for 'else' statements, as the 'return' is immediate
 - Remove examples or sections that are not required
 - Remove comments as possible once the code has been finalized
- Exclude a Data Center in users PAC file
 - Normally the closest Zscaler Data Centers are calculated based on geo-location using the user's IP address
 - The PAC file can be modified to override this and exclude specific Data Centers if necessary

Slide notes

Keep the PAC file as compact and as efficient as possible, for faster retrieval and parsing, for example:

- Place high probability checks at the top (such as the check for private IP address space);
- Avoid excessive use of exclusion functions;
- Consolidate bypass criteria into fewer **if()** statements wherever possible;
- Beware the overuse of **isResolvable()**, **dnsResolve()**, and **isInNet()** to avoid potential DNS performance issues;
- There is no need for **else** statements, as the **return** is immediate;
- Remove examples or sections that are not required;
- Comment lines can be important for understanding the intention of sections or commands within the file, however you can remove comments as possible once the code has been finalized.

Normally the closest Zscaler Data Centers are calculated based on geo-location using the user's IP address. The PAC file can be modified to override this behavior and exclude specific Data Centers if necessary. See the article on Zscaler Community for details (<https://community.zscaler.com/t/exclude-a-data-center-in-users-pac-file/153>).

Slide 55 - 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.