Cisco Cloud Web Security

Standalone Deployment Guide

October 2014

Contents

Introduction

Deploy a simple web security solution that does not require any additional hardware. Connect to Cisco's Cloud Web Security service using existing browser settings and PAC/WPAD files.

*Note: we refer to our cloud proxies as towers. You will see the terms "proxy" and "tower" used interchangeably throughout the document.

Cloud Deployment

Deployment is divided into the following three sections

Additional Redirect Methods

There are 4 additional redirection methods that have corresponding deployment guides. Deployment guides for each redirection methods can be found here, under Technical Collateral.

	**Cisco Integrated Services Router (ISR G2 with CWS Connector)**	Save bandwidth, money and resources by intelligently redirecting Internet traffic from branch offices directly to the cloud to enforce security and control policies. Apply acceptable use policy to all users regardless of location.
	**Next Generation Firewall (ASA/ASAv with CWS Connector)**	Capitalize ASA investments by offloading content scanning to Cisco's cloud through CWS. Apply acceptable use policy to the company, groups or individual users.
	**Web Security Appliance (WSA/WSAv with CWS Connector)**	Integrate CWS and WSA to enable identity information to the cloud and extend other on-premises enterprise features to Cloud Web Security customers.
	**Cisco AnyConnect Secure Mobility Client (AnyConnect)**	Authenticate and redirect web traffic securely whenever the end user is off the corporate network. CWS leverages cached user credentials and directory information when they are away from the office or VPN, ensuring that exactly the same web-usage policies are applied.

Verify connection to a tower

Site-to-tower communication is accomplished over TCP port 8080. HTTP and HTTPS requests are sent to a cloud scanning tower in this method. Therefore, TCP port 8080 outbound is required to be open for all users within the organization. For security reasons, Cisco recommends that port 8080 outbound destinations be limited to the scanning towers provisioned for the customer's account.

Reference video: Verify connection to a tower

Step 1: Log on to a client computer inside the customer's network.

Step 2: Click on the Control Panel and go to Programs and Features.

Step 3: Click Turn on Windows features on or off. Scroll down the list of available features until you find the Telnet Client. Check the box and click OK. Now that the Telnet Client is installed, we can resume our test.

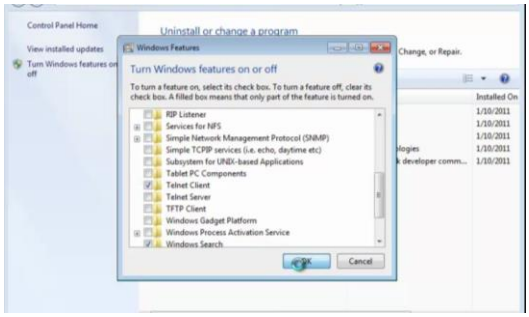

Figure 1.1

Step 4: Open the command line window and type command 'telnet [tower IP address] 8080." A successful connection is noted by a blank screen and blinking cursor.

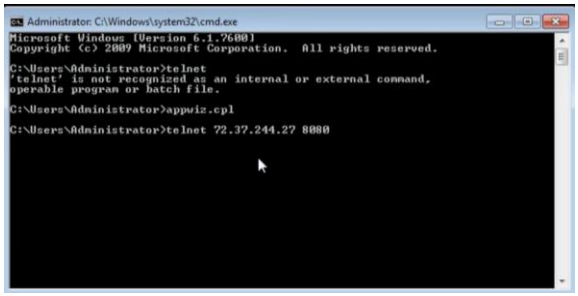

Figure 1.2

Determine your egress IP

When a Cisco proxy receives packets, it must first identify if the received packets originated from a valid customer. To do this, the tower reads the source IP address in the packet and compares it to the IP address registered in its database. If the IP address is found, then the tower receives the packet. If not, then the packet is dropped.

Reference video: Determine your egress IP

Step 1: Open a browser and browse to http://www.whatismyipaddress.com. The returned IP address is the egress IP, or public IP address on the outside interface of your firewall or router.

Step 2: Copy this IP address and log on to the Cisco Cloud Web Security portal at https://scancenter.scansafe.com/

Step 3: From the *Admin* tab, select *Your Account*, then *Scanning IPs*.

Step 4: Paste the IP address in the field provided and click *Submit*. Review the IP address you entered to ensure that it is correct.

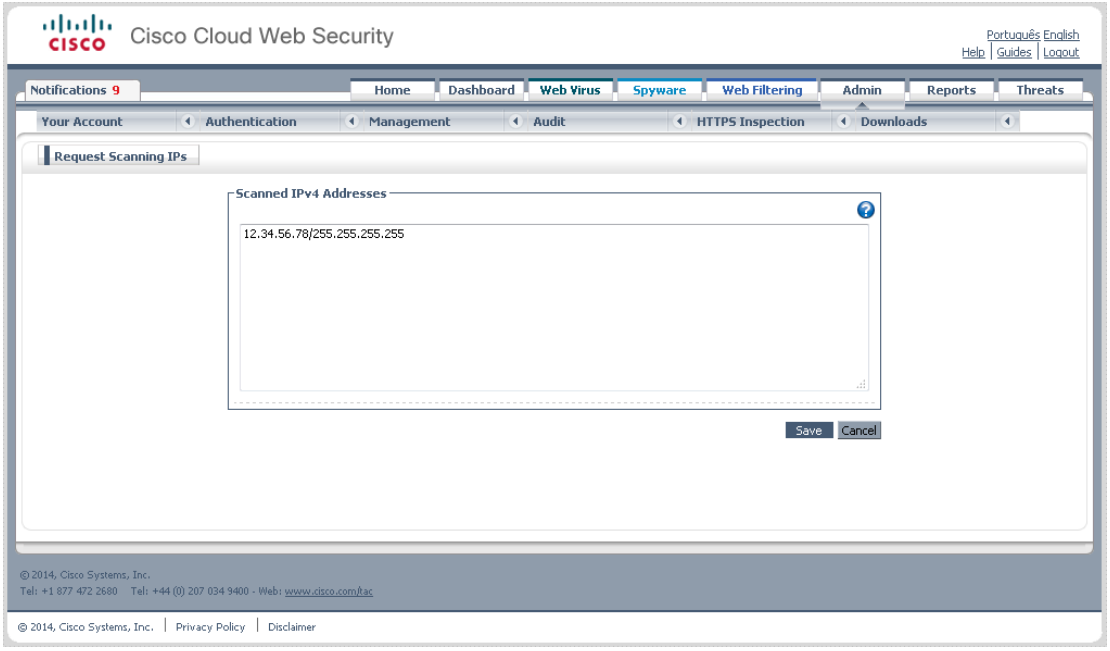

Figure 1.3

*Notice that a subnet mask has been automatically applied to this address. Subnets can be added with the correct subnet mask or in CIDR notation.

Step 5: Next, open Internet Options and set your proxy server settings to your primary scanning tower.

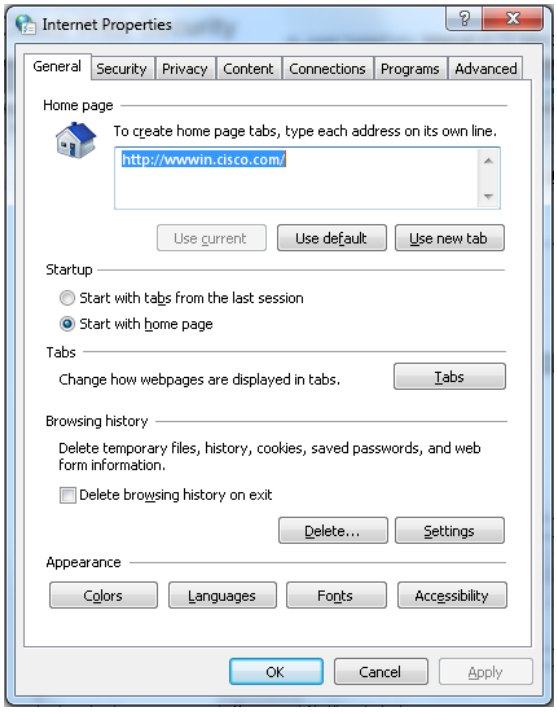

Figure 1.4

Step 6: Check the box *Use a proxy server for your LAN*. Click *OK*.

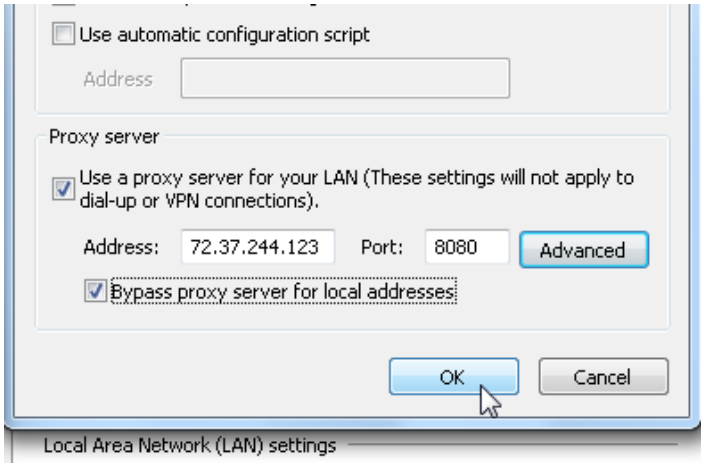

Figure 1.5

Step 7: Return to your web browser and browse to whoami.scansafe.net. You should see your egress IP as your external IP and your company name in the output.

Deploy

Redirect web traffic

The following sections will explain how to redirect web traffic to the cloud using a PAC file. PAC files are used to determine whether to forward a web request to a particular proxy or to send web requests out directly from the client.

This document is intended to provide an overview of the deployment process. For more detailed information and troubleshooting, please refer to the admin guide.

Configure a PAC file

The inner working of a PAC file is JavaScript. Although no script experience is required to craft a PAC file, it is beneficial to understand the fundamentals of how JavaScript works.

Reference video: Configure a PAC file

JavaScript is a cross-platform scripting language usable on Windows, Mac and Linux. As JavaScript is case sensitive, one must use caution when creating and modifying PAC files. Any errors in the PAC file will prevent the PAC file from being read. The browser will simply ignore the PAC file and fail open. A browser that is failed open acts normally, as if no PAC file was configured in the browser at all.

JavaScript Conventions	Description
// (double forward slashes)	a comment; text that is preceded by these characters will not be read
() (open and closed parentheses)	contain arguments for a function
{} (curly brackets)	encapsulates arguments and functions within a function, supersedes parentheses, and cannot be contained within parentheses
|| (double pipes)	a logical OR statement; allows the script to evaluate more than one condition in a function
; (semi-colon)	ends a function, like the period in a sentence

Chart 2.1

A variable is a means of dynamically representing a numeric character value. A variable's value may change each time the PAC file runs.

Declare a variable using the following: var `VARIABLE_NAME=SOME_VALUE`;

Overview of JavaScript Functions

`FindProxyForURL`(url, host)– overall function governing for the PAC file itself. Web browsers look for this function when configured to use a PAC file. The value for the two variables enclosed by the parentheses is defined by the browser. You do not need to define them.

`isPlainHostName(host)`– evaluates the host variable provided by FindProxyForURL to see if the website entered in the browser is simply a host name or a DNS name. If the host variable contains a simple host name and not a DNS name, the value will return True. This is an easy way to determine if web traffic should be forwarded to a proxy or be contained within the LAN.

*Note: Routing of web traffic cannot occur without DNS information. Therefore, a plain host name entered into the web browser is assumed to be in the LAN and NetBIOS will be used to find it rather than DNS.

`isInNet(hostIP, "10.0.0.0", "255.0.0.0")`– evaluates the host variable against two arguments: The first being an IP address or subnet and the other a subnet mask. The purpose is to determine if the host being requested by the client resides at a particular IP address or subnet. If it does, this function returns True.

`shExpMatch(url, "*.domainabc.com*")`– evaluates a host variable against a host, domain, or url. If the two values are the same, the function returns True. Wild cards can be used in `shExpMatch` which make them very useful, but also present more of a load on the client when evaluating them. Consider this when you are determining which function to use in your PAC file.

`dnsDomainIs(host, "vpn.domain.com")`– evaluates the host variable against a static domain host or domain name. DNSDomainis is not compatible with wild cards. This makes them more specific than shExpMatch but also easier on the client to evaluate. If the domain host or domain name appear in the host variable, this function will return True.

`url.substring(0, 4)` ==“ftp;”– evaluates a part of the URL variable string. If there is a match, then it will return true. The first number in the open and closed parentheses defines the start position in the string. Zero is the first character because computers start counting at zero (instead of one). The second number defines how many characters to read from the starting position. For example, if the first four characters in the string are “ftp;”, the function will return True.

`myIpAddress()`– returns the IP address with the highest binding order, which should be the IP address in use. This holds true for OS’s that are not IPv6 enabled. In an IPv6 enabled OS, IPv6 addresses take precedence over IPv4 addresses.

When using the myIpAddress function more than once, use the function in a variable so the PAC file is not harvesting the IP address of the client multiple times.

`isInNet(myIpAddress(), "192.168.1.0", "255.255.255.0")`– rather than evaluating to see if a host is within a certain subnet, it is looking to see if the client is within a certain subnet. This kind of logic can be used to direct groups of clients to different proxies.

Deployment Tip

A conditional statement says that if some condition is true, execute code following the if statement. It is used to make decisions on where to send web traffic.

`return "PROXY 1.2.3.4:8080; PROXY 5.6.7.8 and return "DIRECT";`- tells the PAC file to stop processing code and provide the browser with either a proxy host or group of proxy hosts in fail over order to forward the web requests, or allow the web requests to go to the client. Open a standard PAC file template in Notepad ++ and set the language to Java. The first function is FindProxyForURL.

```
Function FindProxyForURL(url, host) {
      Var hostIP=DNSResolve(host);
// If the requested website is hosted within the internal network
      If (isPlainHostName(host) ||
//          shEXPMatch(host, “*.local”) ||
            isInNet(hostIP, “10.0.0.0”, “255.0.0.0”) ||
            isInNet(hostIP, “10.0.0.0”, “255.0.0.0”) ||
            isInNet(hostIP, “10.0.0.0”, “255.0.0.0”) ||
            isInNet(hostIP, “10.0.0.0”, “255.0.0.0”))
            return “DIRECT”;

// If the hostname matches, send direct.
      If (DNSDomainIs(host, “vpn.dmain.com”) ||
            DNSDomainIs(host, “extranet.domain.com”) ||
            DNSDomainIs(host, “abcdomain.com))
            return “DIRECT”;

// If the URL or protocol matches, send direct,
      If (shExpMatch(url, “*.domainabc.com*”) ||
           shExpMatch(url, “*.domainXYZ.com:*/*”) ||
           url.substring(0, 4) ==”ftp:”)
           return “DIRECT”;

// If the IP address of the local machine is withing a defined
// subnet, send to a specific proxy.
//  if (isInNet(myIpAddress(), “192.168.1.0”, “255.255.255.0”))
           return “PROXY 1.2.3.4:8080”;

// DEFAULT RULE: All other traffic, use below proxies, in fail-over order.
           return “PROXY 1.2.3.4:8080; PROXY 5.6.7.8:8080”;
```

Just below that, there is another variable, `hostIP`. This one is defined in the PAC file and not by the browser. The function dnsResolve`(host)` is taking the host variable and attempting to resolve it using DNS. The result of the DNS lookup is recorded in the variable `hostIP`. Notice the variable is used several times by functions in the script.

Next, is the first If statement. Notice the open parenthesis and accompanying closed parenthesis. All the code contained within is to be evaluated for this If statement. Should the code produce a True result, then the line immediately following the If statement will execute, returning a direct command to the tower. A direct command will tell the browser to perform the web request directly from the client rather than a tower.

The first function in the If statement, `isPlainHostName(host)`, is evaluating if the host variable contains only a host name with no DNS information. If it does contain a host name with no DNS information, then it will return True. The function is followed by a set of double pipes (||). This means that it will additionally evaluate the next function regardless of the result. When functions are chained together with OR statements, only one of them needs to return True for the entire If statement to be true. Therefore, all functions in the If statement need to return false for the If statement to be false.

The first `shEXPMatch(host, “*.local”)` evaluates the host variable against a wild card `*.local`. Notice that it has been commented out. The only reason you should activate and configure this line of code is if you use non RFC 1918 IP addresses inside your LAN and have configured a connection

specific DNS suffix on your client machines.

```
Function FindProxyForURL(url, host) {
      Var hostIP=DNSResolve(host);
// If the requested website is hosted within the internal network
      If (isPlainHostName(host) ||
//          shEXPMatch(host, “*.local”) ||
            isInNet(hostIP, “10.0.0.0”, “255.0.0.0”) ||
            isInNet(hostIP, “10.0.0.0”, “255.0.0.0”) ||
            isInNet(hostIP, “10.0.0.0”, “255.0.0.0”) ||
            isInNet(hostIP, “10.0.0.0”, “255.0.0.0”))
            return “DIRECT”;

// If the hostname matches, send direct.
      If (DNSDomainIs(host, “vpn.dmain.com”) ||
            DNSDomainIs(host, “extranet.domain.com”) ||
            DNSDomainIs(host, “abcdomain.com))
            return “DIRECT”;

// If the URL or protocol matches, send direct,
      If (shExpMatch(url, “*.domainabc.com*”) ||
           shExpMatch(url, “*.domainXYZ.com:*/*”) ||
           url.substring(0, 4) ==”ftp:”)
           return “DIRECT”;



// If the IP address of the local machine is withing a defined

// subnet, send to a specific proxy.

//  if (isInNet(myIpAddress(), “192.168.1.0”, “255.255.255.0”))
           return “PROXY 1.2.3.4:8080”;

// DEFAULT RULE: All other traffic, use below proxies, in fail-over order.
           return “PROXY 1.2.3.4:8080; PROXY 5.6.7.8:8080”;
```

Next is a set of isInNet(hostIP, “10.0.0.0”, “255.0.0.0”) functions to evaluate the defined variable, hostIP against the RFC 1918 subnets and loopback. If the value of hostIP matches any of the four subnets, the function will return True. The logic of this If statement is designed to identify a web request intended for a host residing inside the local network and allow that traffic to originate from the client rather than a tower.

The next If statement is completely commented out. To activate it, remove all the //. This If statement is only performing dsnDomainIs matches, evaluating the host variable against static host names. You may provide a fully qualified domain name if you would like web traffic to originate from the client destined for that domain, or use a DNS host name to be more specific such as cnn.com or images.google.com.

Activate the next If statement in order to configure it. Notice this If statement is If (shExpMatch(url, “*.domainabc.com*”). It evaluates URL strings to see if they are ftp requests. Due to the nature of wild cards, use caution when creating shExpMatch. Make sure to at least include the fully qualified domain name and the leading period so as not to match erroneous fully qualified domain names.

The last If statement: `if (isInNet(myIpAddress(), "192.168.1.0", "255.255.255.0"))` is employing the use of the myIPaddress function embedded in an isInNet function. This will allow you to group computers by subnet to allow you to direct traffic to a particular tower. If your infrastructure can support the myIPaddress function, then you can capitalize on this logic to mimimize the amount of PAC files you need to use in your environment.

At the bottom of the PAC file, there is a lone return function `with` no If statement: return "PROXY 1.2.3.4:8080; PROXY 5.6.7.8:8080";. The return value is an order of proxy servers (i.e. tower) to direct web traffic to. It is the browser instruction function of last resort. This means that if the web request should not originate from a client, it will be forwarded to a tower instead.

Lastly, notice in the proxy string the proxy IP address port numbers are delineated by a semi-colon.

Host a PAC file in the cloud

Hosting your PAC file in the cloud is secure, as only clients originating from one of the scanning IPs configured in the Cisco Cloud Web Security portal will be allowed access to the requested PAC file. It also offers the convenience of being able to host up to 50 PAC files and 5 different versions of each PAC file, as well as offering a single point of management for administrators. There is no need for supporting infrastructure as everything is hosted in the cloud.

Reference video: Host a PAC file in the cloud

Step 1: Go to the *Control Panel* and select *Internet Options*. Click on the *Connections* tab.

Step 2: Click *LAN settings*. Check *Use automatic configuration script*.

Step 3: In the address field, enter the host PAC URL as it appears in ScanCenter (See below).

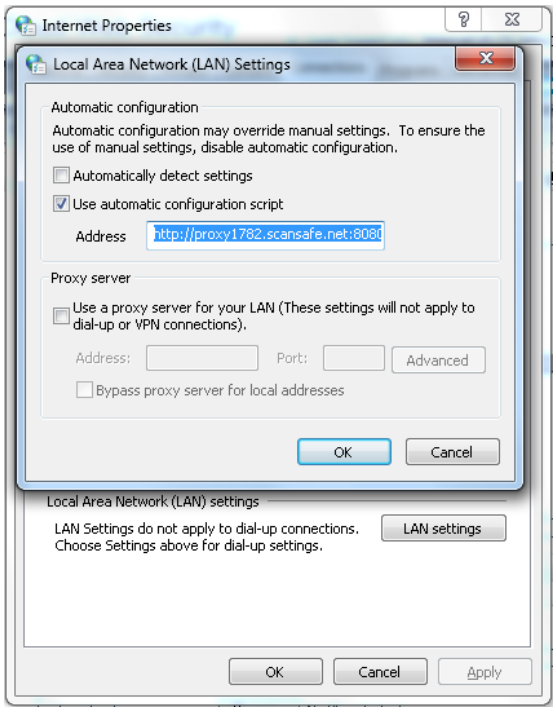

Figure 2.1

Deployment Tip

Notice that the IP is different than the IP of the external IP output. This is because the PAC file is telling the browser to send the web request to the primary scanning tower in the Cisco Cloud . The rest of the world can only see the outbound IP rather than the true egress IP.

Step 4: Open a new browsing session and paste the URL in the address bar to see if the file opens in the browser or returns the download dialogue. If either of these occur, then the browser has access to the PAC file.

Step 5: Browse to http://whoami.scansafe.net. Take note of the external IP.

Step 6: Open another tab and browse to whatismyip.com. Take note of the external IP.

Step 7: Check the PAC file exception by browsing to whatismyip.com. Notice that it returns the true egress IP. That's because the PAC file exception circumvents the Cisco Cloud Web Security Service entirely.

*Note: PAC file exceptions should only be made when the endpoint required is your true egress IP or for a service that is not compatible with a proxy. PAC file exceptions should only be made for trusted entities.

Step 8: Log on to the Cisco Cloud Web Security portal at https://scancenter.scansafe.com. Click the *Admin* tab.

Step 9: Mouse over *Management* and select *Hosted Config*. Click the *Upload Config* subtab.

Step 10: Make sure the resource format is set to PAC. The file name should be unique, contain only alphanumeric characters, and have a file extension of PAC.

Step 11: For file upload, click Browse and select the PAC file that was tested locally. When finished, click *Upload.*

Step 12: Notice the portal returns to the *Manage Config* subtab. Click on the icon of the uploaded PAC file.

Step 13: Check the active box and then select *Save*. Use this process to upload new PAC files, not for modifications or changes to existing PAC files.

Step 14: To make changes to existing PAC files, from the *Edit Config* subtab, use the upload area to upload the updated version of the PAC file.

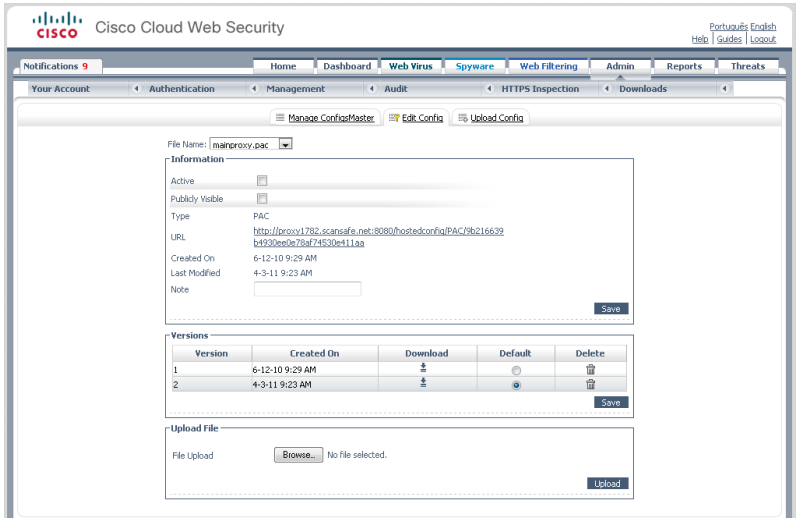

Figure 2.2

Step 15: Once uploaded, the versions area will increment with the new version. Move the default bullet to this version, and click *Save*.

Step 16: From the *Manage Config* sub tab, right click on the URL link and select *Copy link location*.

Step 17: Open *Internet Options*. In the *Connections* tab, click *LAN settings*.

Step 18: In the *Use automatic config script* field, paste the copied URL link. Click *OK*.

Step 19: Open a new web browser and attempt to download the PAC file using the same URL. If the contents of the PAC file appear in the web browser or the download dialog appears, there is direct access to the PAC file.

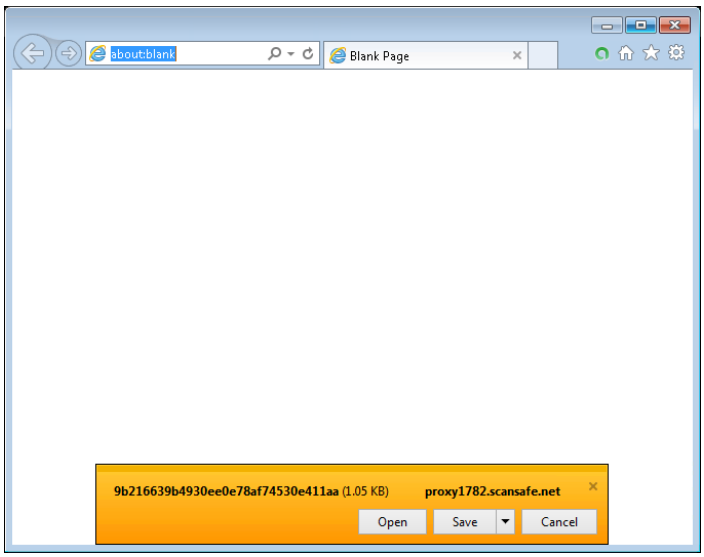

Figure 2.3

Step 20: Browse to http://whoami.scansafe.net and check that you are using the CCWS.

Step 21: Browse to whatismyip.com and see your primary tower's IP address.

Step 22: Browse to whatismyip.com and you should see your true IP due to the PAC file exception for this website.

Deployment Tip

If your domain control is Windows 2003, then you will need to download the Group Policy Manager snap-in from Microsoft. All later versions of Windows Server have this applet built in.

Configure clients to use a PAC file

To configure your managed assets to use a Cisco Cloud Web Security service, configure a GPO to configure a hosted PAC URL to your clients. Do not lockdown proxy settings while in the testing phase.

Reference video: Configure clients to use a PAC file

Step 1: Log on to a domain controller and open *Group Policy Management* to create a new GPO to push your hosted PAC file URL

Step 2: Drill down the tree and select the *Group Policy Objects* folder. Right click on it and select New.

Step 3: Give it a friendly name, such as Cisco Cloud Web Security.

Step 4: Select the GPO. Notice that the links pane is blank because the GPO is not linked to an OU or a domain object.

Step 5: Remove the authenticated users group and add a test user instead. (You only want to test a single user and not the entire organization).

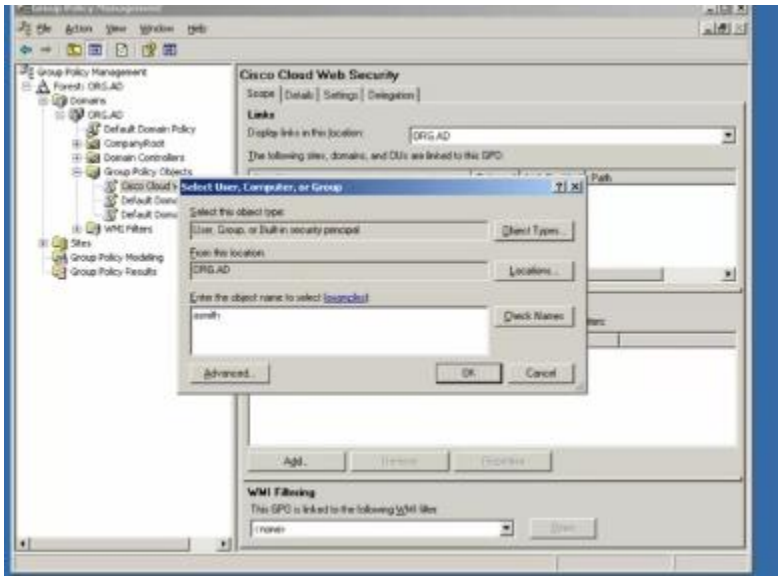

Figure 2.4

Step 6: Link this GPO to the root of the domain by selecting the domain object.

Step 7: Right click the policy and click *Edit* to edit the proxy setting.

Step 8: Drill down through *User Configuration* → *Windows Settings* → *Internet Explorer Maintenance* → *Connection*

Step 9: Remove the check for *Automatically detect configuration settings* and *check Enable automatic configuration.*

Step 10: Paste the hosted PAC URL in the auto proxy URL field.

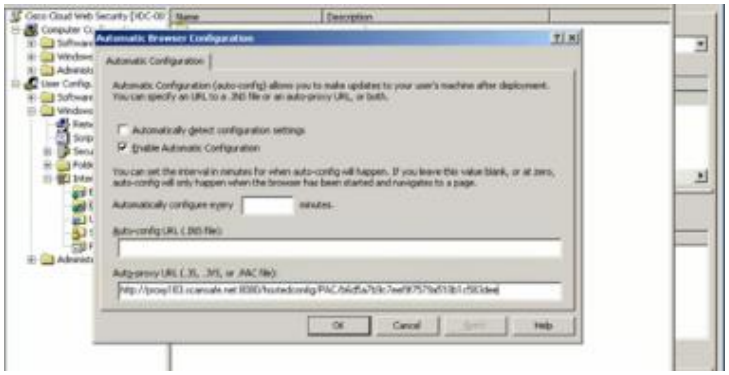

Figure 2.5

Deployment Tip

If the GPO is still not applying, it could be due to a domain controller replication pending or some other GPO issue such as a GPO with a higher priority has taken precedence.

Step 11: Log on to a client whose user account has been associated with the GPO.

Step 12: Open *Internet Options* to see if the proxy settings have been updated. In this example, the proxy settings have not been updated.

Step 13: Open the command line, and run gpupdate/force. As a GPO change is only applying a registry of the client, a log off and reboot is not necessary.

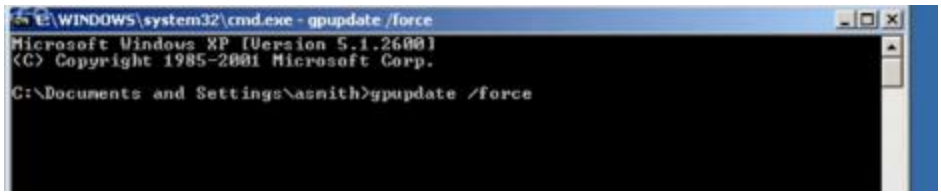

Figure 2.6

Step 14: Go back to the domain controller. Under user settings, expand *Administrative Templates* → *Windows Components* → *Internet Explorer*.

Step 15: Sort the settings to make viewing them easier.

Step 16: Enable the following two settings: *Disable changing automatic configuration settings* and *Disable changing proxy settings*.

Test

Verify web redirection to the cloud

Step 1: From a client machine, browse to whoami.scansafe.net. If a message is displayed, "User is not currently using the service," then the traffic is not redirected to the Cisco cloud. This can be useful in determining if the user is being resolved correctly, any groups being discovered, the internal/external IP of the user/location, and what Connector is in use.

This is an example of a successful whoami.scansafe.net output:

Standalone Deployment Output:

```
authUserName: 12.34.56.78
authenticated: true
companyName: Cisco
countryCode: US
externalIp: 12.34.56.78
groupNames: []
internalIp: 12.34.56.78
logicalTowerNumber: 1782
staticGroupNames:
  - default
userName: 12.34.56.78
```

Standalone Deployment with EasyID/SAML Output:

```
authenticated: true
companyName: Cisco
connectorVersion: Embassy x.x.x.x.x
countryCode: US
externalIp: 12.34.56.78
groupNames: []
internalIp: 12.34.56.78
logicalTowerNumber: 1782
staticGroupNames:
  - default
```

Step 2: From a client machine, browse to policytrace.scansafe.net and enter a URL to see how the web request is processed against the current web filtering policy.

Step 3: With *SearchAhead* enabled in ScanCenter (the CWS admin portal), browse to Google, Bing, or Yahoo and search for something. The *SearchAhead* data should be prepended to each search result in the form of a green, yellow, or red dot. Mouse over the dot to see what information is contained within.

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

Printed in USA C11-727200-00 04/13