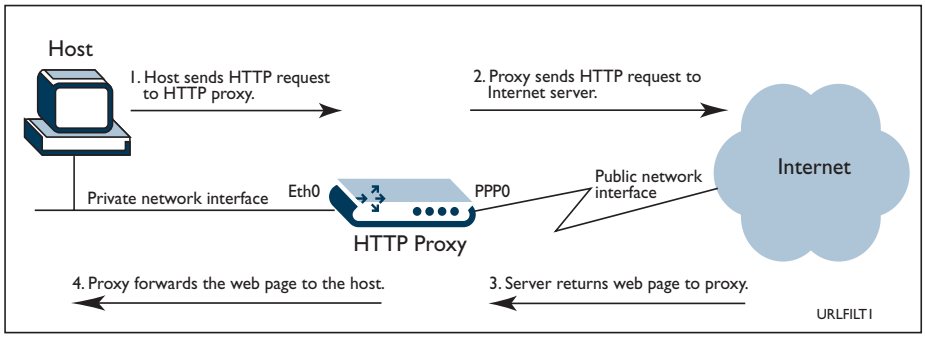
How to Configure URL Filtering Using the Firewall's HTTP Proxy

Introduction

Allied Telesyn switches and routers have a built-in firewall designed to allow safe access to the Internet by applying a set of access rules to traffic passing between the various interfaces on the product. Typically, the firewall has two interfaces; one attached to the Internet (the 'public' network) and one attached to the internal, private network which requires protection.

One feature of the firewall is the HTTP Application Gateway (often referred to as the HTTP proxy). The HTTP proxy acts as an intermediary between the web browsers of users on the private network and the Internet's web servers. When a private-side user attempts to initiate a session with a public-side server (for example, by requesting to download a web page), the session is terminated by the HTTP proxy, which then creates another, separate session with the server. The web page will be returned to the proxy, which then passes it on to the user's web browser. This is illustrated in Figure 1.

Figure 1: Example showing the use of an HTTP Proxy.



This document describes how the firewall's HTTP proxy can be used to filter outbound HTTP sessions based on the URLs requested. This allows the network administrator to prevent users from downloading undesirable resources from the Internet.

Special Feature Licences for Firewall, Application Gateway and HTTP Proxy are required in order to use the HTTP proxy feature of the firewall. These may be obtained by contacting your Allied Telesyn authorised distributor or reseller.

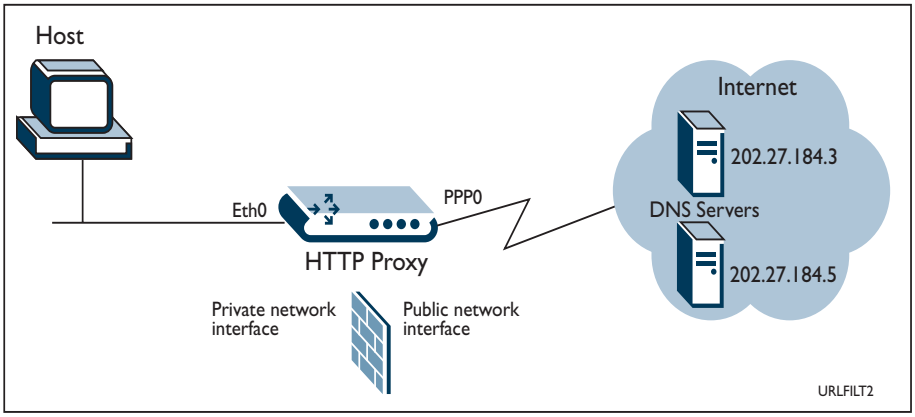
Configuring URL Filtering

Several steps are required in order to configure URL filtering:

1. Specify the DNS server(s) that the HTTP proxy will use.
2. Enable the Firewall.
3. Create the security policy.
4. Add the relevant interfaces to the security policy.
5. Add a firewall HTTP proxy to the security policy.
6. Add an HTTP filter file to the security policy.

The minimum configuration commands for each of these steps follow. Refer to Figure 2 for an illustration of the configuration.

Figure 2: Example of URL Filtering.



The following configuration example assumes that the 'eth0' and 'ppp0' IP interfaces already exist.

1. **Specify the DNS server that the HTTP proxy will use.**

   The HTTP proxy needs to have at least one DNS server in the list of servers it can use to resolve domain names into IP addresses. To add primary and secondary name servers with IP addresses of 202.27.184.3 and 202.27.184.5 respectively, for use as default name servers, use the command:

```
ADD IP DNS DOMAIN=ANY PRIMARY=202.27.184.3
    SECONDARY=202.27.184.5
```

2. **Enable the Firewall.**

   Enable the Firewall using the command:

   ```
   ENABLE FIREWALL
   ```

3. **Create the security policy.**

   Create a security policy named "office" using the command:

   ```
   CREATE FIREWALL POLICY=office
   ```

4. **Add the relevant interfaces to the security policy.**

   Add the Ethernet and PPP interfaces to the security policy "office" using the commands:

   ```
   ADD FIREWALL POLICY=office INTERFACE=eth0 TYPE=private
   ADD FIREWALL POLICY=office INTERFACE=ppp0 TYPE=public
   ```

5. **Add a firewall HTTP proxy to the security policy.**

   Add an HTTP proxy to the security policy "office" using the command:

   ```
   ADD POLICY=office PROXY=HTTP INTERFACE=eth0 GBLINTERFACE=ppp0
       DIRECTION=OUT
   ```

An HTTP proxy can be added to only one security policy.

6. **Add an HTTP filter file to the HTTP filter of the security policy.**

   The HTTP filter is enabled by adding the contents of up to five HTTP filter files to the security policy. The format of these filter files is discussed in "*HTTP Filter Files*" on page 3.

   Add the contents of the file "banned.txt" to the HTTP filter of the security policy "office" using the command:

   ```
   ADD FIREWALL POLICY=office HTTPFILTER=banned.txt
       DIRECTION=OUT
   ```

HTTP Filter Files

The HTTP filter is enabled by adding the contents of up to five HTTP filter files to the security policy. Each filter file is a file type with a .txt extension containing zero or more single line entries. The filter files have two sections:

- Keywords
  This section of the filter file lists keywords that can match any part of a URL. URLs containing any of the listed keywords will be denied unless specifically allowed by an entry later in the filter file.
- URLs
  This section specifies particular URLs to deny, allow or cookie-filter.

The string "keywords:" must be placed at the beginning of the file and is used to start the keywords section. Keywords can be placed on the same line if they are separated by a space or placed on separate lines.

The URL section is indicated by the string "URLS:" as the first word on a line. URL entries may specify a full domain name. If this is the case, there is an implicit "/*" on the end of the domain name. For finer control over the URLs

to match, directory and folder names may be specified. IP addresses may also be specified in the filter file. Only one domain is allowed per line. Options are supplied after the entry and a colon. Each option is separated by a space.

The option keywords that are allowed for each URL entry are:

- Allow
  This option explicitly allows the URL, or part of the URL, given on the line. This is useful for exceptions to a deny filter or a given keyword.
- Nocookies
  This option specifies that the proxy should not accept cookie requests from the domain or URL given, and implicitly allows the URL.

Comments may be placed in the file using a # character at the beginning of the line. White space before and after an entry does not affect the parsing of the file but there must be white space between the URL and colon for the options. After the colon, white space is not needed but there must be white space between each option specified. Empty lines are also allowed.

All URL entries without options are considered to be denied.

How specific the URLs are determines the order of precedence of the entries in the file. For example, *www.plant.com/this/is/a/url/grow.html* would have more precedence than an entry containing *www.plant.com/this*. Also, if the allow option is specified it will have greater precedence than a similar entry with deny. Finally, keywords in the file have the least precedence. They are only applied to sections of the URL not part of the closest fitting URL entry.

Figure 3 on page 5 shows an example of an HTTP filter file.

Figure 3: Example of an HTTP Filter File

```
# The keywords section starts with the string "keywords:".
keywords:
# The keywords can match any part of the URL. URLs containing these entries will
# be denied unless specifically allowed by an entry later in the file.
sex
plants
toys
.nz
# Putting a * in front of the keyword indicates that the string must appear at
# the end of the URL, for the URL to be denied. The following entry would match
www.anything.com/this/is/an/example, but not www.example.com
*example
# The * operator can be used to specify the type of file.
*.mp3
*.jpg

# The URLs section starts with the string "URLS:", and specifies particular URLs
# to deny, allow or cookie filter.
URLS:

# If no explicit deny is put on the end then the URL is denied.
# Note the implicit /* on the end of the domain.
www.plant.com
www.nude.com

# Specific sections of websites can be matched. The sections must be complete
# folder/directory names, so the following entry would match
# www.hacker.com/dosAttack/dos.html but not www.hacker.com/dosAttacks/dos.html
www.hacker.com/dosAttack

# The "nocookies" option denies cookie requests from the domain, and makes an
# implicit allow.
www.acompany.com: nocookies

# The "allow" option can be used to override general URL exclusions.
www.nude.com/this/is/not/porn : allow

# The "allow" option can also be used to override general keyword exclusions.
www.sexy.plants.com : allow

# The "allow" and "nocookies" options can be combined to allow a URL that is
# forbidden by the keywords, but deny cookie requests.
www.acompany.co.nz : allow nocookies
```

Editing the HTTP filter

The HTTP filter is enabled by adding the contents of an HTTP filter file to the security policy. Thus, in order to edit the contents of the HTTP filter list, the HTTP filter file must be deleted from the security policy, edited, then added to the security policy again.

1. Delete the contents of the file "banned.txt" from the HTTP filter of the security policy "office" using the command:

```
DELETE FIREWALL POLICY=office HTTPFILTER=banned.txt
   DIRECTION=OUT
```

The contents of the file "banned.txt" may now be edited.

2. Add the contents of the file "banned.txt" to the HTTP filter of the security policy "office" using the command:

```
ADD FIREWALL POLICY=office HTTPFILTER=banned.txt
   DIRECTION=OUT
```

Note that the HTTP filter file may be edited **before** it is deleted from the security policy. This is because the **contents** of the filter file are added to the security policy using the ADD FIREWALL POLICY HTTPFILTER command. Once the contents of the file have been added to the security policy the filter file itself may be edited or even deleted.

If the HTTP filter file is deleted and the router is restarted, the HTTP filter will not be enabled and an error message will be returned. Similarly, if the HTTP filter is edited and the router is restarted, the HTTP filter will reflect the contents of the edited filter file.

Updating the HTTP filter automatically

It is possible to introduce a degree of automation into the updating of the HTTP filter. This could be achieved in several different ways, depending on the specific needs of the network administrator.

One possible technique is to use a script file to delete the HTTP filter file, download an updated filter file from a TFTP server, delete the filter file from the security policy, then finally add the filter file to the security policy again. An example of the contents of such a script follows:

```
DELETE FILE=banned.txt

WAIT 300

LOAD FILE=banned.txt SERVER=192.168.40.254

WAIT 300

DELETE FIREWALL POLICY=office HTTPFILTER=banned.txt
   DIRECTION=OUT

ADD FIREWALL POLICY=office HTTPFILTER=banned.txt
   DIRECTION=OUT
```

This script could be activated using the trigger facility. For example, if the commands shown above were contained in the script file update.scp, this script could be set to run at 1:00am each day using the command:

```
CREATE TRIGGER=1 TIME=01:00 SCRIPT=update.scp
```