Fujitsu mPollux

WebFront

White Paper

Fujitsu mPollux Version 2.1

June 2016

Table of Contents

1 INTRODUCTION

Fujitsu mPollux™ Security Server is Fujitsu Finland's multifunction security solution, which can be easily adapted for the diverse needs and security level requirements of user organizations. This is made possible by the modular architecture of mPollux™ Security Server: the Server offers a number of Security Options and the user organization chooses the Option(s) that best suit(s) its needs.

This white paper describes the functionality of **mPollux™ WebFront Access Control** that is designed to provide access control in multichannel environments.

2 BASIC FUNCTIONALITY

The mPollux™ WebFront is a module that can be used to implement access control for Web and non-Web applications. mPollux™ WebFront is a dedicated module in separate hardware.
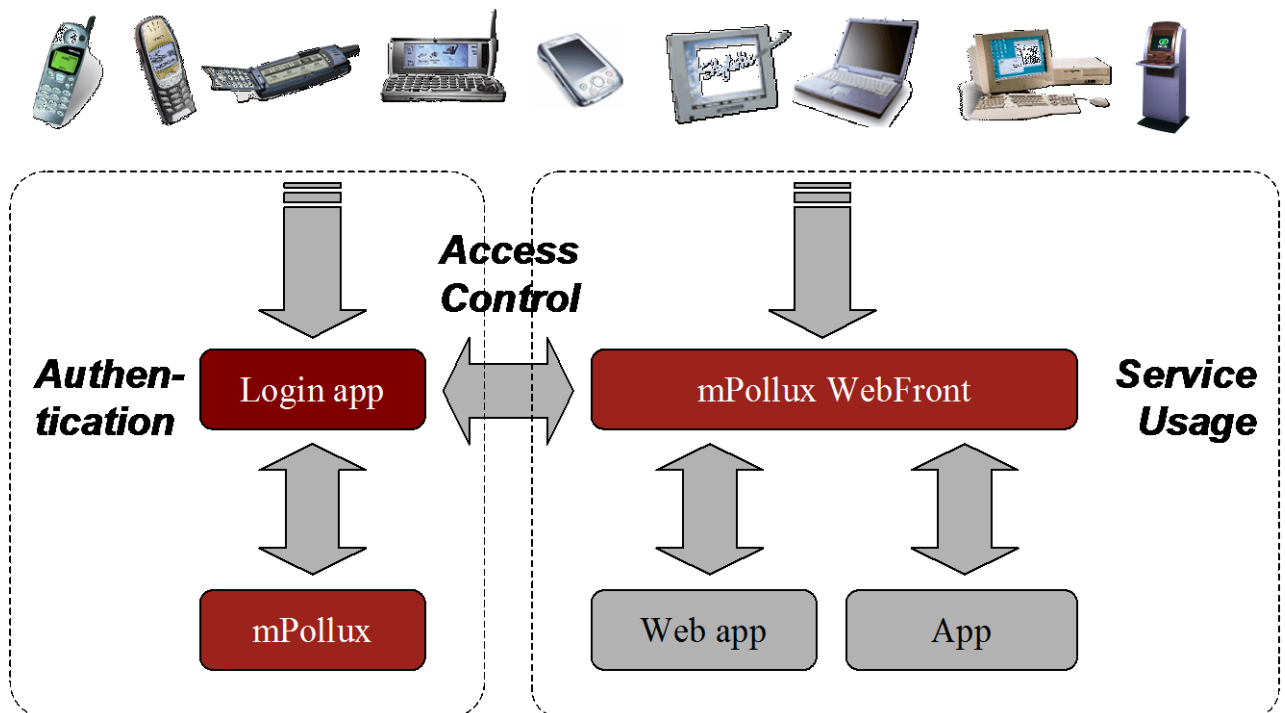
Figure 1. WebFront Usage Environment

The normal working environment for mPollux™ WebFront is described in *Figure 1*.

Users are accessing different applications using a variety of devices. The applications can be either Web applications, i.e. accessed with a Web browser. Or they can be non-Web applications, where they are accessed using a dedicated client.

mPollux™ WebFront is installed from the network point of view in front of the applications so that it is able to control all communication targeted to the applications. mPollux™ WebFront checks whether the users trying to access the applications have been authenticated, and relays the requests of authenti-

cated users to the desired applications. Requests from non-authenticated users are relayed to login applications.

Login applications are responsible for the authentication of the users and controlling mPollux™ WebFront according to the authentication results. Users can be authenticated in different ways utilizing **mPollux™ Security Server**, e.g. using userid and password, mobile phone or smartcards. After successful authentication, the login application will command mPollux™ WebFront to allow the user to access the desired application.

The applications controlled by mPollux™ WebFront are **HTTP proxy mode applications**. These include Web applications, where mPollux™ WebFront is acting as an HTTP reverse proxy controlling all HTTP requests sent to configured applications. It takes the role of the HTTP or HTTPS server it is controlling access to, and verifies authentication tokens included in the HTTP data stream. The authentication tokens are generated for the client's browser application by a login application upon a successful login, and the browser sends the tokens with every HTTP request to prove its authorization. mPollux™ WebFront forwards the request to the actual content server only if the client's authentication token is accepted, or redirects the client back to the login application if the token fails verification.

The following chapter describes more detailed the HTTP proxy mode.

3 WEBFRONT USE SCENARIOS

The use scenarios for ™ WebFront are described using cases where mPollux™ WebFront is controlling access to different types of applications.

3.1 HTTP PROXY MODE

The HTTP Proxy Mode usage scenario covers access control to Web applications in the internal network, i.e. the cases where connectionless HTTP protocol is used. An HTTP session consists of a number of TCP connections, typically one connection per object in a web page, and it is often proxied by caching intermediary systems, causing the source address of the connections to change in an unpredictable fashion. When the source address cannot be relied upon, IP level access control as performed by traditional firewalls and mPollux™ WebFront firewall mode becomes impossible.

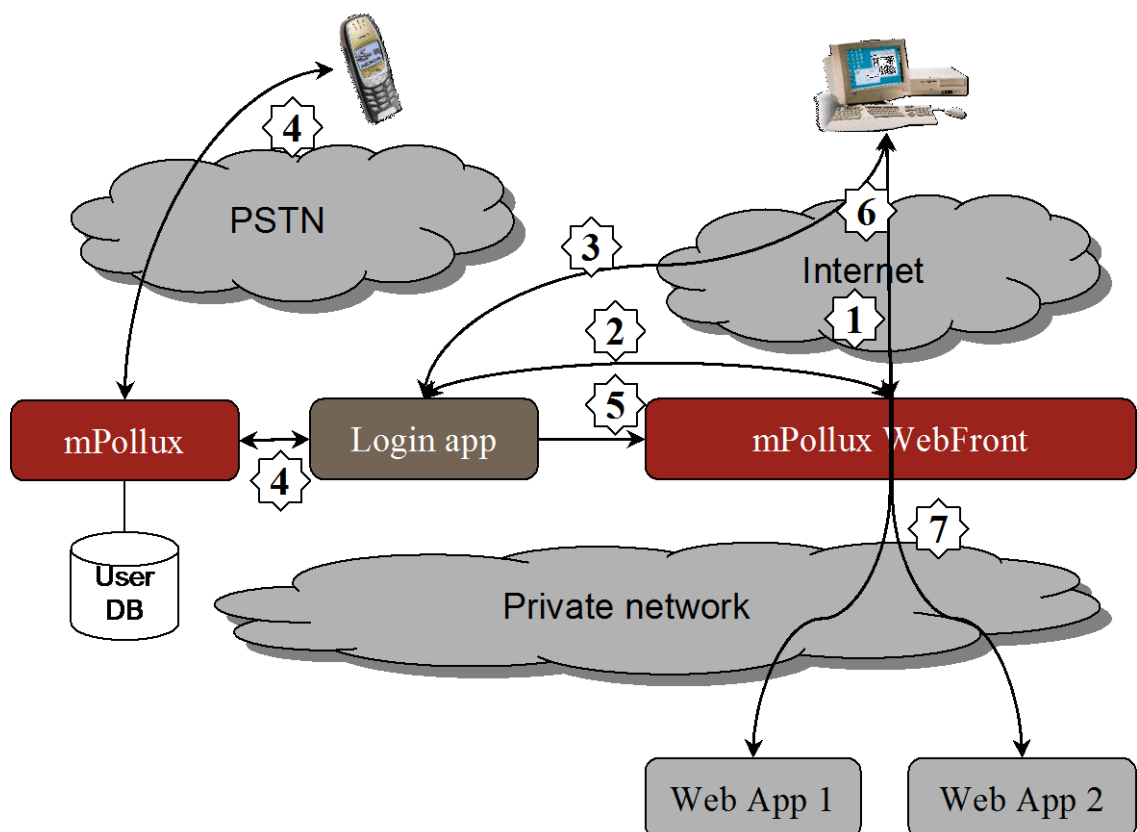There can be several web applications whose access is controlled by mPollux WebFront™.

Figure 2. HTTP Proxy mode

1. The user enters to the browser the URL for the service. mPollux™ WebFront will receive the related HTTP request.

mPollux

2. mPollux™ WebFront recognizes from the missing authentication cookie a non-authenticated user and redirects the user to the login application.
3. The user identifies himself through the web screen provided by the login application e.g. by entering his phone number.
4. The login application authenticates the user utilizing mPollux™.
5. After successful authentication, the login application commands mPollux™ WebFront to create a session for the user. The session information includes an authentication cookie and custom cookies.
6. The authentication cookie is stored into the user's browser as a session cookie and is verified by mPollux™ WebFront for each request.
7. Custom cookies can be used to deliver user information to target applications. They can contain e.g. user ids and groups and other information identified by the login application.
8. Finally, the user id redirected to the target application.

mPollux™ WebFront supports the building of Single Sign-On solutions in different ways. First, mPollux™ WebFront can deliver customizable user identification information to applications. This information includes basic authentication information and customized cookies controlled by login applications. Secondly, the URL used to redirect to the application in the step 8 can emulate a request created by a login page for an application including id and password information. Login applications for mPollux™ WebFront can also perform emulated logins to applications where more complicated logins can be automated.

mPollux™ Security Server supports also storing of SSO related data to its database, such as user ids and passwords for different applications.

4   WEBFRONT BENEFITS

- Access to company's internal applications from Internet
  Using mPollux™ WebFront you can securely provide access to internal applications, such as e-mail, calendar, from Internet.
- Simplify application logic by isolating authentication and access control
  Using WebFront you do not have to implement authentication and/or access control within your application, but in a separate module. In this way, the application logic is much simpler since you do not need to check for session or authentication information in all pages.
- No need to modify applications
  Since mPollux™ WebFront performs authentication and access control in separate modules, you can also secure access to existing applications without the need to modify them.
- Single sign-on, there are several features in WebFront supporting the building of SSO solutions. See the previous chapter for more information.
- Authentication, access control and session management for applications not having them
- Tailorable login applications
  With mPollux WebFront™, you can freely choose the way to authenticate users.

5 WEBFRONT FEATURES

General features:

- Runs on Linux Red Hat Enterprise Edition 6 (64-bit), and Suse 11 (64-bit) operating system
- Up to three 10/100 Mbps or 1 GB Ethernet links
- Up to 15 IEEE 802.1q tagged VLAN interfaces,
- Multiple alias addresses for each interface
- Bandwidth limiter allowing fine-grained traffic flow control;
- Full IP NAT and address redirection control;
- IPSEC for IPv4 traffic encryption and authentication for VPN solutions;
- An optional domain name service (DNS);
- An optional dynamic host configuration protocol service (DHCP);
- Full IPv6 support.
- XML configuration
- Comprehensive logging facility collecting information about connection requests and sessions. Supports also redirection of logs to an external logging server.
- The automatic control of session lifetime. After specific time WebFront will destroy the session and reject requests sent to target applications. Timeout value can be defined as an absolute time or an idle time.
- Control interfaces available as a Java API, C-language API and a COM+- interface.

HTTP Proxy mode features:

- Both HTTP and HTTPS protocols supported. Supports the mixture of protocols, e.g. HTTPS in the public network and HTTP in the private network. In this way also applications supporting only HTTP can be accessed confidentially from public networks.
- Automatic redirection of non-authenticated clients to the pre-configured login page.
- Authentication information stored as default into a built-in cookie, which is encrypted session information. The system supports also the customization of the cookies by adding external cookie validation servers.
- Internal configurable error pages