Memova-based webmail vulnerability: auto-forwarding setting tampering

Rosario Valotta

Matteo Carli

Index

Overview

In this advisory, we have analyzed some webmail applications based on the commercial framework Memova, developed by Critical Path.
By leveraging on some vulnerabilities (descripted later), an attacker is able to tamper victim's mail settings, allowing incoming e-mails to be automatically forwarded to an attacker controlled mail account.
In such a way, is possible to violate the confidentiality of all e-mail communications without using the common attacks based on "identity stealing" (cookie stealing, credential stealing).
All the attacker needs is to send a specially crafted e-mail to a victim.

The danger of such approach is made more critical by these factors:

1. ease of exploitation
2. lack of awareness of the victim
3. spread over the web

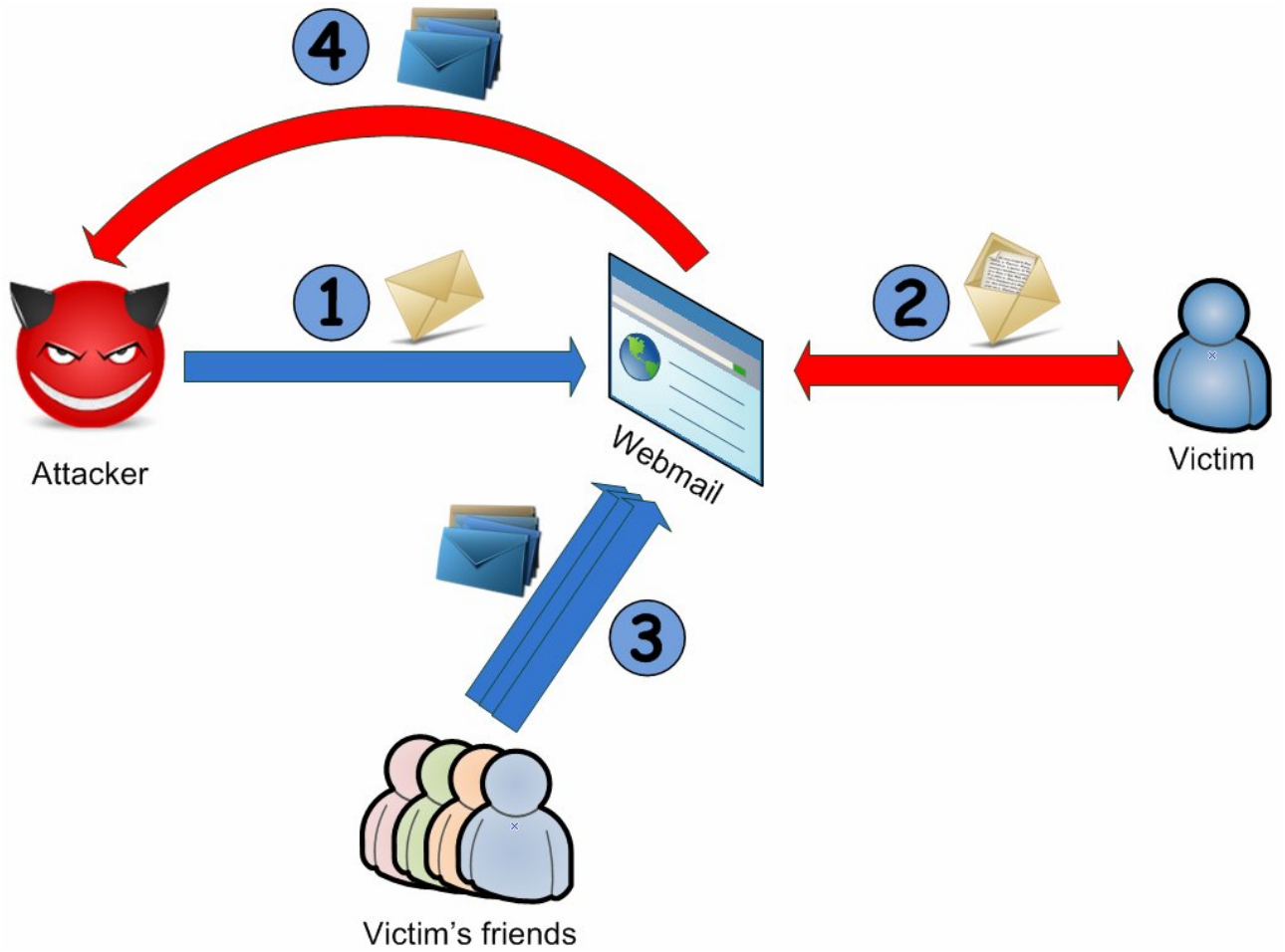

Image 1 – Vulnerability flow chart

Vulnerability exploitation

The attacker only needs to send a specially crafted e-mail to his victim; as soon as the victim opens the mail (no further interaction required) the forwarding settings of his webmail account are silently modified.
In order to craft the email, the attacker can use some widespread tools that allow HTTP tampering (e.g. Firebug, IE http analyzer, OWASP WebScarab, Burp, etc.)

Victim awareness

Auto forwarding settings are usually available under the "Settings" or "Options" menus of the webmail applications. Anyway, these settings are rarely checked by users (usually once, after the mail account has been created), so a stealth tampering of these settings can be unnoticed for a very long time.
In some scenarios (e.g. Libero), users have no chance at all to define mail forward settings through the webmail menu; in such scenarios is impossible for the victim to notice the tampering effects and only with the ISP customer support he/she can restore the former configurations.

Diffusion over the Web

In this advisory, we have analyzed the webmail applications based on the commercial framework Memova, developed by Critical Path. (http://www.criticalpath.net).
Memova is a worldwide spread solution for creating webmail applications: a customers snapshot is available at http://www.criticalpath.net/en/110/ourCustomers/.
They include:

- Tiscali IT/UK/NL
- Wind
- Telecom
- Vodafone
- Swisscom
- Telefonica
- Virgin
- Sonera
- Terra.es
- Telia
- T-Mobile
- FastwebNet
- Ono
- Regione Puglia
- Regione Sicilia
- Some gov.uk domains

Majority of them are worldwide known ISP, with a huge customer base; each of these ISP has deployed on their web portal a webmail service based on Memova framework. Each of these deployments have different look&feel characteristics and some minor differences in terms of features but, all of them are exposed to the same vulnerabilities.
This is the reason why an attacker can send the same e-mail to an extremely long list of recipients (all from different ISP) and can tamper the forwarding settings of million accounts at the same time. The effect of the attack is that the attacker will receive all the incoming mail of the victims as he is in Cc of each message.

In order to get an idea of "live" deployments of Memova messagging platform you can try the following Google search:

http://www.google.it/search?hl=it&q=%22cp%2Fps%2FMail%22&meta=

where "cp/ps/Mail" is a common prefix for all Critical Path Memova deployments.

On the basis of the informations found over the web (ISP and Critical Path references) we can state that more than 40 million mail accounts worldwide are vulnerable to this attack.

Proof of Concept

For the current PoC, we have analyzed three among the more popular italian webmails:

- Tiscali
- Libero (Wind)
- Virgilio (Telecom)

We will show that by crafting a mail with a unique attack vector (a text string capable of bypass Memova security check) an attacker can compromise three different mail account (one for each provider) setting the forwarding of incoming e-mails towards an account controlled by the attacker.

On the basis of an ethical choice the source code of this PoC will not be publicly disclosed.
The aim of this work is only documenting the vulnerabilities and make the customers aware of the risks connected to the use of some webmail applications.

Basics

The attack is based upon two different vulnerabilities found on the analyzed webmails:

- Cross site scripting – XSS
- Cross site request forgery – CSRF

For a deep understanding of these topics, please check the technical documentation available over the web. Basicly, XSS (Cross site scripting) is a vulnerability related to the injection of malicious code (es. Javascript) into a webpage visited by the victim, exploiting a lack of input validation of target webpage. CSRF (Cross site request forgery) is a vulnerability that allows an attacker to trigger stealth arbitrary HTTP requests from a webpage visited by the victim.

Assumptions

The current PoC has been developed and tested using Internet Explorer 7; this means that using a different browser tha code of the PoC must be slightly modified (each browser has a different – not 100% standard Javascript engine).
So we assume that both the attacker and the victims use IE7.
We also assume that the attacker would like to tamper the following mail accounts:

- Victim1@libero.it
- Rosario.valotta@virgilio.it
- Victim3@tiscali.it

Step 1 – mail sending

In order to send the mail to his victims, the attacker uses the account “tentacoloViola” provisioned on Libero webmail; we have chosen this webmail as is extremely simple to tamper mail parameters (no filters are present on server side).
Among the parameters sent in the POST we find:

- HtmlText – contains the HTML code of the mail body
- Text - contains the text of the mail when a non-HTML message is sent

Applying a particular tampering on these two parameters (technical details will not be provided) is possible to escape anti-XSS filters of all tested webmails.
Tough the PoC has been tested only on the 3 webmail descripted above, is highly probable that others webmails will suffer from the same vulnerability using the same or a slightly different attack vector.

Step 2 – Mail reading

Once the victim opens the email, the XSS vector is injected into the HTML code of an iframe responsible for displaying the mail body. The injected code will recall a remote Javascript file (remote XSS scenario) that is downloaded and executed in the iframe context (e.g. mioisp.it).
Two of the three webmail tested have developed a protection mechanism for reducing the impact of a XSS + CSRF vulnerabilities: the iframe domain is different from the webmail domain (es. Mail.mioisp.it)
This trick by leveraging the browser native security restrictions (same origin policy) prevents attacker from:

1. get the session token (mandatory for committing any command to the application) from the document.location of the browser
2. initiate XmlHttpRequest towards webmail domain

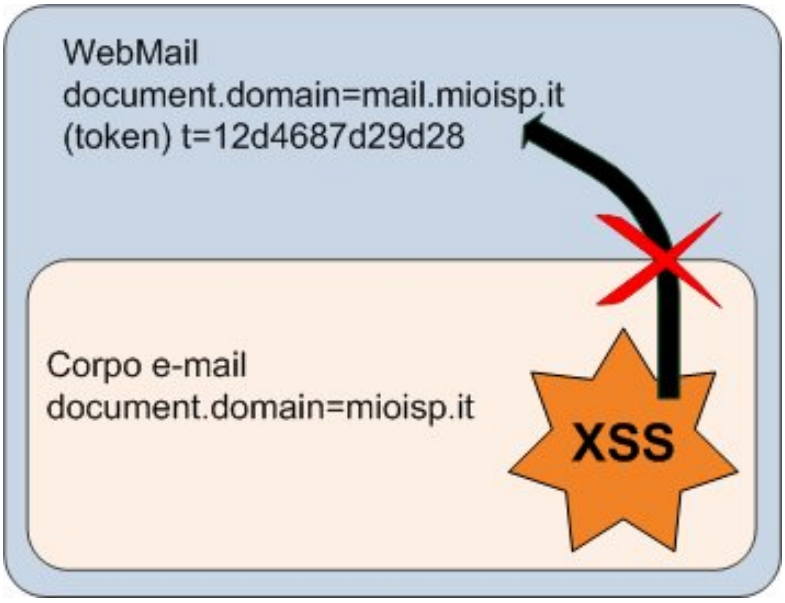

Image 2 - Same Origin Policy prevents the script from accessing data outside his execution domain

There is however a workaraound to bypass this protection:

1. find another XSS (reflected XSS) on the webmail domain (mail.mioisp.it) not tied to the session token (we haven’t it yet…)
2. at runtime we create another iframe nested into the body iframe and we set his source location to the reflected XSS URL
3. the reflected XSS will access the document.location of the webmail (same domain) and read the session token
4. the reflected XSS can now launch a CSRF attack towards “mail.mioisp.it” domain in order to tamper forwarding settings

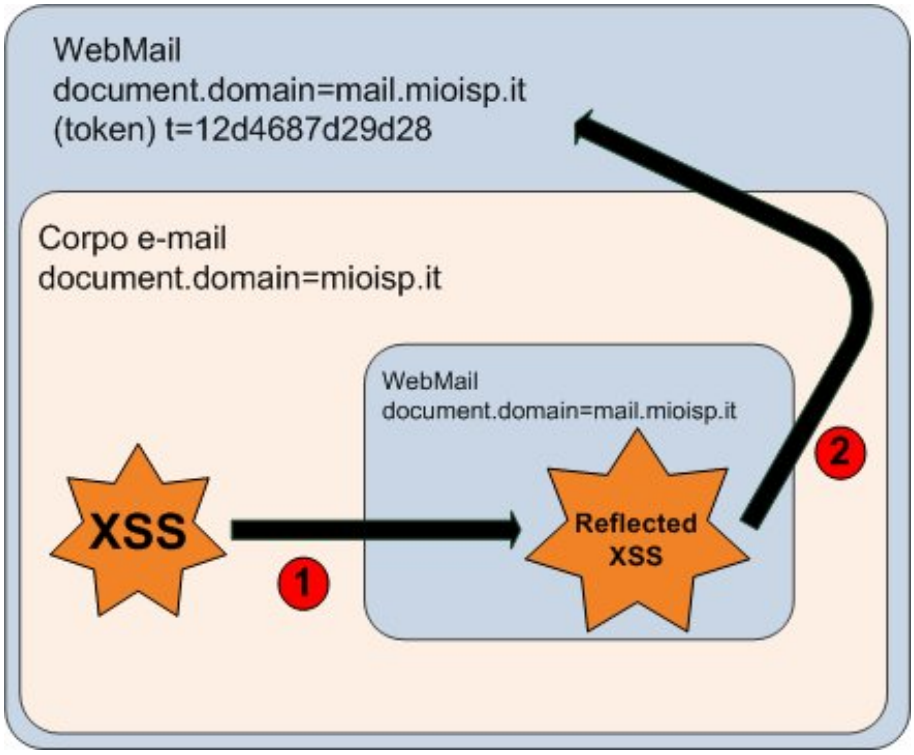

Image 3 - Reflected XSS bypasses Same Origin Policy webmail protection

Step 3 – Forwarding setting

Once the Same Origin Policy has been bypassed and the session token recovered, the reflected XSS can initiate XmlHttp requests towards any resource in the webmail domain (eg. Mail.mioisp.it).
Going into detail, the URL used to set automatic forwarding of incoming mails is:

- POST /cp/ps/Mail/preferences/SetForward?

It is important to underline that this URL is always available (on the tested webmail), even if the forwarding option is not available for the end users (not linked on the web application).
In such a scenario, the tampering of the forwarding settings is completely invisible for the end user and not reversible without ISP customer care support.

Alternative scenarios

Theoretically, these vulnerabilities could be exploited in order to create a worm that steals the victims' contacts and self-propagate to million of e-mail accounts.

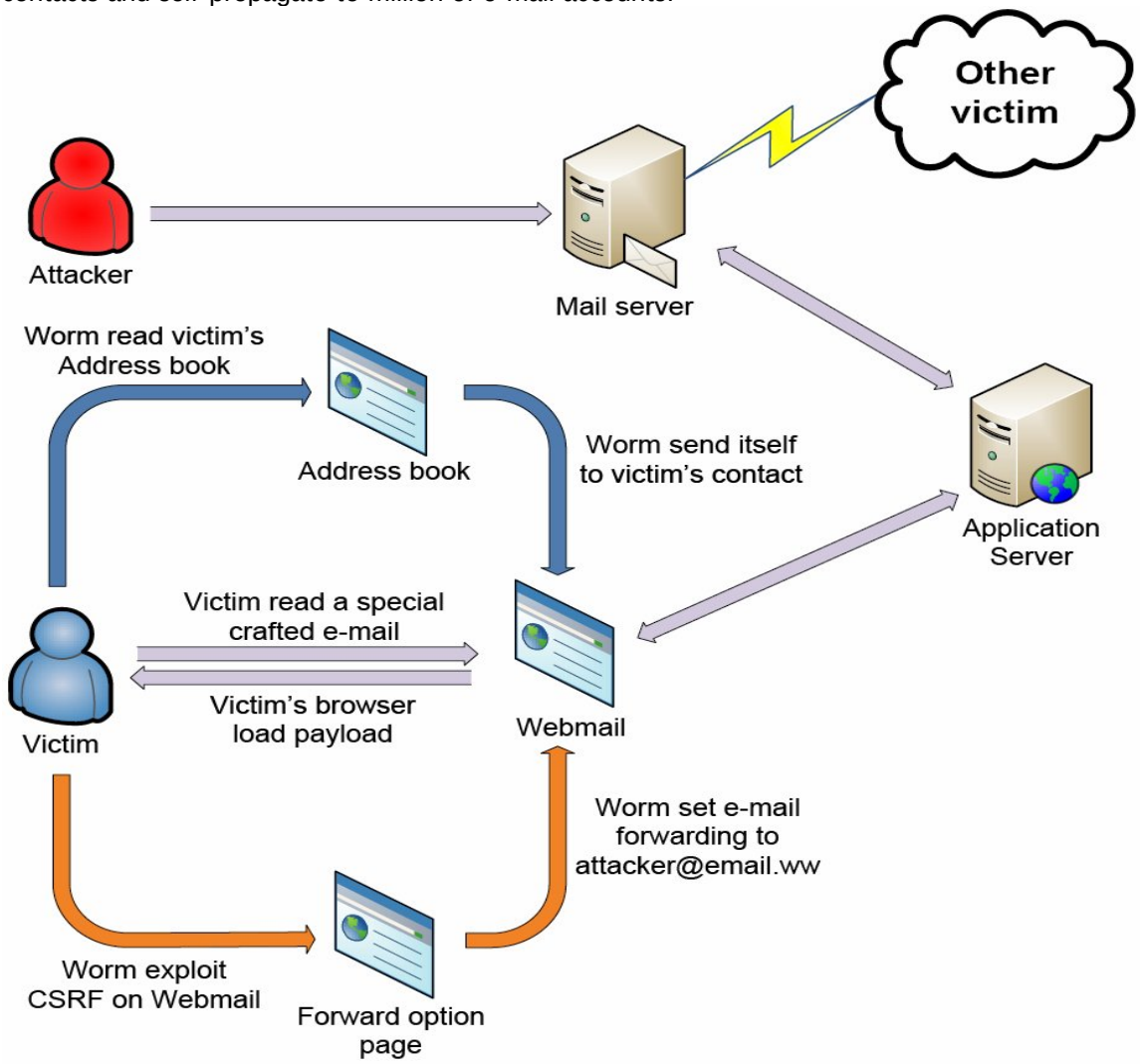

Image 4 - Worm flow chart