Avis de sécurité 2007-003

Objet	Vulnérabilité dans *uPortal*
Référence	ESUP-2007-AVI-003
Date de la première version	25 juillet 2007
Date de la dernière version	16 octobre 2007
Source	liste de diffusion *jasig-members* du consortium JASIG
Diffusion de cette version	Publique
Historique	25 juillet 2007 : réception de la vulnérabilité 27 juillet 2007 : validation de la vulnérabilité sur le *package uPortal-esup* par le consortium ESUP-Portail (Julien MARCHAL) 31 juillet 2007 : accord de Bill THOMSON pour repousser l'annonce publique de la vulnérabilité au 15 août (à la place du 8 août) 3 août 2007 : test du patch proposé et retour (négatif, ne marche que pour uPortal 2.6) à Bill THOMSON (Vincent MATHIEU) 6 août 2007 : diffusion de la vulnérabilité aux correspondants sécurité du consortium ESUP-Portail 15 août 2007 : mise en ligne d'un nouveau correctif (Susan BRAMHALL) 18 août 2007 : validation du nouveau correctif (Vincent MATHIEU) 21 août 2007 : envoi du correctif aux correspondants sécurité du consortium ESUP-Portail 30 août 2007 : annonce de la nouvelle date de la diffusion publique 'Bill THOMSON) 16 octobre 2007 : annonce publique et simultanée de la vulnérabilité par les consortiums JASIG et ESUP-Portail
Pièces jointes	**`ESUP-2007-AVI-003-COR.zip`**

Risque

Usurpation de l'identité des utilisateurs dans *uPortal* par récupération de l'identifiant de session.

Utilisation et diffusion de ce document

Les avis de sécurité du consortium ESUP-Portail portent sur des vulnérabilités des logiciels diffusés par le consortium. Il est de la responsabilité de chacun des destinataires de ce document de ne pas le rediffuser en dehors du cadre pour lequel il a été écrit, pour des raisons évidentes de sécurité des Systèmes d'Information de tous les établissements du consortium ESUP-Portail.

Pour plus de renseignements : ***contact-tech@esup-portail.org***

Systèmes affectés

- Toutes les distributions *uPortal* depuis la version 2.1
- Toutes les distributions uPortal-esup

Résumé

uPortal est distribué avec une configuration *proxy* qui autorise une attaque de type *Cross Site Scripting* (*XSS*).

Description

Un pirate peut introduire du code *Javascript* arbitraire dans le rendu de *uPortal* s'il fait ouvrir par le navigateur client une URL du portail malicieusement construite. Les possibilités d'attaque par le code *Javascript* incluent notamment la capture de l'identifiant de session, autorisant alors l'usurpation de l'identité de l'utilisateur.

Solution

1. Installer les classes du correctif **`ESUP-2007-AVI-003-COR.zip`** dans les sources de *uPortal*. Le correctif peut être téléchargé depuis le site web du consortium ESUP-Portail (*http://www.esup-portail.org*)
2. Positionner la propriété **`org.jasig.portal.serialize.ProxyWriter.resource_proxy_enabled`** du fichier **`portal.properties`** à **`off`**.
3. Commenter ou supprimer la *servlet* **`HttpProxyServlet`** dans le fichier **`WEB-INF/web.xml`**
4. Redéployer *uPortal*
5. Redémarrer *Tomcat*

XSS Vulnerability Proxy Exploit Notification
William G. Thompson, Jr. <wgthom@rutgers.edu>
30 août 2007

This is a private notification of an identified uPortal security vulnerability and workaround. JA-SIG intends to reach as many known uPortal adopters as possible with this information privately prior to public announcement. Public announcement will be made approximately 2 weeks from this notice.

XSS Vulnerability Proxy Exploit

Summary:

uPortal ships with a proxy configuration that allows illicit cross-site scripting. The Adversary can introduce arbitrary JavaScript into a user's portal render cycle if he or she can get the end user's web browser (e.g. via a hyperlink or a redirect) to open a cleverly crafted portal URL. The kinds of things the Adversary could

accomplish with this JavaScript include capture of the user's otherwise secure session cookie, thereby allowing session hijacking.

Versions Affected:

All versions prior to uPortal 2.6.0, including 2.1, 2.2, 2.3, 2.4, 2.5.

Issue:

`HttpProxyServlet`:

uPortal ships with a proxy servlet allowing it to proxy over SSL content that would otherwise be vended in the clear so that the annoying "Mixed content" browser advisory message can be avoided. While some (many?) uPortal deployments are not intentionally making productive use of this servlet, its default configuration is to be nonetheless latently available and so available for exploit using a cleverly crafted URL. You can turn off the feature of proxying resources via the portal.properties property **`org.jasig.portal.serialize.ProxyWriter.resource_proxy_enabled`** but this will not turn off the vulnerability. When this feature is deliberately used, its use involves detecting URLs needing to be re-written via a SAX filter and re-writing them to point at the proxy servlet which then proxies the originally intended content.

CWebProxy:

The web proxy channel's proxying of URLs specified at runtime is already throttled by configuration parameters added to address a previous security vulnerability (that of illicitly proxying local files). Portal administrators should configure each web proxy channel instance as restrictively as possible, such that only trusted content is included within the scope of allowable proxies. Web proxy channels configured such that they will proxy arbitrary content provided by the Adversary can in theory be exploited to execute cross-site-scripting attacks.

Resolution:

Http proxy servlet

The uPortal Http proxy servlet will no longer proxy any content other than elements with a mime type of image. Two changes are involved:

1. `org.jasig.portal.serialize.ProxyWriter`

Summary:

Applets, IFrames and Scripts are no longer rewritten by the serializer to use the proxy servlet.

Description:

When uPortal serializes its response to the browser it passes the response through a filter (**`org.jasig.portal.serialize.ProxyWriter`**) which inserts additional URL path elements as configured by the property, **`org.jasig.portal.serialize.ProxyWriter.resource_proxy_rewrite_prefix`**. In the past the **`ProxyWriter`** class operated on 6 elements: "**`image`**","**`img`**","**`script`**","**`input`**","**`applet`**","**`iframe`**".

The patched version will operate on three element types only: **`image`**, **`img`** and **`input`**. Although it may be convenient to proxy other elements to avoid the mixed content message, this vulnerability response opts for a simpler approach of avoiding entirely these other types of content proxying.

This change comports the proxy writer behavior with the now-more-restricted scope of the **`HttpProxy`** servlet -- this change to **`ProxyWriter`** does not itself secure anything, as the nature of the exploit is to generate proxy servlet instructions by means other than the proxy writer.

2. `org.jasig.portal.HttpProxyServlet`

Summary:

`HttpProxyServlet` will respond with an error code (404) and no content for all objects that do not have a MIME type that begins with: "**`image`**".

Description:

This internal proxy handles requests for the proxied embedded content as the browser renders the response. Previously, the proxy servlet would proxy most anything. With this change, the proxy servlet will proxy only images. Returning only elements with proper mime type of image the browser is able to protect itself from illicitly proxied scripts -- in order for the servlet to proxy, the content must appear to have a mime type of "**`image`**", which will discourage the web browser from executing the content.

Alternative solutions for resource proxy:

Deployments not using the **`HttpProxyServlet`** should remove the class file (in **`WEB-INF/classes`**) and servlet declaration (in **`web.xml`**) from their portals entirely as an unnecessary security vulnerability surface.

uPortal deployers should consider running the proxy on a host other than the portal or other application host, such that if the proxy is exploited to proxy JavaScript, the JS will appear to be coming from an unrelated domain and so in-brower cross-site-scripting protections will apply.

uPortal deployers should consider running external standalone web proxy solutions, such as Squid. External dedicated proxy solutions have their own security issues and configuration needs, but these tend to be well documented and worked through by a larger community dedicated to the problems of proxying.

uPortal 2.6.0 GA ships with a fix similar to the attached patches, pre-applied. An alternative to patching an existing install to deal with this issue is therefore a full upgrade to uPortal 2.6.

Configuration to block the exploit in Web Proxy and `CGenericXSLT` channels

The uPortal Web Proxy and **`CGenericXSLT`** channel includes features for restricting the URLs a channel instance will proxy, as implemented in a previous security fix. uPortal deployments must configure all web proxy and **`CGenericXSLT`** channel publications with a maximally restrictive match prefix. For example, if a web proxy instance proxies a remote application at

`http://www.myschool.edu/apps/dining_hall_signup/startpage.html`

that links to other parallel paths like

`http://www.myschool.edu/apps/dining_hall_signup/pick_a_plan.html`

but not like

`http://www.myschool.edu/other_cool_stuff/dining_menus.html`

An appropriate URI match prefix would be:

`http://www.myschool.edu/apps/dining_hall_signup/`

This would prevent the channel from proxying content at

`http://www.bad.com/exploitive_javascript_bearing_page`

even if a user were to be convinced by the Adversary to click an external link illicitly instructing the channel to proxy that URL.

Patching:

Source Replacement

Replace **`HttpProxyServlet.java`** and **`ProxyWriter.java`** in your local uPortal source tree with the attached files. Rebuild and re-deploy your uPortal. Rebuilding your uPortal class files with this updated source is the recommended way to address this exploit in your local uPortal environment.

Binary Replacement

Replace **`HttpProxyServlet.class`** and **`ProxyWriter.class`** with the attached .class files intended for use in uPortal 2.5.x. When using this approach, you'll need to also fix the issue in your local build environment (e.g. by replacing the .java files as described previously) so that when you perform a new build, the exploit remains fixed.

Previous patch

If you applied a previous patch for this issue, you will first need to undo the steps you performed in applying that patch before applying these steps and this patch.