Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
PREMIER MINISTRE

**S . G . D . S . N**
*Agence nationale de la sécurité des systèmes d'information*
*CERT-FR*

Paris, le 02 mai 2014
N° CERTFR-2014-ACT-018

**Affaire suivie par :**
CERT-FR

# BULLETIN D'ACTUALITÉ

**Objet : Bulletin d'actualité CERTFR-2014-ACT-018**

## 1 - Conséquences d'un poste compromis sur le service de messagerie

La compromission d'un poste utilisateur par un logiciel malveillant peut avoir des conséquences indirectes sur les services offerts par le système d'information, notamment le service de messagerie.

En effet, lors d'un incident traité par le CERT-FR, une institution ne pouvait plus émettre de courrier électronique vers l'extérieur suite à la compromission locale d'un poste. Les courriels émis étaient parfois retournés à l'expéditeur avec un message d'erreur indiquant que celui-ci avait été classé comme indésirable par une plateforme de messagerie.

Un grand nombre de plateformes de messagerie est en effet abonné à des services de listes noires d'adresses IP et de domaines, détectés comme étant émetteurs de courriels indésirables (spam). Ces services peuvent être gratuits, comme pour la Composite Blocking List (CBL) ou The Spamhaus Project.

Les méthodes de détections de sources de courriels illégitimes sont variées :

- le signalements d'utilisateurs ;
- la détection de serveurs de messagerie mal configurés (relais de messagerie ouverts sur Internet) ;
- l'utilisation de Honeypot, ou «pots de miel» qui simule un relais de messagerie ouvert.

Ainsi, un réseau connecté à Internet et dans lequel un poste compromis envoie des spams, pourra faire l'objet d'un référencement en liste noire de ses adresses IP de serveur de messagerie à son insu. Un retour à la normale n'est possible qu'après avoir mis fin à la compromission. La plupart des listes noires ci-dessus retirent les adresses IP et domaines de leur base de données lorsqu'aucun courriel illégitime n'est détecté après quelques heures ou jours.

Pour faciliter la détection de ce type de compromission, le CERT-FR recommande:

- d'utiliser un ensemble restreint de relais de messagerie internes ;
- d'activer la journalisation de ces relais pour connaître et surveiller les volumes de messages envoyés ;
- d'autoriser seulement les flux utilisés par les protocoles de transfert de mail (SMTP, SMTPS) vers et depuis Internet uniquement depuis ces relais.

## 2 - Vulnérabilité critique dans Internet Explorer

Cette semaine, le CERT-FR a diffusé l'alerte CERTFR-2014-ALE-005 concernant une vulnérabilité majeure dans Microsoft Internet Explorer. Cette faille permet d'exécuter du code arbitraire à distance sur les systèmes vulnérables.

Des exploitations de cette vulnérabilité ont été constatées par Microsoft et par la société FireEye. Le code d'exploitation de cette faille, développé en « JavaScript », repose sur l'utilisation du composant « Vector Markup Language » implémenté par la bibliothèque `vgx.dll`.

Microsoft a publié un correctif de sécurité hors cycle (MS14-021) corrigeant cette vulnérabilité. Ce correctif sera également disponible pour Windows XP (dont le support est terminé).

Le CERT-FR recommande l'application du correctif de sécurité MS14-021 dès que possible.

Pour mémoire le CERTA a publié, dans le bulletin d'actualité CERTA-2012-ACT-038, un ensemble de bonnes pratiques à mettre en oeuvre pour la prévention des attaques 0day sur les navigateurs Internet.

**Documentation**

- Bulletin de sécurité Microsoft MS14-021 du 01 mai 2014 :
  https://technet.microsoft.com/en-US/library/ms14-021
- Bulletin d'alerte CERTFR-2014-ALE-005 Vulnérabilité dans Internet Explorer :
  http://cert.ssi.gouv.fr/site/CERTFR-2014-ALE-005/index.html
- Bulletin de sécurité Microsoft 2963983 du 26 avril 2014 :
  https://technet.microsoft.com/en-US/library/security/2963983
- Informations complémentaires de Microsoft du 26 avril 2014 :
  http://blogs.technet.com/b/srd/archive/2014/04/26/more-details-about-security-advisory-2963983-ie-0ay.aspx
- Référence CVE CVE-2014-1776 :
  http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-1776
- Bulletin d'actualité CERTA-2012-ACT-038 :
  http://www.cert.ssi.gouv.fr/site/CERTA-2012-ACT-038/

# 3 - Risques associés aux serveurs d'applications

Un serveur d'applications est une brique logicielle visant à fournir aux applications Web un ensemble de services standards, tels que :

- l'interface de communication entre les clients et le serveur ;
- la répartition des requêtes vers les différentes applications hébergées par le serveur ;
- la répartition de charge ;
- la tolérance aux pannes ;
- etc.

Cet article se focalise sur les serveurs d'applications Java. On peut citer notamment parmi les plus connus : Apache Tomcat, JBoss AS (renommé Wildfly en avril 2013), WebSphere, Weblogic ou encore JOnAS.

Les serveurs d'applications Java sont généralement fondés sur la norme JavaEE (Java Enterprise Edition), standard d'interopérabilité définissant un ensemble de fonctionnalités ainsi que les interfaces associées.

Certains serveurs d'applications Java implémentent intégralement la norme JavaEE, comme JBoss AS, alors que d'autres limitent leur implémentation à certains composants spécifiques. C'est notamment le cas du serveur Apache Tomcat, qui implémente uniquement les parties nécessaires aux Servlets et JSP (JavaServer Pages). Cette approche permet de réduire la surface d'exposition du serveur.

Par ailleurs, il est rare que les applications Web soient conçues intégralement par le développeur. Elles s'appuient en effet le plus souvent sur d'autres briques logicielles fournissant un niveau d'abstraction plus élevé (ex : service d'authentification, services d'accès aux bases de données, etc.). Cela permet de s'appuyer sur des briques logicielles pérennes et réputées robustes. Toutefois, l'intégration de briques tierces implique d'assurer leur maintien en conditions opérationnelles (mises à jour de sécurité notamment).

Il en résulte donc des possibilités d'exploitation pour un attaquant à trois niveaux distincts : le serveur d'applications lui-même, les briques logicielles tierces intégrées et le code spécifique à l'application Web. Bien que la majorité des vulnérabilités présentes sur les applications Web proviennent d'erreurs de développement ou de mauvaises utilisations de briques logicielles tierces, les serveurs d'applications ne sont pas exempts de vulnérabilités. En effet, depuis 2010, 48 vulnérabilités pour le serveur Apache Tomcat et 45 vulnérabilités pour JBoss (JBoss AS et Wildfly) ont été rendues publiques. Ces serveurs étant très largement utilisés, la publication d'une vulnérabilité peut très vite prendre une ampleur considérable (attaques fiables, portables et utilisables à plus grande échelle).

Une configuration réfléchie du serveur d'applications est en conséquence importante puisqu'elle rend plus difficile l'exploitation de vulnérabilités et permet de diminuer la surface d'attaque potentielle. Les points de configuration les plus critiques sont abordés dans cet article et illustrés en se référant aux deux serveurs d'applications Java les plus répandus (Apache Tomcat et JBoss AS). Les principes évoqués sont transposables à l'ensemble des serveurs d'applications.

## Recommandations de configuration

### Console d'administration

Les serveurs d'applications offrent généralement des interfaces d'administration graphiques permettant de faciliter la tâche de l'opérateur. Toutefois, ces interfaces offrent également à l'attaquant une surface d'exploitation potentielle importante (vulnérabilités Web type XSS par exemple). Une alternative à ces consoles d'administration consiste à gérer le serveur de façon maîtrisée via ses fichiers de configuration. Ce mode de fonctionnement est idéal et recommandé. Si cela n'est toutefois pas possible, il faut a minima définir une authentification par mot de passe complexe (modification du mot de passe par défaut trivialement exploitable par un attaquant pour prendre le contrôle du serveur à distance) et si possible déployer les connecteurs d'administration sur un réseau séparé.

#### Apache Tomcat

*Désactivation de la console d'administration*

La désactivation de la console d'administration dans Apache Tomcat se fait en supprimant le répertoire `manager` du dossier de publication Apache Tomcat (sous-dossier `webapps` du dossier `CATALINA_HOME`) ainsi que les fichiers `host-manager.xml` et `manager.xml` présents dans le dossier `CATALINA_HOME/conf/Catalina/localhost`.
De la même manière, il est conseillé de supprimer les autres applications fournies par défaut présentes dans le dossier `webapps` mais non utilisées.

*Définition d'un mot de passe*

La définition des comptes utilisateur se fait dans le fichier `tomcat-users.xml` présent dans le dossier de configuration (sous-dossier `conf` du dossier `CATALINA_HOME`), sous la forme suivante pour un administrateur :

```
<tomcat-users>
<role rolename="manager-gui" />
<user username="<utilisateur>" password="<mot de passe>" roles="manager-gui">
</tomcat-users>
```

*Restrictions d'accès*

Il est possible de restreindre les accès à la console d'administration en spécifiant les adresses IP autorisées à accéder à la console d'administration dans le fichier `CATALINA_HOME/conf/Catalina/localhost/manager.xml` avec le contenu ci-dessous :

```
<Context privileged="true" >
<Valve className="org.apache.catalina.valves.RemoteAddrValve"
       allow="<adresses IP autorisées>" />
</Context>
```

#### JBoss

*Désactivation de la console d'administration*

La désactivation de la console d'administration dans JBoss s'effectue en supprimant le répertoire `management` du dossier de publication JBoss (sous-dossier `deploy` du correspondant au profil du serveur).

*Définition d'un mot de passe*

La définition des utilisateurs se fait via le script `add-user.sh` présent dans le dossier `bin`.

*Restrictions d'accès*

Il est possible de restreindre les accès à la console d'administration en spécifiant les adresses IP autorisées à accéder à la console d'administration dans le fichier `context.xml` se trouvant dans le dossier `WEB-INF` du fichier `.war` contenant l'application, sous la forme suivante :

```
<Context privileged="true" >
<Valve className="org.apache.catalina.valves.RemoteAddrValve"
       allow="<adresses ip autorisées>" />
</Context>
```

#### Protocoles d'administration

Les serveurs d'applications proposent fréquemment un protocole d'administration à distance, comme la technologie JMX (Java Management Extensions) dans le cas des serveurs JavaEE.

Il est recommandé de ne pas activer l'administration à distance si elle n'est pas nécessaire.

De manière générale, il est important de vérifier l'utilité de tous les services activés sur le serveur d'applications, certains n'étant pas forcément indispensables pour le fonctionnement de l'application, et ne faisant qu'ouvrir des portes supplémentaires pour un attaquant (par exemple Java RMI). Ces services non nécessaires doivent être désactivés, ou a minima, filtrés à l'aide d'un pare-feu.

##### Apache Tomcat

Par défaut, l'interface de contrôle à distance JMX de Apache Tomcat n'est pas active tant que le serveur d'applications n'est pas exécuté avec le paramètre `-Dcom.sun.management.jmxremote`.

Cependant, ce paramètre peut être présent dans le script de lancement d'Apache Tomcat ou dans la variable d'environnement `CATALINA_OPTS`. Il est donc important de s'assurer que ce n'est pas le cas pour que la désactivation soit effective.

Si l'utilisation du protocole JMX est nécessaire, il est impératif de mettre en place une authentification avec un mot de passe complexe et de restreindre les autorisations. Ces opérations se font respectivement en ajoutant les arguments suivants :

- `-Dcom.sun.management.jmxremote.authenticate=true`
- `-Dcom.sun.management.jmxremote.jmxremote.password.file=<fichier de mot de passe>`
- `-Dcom.sun.management.jmxremote.access.file=<fichier de contrôle>`

##### JBoss

Par défaut, l'interface de contrôle à distance JMX Jboss n'est pas active. Lors de son activation, l'interface JMX nécessite une authentification.

#### Authentification avec des connecteurs HTTP

Lors de l'utilisation d'applications protégées par mot de passe, les serveurs d'applications peuvent utiliser une méthode d'authentification HTTP. Dans un tel cas, il recommandé d'utiliser la méthode d'authentification `Digest` standard (et non la méthode `Basic`, qui n'offre aucune protection cryptographique du mot de passe).

##### Apache Tomcat

Par défaut, Apache Tomcat utilise la méthode d'authentification `Basic`. Afin de la remplacer par la méthode `Digest`, il faut modifier le fichier de configuration `CATALINA_HOME/server/webapps/<application>/WEB-INF/web.xml` afin qu'il contienne :

```
<login-config>
<auth-method>DIGEST</auth-method>
<realm-name>Application</realm-name>
</login-config>
```

Par exemple, pour l'interface d'administration, le paramètre `real-name` devra avoir la valeur `Tomcat Manager Application`.

##### JBoss

Depuis la version 7.0 de JBoss, la méthode d'authentification `Digest` est utilisée par défaut. Il est possible de vérifier que la méthode d'authentification n'a pas été changée en s'assurant que la partie `management` du fichier de configuration du serveur WildFly ne contient pas le paramètre de configuration `mechanism="PLAIN"`.

## Maintenance des serveurs d'applications

l est important d'inclure les serveurs d'applications dans la politique de mise à jour puisqu'ils constituent une cible pour les attaquants. Pour information, les versions actuelles de Apache Tomcat et JBoss sont :

- Apache Tomcat : 7.0.53
- Wildfly : 8.0 Final

### Rappel des avis et alertes émis concernant Apache Tomcat et JBoss

Depuis Mars 2012, les serveurs d'applications Apache Tomcat et JBoss ont fait l'objet des avis suivants :

- Bulletin d'avis du CERT-FR CERTFR-2014-AVI-198 pour Apache Tomcat : http://www.cert.ssi.gouv.fr/site/CERTFR-2014-AVI-198/index.html
- Bulletin d'avis du CERT-FR CERTA-2014-AVI-032 pour Apache Tomcat : http://www.cert.ssi.gouv.fr/site/CERTA-2014-AVI-032/index.html
- Bulletin d'avis du CERT-FR CERTA-2013-AVI-334 pour Apache Tomcat : http://www.cert.ssi.gouv.fr/site/CERTA-2013-AVI-334/index.html
- Bulletin d'avis du CERT-FR CERTA-2013-AVI-198 pour JBoss : http://www.cert.ssi.gouv.fr/site/CERTA-2013-AVI-096/index.html
- Bulletin d'avis du CERT-FR CERTA-2012-AVI-160 pour JBoss : http://www.cert.ssi.gouv.fr/site/CERTA-2012-AVI-160/index.html

## 4 - Rappel des avis émis

Dans la période du 25 avril au 01 mai 2014, le CERT-FR a émis les publications suivantes :

- CERTFR-2014-ALE-004-001 : Vulnérabilité dans Apache Struts
- CERTFR-2014-ALE-005-001 : Vulnérabilité dans Microsoft Internet Explorer
- CERTFR-2014-AVI-206 : Multiples vulnérabilités dans le noyau Linux d'Ubuntu
- CERTFR-2014-AVI-207 : Vulnérabilité dans Adobe Flash Player
- CERTFR-2014-AVI-208 : Multiples vulnérabilités dans Apache Struts
- CERTFR-2014-AVI-209 : Multiples vulnérabilités dans les produits Mozilla

## Gestion détaillée du document

**02 mai 2014** version initiale.