API Server and Analytics non root user

Motivation/security

In typical deployments, the API Server runs as "root" on a Linux/Unix system. The only real need for this is so it can listen on Internet domain privileged ports (port numbers less than 1024). This is a potential security issue however, as the root user has read/write access to all files on the system. Therefore, if the API Server ever became compromised and could be convinced to return the contents of, or overwrite sensitive files, to the OS would not prevent it. A solution to this is to run the API Server as a non root user, but still allow the API Server process to bind to privileged ports.

Linux Capabilities

Traditional UNIX implementations distinguish two categories of processes:

- *privileged* processes whose effective user ID is 0 (superuser or root)
- *unprivileged* processes whose effective user ID is non zero

Privileged processes bypass all kernel permission checks where unprivileged processes are subject to full permissions checking based on the processes credentials, usually from effective user ID and effective group ID.

More recent versions of the kernel divide up the privileges traditionally associated with the superuser into a set of *capabilities* which can be independently enabled or disabled. This allows a more fine grained control of permissions for a process.

The capability of use for the API Server is CAP_NET_BIND which specifically allows bind to privileged ports. The method by which this capability is set on the API Server is supported from kernel 2.6.24 onwards. If the kernel version is before 2.6.33 then it must be configured with the CONFIG_SECURITY_FILE_CAPABILITIES option enabled.

System Changes

Following are the steps which must be taken to run the API Server as an unprivileged user. For the purposes of the document it is assumed that the new unprivileged user is ***admin*** and the location of your API Server installation is ***/home/admin/gateway***.

Additional API Server Installations

Note that for any additional API Server installations to work, the additional library locations of the new installation must be added and and jvm.xml changes must also be carried out.

Analytics Installations

The instructions are the same for an Analytics installation, just make note of the path changes to your Analytics installation.

Modifications to file ownership

The unprivileged user must have ownership of the API Server files.
If there is a pre-existing API Server installation then ownership of the API Server files must be changed to the new user you intend running the API Server with.

The command to change user and group ownership of all files under the installation directory is:

```
# chown -R user:usergroup /path/to/installation
```

eg

```
# chown -R admin:admin /home/admin/gateway
```

SSL Accelerators or HSM access

The unprivileged user must have access to the device corresponding to the crypto accelerator or HSM card. For the Cavium this is /dev/pkp_nle_drv and for Utimaco /dev/cs2a

In the case of Cavium with an admin user /dev/pkp_nle_drv should have the following permissions:

```
crw-rw-r-- 1 root admin 126, 0 /dev/pkp_nle_dev
```

If the unprivileged user is different to admin then run the following command:

```
# sed -i /admin/d /lib/udev/load_{cavium,utimaco}
```

Setting the CAP_NET_BIND capability on vshell

The Linux capability for to allow the API Server to listen on ports < 1024 must be added to the *vshell* file. To allow the API Server process to verify the license file correctly as a non root user the CAP_SYS_RAWIO capability must also be added.

The command to execute is:

```
# setcap 'cap_net_bind_service=+ep cap_sys_rawio=+ep'
<GATEWAY_INSTALL_DIR>/platform/bin/vshell
```

eg

```
# setcap 'cap_net_bind_service=+ep cap_sys_rawio=+ep'
/home/admin/gateway/platform/bin/vshell
```

(this command should all be on one line).

To verify that the permissions have been set run:

```
# getcap /home/admin/gateway/platform/bin/vshell
```

The output of this command should be:

```
/home/admin/gateway/platform/bin/vshell = cap_net_bind_service,cap_sys_rawio+ep
```

Clearing capability

Be aware that any modifications to the vshell file with chown, chmod, etc will delete any set capabilities. So for example, if the user was to run
chown -R admin:admin on /home/admin/gateway they would then have to rerun the setcap command.

Note for earlier versions of the API Server:
For versions of the API Server prior to 7.1.0 the CAP_SYS_RAWIO capability is not required.

For versions of the API Server prior to 6.3.0 (such as 6.1.2) the path above should be `platform/`**`libexec`**`/vshell` as opposed to `platform/`**`bin`**`/vshell`

Depending on your Linux distribution this may involve installing an additional software package. If the “setcap” command cannot be found then try installing the libcap2 package.

For yum based systems (Redhat Enterprise Linux, CentOS, Oracle Linux):

```
# yum install libcap2
```

For Debian, Ubuntu

```
# apt-get install libcap2-bin
```

Add API Server Library Locations

When executing a file with elevated permissions certain environment variables are disabled as a security precaution. For this reason the locations of the API Server library files have to be made known to the OS. This can be carried out with a few simple steps. Note that if there is **more than one** API Server installation, or a separate Analytics installation, then these steps *must* be carried out for **each installation path**.

1. Create an ld.so.conf file with the API Server shared object locations
2. Execute ldconfig to reload the ld.so.cache file

For a 64 bit installation create a file */etc/ld.so.conf.d/gateway-libs.conf* which contains the following lines:

```
/home/admin/gateway/platform/jre/lib/amd64/server
/home/admin/gateway/platform/jre/lib/amd64
/home/admin/gateway/platform/lib/engines
/home/admin/gateway/platform/lib
/home/admin/gateway/ext/lib
```

For a 32 bit installation create a file */etc/ld.so.conf.d/gateway-libs.conf* which contains the following lines:

```
/home/admin/gateway/platform/jre/lib/i386/server
/home/admin/gateway/platform/jre/lib/i386
/home/admin/gateway/platform/lib/engines
/home/admin/gateway/platform/lib
/home/admin/gateway/ext/lib
```

Note that “/home/admin/gateway” should be replaced with the root of the API Server installation.

After creating the */etc/ld.so.conf.d/gateway-libs.conf* file run the following command to reload the library cache file:

```
# ldconfig
```

Modifying init.d script to use sudo

If there is an init.d script which runs the API Server as a service, then this script must be changed to execute as the unprivileged user. The easiest way is by using `sudo -u admin` in the start command of the script.
For 6.1.2 this would mean changing the following line in the /etc/init.d/vordelgateway script

```
daemon --pidfile $PIDFILE $DAEMON $DAEMON_ARGS >> $LOGFILE 2>&1
```

to:

```
daemon --pidfile $PIDFILE --user=admin $DAEMON $DAEMON_ARGS >> $LOGFILE 2>&1
```

Modify jvm.xml

Edit *system/conf/jvm.xml* under your API Server or Analytics installation. Note that if there is **more than one** API Server installation, or a separate Analytics installation, then these steps *must* be carried out for **each installation path**.

After this line near the top of the file
```
<JVMSettings classloader="com.vordel.boot.ServiceClassLoader">
```

Add an extra line.

For a 64 bit API Server installation add:
```
<VMArg name="-
Djava.library.path=$VDISTDIR/$DISTRIBUTION/jre/lib/amd64/server:$VDISTDIR/$DISTRIBUTIO
N/jre/lib/amd64:$VDISTDIR/$DISTRIBUTION/lib/engines:$VDISTDIR/ext/$DISTRIBUTION/lib:$V
DISTDIR/ext/lib:$VDISTDIR/$DISTRIBUTION/jre/lib:system/lib:$VDISTDIR/$DISTRIBUTION/lib
"/>
```

For a 32 bit API Server installation add:
```
<VMArg name="-
Djava.library.path=$VDISTDIR/$DISTRIBUTION/jre/lib/i386/server:$VDISTDIR/$DISTRIBUTION
/jre/lib/i386:$VDISTDIR/$DISTRIBUTION/lib/engines:$VDISTDIR/ext/$DISTRIBUTION/lib:$VDI
STDIR/ext/lib:$VDISTDIR/$DISTRIBUTION/jre/lib:system/lib:$VDISTDIR/$DISTRIBUTION/lib"/
>
```

Restart the API Server

Start the API Server with the unprivileged user after making the changes described in this document.

Note for Solaris 10

A similar system is in place for Solaris 10 by adding the “net_privaddr” privilege to a user. Note that in this case the user itself is modified (as opposed to the API Server process above).

In this case you can modify an existing user (for example *gwadmin*)

```
# usermod -K defaultpriv=basic,net_privaddr gwadmin
```