Session Sharing

Swivel Session Sharing

White Paper

Contents

Introduction

From version 3.9.5 onwards, Session sharing is included as part of Appliance Synchronisation. See Also Single Channel Session Cache.

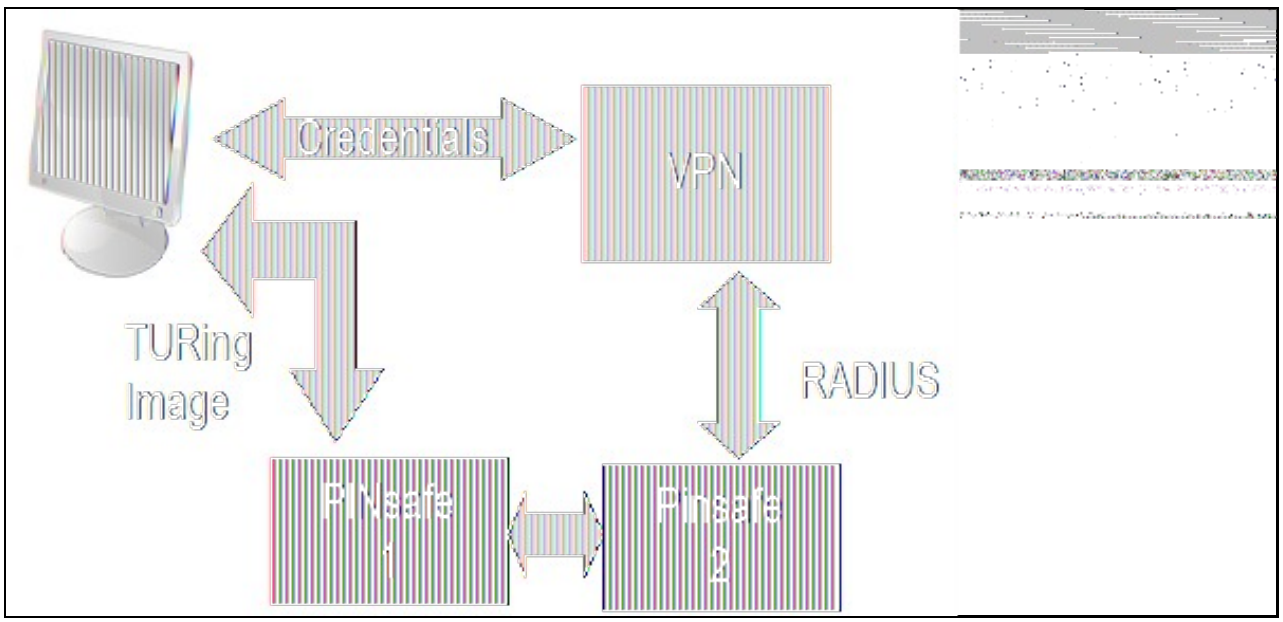
This white-paper explains the session sharing feature that was added in Version 3.5 of Swivel. It explains what this feature is, why you may wish to and how to implement and enable it on Swivel instances (it is not enabled by default).

Note also that PINsafe RADIUS Proxy may also be used in PINsafe 3.8 onwards as an alternative to Session Sharing.

Session Sharing Explained

For Active-Active HA pairs, when using single channel or on-demand dual channel, it has been a requirement that the user authenticates to the same server from which they retrieved the security string. This has prevented Swivel servers from being deployed in a truly load-balanced way. Active/Passive installations would not use session sharing.

Version 3.5 of the Swivel software now removes this requirement. Now when a session is started on one Swivel server, that session can be ?shared? with the other Swivel server, so that either server can accept the subsequent authentication request.



Session sharing means if Swivel server 1 supplied the TURing image the VPN can still authenticate the user via the Swivel 2 server

The mechanism for this is that the sessions that are ?in progress?, ie where a user has requested an image or message, but not yet authenticated, are stored in a cache that is shared between the two servers. In a similar way to the way that the databases are shared in a Swivel active-active configuration.

Swivel servers configured to share sessions must also share a common user-population.

The session sharing can, but need not be, used with an Active-Active pair. Clearly to support the use of this feature there must be sufficient bandwidth, with low-enough latency, between the two servers.

At the network level, session sharing uses IP Multicast, therefore it is a requirement that the network infrastructure supports multicast.

How it works

When a user uses TURing images or on-demand SMS messages to authenticate, the associated security string is only valid for a short time. When a security string is requested in either of these two ways, a session is created within Swivel. This records, in memory, all the information required to allow the user to authenticate using that security string. If the user does not authenticate in time, and the security string becomes invalid, the session is invalidated and destroyed.

Similarly if the user requests another security string, a new session is created and the previous session is deleted.

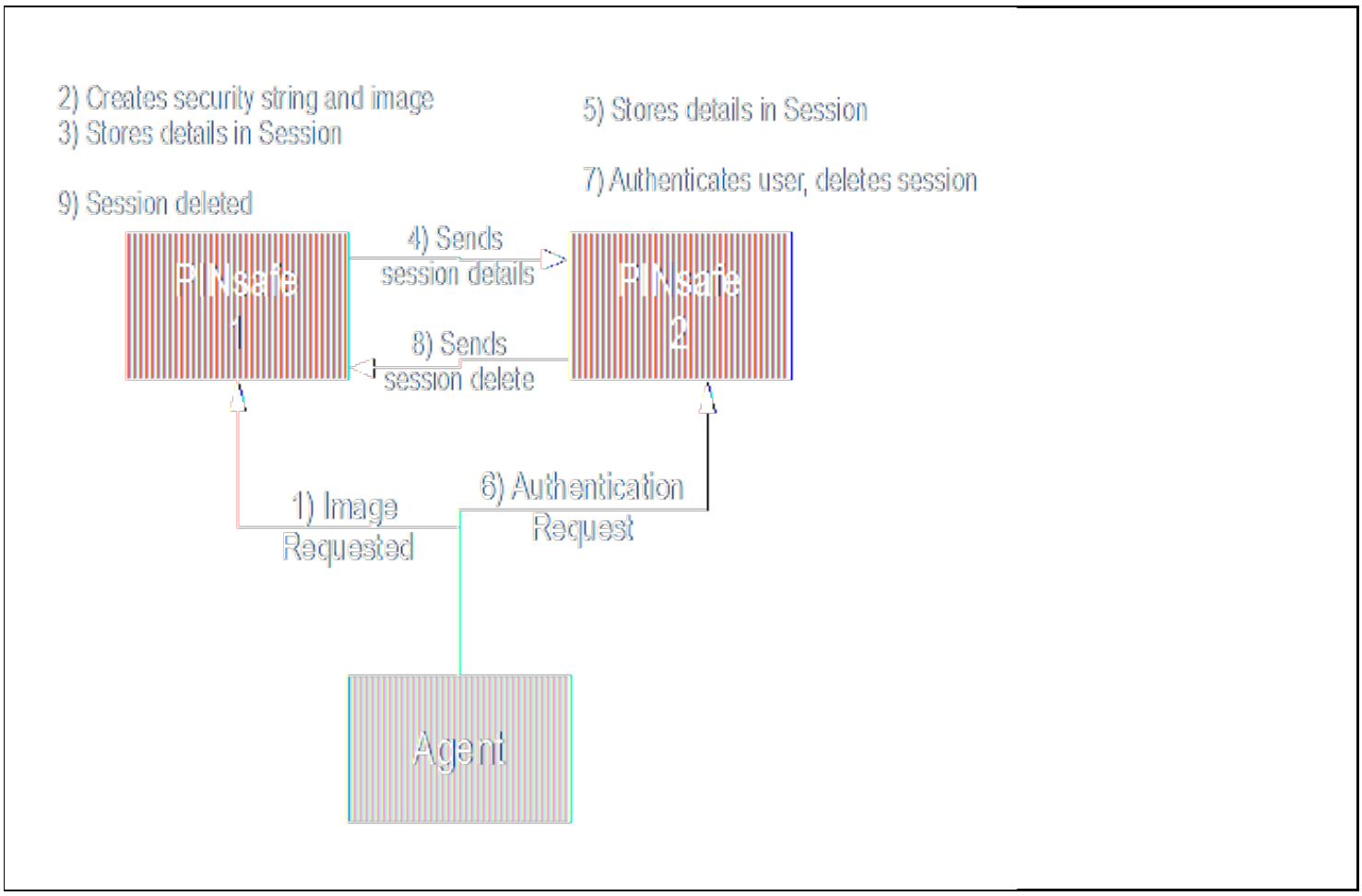
These sessions, as they are short-lived, are stored in memory rather than in a database (unlike the dual-channel security string messages sent out automatically after every authentication attempt, which are stored in the database and therefore replicated across both Swivel servers in an active-active pair).

When a pair of Swivel servers have been set up to share their sessions every time a change is made to the session cache on one server a message is sent across to the other Swivel server informing it of the change. This means that, in effect, both servers share the same session-cache.

This common view of the current active sessions means that a user can authenticate to any server.

Example.

An authentication agent (e.g. a VPN) may be configured to use either one of a pair of Swivel servers



Session sharing example

1. The agent submits a request for a TURing image to the first Swivel server.
2. The first Swivel server receives this request, creates the security string and the required TURing image, which is returns to the Agent.
3. It then stores the details in a session that it stores in memory.
4. In addition, as it is configured for session sharing it sends the details of the session it has created, via multicast, to any other Swivel servers that may need this information. **NB** The use of multicast means that if, for example, the second Swivel server became unavailable for some reason the first Swivel server does not waste time trying to send messages to it. It is possible to set the session sharing to use unicast, but this is not advisable.
5. The second Swivel server, as it also configured for session sharing, receives this information and also creates and stores the session.
6. The authentication request them is sent to the second Swivel server.
7. As this second server has details of the session that was created for the user, it can authenticate the user. Once the user has authenticated the session is no- longer valid, so it is deleted.
8. The second Swivel server therefore sends a message, again via multi-cast, informing any Swivel servers that the session is no longer valid.

Session Sharing Implementation

For Swivel Appliances see Enabling Appliance Session Sharing

For software installations and further assistance see the below.

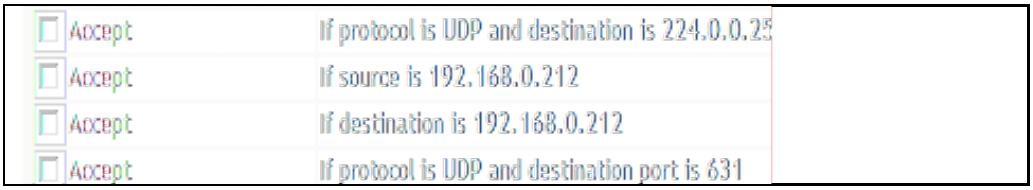
Network Configuration

The Swivel servers need to be connected to a network that supports multicast traffic. Most switches are capable of supporting multicast although some switches / firewalls may be configured to block or disallow multicast.

Firewalls Settings

Note: When changing the firewall settings verify the following Known Issues before changing the firewall rules.

Any firewalls between the two Swivel servers needs to be opened to allow the messages to be passed between the servers. This includes the Linux firewalls on the appliances themselves. A rule must be added to the chain to accept traffic from the other appliance and to allow traffic to be sent to the other appliance.



Additional Firewall rules,

assuming other server is on 192.168.0.212

The above rule means all traffic from the other server will be allowed. A better rule would just open up the multicast port being used. This is 4446 by default and is defined within the cache.xml file multicastGroupPort=4446

Primary Appliance Firewall Configuration

On the Primary Swivel appliance edit /etc/sysconfig/iptables with WinSCP:

After the line "-A RH-Firewall-1-INPUT -i lo -j ACCEPT" add the line:

```
-A RH-Firewall-1-INPUT -i eth0 -s 192.168.0.37 -j ACCEPT
```

(where 192.168.0.37 is the IP of the Standby machine. Replace this example IP address with your Standby IP)

Standby Appliance Configuration

On the Standby Swivel appliance edit /etc/sysconfig/iptables with WinSCP:

After the line "-A RH-Firewall-1-INPUT -i lo -j ACCEPT" add the line:

```
-A RH-Firewall-1-INPUT -i eth0 -s 192.168.0.36 -j ACCEPT
```

(where 192.168.0.36 is the IP of the Primary machine. Replace this example IP address with your Primary IP)

Appliance Firewall Restart

Once you've made these changes, on BOTH machines restart ipchains:

```
service iptables restart
```

Appliance connectivity and listener

You should find that you return a result when running the following netstat command:

```
[admin@primary ~]# netstat -an | grep 4446
udp     1520      0 0.0.0.0:4446                0.0.0.0:*

udp        0      0 0.0.0.0:4446                0.0.0.0:*
```

Time Synchronisation

Sessions are timestamped by the server that creates them, sessions are only valid for finite time, so the clocks on the servers must be synchronised. As clocks drift, the use of an NTP server is essential. NTP servers can be configured from the Appliance menus.

Login to the command line to check and start the NTP service if necessary:

```
[admin@primary ~]# service ntpd status
ntpd is stopped
[admin@primary ~]# service ntpd start
Starting ntpd:                                             [  OK  ]
[admin@primary ~]#
```

See the NTP servers article for further information.

Swivel Settings

There are two elements to the Swivel configuration. First you need to configure Swivel to use session sharing, then select multicast session sharing.

To select session sharing you need to edit the config.properties file, the default settings for this file for Swivel 3.6 onwards are:

```
SECURITY_STRING_GENERATOR = com.swiveltechnologies.pinsafe.server.utility.SecurityString
SESSION_MANAGER = com.swiveltechnologies.pinsafe.server.session.LocalSessionManager
```

The file can can be found under one of the following locations:

- Swivel 3.9.2 onwards on appliance: /home/swivel/.swivel/conf
- Swivel 3.9.2 onwards on software: USER_HOME/.swivel/conf. Example Windows 7 c:/users/<username>/.swivel
- Swivel 3.5 to 3.9.1 or earlier: /usr/local/tomcat/webapps/pinsafe/WEB-INF/conf
- Swivel 3.5 to 3.9.1 or earlier on software: <path to Tomcat>/webapps/pinsafe/WEB-INF/conf

Note many of the transitory data settings were moved in 3.9.1, but the config.properties was moved in version 3.9.2

Take a backup up copy of config.properties

This file includes a setting for Session Manager, the default value for Version 3.5 is;

com.swiveltechnologies.pinsafe.session.LocalSessionManager and needs to be changed to com.swiveltechnologies.pinsafe.session.DistributedCacheSessionManager as shown below:

```
SECURITY_STRING_GENERATOR = com.swiveltechnologies.pinsafe.utility.SecurityString
SESSION_MANAGER = com.swiveltechnologies.pinsafe.session.DistributedCacheSessionManager
```

for Version 3.6 and later the SESSION_MANAGER line needs to be changed from com.swiveltechnologies.pinsafe.server.session.LocalSessionManager to com.swiveltechnologies.pinsafe.server.session.DistributedCacheSessionManager

```
SECURITY_STRING_GENERATOR = com.swiveltechnologies.pinsafe.server.utility.SecurityString
SESSION_MANAGER = com.swiveltechnologies.pinsafe.server.session.DistributedCacheSessionManager
```

To specify multicast session sharing navigate to the pinsafe/WEB-INF/classes folder. By default there will be two files related to session caching; multicast-cache.xml and peer-cache.xml.

To specify the use of multicast you need to copy the multicast-cache.xml file to cache.xml

check file permissions and ownership

Tomcat must then be restarted for these changes to take effect

Repeat this for all Swivel servers.

Once these process have completed on all servers, they should all be sharing a common session-cache.

Testing

With session sharing enabled you can request a security string from one server and authenticate to another (assuming the user exists on both servers). Therefore the easiest way to test is via the Swivel Administration Console. Enter the username in the first login page, start the session, then enter the same username and the resultant one-time code on the second Swivel server. This should succeed. Entering the same one-time code in the first Swivel

server should then fail, as the session should have been deleted from both servers.

Considerations

- The protocols use for the multicast session-sharing are not encrypted. Whereas just sniffing the network will not tell an attacker any useful information, an attacker with the time, expertise and access to the network would be able determine the details of the sessions passed between the servers. However this would only reveal the details of the security strings, this attack would have be combined with an attack that captured the submitted credentials.
- You may need to enable PIM (Protocol Independent Multicast) on your switch or VLAN for Multicast to work.

Swivel Multicast is configured using the RFC implementation with Multicast and IGMP joins for a specific group or groups.

Troubleshooting

ERROR 127.0.0.1:Exception on replication of putNotification. RemoteException occurred in server thread; nested exception is: java.rmi.UnmarshalException: error unmarshalling arguments; nested exception is: java.io.InvalidClassException: net.sf.ehcache.Element; local class incompatible: stream classdesc serialVersionUID = 1098572221246444544, local class serialVersionUID = 3343087714201120157. Continuing...

127.0.0.1:Session started for user: xyz.

This error has been seen in Swivel version 3.9.3 due to control codes in the config.properties file, care must be taken when editing such files such as using Windows third party tools.

java.lang.ClassNotFoundException:

com.swiveltechnologies.pinsafe.server.session.DistributedCacheSessionManager

```
org.apache.catalina.core.StandardContext loadOnStartup
SEVERE: Servlet /pinsafe threw load() exception
java.lang.ClassNotFoundException:
com.swiveltechnologies.pinsafe.server.session.DistributedCacheSessionManager
     at org.apache.catalina.loader.WebappClassLoader.loadClass(WebappClassLoader.java:1386)
     at org.apache.catalina.loader.WebappClassLoader.loadClass(WebappClassLoader.java:1232)
     at java.lang.Class.forName0(Native Method)
     at java.lang.Class.forName(Unknown Source)
     at com.swiveltechnologies.pinsafe.server.config.Config.load(Config.java:87)
     at com.swiveltechnologies.pinsafe.server.config.Startup.init(Startup.java:126)
     at javax.servlet.GenericServlet.init(GenericServlet.java:212)
     at org.apache.catalina.core.StandardWrapper.loadServlet(StandardWrapper.java:1139)
     at org.apache.catalina.core.StandardWrapper.load(StandardWrapper.java:966)
     at org.apache.catalina.core.StandardContext.loadOnStartup(StandardContext.java:3996)
     at org.apache.catalina.core.StandardContext.start(StandardContext.java:4266)
     at org.apache.catalina.core.ContainerBase.addChildInternal(ContainerBase.java:760)
     at org.apache.catalina.core.ContainerBase.addChild(ContainerBase.java:740)
     at org.apache.catalina.core.StandardHost.addChild(StandardHost.java:544)
     at org.apache.catalina.startup.HostConfig.deployDescriptor(HostConfig.java:626)
     at org.apache.catalina.startup.HostConfig.deployDescriptors(HostConfig.java:553)
     at org.apache.catalina.startup.HostConfig.deployApps(HostConfig.java:488)
     at org.apache.catalina.startup.HostConfig.start(HostConfig.java:1150)
     at org.apache.catalina.startup.HostConfig.lifecycleEvent(HostConfig.java:311)
     at org.apache.catalina.util.LifecycleSupport.fireLifecycleEvent(LifecycleSupport.java:120)
     at org.apache.catalina.core.ContainerBase.start(ContainerBase.java:1022)
     at org.apache.catalina.core.StandardHost.start(StandardHost.java:736)
     at org.apache.catalina.core.ContainerBase.start(ContainerBase.java:1014)
     at org.apache.catalina.core.StandardEngine.start(StandardEngine.java:443)
     at org.apache.catalina.core.StandardService.start(StandardService.java:448)
     at org.apache.catalina.core.StandardServer.start(StandardServer.java:700)
     at org.apache.catalina.startup.Catalina.start(Catalina.java:552)
     at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
     at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)
     at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)
     at java.lang.reflect.Method.invoke(Unknown Source)
     at org.apache.catalina.startup.Bootstrap.start(Bootstrap.java:295)
     at org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:433)
```

This error has been seen in Swivel 3.8 due to control codes in the config.properties file, care must be taken when editing such files such as using Windows third party tools.