HITACHI
Inspire the Next

Configuring

F5 LTM Load Balancer

for *Hitachi ID Bravura Security Fabric*

Software revision: 12.2.4
Document revision: 30072
Last changed: 2022-03-01

Contents

1 Introduction

Hitachi ID Bravura Privilege enables users to "check out" temporary access to privileged accounts, granting access to the account via disclosure plugins. When a user accesses these privileged accounts, their activities are recorded by the PAM session monitor (sessmon), which records a variety of account activity that can be retrieved later for auditing purposes.

When checking out an account, up to three separate connections are made to the *Bravura Privilege* server:

- The user's current connection to *Bravura Privilege* through their internet browser, from which the account was checked out.
- When using Firefox or Google Chrome, the native extension that prepares and launches the disclosure plugin. In Internet Explorer, an ActiveX control handles this without requiring a new connection to *Bravura Privilege*.
- The disclosure plugin that grants account access, such as the Remote Desktop window launched when accessing the account.

These three parts of account disclosure are only able to communicate the *Bravura Privilege* server URL information between each other; however all three sessions must be in communication with the same *Bravura Privilege* server that started the request throughout their lifetimes. If this connection is disrupted at any point, functionality such as account session monitoring may fail to operate correctly.

In many load-balanced environments, incoming internet traffic is evenly distributed to available web servers. While this is functional for most environments, it can create issues when a web application creates additional web sessions, as checking out accounts in *Bravura Privilege* does. In this case, the load balancer may attempt to direct these new sessions to other web servers than the one that initiated the account check-out, breaking the functionality.

The various strategies for ensuring that a web session can remain connected to the same server throughout its lifetime are collectively referred to as "session persistence." To ensure session persistence, data from network traffic is used to identify which server is meant to handle each request. Session persistence must be configured by the load balancer; both because it is the only device with full knowledge of available clients and servers, and because the load balancer will override session persistence strategies built into client software. For more information on session persistence, see:

https://www.f5.com/services/resources/white-papers/cookies-sessions-and-persistence

2 Requirements

Before starting, ensure that the following prerequisites have been met:

- BIG-IP F5 Load balancer licensed for Local Traffic Management. (Version 13.1.1 in a single-NIC configuration was used for this testing.)
- *Hitachi ID Bravura Security Fabric* instances installed in replication.
- The load balancer, *Bravura Security Fabric* instances, and all client and target systems have connectivity to each other.
- The BIG-IP load balancer's management port is on a network or subnet that is separate from the *Bravura Security Fabric* instances or their client and target systems.
- The BIG-IP Setup utility has been run to configure initial settings
- The BIG-IP system has network interface, VLAN, server pool, and self IP configurations in place to direct traffic to and from *Bravura Security Fabric* nodes.

For more information on configuring the F5 LTM load balancer, see the product manual and knowledge base available at:

`https://support.f5.com/csp/knowledge-center/software/BIG-IP?module=BIG-IP%20LTM`.

3 Configuring the Load Balancer

Hitachi ID Bravura Security Fabric requires that the following behavior is configured on the network load balancer:

- *Bravura Security Fabric*'s **BASE_IDSYNCH_URL** and **SMON_HTTP_URL** system variables are set to the IP address of the load balancer.
- Network connections through the load balancer use cookie-based session persistence by default, and IP address-based session persistence as a fallback option.
- A traffic handling rule is prepared that enables persistence for API calls to the *Bravura Security Fabric* instance.

Configuring the load balancer to support *Bravura Security Fabric* is fairly straightforward, and involves the following steps:

1. Configure the *Bravura Security Fabric* system variables.

   The *Bravura Security Fabric* system variables **BASE_IDSYNCH_URL** and **SMON_HTTP_URL** allow product administrators to specify where to send requests for features that need to be handled by specific *Bravura Security Fabric* instance nodes. Both privileged account check-out and account access session recording are prominently affected by changes in the server handling their requests,

as in a load balanced environment, these requests have no knowledge of which server is handling their request.

By configuring the **BASE_IDSYNCH_URL** and **SMON_HTTP_URL** to point to the address of the load balancer, we ensure that these requests are always passed through the same traffic management policies used to initiate the request, ensuring that their session persistence is retained.

See Configure *Bravura Security Fabric* system variables for details on configuring these settings.

2. Configure the Cookie Persistence Profile.

   Modern traffic management systems offer a wide variety of policies to enable session persistence, and it is important to know which policy is right for your system. Cookie Persistence is the most popular policy for the majority of web applications, as cookie data is highly configurable, robust, and is typically unaffected by network policies that modify packet data. The main drawback to cookie persistence is that applications must be capable of sending the cookie with every request; a situation that is not always feasible.

   For the F5 LTM, Hitachi ID Systems recommends the **HTTP Cookie Insert** policy of cookie persistence. When using the **HTTP Cookie Insert** policy, the F5 LTM will insert a browser cookie into any traffic that does not already have that cookie defined; if the cookie exists in subsequent request or response headers, the F5 LTM uses the cookie data to associate requests to a server and maintain session persistence.

   By default, the F5 LTM cookie uses the following format:

   `BIGipServer<Pool name>=<Cookie data>`

   where <*Pool name*> is the server pool defined on the F5 LTM, and <*Cookie Data*> is the uniquely-identifying information for this session.

   See Configure the Cookie Persistence Profile for details on configuring the cookie persistence profile.

   See Types of Persistence for more information on available session persistence strategies for the F5 LTM.

3. Create an iRule to Enable Persistence for API Requests.

   While cookie persistence is a robust solution for the majority of web application use cases, it is not functional for applications that are unable to handle cookie traffic, but require a persistent session regardless. In *Bravura Security Fabric*, a prominent example of this scenario is requests made to the *Bravura Security Fabric* API. API calls can originate from a wide variety of client-scripted plugins and applications, and each of these would need to support HTTP cookie retransmission in order for cookie persistence to support these connections.

   As API operations require a persistent connection to their server in order to function properly, a traffic management rule is required to handle API traffic for *Bravura Security Fabric*; in the F5 LTM, these rules are known as “iRules.” iRules are simple scripted solutions that evaluate packet data in order to make various routing decisions and data modifications. To support the *Bravura Security Fabric* API, we require an iRule that performs the following operations for HTTP requests:

   - Confirm the incoming traffic is an API request.
   - Enable persistence for the request.
   - Assign the pool for the request
   - Change the "Host" header for the request, setting it to the address of the *Bravura Security Fabric* server that will receive this request.

The exact functionality of this script may vary depending on the needs of your business, but enabling an iRule to selectively handle API traffic is a minimally-disruptive solution for supporting calls to the *Bravura Security Fabric* API.

See Enable Session Persistence for API calls through iRules for details on configuring API traffic iRules.

4. Configure the *Bravura Security Fabric* Virtual Server for Session Persistence

   In the F5 LTM load balancer, the "Virtual Server" feature presents a reachable address that network traffic can be sent to in order to be load balanced. The F5 LTM's virtual servers are responsible for receiving traffic from both clients and servers, and is responsible for deciding where how that traffic is routed. Because of this, the F5 LTM virtual server responsible for *Bravura Security Fabric* traffic must be updated to enable session persistence, and to make use of the session persistence settings we defined in previous steps.

   See Configure Virtual Server to Enable Session Persistence for details on configuring the virtual server.

4 Detailed steps

4.1 Configure Bravura Security Fabric system variables

1. Identify the publicly-facing address of the load balancer that will handle *Hitachi ID Bravura Security Fabric* traffic; this should be the same address used to log into *Bravura Security Fabric* by your end users. Record this information for later.
2. Log on to Hitachi *Bravura Security Fabric* as a product administrator.
3. Navigate to **Manage the system** → **Maintenance** → **System variables**.
4. On this page, locate the entries for **BASE IDSYNCH URL**. Input the load balancer address recorded in Step 1 in the accompanying text field, in the format:

   `http://<Load balancer address>`.
5. On this page, locate the entries for **SMON HTTP URL**. Input the load balancer address recorded in Step 1 in the accompanying text field, in the format:

   `http://<Load balancer address>/<IDM Suite instance name>/smonc.exe`.
6. Click **Update** at the bottom of the page to commit your changes.

4.2 Configure the Cookie Persistence Profile

1. Log into the BIGIP F5 LTM as an administrator.
2. Navigate to **Main** → **Local Traffic** → **Profiles** → **Persistence**.
3. If you want to use the default cookie profile for managing *Hitachi ID Bravura Security Fabric* session persistence, select **cookie** from this list, and skip to Step 8 below.

   If you want to create a custom persistence profile for *Bravura Security Fabric* traffic, click **Create...** and proceed from Step 4.

4. In the **Name** field, input a unique name for this profile.
5. In the **Persistence Type** drop-down list, select **cookie**.

   The **Configuration** options table appears, with values disabled by default. The **Parent Profile** field also appears.
6. In the **Parent Profile** field, select **Cookie**.
7. Review the **Configuration** table, and enable fields that need to be modified.

   In order to modify these settings, you will need to specify which options will not be inherited from the parent profile. Click the checkboxes on the right to enable modification for individual fields you wish to update, or click the checkbox labeled **Custom** to enable all fields for editing.
8. Under **Cookie Method**, select **HTTP Cookie Insert**.
9. Under **Expiration**, enable **Session Cookie**.
10. Click **Finished** or **Update** to commit your changes.

4.3 Enable Session Persistence for API calls through iRules

1. Log into the BIG-IP administration UI as an administrator.
2. Navigate to **Main** → **Local Traffic** → **iRules**.
3. Click **Create...**
4. Assign a unique name for this iRule.
5. Configure the iRule to modify API traffic, by updating the following packet information:
   - The HTTP "Host" header must be set to the IP address of the server handling the request.
   - The packet must be assigned to a pool, if one was not already assigned.
   - A non-cookie persistence profile must be applied to the packet.

   The following iRule example is capable of updating SOAP API traffic in order to enable session persistence through Destination Address, or "sticky" persistence:

```
when HTTP_REQUEST {
if {[HTTP::header exists "SOAPAction"]}{
     # Set custom persistence profile, as API requests may not support cookies.
     persist sticky 255.255.0.0 3600
        # Confirm that members of “Pool1” are available, and set the pool.
        if { [active_members Pool1] > 1 } {
                    pool Pool1
     }
     # Replace the Host header, as this needs to match the destination node.
     HTTP::header replace Host [LB::server addr]

     # log local0. "SOAP request: [HTTP::header value SOAPAction], sending
  →to: [LB::server addr]."
}
```



```
}
when LB_FAILED {
log local0. "Failed pool selection"
}
```

6. Click **Finished** when your iRule is complete.

4.4 Configure Virtual Server to Enable Session Persistence

1. Log into the BIG-IP administration UI as an administrator.
2. Navigate to **Main** → **Local Traffic** → **Virtual Servers** → **<Hitachi ID Bravura Security Fabric *virtual server*>** → **Resources**.
3. Under **Default Persistence Profile**, use the dropdown list to select the cookie persistence profile you created/modified earlier.
4. Under **Fallback Persistence Profile**, use the dropdown list to select **source_addr**.
5. Click **Update**.
6. Under **iRules**, click **Manage...**
7. Under "Available", select the iRule you created earlier with left-click, and press the **<<** button to enable it for this virtual server.
8. Click **Finished** to confirm your changes.

Appendices

A Types of Persistence

In order to ensure that a load-balanced environment does not disrupt the normal operation of the *Hitachi ID Bravura Security Fabric*, the load balancer must have session persistence enabled for the services that need a predictable connection to a single node. A detailed outline of BIG-IP's supported persistence profiles is available in:

https://techdocs.f5.com/kb/en-us/products/big-ip_ltm/manuals/product/ltm-profiles-reference-13-1-0/4.html

A summary is provided here:

Cookie Persistence Cookie persistence uses the HTTP cookie header to ensure session persistence.

This method is more robust than other persistence options, but requires that client applications are prepared to handle these cookies. More details on cookie persistence in the F5 LTM environment are available in the manual at:

https://support.f5.com/csp/article/K83419154

Cookie Hash This method maps a specific cookie value to a specific node, allowing granular control of how traffic is routed. This requires that the web server creates the web cookie, and send it when new sessions are created.

HTTP Cookie Insert In this method, the BIG-IP injects an HTTP Cookie header into new sessions. Requests that include this header are directed to their respective nodes.

HTTP Cookie Passive In this method, BIG-IP does not interact with cookie data. Instead, the server creates the cookie, which includes the server information and timeout. This method is not recommended for most environments.

HTTP Cookie Rewrite This method intercepts the Set-Cookie header created by the web server, and overwrites its name and content to contain the address and port information needed for persistence.

Destination Address This method directs traffic to the same server based on the destination IP of the incoming packets.

Hash This method uses the data from request and response traffic to generate a hashed value that is used to associate sessions to a specific server.

Host Host persistence uses the HTTP Host header to determine which server to direct traffic to.

Microsoft Remote Desktop This method tracks sessions between clients and servers running the Microsoft RDP service to ensure persistence.

SIP SIP persistence is an application-specific protocol that tracks Session Initiation Protocol messages exchanged by applications who employ this protocol.

Source Address Referred to as "simple persistence", this method routes traffic based on the source IP of a packet.

SSL This method uses the SSL session ID to ensure persistence to a server.

WARNING!: If using load balancers, do not configure any SSL options for transparent synchronization traffic. SSL options should only be configured on load balancers for WebUI traffic, not transparent synchronization. Transparent synchronization is encrypted using a proprietary encryption algorithm. Contact support@Hitachi-ID.com for more details.

Universal This method uses data extracted from request and response packets to establish persistence, but requires that the BIG-IP is able to inspect the packet data in detail.