Jim Knatcher, Senior Client Technical Professional
04 Oct 2011

AVKS TAM Domain Failover Cookies (DFC)

Tivoli. software

Tivoli Technical Enablement
Driving software success through skills enablement

Introduction

- Abstract:
- Domain Failover Cookies for Tivoli Access Manager
- Objectives:
  - After attending the student will know how to:
    - Understand the concept of DFC
    - Understand various configuration parameters and their performance considerations

Concepts

- The failover cookie is not actually a mechanism for maintaining sessions; it is a mechanism for transparently re-authenticating the user. Failover authentication is most commonly used in a scenario where client requests are directed by a load balancing mechanism to two or more replicated WebSEAL servers.

- The replicated servers have identical configuration. They contain replica copies of the WebSEAL protected object space and junction database.

- The purpose of failover authentication is to prevent forced login when the WebSEAL server that has the original session with the client suddenly becomes unavailable. Failover authentication enables the client to connect to another WebSEAL server, and create an authentication session containing the same user session data and user credentials.

Creation

- The failover cookie contains encrypted client-specific data, such as:
  - User name
  - Cookie-creation time stamp
  - Original authentication method
  - Attribute list
- By default, the attribute list contains the user's current authentication level. WebSEAL can be configured to add additional extended attributes to the attribute list.
  - This is useful for non-sticky failover environments where you'll want to add the session ID.
- The cookie is placed on the browser when the client first connects. If the initial WebSEAL server becomes temporarily unavailable, the cookie is presented to the substitute server.
- The replicated WebSEAL servers share a common key that can decrypt the cookie information. When the substitute replica WebSEAL server receives this cookie, it decrypts the cookie, and uses the user name and authentication method to regenerate the client's credential. WebSEAL can also be configured to copy any extended attributes from the cookie to the user credential.

Failover

- The client (browser) attempts to access a protected resource. The client request goes to a load balancer that controls access to the replicated WebSEAL servers.
- The load balancer selects a target WebSEAL server and forwards the user request.
- The client successfully authenticates to WebSEAL using one of the supported authentication methods.
- WebSEAL creates a failover authentication cookie that contains client authentication information, and sends the cookie to the client browser.
- The browser sends the cookie through the load balancer to WebSEAL with each subsequent request. The WebSEAL server processes each request.
- If the load balancer finds that the original WebSEAL server is no longer available, the client request is directed to another replicated WebSEAL server.
- The replicated WebSEAL server is configured to check for the existence of a failover authentication cookie every time it attempts to authenticate a user.
- The replicated WebSEAL server uses the information in the cookie to establish a session with the client, without requiring the client to manually log in again. The client's session data and user credential are built, and the request for the protected resource is processed.
- The change of session from one WebSEAL server to another WebSEAL server is transparent to the client. Because the WebSEAL servers contain identical resources, the client session continues uninterrupted.

Parameters

Failover-include-session-id

Set this to “yes” in non-sticky load balancer configurations

[failover-add-attributes]

AUTHENTICATION_LEVEL = add

To specify authentication strength level (i.e. step up) in the failover authentication cookie, add the authentication level to the WebSEAL configuration file.

session-lifetime-timestamp = add

Adds this value to the failover authentication cookie. When a failover incident occurs, the session lifetime value is used to determine the time remaining for the life of the user's session (the session lifetime value is not reset at failover). If the session lifetime has expired, the user must login.

Parameters (cont'd.)

- [failover] failover-update-cookie = -1
  - When failover-update-cookie is set to 0, the last activity timestamp is updated with each request.
  - When failover-update-cookie is set to an integer less than 0 (any negative number), the last activity timestamp is never updated.
  - When failover-update-cookie is set to an integer greater than 0, the session activity timestamp in the cookie is updated at intervals of this number of seconds.

The third option can severely affect performance. Administrators must choose a balance between optimal performance and absolute accuracy of the timestamp in the cookie. To keep the timestamp most accurate, failover cookies should be updated every time the user makes a request. However, frequent updating of cookie contents incurs overhead and decreases performance.

When you set **failover-update-cookie** to a number greater than zero, ensure that you also set session-activity-timestamp = add. If you do not set session-activity-timestamp = add, WebSEAL will decode the failover cookie on each user access. This repetitive action could adversely affect performance.

Parameters (cont’d.)

```
[authentication-mechanisms]
#failover-password = failover_password_library_filename
#failover-token-card = failover_token_card_filename
#failover-certificate = failover_certificate_filename
#failover-http-request = failover_http_request_filename
#failover-cdsso = failover_cdsso_filename
#failover-kerberosv5 = failover_kerberos_library
#failover-ext-auth-interface = failover_kerberos_library
```

Uncomment the appropriate lines for the authentication mechanisms that require failover authentication and add the appropriate module name.

Tradeoffs

- Session Management Server
  - Centralized view of session cache
    - Who’s logged in
    - Session termination
    - Refresh credentials in a central location
    - Concurrent logins
  - Reduces load on user registry if you’re having many failover events
- Security Risks
  - Enabling domain-wide failover authentication introduces additional security risks to the WebSEAL deployment, because the failover cookie can be sent to any server that is in the same DNS domain as the WebSEAL server. If an attacker controls any Web server in the domain or can compromise the DNS server for the domain, they can hijack failover cookies and impersonate users.

Where to go for more information

- Failover authentication configuration task summary
  - http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=%2Fcom.ibm.itame.doc%2Fam611_webseal_admin401.htm

Questions?

not for distribution