Offline Support Testing Plan

Version 1.5

**From**

**SPAN Systems Corporation**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 24th Aug 2009 | 1.0 | Draft version | Shanmuga Srinivas |
| 31st Aug 2009 | 1.1 | Draft version | Shanmuga Srinivas |
| 8th Sep 2009 | 1.2 | Draft version | Shanmuga Srinivas |
| 16th Sep 2009 | 1.3 | Draft version | Shanmuga Srinivas |
| 7th Oct 2009 | 1.4 | Draft version | Shanmuga Srinivas |
| 14th Oct 2009 | 1.5 | Draft version | Shanmuga Srinivas |

[Overview 3](#_Toc243313549)

[Offline Logon Support 3](#_Toc243313550)

[Offline Support Scenarios: 3](#_Toc243313551)

[Test Strategy 3](#_Toc243313552)

[Test Setup 3](#_Toc243313553)

[New Scenarios: 21](#_Toc243313554)

Offline Support Testing Plan

# Overview

This document describes test scenarios for Offline Support. It also explains the test procedures and approach to test all Offline Support.

# Offline Logon Support

Offline Logon Support policy allows target computers running Linux, UNIX, or Mac OS X to log onto domain accounts when the network or domain controller is unavailable .This setting caches logon credentials and account information in the Likewise authentication daemon.

# Offline Support Scenarios:

The scenarios for Offline Support are

1. Offline support within a single Forest.
2. Offline support between two trusted Forests.

# Test Strategy

* **Testing the Offline support within a Forest, when the domains go down one by one including the domain where user is present.**
* **Testing the Offline support in two different trusted Forests, when the domains go down one by one including the domain where user is present.**

# Test Setup

* To test the above mentioned scenarios, two forests should be configured.
* Each forest should contain a parent domain and two child domains as shown in the below figure.
* Also verify that there should be a transitive trust between two forests.

Note: Before going to test offline support, user must be authenticated from the client machine at least once.

**Scenario 1:** **Offline support within a single Forest.**

**Case 1: Verify that the client machine joined to Child domain CA1 is able to login as other child domain CA2 user with in the forest.**

Fig 1.0

**Observation on windows and Linux:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Test Steps** | **Windows behavior** | **Linux behavior** |
| 1.  2.  3. | Make the child domain CA2 down.  Make the child domain CA2 and parent domain PA down.  Make all the domains down in Forest1. | Client machine sends the Kerberos authentication "AS-REQ" to the child domain CA2. Since domain CA2 is down, Kerberos authentication fails. So, client login using the cached credentials.    Since the Parent domain PA is also down, Kerberos authentication fails. So, client login using the cached credentials.  Since all the domains in the Forest1 is down, Kerberos authentication fails. Client login using the cached credentials. |  |

**Case 2: Verify that the client machine joined to parent domain PA is able to login child domain CA2 user with in the forest.**

Fig 1.1

**Observation on windows and Linux**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Test Steps** | **Windows behavior** | **Linux behavior** |
| 1.  2. | Make the child domain CA2 down.  Make the child domain CA2 and parent domain PA down. | Client machine sends the Kerberos authentication "AS-REQ" to the child domain CA2. Since domain CA2 is down, Kerberos authentication fails. So, client login using the cached credentials.  Since the Parent domain PA is also down, Kerberos authentication fails. So, client login using the cached credentials. |  |

**Case 3: Verify that the client machine joined to child domain CA1 is able to login parent domain PA and child domain CA2 users with in the forest.**

Fig 1.2

**Observation on windows and Linux**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Test Steps** | **Windows behavior** | **Linux behavior** |
| 1.  2. | Make the child domain CA1 down.  Make the parent domain PA down. | Client is able to login using the cached credentials.  Client is able to login using the cached credentials. |  |

**Case 4: Verify that the client machine joined to child domain CA2 is able to login parent domain PA user with in the forest after a reboot of client machine.**

Fig 1.3

**Observation on windows and Linux**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Test Steps** | **Windows behavior** | **Linux behavior** |
| 1. | Make the child domain CA2 down and reboot the client machine. | Client is able to login using the cached credentials. |  |

**Scenario 2:** **Offline support for users in two trusted domains.**

**Case 1: Verify that the client machine joined to CA1 domain in Forest 1 is able to login as the child domain CB2 user in Forest 2.**

Fig 2.0

**Observation on windows and Linux**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Test Steps** | **Windows behavior** | **Linux behavior** |
| 1.  2.  3.  4. | Make the child domain CB2 down.  Make the child domain CB2 and parent domain PB down.  Make the child domain CB2 and parent domains PB, PA down.  Make the child domains CB2, CA1 and parent domains PA, PB down. | Client machine sends the Kerberos authentication "AS-REQ" to the child domain CB2. Since domain CB2 is down, Kerberos authentication fails. So, client login using the cached credentials.  Since the Parent domain PB is also down, Kerberos authentication fails. So, client login using the cached credentials.  Since the child domain CB2 and parent domains PB, PA is down, Kerberos authentication fails. So, client login using the cached credentials.  Since the child domains CB2, CB1 and parent domains PB, PA is down, Kerberos authentication fails. So, client login using the cached credentials. |  |

**Case 2: Verify that the client machine joined to parent domain PA in Forest 1 is able to login as the child domain CB2 user in Forest 2.**

Fig 2.1

**Observation on windows and Linux**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Test Steps** | **Windows behavior** | **Linux behavior** |
| 1.  2.  3. | Make the child domain CB2 down.  Make the child domain CB2 and parent domain PB down.  Make the child domain CB2 and parent domains PB, PA down. | Client machine sends the Kerberos authentication "AS-REQ" to the child domain CB2. Since domain CB2 is down, Kerberos authentication fails. So, client login using the cached credentials.  Since the Parent domain PB is also down, Kerberos authentication fails. So, client login using the cached credentials.  Since the child domain CB2 and parent domains PB, PA is down, Kerberos authentication fails. So, client login using the cached credentials. |  |

**Case 3: Verify that the client machine joined to parent domain PB in Forest 2 is able to login as the child domain CA1 user in Forest 1.**

Fig 2.2

**Observation on windows and Linux**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Test Steps** | **Windows behavior** | **Linux behavior** |
| 1.  2.  3. | Make the child domain CA1 down.  Make the child domain CA1 and parent domain PA down.  Make the child domain CA1 and parent domains PA, PB down. | Client machine sends the Kerberos authentication "AS-REQ" to the child domain CA1. Since domain CA1 is down, Kerberos authentication fails. So, client login using the cached credentials.  Since the Parent domain PA is also down, Kerberos authentication fails. So, client login using the cached credentials.  Since the child domain CA1 and parent domains PA, PB is down, Kerberos authentication fails. So, client login using the cached credentials. |  |

**Case 4: Verify that the client machine joined to child domain CB2 in Forest 2 is able to login as the child domain CA1 user in Forest 1.**

Fig 2.3

**Observation on windows and Linux**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Test Steps** | **Windows behavior** | **Linux behavior** |
| 1.  2.  3.  4. | Make the child domain CA1 down.  Make the child domain CA1 and parent domain PA down.  Make the child domain CA1 and parent domains PA, PB down.  Make the child domains CA1, CB2 and parent domains PA, PB down. | Client machine sends the Kerberos authentication "AS-REQ" to the child domain CA1. Since domain CA1 is down, Kerberos authentication fails. So, client login using the cached credentials.  Since the Parent domain PA is also down, Kerberos authentication fails. So, client login using the cached credentials.  Since the child domain CA1 and parent domains PA, PB is down, Kerberos authentication fails. So, client login using the cached credentials.  Since the child domains CB2, CA1 and parent domains PB, PA is down, Kerberos authentication fails. So, client login using the cached credentials. |  |

**Case 5: Verify that the client machine joined to child domain CA1 in Forest 1 is able to login all the domain users in Forest 1 and forest 2.**

Fig 2.4

**Observation on windows and Linux**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Test Steps** | **Windows behavior** | **Linux behavior** |
| 1.  2.  3. | Make the parent domain PB down and login with the Forest 2 domain users.  Make the parent domain PA down and login with the Forest 1 domain users.  Make all the domains down in forest 1 and forest2. | Client is able to login all the Forest 2 domain users using the cached credentials.  Client is able to login all the Forest 1 domain users using the cached credentials.  Client is able to login all the domain users using the cached credentials. |  |

**Case 6: Verify that the client machine joined to child domain CA1 in Forest 1 is able to login the Grandchild domain CB21 user in forest 2.**

Fig 2.5

**Observation on windows and Linux**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Test Steps** | **Windows behavior** | **Linux behavior** |
| 1.  2.  3.  4.  5. | Make the grandchild domain CB21 down.  Make the child domain CB2 down.  Make the parent domain PB down.  Make the parent domain PA down.  Make the child domain CA1 down. | Client is able to login with the grandchild domain user using the cached credentials.  Client is able to login with the grandchild domain user using the cached credentials.  Client is able to login with the grandchild domain user using the cached credentials.  Client is able to login with the grandchild domain user using the cached credentials.  Client is able to login with the grandchild domain user using the cached credentials. |  |

# 

# New Scenarios:

**Case 1: Verify that the client machine joined to domain 1 in network 1 is able to login using credentials of a user from domain 1, when the client machine moves from network 1 to network 2.**

Fig 3.0

**Observation on windows and Linux**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Test Steps** | **Windows behavior** | **Linux behavior** |
| 1.  2.  3.  4.  5.  6.  7. | Join the client machine to domain 1 in network 1.  Login using credentials of a user from domain 1.  Shutdown the client machine.  Connect to network 2 and switch on the client machine.  Login using credentials of the same user from domain 1.  Clear the cache information in Linux using ./lw-ad-cache –delete-all.  Login using credentials of the same user from domain1 | After step 5, it was observed that the client machine is able to login with the cached credentials of the user from domain 1. | Same  After step 7, it was observed that the client machine is not able to login, since no credentials are available. |

**Case 2: Verify that the client machine joined to domain 1 in network 1 is able to login using credentials of a user from domain 1, when the client machine moves back to network 1 from network 2.**

Fig 3.1

**Observation on windows and Linux**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Test Steps** | **Windows behavior** | **Linux behavior** |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10. | Join the client machine to domain 1 in network 1.  Login using credentials of a user from domain 1.  Shutdown the client machine.  Connect to network 2 and switch on the client machine.  Login using credentials of the same user from domain 1.  Clear the cache information in Linux using ./lw-ad-cache –delete-all.  Bring down the client machine to sleep mode.  Connect the client machine back to network 1.  Restart the network.  Login using credentials of the same user from domain 1. |  | After step 10, it was observed that the client machine is not able to login immediately (taking 5 to 10 min of time).  BUG NO:9402 |

**Case 3: Verify that the client machine joined to domain 2 in network 2 is able to login using credentials of a user from domain 2, when the client machine is connected back to domain 1 in network 1.**

Fig 3.2

**Observation on windows and Linux**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Test Steps** | **Windows behavior** | **Linux behavior** |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10.  11. | Join the client machine to domain 1 in network 1.  Login using credentials of a user from domain 1.  Bring down the client machine to sleep mode.  Connect to network 2 and switch on the client machine.  Restart the network.  Join the client machine to domain 2 in network 2.  Login using credentials of a user from domain 2.  Bring down the client machine to sleep mode.  Connect the client machine back to network 1.  Restart the network.  Login using credentials of the same users from domain 1 and domain 2. | After step 11 client machine is not able to login using credentials of domain 1 user (able to login using cached credentials of domain 2 user). | Same |

**Case 4: Verify that the client machine joined to domain 1 in network 1 is able to login using credentials of a user from domain 1, when the client machine moves back to network 1 from network 2 and by changing the password of domain 1 user in AD.**

Fig 3.3

**Observation on windows and Linux**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Test Steps** | **Windows behavior** | **Linux behavior** |
| 1.  2.  3.  4.  5.  6.  7.  8.  9. | Join the client machine to domain 1 in network 1.  Login using credentials of a user from domain 1.  Shutdown the client machine.  Connect to network 2 and switch on the client machine.  Login using credentials of the same user from domain 1.  Change the password of the domain 1 user in AD.  Connect the client machine back to network 1.  Restart the network.  Login using credentials of the same user from domain 1. | After step 9 it was observed that the client machine is able to login using new password. | After step 9 it was observed that the client machine is still able to login using old password for some time.  After 5 to 10 min of time client machine is able to login using new password. |