

JISC DEVELOPMENT PROGRAMMES**Project Document Cover Sheet****TEST PLAN****Project**

Project Acronym	ASMIMA	Project ID	
Project Title	Adoption of Shibboleth for Multiple Identity Management Applications		
Start Date	April 2005	End Date	March 2006
Lead Institution	Cardiff University		
Project Director	Hugh Beedie		
Project Manager & contact details	Joan Wright wright@cardiff.ac.uk 029 2087 4496		
Partner Institutions	n/a		
Project Web URL	http://www.cardiff.ac.uk/17386		
Programme Name (and number)	JISC Core Middleware Programme (Shibboleth Early Adopters)		
Programme Manager	Nicole Harris		

Document

Document Title	Test Plan		
Reporting Period			
Author(s) & project role	Rhys Smith, IT Officer		
Date	03/11/2005	Filename	TestPlan.rtf
URL			
Access	<input type="checkbox"/> Project and JISC internal		<input type="checkbox"/> General dissemination

Document History

Version	Date	Comments
1.0	29 th November 2005	Deliverable sent to JISC

1. Introduction	3
2. ASMIMA Overview	3
3. Related Documents	3
4. Shibboleth IdP Implementation Overview	4
4.1 Hardware.....	4
4.2 Software	4
5 Overview of Testing	5
5.1 Test Roles	6
5.2 Test Strategy.....	6
5.3 Test Approach.....	6
5.4 Assumptions and dependencies	9
5.5 Scope and limitations of testing	10
5.6 Test environment	10
5.7 Test environment validity analysis	10
5.8 Outline of system logging capabilities	10
5.9 Personnel pre-training needs	11
Appendix 1 - Glossary.....	11

1. Introduction

This document aims to catalogue a comprehensive test plan for the Cardiff University Shibboleth implementation that is a result of the Adoption of Shibboleth for Multiple Identity Management Applications (ASMIMA) project in Cardiff University.

The objective of the testing effort described within is to ensure that the Shibboleth implementation in Cardiff University is ready for full-scale usage within the University: that it is able to perform in a resilient manner when conceivable faults occur within the implementation; and that it is able to function adequately when both expected and (reasonably) extreme levels of usage are placed upon it.

The intended audience for this document is the staff in Information Services (INSRV) in Cardiff University, and anyone with an interest in creating a resilient Shibboleth implementation in a large scale organisation.

2. ASMIMA Overview

Cardiff University has secured funding to develop and implement Shibboleth as part of a coordinated series of JISC projects. Shibboleth is a federated authentication mechanism developed as an open standard by the Internet2/Middleware Architecture Committee for Education (MACE). It enables sites that hold user authentication details to securely verify the identity of individuals wishing to access content at another site that requires authentication. Core to the Shibboleth design is the fact that user authentication tokens do not need to be replicated or reproduced at the content provider.

Shibboleth uses XML and, more specifically, SAML (Security Assertion Markup Language) for inter-server communications on user authentication and access rights. In addition, PKI (Public Key Infrastructure) is used to securely verify the participants in a Shibboleth federation.

The Shibboleth implementation at Cardiff University is intended principally to provide authentication as an “Identity Provider” to the EduServ Athens content “Service Provider”. EduServ Athens is a JISC awarded contract that provides a single point of access management and authentication services to a wide variety of academic content. Beyond this principle Athens usage, the Shibboleth implementation will be used further afield with other Shibboleth Federations and Services, including SPARTA, the UK Academic Federation currently being set up by JISC. In the medium term, the Shibboleth implementation must integrate with a Cardiff University-wide single sign on and security system.

3. Related Documents

- Project Plan - \\Shrinsrv\INSRV\SHARED\Projects\Current\P05_17 Shibboleth\Project_Plan\ProjectPlan2a.doc
- Project Plan Workpackages - [\\Shrinsrv\INSRV\SHARED\Projects\Current\P05_17 Shibboleth\Project_Plan\Workpackages.doc](\\Shrinsrv\INSRV\SHARED\Projects\Current\P05_17\Shibboleth\Project_Plan\Workpackages.doc)
- IDMAN Docs - \\Shrinsrv\INSRV\SHARED\Projects\Current\P03_02 Identity Management\Project Library\Documentation

4. Shibboleth IdP Implementation Overview

This section intends to give an overview of the physical and logical setup of the Shibboleth implementation that we are going to be testing.

Shibboleth is currently implemented by ASMIMA on two identical servers (idp1.cf.ac.uk and idp2.cf.ac.uk). The outside world accesses the IdP through a CName, idp.cardiff.ac.uk, which is a Layer 4-7 switch whose function is to route Shibboleth traffic to a working IdP server, transparent to the entity accessing idp.cardiff.ac.uk. All traffic will be redirected to the same IdP server, unless the Layer 4-7 switch determines that the server isn't functioning correctly, in which case it switches to the alternative server.

The Shibboleth software running on these servers performs authentication and authorisation tasks by communicating with the eDirectory FARAWAY tree, through LDAP. The FARAWAY tree holds information relating to all users of Cardiff University that the Shibboleth IdP needs to function correctly.

4.1 Hardware

Each of the identical servers is a Dell PowerEdge 1850 1U rackmount server, with dual 3.2GHz HT Intel Xeon processors and 2GB RAM, connected to the Cardiff University network at 100MB Full Duplex. Each server has a redundant power supply, the main supply coming through a UPS and the redundant supply plugged into a wall socket.

4.2 Software

The IdP part of Shibboleth itself is implemented as a Java Servlet, and can be configured to work with many different variations of software and operating system.

In our case, OS the servers are running Red Hat Enterprise Linux AS release 4 (Nahant Update 2, kernel 2.6.9), and the software details are as follows:

- Shibboleth IdP v1.3c – The actual Shibboleth software, written as a Java Servlet;
- Java 1.5.0_05 – The Java Virtual Machine (JVM) that runs the Shibboleth software;
- Tomcat 5.5.12 – The Java servlet container that interfaces the servlet code with the JVM;
- Apache 2.0.54 – Used to control the flow of internet requests to Tomcat;
- mod_jk 1.2.14 – The apache module that handles the communication between Apache and Tomcat;
- openssl 0.9.7a – Handles all of the secure connections and encryption used by the Shibboleth IdP; (the SSL certificates used by the servers are issued by a globally recognised Certification Authority – GlobalSign)
- openldap 2.2.26 – Used by the Shibboleth IdP to communicate with Cardiff University's LDAP service.

4.3 Shibboleth-Athens

When accessing the Shibboleth-Athens gateway, two attributes are passed across from our IdP to the Athens SP. These are the “eduPersonTargetedID” attribute (mapped from the “cardiffidentityno” LDAP attribute), and the “CardiffAthensOptionSet” attribute (mapped from the “cardiffathensoptionset” LDAP attribute).

The two LDAP attributes are sourced from the FARAWAY eDirectory tree. “cardiffidentityno” is the user's unique identity number assigned by IDMAN; “cardiffathensoptionset” contains a

string value, currently either “cfula#default” or “cfula#noaccess”. The first currently gives the user access to all Athens resources; the latter none.

5 Overview of Testing

Testing is to be split into two main areas – “System” testing and “Usage” testing.

The system testing area is intended to test the set up of the resilient architecture that we have designed. The main test areas will be:

- Backup and Recovery performance – testing recovery procedures in case of server failure;
- Extreme load testing – checking what throughput the software can handle through scripted access attempts;
- Resilience testing – how well the resilient architecture copes with simulated failure of one or more core Shibboleth components.

The usage testing area is intended to check that the Shibboleth software is working as required. The main test areas will be:

- Account security – checking only valid users can log in through Shibboleth;
- Athens access – checking that users can access Athens resources via Shibboleth log in; and that users with the “noaccess” permission set assigned cannot access any resources;
- Athens access (all resources) – checking that every single Athens resource works correctly;
- Athens multiple access – checking users can access multiple resources simultaneously;
- Classic/Shibboleth Athens – checking how well classic Athens and Shibboleth Athens coexist;
- Crossfire – seeing if Shibboleth login to Beilstein Crossfire will work;
- Endnote – seeing if Endnote will work with Shibboleth Athens;
- Load testing – checking the software can perform to the stated level of throughput in a real life situation – all people in the biggest computing pool room in Cardiff University attempting to log in simultaneously;
- Logout behaviour – seeing what the effect of logging out of one resource while still connected to another is;
- Monitoring – seeing whether the monitoring tools installed provide accurate statistics;
- Multiple federation access – whether Shibboleth will successfully work with multiple federations;
- Multiple simultaneous federation access – accessing multiple federations simultaneously;
- Personalisation – checking whether users can personalise their browsing experience at resources that allow it through the use of the persistent ID;
- Server session behaviour – checking what happens to current Shibboleth sessions when a server fails;

- Session behaviour – checking Shibboleth is working as intended in that credentials do not survive beyond the browser’s run-time, and upon logout/login to PC;
- Statistics – seeing whether the usage statistics (both on our end and those provided by Athens match the real usage).

5.1 Test Roles

The following personnel will take part in the testing of Shibboleth:

- INSRV Technical Staff (ROS) – “Dev Eng”;
- A set of personnel, list drawn up by INSRV (SAS) with expertise in each Athens resource - “Expert User Group”;
- Large group of users at the start of a training class in large computing pool room – “Tutorial Group”.

5.2 Test Approach

The following phases will occur during the testing of Shibboleth:

1. “Dev Eng” will conduct these tests as soon as the implementation is ready to be tested;
2. “Expert User Group” will be asked to test specific Athens Shibboleth-protected resources and report back their success/failure;
3. “Tutorial Group” will be asked at the start of a class to help with the testing of a new system. They will be instructed by a member of INSRV in exactly what to do, this staff member will get all users to login to Shibboleth at roughly the same time. As a “reward” for helping us in this way, member of this group of users will be able to use Shibboleth from then on – before the rest of the university.

5.3 Test Strategy

Testing	Phase	Role	Area	Description	Test	Expected Result
System	1	Dev Eng	Backup and Recovery	Test recovery procedure	Kill live server in some way (e.g. deleting core system files)	Systems team should restore system from backup within an acceptable amount of time
System	1	Dev Eng	Extreme Load Testing	Test the throughput capability of Shibboleth implementation	Use a script to simulate logins at an increasing rate and see what the servers can handle	Should be able to at least handle our stated minimum throughput (10 logins per second)
System	1	Dev Eng	Extreme Load Testing	Test the concurrent capability of Shibboleth implementation	Increase hard-coded maximum threads to see what the servers can handle	Should be able to at least support our stated minimum throughput (10 logins per second)

System	1	Dev Eng	Resilience	Test resilient hardware	On each server, simulate power failure (pull relevant power cord)	UPS keeps server running
System	1	Dev Eng	Resilience	Test resilient hardware	On each server, simulate UPS failure (pull relevant power cord)	Redundant supply not plugged into UPS keeps server running
System	1	Dev Eng	Resilience	Test resilient hardware	On each server, simulate disk failure (pull plug from RAID)	RAID compensates server keeps running
System	1	Dev Eng	Resilience	Test L4-7 switchover	On each server, simulate software failure (kill vital processes: Apache, Tomcat, etc.)	L4-7 switch automatically diverts requests to the other server
Usage	1	Dev Eng	Account Security	Check valid users can login	Attempt to login to Athens resources through shibboleth using a valid user account	Login successful, shibboleth session established
Usage	1	Dev Eng	Account Security	Check invalid users cannot login	Attempt to login to Athens resources through Shibboleth using an invalid user account	Login unsuccessful, no shibboleth session established
Usage	1	Dev Eng	Account Security	Check valid, but suspended (disabled) accounts cannot login	Attempt to login to Athens resources through Shibboleth using a valid user account that has been suspended (disabled)	Login unsuccessful, no shibboleth session established
Usage	1	Dev Eng	Athens access	Check that a given username can log in to Athens protected resources	Log in with a normal user's credentials, attempt to access an Athens protected resource	Able to access the resource
Usage	1	Dev Eng	Athens access	Check that a given username with the "noaccess" permission set can log in but not access Athens protected	Log in with the credentials of a user with the "noaccess" permission set, attempt to access and Athens protected	Unable to access the resource

				resources	resource	
Usage	1	Dev Eng	Athens multiple access	Check that accessing multiple Athens resources work in the same Shibboleth session	Login to Athens through shibboleth, access several resources	Able to access all resources without being asked for credentials after the initial login
Usage	1	Dev Eng	Crossfire	Check if Crossfire works with Shibboleth Athens	Load Crossfire, attempt to access Athens-Shibboleth resources	Crossfire should work
Usage	1	Dev Eng	Endnote	Check if Endnote works with Shibboleth Athens	Load Endnote, attempt to log in to Athens using Athens-Shibboleth credentials	Endnote should be able to access Athens resources
Usage	1	Dev Eng	Logout behaviour	Check behaviour when logging out of one resource when connected to another	Login to two Athens resources concurrently, log out of one, check whether still logged into other	Should stay logged in on other resource
Usage	1	Dev Eng	Monitoring	Check accuracy of statistics produced by reporting tools	Use Shibboleth server a specified amount, check logs to confirm, then check reported statistics and compare accuracy	Statistics should be accurate and correct
Usage	1	Dev Eng	Multiple Federation Access	Check Shibboleth works correctly with multiple federations	Attempt to access shibboleth protected resources on different federations in the same Shibboleth session	Gain access to Shibboleth protected resources, providing credentials the initial time only
Usage	1	Dev Eng	Multiple Simultaneous Federation Access	Check Shibboleth works correctly when accessing multiple federations simultaneously	Login to shibboleth, then attempt to access resources of two federations (e.g. Athens and SDSS) simultaneously	Gain access to Shibboleth protected resources, providing credentials the initial time only
Usage	1	Dev Eng	Personalisation	Check that users can personalise resources using the Shibboleth persistent ID	Login to an Athens resource with personalisation options, apply personal settings, re-login in a different	Personalisation of the Athens resources should work

					Shibboleth session	
Usage	1	Dev Eng	Server Session behaviour	Check current session loss	Login to Shibboleth, simulate failure of current idp server, then attempt to access Athens resource	Shibboleth session should have persisted, resource viewable
Usage	1	Dev Eng	Server Session behaviour	Check currently logging-in session loss	Login to Shibboleth, during login process (post AuthN and pre AuthZ) simulate failure of current idp server	Shibboleth session should not be properly established, resulting in having to re-provide credentials
Usage	1	Dev Eng	Session behaviour	Check correct credential destruction behaviour	Login to Shibboleth, close browser, attempt to access Athens resource	Prompted to re-enter credentials
Usage	1	Dev Eng	Session behaviour	Check correct credential destruction behaviour	Login to Shibboleth, logout of computer, re-login, attempt to access Athens resource	Prompted to re-enter credentials
Usage	1	Dev Eng	Statistics	Check accuracy of provided statistics	Keep track of Shibboleth usage during earlier tests, compare to reported statistics	Usage statistics should be accurate and correct
Usage	2	Expert User Group	Athens access	Check that all Athens resources work with our Shibboleth implementation	Assign each Athens resource to someone (list provided by SS) and check they all work	Able to access all resources
Usage	3	Tutorial Group	Load Testing	Test the throughput capacity of our Shibboleth implementation in a real life situation	Have a large group of users simultaneously attempt to sign in to Athens using Shibboleth	All logins should be successful.

5.4 Assumptions and dependencies

For this testing to take place, the following must be in place:

- The Shibboleth implementation must be ready and working (including backup procedures, resilient implementation and software);

- Every user who is to take part in the testing (“Dev Eng”, “Expert User Group” and “Tutorial Group” must have the “cardiffidentityno” and “cardiffathensoptionset” attributes set correctly on the FARAWAY tree;
- The “Expert User Group” list must be drawn up and the members of it shown how to use Shibboleth
- The INSRV staff members taking charge of the “Tutorial Group” testing must be given a set procedure to follow outlining the process of the test to take place;

5.5 Scope and limitations of testing

The test strategy defined above, while attempting to test the Shibboleth implementation as thoroughly as possible, do not (and can not) test the implementation with a usage level distribution. However, since our testing should test the implementation at usage levels far above the expected, this is not seen as a problem.

5.6 Test environment

All tests by “Dev Eng” will be conducted on INSRV machines, running the stock INSRV Windows XP Image (820), Linux, and Mac OS X (10.4). All tests will be run in Microsoft Internet Explorer 6, Mozilla Firefox (1.0.x and 1.5 RC x), Opera (7 and 8), and Safari (as appropriate to the platform). Additionally, the appropriate tests will be run from on and off the Cardiff University network (from the home broadband connection of a selection of INSRV staff, in order to check it works on multiple ISPs – including AOL).

The tests run in the computing pool room will be on the stock INSRV Windows XP Image (820), running the web browser of the user’s choice – to make it as realistic as possible.

5.7 Test environment validity analysis

As the tests are going to be run on pretty much every major supported web browser and OS combination any Cardiff University member is likely to use, the test environment is pretty close to future real life use of Shibboleth.

5.8 Outline of system logging capabilities

The following tools will be used to log the testing as it happens, and analyse the results of the testing:

- Built in Shibboleth logging – the log files will show all Shibboleth usage;
- Built in Tomcat logging – the log files will show all Tomcat processes;
- Built in Apache logging – the log files will show all connection attempts;
- Built in mod_jk logging – the log files will show all information relating to Apache – Tomcat interaction;
- Built in linux system logging – the log files will show all information relating the processes happening on the servers;
- MRTG – MRTG will be used for live system monitoring, which includes server information (CPU/Memory/Disk/etc usage) as well as Shibboleth usage through Perl scripts which parse the Shibboleth log files;
- Athens provided statistics – Statistics provided by EduServ that show usage of the Shibboleth – Athens gateway from their end of the chain.

5.9 Personnel pre-training needs

The following training needs to take place before the testing can be completed:

- The members of the “Expert User Group” need to be shown how to Login to Shibboleth and access the particular Athens Resource assigned to them;
- The staff member(s) responsible for leading the “Tutorial Group” needs to be trained in the usage of Shibboleth and what common errors may be received, so they are able to quickly diagnose any errors that may occur during the test.

Appendix 1 - Glossary

Apache – Web server software

ASMIMA (Adoption of Shibboleth for Multiple Identity Management Applications) – The JISC funded project at Cardiff University, part of the Core Middleware Program, specifically as part of the Shibboleth Early Adopters scheme

Athens - Athens is an Access Management system for controlling secure access to web based services

eDirectory - Novell eDirectory (formerly called Novell Directory Services) is an X.500 compatible directory service software product released in 1993 by Novell for centrally managing access to resources on multiple servers and computers within a given network

EduServ - A not-for-profit IT services group, providing support and solutions for business critical hosting, for e-learning, e-government and e-commerce, and for network identity management. Currently hold the JISC contract to run Athens

FARAWAY Tree – An eDirectory tree used in Cardiff University by web applications that need access to only a small proportion of the attributes held in the full tree

IdP – Shibboleth Identity Provider

LDAP - A protocol used to access a directory listing, in this case, the FARAWAY tree

Linux (properly GNU/Linux) – An open source operating system

Mod_jk – An interface between Apache and Tomcat

MRTG (Multi Router Traffic Grapher) - Free network traffic monitoring system which displays the results as graphs on the Web

SDSS (Shibboleth Development Support Service) – A development Shibboleth federation for managing access to UK academic online resources. Will eventually become SPARTA

SP – Shibboleth Service Provider

SPARTA – A UK-wide Shibboleth federation for Higher Education institutes and resources

Tomcat – A Java Servlet container and web server