Implementation of an Authentication and Authorization Architecture for Mobile Internet Users

Thomas Spreng, Torsten Braun, Marc Steinemann and Attila Weyland
Institute of Computer Science
and Applied Mathematics
University of Bern,
Neubrückstr. 10, CH-3012 Bern
Email: {spreng, braun, steine, weyland}@iam.unibe.ch

***Abstract*— Internet Users' mobility requirements have increased in the past few years. Roaming using a single authentication procedure within different universities has become more and more important. The problem is that many universities are implementing solutions of their own, which cannot always be used by visitors. Therefore, we have looked into designing a mobile user authentiction scheme based on already deployed technologies that will use a single and homogenous authentication and authorization procedure everywhere. The chosen solution consists of a web-based network access portal, which uses Shibboleth as an authentication and authorization framework. This allows the users to easily get Internet access using their mobile devices at any participating organization using a secure authenitcation method.**

1. Introduction

1.1. Motivation

Internet users' habits are changing in the way that many users prefer working with mobile devices. Many mobile devices already include wireless network adapters. In order to satisfy these needs, wireless docking networks become more and more important. In many universities radio networks for mobile Ethernet users are mostly reserved to users of the respective university. The Swiss research network provider SWITCH [1] offers a workaround for connecting mobile devices by means of virtual private network (VPN) tunnels back to the users home university. This workaround has several drawbacks, especially the installation and maintenance costs of the virtual private network gateways are high. Additionally, users do not get access to local services offered by the hosting university and cannot roam. It only works in universities that prepare their wireless LAN for this workaround. SWITCH is implementing a Swiss-wide authentication and authorization infrastructure (AAI) [2] for universities that allows users to access enabled services from any place where Internet connection is available. This infrastructure is built upon the Shibboleth middleware [3] and operates in a productive environment since June 2004. At this time it is primarily used for protecting access to e-Learning web resources from the participating universities. Logging-on to wireless networks however is not addressed in the present solution. Users should be enabled to easily connect and use their mobile devices in all university campuses with the same security level they are used to in their home university.

1.2. Goals

The goal of this work is to develop a concept and a prototype implementation of an authentication and authorization procedure for mobile Internet users. The focus is based upon the situation that exist among Swiss universities, which means that already deployed infrastructures and technologies should be considered. Nevertheless any solution must be able to be adapted to work in different scenarios, not only within universities. Five main objectives have been defined to be achieved:

1) Users of mobile devices should be able to access home and foreign wireless and wired LAN networks in a single authentication and authorization procedure. This means that no preliminary action is required for the user before accessing a network.
2) Users should be able to use the same credentials as they use in their home universities. Any type of a local user directory used for authentication purpose must be avoided since it will make any solution unscalable and dangerous in terms of privacy.
3) Users should be able to roam in the visited network without re-authenticating at each hand-over.
4) No third-party software should be used on the client side. This is also to support that the solution will be as platform and application independent as possible.
5) Users should have access to local services offered by the hosting organization such as printers for example. Access to such local resources may be handled differently in each organization since it depends on their policy, network topology, etc. The solution however should take this into account and show how local resources can be accessed.

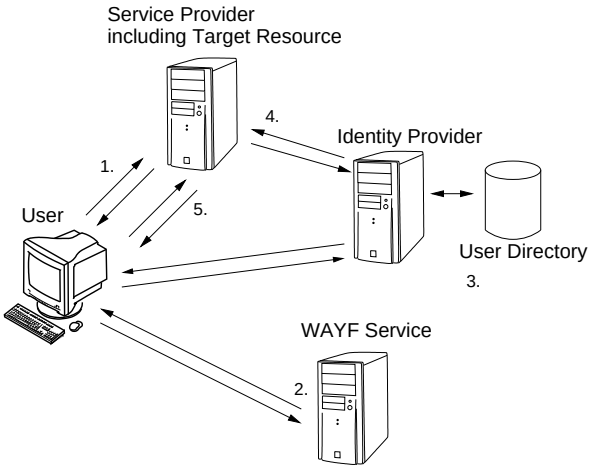

Fig. 1. Shibboleth Authentication and Authorization Message Exchange

1.3. Shibboleth Authentication and Authorization Infrastructure

Shibboleth [4] will be used as an authentication and authorization backend in the schemes that will be explained in 2.

Shibboleth is a web-based authentication and authorization infrastructure (AAI) mainly developed by Internet2/MACE. A key aspect is the federated administration, which means that user identities and attributes are controlled and administrated by a single entity, the identity provider. Any resource relies on this entity that delivers information about a user to support authorization decisions. Also users are only authenticated at their respective identity provider, never at a resource locally. Information must be exchanged in a secure way using open, standard-based solutions like OpenSAML and OpenSSL.

A Shibboleth authentication and authorization procedure is shown in Figure 1 and works as follows. If a user tries to access a Shibboleth-protected resource (also known as service provider), it tries to authenticate the user first (step 1.). Most likely this will not succeed unless the user has already visited this resource during the current session. Since the identity is unknown, the service provider part will redirect the user to the “Where Are You From” (WAYF) server (step 2.). This service is maintaining a list of all participating organizations and their corresponding identity providers within the Shibboleth federation. It is the only centralized service used in this infrastructure. The user then has to choose its home organization and immediately gets redirected to the respective identity provider (step 3.). Once authenticated the Handle Service at the identity provider creates a handle, which acts as a reference to the user and sends it to the service provider. The request to send this handle back to the resource was sent by the WAYF server. The part, which is responsible for acquiring user handles at the service provider, is the Shibboleth Indexical Reference Establisher (SHIRE). Once the SHIRE has succeeded the impersonation checks on the handle it passes it to the Shibboleth Attribute Requester (SHAR) component at the service provider. The SHAR then can send attribute query messages (AQM) to the identity provider to get information about this user (step 4.). Based on these attributes the service provider may grant access to this resource (step 5.).

While the design of the Shibboleth protocol is very nice in terms of federated administration, scalability and security aspects, it has one major drawback: The current implementation is completely browser based. This means that every user needs to use a HTTPS capable web browser in order to complete the authentication procedure since the user interaction is based on Security Assertion Markup Language (SAML) [5] browser profiles. These profiles work with HTTP redirects and HTTP POST and GET requests, hence a browser is needed.

2. Authentication Schemes for Mobile Internet Users

There are various approaches to design and implement an inter-institutional wireless network access architecture. This is given by the fact that there is no standard solution for this problem. Many different authentication infrastructures and protocols can be used in such a scenario and some are already being deployed such as Shibboleth even though it is a browser-based solution. This section presents an analysis of some schemes that have been considered for such a scenario.

2.1. Web-based Authentication using a Shibbolized Portal

In this scheme the already deployed Shibboleth authentication and authorization (AA) infrastructure is used to provide access contol decisions for mobile internet users.

The mobile users attach their devices to a so called docking network, a separated part of the institutional network dedicated to home and foreign mobile user access. This docking network should be considered hostile and therefore be a part of a public demilitarized zone (DMZ). Network access should be possible through Ethernet ports and wireless access points, which must share a common SSID within the whole federation. All data traffic directed to other network parts or the Internet is routed through a firewall/gateway server, which connects the docking network to the rest of the demilitarized zone. In order to grant network access, all unauthorized HTTP(S) traffic is caught at the gateway server and redirected to a Shibboleth-protected web server. This captive portal can either run on the gateway itself or on a dedicated server. This portal acts as a Shibbolized resource in the infrastructure where each user needs to be successfully authenticated by his respective identity provider in order to get access to the protected web server. Authorization decisions can then be made based on user attributes provided by Shibboleth. That way role based access control can be achieved. Upon successful authorization the web server triggers the firewall to grant network access for the respective client.

The web-based Shibboleth portal features a single, identical and easy to use network access procedure, where no additional software is needed except a HTTPS capable browser. The

open sessions however are vulnerable to some hi-jacking and eavesdropping attacks.

2.2. IPSec Solution Based on Client Certificates

This scheme uses a local IPSec gateway using client certificates to a provide secure network access architecture.

Like in the previous scheme a separate docking network is used to provide access ports (be it wired or wireless) for mobile users. All wireless access points must share the same SSID that is broadcasted. The docking network is separated from the institution's network by a firewall/IPSec gateway. Only incoming IPSec ESP packets are able to pass the firewall and are routed outside the docking network. In order to establish such a local IPSec tunnel each user must be provided with a client certificate. One of the major drawbacks of a full-featured Public Key Infrastructure (PKI) is the scalability and administrative overhead. This can be greatly improved if local gateways are used and no certificate revocation lists need to be maintained. The latter can be achieved by issuing client certificates with a short validity period. Every new user will be redirected to a Shibboleth-protected portal by the firewall, if no IPSec tunnel has been established yet. The portal's web server is configured as a Shibboleth resource and hence a user must be authenticated by his identity provider and authorized by the resource to get access to it. The purpose of this portal is to generate and distribute public key pairs and certificates with a short validity period, which are then used to set up the client's IPSec configuration. Once the user has obtained such a certificate and configured its VPN client, a tunnel to the local IPSec gateway can be established, if the validity of the client certificate can be verified by the local certificate authority (CA) or via a certificate chain to the issuing CA. Any data traffic encapsulated in such a IPSec tunnel is then passed by the firewall and routed outside of the docking network.

The IPSec scheme offers a very mature and secure network access control in contrast. But each user must have an IPSec client installed and configured in advance before Internet access is granted. Also the used VPN configuration could conflict with already existing set-ups.

2.3. 802.1x using EAP-TLS and Client Certificates

This scheme uses a similar approach than the previous IPSec based scenario. Network access is controlled by 802.1x network ports using EAP-TLS with client certificates as an authentication protocol.

First of all the docking network needs to be equipped with 802.1x capable access points and Ethernet ports. All network ports (wired and wireless) must be in an unauthorized state so that no client can access the network without using the 802.1x protocol. Since client certificates are used in order to authenticate the users and no network access is possible without them, they must be obtained in advance. For that reason a central certificate distribution server is needed. This can be achieved by a Shibboleth protected web server that is accessible from everywhere within the federation. Upon a successful Shibboleth authorization it generates a public key pair and a certificate for each user, which will be used for the 802.1x authentication. In order to reduce the complexity of a full-featured PKI the same limitations as in the previous scheme can be used, namely a short validity period and no signing purpose for client certificates. After the user has installed such a certificate and configured his EAP-TLS capable client, 802.1x port based access is possible within the docking network. As soon as the client is attached to a network port the EAP initiation phase with the authenticator (most likely an access point) takes place followed by EAP-TLS authentication messages. These EAP-TLS requests are forwarded to a local authentication server, which is able to verify the validity of the supplied client certificate. Finally, an EAP success message is sent back to the authenticator and the port is put in an authenticated state, allowing full access to the client.

The 802.1x based scenario also offers a very secure access control mechanism. But an additional EAP client might be needed depending on the client's operation system. The major drawback is the configuration phase, which has to be done in advance before attaching to the docking network.

2.4. Rewriting SSL Proxy

This scheme is using using an Shibboleth-protected SSL proxy that acts as a limited VPN gateway for HTTP sessions.

The mobile users attach their devices to a docking network, which is separated from the rest of the demilitarized zone by a firewall gateway server. The access to the docking network itself is unrestricted but any unauthorized HTTP traffic is redirected by the firewall to the Shibboleth-protected SSL VPN proxy. Only packets originated from this proxy server are passed by the firewall. Other connections to the outside of the docking network are blocked but Shibboleth authentication messages will not be blocked. The use of the SSL VPN proxy is restricted to authorized Shibboleth users. If this procedure has been successful the user opens a HTTPS connection to the SSL VPN's proxy page. There he may access any URL he wants to visit and the proxy will connect to this URL on behalf of the user and pass the output back to the users browser. The proxy als needs to rewrite any anchor tags to connect via the SSL VPN proxy instead of a direct connection. This is not very easy to achieve since this does not only affect HTML but also other languages like Javascript, which are often used to generate links on the client side. Therefore some web sites may not work as intended.

The scheme with the SSL VPN proxy actually does not have any of the drawbacks of the other three proposals but only it can be used with HTTP sessions.

3. Mobile User Authentication using a Shibboleth-protected Portal

Given on an evaluation of the previous schemes we present a solution, which is based on the scenario described in Section 1, using a Shibboleth-protected captive portal to offer role based network access.

This solution uses a web-based authentication scheme in order to grant access to the docking network for mobile

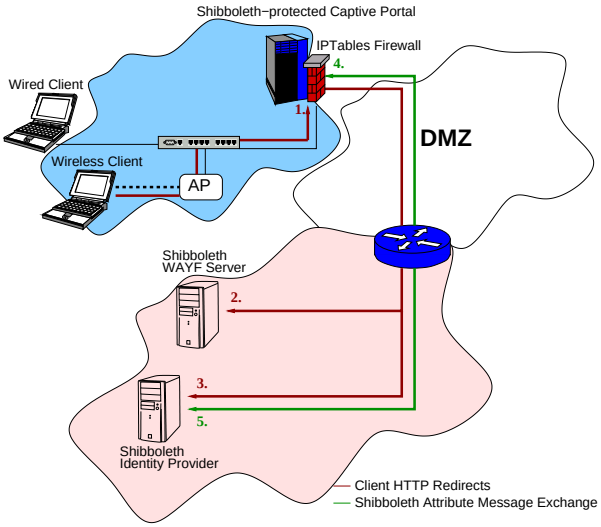

Fig. 2. Possible network topology using the proposed solution

Internet users. As the authentication and authorization backend, Shibboleth seems the most suitable one although other technologies like RADIUS or Diameter could be used as well. The advantage of using Shibboleth is that it is already deployed in many Swiss Universities and features a secure, web-based authentication and authorization infrastructure, which is suitable for this scenario.

3.1. Network Topology

The mobile users connect their devices in a separate network designed for this purpose, the so called docking network. This network is separated from the private parts of the organization's network. It is recommended that a IEEE 802.1Q VLAN is deployed for the docking network. This has some administrative advantages and security aspects when introducing new technologies for authenticating and authorizing users. A new and separate VLAN may then be set up in parallel without physically altering of the Ethernet cabling, which will not be possible otherwise. The docking networks within all the participating institutions should share the same SSID for access points. In addition to that a DHCP service has to be provided for easy network configuration.

3.2. Access Control

Accessing the docking network does not need any authorization. Any device either associated with an Access Point or attached to an Ethernet port has full access to the docking network as shown in Figure 2. Only packets leaving the network need to pass the firewall/gateway and are blocked by default except packets used for Shibboleth authentication messages. The gateway server hosts a Shibboleth-protected web resource combined with a packet filter (firewall) and a captive web portal software. HTTP traffic of unauthorized users is intercepted and redirected to the protected resource (Figure 2, step 1.). There users are redirected to the WAYF server (Figure 2, step 2.) and then are directed to authenticate themselves at their identity provider (Figure 2, step 3.), which is usually at their home organization. Actually a link to a Shibboleth-protected area on the portal server's web server is presented. By following this link the Shibboleth infrastructure makes sure that all users are authenticated and the server gets all the required user attributes it demands (Figure 2, steps 4. and 5.). Based on that information the packet filter may be triggered to grant access to the Internet. Since the resource can ask for any attributes from the user's identity provider, such as the organization name, unit, user role and so on, role-based authorization is perfectly possible. This means that the access is not only granted because a user is authenticated but it also depends on what attributes he provides. One problem is that the user has to authenticate himself directly at the respective identity provider. This means he must be able to access his home organization and the Shibboleth WAYF server already during an unauthenticated state. Therefore all gateways must maintain a list of all identity providers and WAYF servers within the federation. This might become a scalability problem when there are thousands of sites to maintain but in a Swiss-wide federation this should work without a problem since it is already done automatically by the Shibboleth resource software itself.

Another function of the portal is to make sure that users who left the docking network get disconnected appropriately. Such a disconnect link is provided on the web portal and when followed by the user all the corresponding packet filter rules will be flushed. But the problem is that one cannot rely on this manual log-out feature alone. Users may forget to click such a link when leaving the network or even worse may be a victim of a session hijacking attack. Therefore, a disconnect or re-authentication policy has to be applied. Each authorized access has a maximum lifetime, after which a new re-authorization must be taken place. Since Shibboleth integrates Pubcookie [7] this might be done without any user interaction. This maximum lifetime can be reduced by the result of activity tests that will be run in the background. These tests may consists of ICMP requests, ARP look-ups and traffic measurements.

3.3. User Interface

When a new user wants to connect to the docking network he either plugs his device to a LAN port using a RJ45 cable or associates with a access point using the common SSID. Once connected the user has to open a web browser and connect to any server. Since the device is still in a unauthorized state any HTTP(S) traffic is captured by the firewall and redirected to the Shibboleth-protected portal. There a login link is presented to the user, which leads to a Shibbolized part of the portal that forces him to authenticate at his identity provider if no active session exists. Once authenticated both methods lead to a page that tells the user whether he is authorized to access the network or not. This decision is based on the configuration settings of the portal. As already mentioned before, access may be granted based on user attributes and not only on authentication state. If access is granted the user is presented a page with information about his access state and a link to

log out when he wants to leave the docking network.

3.4. Guest User Handling

People that are not members of the participating organizations should still be able to get network access if needed. There are more or less two possible solutions. One would be to add temporary accounts to a identity provider's user directory. That would make them act like members of a participating organization and thus the docking network portal would not make any distinction between those two user groups. The second solution is to create an own identity provider dedicated to guest users for the whole Shibboleth federation [6].

3.5. Interconnection with other Solutions

Since this scheme is not meant to be the only solution that will be used in the future, it is more of an intermediate solution until one of the more secure, seminal methods are widely implemented and adopted. Until this happens this scenario will have its uses as a very user friendly and easy to use implementation for authenticating mobile Internet users. Since most enterprise level access points supports multiple VLANs it is possible to build up new solutions aside this one without interference. Many organizations offer VPN connections to be made to their private address space for its members. Such VPN tunnels are possible in this scenario even though they might be limited if a network address translation (NAT) is performed on the gateway.

3.6. Security Concerns

The major drawback of this solution is that the network access session is vulnerable to some spoofing, session hijacking and man-in-the-middle attacks. The user authentication and authorization procedure provided by Shibboleth however is not known to be vulnerable to such attacks. Some of these threats can not be completely prevented since a layer 3 access control is performed. Therefore it would be important to passively detect attacks if possible or any other suspicious events. That way an operator could react upon such incidents and prevent any further abuse.

Securing the communication channel between the mobile client and the wireless access point using WPA or 802.11i is not really possible at the moment. There are two ways of deriving keys in order to secure the communication. The first one is by setting up a pre-shared key. This is only viable in a local home or in an ad-hoc environment but it is useless in a larger scope. The second possibility is to establish a session key based on a 802.1x authentication. That way a dynamic per session key can be used between a single supplicant and the authenticator. Unfortunately Shibboleth needs at least limited Internet access to the WAYF and the identity providers during the authentication phase but 802.1x prohibits any traffic but EAP messages on an unauthorized port. There is no way of relaying or encapsulating Shibboleth authentication messages in EAP packets since the procedure is completely browser (HTTPS) based at the moment and requires user interaction.

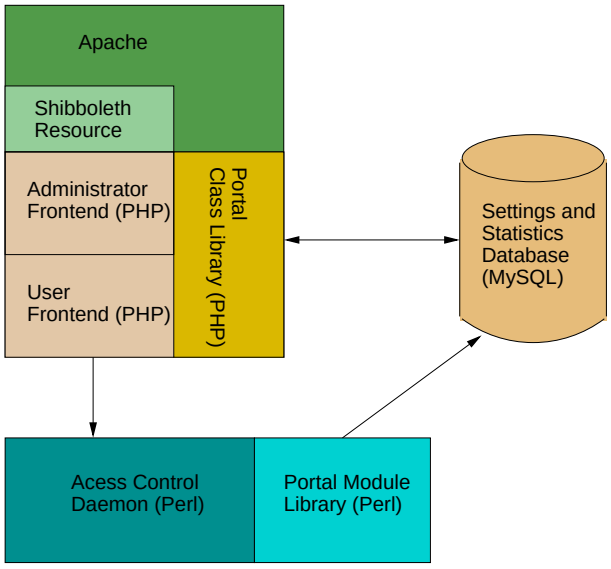

Fig. 3. Main software components

4. Implementation of a Shibboleth-protected access portal

All components have been implemented on a single server using a Debian (3.1) operation system running a 2.4 Linux kernel. The portal is using IPTables to filter packets and thus is limited to run on Linux-based operating systems.

The complete portal software components can be roughly divided into three parts:

- Third party software
- Portal webpage scripts written in PHP v4.x
- Network access control scripts written in Perl v5.8

Figure 3 shows a schematic overview of the portal webpage scripts and the network access control scripts components.

4.1. Third Party Software

The portal implementation is based on various third party software, which are:

- Apache web server
- IPTables packet filter
- DHCP daemon
- MySQL database
- DNS proxy

The web server chosen by this implementation is Apache 2.0 with mod_php4. The web server is used to host the Shibboleth protected resources of the portal. The actual access control on the portal is done by IPTables. This is a Linux specific packet filter, which serves all the needs that are demanded by the portal software such as stateful filtering, network address translation. The DHCP daemon is actually not an essential part of the portal in terms of that it would not work without it. But from a user point of view it provides ease of use because the client's IP configuration is done automatically without any interaction. The MySQL database is used to store information about currently open sessions and is mostly queried by the

administrator interface. In order to reduce redundant data and to provide data privacy only the users Unique ID is stored in the database. This ID is an attribute that uniquely identifies a user in a Shibboleth federation. Besides this Unique ID, user accounting information is also stored in the database. The last third party software used in this implementation is a DNS proxy service. This way the DNS port (53) does not need to be open by default.

4.2. Portal Web Page Scripts

The portal's web page scripts can be divided into three different parts, namely the captive user interface, the administrative interface and the portal class library.

4.2.1. Captive User Interface: This interface is kept as small as possible since the only function to the user is to either log in or log out. If the user connects to the captive portal he is presented a portal welcome page with a log-in link that leads to a Shibboleth-protected area of the web server. If the user follows this link he is passed to the Shibboleth authentication procedure. Upon this authentication and authorization process an information page is displayed showing the access control decision and if it was successful a link to manually log out.

4.2.2. Administrator Interface: It is used in order to adjust the portal settings or to monitor the users and their sessions. Unlike the user interface it can also be reached from outside the docking network. Only users that have the privilege bit set in the portal database may access this interface. The functionality contains of listing all users and their session data such as how much traffic was used in each period. This data can possibly be used for accounting purposes. Also a list of currently online users is available featuring a live network traffic graph. The third function is the configuration page where fine grained access control decision can be made. These access control decisions consist of a set of rules that either deny or grant access based on user attribute values. At last there is also some functionality to maintain and clear the database if needed.

4.2.3. Portal Class Library: The library provides functionality that is not done by the portal scripts themselves. This is an object oriented class library written in PHP, which provides encapsulation for the portal's core functionality. Since PHP does not feature very mature object oriented programming support the library is kept pretty simple, also in order to make it work with different PHP versions.

4.3. Network Access Control Scripts

There are some portal access control scripts written in Perl that handle the interaction with the operating system and the third party software mainly. While the PHP part of the portal software is responsible for the interaction with the users and Shibboleth, the access control scripts are primarily used to access and modify the packet filter and the network address translation rules.

One script is used to get a list of IP addresses of all identity providers that are part of the current Shibboleth federation. These IP addresses will then be added to the IPTables rule by another script that permits access to those sites per default. These are the only IP addresses that need to be accessible from the docking network even for users that are not yet authorized.

There is a script that is used to bootstrap the first administrator access. Since the administrator interface is also protected by Shibboleth and hence can only be accessed by an user that is part of the Shibboleth federation, the first administrator privilege must be granted by this script. As soon as there is at least one administrator any further privileges can be handled via the admin web interface.

The most frequently called script basically just grants network access to a user. This script is also called by the user interface whenever a user is authorized by the captive portal. This program must run as root, because it alters the firewall table and it is recommended to use the "sudo" package to give the PHP interpreter the rights to run it as root. This is also suggested in the installation notes and there is already a sample "sudo" configuration file, which can be used without any modification if needed.

The last access control script acts as an access control daemon, which means it detaches itself from the console and runs in the background. This daemon is the main access control program, it keeps track of all users and sessions that are currently on-line and it synchronizes the user database with the IPTables sessions in order to maintain a secure state of the access control chain. Another functionality of the script is to measure the real-time traffic that is is being generated by each user session. This data is used twofold. First of all it can be used to integrate accounting functionality into the portal since all network traffic is exactly measured for each user and it is also displayed in the administrator interface in a real time network graph. The second purpose of this data is that it is being used to analyze each users activity and if the user is inactive for a certain period of time, the network access session will be closed by the daemon in order to prevent session hijacking attempts.

All these access control scripts use a custom Perl module library, where the core functions are encapsulated.

4.4. Performance Measurements

Since this portal is about authenticating and authorizing users for network access most performance tests do not apply here. It would not make sense to measure the response time the portal is delivering or the time in which the portal interface can be retrieved under heavy load. More important questions are how the access control daemon scales or how long the duration of a roaming session handover is.

4.4.1. Scalability of the Access Control Daemon.: It is important to see how the access control daemon scales with an increasing number of concurrent clients. Upon these results one can know what hardware it would require in order to make sure that the portal is still working properly even if the whole docking network address space is covered by clients. The design of the access control daemon and the corresponding modules was done with scalability in mind, so this is not expected to be a problem and measurement tests have shown

that a single server with modest hardware is capable of handling hundreds of concurrent clients.

4.4.2. Session Handover Time: This proposed solution features re-authentication and re-authorization without user interaction. Thus a session handover between multiple docking network is possible and it is very interesting to know what time such a procedure would take. One must notice that the timing highly depends on the underlying architecture and the involved technologies. Such a session handover can be divided into 4 different parts.

1) 802.11 MAC layer handover
2) IP layer reconfiguration
3) Shibboleth re-authentication
4) Portal re-authentication

The 802.11 MAC layer handover only applies if the docking network is using a 802.11(b) layer of course. This measurement relies on that technology, because it is the most widely used one. But it is up to the operator to choose the underlying network type as long as it can be used with a TCP network stack. The 802.11b handover can be further divided into the discovery, search and handover execution phase. According to [8] this is also highly hardware dependent and can take from 1104 ms up to 1920 ms.

The second reconfiguration procedure (IP layer) is done by the DHCP protocol. In general it works as follows. The client sends a DCHPDISCOVER packet, which will be responded by a DHCPOFFER packet from the server. Then the client answers with a DHCPREQUEST and if this requests is accepted by the server it will sent a final DHCPACK packet. Then the client may adjust its IP configuration according to the information that has been sent with the DHCPOFFER packet. Unfortunately this can take several seconds and the time it takes varies on different operating systems. Tests have shown that using some operation systems it takes a constant period of about 4000 ms while other systems such as FreeBSD take from about 200 ms up to 9000 ms to reconfigure using DHCP.

The third phase is the Shibboleth re-authentication procedure. This is based upon HTTP redirects and consists of the following messages. The client browser is forced to make a HTTP connection to the portal server and is redirected to the Shibboleth WAYF server. Then the client gets redirected further to the corresponding Shibboleth Identity Provider where re-authentication takes place. Based upon the configuration of the Identity Provider the latter procedure can be done transparently without any user interaction. Then the browser gets redirected back to the portal and the client is re-authorized and is granted access in the new docking network. Since this phase is handled by Shibboleth alone and the redirects may be different for several users, it is very difficult to get a general time frame for this procedure but typically this will be a matter of about 1000-2000 ms if no user interaction is required.

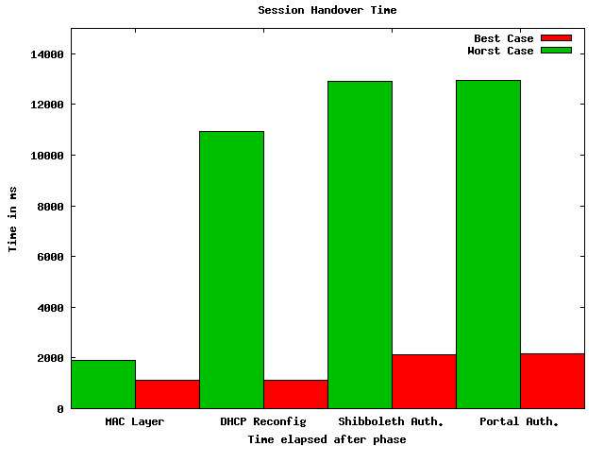

Fig. 4. Total handover time during the different phases. Worst and best case

Finally the portals re-authorization is done based on the attributes provided by Shibboleth but this can be neglected in relation to the other procedures because it will take place in under 20 ms using “normal” hardware.

These measurements (Figure 4) clearly show that the main time consumers of such a session handover are the first three procedures. They also heavily depend on the network topology, the used operating systems and different software. The conclusion is that the whole session handover procedure might be very slow and its time consumption is given by the used protocols.

REFERENCES

[1] SWITCH. SWITCHmobile. [Online]. Available: http://www.switch.ch/mobile/

[2] SWITCH. SWITCH Authentication and Authorization Infrastructure. [Online]. Available: http://www.switch.ch/aai/

[3] INTERNET2. The Shibboleth Project. [Online]. Available: http://shibboleth.internet2.edu/

[4] R. L. Morgan, S. Cantor, S. Carmody, W. Hoehn, K. Klingenstein: Federated Security: The Shibboleth Approach. EDUCAUSE Quarterly, Vol. 27, No. 4 (2004)

[5] OASIS. Assertions and Protocol for the OASIS Assertion Markup Language (SAML). Committee Specifi cation 01. 2002.

[6] SWITCH VHO. SWITCH Virtual Home Organization. [Online]. Available: http://www.switch.ch/aai/vho/

[7] Pubcookie. Open-source software for intra-institutional web authentication. [Online]. Available: http://www.pubcookie.org/

[8] Héctor Velayos, Gunnar Karlsson: Techniques to Reduce IEEE 802.11b MAC Layer Handover Time. KTH Technical Report TRITA-IMIT-LCN R 03:02 (2003), ISSN 1651-7717

[9] M. A. Steinemann, T. Braun: A generic broker portal linking authentication and authorization infrastructures and resources. European Journal of Open and Distance Learning (EURODL) (2004), ISSN 1027-5207.

[10] TERENA. Trans-European Research and Education Networking Association. [Online]. Available: http://www.terena.nl/tech/task-forces/tf-mobility/