

Case Study - An Open Source solution for network user authentication

Manuel ŞUBREDU, Octavian RUSU, Valeriu VRACIU and Florin Manolache¹

Abstract - In the last years the number of online services provided to their users by educational institutions and companies have greatly increased. Typically each such service comes with its own separate authentication mechanism. Checking and enforcing security policies, as well as password management in such environments is tricky for users and system administrators. This paper shows our solution to implement a Single User Single Password solution that can be applied for almost all network services.

Keywords — ldap, authentication, 802.1x, radius

I. INTRODUCTION

Over the last years, the number of online services provided by educational institutions had greatly increased. Since these services have different authentication methods and mechanisms, enforcing the security policies and a good password management is a cumbersome task, for both users and technical staff alike. We propose a Single User Single Password solution, based on OpenSource software, that can be implemented in environments that are friendly to open standards. The main advantages of our solution are:

- ⑩ security
- ⑩ strong authorization
- ⑩ interoperability (with eduroam)

The solution proposed by us, is currently used for services on the Romanian Education Network and the Alexandru Ioan Cuza University networks. Our solution uses, OpenSource software, open standards and scripts developed by us to achieve its purpose. The main software components are:

¹ Manuel ŞUBREDU is a network engineer at RoEduNet Iasi Branch and a programmer at the Digital Communications Department from the Alexandru Ioan Cuza University, Iasi, Romania (phone: 40/232/201007; fax: 40/232/201200; e-mail: manuel.subredu@roedu.net).

Octavian RUSU is the CEO of RoEduNet and Digital Communications Department from the Alexandru Ioan Cuza University, Iasi, Romania (e-mail: octavian@roedu.net).

Valeriu VRACIU is a network engineer at RoEduNet Iasi Branch and at the Digital Communications Department from the Alexandru Ioan Cuza University, Iasi, Romania (e-mail: valeriu.vraciu@roedu.net).

Florin B. Manolache is with the Mellon College of Science, Carnegie Mellon University, Pittsburgh, PA, USA (e-mail: florin@cmu.edu)

- ⑩ OpenLDAP
- ⑩ FreeRadius
- ⑩ BIND
- ⑩ OpenCA
- ⑩ Apache
- ⑩ PERL

The hardware used by us includes Cisco network equipments and Intel servers. At this moment our solution is applied to:

- ⑩ email accounts on GNU/Linux servers (OpenSuSE 10.x);
- ⑩ authentication on internal services;
- ⑩ wireless secure authentication.

II. MAIN COMPONENTS

A. OpenLDAP

OpenLDAP¹ is the central piece of our solution. It stores all the information associated with the accounts. We opted to use the Berkeley Database as OpenLDAP backend.

Since OpenLDAP is the mechanism of choice for the network services, performance is mandatory. To be able to scale our solution as needed, we created a hierarchical network of OpenLDAP servers with one master and a number of slaves. The master is optimized for writing and content replication, and the slaves are optimized for reading the OpenLDAP database. All the modifications (add, delete and modify operations) are made by the master and all reading requests are handled by the slaves. The slave selection is made using DNS RR algorithm.

To take advantage of OpenLDAP hierarchical network, each department should have at least one OpenLDAP slave¹¹.

B. OpenLDAP – Custom schema

Since we use OpenLDAP in as many online services as possible, we had to create a custom schema to be able to store within OpenLDAP all the information required. For example our custom schema includes employe ID, fields who defines if a user has the right to access online pbx application, etc.

The first step into creating a custom schema, is to obtain a ldap ID from IANA. That ID will be the root of all schema related information. In our case the root ID is

1.3.6.1.4.1.20816¹⁰.

We mapped our definitions using .10.1 (attributes) and .10.2 (objects) for better schema management. Example:

```
objectidentifier    uaic 1.3.6.1.4.1.20816
objectidentifier    uaic.Definitii uaic:10
objectidentifier    uaic.Definitii.Atribut uaic:10.1
objectidentifier    uaic.Definitii.Obiecte uaic:10.2
```

To accommodate LDAP into our organisation, we created multiple ldap classes. We have classes for accounts, people, servers and network access (both wired and wireless). Each class contains multiple required and optional attributes.

At this moment, an account can be used for:

- ⑩ shell access on the mail server
- ⑩ email
- ⑩ wireless network access
- ⑩ access to web based network management software
- ⑩ access to internal services

Our schema contains information that can store both AAA (access, authentication and authorization) and general user information like parents information, highschool grades, employee data, user picture, user certificate, user GPG key, etc. This way, our LDAP database is designed to be easily used by other departments in our institution.

C. FreeRadius

FreeRadius² is part of the authentication process. Using Radius, the network equipments authenticate an account and allow access to certain resources (eg: vlan membership).

D. OpenCA

OpenCA⁴ is the only real OpenSource CA software who can be used to set up a custom Certification Authority (CA). Certificates are important in our solution since the communication between the client (radius) and the server (ldap) is encrypted to provide sniffing protection. Since we require our users to install SecureW2 in order to gain access to wireless network, it is a great opportunity to distribute (and install) our CA public certificate to our users.

E. Custom scripts

Since the customization plays a major role in our solution, we had to develop a set of perl scripts to :

- ⑩ add users to LDAP directory;
- ⑩ reset and change user passwords;
- ⑩ perform account management via cgi/web based utilities;
- ⑩ convert users from passwd/shadow to LDAP users;
- ⑩ maintain the consistency of the configuration files across machines

All the perl scripts, use Net::LDAP and CGI CPAN modules to ensure greater portability. At this moment, we provide the technical details of our solution (configuration

files and source files of custom scripts included), only to trusted partners.

III. HOW IT WORKS

A. wireless

To better understand how our solution works, we will present its implementation on the wireless network.

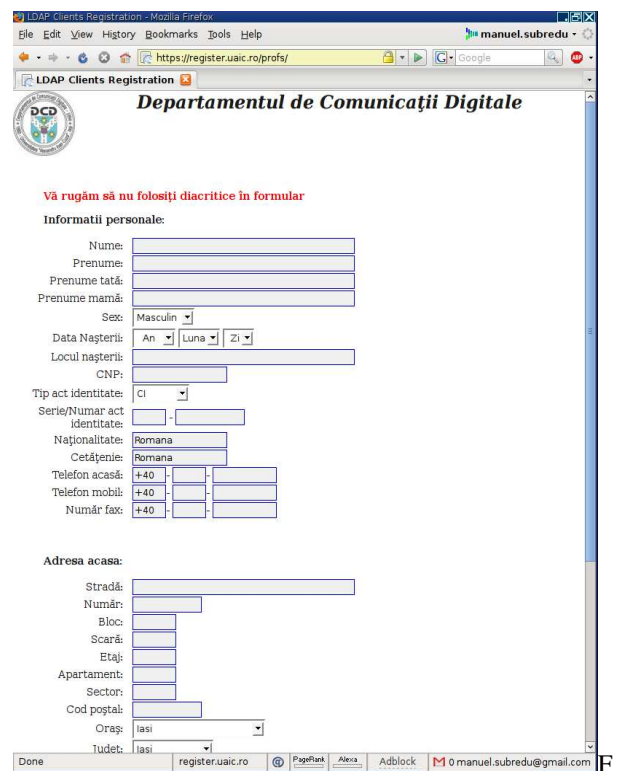
When a computer scans for available wireless access points, a single open network named “uaic” is available. This network is not secured and can be accessed without a password.

After the computer connects to the uaic wireless network, it receives the network settings through dhcp.

B. the registration

Using a BIND³ feature called views, we resolve any dns request into a ip address allocated to the registration server. This way, the user will be redirected to the registration page, no matter where he tries to go on the web.

The registration website⁵, contains all the necessary and relevant information for the user (like security policies, rules and regulations, etc).



ig. 1 – registration form for teachers

After the user has been through all the required documentation, he completes a registration form, in which all the required information is filled in. When the form is submitted, two emails are generated: one goes to the support team informing them that a new user has been created and, the other goes to the trouble ticket system, for future reference.

Before the account becomes valid, some information is verified by our support team (like personal numeric code,

etc). If the information is correct, the account is activated.

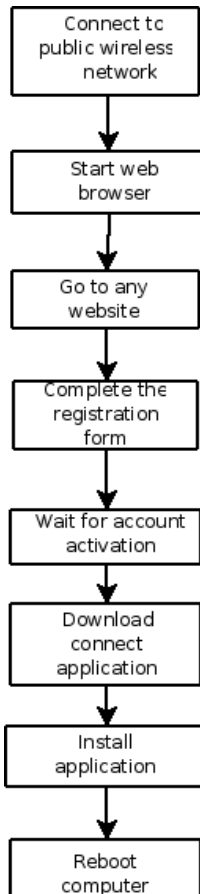
The user, can check at any time if his account has been activated, using a form available on the registration website. Usually, the accounts are activated within the same business day, if the accounts were created in the working hours.

C. using the network

After the account has been activated, the user has to download a program and to install it on his computer. The program is SecureW2¹² with custom configuration. The configuration includes wireless network parameters like: SSID, encryption type, preference, etc. During the installation, the user is asked about a username and password. After this point, the user can connect to the wireless network (and use it), at any time.

The same credentials, will be used to access the students email service, the VoIP service, and any other services that will be provided.

A scheme of the registration process (as viewed by the user) is shown below:



D. Some numbers

At this moment, our OpenLDAP directory, has around 1300 staff accounts (faculty and technical staff) and around 100 students accounts, with 10 to 15 new students registering on a daily basis. All students accounts are used to access the wireless network available in two student hostels.

The staff accounts, are used on the official mail server.

The ldap replicas (who respond to authentication requests coming from email server), receive around 8-9 requestes per second (this means more than 500kreq per day).

E. Configuration of the wireless routers

As mentioned before, we are using wireless routers to offer to our users wireless connectivity in a secure environment. Here is (a sample) how our equipment is configured:

```

dot11 vlan-name wlstud vlan 750
dot11 ssid wlstud
    vlan 750
    authentication open eap rad_auth
    authentication network-eap rad_auth
    authentication key-management wpa
    accounting default
!
interface Dot11Radio0
    no ip address
    no ip route-cache
    load-interval 60
    encryption vlan 750 mode ciphers tkip
    broadcast-key vlan 750 change 600 membership-
termination capability-change
    ssid wlstud
interface Dot11Radio0.750
    encapsulation dot1Q 750
    no ip route-cache
    no snmp trap link-status
    bridge-group 75
    bridge-group 75 subscriber-loop-control
    bridge-group 75 block-unknown-source
    no bridge-group 75 source-learning
    no bridge-group 75 unicast-flooding
    bridge-group 75 spanning-disabled
!
interface FastEthernet0.750
    encapsulation dot1Q 750
    no ip route-cache
    no snmp trap link-status
    bridge-group 75
    no bridge-group 75 source-learning
    bridge-group 75 spanning-disabled
!
  
```

IV. FUTURE WORK

As we speak, we try to implement 802.1x⁸ authentication⁷ on our wireless network, using two student hostels as testbed. If the tests are successful, we plan to expand the wireless network on the other buildings of our institution.

With the expansion of wireless network and the growth of the number of people registered into our LDAP server, we are going to offer email service, to all students (aprox 70000 accounts).

Long term plans, include 802.1x AAA on the wired network and migration of all AAA services (VoIP included) to LDAP.

V. CONCLUSION

This case study, illustrates the implementation of a solution that allows an user to have a single account name and password to authenticate on all the services provided by a institution. This solution, can be implemented using only OpenSource software and open standards.

The list of applications that can use the authentication solution proposed here includes any service that needs authentication.

REFERENCES

[1] OpenLDAP: <http://www.openldap.org/>

[2] FreeRadius: <http://www.freeradius.org/>

[3] BIND: <http://www.isc.org/sw/bind/index.php>

[4] OpenCA: <http://www.openca.org/>

[5] Apache: <http://www.apache.org/>

[6] IEEE 802.1x: <http://www.ieee802.org/1/pages/802.1x.html>

[7] 802.1x overview: <http://en.wikipedia.org/wiki/802.1x>

[8] Extensible Authentication Protocol (EAP) :
<http://tools.ietf.org/html/rfc3748>

[9] Gerald Carter, LDAP system administration, O'Reilly 2003, ISBN 13: 9781565924918

[10] OpenLDAP Admin Guide:
<http://www.openldap.org/doc/admin23/>

[11] SecureW2: <http://www.securew2.com/>