Security Information Architecture for Automation and Control Networks

Eduardo Feitosa[1,2], Luis Oliveira[1], Bruno Lins[1], Ademir Junior[1], Rodrigo Melo[1], Djamel Sadok[1], Ubiratan Carmo[3]

[1]Centro de Informática
Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851 – Cidade Universitária - Recife - PE

[2]Departamento de Ciência da Computação
Universidade Federal do Amazonas (UFAM)
CEP 69077-000 Campus Universitário - Manaus - AM

[3]Companhia Hidroelétrica do Rio São Francisco (CHESF)
Rua Delmiro Gouveia, 333 - Recife - PE

{efeitosa,eduardo,bruno,ademir,rodrigodma,jamel}@gprt.ufpe.br,
uacarmo@chesf.gov.br

Abstract. *Ongoing automation and the open access implementation of critical systems are increasing security vulnerability in automation and control networks employed in the electric power sector. This work is of the view that the integration between policy-based management and access control mechanisms is necessary to take us closer to a more effective solution for the combat against security threats. This paper introduces a modular architecture based on the XACML framework and applies this to automation and control networks used in the electric power industry. This security architecture is described and its components are individually analyzed and tested in a real power production environment. This paper also describes relevant encountered difficulties, provides results and new insights into access control when applied to industrial critical network infrastructures.*

Resumo. O atual nível de automação e o livre acesso as implementações de sistemas críticos estão aumentando as vulnerabilidades de segurança em redes de automação e controle empregadas no sector de energia elétrica. Este trabalho é da opinião que a integração entre o gerenciamento baseado em políticas e mecanismos de controle de acesso caminha lado a lado na busca por uma solução mais eficaz na luta contra as ameaças de segurança. Este artigo introduz uma arquitetura modular baseada em XACML e que é aplicada a redes de automação e controle utilizadas na indústria de energia elétrica. Essa arquitetura de segurança é descrita e seus componentes são analisados individualmente e testados em um ambiente real de produção. Este artigo também descreve as dificuldades encontradas, resultados e fornece novas perspectivas sobre controle de acesso quando aplicada a redes de infra-estruturas industriais críticas.

1. Introduction

For several years, the isolation induced by the diversity of equipments, protocols, and proprietary systems kept the network infrastructure of critical industries (electric power, gas and oil, public transport, etc.) relatively safer. A potential attacker would have to understand all this diversity to gain access to restrict data hence such isolation had its benefits. However, the introduction of competition, effect of the globalization, in these industries resulted in a big pressure to improve their financial performance, to provide a better assistance to the productive process, and to reduce operational costs.

In order to achieve these “new targets”, critical industries have invested heavily in the automation of their installations, integration of operations through SCADA (Supervisory Control And Data Acquisition) systems, and network infrastructures integration using TCP/IP protocol suite. Nevertheless, all of this “modernization” comes at a price. New security problems in the ambit of critical network infrastructure, previously unheard of, popped up mainly due to the adoption of “well-know” technologies.

De facto protocol stacks such as the Ethernet and TCP/IP are notoriously known for the lack of any security driven design. In addition, there aren’t any recommendations or standards to define the degree of reliability and availability of SCADA systems. Moreover, critical networks are different in their nature. They have been designed to transport critical information control of productive processes and not data information such as e-mail and web pages. Information confidentiality, availability, and integrity of are seen as primordial. To finish, it is important to emphasize that the existence of vulnerabilities and security flaws in critical infrastructure networks can result in a literally dark scenario like, for example, blackouts.

The present work addresses the lack of well designed security architecture for SCADA-based control networks that operate via open standards (e.g., TCP/IP). We propose SIRCAM, a security modular architecture for automation and control networks, designed and applied in electric power sector.

The rest of this paper is organized as follows. Section 2 describes some activities developed by electric power sector to expand the security in critical networks. In section 3 we present an overview about SIRCAM architecture. Section 4 describes each of the components and reveals the interactions between them. Section 5 presents the SIRCAM implementation of main components. In section 6 are discussed the initial results of our architecture. Lastly, in section 7, the conclusions and some suggestions for future works in this area are presented.

2. Background

After September 11, 2001, the issues related with the security and reliability of the electric infrastructure became more relevant. Thenceforth, electric power associations, standardization institutes, and committees have proposed a set of standards and devices to deal with security issues, specifically for the control and automation network area. The IEEE Power System Relay Committee published the report Cyber Security Issues for Protective Relay 0, which focuses on electronic security in information exchanges between protection relays. The IEC (International Electrotechnical Commission)

published technical specifications (IEC 62351-6) 0, which dealt with security in the new automation protocol IEC-61850[1] [IEC 2002]. The Cigré (International Council on Large Electric Systems) created a Joint Work Group 0 to identify and define information security issues for the electric power industry, such as main domains for Utility Information Systems and Telecontrol (control center and substations).

Although there are already standards, devices, and software [He 2005] to address security issues, these initiatives in networks of automation and control for critical systems are still timid. For instance, there aren't any recommendations or standards to define the degree of reliability and availability of SCADA/EMS (Energy Management System) systems. In Brazil, this issue is seen as a secondary one for companies working with the production and transmission of electric energy.

Other initiatives focused in safety, reliability and availability of critical infrastructures are CRUTIAL and TCIP. CRUTIAL project [CRUTIAL 2008] aims to assist on the development of new electric power grid, which components will be interconnected with the corporate networks (intranets). TCIP (Trustworthy Cyber Infrastructure for the Power Grid) [ITI 2008] is a project focused on securing the low-level devices, communications, and data systems in Electric power grid systems. This project is divided in different lines such as the design of new functionalities in hardware to detect attacks and failures, new techniques to detect, react, and recover failures caused by cyber attacks, new authentication and authorization techniques, and so on.

3. SIRCAM Overview

A careful and deep examination of the current infrastructure networks of electrical power sector reveals the complexity and difficulty in to design, develop, and implant security solutions.

First, there are different automation levels of the installations. For instance, while there are modern electrical substations fully digitized, others have a single bay retrofitted to Programmable Logic Controller (PLC). Second, normally the security policies are set up to protect against external threats by firewalls, gateways, and filters. Access control schemes and protections against domestic incidents are whether "delegated" to the end users or simply not exist. As result, internal security incidents are common and frequent. Third, any proposed security solution must cover all existent infrastructures and interact with other networks and services such as management, measurement and billing.

To embrace all these peculiarities, our approach is based in three aspects. First, we consider that there are "**automation islands**" in different evolutionary stages and must receive differentiated treatment of the security aspects when applied to people, processes, and equipments. Control centers, substations, relay room, and automation and protection equipments are examples of automation islands. In addition, these islands use TCP/IP protocols.

Second, to minimize internal security incidents our approach employs the concept of Policy-Based Management (PBM) to provide access and admission control.

[1] IEC 61850 is a standard for the design of substation automation.

Basically, PBM methodology use policy rules to manage the configuration and behavior of one or more entities and even entire networks 0. In other words, it allows monitoring, controlling, and enforcing the adequate use of network resources and services where the expected result may be: user admission, transfer to other subnet, or access being denied. PBM methodology provides better security with the proliferation of the number of users and applications, management of device, traffic, and services complexity, simplifies the implementation of time-critical network functions, and handles traffic more intelligently. Furthermore, policies are text-based declaratives, i.e., this permits that they can be interpreted and adapted at run-time, increasing the ability to capture all behaviors of the managed system.

Lastly, to keep the interoperability with the legacy network infrastructure, after a series of interviews with engineers, managers, and SCADA/EMS operators of the CHESF Company and to evaluate performance and compatibility issues, we adopt the following technologies:

- RADIUS (Remote Authentication Dial-In User Service) Protocol [Rigney 2000] as a framework to connect the hosts with access devices;
- IEEE 802.1X [IEEE 2001] to implement access control based on physical access device ports;
- EAP (Extensible Authentication Protocol) [Aboba 2004], a universal framework for authentication used in order to provide common functions and schemes for authentication through EAP methods.
- IPSec [Kent 1998] to permit a pair of communicating entities (host-to-host or host-to-service), inside or outside of the automation island, to use whichever algorithms to provide the security services appropriate for such communication.
- XACML (eXtensible Access Control Markup Language) [OASIS 2005], an XML-based standard, as a policy language and an access control decision and response language. It is designed to provide a universal language for authorization policy to enable interoperability with a wide range of administrative and authorization tools.

4. SIRCAM Architecture

The SIRCAM architecture was designed to be modular, to ease future modification, and the seamless addition of new components. In spite of designed and applied in electric power sector, it can be implanted in most different environments and this must be seen as one of the advantages of our architecture.

As shown in Figure 1, the architectural components were broken down in accordance with their role. The main entities that compose the SIRCAM architecture are:

- **Agent** – represents the system entity responsible for aggregating and managing all functions related to access control for end hosts, HMI (Human Machine Interface), operators, and end users in automation island networks. Due to the diversity of equipments and operational systems, the agent must be adaptable.

- **Access Device** – represents the system entity responsible for access and traffic control. It makes access requests and enforces the decisions taken by the access control service. Manageable switches and access points are examples. In this context, an access device corresponds to an IETF Policy Enforcement Point (PEP), defined in RFC 3198 0.
- **Policy and Access Control Service (PACS)** – represents the main entity of the SIRCAM architecture and is seen as the heart and brain of the project. It is responsible for evaluating all access requests in accordance with the applicable policies and for responding with an authorization decision (permit or deny). This entity is based on the IETF Policy Decision Point (PDP), also defined in RFC 3198.
- **Remediation Service –** represents the system entity responsible for keeping an end host working in accordance with the defined policies. At first, a remediation service can be seen as an additional and specific service capable to communicate with all agents to maintain them capable of working properly. Examples of the controls performed by a remediation service are anti-virus version control and updating. More information about remediation can be found in references 0 and 0.

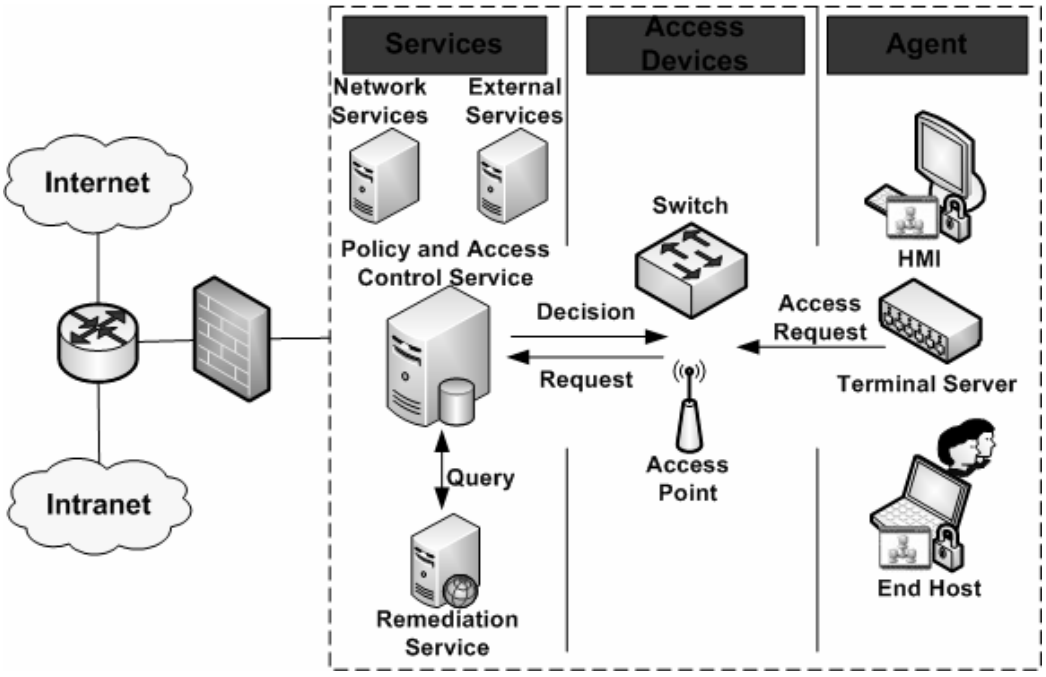

Figure 1. SIRCAM Architecture Overview.

In addition, the SIRCAM architecture interacts with other services or systems such as:

- **Network Services** – represents all services necessary for network operation. We can cite DHCP, DNS, and HTTP.
- **Data Repository** – represents the database that contains hosts and user information.

- **Policy Repository** – represents the database that contains policy rules, actions, conditions, and related data.
- **External Services** - represents others services outside the security context but that are essentials to the operation of the network and systems. As an example, we mention the SAGE (Sistema Aberto de Gerenciamento de Energia) system 0, a Brazilian open source software responsible for managing electric power all the way from the substations to the control centers.

4.1 Agent

Considering a great number of security incidents arising from inside of the installations (IP address misconfiguration, login and password bad utilization, virus, for example), we believe that a rigid access control cannot cease such problems but considerably reduce the number of incidents. This way, the SIRCAM architecture presents an agent to deal with all the access control process (authentication and authorization) of hosts and also users. It was designed to set up and manage access control in existent automation island equipments (end hosts, terminal services, and HMI). It can be seen as a front-end of the architecture since it represents the access gate to the architecture's services and resources.

Architecturally speaking, the SIRCAM agent is divided in two modules: ChesfSupplicant and VisualSupplicant. The former module runs continually when the host is active, i.e., it is implemented as a daemon or service. It is responsible for managing host and user access control, the remediation process, and any other function that involves hosts or users. The last provides a friendly graphical user interface (GUI) to allow operators and end users to gain access to all available services in ChesfSupplicant. It permits user authentication provided by ChesfSupplicant, but only operational through it.

4.2 Access devices

Access devices are used to protect the access to one or more resources and to enforce the decisions taken by the architecture, i.e., by PACS element. Within the SIRCAM architecture, a host must identify itself to the access device associated to it before obtaining access to network resources. Then, the access device consults the PACS to know whether this particular host is allowed to join the network, what services it is able to use, and what controls will be applied over it.

Basically, access devices receive only two types of decisions: permit or deny. The deny decision prohibits the host to access any network resource. In this case, the host is not a network member. The permit decision can results in two situations. In typical case, the access request is fully accepted. This means that the host information was approved. In other situation, the access request is partially validated by PACS. This means that a remediation process will be started to perform the necessary software updates on this host. Often, this is achieved through the establishment of a specific VLAN between host and remediation service. A new host validation will be necessary when the remediation process is over.

Access devices are seen as "men of the battlefield" since they enforce security policies and ensure that each host attached to their network is compliant, protected and

safe to a degree. It is important to highlight that the access devices are network devices such as switches and wireless access points, and their functionality will not be altered in this work.

4.3 Policy and Access Control Service (PACS)

The core of the SIRCAM architecture is its Policy and Access Control Service (PACS). PACS has as main function to take independent access control decisions based on policies usage, i.e., it verifies the credentials of hosts and users, checking if they are able to access a particular network resource according to predefined policies and allows or denies access to these resources. Moreover, it should maintain control over the status of authenticated and connected hosts and users.

Examples of other actions taken by a PACS include:

- Locating and evaluating a set of rules that is applicable to a given host being managed by PACS.
- Keeping a register about which hosts are associated to what access devices and which users are associated to what hosts (e.g. for the session control).
- Control the establishment of secure communication between agents (hosts) and services inside or outside of automation islands.
- Enforcement of new policies based on information sent by, for example, the remediation service.
- Keeping a policy database updated.
- Resolving policy conflicts when possible, and reporting them to a policy management console and administrators.

Furthermore, PACS can also be responsible for coordinating the execution of other functions/services such as bandwidth management, event logging, remediation actions, firewall setup rules, quarantining, and so on.

PACS architecture

The internal PACS architecture basically consists of two modules: Context Handler and Policy Decision Engine (PDE). PACS Context Handler intermediates access requests from hosts and PDE. Basically, it acts as a translator, receiving access requests in different formats according to the access device where it is connected (typically RADIUS) and converts them to policy language requests. Next, it sends access requests to PDE and waits for a response.

PDE module is the brain of PACS. It evaluates the policies stored in the policy repository, determines which policies are applicable, and returns a response (decision) to access device associated to the received request. When a request is sent to the PDE, host identity, user information, access device information, conformity status, certificates, and other possible data are extracted from the request message to carry out verifications. First, the PDE needs to check whether a given host or user exists in its data repository. If not found, PDE formulates a negative response and sends it to the Control Handler to inform the access device. Otherwise, the PDE searches its policy repository to find policy or set of policies applicable to the requester. If at least one is found, a decision is

chosen and sent to the Context Handler. It is important to highlight that only the PDE communicates with the data and policy repository to prevent fraud and illegal access to the network.

4.4 Remediation Service

The main goal of this component is to keep hosts "clean" and in accordance with the established policies. This service can be seen as a database to store antivirus, anti-spyware, operational systems patches, etc. It can and must be viewed like a specific system that permits a host always to stay updated. At CHESF, due to operational restrictions the remediation service is used to alert the operators and user about SAGE system updates. In the SIRCAM architecture, the remediation service acts jointly with PACS during the access control process or where there is need to update some software.

4.5 Access Control Process (Interaction between components)

To help the reader gain familiarity with the SIRCAM architecture, we give next information on the interaction among these components during the access control process both for hosts and users. Figure 2 exemplifies the host access control process and Figure 3 show the user access control process.

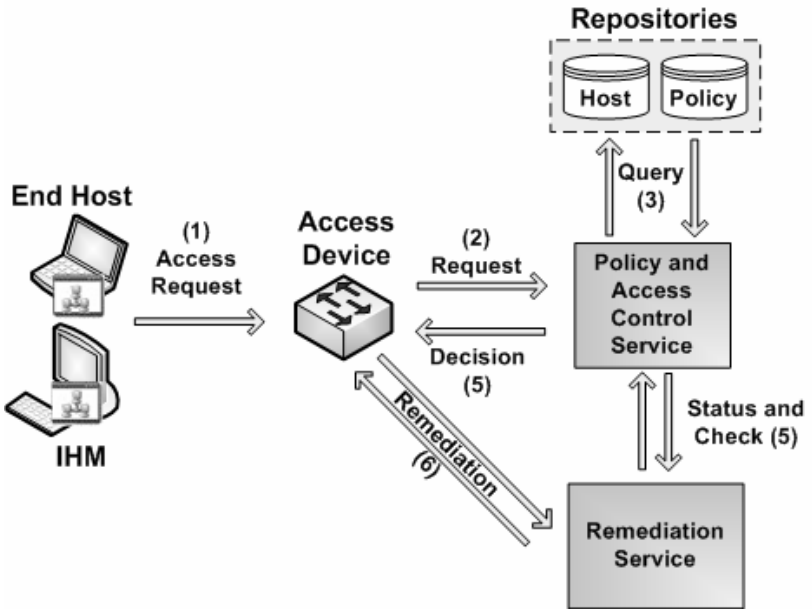

Figure 2. Host Access Control process.

In order for a host to obtain network access (Figure 2) it must identify itself to the target system, via the access device where it is connected (1). This initial step involves message exchanges between agent and access devices. So, the access device creates access request containing host information and self identities and forwards (2) it to the policy and access control service (PACS) to evaluate whether the host has access rights and what it can do. PACS then analyzes all contained information in the request (3) via queries in repositories (hosts and policies) and through conformity status validation in the remediation service (4). Once verified, PACS returns to the access device a decision containing the action to be executed (5). When access is allowed, there may be two possible actions: to permit the host ingress on network or to trigger

remediation activities of any host software (6). For this, a special VLAN is established between the host and the remediation service.

The user authentication process (Figure 3) is simpler but requires the prior host authentication and does not involve directly access devices and the remediation service. Thus, via the agent, the user sends its identities (username, password, and credentials) (1) directly to PACS. PACS analyzes all contained information in the request (2) via queries in repositories (users and policies). After analyzed, PACS returns to the user (agent) a decision containing the action to be executed (3).

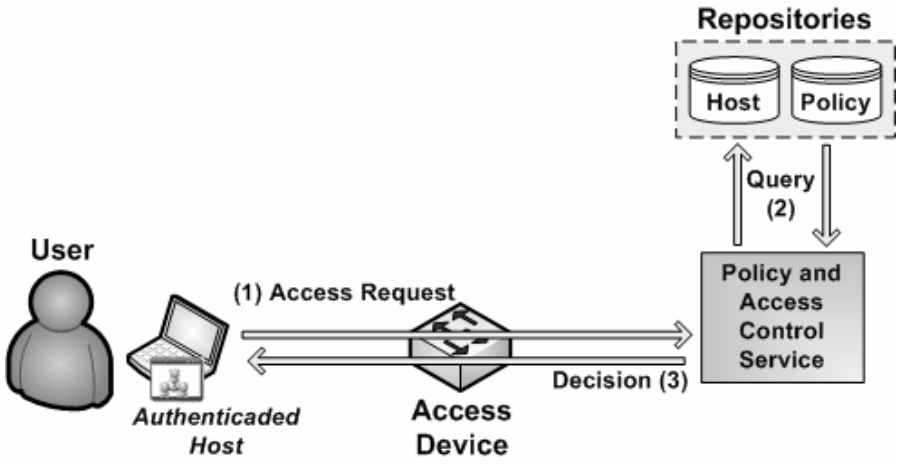

Figure 3. User Access Control process.

5. Implementation

An initial SIRCAM prototype has been developed. We are using Java (version 1.6 or superior) and SUN XACML Java API 0. We also make use of:

- XSupplicant 0, an open source tool that implement 802.1X supplicant functions. ChesfSupplicant agent applies control over XSupplicant to negotiate host authentication with access devices using IEEE 802.1x and EAP protocols.
- FreeRADIUS 0, an open source RADIUS server that offer communication between access devices and PACS, i.e., it receives hosts access requests and forwards them to PACS;
- JRadius 0, an open-source Java API to intermediate access control negotiation between FreeRADIUS and the PACS Context Handler. In other words, it makes available an interface for that RADIUS access request for analysis by PACS;
- OpenLDAP 0 software as a lightweight directory service (LDAP directory) to implement host, user and policy repositories.

To integrate all components using policy-based management concept and to attend all CHESF requirements, we extend some functions in XACML API. First, the XACML communication model only permits synchronous messages (request and response communication). However, in some situations the interaction between SIRCAM components must be done by asynchronous messages (notifications). XACML notifications are necessary because PACS is reactive and is necessary make decisions even if no requests are received, taking on a proactive behavior. For instance, agents can

receive notifications that were disconnected by the human operator. Second, the XACML combining algorithms existent are not appropriate to deal with different levels of priority. Then, we defined a new priority-based policy-combining algorithm. Third, we implemented a LDAP schema that supports XACML policies based on the LDAP Profile for Distribution of XACML policies draft 0. Lastly, we also created a simple management tool that converts the abstract policy into entries that can be populated into an LDAP Directory. This tool was incorporated to PACS.

5.1 SIRCAM Implementation

Agents

As explained previously, the VisualSupplicant agent depends of ChesfSupplicant agent to exhibit for users all implemented functions. This way, the communication between them uses one specific interface implemented via sockets. The message exchange includes user access request (authentication), user logout, set up of connection parameters, host disconnection request, and notifications such as the keep-alive message and ChesfSupplicant disconnection. All communication between agents is encrypted. Except for ChesfSupplicant, two socket interfaces are provided by PACS console: one is used to send XACML request-response (user access control) and the other one is used to carry XACML notifications such as for remediation.

PACS

To keep session control of all elements, PACS implement two hash tables (hosts and users). In host hash table, each element is composed by IP address, hostname, IP address of device access where the host is connected, physical port of device access, and date. For user hash table, only login, hostname where the user is connected, and date are stored. All these information are obtained during authentication and authorization process.

Regarding communication with other architectural components, PACS use three specific interfaces implemented through transport sockets: one of them is used to wait for access requests-response, the other one to wait for XACML notifications, and the last one for send XACML notifications to the agents. All communications are encrypted.

With regard to PACS Console modus operandi, basically there are following the four steps:

1. The Context Handler module active JRadius interface establishing communication with FreeRADIUS server.
2. The PDE module is initialized.
3. After, all repository data is loaded in memory. Policy rules are instanced in the “AbstractPolicy” element of the XACML API. Host and user data are allocated in hash tables. This procedure permits to considerably increase the console’s performance.
4. Lastly, the PACS Console interface is activated.

6. Evaluation and Results

Due to the nature of SIRCAM architecture, its evaluation is somewhat subjective and non-trivial since that the final result is basically permit or deny hosts and users to access network resources. Depending of the amount of policies used, access type, and network traffic, the access control performance can varies. This section presents SIRCAM experiments and some validation results.

6.1 Testbed environment

To cover the variety of scenarios and to fairly evaluate the robustness of this architecture, we set up an elaborated confined testbed consisting of real machines within our laboratory. The idea was to create a controllable environment that resembles as much as possible a realistic network topology. Our testbed contains 16 PCs (enumerated from 1 to 16), 2 servers (Athlon XP 4200+ 64bits processor, 2 Gb of RAM, and 160 Gb of HDD), 1 switch with 24 10/100 Mbps interfaces and two Giga uplink interfaces, and 1 wireless access point 802.11b/g. The PC nodes running some Linux distribution (Fedora, CentOS, Debian, and Gentoo, for example) and they have installed ChesfSupplicant and VisualSupplicant agents. Each server runs Fedora operational system and contains the PACS server, the LDAP directory and the FreeRADIUS server. In CHESF environment, our tests were in CROL (Centro Regional de Operação do Sistema Leste) with the similar infrastructure, except that the hosts and servers are SUN UltraSparc IIIi, 8 GB of RAM, and 830 Gb of HDD.

In our experiments, the switch was set up to support three VLAN (Virtual Local Area Network). This way, we ensure that device restrictions and policy decisions can be applied correctly.

6.2 Performance and Scalability results

In order to evaluate the performance and scalability of our architecture, we use traditional metrics of software evaluation such as memory and CPU consumption. However, our prototype presents some peculiarities due to the restraints and performance and scalability aspects. This way, is necessary measure, for instance, the time spent to load all policies in memory and the spent time to PACS evaluate simultaneous requests.

6.2.1 PACS analyses

To evaluate the time spent by PACS to load all policies in memory[2], we populate the LDAP repository with 15600 hosts (originated from a mathematical permutation without repetitions among all alphabet letters in groups of 3 elements - 26*25*24), 15600 users, and 31200 policy rules entries (one for each host and user), making a total of 62400 entries. In our testbed, the passed time was approximately 17.2 seconds. Taking advantage of this massive data in memory, we observe that they occupy 23% of total, i.e., 471 Mb of a total of 2 Gb. Moreover, PACS in operation utilizes 7.5% of memory (155 Mb).

[2] This procedure involve access LDAP repository, carry out all policies, translating them to XACML object, and store this object in memory. It was tested and is more efficient than making one query at a time on LDAP repository.

In addition, we also verify the percentage of memory occupied by each element (host and user) active in SIRCAM architecture. To obtain this measure, we first observe how each element can occupy in hash tables. For host, one element may use between 65 and 82 bytes of memory. For user, the memory occupation varies from 28 to 55 bytes. Assuming the worse case, where all elements use maximum available size. Next, we suppose a scenario where all 15600 hosts and 15600 users are connected (authenticated and authorized). In this case, the memory occupation by host hash table achieves approximately 1.25 Mb (82 bytes * 15600 elements) and by user hash table achieves approximately 838 Kb (55 bytes * 15600 users). So, accounting all estimated usage of memory, we consider that our prototype is very scalable due to the fact that to deal with 31200 hosts and users, simultaneously connected, it fill approximately 628 Mb of memory (471 Mb of the loaded XACML objects, 155 Mb of operational usage, and 1.25 Mb and 0.83 Mb of hash tables) and that in CHESF environment, the number of hosts and users working not surpass 250 by automation island. Moreover, as this architecture is focused in critical server environments, the amount of spent memory to run PACS application does not represent a significative quantity.

In relation to CPU, we observe that the PACS reaches approximately 88% of CPU usage during start process where it executes four start stages as seen previously. After this it falls down dizzily for 3%.

However, also is necessary to evaluate the scalability regarding CPU consumption and the spent time to PACS to response simultaneous access request. For this, we generate 12, 25, 50, and 100 concurrent requests. Figure 4 illustrates the result.

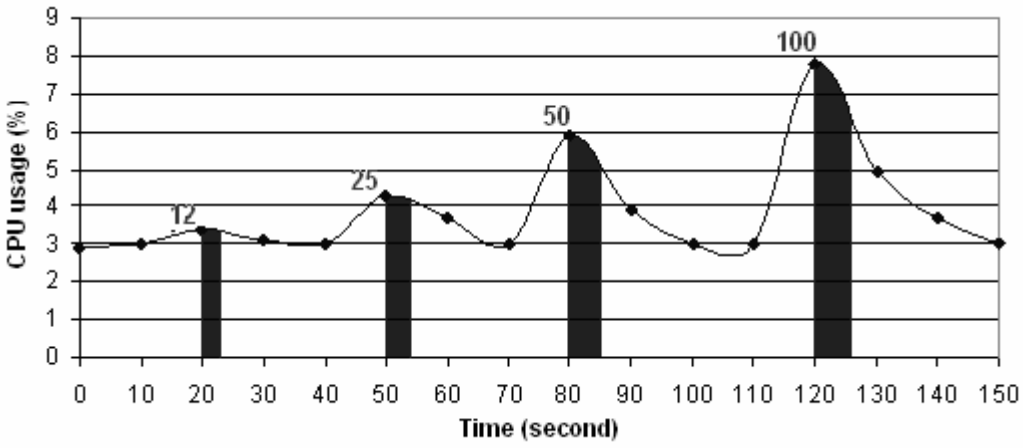

Figura 4. CPU and Time response percentual.

We utilize 12 synchronized access requests and measured all message exchange since first host sent packet until the last decision to be received. The time consumed was exactly 2.3 seconds. Using 25, 50, and 100 synchronized access requests, the time to process and response was 3.7, 5.0, and 6.7 seconds. It is necessary to explain that these values correspond to experiments without access device interference (any activity related with action or enforcement of policies are executed).

In relation to CPU usage, figure 3 reveal the quickly increase after receive the first requests. However, its processing also is very fast and, for this motive, we consider

adequate and effective to work in critical industrial environments. It is important emphasize that all policies and host and user date are loaded to memory.

6.2.2 Agents analyses

In relation to memory and CPU of agents, we observe that ChesfSupplicant agent utilizes 5.5% of CPU when starting and less than 0.7% after this. The memory usage is inferior to 2% during all operation, even when it receives different requests and notifications. VisualSupplicant agent always uses around 2.4% of memory and its CPU use varies between 25% and 2.5% during and after the start process, respectively. This great variance of CPU values is due to start GUI components.

7. Conclusion

We presented an access control network infrastructure applied and tested in CHESF electric power company, but easily portable for any network infrastructure, critical or not. Our current proof of concept implementation currently combines the use of policy-based management with the XACML access control framework to provide a high-level advanced solution.

Towards this end, our SIRCAM architecture makes the following contributions. In our work we consider some dependencies and operational requirements to describe a model of access control to permit or forbid network ingress actions. In our model, agents make requests to gain network access through common access protocols and standards without the need for infrastructure changes. We extended the XACML communication model to accept asynchronous messages and consequently to offer a more complete management of resources. We also created a simple and efficient LDAP schema to store text policies rules based on XACML. The overall system also shows the effectiveness of a policy based access control mechanism.

Currently, SIRCAM architecture is in operational stage at CHESF. As a future work, we plan to develop an agent for the Windows operational system due to the fact that many of CHESF's operator notebooks use this operational system, and also work on improving interaction between policies.

8. Acknowledgment

This work was supported by Companhia Hidro Elétrica do São Francisco (CHESF).

References

Aboba, B., Bluck, L., Vollbercht, J., Carlson, J., Levkowetz, H. (2004) "Extensible Authentication protocol (EAP)", RFC 3748, July.

CEPEL. (2004) "SAGE", http://www.sage.cepel.br.

Cigré. (2008), http://www.cigre.org.

CHESF. (2008) "Companhia Hidro Elétrica do São Francisco", http://www.chesf.gov.br.

CRUTIAL. (2008) "CRitical UTility InfrastructurAL resilience", http://crutial.cesiricerca.it/

FreeRADIUS. (2008) "The FreeRADIUS Project", http://www.freeradius.org.

IEC 61850. (2002) “Communication networks and systems in substations”, IEC, February.

IEC/TS 61850-6. (2007) “Power systems management and associated information exchange - Data and communications security - Part 6: Security for IEC 61850”, June.

IEEE. (2001) “Standards for Local and Metropolitan Area Networks: Port based Network Access Control”, IEEE Standard 802.1X-2001, June.

He X., Wang G., Zhao J. (2005) "Research on the SCADA /EMS System Data Warehouse Technology", In: Proceedings of Transmission and Distribution Conference and Exhibition: Asia and Pacific, IEEE Power Engineering Society, China.

Helfrich, D., Ronnau, L., Frazier, J., Forbes, P. (2006) “Cisco Network Admission Control, Volume I: NAC Framework Architecture and Design”, Cisco Press.

ITI. (2008) “TCIP: Trustworthy Cyber Infrastructure for the Power Grid Center”, http://www.iti.uiuc.edu/tcip/index.html.

JRadius. (2008) “The Open Source Java RADIUS”, http://coova.org/wiki/index.php/JRadius.

Kent, S., Atkinson, R. (1998) “Security Architecture for the Internet Protocol”, RFC 2401, November.

Microsoft. (2007) “Network Access Protection Platform Architecture”, http://microsoft.com/technet/network/nap/naparch.mspx.

OASIS. (2003) “LDAP profile for distribution of XACML policies (Working draft 01)”, October.

OASIS. (2005) “eXtensible Access Control Markup Language (XACML) Version 2.0”, February.

Open1x. (2008) “Open Source Implementation for 802.1X”, http://open1x.sourceforge.net.

OpenLDAP. (2008) “OpenLDAP community developed LDAP software”, http://www.openldap.org.

Rigney, C., Willens, S., Rubens, A., Simpsom, W. (2000) “Remote Authentication Dial in User Service (RADIUS)”, RFC 2865, June.

SUN. (2008) “XACML’s Implementation”, http://sunxacml.sourceforge.net.

Strassner, J. (2004) “Policy-Based Network Management: Solutions for the Next Generation”, Morgan Kaufmann Publishers.

Ward, S., O’Brien. J., et al. (2007) “Cyber Security Issues for Protective Relays”, In: Proceeding of IEEE Power Engineering Society General Meeting, June.

Westerinen, A., et al. (2001) “Terminology for Policy-Based Management”, RFC 3198, November.