DESIGN AND IMPLEMENTATION OF AN UNAUTHORIZED INTERNET ACCESS BLOCKING SYSTEM VALIDATING THE SOURCE INFORMATION IN INTERNET ACCESS LOGS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

YUSUF UZUNAY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

SEPTEMBER 2006

Approval of the Graduate School of Informatics

_____

Assoc. Prof. Dr. Nazife BAYKAL

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

_____

Assoc. Prof. Dr. Yasemin YARDIMCI

Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Assoc. Prof. Dr. Nazife BAYKAL

Supervisor

Examining Committee Members

Prof. Dr. Ersan AKYILDIZ	(METU, IAM)	_____
Assoc. Prof. Dr. Nazife BAYKAL	(METU, II)	_____
Dr. Kemal BIÇAKCI	(METU, II)	_____
Assist. Prof. O. Ayhan ERDEM	(GAZI, TEF)	_____
Dr. Altan KOÇYİĞİT	(METU, II)	_____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Yusuf UZUNAY

Signature : _____

ABSTRACT

DESIGN AND IMPLEMENTATION OF AN UNAUTHORIZED INTERNET ACCESS BLOCKING SYSTEM VALIDATING THE SOURCE INFORMATION IN INTERNET ACCESS LOGS

Uzunay, Yusuf
M.Sc., Department of Information Systems
Supervisor: Assoc. Prof. Dr. Nazife Baykal

September 2006, 101 pages

Internet Access logs in a local area network are the most prominent records when the source of an Internet event is traced back. Especially in a case where an illegal activity having originated from your local area network is of concern, it is highly desirable to provide healthy records to the court including the source user and machine identity of the log record in question. To establish the validity of user and machine identity in the log records is known as source authentication.

In our study, after the problem of source authentication in each layer is discussed in detail, we argue that the only way to establish a secure source authentication is to

implement a system model that unifies low level and upper level defense mechanisms. Hence, in this thesis we propose an Unauthorized Internet Access Blocking System validating the Source Information in Internet Access Logs. The first version of our proposed system, UNIDES, is a proxy based system incorporating advanced switches and mostly deals with the low level source authentication problems. In the second version, we extend our system with SIACS which is an Internet access control system that deals with the user level source authentication problems. By supplementing the classical username-password authentication mechanism with SSL client authentication, SIACS integrates a robust user level authentication scheme into the proposed solution.

Keywords: Real-time communication security, source authentication, TCP/IP security, audit system, network security

ÖZ

İNTERNET ERİŞİM KAYITLARINDAKİ KAYNAK BİLGİSİNİ DOĞRULAYAN YETKİSİZ BİR İNTERNET ERİŞİM BLOKLAMA SİSTEMİNİN UYGULANMASI VE TASARIMI

Uzunay, Yusuf
Yüksek Lisans, Bilişim Sistemleri Bölümü
Tez Yöneticisi: Doç. Dr. Nazife Baykal

Eylül 2006, 101 sayfa

İnternet üzerinden gerçekleştirilen bir olayda, olayın kaynağını tespit etmek için başvurulan en önemli kaynak yerel ağdaki İnternet erişim kayıtlarıdır. Özellikle yerel ağ üzerinden gerçekleştirilmiş hukuka aykırı bir olay söz konusu olduğunda, İnternet erişim kayıtları kullanılarak olayı gerçekleştiren şahsı ve olayın vukuu bulduğu bilgisayar sistemini doğru bir şekilde tespit etmek daha da büyük bir önem arz etmektedir. İnternet erişim kayıtlarındaki kullanıcı ve bilgisayar sisteminin kimliğini bu şekilde ispatlamaya kaynak doğrulaması adı verilmektedir.

Çalışmamızda öncelikle kaynak doğrulamadaki mevcut problemler detaylı bir şekilde incelenmiş, güvenli bir kaynak doğrulama mekanizması gerçekleştirmenin tek yolunun üst seviye ve alt seviye güvenlik mekanizmalarının birleştirilmesi olduğu kararına varılmıştır. Bu nedenle bu tez çalışmasında "İnternet Erişim Kayıtlarında kaynak bilgisini doğrulayan Yetkisiz İnternet Erişim Bloklama Sistemi" isminde yeni bir sistem önerilmektedir. Önerilen sistemin ilk versiyonu olan UNIDES, ileri düzey anahtarları barındıran Proxy tabanlı bir sistemdir ve daha çok düşük seviye kaynak doğrulama problemleriyle ilgilenmektedir. İkinci versiyonda sistem, SIACS isminde kullanıcı seviyeli kaynak doğrulama problemleriyle ilgilenen bir İnternet erişim denetim mekanizması sunan ekstra bir sistemle desteklenmiştir. SIACS, önerilen çözüme kullanıcı-parola doğrulama mekanizmasının yanı sıra SSL istemci doğrulama mekanizmasını da entegre ederek çok sağlam bir doğrulama sistemi sunmaktadır.

Anahtar Kelimeler: Gerçek-zamanlı iletişim güvenliği, kaynak doğrulaması, TCP/IP güvenliği, kayıt sistemleri, ağ güvenliği

This thesis is dedicated to:

My Lovely Mother
My Brave Father
My Doctor Brother
And
My Youngest Brother (kazandibi)

For their endless support,
For their love…

ACKNOWLEDGEMENT

It is a pleasure for me to express my sincere gratitude to Dr. Kemal Bıçakcı for his patience, encouragement and guidance throughout the study (especially for my first paper UNIDES). I greatly appreciate his helps when I take my first steps in Academic Writing and always being with me in every phase of this Thesis.

I would like to also express my special gratitude to my supervisor Dr. Nazife Baykal for her support, guidance, helps and suggestions throughout my research.

I am very grateful to Davut İncebacak for the reviewing of my thesis and continuous support. My special thanks go to Mustafa Koçak, Cahit Güngör, Ayhan Çankaya, Mahir Ersöz, Mustafa Eker, Lütfü Ersoy, Yüksel Türkal and the officers of Ankara Police Department Computer Unit for their encouragement and helps in my job.

I will never forget the support of my homemates Hasan Fatih Solak, Mesut Muhammet Solak and Mahmut Şevket Solak.

Also, I owe much to the committee members Dr. Ersan Akyıldız, Dr. Altan Koçyiğit and Dr. O. Ayhan Erdem for helpful comments and discussions.

I appreciate Ayse Ceylan, Sibel Gulnar and Ali Kantar in the institute for their kindness since the beginning of my M.Sc. study.

Finally, I will also never forget the unending support my family has provided me with during all the hard times.

TABLE OF CONTENTS

ABSTRACT iv
ÖZ vi
DEDICATION……..……..……...viii
ACKNOWLEDGEMENT ix
TABLE OF CONTENTS x
LIST OF TABLES xiii
LIST OF FIGURES xiv
LIST OF ABBREVIATIONS AND ACRONYMS xvi
CHAPTER
I INTRODUCTION 1
I.1 Security in Local Area Networks 2
I.2 Internet Access and Authentication 3
I.3 Internet Access Challenges in Local Area Networks 4
I.4 Source Authentication in Internet Access Logs 5
I.5 Scope of This Thesis 7
I.6 Outline of This Thesis 9
II BACKGROUND 11
II.1 Cryptography 12
II.1.1 Secret Key Cryptography 13
II.1.2 Public Key Cryptography 14
II.1.3 Message Authentication Code 15
II.1.4 Pseudorandom Number Generators 15
II.1.5 One Way Hash Functions 16

II.1.6 Digital Signatures and Certificates 16
II.2 Audit Systems 17
II.3 Authentication 19
II.3.1 Password Authentication 20
II.3.2 Credentials 23
II.3.3 Web Authentication 24
II.3.3.1 Web Password Authentication 25
II.3.3.2 Single Sign-on 26
II.3.3.3 Secure Socket Layer (SSL) 27
III PROBLEM DEFINITION 30
III.1 Source Authentication Problem 31
III.1.1 Source Identity Authentication 31
III.1.1.1 Problem of Application Layer User Authentication 31
III.1.1.2 Problem of Weak User Authentication Schemes 32
III.1.1.3 Problem of Real Time User Authentication 33
III.1.2 Source Machine Authentication 33
III.1.2.1 Address based Host Authentication 35
III.1.2.2 MAC Address Authentication 36
III.1.2.3 Cryptographic Solutions at Host Level 37
IV RELATED WORK 39
IV.1 RELATED WORK 40
IV.1.1 Secure Address Resolution Protocol 40
IV.1.2 Static ARP Tables 41
IV.1.3 ARPWATCH 42
IV.1.4 MAC Binding on the Switch 42
IV.2 RELATED METHODS 43
V OUR PROPOSED SOLUTION 46
V.1 UNIDES-Unauthorized Internet Access Detection and Blocking System 47
V.1.1 The Framework Model of UNIDES 48
V.1.1.1 Topology 48
V.1.1.2 Proxy Server and Firewall 49

V.1.1.3 Advanced Switches and Port Security Mechanism 51
V.1.1.4 DHCP Server 53
V.1.2 Operation of UNIDES 53
V.1.3 Management Interface 55
V.1.4 Advantages of UNIDES 58
V.1.4.1 Security 58
V.1.4.2 Intranet Management and Performance Analysis 59
V.1.5 SUMMARY 62
V.2 SIACS: SSL Based Internet Access Control System to provide User Level Source Authentication in Internet Access Logs 63
V.2.1 Framework of SIACS 64
V.2.1.1 Firewall in front of the Proxy Server: 65
V.2.1.2 Proxy Server 66
V.2.1.3 Certificate Server 67
V.2.2 Operation of SIACS 68
V.2.2.1 Definition of a New User 68
V.2.2.2 Certificate Installation on Client Side 69
V.2.2.3 Internet Logon 72
V.2.2.4 Session Termination in SIACS 75
V.3 Development Environment 76
V.4 Robustness of Our Proposed Solution 77
VI CONCLUSION and FUTURE WORK 79
REFERENCES 85
APPENDICES
A. SCREEN SNAPSHOTS of UNIDES 90
B. SCREEN SNAPSHOTS of SIACS 95
C. CURRICULUM VITAE 100

LIST OF TABLES

Table 1: Measurement of Data Input using UNIDES 59
Table 2: Measurement of Data Input using manual ways 59
Table 3: Total Performance Gain in Data Input by using UNIDES 60
Table 4: Measurement of Data Update using UNIDES 60
Table 5: Measurement of Data Update using manual ways 60
Table 6: Total Performance Gain in Data Update by using UNIDES 61
Table 7: Measurement of Data Deletion using UNIDES 61
Table 8: Measurement of Data Deletion using manual ways 62
Table 9: Total Performance Gain in Data Deletion by using UNIDES 62

LIST OF FIGURES

Figure 1 : SSL Handshake without Client Authentication 27
Figure 2 : Warning message when the certificate is not in the browser's trust list..... 29
Figure 3 : Internet Stack Layers, Authentication Mechanisms and the Security Threats 34
Figure 4 : MAC-IP Scan Graphics 45
Figure 5 : The Topology of UNIDES 49
Figure 6 : Operational scheme of UNIDES 54
Figure 7 : The Time Graphic for Data Input 60
Figure 8 : The Time Graphic for Data Update 61
Figure 9 : The Time Graphic for Data Deletion 61
Figure 10: The Topology of SIACS 64
Figure 11: SSL Client Authentication Configuration in SIACS Apache Web Server 65
Figure 12: Client Proxy Settings for SIACS 67
Figure 13: Certificate Import Wizard 70
Figure 14: Import of the file incorporating user's certificate 70
Figure 15: Password determination for private key 71
Figure 16: Certificate Store Determination 71
Figure 17: Completion of Certificate Import 72
Figure 18: Certificate Selection in SSL Client Authentication 73
Figure 19: Flowchart of SIACS Internet Logon 74
Figure 20: Data Input(Upper) and Unauthorized Internet Access Alarm(Lower) frames 90
Figure 21: The View of a Record (Yusuf UZUNAY) while it is being defined..... 91

Figure 22: View of Configuration Error in a Switch 91
Figure 23: View of a Configuration Error occurred on DHCP Server 92
Figure 24: The view of Configuration Errors both in the Switch and in the DHCP Server 92
Figure 25: The View of the Records having no problem 93
Figure 26: Record Search 93
Figure 27: The View of Detected and Blocked Records by UNIDES 94
Figure 28 : SSL Client Authentication and Certificate Registration Control 95
Figure 29: Authentication failure when the given certificate is not registered to SIACS 95
Figure 30 : SIACS Username-password Authentication Interface 96
Figure 31: Authentication failure when there is incompatibility between certificate and user information 96
Figure 32: Authentication failure in user status control 97
Figure 33 : Data Transfer to the Server 97
Figure 34: Definition is being performed on the server 98
Figure 35: Authentication succeeded and Internet access is given to the user 98
Figure 36: Log out process is initialized 99
Figure 37: Log out process finished and user's Internet connection ended 99

LIST OF ABBREVIATIONS AND ACRONYMS

ARP	: Address Resolution Protocol
AH	: Authentication Header
CA	: Certificate Authority
CN	: Common Name
DB	: Database
DES	: Data Encryption Standard
DHCP	: Dynamic Host Configuration Protocol
DMZ	: Demilitarized Zone
DoS	: Denial of Service
ESP	: Encapsulation Security Payload
HTML	: Hyper Text Markup Language
HTTP	: Hyper Text Transfer Protocol
IDS	: Intrusion Detection Systems
IP	: Internet Protocol
IPSec	: Internet Protocol Security
ISP	: Internet Service Provider
LAN	: Local Area Network
MAC	: Media Access Control
MAC	: Message Authentication Code
NAT	: Network Address Translation
NTP	: Network Time Protocol
OSI	: Open Systems Interconnection
PIN	: Personal Identification Number
PKCS	: Public-Key Cryptography Standards

PKI	: Public-Key Infrastracture
RSA	: Rivest-Shamir-Adleman
SIACS	: SSL Based Internet Access Control System
SSL	: Secure Socket Layer
SSO	: Single Sign On
TCP	: Transmission Control Protocol
UNIDES	: Unauthorized Internet Access Detection and Blocking System
URL	: Uniform Resource Locator
VLAN	: Virtual Local Area Network
VPN	: Virtual Private Network

CHAPTER I

INTRODUCTION

"Wisely and slow; they stumble who run fast"
[Shakespeare Rom & Jul, Act ii, Sc.3]

"A good beginning is half the finish."
[Turkish Proverb]

With the developments in technology, most of the daily activities of human beings have began shift towards digital environment and concepts such as e-commerce, e-banking, e-government and the other "e" words have come to the forefront. The proliferation of Internet services has gradually led to the distribution and marketing of all kind of materials especially the ones including critical data through Internet. This is the main reason why information security has become a very important issue in the last decade.

"Security" can be defined as to protect certain assets in spite of certain threats and attacks. Broadly speaking security has 3 main targets:

- Confidentiality
- Integrity
- Availability

To define briefly, confidentially is the prevention of unauthorized disclosure of information, integrity is the prevention of unauthorized modification and availability is the prevention of unauthorized withholding of information [1].

There is another term called “authentication” which we encounter very often in security domain. The reason why we do not include this term in three main security targets is that the authentication has to be involved in each of them. In other words without authentication, the other security targets can not be provided. Authentication is the process of determining if an entity is who it claims to be.

I.1 Security in Local Area Networks

In a general perspective; security threats can be classified into two categories. One of them is external threats and the other is internal threats. External threats stand for the threats the origin of which is outside of an organization i.e. an outside hacker attempting to break into the web server of an organization. Internal threats, on the other hand, stand for the threats the origin of which is inside the organization. Local area network attacks are the most prominent type of internal attacks.

The number of attacks against information systems increases rapidly especially in the last years. It is stated that nearly 80% of these attacks are caused by the insider abuse of net access according to CSI/FBI survey reports [2]. This is the reason why security in local area networks is an issue of paramount importance.

Most of the external attacks can also be performed inside the local area network. But LAN specific attacks (i.e. data link layer (layer 2) attacks) can not be performed from outside of the network of an organization. Data link layer is an OSI layer responsible for the communication between adjacent nodes. Date link layer address, also known as MAC address, is used to send layer 2 frames to the nodes which can be reached directly without any routing operation. This means that data link layer

communication is established between two nodes which are on the same network. Therefore; it is infeasible to perform a data link layer attack from a node outside of the organization network.

We are confronting a lot of LAN attacks for years. The most famous ones are MAC spoofing, MAC cloning and ARP attacks. The reason why these attacks are still effective is that Internet protocols were not designed security in mind when they were first proposed. In the time, when Internet is newly emerging, the focus was to create simple, easy to use and effective protocols to provide usability in all over the world. Security was not an issue at that time. Because of the fact that most of the applications and protocols are designed over the existing Internet protocols and they are still in use, it is very difficult to design new protocols instead of the old ones from scratch. So; security people mostly try to strengthen the available protocols by adding new features without violating the compatibility with the original ones or find solutions on top of the available protocols.

I.2 Internet Access and Authentication

Internet access and authentication are two main concerns of this thesis. Although we mostly deal with these issues in local area networks, we are going to give an overview of Internet access types in this section, to provide a general view before focusing on more specific issues. Technically, Internet access is provided to end users in following three ways [3]:

- **Residential Access:** Type of access connecting home end systems into Internet.
- **Company Access:** Type of access connecting end systems in a business or educational institution into Internet.
- **Mobile Access:** Type of access connecting mobile end systems into Internet.

In residential access, individuals generally have a direct connection to the Internet through some kind of gateway devices such as modems. But in company access users have to first connect to an internal network through the network interface cards of their machines. Mobile access is mostly done through some mobile devices such as mobile phones, PDAs, laptops having wireless NIC and etc.

Regarding this highly distributed Internet access architecture where users connect with a broad range of devices through different wired and wireless networks, it is likely that unauthorized intrusion and access to private personal data by a third party or other misuse may occur to all types of Internet accesses. Today, we observe that Internet is also used by malicious users, terrorists, smugglers, thieves and etc. The number of crimes performed by using Internet and communication technologies is increasing day by day.

As a matter of fact implementing security in Internet accesses is a very important issue. Especially authentication mechanisms which prove the identity of the users having an Internet connection are required in all type of Internet accesses.

I.3 Internet Access Challenges in Local Area Networks

In this thesis, because of that we deal with source authentication in Internet access logs in a local area network, our focus will be on company access which is the type of Internet access provided in the organizations. In a more detailed view, a user who wants to have an Internet access has to first connect to the internal network of the organization. By default, in order to make connection in the Internet a real IP address is needed. Today's IP address scheme, IPv4, uses 32 bits for Internet protocol addresses. But everyday the number of Internet users increases and this address space is deemed to have ran out in the near future.

One important invention to use IP addresses in a thrifty fashion is the NAT (Network Address Translation) technology. NAT provides Internet connection to a lot of

machine using limited real IP addresses (i.e. one or two real IP addresses). This real IP address is mostly assigned to the gateways within the network. In this structure, clients are assigned an IP address which is not used in the Internet. This is called virtual IP address. When connecting to the Internet, each user has the IP address of the gateway. There is a translation methodology in NAT. Whenever a user having a virtual IP address wants to make a connection through gateway, the IP address of the packets are changed by the gateway with real IP addresses. This translation is also kept in a NAT table in order to be able to send the reply packets to the original source back.

Albeit the usefulness of NAT technology, it also involves a lot of risks in terms of security. Because when a criminal activity is performed from the local network of an organization to a site at the outside, the IP address of the Gateway is detected and if there is not any security implemented system to prove which of the user is responsible for this activity, it is impossible to detect the criminal. So, this case explains why the authentication and audit in local area networks has a paramount importance.

I.4 Source Authentication in Internet Access Logs

Suppose that the cyber crime investigators claim that they have strong evidence that a credit card fraud has happened originated from a machine located in your company's intranet. For the sake of protecting your company's high reputation, you should find out which one of your employees is responsible for. Hence you consult to your audit records and find the IP address of the aforementioned machine. You also have a reliable record showing to whom this IP address was assigned to. However, the employee in suspect denies the claim and says that although it seems impossible that another employee can login to his computer (a strong biometric authentication is in place), his IP address must have been used by someone else. Did he say the truth? How can you solve this dispute?

When the auditing of Internet activities is of concern, the most important security issue is the source authentication, establishing the identity of the owner of recorded activities and the connection machine. A naive solution to this problem is to authenticate every user before allowing his machine to have an Internet access. Unfortunately this works only if there is no attack to the system! An attacker with access to the lower levels can easily subvert the authentication at the user level [1]. For instance, without any further protection, it is as simple as modifying the IP packet headers so that it appears that packets are coming from another machine.

How can we establish the authenticity and the auditability of Internet accesses in a local area network? Most of the time audit in Internet access logs is implemented by recording which IP address has connected to which site on which date and at which time. For example "proxy" is the easiest way of keeping the Internet audit trails of IP addresses. More advanced commercial audit systems do exist; providing detailed audit reports and statistics.

Various authentication techniques are in use today to provide the identity of the user or machine in order to be able to trace back to source from audit logs, namely user level authentication techniques (i.e. username-password, smartcards, biometrics and etc), authentication servers (i.e. Kerberos, VPN), IP address based authentication, MAC Address Based Authentication and etc. However, all these authentication schemes suffer from some serious problems when the source authentication of Internet audit records is of concern.

As we previously mentioned, using only upper layer authentication mechanisms enables an attacker to break in the system by gaining access to the lower levels. In order to prevent this, we can supplement upper level authentication mechanisms with low level authentication mechanisms such as IP address based authentication in network layer or MAC address based authentication in data link layer. But because of the inherited deficiencies in most of the existing Internet protocols, to perform a healthy low level authentication is very difficult. For example IP addresses can easily

be forged by using IP Spoof Attacks or data link layer can easily be exploited by a lot of attacks such as MAC spoof, MAC cloning, MAC flooding attacks. Using cryptographic solutions is also problematic in source authentication of Internet access logs. Internet audit systems, by their nature, are the systems recording Internet activities of the users in real time. Hence, cryptographic solutions having heavy computational requirements, is not appropriate in source authentication of Internet access logs because of the efficiency limitations.

In order to establish a healthy source authentication, we need to implement a security mechanism that provides both upper layer and low level authentication mechanisms in an integrated fashion. This system should also be efficient enough to be used in source authentication in real time Internet auditing.

When the literature is reviewed, we could not find any effective solution to this problem. In Chapter 3, we will examine all the existing authentication techniques in detail and see where they have bottlenecks.

I.5 Scope of This Thesis

In this thesis, our objective is to find out a secure and efficient solution to the problem of trusted source authentication in Internet access logs. Instead of offering an abstract solution, we prefer to construct a real architecture that can be used in today's networks easily. Thus, we give due attention to design our system on top of the available protocols and systems.

Another issue is the simplicity of the system. We believe that if the design is simple, not only the practicability of our solution is increased but also the overhead on the existing network resources is minimized.

From security perspective, we believe that the standardized protocols, the robustness and security of which are approved worldwide are always better than home made

protocols. Hence, we try to implement our idea using the well known open source protocols. In more precise terms, our contribution is not to make improvements in the available protocols but to build a special architecture integrated with secure protocols that provides high level of assurance in source authentication of Internet audit logs.

Considering all these issues, we propose an Unauthorized Internet Access Detection and Blocking System validating the Source Information in Internet Access Logs. In the first version of our system (UNIDES), we have built an unauthorized internet access detection and blocking mechanism which provides low level source authentication [4]. And then we have extended UNIDES with a second system called SIACS: SSL Based Internet Access Control System in order to add a robust user level authentication mechanism to our proposed solution.

UNIDES functions at the bottom line of the architecture and mostly deals with the low level authentication problems such as IP address authentication and MAC address authentication. It also prevents most of the attacks originating from the vulnerabilities of data link layer protocols. UNIDES has a proxy based architecture incorporating the advanced switches, the port security mechanisms of which are activated. In the framework of UNIDES, every user is first registered into the system by assigning them a constant IP address recorded together with his machine's MAC address. Whenever a user changes his/her IP address, the system automatically detects and blocks his Internet access. Integrated to ease the administration, there is also an interface in UNIDES to manage all the relevant security and administrative processes.

SIACS is a user level source authentication system on top of the UNIDES. It supports UNIDES to differentiate each user using the same computer and implements a very robust user authentication scheme. In order to have an Internet access, user has to be first connected to SIACS web server and authenticate himself. SIACS authentication is composed of SSL client authentication and username-password authentication.

Only if all the given credentials are true, Internet access permission is then given to the user.

By implementing these two systems in a correlated fashion in our architecture, we obtained a complete system providing the security of the source authentication mechanisms in all layers of Internet stack. For real time requirements of Internet Audit logs, we do not follow up a methodology to make use of a real time authentication mechanism because of the efficiency concerns, but instead, we have constructed a system model that detects any unauthorized action almost in real time and blocks it. In other words, If we sure that the user and the machine identity is not changed after the first authentication, we do not need to perform a real time authentication.

I.6 Outline of This Thesis

This dissertation is composed of 6 chapters. First chapter is the introduction giving information about the leading factors why we need to publish this thesis. In this chapter, we first define security concepts and introduce the main security targets. The importance of security in local area networks is then emphasized. Because of the fact that Internet access of the users is the main issue in our study, we take a look at how Internet access is provided to end users both in general and local perspectives. And after a brief evaluation of source authentication in Internet audit logs which is our main concern, we lastly present scope and outline of this thesis.

We have dedicated chapter 2 for background information. Background chapter is divided into 3 sections. In order to understand this thesis better, we first give an introductory information about cryptography and define the most important terms briefly. A section trying to introduce audit systems follows up the cryptography section. Than we end the background chapter with the authentication section. In authentication section some of the most used authentication schemes are mentioned and their pros and cons are discussed.

Chapter 3 tries to define the problem of source authentication and argue why it is difficult to have a healthy source authentication. Source authentication is examined into two parts. One of them is source identity authentication; the other is source machine authentication.

In chapter 4, we give an overview of previous studies. Different solutions are proposed to solve problems in especially low level authentication mechanisms such as IP and MAC address based authentication. One previous work has proposed another protocol instead of ARP to provide secure address resolution. Another has proposed using static ARP tables and yet another has constructed a mechanism to detect the changes in the MAC/IP pair. But all these solutions suffer from some problems.

Our proposed solution, Unauthorized Internet Access Blocking System validating the Source Information in Internet Access Logs, is introduced in a detailed fashion in Chapter 5. In the first version of our proposed solution (UNIDES), low level authentication problems and in the second version (UNIDES extended with SIACS) user level authentication problems are prevented.

Finally we wrap up our thesis with a conclusion and future work in Chapter 6. We believe that our systems can also be used in other applications where healthy source authentication is an issue of importance.

CHAPTER II

BACKGROUND

"The one who is unaware of his past
can not make any comments about his future."
[Turkish Proverb]

Before going into more detail about the problem of source authentication and the previous studies in the next section, we in this chapter give some background information about the core subjects on which our discussions in this thesis focus.

We start with cryptography which we believe as an indispensable part of every security subject. Cryptography has a long history dating back at least as far as Julius Caesar. It is a very deep issue and there exist a lot of good resources giving introductory information about cryptography such as [5,6,7,8,9,10,11]. We will try to define the most basic concepts of cryptography in order to provide clarity of the issues in this thesis. Than we will handle the subject of audit systems in order to understand why audit systems are important in the security domain and how audit is performed in most of the networks. As one of the main goals of this thesis is source authentication in Internet access logs, we see it is useful to understand the issue of authentication. Since we implement a web based authentication scheme in our proposed solutions, after some brief information about authentication, we mostly focus on the web based authentication techniques throughout section II.3.

II.1 Cryptography

Just as a doctor needs to understand physiology as well as surgery, so a security engineer needs to be familiar with cryptology as well as computer security (and much else) (Ross. J. Anderson). The word "cryptography" is derived from Greek and when literally translated, means "secret writing." Cryptography was primarily used by the military purposes. With the advances in modern communication, information transportation can be easily carried out at a very low cost via public networks such as Internet. This development also brings about the risks of data exposure during the transportation. Hence, it becomes an obligation for businesses to make sure that sensitive data is transferred from one location to another in a secure manner. Cryptography can help us to achieve this goal by making messages unintelligible to all by using some techniques so that only the intended recipient(s) can recover these messages to original form and understand.

Encryption refers to the transformation of data in "plaintext" form into a form called "ciphertext," which renders it almost impossible to read without the knowledge of a "key," which can be used to reverse this transformation. The recovery of plaintext from the ciphertext requires the key, and this recovery process is known as decryption. This key is meant to be secret information and the privacy of the ciphertext depends on the cryptographic strength of the key [12].

There are two main types of cryptography. These are Secret Key Cryptography and Public Key Cryptography. Before moving on detail about these issues, let's introduce Alice and Bob whom we will use very often in our thesis. They are well-known fixtures in the security community. They stand for two objects (two people, two machines, two services or one people one machine and etc.) which want to make secure communication with each other.

II.1.1 Secret Key Cryptography

In our first type, assume that Alice wants to send an encrypted message to Bob. She encrypts her plain text message with a shared key which is also known by Bob then she sends the cipher text. When Bob receives the cipher text, he decrypts the cipher by using the same shared key. In this analogy because of that the encryption and decryption operation are performed by using the same shared key, the secrecy of the key is very important. Hence, this cryptographic algorithm is known as *Secret Key Cryptography*. Because the keys are same, it is also known as *Symmetric Cryptography*. Well known examples of secret key algorithm are DES, 3-DES, RC4, RC5 and etc.

Secret Key Cryptography is widely used and popular. Provided that the key is secure and relatively strong, the algorithm is considered secure. Secret Key Cryptography is also very fast relative to public key cryptography. The most important problem in secret key cryptography is how to distribute the shared key to the participants [13]. This is known as **key distribution problem**.

In cryptography, one basic principle is that the security of an algorithm should depend on only the secrecy of the key not the secrecy of the encryption algorithm. This is the reason why open source algorithms are very popular today and used in very important Internet protocols.

Another important issue is the key length. Because the security of an encryption algorithm is directly related to the key used, the only way to crack an algorithm is to find the encryption key. If any other intelligence is not of concern, the only way to find the key is performing Brute Force Attack. The most secure (secret-key) cipher is the one which can not be broken by a less effort than the brute force attack. The susceptibility of a cipher to a brute force attack is directly related to the key length e.g., an algorithm with a 40-bit key length can be broken by about a trillion trials. Of course the longer the key, the harder to break the system by brute-force attack, but on

the other hand using a key longer than what we need slows down the encryption/decryption operations and is inconvenient to store and to transmit [14].

II.1.2 Public Key Cryptography

Public Key Cryptography is another type of cryptography which uses one key for encryption and a different but correlated key for decryption operations instead of using the same shared key. Public Key Cryptography is first proposed by Diffie and Hellman in 1976 and accepted as one the most important promotion in the Cryptography domain [15]. Public Key Cryptography is based on mathematical functions instead of basic bit operations.

The most important benefit of public key encryption is to solve the key distribution problem in secret key cryptography. There are two different keys in Public Key cryptography. One of them is private key, the other is public key. As the names implies, the private key is kept secret and the other one is distributed that means whoever wants can easily obtain the public key. In order to send an encrypted message to Bob, Alice first obtains the public key of Bob which is open and then encrypts her plain text message with this public key. After receiving the cipher text, Bob decrypts the cipher by using his private key. In Public Key Cryptography the cipher text which is encrypted by a public key can only be decrypted by using its correlated private key. Because of that Bob is the only owner of the private key, no one other than Bob can decrypt the cipher text.

In this scheme, the number of keys managed by each user is much less compared to secret key cryptography. But because of that public key encryption requires some heavy mathematical calculations, it is much slower than secret key cryptography and the cipher text is much larger than the plain text, relative to secret key cryptography.

II.1.3 Message Authentication Code

Assume that Bob wants to ensure that the message sent by Alice has not been altered on the way and really sent by Alice. In a technical expression, Bob wants to authenticate the Alice and ensure the integrity of the message. These two security functions can be basically established by using Secret Key Cryptography.

After creating the plain text, Alice processes message content with her secret key. This is as same as the secret key encryption operation. But in this case instead of sending only the cipher text, Alice adds cipher text at the end of the plain text and sends both of them. The added cipher text is called MAC (Message Authentication Code). Received the message, Bob also calculates MAC by using his secret key which is shared between Alice and him. If two MAC is same, this means that the message has not been altered on the way and because only Alice has the same secret key, the message has been sent by Alice. Note that MAC does not provide secrecy but provides authentication and integrity.

II.1.4 Pseudorandom Number Generators

Almost every security protocol that uses cryptography needs for reliable random numbers. That is, given a stream of outputs from a random number generator, it should be infeasible to predict what the next output value will be. The security of a cryptographic algorithm, therefore, is directly related to how strong random numbers are used.

In reality, random number generators are theoretical objects, existing only abstractly. In practical circumstances we create an algorithm which attempts to approximate a random number generator; the outputs are then called **pseudorandom numbers** and the generator is called a **pseudorandom number generator.**

II.1.5 One Way Hash Functions

As we have just mentioned, by utilizing secret key, the authentication of the message content and the sender can be carried out. But sometimes it may be needed to authenticate only the message content. In this case "One way Hash Functions" which is a very effective method still preferable in many applications today, can also be utilized without any need to a key. In one way hash functions it is very easy to calculate H(x) from any x value but it is mathematically infeasible to calculate x from a given h value in the H(x)=h equation. The most important feature of hash functions is that they take any message with different lengths as an input and produces output at a constant length. Therefore one way hash functions are also used in increasing the efficiency of cryptography algorithm in the public key cryptography by reducing the high sized files to a constant value.

II.1.6 Digital Signatures and Certificates

How to provide authentication and integrity of message using public key cryptography? This is where digital signature comes into play. Digital signature operation is somewhat reverse of public key encryption operation. While we process the plain text with receiver's public key in encryption, now we process the plain text with sender's private key. Similar to MAC, we add the output (the cipher text), which is called as digital signature, at the end of the message and send both of them. Bob, receiver of the message, can verify the message by processing digital signature and public key of Alice (sender). Because only the Alice has the correlated private key, Bob ensures that the message is signed by Alice and the message has not been altered since it was signed. RSA which is the first proposed digital signature scheme has been widely used still today [16].

Because public key cryptography requires heavy mathematical calculations, it is not efficient to process whole plain text with private key. At this point one way hash

functions come to the help. Most of the digital signature algorithms first calculate one way hash of the plain text and then process the signing algorithm.

The biggest problem of digital signature is whether the aforementioned public key which is reachable by everyone and used in digital signature is really belongs to the owner claimed. Trudy who wants to impersonate Alice can produce a key pair (public and private key) herself and distribute the public key as Alice's public key. If Bob does not recognize this, Trudy can easily send signed messages instead of Alice. In order to prevent this kind of malicious attempts, digital certificate concept has come out.

A *digital certificate* is composed of a public key, some identity information about the owner of this key and the signature of a trusted third party which ensures the validity of this certificate. We call this trusted third party as "*Certificate Authority-CA*". Also the infrastructure needed to maintain this certificate based operations is called as "*Public Key Infrastructure-PKI*".

II.2 Audit Systems

The primary usage of audit systems is to detect the occurrence of a threat and its origin. According to the detail of the audit log, the analysis and traceability will be much easier. Assume that an attack is performed to a web server in our network. By analyzing the intrusion detection system audit logs we can reach which type of attacks are performed, the details of attack packets, the source IP addresses of these packets, the destination ports of the packets and other similar information about the attack. This log will give us an overview of the attack. By analyzing the firewall logs, we can reach the access records of the IP address that is detected in the intrusion detection systems logs. For example in one week period before the attack, is there any traffic to our network from the aforementioned IP address? How many packets of this IP address is blocked? By analyzing the audit logs of the server itself, we can detect

the unsuccessful or successful logins, the date and time of the login, the identity of the user who logged into the system in a specific time, the accesses to the critical records.

In order to increase the traceability of a threat the correlation between audit records are very important. In order to be able to set up some kind of correlation, it is needed to analyze more than one audit records. In other words, the more systems that keep audit records, the more chance you have to detect the origin of a threat.

As it is seen in our previous example, audit can be implemented in various systems such as firewalls, IDS, servers, hosts, network devices (e.g. routers, gateways) and etc. But the most important thing in audit systems is the time synchronization between all these system clocks because time knowledge is the prominent data to analyze and trace a threat. There are various ways to set up healthy time synchronization within your network. One of the most known one is NTP (Network Time Protocol) [17].

Up to know, we have seen that we can use audit systems in order to get detailed information about a threat after the event has occurred. But sometimes it is needed to detect the event in real time. These kinds of audit systems are called real time monitoring systems. In most of the real time monitoring systems, there are sensor devices which perform the real time detection task and a central monitoring system. When a threat is detected by the sensors, the threat information is transferred to the central monitoring system and monitoring system has the ability to warn the system administrators by using different ways such as sending an email or SMS, calling to an emergency center, giving alarm and etc. Intrusion detection systems are the well known real time monitoring systems.

In the scope of this thesis, we will mostly concern with Internet audit systems. By definition Internet audit systems are the systems that keep trails of the Internet activities of the users. For example which user has connected to xxxbank.com web

site at a specific date and time? To keep healthy Internet audit records is very important for system administrators especially when, an illegal activity (i.e. e-banking fraud) having originated from the local network is of concern. We will return back to this issue in the ongoing chapters.

Audit systems are beneficial not only for the detection and analysis of a threat but also for network analysis. They provide network administrators with statistics indicating that the network and its resources are functioning properly. This can be done by an audit mechanism that uses the log file as input and processes the file into meaningful information regarding system usage and security. Due to their volume and complexity, it can be very difficult to analyze and effectively utilize the audit data files and/or audit reports. The audit services should include tools that condense and organize the audit data to allow for ease of study. Tools should also provide the capability of sorting audit entries by categories (user entries, file system entries), date/time, physical location, etc [18].

II.3 Authentication

Authentication is the process that establishes the identity of some entity under scrutiny. For example in the airport, the passenger gives his passport to the border guard to prove his identity. The border guard checks the serial number of the passport in the database and looks at the photo of the passenger attached on the passport and validates the identity of the passenger. This validation is a form of authentication.

On the Internet, authentication is somewhat more difficult than the physical world. Because the entities do not have physical access to the parties they are authenticating. Malicious users or programs can obtain sensitive information, disrupt the services or impersonate the valid entities by using some kind of techniques. Distinguishing these malicious parties from valid entities is the role of authentication, and is essential to network security.

Successful authentication does not always mean to give access to the authenticated entity. There are other issues to check in order to decide if the entity will be able to access to a certain location. Again if we turn back to our airport example, after validating the identity, the availability of visas, the past criminal record of the passenger, the government's political issues are all the other important issues that will affect the access permission. In the literature this is called **authorization**.

While the preceding discussion has focused on entity authentication, it is important to note that other forms of authentication exist. In particular, message authentication is the process by which a particular message is associated with some sending entity. Another major difference between these two is that message authentication itself does not provide any timeliness guarantees whereas entity authentication confirms the identity in real-time (while the verifying entity awaits) [14].

Let's assume that Alice want to authenticate herself to Bob. Alice and Bob may not be users, but computers. For example a computer must authenticate itself to a file-server prior to being given access to its contents. No matter whether Alice is a computer or not, she must present some kind of evidence to prove her identity to Bob. This evidence is called **credential**. Bob evaluates the evidences and decides if the evidences are enough to validate the Bob's identity. If the validation is successful Alice is then called **authentic** to Bob.

II.3.1 Password Authentication

The authentication credentials differ from application to application. The most widely used authentication credential is password. In UNIX authentication framework, think that Alice wants to authenticate herself to Bob (UNIX system). In the time of the login Alice is requested a password. Than Bob checks if the given password is the one it knows that it belongs to Alice before. In password schemes because of that Bob believes Alice is the only one who posses this password, the authentication process is deemed as successful if the password is right.

In reality Bob's claim that Alice is the only one who could supply true password is not always true. There are a lot of attack types to circumvent the existing password authentication mechanism. For example password guessing attack is the most basic one. In this attack, some other third party tries to guess the password of Alice and tries to logon to the system on behalf of her. This goes on until the right password is matched. Most of the system defend against this kind of attacks by checking a threshold value of failed password and do not accept more trial or slow down the authentication process whenever a wrong password is entered.

Dictionary attacks and brute force attacks are more serious attacks. In most of the authentication schemes of the operating systems a salted hash of the password is stored in a file. If some adversary can acquire these hashed values, brute force or dictionary attacks can easily be initiated to find the password.

In dictionary attacks, there is a password dictionary including the most used passwords or words used in a specific language. The logic comes out from the idea of that people have a tendency to choose a password that is easily be remembered. So it is quite likely that the password might be one of the known words that is in the dictionary. In order to find out the password, all the words in the dictionary are tried one by one until the right one is matched.

If the dictionary attack fails, then brute force attack can be performed. In brute force attacks all the combinations of numeric and alphanumeric values are tried respectively. In this scheme the longer the length of the password the more difficult it is cracked.

Passwords are subject to more fundamental attacks. In one such attack, the adversary simply obtains the password from Alice directly. This can occur where Alice "shares" her password with others, or where she records it in some obvious place. In the case of authentication, failure of protecting credentials from misuse can result in system compromise.

Albeit a lot of security measures have been proposed heretofore [19,20,21,22,23], password vulnerabilities do still exist [24]. While system related vulnerabilities such as cleartext password transmission can easily be avoided using cryptographic protocols, the problems associated with end user behaviors bring a more challenging threat. In other words, it is a very well known fact that currently end users are the weakest link in the security chain and there is not any evidence that this situation will change in the near future. Four main password mistakes continuously done by the end users are listed below:

- Low-Entropy Password Choice
- Lack of Physical Protection
- Giving Away Passwords (Social Engineering Attacks)
- Password Reuse in Different Applications

Low Entropy Password: It is the password that is easily guessed or can be easily cracked by dictionary or brute-force attacks using tools such as Crack [25] and John the Ripper [26].

Lack of Physical Protection: For example writing password on a post-it note and stick it next to the monitor.

Giving Away Passwords: It is not difficult to convince unsuspected users to disseminate his password for instance by introducing himself as the system administrator over the phone. Another notable example for these so-called social engineering attacks is the new scam of phishing emails asking password and other sensitive information from the users by falsely claiming that these are necessary for instance to reactivate the user's account.

Password Reuse in Different Applications: Users mostly choose to use the same password for a number of different applications. If this is the case, the attacker's job becomes much easier. Any interesting looking web site is more than enough to collect

username password pairs that have previously been used for other - potentially security critical - applications. So after acquiring a password from a user, the attacker can try to authenticate himself in different applications by using the acquired password i.e. to learn someone's password used in a web site offering white papers upon registration and try it in the Internet applications of major banks.
Broadly speaking, passwords are the least secure way of authenticating people. Then why is this method still preferred in many applications including highly security sensitive ones?

The answer can be summarized as follows:

- Username-password scheme is easy to implement because it has a simple structure.
- It is very easy to use from users' perspective.
- It is inexpensive compared to other authentication schemes.

Due to the fact that passwords are the most used and the least secure authentication method, it has frequently come under fire by the attackers.

II.3.2 Credentials

Authentication is performed by the evaluation of credentials supplied by the user (i.e. Alice). Such credentials can take the form of something you know (e.g., password), something you have (e.g., smartcard), or something you are (e.g., fingerprint). The credential type is specific to the authentication service, and reflects some direct or indirect relationship between the user and the authentication service [27].

Most of the time, credentials are seen as the shared secret knowledge between the authenticating parties. As we previously mentioned, in UNIX systems this secret is a password. In general such secrets need not to be defined statically. For example, in One Time Password Authentication, user does not present the secret text directly but demonstrate knowledge of it (e.g., by presenting evidence that could only be derived from it). The value of this knowledge is only valid for a single authentication.

Other than secret knowledge, credentials can be something you posses. There is also a shared secret in this method but this secret is generally stored in a physical object. This physical object is mostly in the form of cards which are known as access cards. Many types of access cards have been developed. The newest types of access cards are called **smart cards**. These cards contain microchips that consist of a processor, memory used to store programs and data, and some kind of user interface. Sensitive information such as user's PIN code is kept in a secret zone of the read-only memory. This zone is encoded during manufacturing, using cryptographic techniques, and is inaccessible even to the card's owner [28].

Every person has some differentiated physical properties. The use of these specific characteristics to provide personal identification is known as **Biometrics**. This credential type is referred as "something you are" indicating the physical characteristics you have. In this scheme computerized biometric devices are used. These devices measure or detect the specific physical properties and compare them with the records that have been previously stored in a database. In most of the biometric methods, there is a threshold value which is scientifically defined. If the matching value is over this value the authentication is deemed as successful. The most known patterns that biometric devices use include fingerprint, retina, iris and face. The biometric authentication is the most expensive technique among the other authentication techniques. Hence the usage of biometrics is limited in the Internet. Today the most preferable application field of biometrics is physical access control.

II.3.3 Web Authentication

Web transactions constitute a large amount of Internet traffic. Access to the web is performed through special protocols such as HTTP. Sometimes it is necessary to restrict access to some web pages. This is done by utilizing several web authentication mechanisms. Let's see the most important ones:

II.3.3.1 Web Password Authentication

Most of the web server programs utilize some kind of password authentication mechanism. Now we will introduce a popular authentication scheme, the Apache web server password authentication:

In Apache, password authentication can be implemented easily. The mechanism is called “basic authentication”. Access to content protected by basic authentication in the Apache web server is indirectly governed by the password file. This password file is managed by a special utility called “htpasswd”. After passwords are created, it is assumed that passwords are given to the users using out of band channel (via email, phone). Web server should also define which web content will be password protected. This is usually done by using .htaccess file. The .htaccess file defines the authentication type and specifies the location of the relevant password file. The following definition depicts to restricting accesses to the “/home/yusuf/public_html/securearea” directory.

```
AuthName "Restricted Area"
AuthType Basic
AuthUserFile /home/yusuf/public_html/securearea
require yusuf mustafa
```

You will need to restart your Apache server in order for the new configuration to take effect, if these directives were put in the main server configuration file (i.e. httpd.conf). Directives placed in .htaccess files take effect immediately, since .htaccess files are parsed each time files are served. The next time that you load a file from that directory, you will see the familiar username/password dialog box pop up, requiring that you type the username and password before you are permitted to proceed. Note that in addition to specifically listing the users to whom you want to grant access, you can specify that any valid user should be let in. This is done with “Require valid-user” expression in the configuration [29].

In basic authentication scheme passwords are sent clear text which means they are vulnerable to eavesdropping attacks. So this mechanism should not be used to protect sensitive or valuable data. However, to use basic authentication over more secure protocols such as SSL will mitigate most of the attack risks.

In order to provide ease of use in the authentication mechanisms, some web sites store encrypted user passwords in cookies so that the user can make further logins to that site without any password. Because of the fact that cookies are stored on client side, a lot of security risks emerges. In this example any one who uses the same computer can easily log into the same password protected site without any authentication. Today we also know that cookies can also be easily captured and replayed back to the web sites [30].

In order to mitigate password authentication limitations, Franks et al. has proposed another type of authentication which is called **Digest Authentication** [31]. In digest authentication passwords are not transmitted directly. When Alice wants to authenticate herself to Bob, he takes a random number (nonce) from Bob. Then she concatenates the random number and her password and sends the hash value of this to Bob. After receiving the hash value Bob also makes the same process and calculates a hash value. Note that Bob knows the password of Alice that is a shared secret between them and also knows the random number. If the two hashes matches, authentication is deemed successful.

II.3.3.2 Single Sign-on

Basic authentication scheme is the most used web authentication application in the Internet. But classical username-password vulnerabilities are also affecting this authentication mechanism. As we have previously mentioned, the rate of the mistakes originated from the end users is very high. Users are often faced with the difficult and error prone task of maintaining a long list of usernames and passwords and they

mostly choose to use the same password for a number of different applications creating a lot of security risks (See Section II.3.1).

A single sign-on system (SSO) defers user authentication to a single, universal authentication service. Users authenticate themselves to the SSO once per session. Subsequently, each service requiring user authentication is redirected to a SSO server that vouches for the user. Hence, the user is required to maintain only a single authentication credential (e.g., SSO password). Note that the services themselves do not possess user credentials (e.g., passwords), but simply trust the SSO to state which users are authentic. While single sign-on services have been used for many years, the lack of universal adoption and cost of integration has made their use in web applications highly undesirable [32].

II.3.3.3 Secure Socket Layer (SSL)

Secure Sockets Layer protocol (SSL) was invented by Netscape Communications to include security in its browser products in order to make communication safe. SSL was originally intended for use with HTTP protocol used by Web servers and browsers but is now a significant component in all kinds of secure Internet communication.

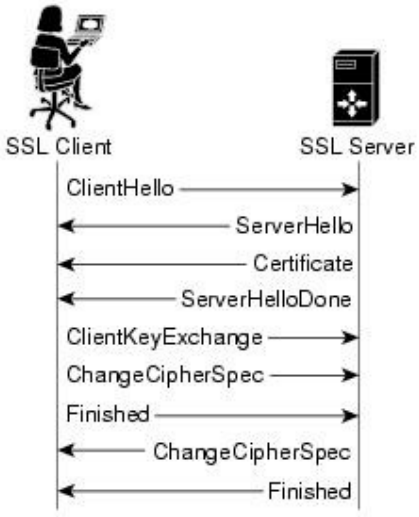

Figure 1 : SSL Handshake without Client Authentication

SSL has become the de-facto standard for secure Internet communication on the application level. There is a multitude of application protocols being run on top of SSL. The most well know is the HTTP protocol, commonly known as HTTPS when run over SSL, but many other protocols are commonly being run over SSL in order to provide security. SSL provides 3 main features [33]:

- SSL server authentication: This allows a user to confirm a server's identity.
- SSL client authentication: This allows a server to confirm a user's identity.
- An encrypted SSL session: All information sent between browser and server is encrypted.

Let's briefly look at how SSL server authentication and encryption work over web (see Figure 1 [34]). When Alice wants to communicate securely with Bob's server, she logs into the site by typing https rather than http. Alice's browser and Bob's server then run SSL handshake protocol, which authenticates the server and generates a shared symmetric key. In the SSL handshake process, the browser first sends its SSL version number and cryptographic preferences to the server. This is important because the client and server should negotiate which symmetric key algorithm they are going to use. After that, the server also sends its SSL version number, cryptographic preferences and also its certificate. This certificate includes RSA public key of the server and is certified by a Certificate Authority which means that the certificate was signed by the CA's private key.

Received the server's certificate, browser tries to validate the certificate and get the public key of the server. Each browser maintains a trusted CA list and the public keys of each CA. In order to validate the server's certificate, the browser checks its CA list. If the CA of the certificate is in the list, the validation process is started by using the public key of the CA. If not, a warning message as in Figure 2 is displayed on the screen. After successful validation, the browser ensures that the public key in the certificate really belongs to the targeted server. Browser then generates a random symmetric session key, encrypts this with the public key of the server and sends it.

With this message browser states that the ongoing session is going to be encrypted using this session key.

Figure 2 : Warning message when the certificate is not in the browser's trust list

Browser also sends a separate message indicating that the browser portion of SSL is finished. After receiving these messages, server first sends an acknowledgement that shows that it received the session key and is also going to encrypt the ongoing messages using this session key. The SSL handshake process is completed by a separate message sent from server indicating the server portion of the SSL handshake is finished.

CHAPTER III

PROBLEM DEFINITION

"To understand what the problem is the half of to solve it"
[Turkish Proverb]

Internet audit records are the most important data when the source of an Internet activity is to be traced back in order to reach the identity of the owner of an Internet activity and the machine on which the action took place. Especially for a criminal activity it is an obligation for system administrators to provide healthy records to the court to prove who has performed the unlawful activity in question.

The challenging question at this point is how one can be hundred percent sure to whom the log records belong and from which machine the activity is originated from. In our framework we call this process as **Source Authentication.** Note that source includes both the user who performed the activity located in the audit records and the machine on which the aforementioned action took place.

In this chapter we try to introduce the main reasons underlying why it is difficult to establish a successful source authentication.

III.1 Source Authentication Problem

As the number of the computer crimes increases, need for source authentication becomes much more important. Although a lot of security technologies do exist today, we observe that the problem still goes on and we are unaware of a previous study that proposes an effective solution to this problem when the literature is reviewed.

We will investigate the problem of source authentication in two different sections. One section is for source identity authentication and the other is about source machine authentication.

III.1.1 Source Identity Authentication

Source identity authentication is the process to establish the user identity of the owner of a specific log record. Source identity authentication can also be regarded as user authentication. We examine the problems of user authentication in following 3 sections:

III.1.1.1 Problem of Application Layer User Authentication

One of the widely used solutions is to implement user authentication at the application layer. This can be any authentication scheme that gets the authentication data from the user in the application layer e.g. username-password, biometric authentication. In this authentication type, because the trusted data is handled at the most top layer, it is very difficult to prevent an attacker getting access to a lower layer. As Gollman stated, an attacker with access to the "layer below" is in a position to subvert protection mechanisms further up [1]**.** For example once you gain system

privileges in the operating system, you will usually be able to change the fields of the TCP/IP packets such as IP Address before they are put on to the wire. Thus, no matter how complex user-level authentication is, an attacker, who has taken control of the system and has modified relevant processes, can mount a successful attack to break the security.

III.1.1.2 Problem of Weak User Authentication Schemes

In security, one of the most known tradeoff is the one between user convenience and security. The more security precautions you implement, the more you violate the user convenience. A security administrator should carefully inspect his network and after finishing the risk assessment and the other necessary analysis, he should create a "security policy" covering these analysis and their results. Because of the fact that absolute security is impossible, we can say that the security of an organization is adequate only if all the criteria stated in the security policy are met. This also means that there is no any other security risk that can not be managed by the organization.

So why do the security problems still exist in most of the organizations? The problem is that the security policy can not be properly created and managed because of the great lack in the number of security experts who is responsible for managing security in the organizations. Today we observe that there is not an information security department in most of the organizations. So as a matter of fact, weak authentication schemes are still widely deemed to have provided adequate security. The well known example is username-password scheme. As we have previously mentioned, because of the fact that username-password scheme is very easy to use and implement and also inexpensive compared to the other authentication schemes, it is the most widely used authentication scheme in the networks. Unaware implementation of username-password framework constitutes a lot of security risks which we have handled in section II.3.1.

III.1.1.3 Problem of Real Time User Authentication

All user authentication schemes are generally built on the idea of establishing the identity of the user mostly in access control and authorization frameworks. Today's user authentication schemes are appropriate for these reasons but not for systems with stringent real-time requirements. In this thesis our focus is the source authentication of the records in Internet audit logs. Internet audit systems, by their nature, are the systems recording the Internet facilities of the users in real time. If we think that a lot of Internet connections can be set up in a minute, the accuracy and effectiveness of the audit and authentication methods are gaining more and more importance. If any deviation in the time of records occurs, it is quite likely that a wrong identity can be detected.

Assume that we have no problem with the authentication scheme itself. It is as strong as we need. How can we provide real time authentication of users? Actually real time authentication is very difficult (if not impossible) by using only user authentication methodologies. We can implement solutions where the users are authenticated at the time of the Internet logons and perform reauthentication in a specific time interval. But no matter what kind of user authentication scheme we are implementing, to keep the authentication time interval as small as the seconds in a wide topology is infeasible in today's technology because of the extreme overload over the network resources.

III.1.2 Source Machine Authentication

User authentication is an indispensable part of source authentication, but as we have seen in the previous chapter it is not adequate for source authentication of Internet audit logs by itself. As we will explain, the desired solution is to supplement user authentication with additional mechanisms at lower layers.

Figure 3 shows we Internet stack layers, authentication schemes that can be implemented in these layers and various attacks to these schemes. Actually, the previous section source identity authentication or briefly user authentication is related with the top most layer of Internet stack, which is application layer. The lower level authentication schemes such as IP address based authentication, IPSEC, MAC authentication are the authentication types which are directly related with the machine. So we will examine all these authentication schemes under source machine authentication.

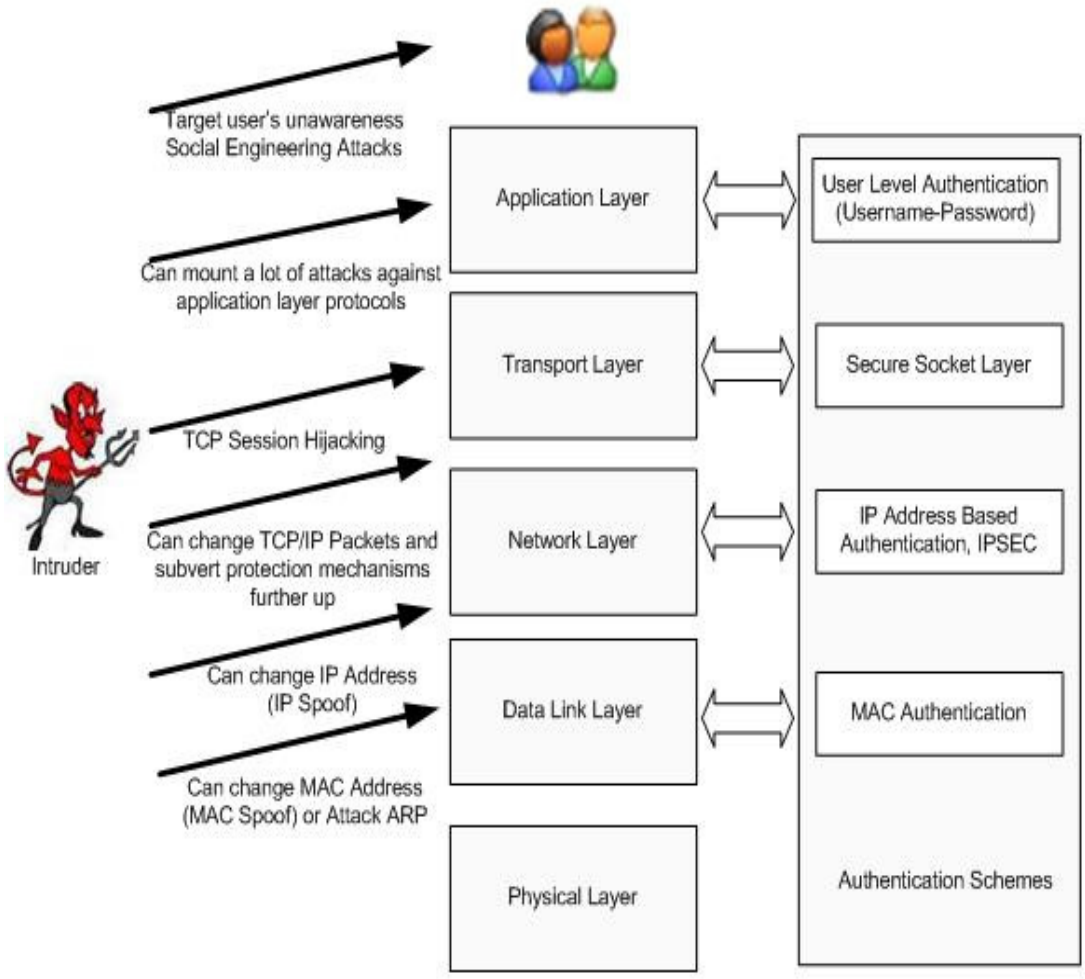

Figure 3 : Internet Stack Layers, Authentication Mechanisms and the Security Threats

The prominent reason to implement source machine authentication is to supplement user authentication with lower layer authentication. But there is another important reason. In a cyber crime case the digital evidence having been captured from the suspicious machine are sometimes the most important supporting materials in the time of trial. Although the evidence captured from the other network devices strongly point some decisions, the original crime machine should be definitely acquired for being analyzed later by computer forensics investigators. So for a cyber crime investigation in a local area network, detecting the crime machine successfully is a substantial issue.

Now let's see each authentication scheme respectively and discuss the pros and cons of them:

III.1.2.1 Address based Host Authentication

In order to prevent lower layer attacks, the idea of authenticating the system underneath the user-level is to be considered. Due to its low cost and convenience, many of today's network services use address-based host authentication which means that hosts often decide to trust other hosts based on their IP addresses [37,38]. Berkeley "r-commands" (remote login, remote shell, remote copy and etc.) is a notable example to these kinds of services [39]. If we look at Rlogin, it is a simple client-server based protocol that uses TCP as its transport layer. Rlogin allows a user to login remotely from one host to another, and, if the target machine trusts the other, rlogin allows the convenience of not prompting for a password. It authenticates the client via the source IP address instead [36, 40].

IP is the connectionless, unreliable network protocol in the TCP/IP suite. It has two 32-bit header fields to hold address information. IP's job is to route packets around the network. It provides no mechanism for reliability or accountability, for these, it relies on the upper layers [40]. Because of the fact that IP addresses were not

designed to provide authentication, an adversary can easily exploit the services by address impersonation. Address impersonation is technically called as **IP Spoof**.

IP Spoof can be performed in two ways. One of them is to use Proxy/Socks and the other is to edit the IP Packets. When a proxy server is used, the connections are made by proxy server instead of the original client. So that in the logs of the destination, only the IP address of proxy server can be seen. Editing the IP Packets is the real IP Spoof attack. It is basically done by sending special packets by changing the source IP address part of the IP header. IP Spoof attack performed by editing the packets is a kind of **blind attack**. If the attacker has not any chance of seeing the packets that are destined to victim's IP address (the one which attacker has impersonated.), to get the reply packets will be impossible. This kind of IP Spoof is generally used in the attacks where the reply of the packets is not an issue i.e. Denial of Service Attacks.

If the attacker wants to really impersonate someone by stealing all his session, than a more complex attack is needed. **TCP Session hijacking** is one of these type attacks. Broadly, it is done by predicting the sequence and acknowledgement numbers of the victim's packets. If properly implemented, it is feasible to completely steal someone else's session.

III.1.2.2 MAC Address Authentication

The desired solution might be to supplement IP with the address at a lower layer, MAC (Media Access Control) address. In order to determine the MAC address of an IP address, Address Resolution Protocol (ARP) is widely used in today's networks and especially in Internet [41].

ARP operates by sending out ARP request packets and receiving ARP reply packets. When a new MAC address of XXX IP address is to be requested, ARP request packet includes the question "What is the MAC address of the machine which has an IP

address of XXX". This request packet is broadcasted to whole network even in switched networks. The machine which has the IP address of XXX, replies this question with an ARP reply packet including its MAC address. To minimize the number of ARP packets being broadcasted, operating systems keep a cache of ARP replies. When a computer receives an ARP reply, it updates its ARP cache with the new IP/MAC association. As ARP is a stateless protocol, most operating systems update their cache if a reply is received, regardless of whether they have sent out an actual request [42].

The structure of ARP is very simple in order to provide efficiency, which makes it also vulnerable to a lot of security attacks. The main attack methods are the followings [42]:

- *ARP spoof:* It means to construct forged ARP request and reply packets. By sending forged ARP replies, a target computer could be convinced to send frames destined for computer A to instead go to intruder's computer B. ARP spoof causes man-in-the-middle attacks, sniffing in the switched networks, denial of service attacks, session hijacking etc.
- *MAC flooding:* By sending spoofed ARP replies to a switch at an extremely rapid rate, the switch's port/MAC table would overflow. Some switches can revert into broadcast mode at this point and sniffing can then be performed.
- *MAC cloning:* Normally MAC addresses are considered to be unique and burned into the ROM of each interface. Today, however MAC addresses can easily be changed [43]. An attacker can issue DoS attacks assigning himself the MAC and IP address of a target computer.

The systems using MAC based authentication must definitely take into account of these problems.

III.1.2.3 Cryptographic Solutions at Host Level

In some cases using cryptographic solutions might be seen as the best. When the source authentication is of concern, the IPSec protocol can be considered [3][8].

IPSec is a suite of protocols that provides security at the network layer and performs source authentication based on two principle protocols *Authentication Header (AH) protocol and Encapsulation Security Payload (ESP) protocol.* In both protocols, the source and destination hosts handshake and create a network layer logical connection based on either pre-shared keys or public key methods. Using pre-shared keys does not scale and incorporates the difficulty of maintenance the privacy of the key. Using public key authentication requires secure storage of the corresponding private key and trusted distribution of public keys. Public Key Infrastructure (PKI) is based on trusted parties that certify public keys of verified owners [44]. Public key infrastructure is a complex concept and not widely deployed.

IPSec also requires special-purpose client software, which in most cases replaces or augments the client system's TCP/IP stack. In many systems this introduces the risk of compatibility issues with other system software. Because of the way IPSec was created – generally lacking conformance to a standard – nearly all IPSec implementations are not compatible with each other [45].

Another drawback of IPSec, as well as other cryptographic solutions is the efficiency problem. There are practical limitations to use cryptography in real-time activities such as source authentication of Internet audit records because of its computational requirements.

CHAPTER IV

RELATED WORK

"If you reveal your secrets to the wind
you should not blame the wind for revealing them to the trees."
Kahlil Gibran

In the previous chapter, we have tried to obviously bring up the problem of source authentication in Internet access logs. In this thesis we try to overcome all these problems and design a system that will detect and block unauthorized Internet access in a local area network and validate the source information in Internet access logs. But before; we will take a look at the previous studies in the literature and discuss their shortcomings in this chapter.

This chapter is divided into two sections. One of them is related work; the other is the related method. The reason why we have also analyzed related methods besides related work is that the efficiency of the methods used in the systems is very important because we try to tackle with the source authentication in real time Internet audit records.

IV.1 RELATED WORK

When the previous studies are examined, we see that they mainly concentrated on the lower level authentication problems. This is because a lot of different user level authentication schemes are available today ranging from the most basic one such as username-password to complex ones such as biometrics, but it is very difficult to find systems offering an efficient and secure low level authentication solution.

As we have stated in previous chapters, most of the vulnerabilities in the lower level authentications such as IP Address Based Authentication, MAC Address Authentication are originating from the inherited security deficiencies in the structure of Internet protocols. This is actually the result of that the efficiency and simplicity are the main concerns and given more importance than the security at that time.

Now let's see what the previous studies have proposed us and their shortcomings in the following sections:

IV.1.1 Secure Address Resolution Protocol

A Secure Address Resolution Protocol has been proposed by Gouda and Huang in 2002 [46]. This protocol consists of a secure server connected to Ethernet and two sub-protocols: an invite-accept protocol and a request-reply protocol. Each computer connected to the Ethernet can use the invite-accept protocol to periodically record its IP and hardware address in the database of the secure server. Each computer can later use the request–reply protocol to obtain the hardware address of any other computer connected to the Ethernet from the database of the secure server. The server shares unique secrets with each computer on the Ethernet. The two sub-protocols provide the security of the messages transmitted between the server and clients by calculating some hash values using these unique secrets and some nonce values.

First of all, we note that this protocol does not use ARP caches, which means that in any host-to-host communication, a query to the server is required. Furthermore, in every communication a message digest algorithm is executed to calculate the hash values, which means additional real-time inefficiency no matter how efficient the algorithm is implemented (similarly IPsec has this drawback). In the invite-accept protocol, the secure server sends a periodic invite message to every host on the Ethernet every T seconds to gather the MAC and IP addresses of the machines in a secure way. This process is rather long and spends the bandwidth by generating a lot of network traffic.

Another issue is the maintaining the secrecy between the clients and server. The protocol has the inherent problems of secret key cryptography like key distribution. Gouda and Huang have stated that these shared secrets can be renewed once in a long period, for example a month. Assume that an intruder from a local network has registered himself to this system and obtained a shared secret. At the next request from the server, the intruder can send fake MAC and/or IP addresses by adding a message digest. When the server receives this information it checks the digest and using the shared secret confirms that this information is true. The server fully trusts and updates its database with the new record. This is all an adversary needs to perform ARP spoofing through the whole month.

IV.1.2 Static ARP Tables

Another proposed method is to use static ARP tables, which means permanent entries for trusted hosts are stored in the ARP tables of all computers in a LAN [47]. This is not a practical solution especially in large networks because when an IP address of a machine changes or a new machine is added, ARP tables of all computers must be updated. Static ARP tables also do not support the detection and blocking of malicious attempts.

IV.1.3 ARPWATCH

Aside from these two proposals there is another mechanism called ARPWATCH [48], which is a bit similar to our proposal. ARPWATCH is a free UNIX program which sniffs for ARP replies on a network and generates a table consisting of IP/MAC associations. When an IP address associated with the MAC address changes, it logs and sends an e-mail to the administrator. If we take a look at the drawbacks of ARPWATCH, we see that it creates the initial table itself automatically which is a very unreliable method. When an adversary joins to the network with a wrong MAC or IP address or an adversary already defined changes his MAC and IP address at the same time, the system could not detect this action and logs this as a new machine having joined to the network. Another issue is that ARPWATCH uses sniffing method which requires a high configuration for a machine to catch all the ARP activities in real time especially in large networks. The last but not the least, ARPWATCH provides only detection but we need also blocking in our system.

IV.1.4 MAC Binding on the Switch

Another related work with the idea of preventing MAC address changes was by Beck in 1999 [49]. Using Kerberos, Beck's system incorporates user authentication into the system. MAC address changes are prevented by binding each port of the switches to one Ethernet address. When a problem such as IP address change occurs during the authenticated session, the session is canceled and the user has to reauthenticate himself. This may be seen as a partial solution but not a full blocking mechanism because when the user authenticates correctly himself again, he can continue using the Internet. This system is also not so user-friendly in management aspects. When a new computer is to be defined, its MAC address should be defined to the switches manually.

Another drawback of Beck's system is that it uses Kerberos as a user level authentication. Kerberos, in most cases, requires special software or modifications on client's side and you can only use workstations that are specially configured to run the exact version of Kerberos.

IV.2 RELATED METHODS

As we are going to see in the next section, we propose a novel unauthorized Internet access detection and blocking system validating the source information in Internet access logs. In this system, we remove the need for real time authentication in Internet access logs, by preventing possible attacks to the authentication mechanisms in all layers of Internet stacks. By this way, we ensure that neither user nor machine identity can be changed after the first authentication, thus, we do not need to worry about implementing an authentication mechanism in a real time manner which is infeasible because of the efficiency problems. But now there is a second problem of how to find an effective methodology to detect forges in authentication mechanisms.

In order to detect unauthorized Internet accesses, we propose a subsystem called UNIDES which detects the MAC-IP forges in the network. We do not move into detail about UNIDES now (see chapter 5), only discuss the methodology that UNIDES uses and the other related methods. In order to perform detection, UNIDES should be informed with corresponding MAC addresses of the IP addresses in the network periodically. The important point is here how to detect MAC-IP address pairs in a network efficiently. We can say that the more efficient methodology we implement, the more we approach to real time responsiveness.

This can be accomplished using different methods. When previous studies are examined, we see that the most used techniques are based on a network program that sends a broadcast message to all active machines in the network and gathers their MAC and IP addresses as responses. This technique has some serious drawbacks

especially in terms of efficiency which is one of the important criteria in most of the security applications as well as in UNIDES.

The first issue affecting the efficiency is the fact that a lot of broadcast traffic spreads into the network. The second issue is the time spent during the information gathering process. This issue is also an essential one in terms of real time responsiveness which is the indispensable feature of these systems because the goal is to detect unauthorized actions almost instantly at the time they occur and react accordingly just after. To cope with this stringent requirement we should keep the information gathering time very short.

To speak in more precise terms, two network programs ("Knowlan" [50] and "R3x" [51]) are analyzed. Knowlan sends ARP REQUEST packets to the LAN and receives ARP REPLY packets from the currently running machines. As a result, by time it gets all IP and MAC address pairs of all online machines. On the other hand, R3x scans large networks by sending UDP query status to every IP and wait for responses in which MAC address of that IP can be found.

We established a small experiment to determine how much time each program spends to find all the MAC-IP pairs in a LAN. Programs are run in a network having IP's range between 192.168.0.1-192.168.8.254. We found that Knowlan had finished scanning in 51 seconds and R3x had finished scanning in 41 seconds as shown in Figure 4.

After these performance results, we conclude that utilizing these network programs does not satisfy the high efficiency and real-time functionality requirements of our system. We should come up with an idea that would not overwhelm the network. After some researches, we see that the solution we are looking for is not so complex. Why not to use the ARP table of a machine to know all the IP-MAC addresses of computers that have an access to the Internet at the given time? If this is achievable,

in order to obtain the MAC-IP pairs, it is adequate to run only a query command to the ARP table, the result of which can be reached in 1 or 2 seconds.

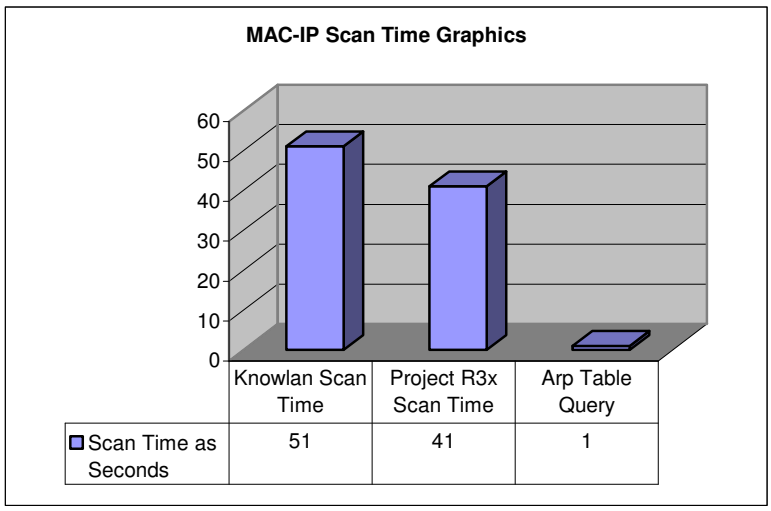

Figure 4 : MAC-IP Scan Graphics

An important issue here is that we should set up a structure so that all the MAC-IP address pairs can be reached in one system's ARP table. The practical solution is to determine a central system (generally referred as a gateway or a proxy) and construct a topology that compels all the local users to pass through it in order to have an access to the Internet.

To summarize, we report the following advantages of using ARP table of a central server instead of using a network program in order to obtain IP-MAC addresses of the computers connected to Internet in a network:

- There isn't any traffic through the network wasting the bandwidth because the system scans its own ARP table.
- Since getting the MAC-IP pairs takes only a few seconds, we can respond almost in real-time.

CHAPTER V

OUR PROPOSED SOLUTION

"In science one tries to tell people, in such a way as to be understood by everyone, something that no one ever knew before. But in poetry, it's the exact opposite"
Paul Dirac

In this chapter, in order to solve the problem of source authentication in Internet access logs, we propose an unauthorized Internet access detection and blocking system validating the source information in Internet access logs which is formed by two different but complementary subsystems. The first system is ***UNIDES: UNauthorized Internet Access DEtection and blocking System*** which has a specific proxy-based architecture to solve the problem of source authentication in low levels without any need to modify the underlying network protocols. UNIDES is a highly efficient real-time security system based on MAC authentication but it prevents the attacks originating from the vulnerabilities of ARP protocol. Firstly, every user is registered into the system by assigning them a constant IP address recorded together with his machine's MAC address. Whenever a user changes his/her IP address, the system automatically detects and blocks his Internet access. The goal of this thesis is not to provide a full ARP security solution or to detect local attacks, but instead, to provide reliable Internet audit records and enforce a security policy integrated in a topology based on advanced switches. Integrated to ease the administration, there is also an interface in UNIDES to manage all the relevant security and administrative processes.

The second system is called **SIACS: SSL Based Internet Access Control System to provide User Level Source Authentication in Internet Access Logs**. In this system each user who wants to have an Internet access is assigned a username-password and also a digital certificate. By default none of the users is allowed to have an Internet access by the firewall. Whenever a user wants to be connected, he has to first login to a web based Internet access control program and give his credentials (username-password and digital certificate). Only when he is authenticated by the system, Internet connection can be established. In this framework SSL Client Side Authentication is implemented together with the username-password authentication. Internet remains active for that user unless the user logs out the system. The system also has a timeout value. If the computer of the user is shut down without a logout, the system automatically detects that user and blocks his Internet connection. The SSL User Certificates are stored in a .p12 file (in PKCS12 format). This file is password protected and all passwords in the framework of SIACS are only known by system administrators. When a certificate is to be installed to the client's machine, system administrator installs it by entering admin password of the file so that the certificate can not be moved from one computer to another because of the password protection. This provides us to bind each machine with one particular user identity.

Now let's have a closer look at to each proposed system and see the details:

V.1 UNIDES-Unauthorized Internet Access Detection and Blocking System

In this section we present UNIDES (**UN**authorized **Internet** Access **De**tection and blocking **S**ystem). UNIDES concerns with the low level authentication part or let's say machine authentication part of the source authentication problem and offers an efficient methodology for the detection of unauthorized Internet accesses almost in real time.

The rest of the section is organized as follows. Section V.1.1 gives an overview of the framework model for describing UNIDES. Operation and management of UNIDES is examined in section V.1.2 and V.1.3. We highlight the advantages of UNIDES in section V.1.4. In section V.1.5, we end by summing up our work.

V.1.1 The Framework Model of UNIDES

UNIDES is not only a real time security system that detects and blocks unauthorized Internet access, but also a very efficient intranet management system. Before explaining the operation of UNIDES, we describe the various entities in our framework model.

V.1.1.1 Topology

The topology of the underlying network and the entities taking active role in the operation of UNIDES is illustrated in Figure 5. We believe it is reasonable to expect similar scenarios in most of the intranets currently in use today.

As seen from Figure 5, there are three main zones in the topology; local, DMZ and Internet. In the central point of these three zones, firewall (also gateway) is located. In local zone we need advanced switches, a proxy server, a DHCP server and a database. In our framework, we assume proxy and DHCP servers run on the same machine. On this machine we have also a web server, because our management interface would be a web based one.

To give you an early idea of our system, here we would like to mention also the software programs we would need. For detection phase, some shell scripts are needed to gather the MAC-IP pairs from the ARP table and compare them with the records in the authorized user database.

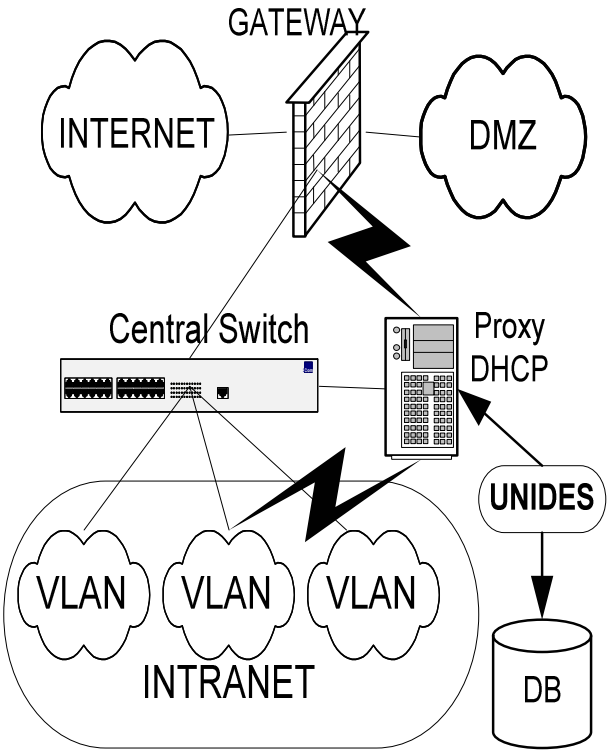

Figure 5 : The Topology of UNIDES

For blocking phase, we need extra programs to be able to manage switches' port security mechanisms. In UNIDES we use Expect Script Language [52] which eases our job since connecting to switches from a script and making any configuration we want becomes so straightforward.

Let us now show the entities taking active role in our framework in more detail:

V.1.1.2 Proxy Server and Firewall

In UNIDES we use proxy server as a central point through which all the users must pass in order to have an Internet connection. Proxy servers were initially used to increase the efficiency of Internet connections by caching the web pages accessed before. Today's proxies are much more functional providing some security

mechanisms such as URL filtering, audit systems and much more. We can perform a lot of controls using a proxy system. To understand the usage of proxy as an audit system, let us briefly examine the operation of squid [53], an open-source proxy software mostly used on UNIX systems.

There is a file called "squid.conf" where all the configurations of proxy can be performed. In this file, there is a definition like:

 # cache_access_log /usr/local/squid/logs/access.log

This line shows the path and name of the squid log file including all the records showing which IP addresses connected to which URLs on which date and time can be found in the following format:

192.168.2.105 - - [05/May/2004:19:08:18 +0300] "GET http://www.abc.com.tr/images /kapak/ipucu.jpg HTTP/1.0" 304 478 TCP_MISS:DIRECT

In UNIDES our central point will be the proxy server in order to capture all the MAC-IP pairs in the network. But the problem has not yet resolved. We must compel all the users to use this proxy which can be done by using several methods. To name a few; transparent proxy, port forwarders or firewall policies are all possible and effective. We would use the firewall policy and add a rule to our firewall that blocks all the packets from the local network except the packets coming from proxy. Hence, anyone who does not use proxy will not have an access to the Internet. This implies that to have an access to the Internet, every user must first be connected to the proxy which also means every IP and MAC address pairs must be at least once entered to the proxy's ARP table.

Let's look at how we can view the ARP table in a UNIX system:

```
# arp –an
(192.168.2.11) at 00:04:45:d0:4f:d5 on em0 [ethernet]
(192.168.2.16) at 00:02:a7:28:b5:33 on em0 [ethernet]
(192.168.2.19) at 00:c4:35:65:c1:a5 on em0 [ethernet]
```

V.1.1.3 Advanced Switches and Port Security Mechanism

Our experience and the previous studies have showed us that by using only higher level software solutions, it is very difficult to defend against the attacks originating from the vulnerabilities in the ARP protocol. Today, MAC addresses, which were considered as unique and permanent, can be easily changed in a few seconds. So a mechanism is needed to prevent changing the MAC addresses which can be either protocol based or hardware based. In UNIDES, we choose the second approach in order to build our system on top of the available protocols and systems.

The best hardware solution to prevent MAC address changes and ARP attacks is MAC binding on the ports of the switch. This is also called the “port-security” mechanism. Port-security enables us to define the MAC addresses of the machines connected to the ports of the switch. Each port has a different MAC table. Note that this mechanism is not supported by all the switches. We call a switch having this property as “advanced switch”.

In our solution we construct our topology on this kind of switches. The drawback of this mechanism is that when a host is to be connected to the Internet, the MAC address of that host must be defined in all of the switches on the way to the Internet. But in UNIDES this problem has been resolved by using a management interface. When a user is to be defined, the administrator enters only the required information and only once i.e. he does not concern about which switches or on which ports the MAC address should be defined. Other boring and tiresome functions to define that MAC address on the switches can be performed by the system because the topology is already known.

In UNIDES we use advanced switches not only to prevent MAC address changes or ARP attacks but also in blocking the detected unauthorized Internet accesses.

Let's look at basically how we can define MAC address 00:20:18:b8:7a:5b to the port Fa0/2 on a Cisco [20] switch:

Central-Switch#conf t
Central-Switch(config)#int Fa0/2
Central-Switch(config-if)#no switchport port-security
Central-Switch(config-if)#switchport port-security maximum 1
Central-Switch(config-if)# switchport port-security mac 0020.18b8.7a5b
Central-Switch(config-if)#switchport port-security

So in the example above only the MAC address 00:20:18:b8:7a:5b can pass through the port Fa0/2 of the switch.

The advantages of port security mechanism can be summarized as follows:

- Someone connecting his computer cable to the network without giving his MAC address to the system administrator is not permitted to connect to the Internet. This would provide a good central administration and prevents unknown users joining to the local network.
- The attacks such as ARP spoofing, MAC flooding, MAC cloning, etc. would be prevented and as a result some malicious actions like sniffing can not be performed.
- Each machine can only log into the Internet through the port on which its MAC address is defined. And by binding the IP addresses to the MAC addresses and preventing any change, we ensure the machine identity of a specific IP address (remember source machine authentication problems in section III.1.2).

V.1.1.4 DHCP Server

We need also a DHCP server [54] in our network. The main function of a DHCP server is to deliver automatic TCP/IP configurations to the users. In our framework, we benefit from static IP-MAC address configuration in DHCP. Let's briefly look at how we can define an IP-MAC address pair on a UNIX DHCP server:

host YusufUZUNAY-5256 {
hardware ethernet 00:20:18:b8:7a:5b;
fixed-address 192.168.4.28; }

So according to this configuration if the host having a MAC address of 00:20:18:b8:7a:5b requests an IP address from the DHCP, server will always assign the IP address of 192.168.4.28 to that host.

V.1.2 Operation of UNIDES

UNIDES is based on two main daemons running on proxy server. One of them is the management daemon which is used to define, delete and update user's information in the related network systems (switches, dhcp, proxy), and the other is the detection daemon which is used to detect the unauthorized Internet accesses and trigger the other programs for blocking.

Let's take a closer look at how a user is registered to UNIDES. After the required information is entered to the system using the management interface of UNIDES, system determines an IP address for that user from the unused IP addresses table and initiates data input modules. First module enters these data including the assigned IP address and MAC address into the database and second module enters the data into data file placed in Proxy-Dhcp server. After the input process, the second module sets the value in the action file to "1" which would be used to trigger the management

daemon (See Figure 6). The functions up to this point are initiated by the Web interface.

On the server side two main daemons running at background watch the values of records in some action files. When the management daemon notices the value change in the action files and if this value is set to “1” representing data input, it takes the data from the data files, adds a configuration into the dhcpd.conf file and initiates the Expect scripts with the required parameters. Expect scripts make the necessary configuration in the switches. This is the final step of data input part of UNIDES. The other processes like data update and data deletion are performed similar with the data input process.

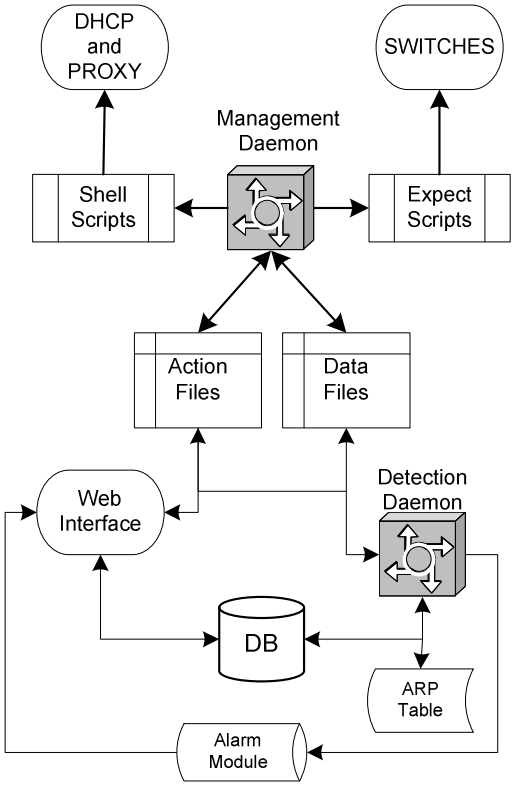

Figure 6 : Operational scheme of UNIDES

In detection phase, a daemon called detection daemon runs on the proxy server. This daemon checks the ARP table and compares all the MAC-IP pairs with the records in the database. If an unmatched record is found, daemon edits the data and the action files. Then the shell and Expect scripts run and perform the required functions to delete the relevant record from the related switches, proxy server, DHCP server and from the database. After deletion processes, a record indicating an unauthorized Internet access is entered into the detected IP database. In the web interface an alarm module frame updates itself continuously and checks if there is a new record in the detected IP database or not. If there is, system alarms to warn the administrator and prints out the records including the detected MAC addresses.

V.1.3 Management Interface

The management interface of UNIDES is based on a web application running on the proxy server and it provides a user-friendly and efficient network management mechanism integrated with security.

The main services provided by management interface are these:

1. *Data Input:* The information like user id, name, surname, department, MAC address are entered into the database by means of a data input form (Figure 19 in the Appendix). Administrator only fills this form and presses submit button. Then the rest of the process to enter this information in the relevant systems is automatically performed by UNIDES. For example it is not needed for the administrator to know which constant IP address he should give to user or on which port of the switches the MAC address should be defined. There is an IP management mechanism in UNIDES, that knows which IP blocks belongs to which departments, and maintains IP pools for each department. UNIDES also knows the topology so that it can determine on which switches or on which port of the switches the MAC address is to be defined.

UNIDES also supports an error detection mechanism in data input process. When a new record is entered from the data input form, UNIDES runs a procedure to initialize the configuration modules on the server. These modules configure the switches and the DHCP server. There are two columns in the record table called Switch and DHCP indicating possible errors in data input process. While the configuration modules are running, the information of the new record is seen on the screen in a table and the switch and DHCP columns indicate that data input process is going on (Figure 20 in the Appendix). When an error has occurred in either DHCP or switches, the columns indicate it, accordingly (Figures 21,22,23 in the Appendix).

2. *Data Edit:* By the help of management interface of UNIDES, data update processes can be carried out in a very short time. As we know, the more security mechanisms you have, the more difficult it is to manage the network. Assume that a user has changed his department which is connected via another port on the central switch. The following actions should be performed manually to update the system if UNIDES is not used.
   - Detect which switches and which ports of those switches are being used for the connection of that user.
   - Connect to the switches and learn the maximum numbers of the ports.
   - Disable port security mechanism on those ports.
   - Enter those ports and delete the MAC address from all of them.
   - Decrease the maximum by 1 and enable port security mechanism.
   - Detect which switches and which ports on those switches are to be used for the connections of the user's new department.
   - Learn the maximums of the new ports.
   - Enter those ports and disable port security mechanisms and increase maximum by 1.
   - Add MAC address on to those ports and enable port security mechanisms.
   - Determine a new IP for the user searching the IP records of the databases.

- Connect to the DHCP server, find the configuration of the user and change.

As we see this requires a very long and boring process that affects the efficiency of network management. UNIDES performs all these actions in once click, instead.

3. *Data Delete:* For deletion of a record from all of the relevant systems, the administrator only finds the record from the web interface and click delete (Figure 24 in the Appendix).

4. *Data Search:* The interface provides to search all the records by giving different keywords (Figure 25 in the Appendix).

5. *Management of Detected Records:* When an unauthorized Internet access is detected, the records of that MAC address are deleted from the systems and entered into the detected user database with the information on which date IP address is changed. If the system administrator decides to define this user again, it is enough only to click the activate button from the management interface (Figure 26 in the Appendix).

6. *Alarm Mechanism:* The interface is running on a page formed by two frames. The upper frame provides the management facilities and the lower provides alarming mechanism (All Figures in the Snapshots of UNIDES in Appendix). When an unauthorized Internet access is detected, its record including the MAC address of the machine used is added to the detected user table. The lower frame refreshes itself periodically and checks if there is a change in the number of the maximum records in the detected user table. If there is, system gives alarm and prints out the information of the detected users.

V.1.4 Advantages of UNIDES

UNIDES has an integrated architecture which has a lot of advantages especially with respect to security and network management.

V.1.4.1 Security

The architecture of UNIDES built on advanced switches provides the following advantages regarding network security:

- Prevents users changing their IP addresses or MAC Addresses.
- Supports source authentication in Internet access logs.
- Prevents ARP based attacks such as ARP spoofing, MAC flooding, MAC cloning.

The management interface of UNIDES itself also implements some security mechanisms. For instance there is an admin based control mechanism. In other words, using UNIDES one can define different admin levels each of which has different permissions on the management program. For example one super admin can change all the records, the other can only change one department's records or another admin can only see the records but can not make any changes at all.

In order to be able to analyze the auditing of the management interface itself, every action is recorded to the database such as which admin performed which activity on which date and at which time. So we can fully see one record's life circle. Another important security mechanism of the management interface of UNIDES is the deleted records table. When a record is deleted from the system, in fact it is not really deleted. That record is taken to the deleted record table. This provides us to trace back the past records of an IP address such as in a certain time at the past, to whom this IP address was assigned.

V.1.4.2 Intranet Management and Performance Analysis

UNIDES also provides a very functional intranet management mechanism. Because all the local users are registered into a central database, the user information can easily be managed and the system configurations like DHCP or advanced switches can be easily done by using this management interface.

As we have mentioned before, in UNIDES the administrator only communicates with the web interface by entering some information from the forms or clicking to some buttons. All the other functions and configurations are performed by the system. We have carried out some experiments to determine the performance gain of the UNIDES. We perform some Data Input, Data Update and Data Deletion tasks using manual ways[1] and using management interface of UNIDES. In the experiments, only one advanced switch is configured. Let's look at the results of our experiments:

Table 1: Measurement of Data Input using UNIDES

Test 1	68 seconds
Test 2	71 seconds
Test 3	69 seconds
Test 4	58 seconds
Test 5	72 seconds
Avarage	**67.6 seconds**

Table 2: Measurement of Data Input using manual ways

Test 1	294 seconds

[1] Manual ways mean the manual configuration of all the systems used in UNIDES by a system administrator having enough experience.

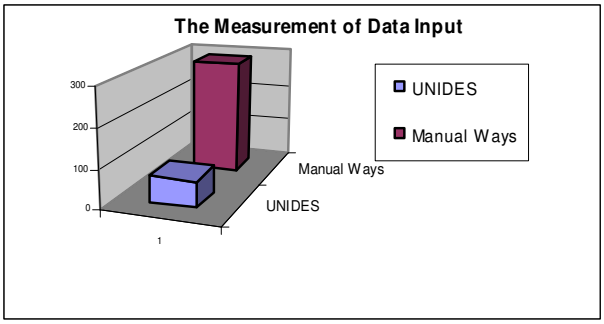

Figure 7 : The Time Graphic for Data Input

Table 3: Total Performance Gain in Data Input by using UNIDES

Total Performance Gaining	77 %

In data update experiment, we have chosen to change Vlan and ID No information of the records because this update requires changes in the configuration of switch, DHCP and also proxy.

Table 4: Measurement of Data Update using UNIDES

Test 1	42 seconds
Test 2	46 seconds
Test 3	48 seconds
Test 4	62 seconds
Test 5	50 seconds
Average	**49.6 seconds**

Table 5: Measurement of Data Update using manual ways

Test 1	328 seconds

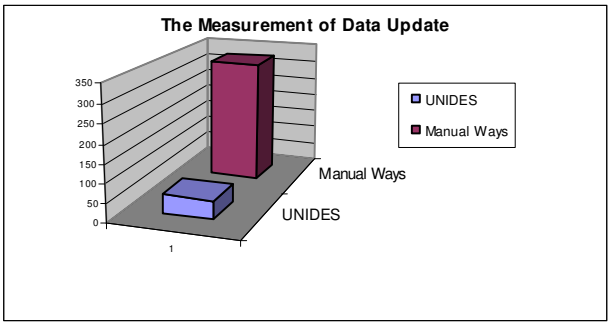

Figure 8 : The Time Graphic for Data Update

Table 6: Total Performance Gain in Data Update by using UNIDES

Total Performance Gaining	84.87 %

As the third and last experiment, we perform the data deletion task.

Table 7: Measurement of Data Deletion using UNIDES

Test 1	25 seconds
Test 2	19 seconds
Test 3	28 seconds
Test 4	18 seconds
Test 5	17 seconds
Avarage	**21.4 seconds**

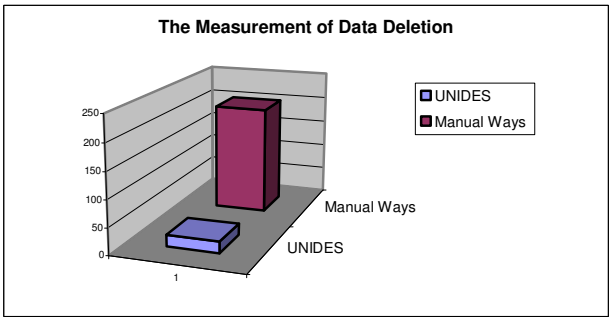

Figure 9 : The Time Graphic for Data Deletion

Table 8: Measurement of Data Deletion using manual ways

Test 1	203 seconds

Table 9: Total Performance Gain in Data Deletion by using UNIDES

Total Performance Gaining	89.45 %

V.1.5 SUMMARY

In this section, we have presented a highly efficient real-time security system called UNIDES to detect and block unauthorized Internet access. UNIDES is based on a proxy-based architecture and it supports source authentication of Internet access logs in the low level.

In another view, UNIDES can be regarded as a hardware based model incorporating advanced switches with port security mechanism into the proposed solution. By the help of these switches we have prevented the ARP attacks. For source authentication we have associated the MAC addresses of the users to the constant IP addresses which are then entered into a central database with the additional user information. We have prevented these IP addresses to be changed by the detection and blocking mechanisms of UNIDES.

In the detection mechanism, ARP table of a central system (proxy) through which all the user must be connected in order to have an access to the Internet is used. This architecture has provided us the efficiency and real time functionality in the process of finding all the MAC-IP pairs of the connected users. The system periodically verifies the records in the ARP table by comparing them with the authorized records located in a secure database. Whenever an unauthorized Internet access is detected, the Internet access of that machine is being blocked by deleting all of its records from the switches and from the servers.

The drawback of UNIDES may be the need for a specific architecture and the need of administration of all the relevant activities. In order to ease the administration we have integrated a web based management interface to UNIDES which might be also considered as an intranet management system because all the configurations of the servers, switches and the user database can be performed using it.

V.2 SIACS: SSL Based Internet Access Control System to provide User Level Source Authentication in Internet Access Logs

In the previous section, UNIDES is introduced as a system model that performs MAC address authentication by restraining threats originating from the inherited vulnerabilities of Internet protocols. Considering that a computer is used only by a specific user, we may assume machine authentication also provides user authentication. By this view, we can say that UNIDES is very effective source authentication solution itself especially in the places where robust physical authentication of personal computers is of concern. UNIDES is also a very user friendly system that does not bother user with extra authentication mechanisms. The only think that user does in order to have an Internet access is to login his personal computer. So why do we extend our system with SIACS which is an extra system on top UNIDES providing user level authentication? The answer can be summarized as follows:

- In order to be able to use our system also in other places such as laboratory environment where a computer is used more than one user.
- In the first version of UNIDES (without SIACS), the security of personal machine is the responsibility of the user. In this scheme if an adversary succeeds to logon to the PC of another user, he can directly gain access to Internet because there is not an extra user level authentication mechanism in place.

So we have decided to develop also SIACS, SSL Based Internet Access Control System that provides user level source authentication in order to supplement UNIDES for providing a robust source authentication.

Now, we are going to first introduce you the framework of SIACS and then move into details how it operates.

V.2.1 Framework of SIACS

SIACS provides user level authentication by implementing both SSL client side and username-password authentication schemes. it is also a topology specific access control system such as UNIDES.

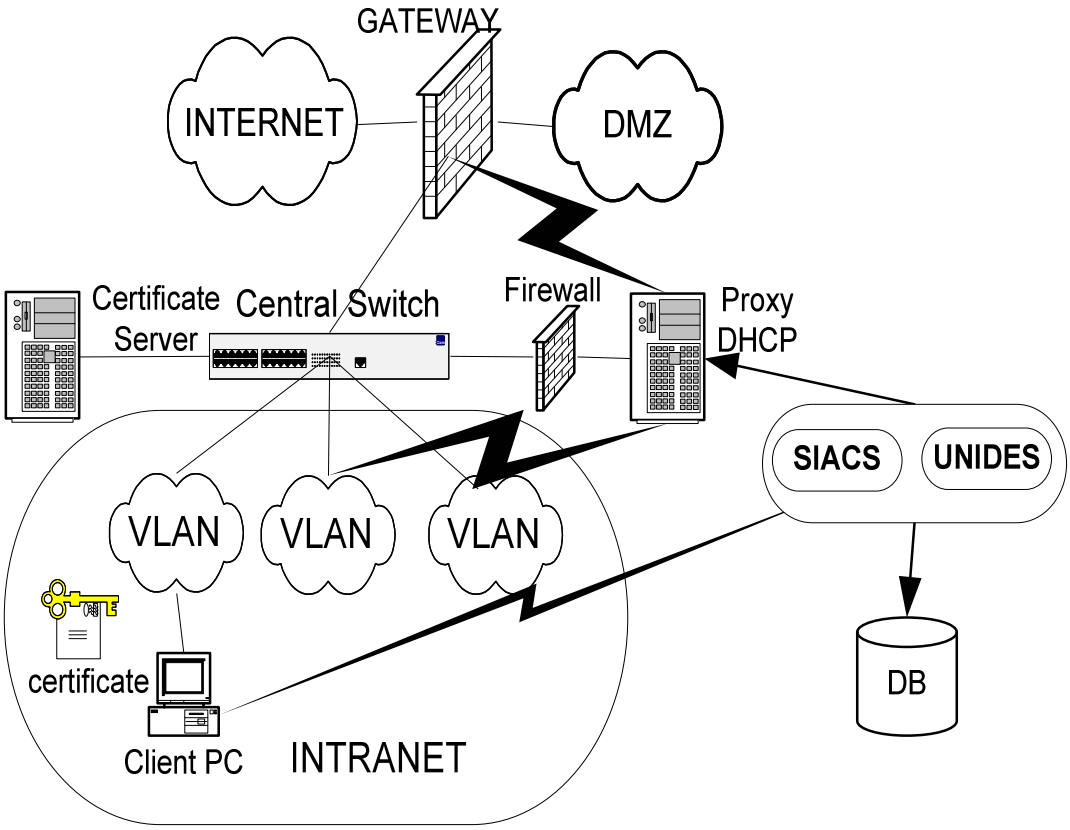

Figure 10: The Topology of SIACS

SIACS is located on top of UNIDES in order to supplement the low level authentication with user level authentication and provides a robust solution to source authentication problem. The topology of SIACS is similar to the topology of UNIDES. Only some extra entities such as Certificate Server and a second firewall in front of the proxy server are added (See Figure 10).

SIACS is a web based access control system written in PHP and UNIX shell script. It runs on the proxy server as UNIDES does and uses the same database. Different from UNIDES, when making web configuration in SIACS, we extra configurations are necessary in order to support SSL client authentication (See Figure 11).

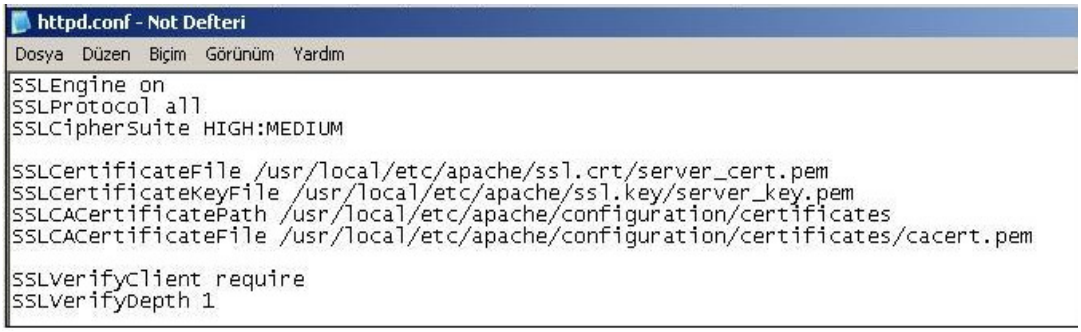

Figure 11: SSL Client Authentication Configuration in SIACS Apache Web Server

Now let's give some detail where and why we have used the extra entities in the topology:

V.2.1.1 Firewall in front of the Proxy Server:

When implementing UNIDES, remember that we have constructed a specific structure where each user should first connect to the proxy server in order to have an Internet access. This rule is still active in the topology of SIACS. But this time, we also need another blocking mechanism which will allow users to be connected to Internet only when they are authenticated to SIACS web based interface.

At first glance, it might be considered to implement blocking on the central firewall (also gateway) in order to avoid the requirement for an extra system. But it is not feasible because of the following reason; In order to block or give access to the authentic users, we first need to detect the IP address of the user's machine and then create rules utilizing this IP address. In such a case, if we implement blocking on the central firewall, we are not able to differentiate the users' IP addresses because of the fact that every user first connects to the proxy before they go to gateway. All the incoming connections in the firewall of the gateway comes from the proxy server. So we have decided to implement this blocking mechanism on an extra firewall running in front of the proxy server (see Figure 10).

In this firewall, by default, every connection that goes to port 8080 (proxy service port) of the proxy server is disallowed indicating that nobody has a proxy access. It also means that nobody can have an Internet connection because of not being able to use proxy. Whenever a user has a success authentication, a rule that permits the IP address of the user's machine to reach to the port 8080 of proxy server is added before the blocking line in the firewall, henceforth Internet connection is activated. Whenever the user logs out, the associated rule for that user is deleted from the firewall and Internet connection is blocked.

V.2.1.2 Proxy Server

SIACS runs on proxy server in our scheme. This is not an obligation, the design allows other scenarios that can be implemented. The reason why we have preferred proxy server is to keep all our programs (UNIDES, SIACS) together in one server. Proxy is a key element in our solution because both UNIDES and SIACS heavily uses it. UNIDES uses it as a central point to get MAC-IP pairs from its ARP table and SIACS uses it as an Internet access control mechanism.

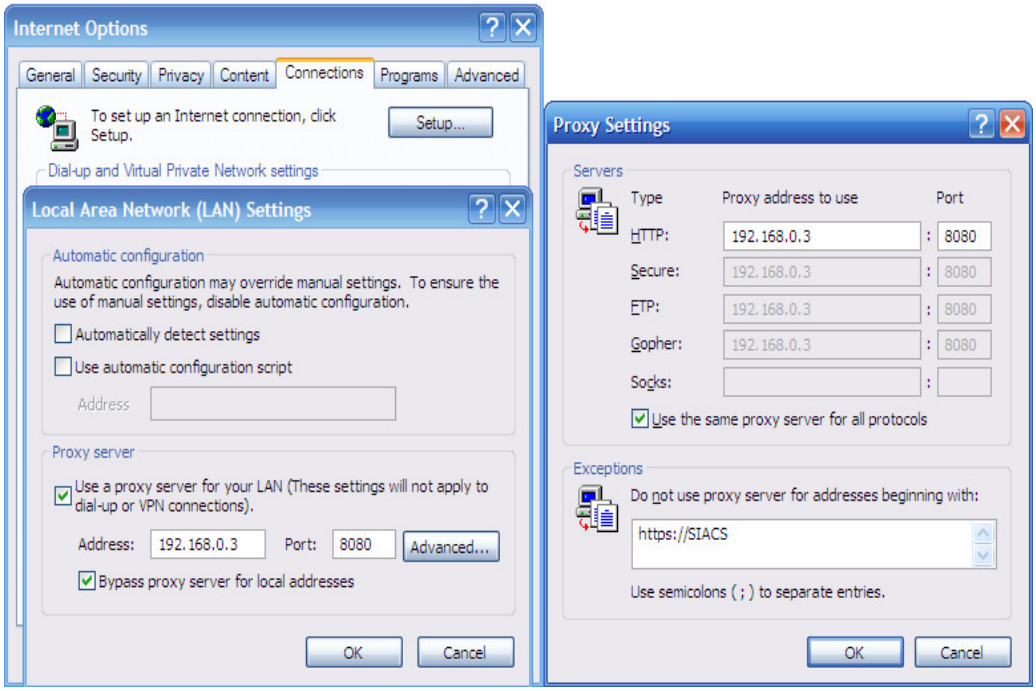

Figure 12: Client Proxy Settings for SIACS

When implementing the blocking scenario mentioned in the previous section, we ran into some difficulties: Since all the users' browsers were configured to use proxy, users could not also login to SIACS web based interface. Because their connections try first go to proxy server. As it is seen in Figure 12, we have solved this problem by configuring the browser proxy settings as to bypass addresses starting with the address of the SIACS web based interface (https://SIACS).

V.2.1.3 Certificate Server

Because we implement SSL Client Side Authentication, we need to create a digital certificate for each Internet user signed by a Certificate Authority defined in web configuration of SIACS. In our framework, all the user certificate signing and other relevant processes are carried out on the certificate server seen in Figure 10.

V.2.2 Operation of SIACS

We are going to examine how SIACS operates into four different subsections:

- Definition of a new user
- Certificate installation on client side
- Internet logon
- Session termination

V.2.2.1 Definition of a New User

When a user is to be defined to the system, we first need to create a new certificate request in the certificate server. In this request, we enter all the informative data about the user which will be used in the certificate. In the definition process, the important point is the CN (Common Name) part of the certificate. In our framework we have created a specific format for CN as "UserID UserName UserSurname". We will later use this information in SSL Client Authentication in order to determine whether the user having the given parameters (UserID, UserName and UserSurname) is registered into SIACS or not.

After creation of certificate request, the second step is to sign this certificate by using the CA's private key. This CA should be the one which is configured in web server configuration file in order to be used in SSL Client Authentication. After this process is finished, a user certificate is ready to use.

In order to perform SSL client authentication, we need to install user certificates in clients' browsers. For this purpose we will use PKCS12 which is a file format commonly used to store private keys with accompanying public key certificates protected with a password-based symmetric key [55]. So, first we need to extract private key from the user certificate and then generate PKCS12 file in order to be able to have both user certificate and private key. In the process of PKCS12 file creation, a

password is determined. This password will be used later when the certificate is to be installed on client's machine. In SIACS, we keep this password secret and let only the system administrators know it. When an installation is performed, system administrator installs the certificate on behalf of the user. The reason why we implement this method is to prevent certificates to be moved from one machine to another.

SIACS also implements username-password authentication as well as SSL client authentication. So after certificate and PKCS12 file creation processes have been completed, the second step is to assign a username-password for the user. We perform it using SIACS password generation utility. These two processes are sufficient for SIACS user definition.

Now, the user has been registered to SIACS. But Internet is still not allowed because user has to authenticate himself also to SIACS web based interface as a user level authentication step. Before seeing how this is done in section V.2.2.3, let's first introduce you the client certificate installation operation in the next section.

V.2.2.2 Certificate Installation on Client Side

Figure 13 to 17 illustrates the certificate installation process on Client Side. Let's see step by step how it is done:

In order to start installation, we first need to move the pkcs12 file to the client's machine.

Figure 13: Certificate Import Wizard

As seen in Figure 13, when we double click the PKCS12 file, a certificate import wizard is opened in order to let us choose the path of the file.

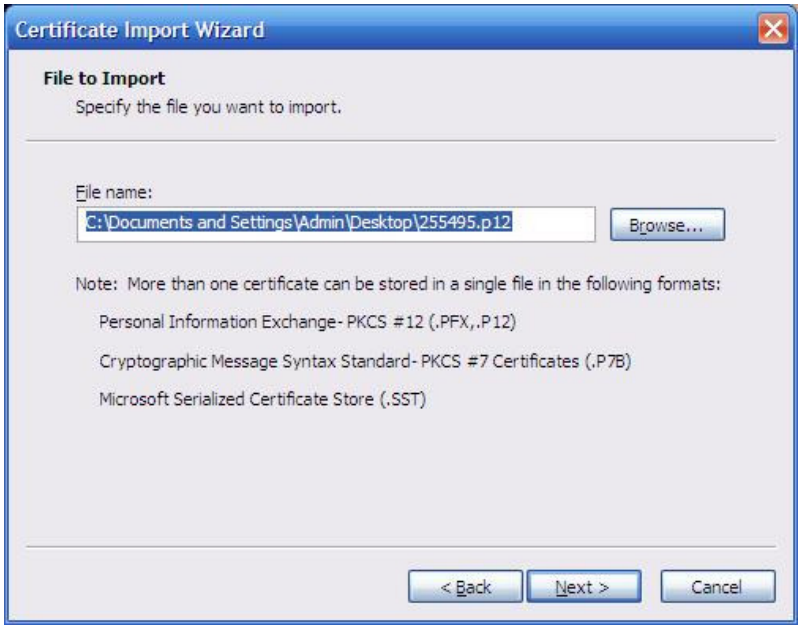

Figure 14: Import of the file incorporating user's certificate

Now a password for the private key is requested in order to continue to the process (Figure 15). This password is the one known only by the system administrators. System administrator enters a password here and clicks next without choosing “Mark this key as exportable” option.

Figure 15: Password determination for private key

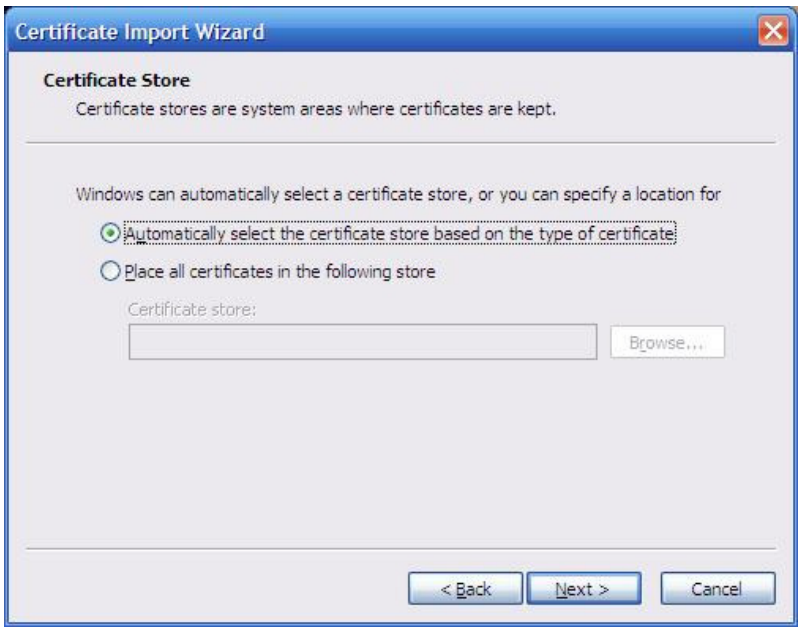

Figure 16: Certificate Store Determination

In Figure 16, we let the system automatically choose the certificate store type. This is actually “Personal” store when automatic option is chosen.

Figure 17: Completion of Certificate Import

Finally, in Figure 17, we finish certificate installation process.

V.2.2.3 Internet Logon

The flowchart of Internet logon process in SIACS is depicted in Figure 19. In order to have an Internet connection, user should first authenticate himself to SIACS. This is done via SIACS web based Internet access interface. Because of that SIACS implements SSL client authentication, when connected to interface, user first confronts with a pop-up window requesting user’s certificate. In this window, all the certificates having been installed to the browser can be seen (see Figure 18). User chooses the certificate belonging to him and continues his session.

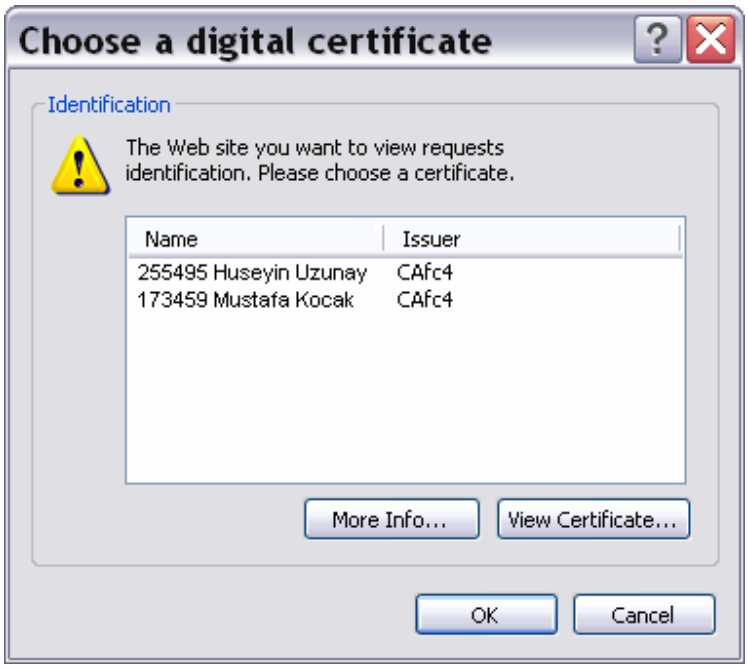

Figure 18: Certificate Selection in SSL Client Authentication

After receiving the client's certificate, SIACS first checks whether the certificate is signed by a CA that has previously been defined in its configuration. In SIACS before Internet permission is given, various checks are performed. If one of them fails, the logon process is cut off and user is not able to have an Internet access (see Figures 29 and 32 in Appendix).

After CA is validated, certificate information is extracted and checked whether the user having this certificate is registered into SIACS database or not (see Figure 28 Appendix). If everything is okay up to this point, SSL client authentication is deemed successful.

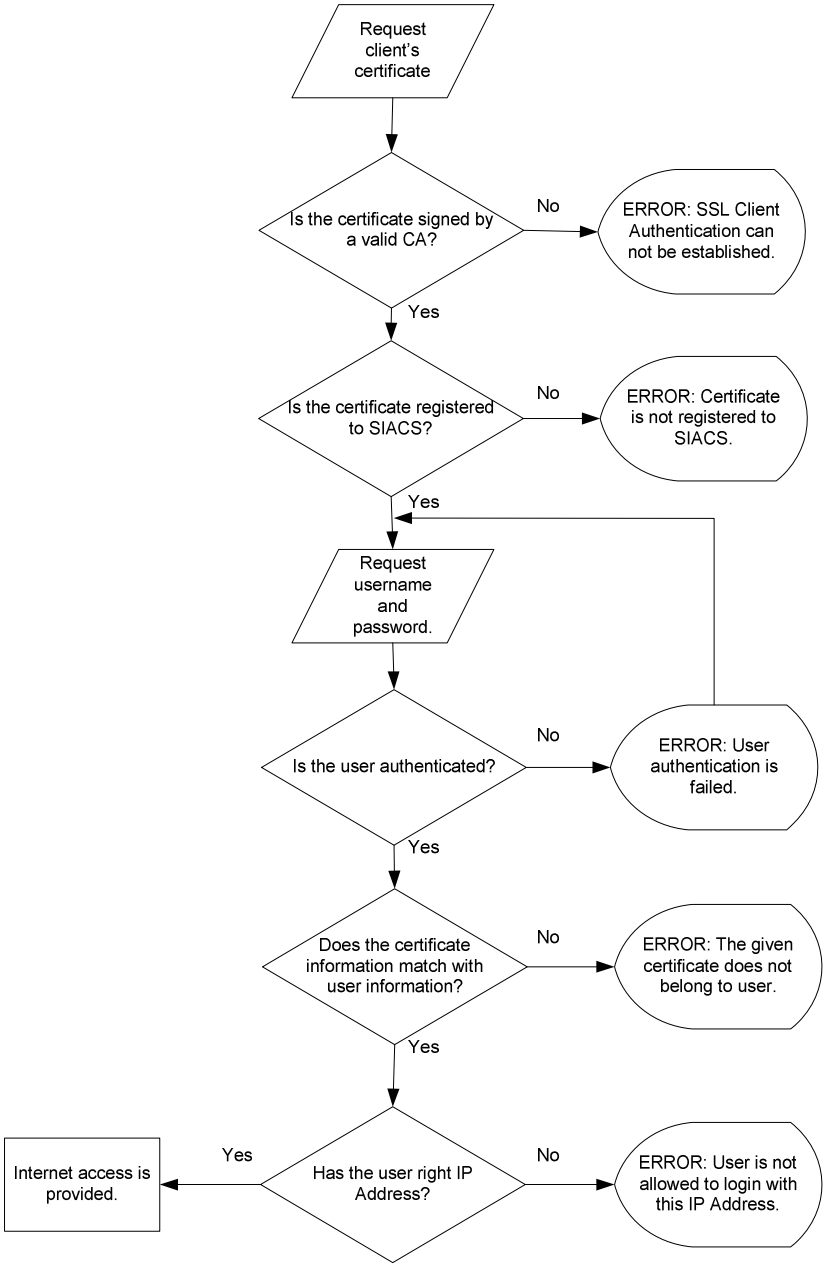

Figure 19: Flowchart of SIACS Internet Logon

Now the second authentication scheme, which is username-password, takes place (see Figure 30 in Appendix). User enters his credentials via a login interface and tries to authenticate himself. SIACS first performs authentication and then checks whether the user's information matches with the user information in the previously given certificate (see Figure 31 in Appendix). If no problem, SIACS finally checks the IP address of the connection (see Figure 33 in Appendix). If it is the one assigned for that user in the database, Internet access is then given for that IP address (see Figures 34 and 35 in Appendix). This process includes adding a new line in the firewall rule file that permits the packets coming from the user's IP address and having a destination port as the proxy port (8080 in our framework).

V.2.2.4 Session Termination in SIACS

After Internet logon is performed successfully, a new record including user id, user's login date and time, logout date and time, last refresh timestamp and user status is added to a log table in the database. At the initial phase, the logout date and time field is null and last refresh timestamp indicates the login time. On the client side SIACS web based interface is still open and refreshes itself in 30 seconds continuously. Note that in order to keep the system running, the interface should not be closed. In every 30 second periods, user's session id, session username and session user id are checked and the timeout value in the associated log record is updated.

In SIACS, there are 2 kind of session termination:

- Graceful Termination.
- Nongraceful Termination.

Graceful Termination: In graceful termination, user wants to terminate his Internet session and clicks logout button in the SIACS web interface (see Figure 35 in Appendix). When logout button is clicked, SIACS first gathers the user's information and then connects to the firewall, deletes the line with the user's IP address. Afterwards it enters the relevant date and time data into the logout date and time field

in the database and make the status of the user "0" which indicates that the session of the user is terminated (see Figure 36 and 37 in Appendix).

Nongraceful Termination: Nongraceful termination is performed when one of two types of event has occurred. The first one is when SIACS web interface is closed because of an accidental behavior of the user (i.e. close browser) or a shutdown in computer (i.e. an electrical cut). The second one is occurred when an unauthorized access is detected.

So let's explain how termination process is done in SIACS, when such cases happened. For the first case, a control daemon runs continually at the background on the server and checks the refresh timeout fields of the active records (the records the status field of which is "1") in the database. In this process a comparison between the current timestamp and the timestamp in the database is done. If the difference is more than 40 seconds (this value can change according to the volume of the networks), a timeout process is initiated.

The operations in the timeout process are as same as the operations in graceful termination. First the firewall configuration is made in order to delete the rule with the IP address of the user whose session is expired and then a record indicating the logout date and time is entered into the database.

In the second type of event, either a feedback is received from UNIDES indicating that an unusual record is detected with the relevant IP address or a change in the IP Address is detected while refreshing. For both type of the events, the session of the user is terminated in SIACS.

V.3 Development Environment

In this section, we would like to briefly mention about the development environment of our proposed system. In both UNIDES (version 1) and SIACS, we have used php4

for the web based system operations. As a web server, we have used Apache 1.3.29 with MODSSL support.

UNIDES and SIACS use the same database. Because we need fast read-write database operations in our framework, we prefer using MYSQL.

All our software are based on the systems having FreeBSD operating system. In order to perform some system operations on FreeBSD servers, we have used shell script language.

In UNIDES, we need also to connect to the switches remotely and perform some switch based operations. For this purpose, we utilize expect scripting language.

V.4 Robustness of Our Proposed Solution

In our proposed system, Unauthorized Internet Access Blocking System validating the Source Information in Internet Access Logs, we have both implemented SIACS and UNIDES. Thus, we have a system which is supported by both user level authentication and low level authentication. Our proposed architecture provides security of the authentication mechanisms in all layers (see Figure 3).

In this framework only the authorized users having a digital certificate and a username-password can have an Internet access. Furthermore user can only access to the Internet over the machine determined by the system administration because the user certificate can not be moved from one machine to another without the certificate admin password. Consider that an adversary finds a way to move a certificate to another machine. While he is trying to logon to SIACS, the IP address of the user will not match with the IP address assigned for that user in the database, and logon process will not be continued. The user can not change MAC or IP address of the machine, because UNIDES only allows a MAC address from a specific port of a switch and checks the MAC and IP address pairs. Whenever an invalid match is

detected, it blocks the Internet connection of that MAC address and gives alarm. So, this provides us to ensure over which machine, the user has logged into Internet, namely source machine authentication.

User authentication provided by SIACS includes both username-password and SSL client authentication. So vulnerabilities of username-password scheme are prevented. Because of the fact that the security of SSL protocol is well established, we can say that user level authentication scheme in our framework is rather robust.

Other than source authentication, UNIDES has also some benefits in terms of LAN attacks. Although this is not our primary goal in this thesis, we would like to mention some of them in order to give you an idea about the extra advantages of UNIDES. Assume that we only use SIACS and do not implement any low level security mechanism (First version of UNIDES) at the bottom line. In this case, an adversary first can authenticate himself to SIACS from his machine and have an Internet access. He also uses another program that can generate layer 2 attacks such as MAC spoof, IP spoof and etc. By using this program he can perform MAC flooding attack to the switch and after the MAC table of the switch is overflowed, he can sniff all the ports of the switch.

In another scenario, because of that there is not any control for MAC and IP address matches, the adversary can take a MAC address of another machine connected in the same switch. This is called MAC Cloning attack. Now all the packets having destination for that MAC address will also come to the machine of adversary. Adversary can now perform sniffing attack. He can also perform Denial of Service attack by sending a lot of forged packets to the network. SIACS can not detect all these attacks lonely because the adversary does not change his IP address in the connection between SIACS web interface and him.

Hence we can come up with a conclusion that it is required to implement a low level security mechanism (First version of UNIDES) at the bottom line of our framework.

CHAPTER VI

CONCLUSION and FUTURE WORK

"How far that little candle throws his beams,
So shines a good deed in a naughty world."
Shakespeare

In this dissertation, we have dealt with the source authentication problem in Internet Access Logs. In order to overcome the problem, we have designed and implemented an Unauthorized Internet Access Blocking System Architecture validating the Source Information in Internet Access Logs.

Our proposal is composed of two different but related subsystems. One of them is UNIDES: Unauthorized Internet Access Detection and Blocking System and the other is SIACS: SSL based Internet Access Control providing user level source authentication in Internet Access Logs. Both systems have the capability of running independently in order to provide an Internet access control in local area networks but they are not adequate alone when the source authentication of Internet access logs is of concern. So in our design, we use both of them and see that when implemented correctly source authentication problem can be solved in Internet Audit Logs. Our first proposal UNIDES, lying on the bottom of the architecture, is a proxy based system that solves the source authentication problems in lower levels. When a user is

to be given Internet, first the MAC address and the other relevant information is registered by UNIDES. The registration process involves defining the MAC address of the user to all related ports of the switches and making the necessary configuration in proxy and DHCP servers. UNIDES is initially informed with the existing topology, so all the required configurations are done by UNIDES itself not by the administrator. From this point of view, UNIDES can also be seen as an efficient intranet management system.

In another view, UNIDES can be regarded as a hardware based model incorporating advanced switches with port security mechanism into the proposed solution. By the help of these switches, other than supporting user level authentication at lower layers and providing robust source authentication, we have also prevented most of the data link layer attacks such as MAC spoofing, MAC cloning, MAC flooding and etc. For source authentication, we have associated MAC addresses of the users to the constant IP addresses which are then entered into a central database with the additional user information. We have prevented these IP addresses to be changed by the detection and blocking mechanisms of UNIDES. Whenever an incorrect IP-MAC address match is detected, UNIDES blocks the Internet connection of that MAC address by deleting it from the switches and servers. UNIDES also has a web based alarm mechanism. If an unauthorized Internet access is detected and blocked, an alarm plays at the administration interface and the detected record is viewed on the screen.

UNIDES uses a very effective mechanism to solve low level authentication problems. When it is implemented independently, it is assumed that each machine is registered to one user and nobody else can use another one's machine. But sometimes in the organizations one machine is used more than one user. Another point is that at the upper layer to implement a user level authentication scheme will facilitate to accurately detect the source user in Internet access logs. So these requirements led us also design SIACS: SSL based Internet Access Control System which provides user level source authentication in Internet Access Logs.

SIACS works at top of our architecture and implements a very robust user level authentication scheme including SSL Client authentication and username-password authentication. In SIACS every user has a digital certificate and Internet login password. Whenever a user wants to connect to Internet, he has to first connect to SIACS web based Internet access control interface. He then presents his certificate and password to the system. SIACS implements a lot of controls over the given credentials. If the authentication succeeds, Internet access is given to the user.

SIACS also implements a timeout mechanism. If the computer of the user is shut down without a logout or the web based interface is closed, system waits for a timeout value and then automatically revokes that user and blocks his Internet connection. SSL User Certificates are stored in PKCS12 formatted files. This file is also password protected and passwords are known only by the system administrators. When a certificate is to be installed to the client's machine, system administrator installs it by entering the admin password of the file. Thus, in this scheme, certificates can not be moved from one computer to another because of the password protection. This mechanism provides us to ensure source machine of user in Internet access logs.

Unauthorized Internet access detection is performed mostly by UNIDES. Whenever it detects an unauthorized record, it informs also SIACS while it performs blocking in the low level itself. Having been informed by UNIDES, SIACS also ends the user session and implements blocking on firewall.

For the real time requirements of source authentication in Internet audit logs, we do not follow up a methodology to make use of a real time authentication because of the efficiency concerns, but instead, we have constructed a system model that detects any unauthorized action almost in real time and blocks it. In other words, If we are sure that the user and the machine identity is not changed after the first authentication, we do not need to perform a second authentication.

As a conclusion, we can say that in our architecture, we have unified low level and upper level defense mechanisms and tried to cover all the source authentication problems in the layers of Internet stack. One of the prominent advantages of our architecture is that we have implemented our solution on top of the available systems and protocols and used a very well known topology. So this system can easily be put into practice in most of the local area networks without any need to change in the available protocols.

By providing healthy source authentication we have had a lot of gaining. The first one is that we have provided the nonrepudiation of source in Internet access logs because we have the ability to prove which user is the owner of a specific audit record. Hence when an unlawful Internet activity is detected from our network, we can make precise detections without any doubt. This also led us to find another important gaining of our system which is to make the Internet audit logs used as digital evidence in a court of law. Evidences, by definition, must meet certain standards to be admitted legally. If source authentication can not be established well enough, the admissibility of these log records as evidence will be very difficult.

Another benefit of our system is that we can also detect on which machine the Internet action specified in the log record took place. In more precise terms, we have provided source machine authentication in Internet access logs. This is done by the help of the port security mechanism of UNIDES which disallows a machine to connect to Internet from another place and immoveable certificates of SIACS. Precise detection of source machine is very important especially in a cyber crime case where original crime machine is needed to be obtained in order to make forensics analysis and reach some digital evidences.

Other than source authentication, our system also facilitates secure Internet management in a local area network because only the authorized users which are registered to the system by the system administration can have an Internet connection.

All definitions and configurations are carried out automatically by the system itself by easing the system administrators' tasks.

One promising future work should be to increase the efficiency of user level authentication by implementing different SSL schemes such as Reverse SSL [56]. This will be really a good contribution because it is quite likely that there will be intensive traffic and overload when all users try to authenticate himself to SIACS especially in the beginning of the working hours.

Our proposed solutions in this thesis are mostly constructed over a specific topology. Another future work will be to tailor the system to work on top of any topology. For example, there should be a topology drawing part in the system. Initially, every organization can create its own topology by locating the advanced switches wherever they want. Thus, system should start functioning according to this newly created topology.

Another good future work should be studying on a mechanism that can be used instead of advanced switches. In our framework, every switch is advanced switch having port security feature. The cost of this type switches is rather high when compared to standard switches. Today because of that the most effective methodology to defense against data link layer attacks is MAC binding on these advanced switches, we had to use them. But in the future, a security protocol for ARP can be proposed to avoid using a MAC binding methodology.

In the current form of our proposed system, we do not deal with source authentication of devices having wireless network connection. Actually, SIACS, a user level authentication system which we have added to extend first version of UNIDES, can also be used in wireless networks because in SIACS, clients do not need any physical interaction. Each wireless device should have a digital certificate approved by system administration in order to have an Internet connection. But as a future work we plan to focus our studies more on wireless networks and the security issues in wireless

authentication schemes and then implement a robust system that can also be used in wireless networks.

In order to prevent user's connecting to Internet from another machine, SIACS implements password protected PKCS12 files which include user's certificate and private key. The passwords of these files are only known by the system administrators. So whenever a certificate is to be installed, a system administrator must perform this process. Maybe a mechanism should be devised to automatically install user certificates remotely without any need to system administrators but without revealing the password.

One other interesting future work is on the issue of user privacy. In its current form, our solution does not support user privacy. In a future version of this system, we are planning to include a new functionality so that the identity of the owner of a record can be established only in case of a legal dispute. Otherwise, due to privacy reasons, no single entity including the system administrators by themselves would establish the identity of the owner of Internet records.

As a final future work, a secure Internet audit log scheme [57,58,59] should be implemented to Internet access logs as well as source authentication. Legally in order to provide Internet access logs as digital evidence, two important issues are to prove to whom these log records belong (in our scheme this is called as source authentication) and to provide the integrity of the logs that is to prove that the logs have not been altered since they were created. In our thesis, we have established source authentication part. If our study is extended with a system that provides integrity and the security of the logs, the legal admissibility of the Internet access logs as digital evidence will have been maximized when an illegal activity is of concern.

REFERENCES

[1] Gollmann, D. (1999). *Computer Security*. Chichester: John Wiley & Sons.

[2] Computer Security Institute (2003). *2003 CSI/FBI Computer Crime and Security Survey.* Retrieved August 8, 2006, from http://www.reddshell.com/docs/csi_fbi_2003.pdf

[3] Kurose, J.F. & Ross, K. W. (2003). *Computer Networking, Second Edition*. USA: Pearson Education.

[4] Uzunay,Y. & Bicakci, K. (2005). UNIDES: An Efficient Real-Time System to Detect and Block Unauthorized Internet Access", *1st International Workshop on Security in Networks and Distributed Systems (SNDS) in conjunction with ICPADS 2005.* IEEE, Computer Society, Fukuoka, Japan

[5] Schneier, B. (2000). *Secrets and Lies, Digital Security in a Networked World*. New York: Wiley Computer Publishing.

[6] Menezes, A., Van Oorshot, P. & Vanstone, S. (1996). *Handbook of applied cryptography, CRC Press series on discrete mathematics and its applications*. Boca Raton: CRC Press.

[7] Anderson, R. (2001). Security *Engineering: A Guide to Building Dependable Distributed Systems*. New York : John Wiley and Sons.

[8] Kaufman, C., Perlman, R. & Speciner, M. (2002). *Network Security, Private Communication in a Public World, Second Edition*. Englewood Cliffs, NJ: Prentice Hall Series.

[9] Stinson, D. R. (2002). *Cryptography: Theory and Practice, Second Edition*, Boca Raton: CRC Press, Inc.

[10] Schneier, B. (1996). *Applied Cryptography*. New York: John Wiley & Sons.

[11] Burnett, S. & Paine, S. (2001). *RSA Security's Official Guide to Cryptography*. Berkeley: McGraw-Hill.

[12] Atreya, M. *Introduction to Cryptography*, Retrieved August 1, 2006, from http://www.rsasecurity.com/products/bsafe/overview/IntroToCrypto.pdf.

[13] Stallings, W. (2002). *Network Security Essentials Applications and Standards*. New Jersey: Prentice Hall.

[14] Bicakci, K. (2003, Sept.). *On The Efficiency of Authentication Protocols, Digital Signatures and Their Applications in E-Health: A Top-Down Approach*. Doctoral dissertation, Informatics Institute, Middle East Technical University, Ankara.

[15] Diffie, W. & Hellman, M.E. (1976). "New Directions in Cryptography", *IEEE Trans. Information Theory, 22(*6), 644-654.

[16] Shamir, R.L. & Adleman, L. (1978) "A method for obtaining digital signatures and public-key cryptosystems", *ACM, 21,* 120-126.

[17] Network Time Protocol, ftp://ftp.rfc-editor.org/in-notes/rfc958.txt

[18] Laroche, M. (1996, Jan.). *Network Security Analysis and Implementation*, Retrieved March, 12,2006 from http://www.cse-cst.gc.ca/documents/publications/gov-pubs/itsg/mg1.pdf

[19] Bellovin, S.M. & Merritt, M. (1993, Nov.). Augmented encrypted key exchange: a password-based protocol secure against dictionary attacks and password file compromise, *Proceedings of the 1st ACM conference on Computer and communications security,* Fairfax, Virginia, United States.

[20] Morris, R., & Thompson, K. (1979, Nov.). Password security: A case history, *Communications of ACM, 22(11),* 594–597.

[21] Singh,K. (1985, Jan.). On improvements to password security, *ACM SIGOPS Operating Systems Review, 19(1)*, 53-60

[22] Spafford,E.H. (1992). Opus: Preventing weak password choices*, Computers & Security, 11(3)*, 273-278.

[23] Yeh,H. & Sun, H. (2002, Oct.). Simple authenticated key agreement protocol resistant to password guessing attacks, *ACM SIGOPS Operating Systems Review, 36 (4)*, 14-22

[24] Howard, J.D. (1997). *An analysis of security incidents on the Internet, 1989-1995*, Dissertation, Engineering and Public Policy, Carnegie Mellon University.

[25] *Crack*. Retrieved November 2, 2005, from ftp://coast.cs.purdue.edu/pub/tools/unix/pwdutils/crack/

[26] *John the Ripper password cracker.* Retrieved November 2, 2005, from http://www.openwall.com/john/

[27] McDaniel, P. (2001, May). *Authentication*, At&T Labs.

[28] Russell, D. & Gangemi Sr, G.T. (1991). *Computer Security Basic.* USA: O'Reilly.

[29] *Authentication, Authorization, and Access Control.* Retrieved November 31, 2006, from http://httpd.apache.org/docs/1.3/howto/auth.html

[30] Fu, K., Sit, E., Smith, K., & Feamster, N. (2001). Dos and Don'ts of Client Authentication on the Web, *10th USENIX Security Symposium,* Washington, DC, USA.

[31] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., & Stewart, L. (1999). *HTTP Authentication: Basic and Digest Access Authentication*, Internet Engineering Task Force, RFC 2617.

[32] McDaniel, P. (2001, May). Authentication, At&T Labs.

[33] Thomas, S. (2000). *SSL & TLS Essentials: Securing the Web*, , Alpharetta: John Wiley & Sons

[34] *Overview of CSS SSL.* Retrieved August 8, 2006, from http://www.cisco.com/en/US/products/hw/contnetw/ps792/products_configuration_guide_chapter09186a0080579e21.html

[35] Northcutt, S. & Novak, J. (2000, Sept.). *Network Intrusion Detection, Second Edition*, USA: Sams.

[36] Heberlein, L.T. & Bishop, M. (1996, Oct.). Address Spoofing, *Proc. of the 19th National Information Systems Security Conference* (pp. 371-377)

[37] Bellovin, S. (1994, Aug). *Security Concerns for IPng*, RFC1675.

[38] Bellovin, S., Schiller, J. & Kaufman, C. (2003, Dec.). *Security Mechanisms for the Internet*, RFC 3631.

[39] *UNIX User's Reference Manual* (1986). Computer Systems Research Group, Berkeley Software Distribution Virtual Vax-11, Computer Science Division, UC Berkeley.

[40] *IP-spoofing Demystified: Trust-Relationship Exploitation*, (2002, March). Guild Productions, Phrack Magazine Retrieved March 21, 2002, from http://www.fc.net/phrack/files/p48/p48-14.html.

[41] Plummer, D. C. (1982, Nov.). *An Ethernet Address Resolution Protocol or Converting Network Protocol Address to 48bit Ethernet Address for Transmission on Ethernet Hardware*, RFC826.

[42] Whalen, S., (2001, April). *An introduction to ARP spoofing*, Retrieved August 15, 2000, from http://www.node99.org/projects/arpspoof/

[43] *Etherchange*, Retrieved August 11, 2006, from http://ntsecurity.nu/toolbox/etherchange/

[44] Saarinen, M. (2004). *Legacy User Authentication with IPSEC*, Master Thesis, Helsinki University of Technology, Finland

[45] *A Comparison of VPN Solutions- SSL VS IPSEC*, (2003), Netilla Networks, Retrieved May 13, 2005 from http://www.netilla.com/downloads/wp_ipsec-vs-ssl.pdf

[46] Gouda, M. G., & Huang, C. (2003, Jan.). A Secure Address Resolution Protocol, *Computer Networks, vol.41, no.1.* (pp. 57-71)

[47] de Vivo, M. de Vivo, G. O., & Isern, G. (1998, April). Internet Security Attacks at the Basic Levels, *Operating Systems Review, Vol.32, No.2, (pp.4-15) SIGOPS,ACM*

[48] *ARPWATCH*, Network Research Group, Lawrence Berkeley National Laboratory, Retrieved June 12, 2006 from ftp://ftp.ee.lbl.gov/arpwatch.tar.gz

[49] Beck, R. (1999). *Dealing With Public Ethernet Jacks-Switches, Gateways and Authentication*, Proceedings of LISA'99: 13th Systems Administration Conf., (pp. 149-154) Seattle, Washington, USA.

[50] *Knowlan*, Retrieved May 31,2005, from http://www.enderunix.org/knowlan

[51] *Project R3x*, Retrieved June 5,2005, from http://soul4blade.home.ro

[52] *Expect Scripting Language*, Retrieved August 28, 2006, from http://expect.nist.gov

[53] *Squid*, Retrieved June 6,2005, from http://www.squid-cache.org

[54] Droms, R., (1997, March). *Dynamic Host Configuration Protocol*, RFC 2131.

[55] *PKCS12 Definition*, Retrieved August 28, 2006 from http://en.wikipedia.org/wiki/PKCS

[56] Bicakci, K., Crispo, B. & Tanenbaum, A. S. (2006). *Reverse SSL: Improved Server Performance and DoS Resistance for SSL Handshakes*, Cryptology ePrint Archive, Report 2006/212,

[57] Schneier, B. and Kelsey, J. (1998). Cryptographic support for secure logs on untrusted machines, In *Proceedings of the 7th USENIX Security Symposium,* (pp. 53–62)

[58] Schneier, B. and Kelsey, J. (1999). Minimizing bandwidth for remote access to cryptographically protected audit logs. In *Web Proceedings of the 2nd International Workshop on Recent Advances in Intrusion Detection.*

[59] Schneier, B. and Kelsey, J. (1999). Secure audit logs to support computer forensics. *ACM Transactions on Information and System Security (TISSEC), 2(2):159–176.*

APPENDICES

APPENDIX A. SCREEN SNAPSHOTS of UNIDES

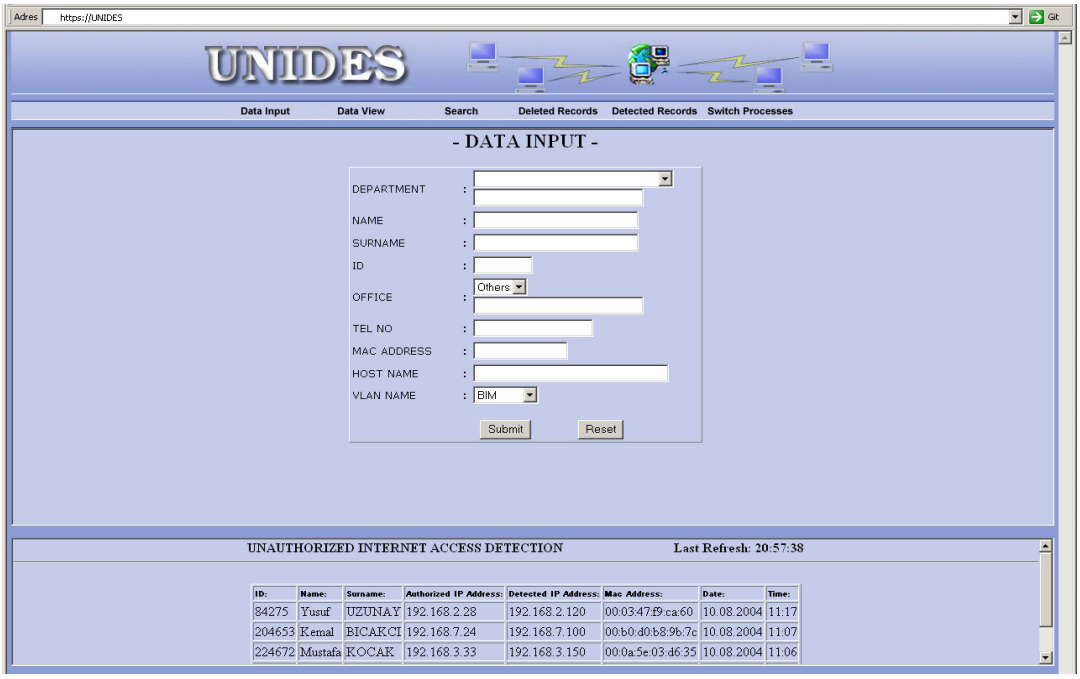

Figure 20: Data Input(Upper) and Unauthorized Internet Access Alarm[2](Lower) frames

[2] UNIDES Alarm frame is seen in the lower frames of all figures

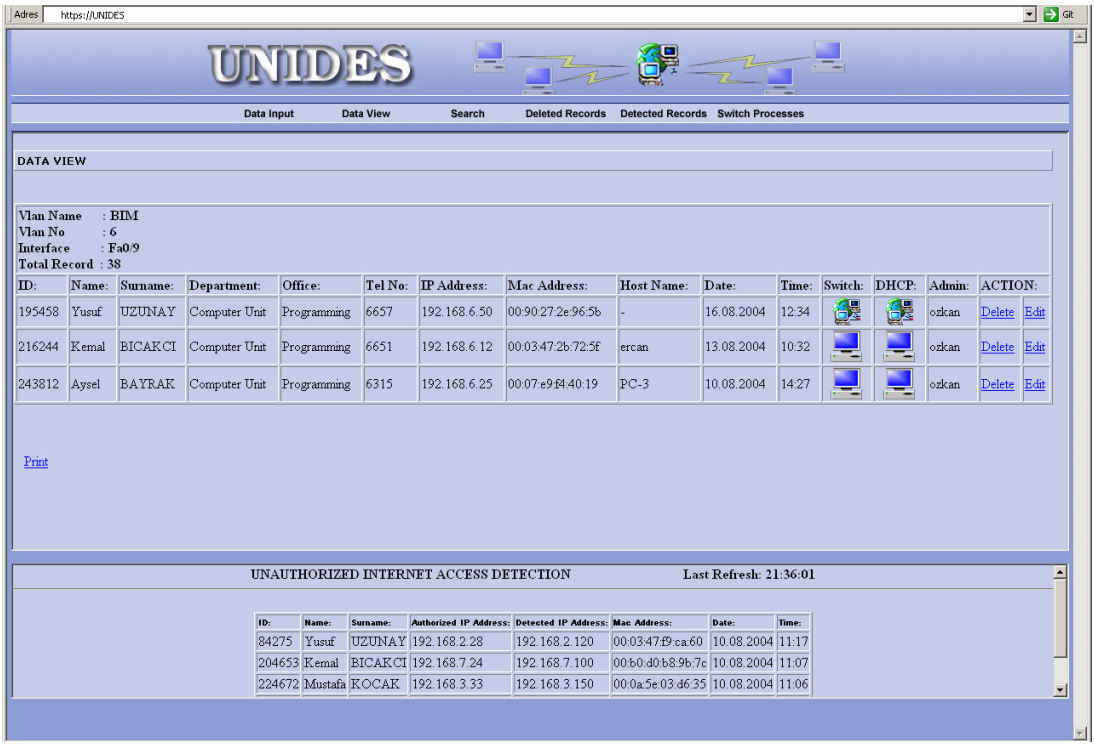

Figure 21: The View of a Record (Yusuf UZUNAY) while it is being defined.

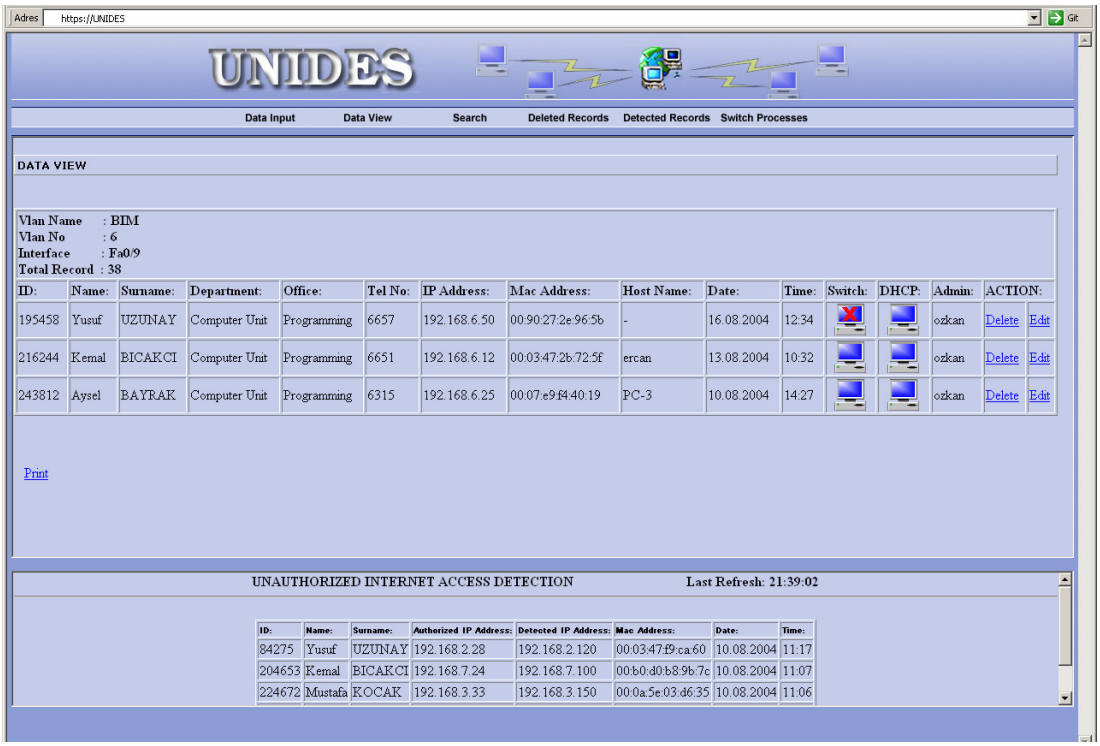

Figure 22: View of Configuration Error in a Switch

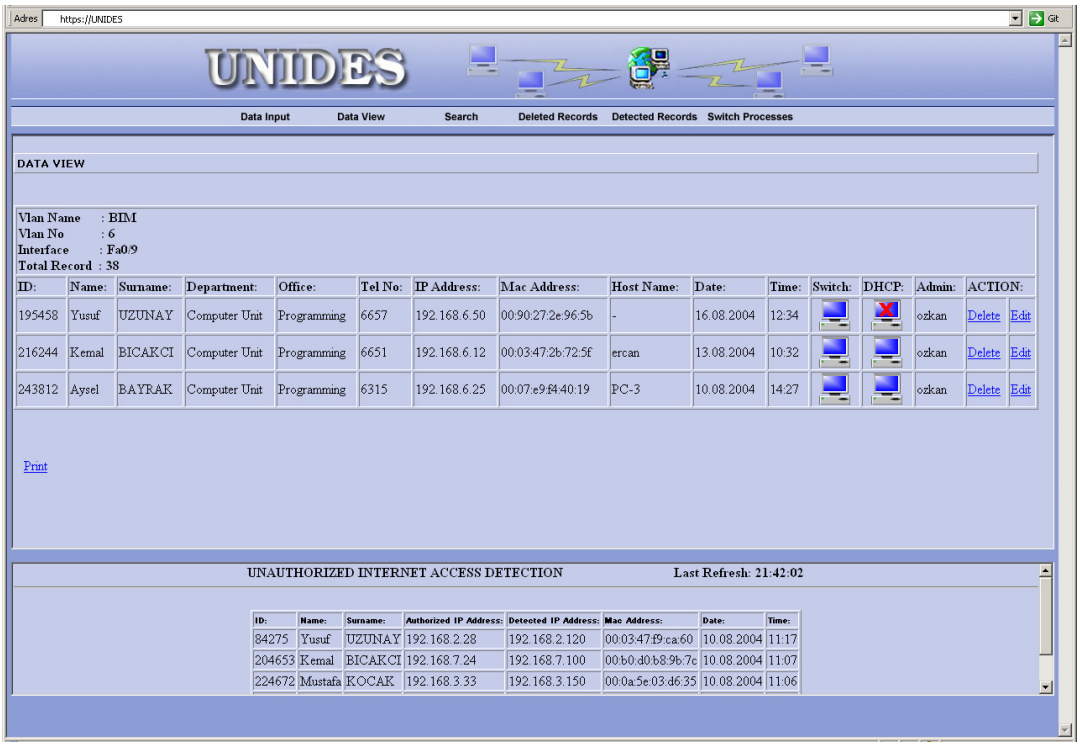

Figure 23: View of a Configuration Error occurred on DHCP Server

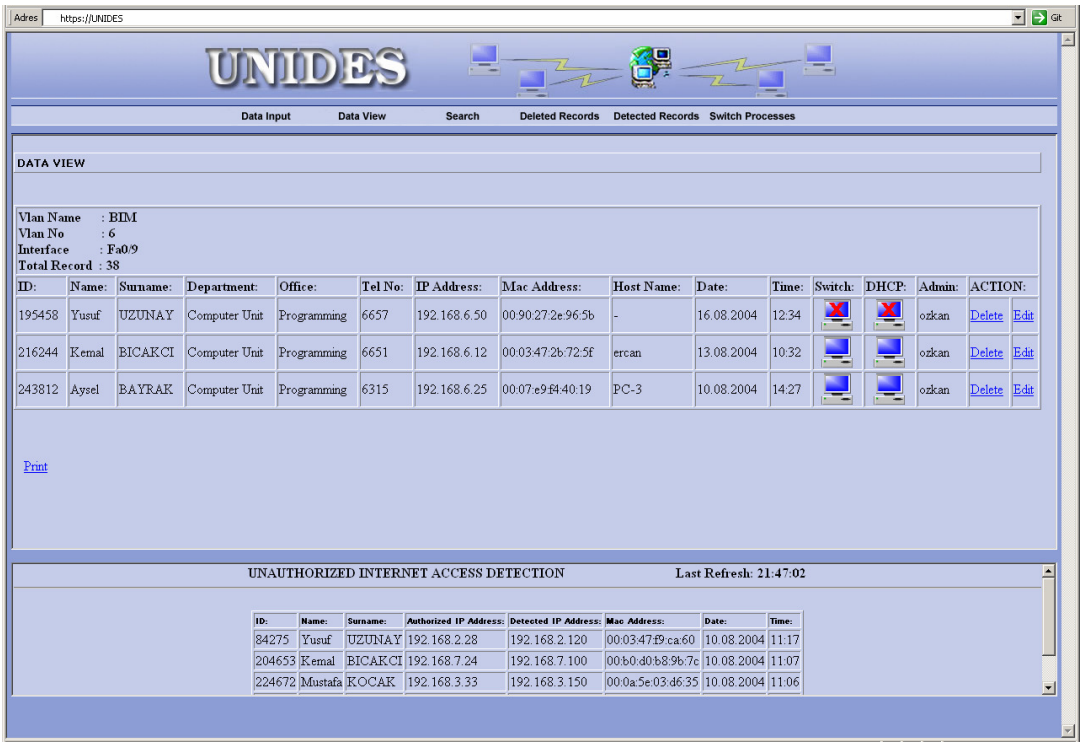

Figure 24: The view of Configuration Errors both in the Switch and in the DHCP Server

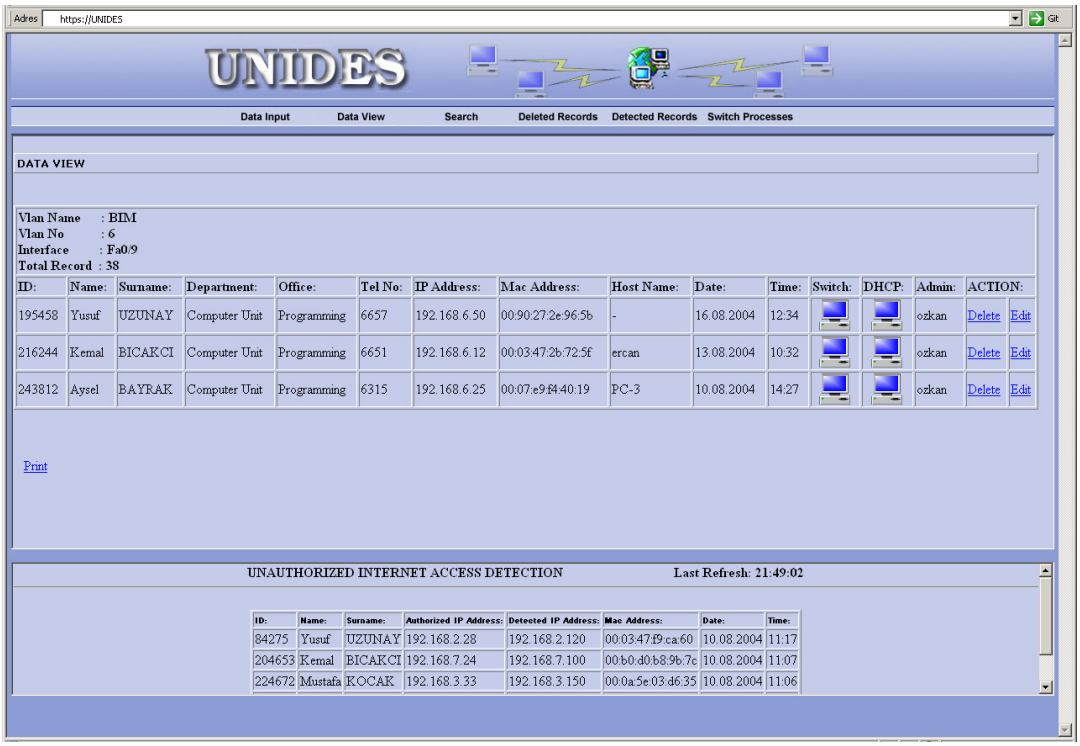

Figure 25: The View of the Records having no problem

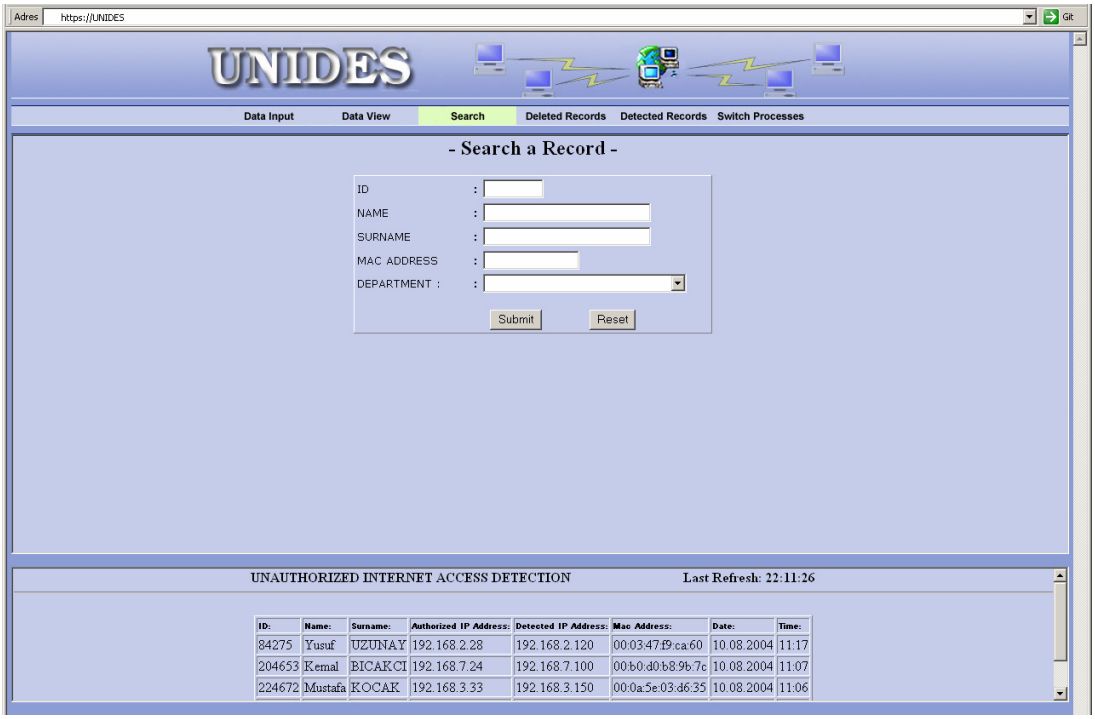

Figure 26: Record Search

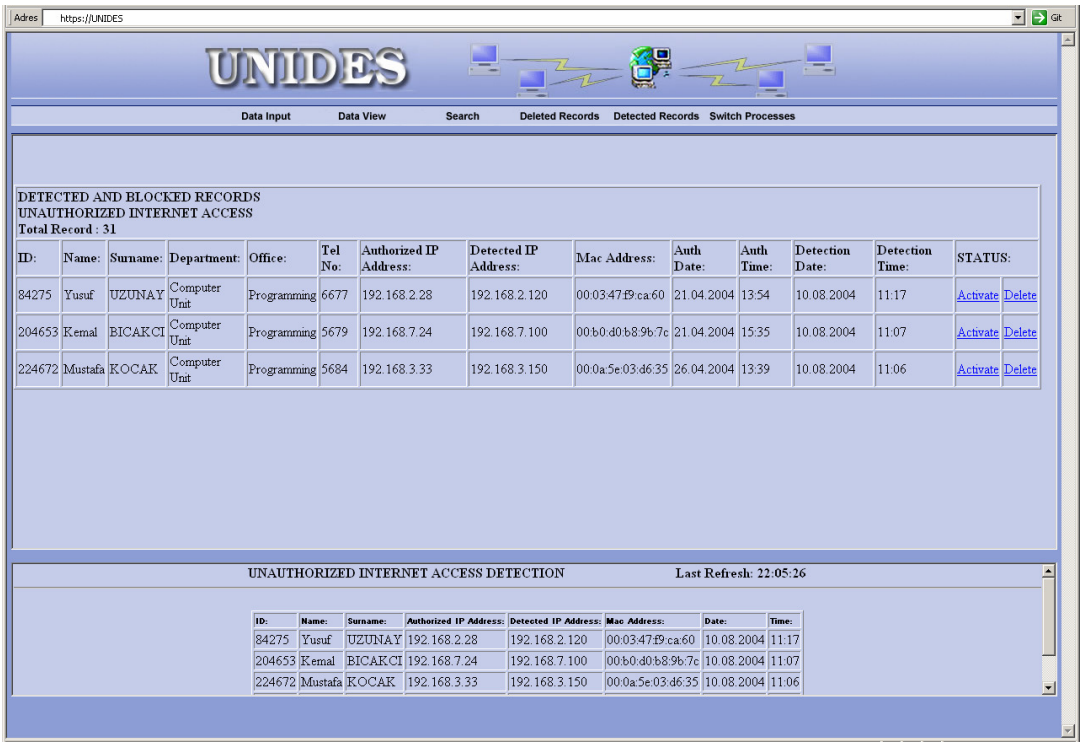

Figure 27: The View of Detected and Blocked Records by UNIDES

APPENDIX B. SCREEN SNAPSHOTS of SIACS

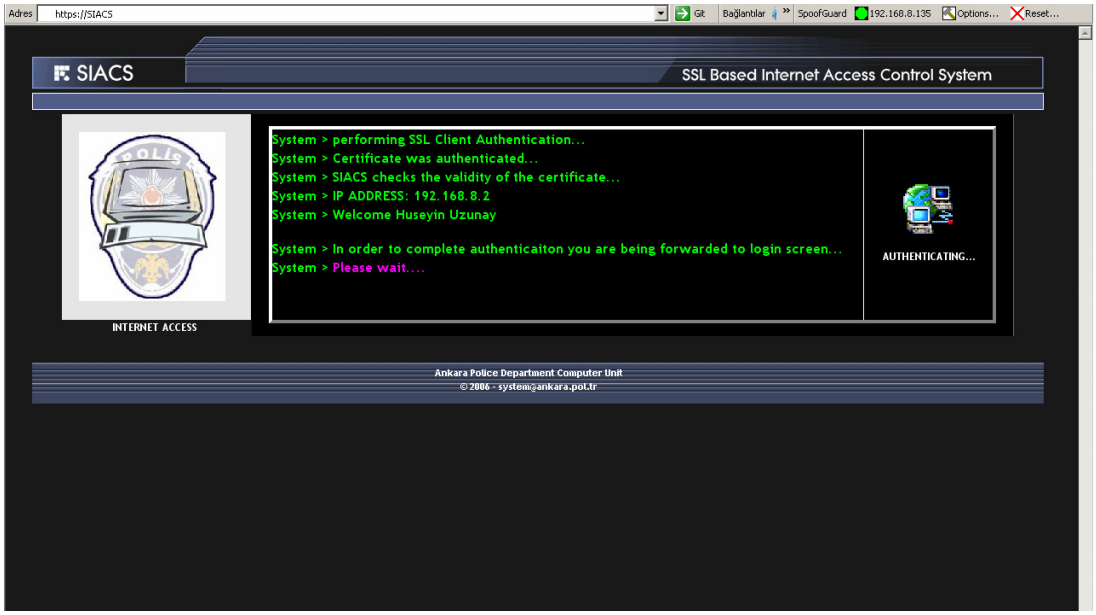

Figure 28 : SSL Client Authentication and Certificate Registration Control

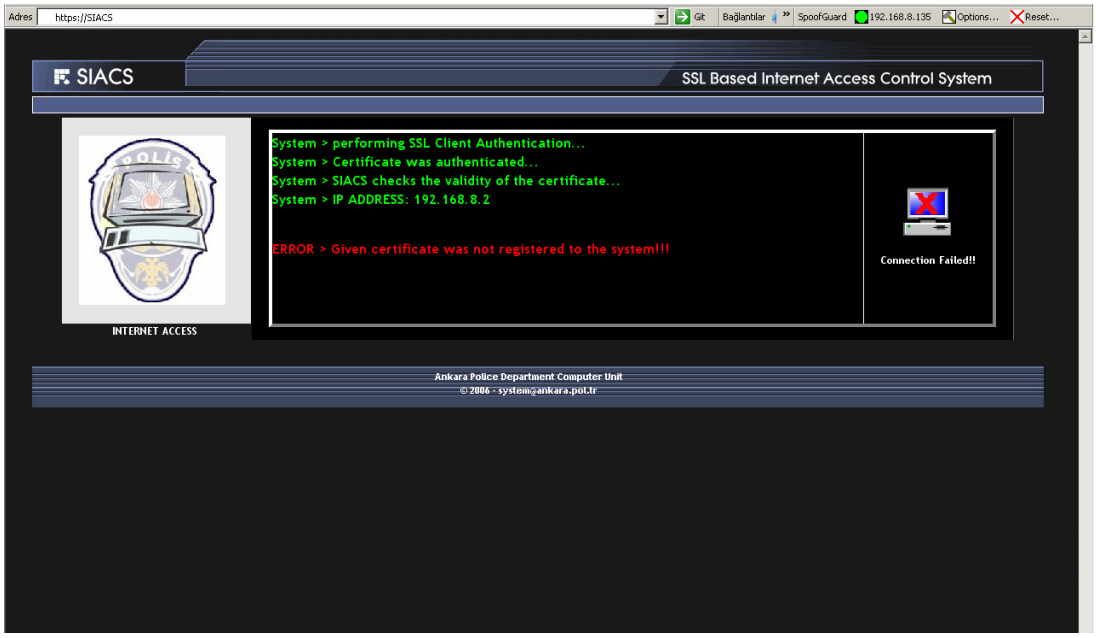

Figure 29: Authentication failure when the given certificate is not registered to SIACS

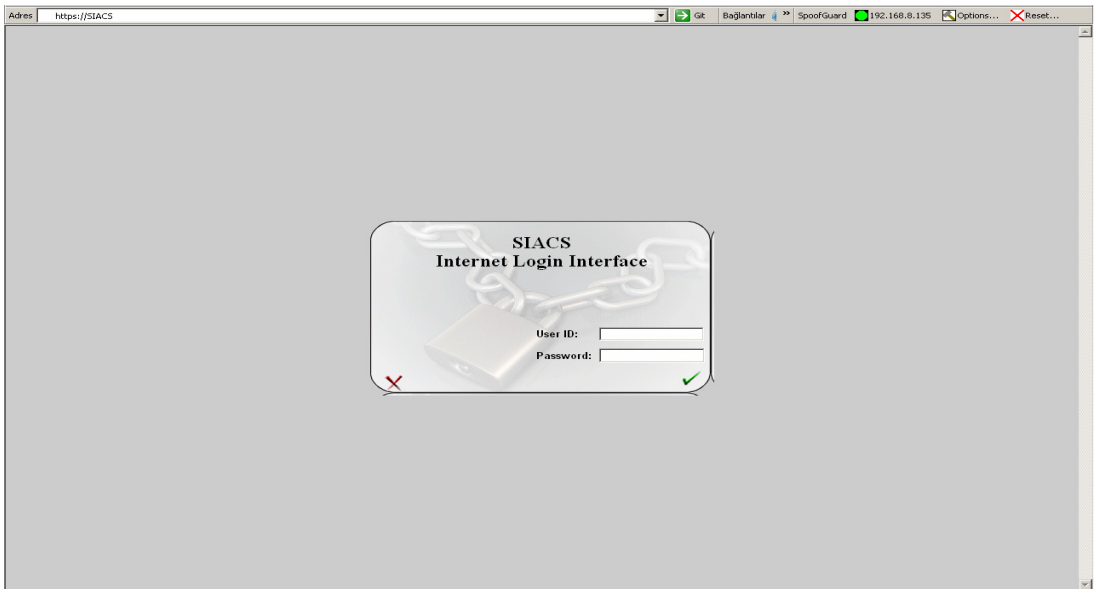

Figure 30 : SIACS Username-password Authentication Interface

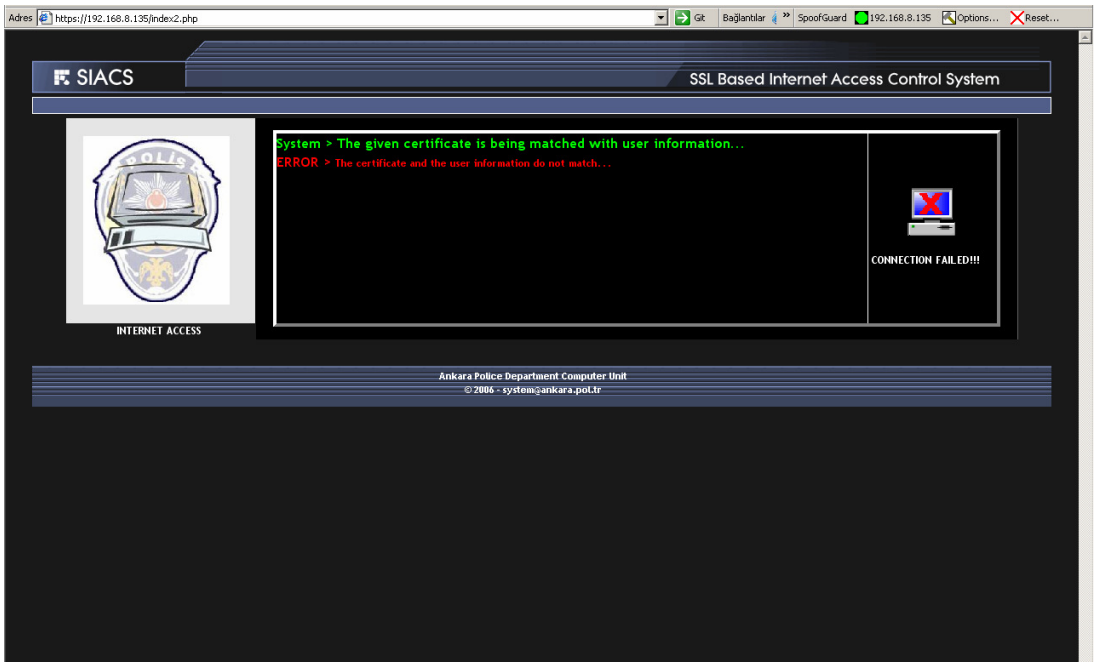

Figure 31: Authentication failure when there is incompatibility between certificate and user information

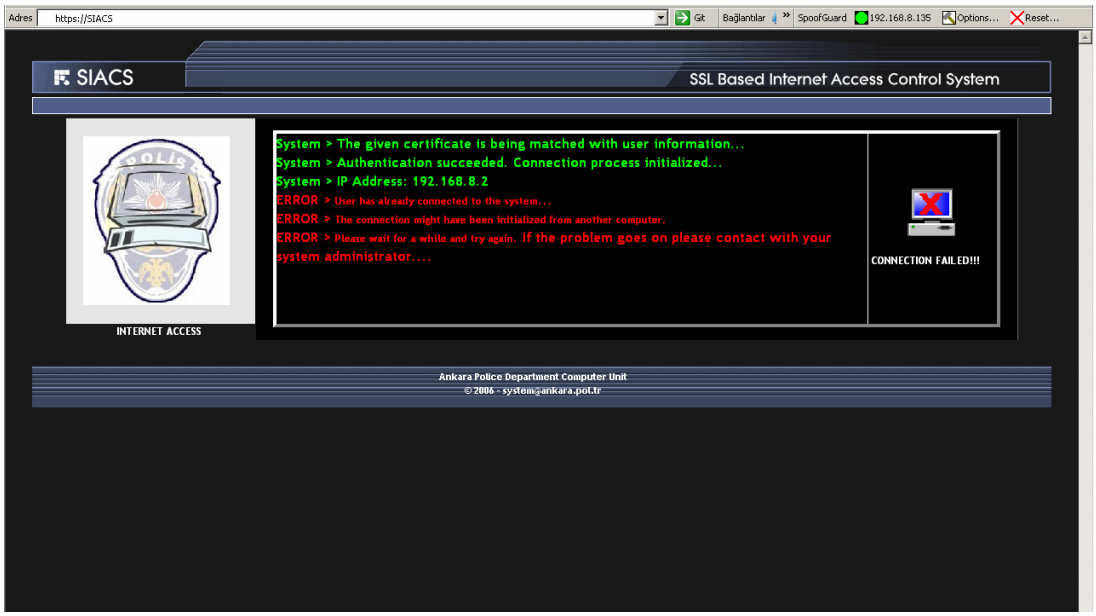

Figure 32: Authentication failure in user status control

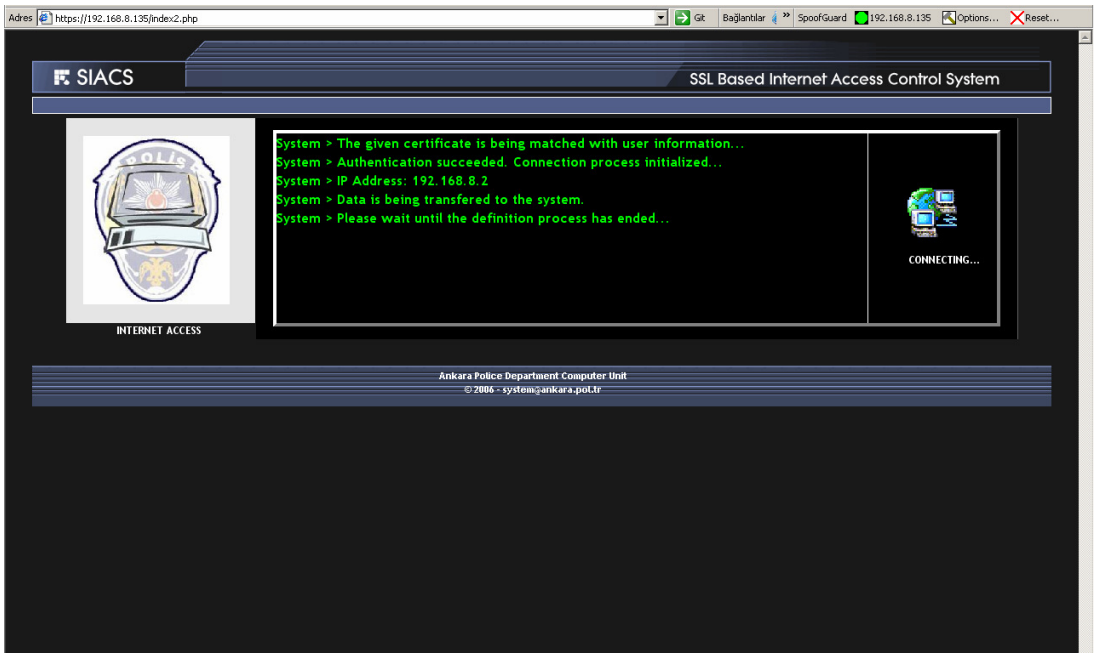

Figure 33 : Data Transfer to the Server

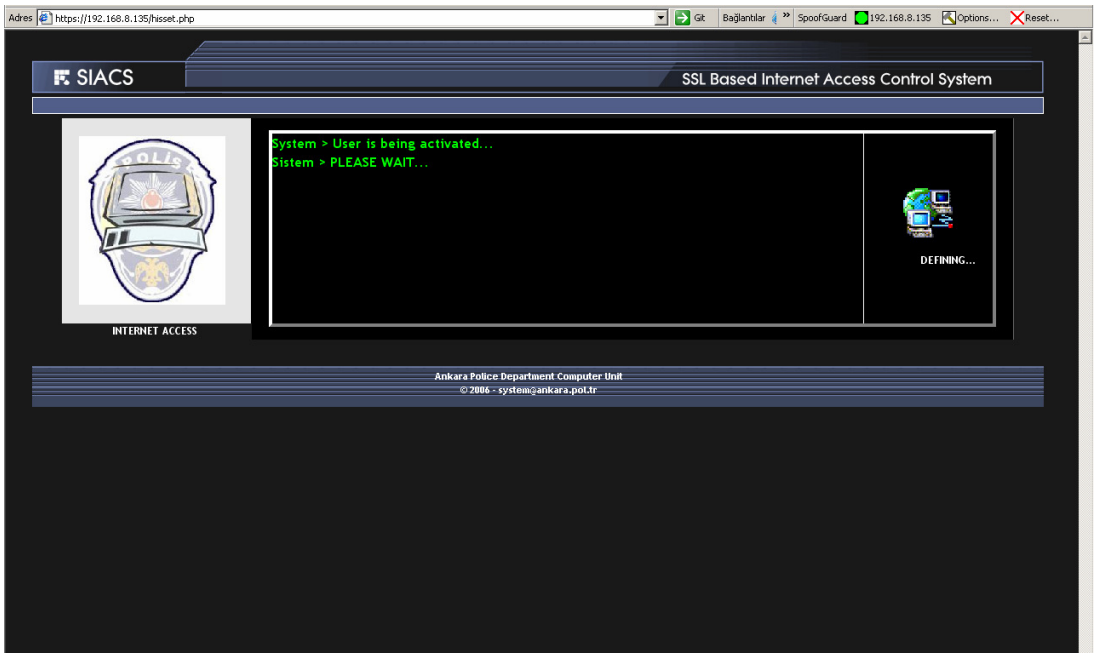

Figure 34: Definition is being performed on the server

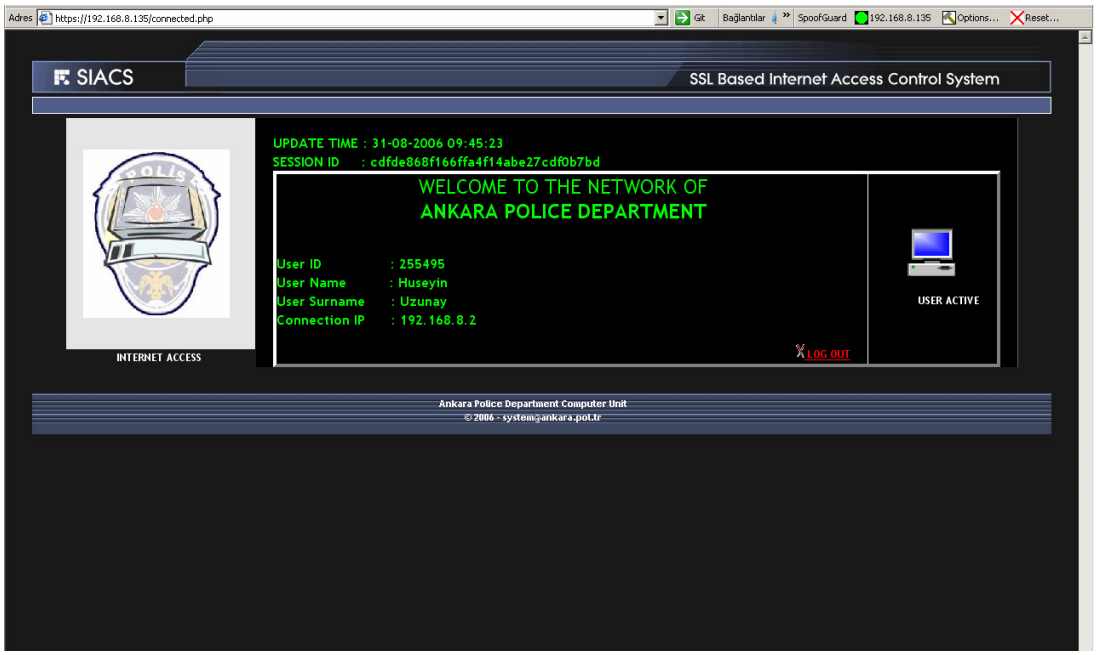

Figure 35: Authentication succeeded and Internet access is given to the user

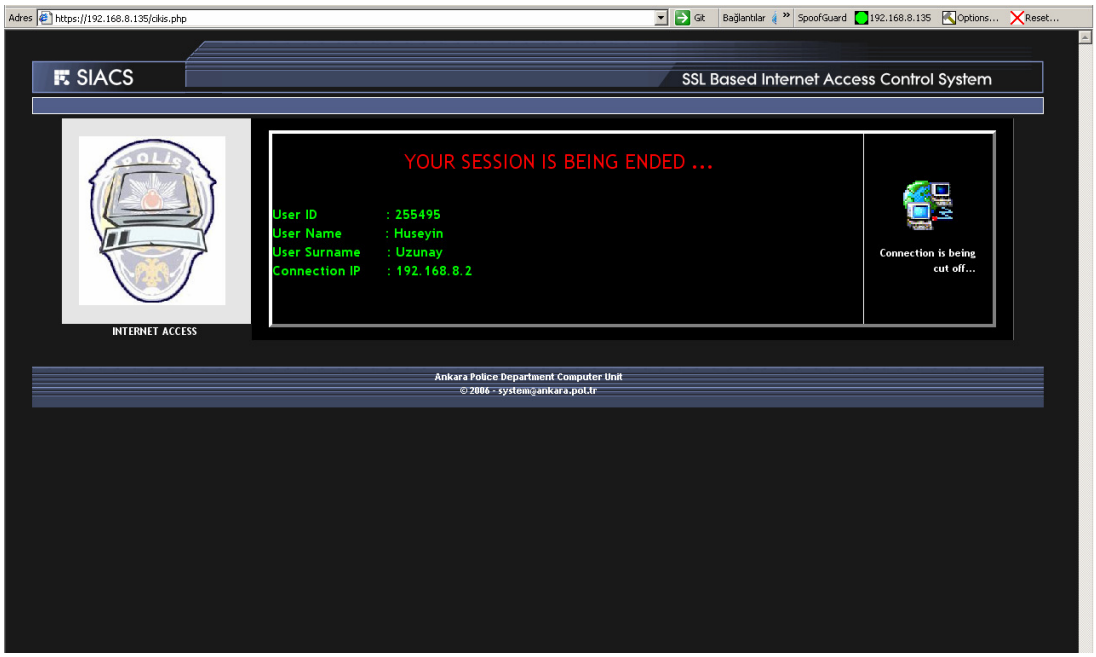

Figure 36: Log out process is initialized

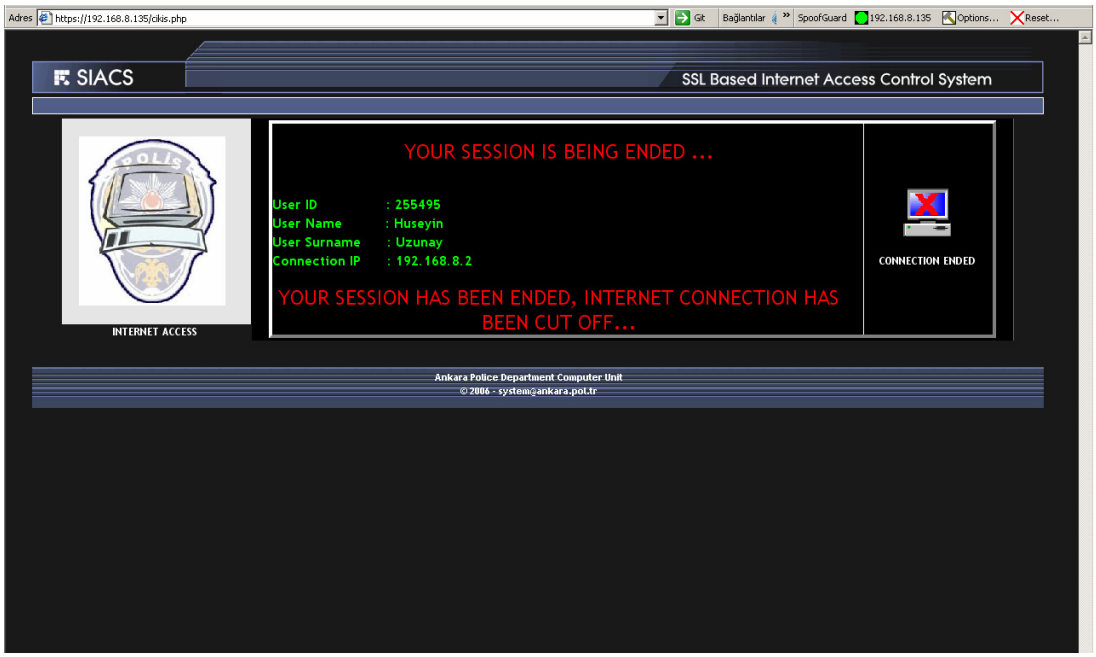

Figure 37: Log out process finished and user's Internet connection ended

APPENDIX C. CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Uzunay, Yusuf
Nationality: Turkish (TC)
Marital Status: Single
email: yuzunay@ii.metu.edu.tr

EDUCATION

Degree	Institution	Year of Graduation
BS	Gazi University Technical Education Faculty Electronics and Computer Department	2002
High School	Ankara Police College	1998

WORK EXPERIENCE

Year	Place	Enrollment
2002- Present	Ankara Police Department Computer Unit	Chief of Internet and Programming Departments

FOREIGN LANGUAGES

English, German

PUBLICATIONS

1. Yusuf Uzunay, Kemal Bicakci, "UNIDES: An Efficient Real-Time System to Detect and Block Unauthorized Internet Access", *1st International Workshop on Security in Networks and Distributed Systems (SNDS 2005) in conjunction with 11th International Conference on Parallel and Distributed Systems (ICPADS 2005).* IEEE, Computer Society, July *2005,* Fukuoka, Japan
2. Yusuf Uzunay, Kemal Bicakci, "*A3D3M: A PKI Based Digital Evidence Verification Model*", *ABG2005: National Network and Information Security Symposium*, June, 2005, Istanbul, Turkey
3. Yusuf Uzunay, Mustafa Kocak, "Child Pornography on Internet and the Difficulties in Combating", *Turkish Journal of Police Studies,* Vol:7 (2), 2005, pp. 97-116, Ankara, Turkey
4. Yusuf Uzunay, *"Network Forensics", Computer Forensics Workshop'05,* May 2005, Izmir Turkey

5. Yusuf Uzunay, Mustafa Kocak, "Digital Evidences in the Domain of Cyber Crime", *AB'05 Academic Informatics Conference,* February 2005, Gaziantep, Turkey
6. Yusuf Uzunay, "Digital Evidence Investigation Process", *The 2.nd Police Informatics Symposium*, April, 2005, Ankara, Turkey
7. Yusuf Uzunay, "Digital Attacks, It's Importance to Security Forces and Ways of Protection", *Turkish Journal of Police Studies,* Vol:5 (2) 2003 Page:131 Ankara, Turkey

HOBBIES

Table Tenis, Guitar, Poetry