Aalto University
School of Electrical Engineering
Department of Communications and Networking

Hammad Kabir

Security Mechanisms for a Cooperative Firewall

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology.

Espoo 25.02.2014

Thesis supervisor: Prof. Raimo Kantola
Aalto University

Thesis instructor: D. Sc. (Tech.) Nicklas Beijar
Aalto University

Aalto University School of Electrical Engineering **Abstract of Master's Thesis**

Author: Hammad Kabir

Name of the Thesis: Security Mechanisms for a Cooperative Firewall

Date: 25.02.2014 **Language:** English **Number of pages:** XII + 118

School: School of Electrical Engineering

Department: Department of Communications and Networking

Professorship: Networking Technology **Code:** S.38

Supervisor: Prof. Raimo Kantola, Aalto University

Instructor: D. Sc. (Tech.) Nicklas Beijar, Aalto University

The growing number of mobile users and mobile broadband subscriptions around the world calls for support of mobility in the Internet and also demands more addresses from the already depleting IP address space. The deployment of Network Address Translation (NAT) at network edges to extend the lifetime of IPv4 address space introduced the reachability problem in the Internet. While various NAT traversal proposals have attempted to solve the reachability problem, no perfect solution for mobile devices has been proposed.

A solution is proposed at COMNET department of Aalto University, which is called Customer Edge Switching and it has resulted in a prototype called Customer Edge Switches (CES). While it addresses many of the current Internet issues i.e. reachability problem, IPv4 address space depletion, so far security has generally been considered out of scope.

This thesis aims at identifying the security vulnerabilities present within the CES architecture. The architecture is secured against various network attacks by presenting a set of security models. The evaluation and performance analysis of these security models proves that the CES architecture is secured against various network attacks only by introducing minimal delay in connection establishment. The delay introduced does not affect the normal communication pattern and the sending host does not notice a difference compared to the current situation.

For legacy interworking a CES can have the Private Realm Gateway (PRGW) function. The security mechanisms for PRGW also generate promising results in terms of security. The thesis further contributes towards security by discussing a set of deployment models for PRGW and CES-to-CES communication.

Keywords: IP, PRGW, CES, Security, Traversal, NAT, Reachability, DoS

Acknowledgements

I would begin by praising Almighty Allah, the most beneficent, the most merciful. Special praise to Prophet Muhammad (Peace be upon Him) who is forever a torch of knowledge, inspiration and guidance, for humanity as a whole.

I would like to thank my supervisor, Professor Raimo Kantola, for the opportunity to work on this thesis and for his valuable insight during the thesis. His guidance, suggestions, broad experience, depth over the subject and profound knowledge was crucial for realization of this thesis.

I would also like to appreciate the efforts of Nicklas Beijar, who had a great role in identifying the thesis topic and defining the scope of this thesis. His experience and authority over the subject was commendable and inspiring.

I would like to express my gratitude to Jesús Llorente Santos for his valuable input during the thesis. His past experience with this research project and detailed replies to my queries were crucial for laying the foundation of this thesis.

I would also like to thank my friends for their support during my studies in Finland. Their support and constructive criticism has helped me improve many aspects of my life.

Finally, I would like to thank my parents for their endless support and encouragement during all these years. Their love, support, prayers and kindness has always been a great source of strength.

Table of Contents

List of Figures

List of Tables

List of Acronyms

AAA	Authentication Authorization Accounting
ACL	Access Control List
AES	Advanced Encryption Standard
ALE	Annual Loss Expectancy
AVP	Attribute Value Pair
CA	Certificate Authority
CES	Customer Edge Switches
CN	Customer Network
CSR	Certificate Signing Request
DDoS	Distributed Denial of Service
DNS	Domain Name System
DoS	Denial of Service
DSA	Digital Signature Algorithm
EAC	Estimated Annual Cost
EIR	Equipment Identity Register
FQ	Full Query
GTO	Global Trust Operator
HSS	Home Subscriber Server
HTTPS	Hypertext Transfer Protocol Secure
ICE	Interactive Connectivity Establishment
iCES	inbound CES
IDS	Intrusion Detection System
IMEI	International Mobile Equipment Identity
IMS	IP Multimedia Subsystem
IoT	Internet of Things
IPS	Intrusion Prevention System
ISN	Initial Sequence Number
ISP	Internet Service Provider
LSU	Link State Update
MD	Message Digest
MITM	Man-in-the-middle
NAT	Network Address Translation
oCES	outbound CES
PRGW	Private Realm Gateway
QN	Querying Network
RADIUS	Remote Authentication Dial In User Service
RFC	Request For Comments
RLOC	Routing Locator
RR	Resource Record
RTT	Round Trip Time
SHA	Secure Hash Algorithm
SPN	Service Provider Network

STUN	Session Traversal Utilities for NAT
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TTL	Time To Live
TURN	Traversal Using Relay around NAT

1. Introduction

Today Internet faces some new and rather strong challenges in the wake of recent scientific and technological developments. Recently, as per ITU-T, mobile users and mobile broadband subscriptions around the world are growing at a much faster rate and are replacing fixed phone and fixed broadband subscriptions[1]. This growing number of mobile users calls for support of mobility in the Internet architecture and also demands more addresses from the already depleting IP address space. The deployment of NAT at network edges to prolong the IPv4 address space life time introduced the reachability problem in the Internet. The issue is raised when a host in the public realm wants to reach a host in a private network without any prior mapping in the NAT for forwarding packets to the destination. While various NAT traversal proposals have attempted to solve the reachability problem i.e. STUN [1], TURN [2], ICE [3] etc, no perfect solution for mobile devices has been proposed.

Security has always been one of the core issues in the Internet. The marginal interest towards security in the Internet and absence of authentication mechanisms in traditional TCP/IP stack has hurt internet in many ways, including long periods of dis-connectivity because of Denial of Service (DoS) attacks. Huge spam volumes, Man-in-the-middle attacks, Internet fraud and a wide range of malicious activities owe themselves to feeble security implementations in the Internet. Today when Internet is a hub of various commercial activities, an essential part of everyday life and this reliance is only to grow with time, we argue that security must be an integral part of any Future Internet design.

To survive its expansion rate and meet the changing paradigms, the Internet needs to address the challenges related to its architecture. Realizing these challenges, a research was conducted at COMNET department of Aalto University, supervised by Raimo Kantola, for transition of Internet towards trust-to-trust principle rather than traditionally followed end-to-end principle. Implementing this concept, a prototype has been developed called Customer Edge Switching [4] [5] [6].

[1] (2013, Aug.) ICT STATISTICS. http://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx

1.1 Research Problem

Security has been an overlooked aspect in the TCP/IP stack, which has resulted in huge volumes of unwanted traffic, spam, man-in-the-middle attacks, internet fraud and a wide range of malicious activities across the Internet. CISCO 2013 Annual security report indicates the growing attack volumes and increased attack sophistication and at the same time asserts the need for security by indicating that the growth and convergence of people, services, data and things have made network connections more valuable than before [7].

While the CES architecture solves many of current Internet problems i.e. reachability issue, IPv4 address space depletion etc. it proposes the use of CES as collaborative firewalls to counter different network attacks. The currently developed CES prototype only offers policy based admission control to communicating ends and implements minimalistic security using return routability checks. However, specifics of security has generally been considered out of scope [5].

1.2 Objective and Scope

This Master thesis is an extension to the research carried on Customer Edge Switches (CES). This thesis is aimed at securing CES against different network attacks on its architecture.

The thesis analyzes the CES prototype to identify security vulnerabilities present in CES. Next, the thesis presents a set of security models comprising of different security mechanisms to secure CES against vulnerabilities in its architecture. These security models are submitted for analysis based on a set of test cases. A detailed summary of conducted tests and performance analysis will demonstrate the effectiveness of these security models. In addition, the thesis also contributes towards security by presenting various deployment models that further strengthen the security of CES.

1.3 Structure

The thesis is divided into following chapters.

Chapter 2 introduces the concept of network security, describes various known security threats and corresponding countermeasures used to ensure network security. Chapter 3 presents an overview of DNS and NAT concepts and analyzes the consequences of NAT deployment in the Internet. Chapter 4 introduces the concept of Customer Edge Switching (CES) which solves the NAT reachability issue and aims to provide trust services between disparate networks. The chapter describes the CES architecture and presents the detailed packet flow through CES.

Chapter 5 describes the CES-to-CES communication and the circular pool model of CES for inter-working with legacy hosts. The chapter highlights the security vulnerabilities present in the circular pool model and in the CES-to-CES communication. Chapter 6 describes various security mechanisms added to secure the CES architecture against the vulnerabilities present in its architecture. The chapter concludes by presenting a security model for Circular Pool and CES-to-CES communication to secure CES against attacks on its architecture.

Chapter 7 evaluates the different aspects of the security models and demonstrates the effectiveness of these models through a set of test cases and admission policies. Chapter 8 concludes the thesis and indicates the future research in this topic.

2. Fundamentals of Internet and Network Security

This chapter presents an overview of network security and discusses its various aspects. Next, a description of various attacks threatening the network's security is presented followed by a detailed description of various security mechanisms deployed to counter the network security threats.

2.1 Network Security

The advent of Internet and popularity of its services i.e. web and e-mail, has subjected the Internet to sporadic adoption by masses. Today, when the Internet offers mission critical services and is a hub of various commercial activities, it has become a fundamental component of everyday life. With businesses relying on it, the development of Internet of Things (IoT) and the presence of various cloud based services, this dependency is only to grow with time. While the Internet offers such valuable services, the other side of the mirror tells about the growing level of spam and malicious activity in the Internet. The unauthorized attempts, spam volumes, phishing attempts, Denial of service (DoS) attacks, Trojan horses, botnets etc. are on the rise and are more sophisticated than before [7]. These attacks are usually aimed at stealing personal information, thwart business secrets or restricting legitimate access to a service. We take a look at various aspects of network security in subsequent sections,

2.1.1 Network Security Architecture

The ITU-T's published security framework X.805, presented in Figure 2.1, defines a network security architecture using layers and planes. Due to the layering design, the architecture can be applied to various network types regardless of communicating end points and the underlying network. The three layers defined by the framework are the infrastructure layer, the services layer and the application layer. The infrastructure layer deals with the security issues of the network transmission facilities and individual network elements i.e. routers, switches etc. The services layer deals with the security of the services offered by the Internet to the user, whereas the application layer security deals with the security challenges faced by network based applications i.e. services that run on collaborative principles.

The security planes in the X.805 framework address the security activities performed in the network. The management plane deals with Administration, Operation and Maintenance related activities. The control plane provides security for signaling aspects of connection i.e. connection establishment etc. The end user plane is concerned with the security of the network access and the use of the network by customers, as well as protecting the end-user data flows [8].

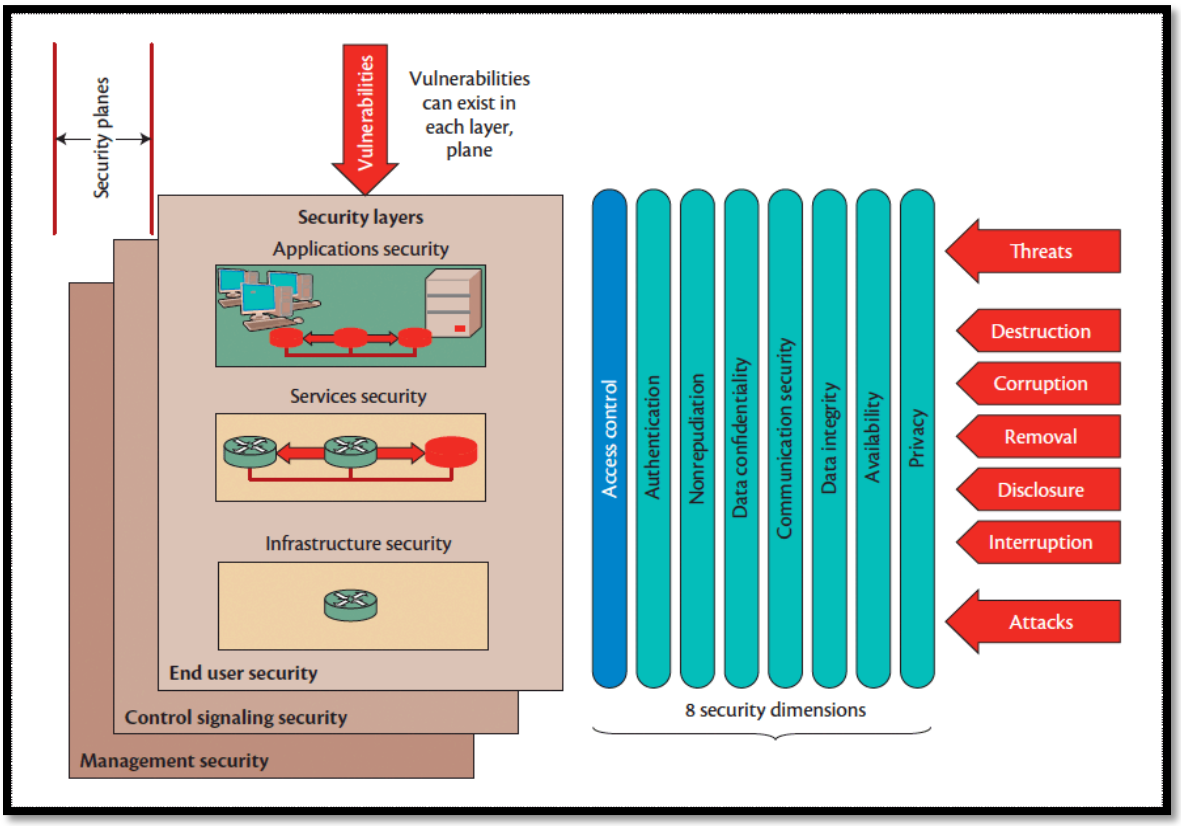


Figure 2.1 Security framework presented in ITU-T X.805 framework

2.1.2 Network Security Dimensions

The X.805 security framework identifies eight dimensions to the network security. Privacy is identified as the prime motivator for security, whereby the user restricts the amount and the kind of information available to others. Whereas, the data confidentiality is aimed at controlling the unauthorized access to user's data i.e. using encryption or access control mechanisms. The data integrity dimension ensures the receiver that the data received has not been altered by an unauthorized party i.e. man-in-the-middle attack.

Non-repudiation is the feature of security which binds an action to the user, and hence does not let the user refute this action later on. This action can be sending or receiving data, content creation, making calls etc. The availability dimension guarantees that a resource or a service will remain available to a legitimate user at all times. The access control security dimension protects the network against unauthorized use or access of network resources.

Authentication, which generally follows identification, is an important security dimension and guarantees that the claimed identity of an entity i.e. user, services or applications, is true. This is an important check against the attacks where a hacker attempts to masquerade as a legitimate user and hence access the network resources. Network security imposes the condition that data is being exchanged with the claimed legitimate user [8].

2.1.3 Security Threats and Risks

A network's security is at risk only when a security vulnerability is combined with a security threat. While various approaches have been designed to determine and analyze the security risks of a system, these approaches can be divided into quantitative and qualitative risk analysis. Quantitative risk analysis approaches try to compute Annual Loss Expectancy (ALE) or Estimated Annual Cost (EAC) by multiplying the probability of an event with the likely loss that will occur. Whereas, the Qualitative risk analysis computes the risk by identifying threats and vulnerabilities of a system.

The vulnerability in a system is a potential attack point which can be the result of various factors i.e. weak system design, buggy code or minimal attention to security details of the system. Whereas, a security threat is identified as an activity which aims to violate the network security policy i.e. unauthorized access etc. The security threats that do not change the state of the system are called passive security threats i.e. eavesdropping or passive monitoring etc. Whereas, the security threats that change the state of the system are known as active security threats. Examples of active security threats include unauthorized access, DoS attacks etc. as such attacks cause loss of data integrity or affect the system availability [9].

A vulnerability and a security threat do not risk the system's security, when viewed separately. However, the security of a system is at risk when a security threat combines with the corresponding security vulnerability. The risk to a system's security is often explained in literature by different variants of the following equation [10]:

Risk = Threat * Vulnerability

2.2 Network security threats

As discussed previously, a security threat is an activity aiming to subvert the system security. Various network attacks have been developed over the time span of the Internet, which can be classified into various categories based on the attack objectives. While various basic and advanced attacks are launched to compromise the network security, this section only presents an overview of some of the common network attacks.

2.2.1 Denial of Service Attacks

Denials of Service (DoS) attacks target the availability of system resources i.e. to make system resources unavailable for legitimate users. The principle behind a DoS attack is to flood the victim host with excessive connection requests, more than it can serve. As a result, when a connection request from a legitimate user arrives it is not served because all of the victim's resources are reserved for connection requests from an attacker that are never completed. This results in denial of service to a legitimate user. A denial of service attack is often carried in combination with other attack types to increase its sophistication i.e. IP address spoofing, Smurf attack [9] [11] etc. The denial of service attack and its different variants are described in the following sections.

TCP SYN Flooding attack

TCP connection establishment process requires the exchange of TCP packets SYN, SYN/ACK and ACK packets between the source and the destination of a connection, respectively. Data packets are exchanged only after the connection is established. The server hosting a resource normally allows a limited number of simultaneous connection requests and any new connection request is served only after earlier queued requests have been served. Attackers make use of this knowledge and they bombard the victim with TCP SYN packets, putting the victim in a half-open state. Following the protocol, the victim host replies with TCP SYN/ACK packet to the claimed sender. The next expected step from a legitimate user is to send a TCP ACK packet to the destination host in order to complete the connection establishment.

An attacker chooses not to perform the ACK part of connection establishment and this places the victim machine in the half-open connection state. Once the number of half-open connections goes beyond the maximum simultaneously servable connections, any new connection request received is dropped by the victim and this results in denial of service to a legitimate user [9] [12].

Distributed Denial of Service Attack

Distributed Denial of Service (DDoS) attack is a sophisticated version of a single source based DoS attack, where usually a large number of hosts participate to launch an attack on the victim machine i.e. by flooding with connection requests. An attacker normally subverts a large number of host machines over a period of time and installs the attack software in them, after elevating the access privileges over the machine. The installed attack software puts these compromised hosts in Master-Slave configuration with the attacker's host. These compromised hosts are referred to as amplifiers in DDoS attacks. At a predetermined time or at the command from the master (i.e. attacker), these compromised hosts bombard the victim host with unsolicited packets. This results in the depletion of resources at the victim host, making it unable to serve any new connection request and hence launching a denial of service for the legitimate traffic [13]. Figure 2.2 presents a scenario where an attacker bombards the victim server with the help of amplifier hosts to launch a distributed denial of service attack.

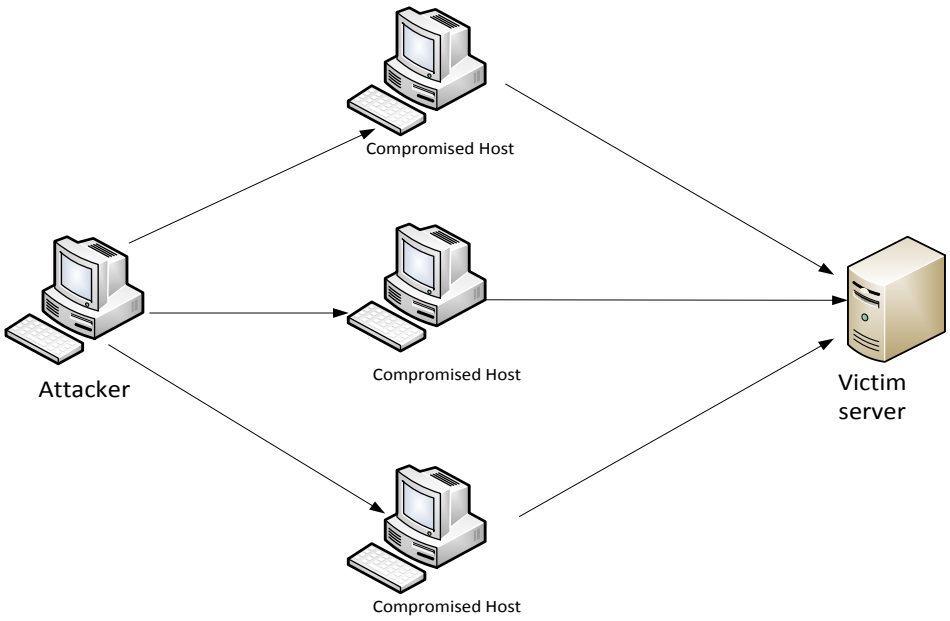


Figure 2.2 Distributed Denial of Service attack

When hundreds of compromised hosts act simultaneously, the result is a huge volume of traffic effectively taking the victim's machine down. Huge volumes i.e. in hundreds of Gigabits/sec of current denial-of-service attacks are because of this distributed nature of attack [7]. The difficulty in tracing a DDoS attack is that the amplifier hosts are generally legitimate hosts and are not part of the attack by intent, but unknowingly. A well planned attack may program the amplifier hosts to spoof their addresses to further hinder tracing of the original attack source.

The DDoS attack consumes a vast amount of network resources in Internet service provider (ISP) networks. A DDoS attack targeted even at a minor web server has the potential to bring the whole ISP's network down, and hence can affect thousands of ISP customers. Similarly, a DDoS attack against the services like Domain Name System (DNS) or any other single point of failure can affect large portions of the Internet. Cookie mechanisms and *ICMP traceback messages* among others have been proposed as the countermeasure against DDoS attacks [13] [11].

Smurf attack

Just like TCP, the ICMP protocol can be used to launch DoS attacks. In a smurf attack, the attacker sends the ICMP echo request (ping) messages containing the victim's address forged as source address towards the broadcast address of a network. This broadcasts the ICMP echo request message to all the hosts in the network. Since the ICMP protocol defines the ICMP echo reply message in response to ICMP echo request message, so a large number of ICMP reply messages are received by the victim's host. The huge frequency and volume of these unsolicited reply packets results in slowing down of the victim's host and denial of services to the user of this host [13].

Ping of death is also an example of flooding DoS attacks where the victim host is flooded with malformed ICMP ping packets in hope to crash the victim's system [11].

2.2.2 Man-in-the-middle attacks

A Man-in-the-middle (MITM) attack happens when a hacker sits in the connection established between two communicating ends. Such attacks are launched with tools like packet sniffers, ARP spoofing or DNS cache poisoning, which route the traffic to the MITM's host and let the attacker eavesdrop on the exchanged traffic. Figure 2.3 illustrates a man-in-the-middle attack

where the MITM host intercepts the communication between two trusted hosts Host-A and Host-B. After intercepting the communication, the MITM host can corrupt or manipulate the data packets and forward them to the destination. In some cases, the MITM attacker impersonates a trusted party to establish a connection with the victim host. If successful, the attacker can access the victim's confidential information. Such attacks are often used for theft of information, session hijacking or for disrupting an ongoing communication [9].

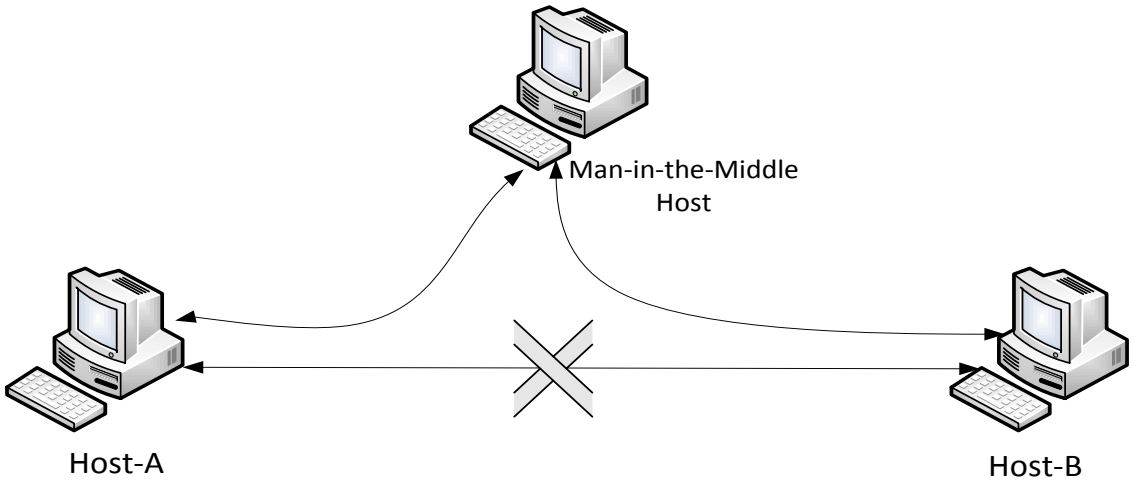


Figure 2.3 Man in the middle attack

2.2.3 IP Spoofing Attacks

In spoofing attacks, an attacker replaces the source address of the packet to conceal its identity. The IP spoofing can be aimed to masquerade as a trusted party or for launching an attack at the victim host. While IP spoofing concept seems trivial, it serves as launch point for various attack types. The Denial of Service (DoS) attacks discussed in section 2.2.1 and various other network attacks use address spoofing to conceal the identity of the attacker. This hardens tracking down the attacker as the hacker's identity is never revealed [13].

2.2.4 Trojan horse Attacks and Viruses

The term Malware refers to a malicious software that is attached or often disguised in a program or in an e-mail content to execute unwanted activity on the recipient host. Malware can be classified in different categories discussed next,

Viruses and worms are two types of malicious software that have the ability to replicate their own copies and propagate. Both can create the damage ranging from minor irregularities to

corrupting application, deleting data or causing denial of service conditions. The difference between virus and worms comes from operational perspective. Viruses need to be executed before they can cause damage, either by a program or by human help i.e. by tricking the user. Whereas, worms are self-sufficient and do not need any assistance from a program or a human to propagate and execute themselves[2].

A Trojan horse is another form of malware which tricks the user to executing them i.e. by being part of software or e-mail. A Trojan horse can infect the victim host in different ways i.e. stealing or damaging the private data, creating back-door accounts and making the system prone to various security threats. Bots are a more lethal form of malware, as they automate the steps involved in an attack and provide the attack services that otherwise would have required human intervention [12]. Bots are much more versatile than other malware types and can cause damage ranging from logging keystrokes, gathering passwords to launching DoS attacks. The backdoor accounts created via Trojan horses or Bots can later be used for launching the DDoS attack, where the infected host serves as an amplifier host in the attack.

2.2.5 Spam

Spam over the Internet exists in many forms i.e. a fake website, unsolicited promotional e-mail or viruses hidden in tempting graphics etc. Spam initially started by sending unsolicited advertisements in e-mails, but with time, it has also evolved and today it has become a source of distributing viruses, worms and other malicious codes that impact the system security in a negative way. Spam over the Internet is a wide-spread problem and it affects both the users and service providers in terms of time, privacy and system security. Spam exists in many forms besides e-mails i.e. DNS cache poisoning, where an attacker can give incorrect information about the address of a company webpage and then can redirect the traffic to another site [14]. Cisco Annual Security Report for 2009 reveals that social media has also contributed to the rising spam volume and has given a new dimension to the spam.

2.2.6 Eavesdropping

Eavesdropping can be categorized in two types, active and passive eavesdropping. Passive eavesdropping is where an attacker listens on the packets exchanged between the communicating

[2] (2013, Jul.) Cisco. http://www.cisco.com/web/about/security/intelligence/virus-worm-diffs.html

end points. It is an example of passive security threats as the state of the victim host does not change because of this attack. Whereas, Man-in-the-middle attack described in section 2.2.2 is an example of active eavesdropping, where an attacker tries to intercept and manipulate a data flow, and it masquerades as a legitimate sender. Eavesdropping is generally useful against plain text protocols, data flows or insecurely established connections. An intruder can use packet sniffers, IP spoofing or other attack tools to eavesdrop on a connection [9].

2.2.7 Infrastructure Attacks

Internet consists of various components i.e. routers, DNS, NAT etc., which are used to provide connectivity services between communicating end points. Attackers often target these infrastructural components to launch an attack on the victim. If planned well, these attacks have the potential of large scale catastrophe on security and economic vitality of a society.

Routing based attacks are one such example of infrastructural attacks. Routing misconfigurations result in frequent routing troubles. Potential threats to the routing infrastructures also come from spoofing attacks, where an attacker sends false routing updates about the victim that can redirect the traffic to an invalid address and hence result in DoS to users of a service. Similarly, a compromised router can listen and alter the packets passing through it or simply drop all the packets, and hence make the destination unreachable. Routers often generate Link State Updates (LSU) to notify the status of the links. These updates are sent to all neighboring routers, who based on the received information update their routing table. A malicious or a compromised router can send bogus routing updates to neighboring routers, and can cause all the traffic to redirect to itself and then eavesdrop or simply drop the traffic. This is known as poisoning of routing tables, and the resulting mishandling of packets can put a network in congestion or in the denial of service state [15] [16].

DNS is another critical component of the Internet which translates domain names to corresponding IP addresses. DNS cache poisoning is also an infrastructure attack where false information fed to a DNS server can redirect the traffic to an invalid destination. An attack against the Internet infrastructure can target large portions of the Internet, and hence can affect thousands of users [16].

2.2.8 Social Engineering

A non-technical type of security threat comes from social engineering. Social engineering is an art of tricking users or employees into performing actions which endanger the system or network security. All the security protections designed for safeguarding the network or system cannot guarantee the system security alone if the user of the system gets tricked into aiding the attacker. Phishing user's confidential information through an e-mail message that appears to have come from a legitimate source or redirecting the user to a fake website based on hyperlink in the email content are example of social engineering attacks [17]. Compliance to security policy and security awareness culture among users and employees is known as the best defense against social engineering attacks [7].

2.2.9 Other Attacks

Internet attacks can be classified into various categories based on the attack objectives. A set of these attacks have been described in previous sections, but various attack types are left out in discussion i.e. TCP sequence guessing, TCP RST based attacks, LAND attacks etc. The presence of these attacks is mostly an attribute of the absence of Identification, authentication and authorization mechanisms in the TCP/IP protocol stack. Several security mechanisms have been developed as countermeasures against these attacks. While identification of new security attacks and development of corresponding protection mechanisms is an ever evolving field, we take a look at some of the current security practices against well-known security threats in the next section.

2.3 Network security protections

Security protections or security mechanisms are the processes or techniques used to enforce system security and thwart an attacker's attempt to subvert the security of a system. Network security mechanisms comprise of three basic elements: *prevention*, *detection* and *response*. All the security mechanisms or security policies are designed to implement this security trinity and hence protect the system against security threats [9]. In this section, we take a look at different security mechanisms used to ensure the network's compliance with security requirements.

2.3.1 Cryptography

Cryptography is a word of Greek origins which means "secret writing". In network security, this term refers to the science and art of transforming plain text messages into a form which is secure and immune against attacks from a hacker [18]. Cryptography consists of a set of different techniques which are aimed to meet the security dimensions of the X.805 security framework, described in section 2.1.2 i.e. to provide data confidentiality, data integrity, non-repudiation and privacy. We will take a look at these techniques in the following sub-sections.

Symmetric and Asymmetric Key Encryption

Encryption refers to the process which converts a plaintext message to an unintelligible text called ciphertext. The original message can be derived from the ciphertext by decryption, using the shared secret. To keep the privacy of the message, the parties involved in the communication must keep the "key" secret. The encryption techniques can be divided into two types, based on keys involved in the process: symmetric (private/secret) key encryption and asymmetric (public) key encryption techniques.

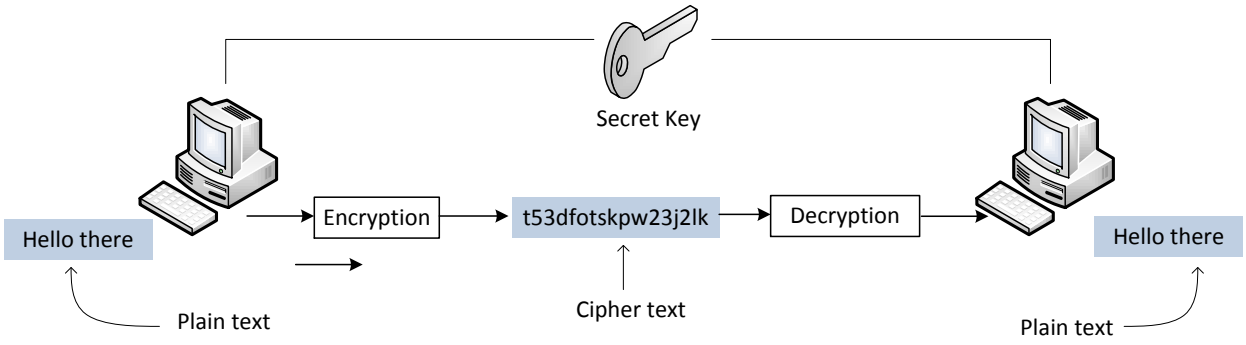


Figure 2.4 Encryption/Decryption process

Figure 2.4 describes symmetric key encryption, where the sender and the receiver share the same key to encrypt and decrypt the communication. Symmetric key encryption can be used with either stream ciphers or block ciphers. Stream ciphers perform a bit wise operation when encoding a message stream, whereas block ciphers encode a block of data at once. The biggest drawback of symmetric key encryption is the distribution of the key, as both the sender and the receiver need to be aware of the same key. A compromise during key distribution can result in decryption of all the encrypted flows to an unauthorized party. DES, 3-DES, AES, CAST etc. are example of symmetric key encryption algorithms [18].

Asymmetric key encryption, also called public-key cryptography, uses a set of two distinct keys: the private-key and the public-key. The private-key is kept secret and is never disclosed to anyone while the public-key is can be accessed by anyone. The sender encrypts the communication with the public-key of the receiver, which is publicly available, and then the receiver can decrypt the communication using its private-key, which is available to none else. This ensures the privacy of a communication using public-key cryptography.

While symmetric key cryptosystems are limited to ensuring the privacy of a communication, the public-key cryptography offers much more versatile roles. Public key cryptosystems offer the support for digital signatures and key exchange algorithms, in addition to encryption/decryption. Hence, they provide authentication and non-repudiation services to the sender and the receiver. Some of the most widely used public-key algorithms are Diffie-Hellman, RSA and the Digital Signature Algorithm (DSA) [11].

Message Digest

In cryptography, message digest functions are used to preserve the integrity of a message. These functions take a message of an arbitrary size in input and generate an output of fixed number of bits, called Message Digest (MD) or hash value. A hash function is a one-way function and the original message cannot be recovered from the corresponding MD. A hash function guarantees that no two unique messages would result in the same message digest. Because of the second property, MD is often known as the fingerprint of the message, as it is uniquely associated with a message. SHA1 [19] and MD5 [20] algorithms are two examples of message digest functions used in cryptographic security.

In computer networks, a message digest computed at the sender accompanies the message sent towards the destination. For a corrupted reception, the message digest received in the packet would be different from the MD computed over the received message, and hence the compromise of message integrity is detected at the receiver.

Signature

As described before, message digest is used to ensure the message integrity. However, an MD is not simply transmitted over the communication channel in cryptosystems. Rather, cryptosystems compute the message digest of the data and encrypt it with sender's private-key. This generates the digital signature of the message, which is then transmitted over the communication channels.

The use of digital signature ensures the receiver that the message has indeed come from the claimed sender, as only the sender's private-key could have generated this signature. This concept not only ensures message integrity but also guarantees authentication and non-repudiation, since only the claimed sender could have accessed the private-key which generated this signature [18].

The receiver verifies the digital signature by decrypting the signature received, using sender's public-key. Next, the MD is re-computed over the received message and the result is compared to the MD decrypted from the digital signature. Once verified, this guarantees the message integrity and also authenticates the sender [18]. When using the signature, the rest of the message is still sent as clear text and hence the data confidentiality is not provided and eavesdropping is still possible.

Certificate Authority

The success of the public key cryptography is centered on the principle that the private-key remains secret to the entity while the public-key can be broadcasted to the recipients of a communication. The lack of authentication mechanisms on the received public key leaves a window of opportunity for an attacker who could distribute a false public-key to the victim host and hence could easily decipher the encrypted confidential information of the victim.

The solution proposed to counter this vulnerability is to embed the trust in a trusted third party, often called Certificate Authority (CA). An entity can seek a digital certificate from a CA by providing its public-key and a necessary set of information in a Certificate Signing Request (CSR). The CA issues a digital certificate to the certificate requestor after performing necessary validation checks on the provided information. The digital certificate issued by the CA binds the identity of an entity with the public-key, and hence provides an independent confirmation that the entity is who it claims to be. The issued digital certificate is signed by the CA's private-key and can be verified using the public-key certificate of the CA, which is publicly available. After the certificates have been acquired, the participants in a communication can exchange the certificates instead of the public-keys, and the receiving entity can verify these certificates using public-key certificate of the CA. A number of different entities issue digital certificates in the Internet world e.g. VeriSign, GTE, AT&T and Microsoft [11].

X.509 Certificates

With the introduction of certificates, different certificate authorities started issuing digital certificates in different formats. For something to be used globally, the demand for a standard certificate format was pressing. ITU-T made the first attempt to launch a standard for public key infrastructure and specified a structured format for digital certificates called X.509 certificate. The X.509 certificate format has been updated thrice and the current version number is 2 [18]. A X.509 certificate consists of the following fields,

Version: This field defines the version of X.509 certificate, the current version number is 2.

Serial Number: This field contains a positive unique number assigned to each certificate.

Issuer: This field identifies the certification authority which has issued and signed the certificate. The field describes in a hierarchical manner: country, state, organization, department, and so on.

Period of validity: This field defines the starting and ending times when the certificate is valid.

Subject: This field carries the identity of the entity to which the public key belongs. It is also a hierarchy of strings, similar to the 'Issuer' field, defining the beholder of the key.

Subject's public key: This field carries the public key associated with the "subject". It also defines the algorithm and corresponding parameters to be used with the key.

Issuer unique identifier: Two issuers of the certificate can use the same issuer field value, if the issuer unique identifiers are different.

Subject unique identifier: Similarly, this optional field allows two different subjects to have the same subject field value, if the subject unique identifiers are different.

Extensions: The extensions field defined in X.509 v3 certificates provides the methods for associating additional attributes with the users or certificates.

Certificate signature: The certificate signature field contains the digital signature computed by the CA using its private key. By generating this signature, the CA endorses the binding of subject identity and the corresponding public key in the certificate.

Certificate Signature Algorithm: The signature algorithm field contains the identifier for the cryptographic algorithm used by the CA to sign the certificate. The algorithm identifier is defined by the ASN.1 structure [21].

2.3.2 Firewalls

A firewall is a fundamental security component which isolates a private network from the public Internet. The isolation guards and protects the internal network against attacks from the public Internet. The firewall, usually installed at a gateway between two networks, filters the traffic flowing in and out of the private network. The ingress and egress filtering in a firewall monitors/filters the incoming and the outgoing traffic of the network and allows or disallows the traffic based on a pre-configured set of rules in the firewall.

A firewall is a combination of hardware and software deployed to protect the private network against possible intrusion from hackers in the external network. Based on the location of deployment, firewalls can be classified in two categories: network based firewalls and host-based firewalls. Network based firewalls are deployed at network edges and they filter the traffic going in and out of the network. Whereas, host-based firewalls implement the host specific traffic filtering policies. The host and the network based firewalls when combined can provide defense-in-depth against unauthorized attempts. While a firewall aims to protect a network or a host against attacks, it adds delay to the communication because of the processing involved in each flow. Based on the processing involved, firewalls can be classified into Network level firewalls and Proxy Firewalls [11].

A network level firewall operates at the network layer and the transport layer of TCP/IP protocol suite. The network level firewalls are usually the screening routers which filter each inbound/outbound packet based on the IP address or the port numbers. Network level firewalls offer two types of filtering: static packet filtering and stateful packet filtering. The static packet filtering employs a set of fixed rules to filter the Internet traffic and these rules once configured remain unchanged regardless of the traffic nature passing through the firewall. While the stateful packet filtering keeps track of earlier traversed traffic, and hence can employ more sophisticated checks for malicious activity in the Internet traffic. Moreover, the stateful firewall uses a dynamic set of rules, which can change as per traffic condition [11] [18] [9]. Figure 2.4 presents both types of network firewalls deployed to protect network against attacks.

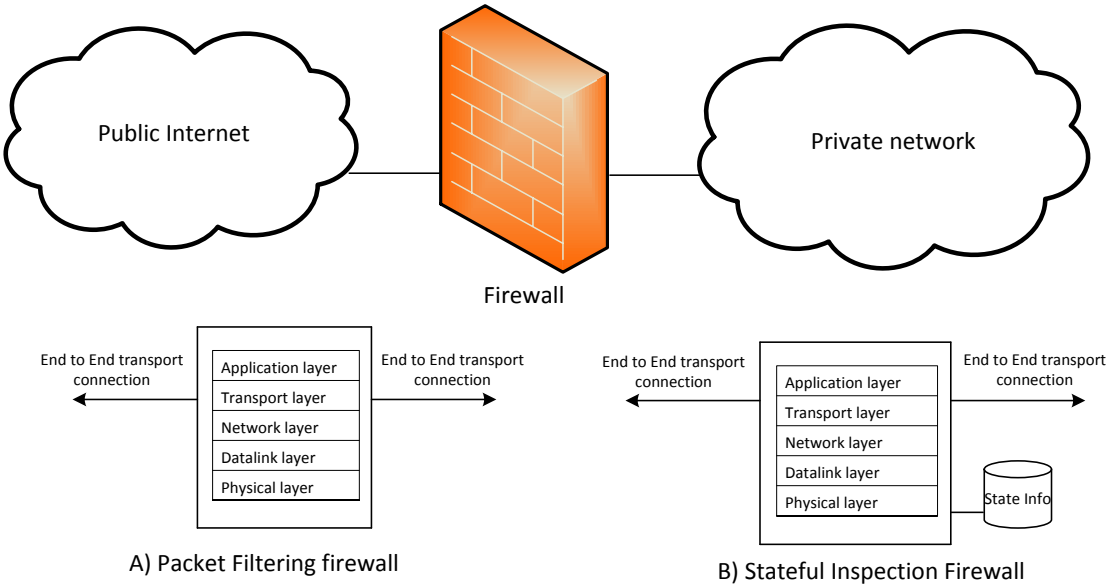


Figure 2.5 Packet Firewalls and Stateful Firewalls [9]

A proxy server stands between a customer network and the public Internet during a connection, acting as the Man in the middle. Therefore, there is no direct connection between a private network host and the remote communication host. The proxy runs on the firewall allowing controlled access. A proxy server has two implementation types: Application level firewalls and circuit level firewalls. Application level firewalls, or gateways, come in handy when there is a need to filter a message based on the information available in the payload i.e. at the application layer. This provides a deeper level of packet inspection than any other firewall type. Whereas, circuit level firewall filters a packet at the transport layer of the TCP/IP stack. They add many services to packet firewalls and are more prohibitive in nature i.e. due to encryption of traffic flows [11] [22] etc.

2.3.3 Intrusion Detection Systems

An Intrusion Detection System (IDS) is a device or a special purpose software which detects malicious activities or attack attempts in the network traffic, and reports it to the network administrator. The intrusion detection approach is based on the assumption that an intruder's behavior differs from a legitimate user in ways that can be quantified. Based on this fact, the intrusion detection employs techniques which are a combination of monitoring, analysis and response. Monitoring and analysis are passive techniques as they can be carried out independently, whereas, the response involves sending alerts to the system administrator, or

configuring an updated set of rules to counter the attack. Detection accuracy is a critical factor in the IDS performance and it needs to be maintained continuously to minimize the false positives and false negatives detections i.e. detecting false attack and neglecting an actual attack, respectively [23].

IDS can be categorized in two ways. An approach similar to firewalls categorizes IDS in Host-based IDS and Network-based IDS. A Network-based IDS detects an attack targeted at the network and a Host-based IDS handles an attack against the host. Intrusion detection systems can also be categorized based on the detection approach used by the IDS to spot an attack. Traditionally these approaches are signature-based detection and anomaly-based detection.

Signature-based intrusion detection relies on the pattern that uniquely identifies an attack. If an activity matches to a known signature, the IDS identifies and reports it as an attack. However, the drawback of signature-based IDS is that they can only detect the attacks with known patterns and are immune to any new attack type. Statistical-based IDS use anomaly-based detection approach to identify attacks. Statistical-based IDS is preferred over signature-based IDS because of its potential to detect and recognize new attacks, even without a known pattern. The basic principle used here is to define a "normal" behavior statistically, with certain allowable deviations, and any activity that goes beyond these deviations is detected as an intrusion [11] [23]. The false positive or false negative detection of the IDS depends on how strict or how loosely the attack pattern or the 'normal' behavior is defined in the IDS.

An intrusion prevention system (IPS) reacts more actively towards an attack by implementing 'prevention' aspect of the security trinity. The IPS prevents an attack against the network by combining the traditional monitoring, analysis and detection aspects of the IDS with more active automated responses, i.e. automatically reconfiguring firewalls to block the attack or to carry a deeper packet inspection [23].

2.3.4 Logging, Auditing and Reporting

Logging is the process of recording the network activity in log files. This is an important concept in network security where network activity is recorded for a later analysis. Firewalls or IDS often use logging to report the attacks or vicious activities to the network administrator or for a third party audit. A set of processes and techniques are then employed to detect attacks in a specific environment, using logs as the primary source of information [24].

An audit by definition is an independent review of a given subject. The audit process in network security reports divergence or conformance of network activities to the established security standards. The auditing process can span over many areas i.e. operational audits, system audits, activity and usage audits. The audit process heavily relies on the logs, and given well maintained log files, the audit process verifies compliance to network policies and can report if the network's security procedures and practices need to be updated. Since log files can be subjected to alteration by the hacker, which would result in a wrong audit, log files must be copied to a secure location. The audit process serves as a feedback for the network administrators, to employ efficient security mechanisms against the changing network attacks. Besides log files, auditing may involve employing a team of white-hat-hackers who fake an attack against the network to check the network's compliance with security policies [23].

2.3.5 Access Control

In terms of network security, access control refers to the processes which guarantee that only a legitimate user gets access to the network resources and performs activities within an authorized level. The access control mechanisms consist of three steps: 1) authentication of users 2) authorization of privileges and 3) accounting (or auditing) of user actions. Lack of authentication mechanisms in the TCP/IP protocol suite has given birth to different attacks and has risked the security in the Internet. The access control mechanisms attempt to better the situation by authenticating the sources, digging out relevant access privileges and then keeping track of the resources used.

Authentication is the process of verifying the source identity, and it generally follows the identification. The source first identifies itself by providing an identity, and the authentication determines if the provided credentials belong to the claimed entity. Password, PIN, token or digital certificate are examples of authentication mechanisms.

Authorization determines the level of access an authenticated user has to the network resources i.e. permissions to read, write, or execute etc. Discretionary privileges can be defined by an Access Control List (ACL), which determines if a source should be granted or denied the access to resources. Accounting refers to the process that keeps track of network resources consumed by the user. This may involve recording the user activities in audit trails or logs, which can be later used to determine the user's compliance or deviance from the network security policy. For example, numerous failed logins by a user can indicate an intruder's failed impersonation

attempts. Besides auditing users, the accounting process can benefit a network in many aspects i.e. in capacity planning and billing the users [23] etc.

The emergence of new technologies and applications, such as wireless networks and mobile IPs, have increased the requirements for authentication and authorization, and access control mechanisms have grown in complexity. Network security refers to the term AAA (Authentication, Authorization and Accounting) to define the access control architectures. The AAA architectures normally consist of three entities: the user requesting the access, an Access Server at the network edge controlling access to the network and an AAA server that grants or denies the access based on the access credentials provided by the user [25]. The AAA architectures require a standardized protocol between the access server and the user information repository in the AAA server in order to exchange the access control related information. The information exchanged via this AAA protocol is used to decide the fate of an incoming user request. Two of the well-known AAA protocols used to exchange access control information are RADIUS [26] and DIAMETER [27]. The next section presents an overview of the DIAMETER protocol.

2.3.6 DIAMATER Protocol

The RADIUS protocol was proposed and designed to exchange AAA capabilities, but the evolution of network applications and protocols gave birth to new requirements and hence new mechanisms were required to authenticate the users. The need for a more extensible and generic AAA protocol was realized and met using the DIAMETER protocol, which inherited many features from the RADIUS protocol.

The DIAMETER protocol defines diameter messages for carrying AAA related information, in an attribute-value pair (AVP) format. The DIAMETER protocol allows the definition of new Diameter applications by extending the DIAMETER base protocol, defined in RFC 3588. Each application is identified by its application identifier and can add new command codes and new mandatory AVPs to the base protocol.

Unlike the client-server based RADIUS protocol, the DIAMETER is a peer-to-peer protocol where a Network Access Server (NAS) residing at the network edge usually acts as a DIAMTER client. The NAS acts as gateway and hence controls the access to the private network. During the authentication, the DIAMETER client sends the received user credentials as DIAMETER Access Request messages to the DIAMETER server and requests the authorization. The DIAMETER

server is responsible for processing the request message, authentication of the user and returning the access parameters necessary for the DIAMETER client to deliver the services. Upon receiving the access request, the DIAMETER server carries the verification process locally and responds with: access denial, access granted with authorization parameters, or throws additional authentication queries to the user requesting the access [25]. Figure 2.6 presents a DIAMETER based access control setup, where the remote client Host-A requests a service in the private network and is being authenticated by the Network Access Server (NAS) and the Diameter Server, using the DIAMETER protocol.

For accounting purposes, the DIAMETER node that receives a successful authentication or authorization from the DIAMETER server collects the accounting information for the session. The Accounting-Request message transmits the accounting information to the DIAMETER server, which replies with the Accounting-Answer message to confirm the reception. The DIAMETER server also conveys the DIAMETER client about the expected behavior of accounting messages i.e. how often the accounting record should traverse from the client to the server [27]. As mentioned before, this accounting information can serve multiple purposes i.e. billing, capacity planning and auditing the user for access services etc.

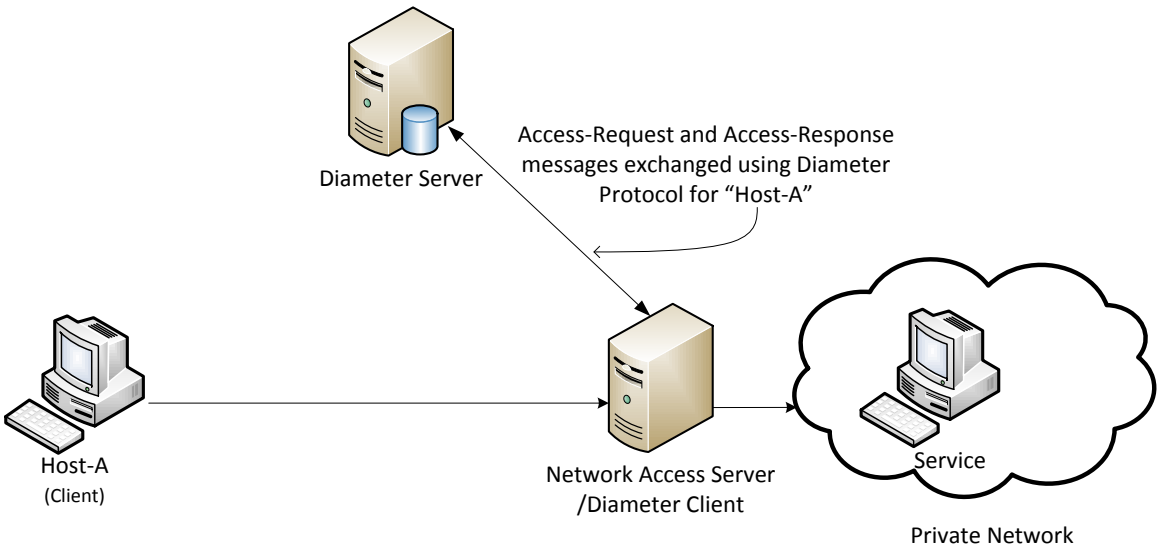


Figure 2.6 DIAMETER protocol usage for Access Control [25]

2.3.7 Honey Pots

A honeypot is a system or a network deployed often to trick the hacker in believing to having found a potentially vulnerable target while the actual network runs safely apart. A honeypot is equipped with comprehensive and reliable capabilities for monitoring and logging all the activities. The logged information in honeypots can be used to learn the attack tactics and then the corresponding security mechanisms can be deployed to foil such attacks in the future. This protection mechanism acts as both, a forensics tool and a line of defense from the network security perspective [23].

2.3.8 Home Subscriber Server

In mobile communication networks, a Home Subscriber Server (HSS) is a central repository which stores user-related information. The information in the HSS is required to handle calls and multimedia sessions, and it includes location information, authentication and authorization information, user profile information, subscribed services and name/address resolutions [28] etc. In terms of network security, the HSS is and can provide security services necessary to authenticate the users. The usage of HSS is becoming more and more important in the wake of diminishing boundaries between IP networks and Mobile networks.

2.3.9 Blacklisting/Whitelisting of Sources

The approach maintains a list of entities that would be granted or denied the access to a particular resource. This approach is practiced in mobile networks where an Equipment Identity Register (EIR) is used to grant or deny the access to a user. The EIR register consists of three databases: 1) “white list” that contains all the legitimate mobile stations, 2) the “black list” contains International Mobile Equipment Identity (IMEI) of the stolen or barred mobile stations and 3) the “grey list” maintains the list of mobile station which are to be traced [29]. A similar concept is also applicable in network security where firewalls, IDS or AAA servers maintain the lists of whitelisted and blacklisted sources, to decide the access denial or access grant upon receiving a request from a remote source [23].

3. Elements of Internet Technology

The chapter establishes some of the basic concepts of the Internet. The chapter starts by introducing the concept of Domain Name System (DNS) and describes various structural components of DNS. Next, an overview of Network Address Translation (NAT) is presented and the issues with NAT reachability are highlighted.

3.1 Domain Name System

The section introduces the concept of Domain Name System (DNS). First, an overview of DNS is presented and then DNS infrastructure and protocol are explained, followed by a detailed description of the name resolution process.

3.1.1 Overview

The Domain Name System (DNS) is a directory lookup service that provides the mapping between (human readable) names of the hosts and their corresponding Internet address e.g. the DNS service is used by both the end users and the Internet services to locate the remote end of the communication and its deployment in the Internet is motivated by human unease to remember Internet addresses i.e. numerals over the string literals (domain names).

DNS has a distributed and hierarchical architecture of interconnected name servers. It defines a client-server protocol to extract the requested information from a name server. The three major components that comprise DNS are Resolver, Name Server and the Domain Name Space [30].

3.1.2 Domain Name Space

DNS uses a hierarchical name space to ensure that each address maps to a unique host name. The hierarchical name space or the Domain Name Space has an inverted tree like structure with root at top. The root is extended by subdomains which may contain several subdomains as well, and each node in this name space has a label (a string of maximum 63 characters) and a domain name. A Fully Qualified Domain Name (FQDN) which uniquely identifies an endpoint is defined as a sequence of labels from the last node up to the root node, separated by dots [18].

3.1.3 Resolver

As mentioned before, DNS is designed as a client/server application. A resolver program performs the client side of DNS operations on behalf of an end user. The resolver queries a name server for a certain record type as per user request, then receives the response and interprets it either as a valid response or an error response. The response is finally delivered to the requesting entity. In some cases, a DNS response may refer the resolver to another name server to perform the DNS query.

3.1.4 Name Servers

A DNS Name Server performs the server side of operations in DNS. Similar to the name space hierarchy, the name servers are also interconnected in a hierarchical manner and each name server has an authority over a certain portion in the domain name space and this area of authority is referred to as zone.

DNS defines two types of name servers: primary and secondary name servers. A primary (master) name server is responsible for creating, maintaining and updating the zone information in the zone file, located on its local disk. Whereas, a secondary (slave) name server, also an authoritative name server for the zone, is deployed to implement the redundancy in DNS by copying the same zone information from the primary name server to multiple secondary name servers.

3.1.5 DNS Message Structure

DNS has two types of messages: query and response. The query and response message share the same header format with some fields being absent in the query message. The query message contains a fixed header of 12 bytes and question records only, whereas the response message can contain answer records, authoritative records and additional records in addition to the fixed 12 byte header and the corresponding question record.

Figure 3.1 presents the DNS message format, where the identification field in the fixed header format is generated when a client generates a DNS query, and it is used to match the DNS response with the corresponding DNS query. Various flags under the “Flag” field define the nature of a DNS message i.e. message type (query or answer), type of resolution requested

(recursive or iterative) etc. The next four fields in the message structure identify the total number of forthcoming question, answer, authority and additional resource records in the DNS message.

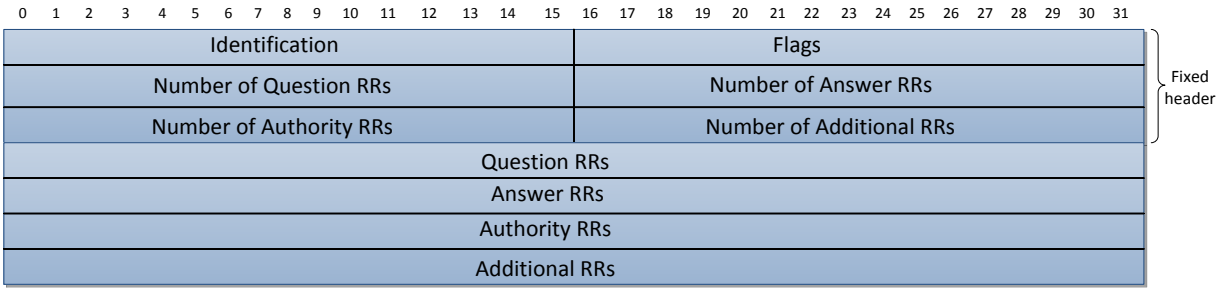


Figure 3.1 DNS message structure

The next four sections in the message header are encoded in Resource Record (RR) format. The question section carries one or more queries sent to the DNS server. The answer section consists of one or more RRs sent in response to the earlier query. The authority section carries one or more RRs that inform about the authoritative name servers for the queried resource. The additional section may contain multiple RRs providing additional information to aid the resolver in the resolution process [18].

3.1.6 Resource Record

The information related to a domain or a zone is held in Resource Records (RRs). Each RR consists of a tuple of information which includes: name, type, class, time to live and the resource data. Based on the name and type parameter in the query, an RR can carry the resource data from the domain. The TTL field value indicates the duration for which an RR is valid when cached, after which a DNS query must be directed towards the authoritative name server. While a long list of RFCs have introduced different resource record types, we only present a brief description of the most commonly used RR types in Table 3-1.

Type	Description
A	Maps the hostname to the corresponding IPv4 address of the host
AAAA	Maps the hostname to the corresponding IPv6 address of the host
PTR	Used for reverse DNS lookup i.e. Mapping an IPv4 or IPv6 address to a hostname
NS	Indicates the authoritative name server for a zone
NAPTR	Allows the usage of regular expressions to generate domain names i.e. as URI
CNAME	The canonical name record contains an alias of a name
MX	Specifies the mail exchange server accepting the emails on behalf of the queried domain
TSIG	Used for authenticating updates/responses coming from an approved client or DNS server

Table 3-1 NS Resource records

3.1.7 Name Resolution

The process of contacting a name server to retrieve the resource data of a particular domain is called name resolution. Every DNS query begins with a resolver located in the user host system. The resolver is configured to know the name and address of a local name server. If the resolver does not have the queried resource in its cache, it forwards the query to the local name server which may return an answer or further query the domain name space to resolve the DNS query.

The resolution process follows the inverted tree like structure of the domain name space. It starts by contacting the top level domain and then moves one step down based on referrals from each level name server until the given resource record is located. The name resolution process follows either of the approaches: recursive name resolution or iterative name resolution.

3.1.8 Recursive Resolution

In recursive name resolution, the resolver offloads the name resolution process to the queried name server. The name server either responds with the requested resource or contacts with the root name server when the record is missing in its cache. The root name server returns the referral to the next level name server. The local or querying name server follows the referral and forwards the same query to the next referred name server, which may return a referral to another next level name server. The procedure is followed until the authoritative name server of the domain is reached, which responds with the requested resource. This response is then forwarded to the resolver by the queried name server.

Figure 3.2 presents a scenario where recursive name resolution is used to resolve the query for the domain name ‘www.aalto.fi’, sent by the host. Local DNS server contacts the Root-DNS that refers the query one step below in the domain name space to the ‘.fi’ name server. Similar to the previous step, the procedure is repeated between the local name server and the referrals until the authoritative name server ‘aalto.fi’ for the queried domain is reached. The response from the authoritative name server is then forwarded by the Local DNS server to the requesting Host [18].

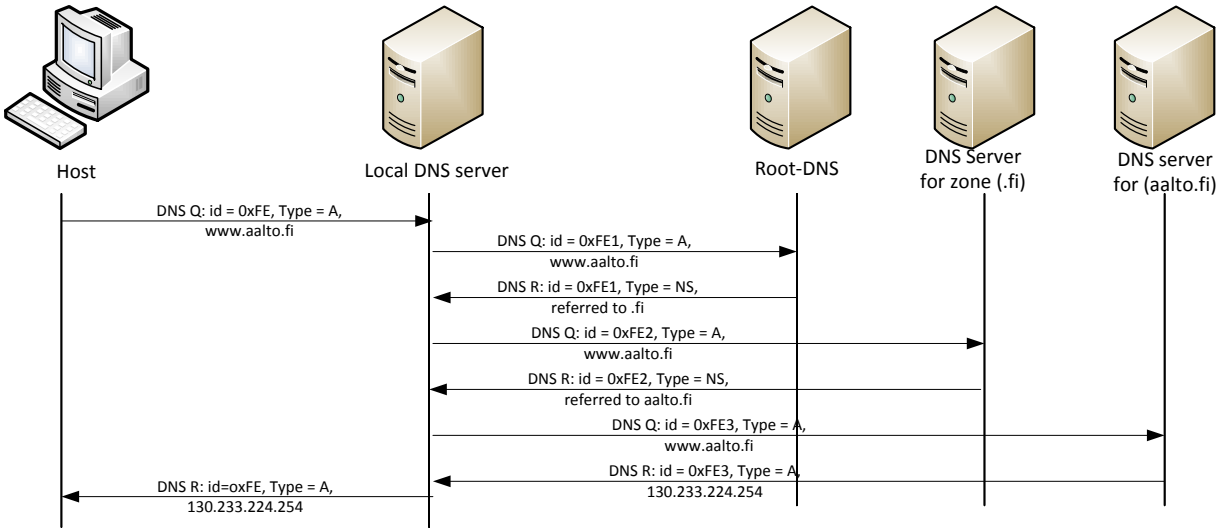


Figure 3.2 Recursive name resolution of the DNS query

3.1.9 Iterative Resolution

In iterative name resolution, the local name server does not contact the root name server when it does not host the requested resource, rather the closest name server in the name space hierarchy is approached with the DNS query. The name server either responds with the resource record or returns a referral to the name server that may resolve the query. The requesting name server then forwards the DNS query to the new name server, and repeats this for each referral received from the earlier contacted name server until the authoritative name server is reached. Eventually, the response from the authoritative name server is forwarded to the DNS resolver [18].

3.2 Network Address Translation

The section briefly describes the motivation behind the development of Network Address Translation (NAT) in the Internet, followed by its operations and different NAT mapping behaviors. Next, the NAT traversal issue is highlighted and an analysis of NAT traversal schemes is presented.

3.2.1 Motivation

The growth and popularity of the Internet drew more users to the Internet, and this increasing number resulted in rapid depletion of the IP address space. With shortage of the available addresses, a solution was proposed to prevent the IP address space exhaustion called Network Address Translation (NAT) [31]. NAT enables a private network to use private addresses for internal communication and a set of public (or global) addresses for communication with hosts in the public Internet [18].

Besides alleviating the shortage of addresses, NAT hides the private network from the rest of the Internet. This is because a host in the public Internet cannot unilaterally address a host in the private network. From a network administrator perspective, it is beneficial as it makes difficult for an attacker to intrude the host. However if inbound connections have to be allowed from the Internet to the private network, the same thing is perceived as the reachability problem created by NAT.

3.2.2 Operations

NAT defines the usage of the private addresses inside a private network. These private addresses reserved by IANA can be reused inside any private network, and therefore are not unique globally. While a communication between hosts in the same network uses private addresses, the communication with a host in the public network requires translation of the private address to one of the public addresses used by the NAT.

In the NAT architecture, a NAT device is installed at a network border of the private network and it contains a translation table, which maintains a tuple of information: private IP address, private port, public IP address, public port and the protocol used. When an internal host initiates an outgoing connection, the NAT replaces the private IP address and the port of the host with

one of the public IP addresses and port number in the packet, and stores the corresponding tuple in the translation table. This tuple of information is referred as “NAT binding” or “mapping” in the NAT. This mapping is used to perform address translation for forwarding packets between the private host and the remote host. This address translation by NAT is necessary, as no router in the public Internet would forward a packet containing a private address [18].

3.2.3 NAT Address Assignment

The NAT devices can be divided in two categories: the basic NAT and the Network Address Port Translation (NAPT). The basic NAT only performs IP based translation services while the NAPT performs IP address and port dependent translation services. A NAT device can define the mapping behavior in the NAT using any of the following approaches, defined in RFC 4787.

Endpoint-Independent Mapping

The NAT reuses the same mapping for all the connections originating from the same private IP address and port number destined to any external IP address and port number.

Address dependent mapping

Here, the NAT reuses the same mapping for all the connections originating from the same private IP address and port number to the same external IP address, regardless of the external port number.

Address and Port Dependent Mapping

In this mapping scheme, the NAT reuses the same mapping as long as a connection packet originates from the same private IP address and port to the same public IP address and public port. Or, a new mapping is created in the NAT every time either of the private IP address, the private port, the public IP address or the public port changes [32].

3.2.4 NAT Reachability Problem

When a private host behind the NAT initiates a communication with an external host, the NAT creates a mapping in the translation table and forwards the packet to the destination. The response from the public host is received by the NAT, which is forwarded to the private host after applying the mapping in the NAT.

However, for an inbound connection, initiated by an external host, the incoming packet would be dropped by the NAT because of the absence of a prior mapping in the NAT. This is known as NAT traversal issue or the reachability problem, where an external host cannot reach the host behind the NAT.

The reachability problem introduced by NAT affects various protocols and applications in the Internet. For example, the protocols that carry IP addresses in their payload for connection establishment, i.e. SIP, are affected by NAT. This is because NAT does not operate above layer 4 and hence cannot modify the content in the protocol payload. Similarly, the peer-to-peer applications, unlike client-server applications, require bidirectional connectivity and are affected by the NAT reachability issue. The reachability problem also hinders hosting a service inside the private network, as all the packets from its clients in the public Internet will be dropped for not having a prior mapping in the NAT.

3.2.5 NAT Traversal Protocols

With the introduction of the NAT reachability problem, several proposals were aimed at solving the NAT traversal issue. These solutions or techniques are referred as NAT traversal protocols. A brief summary of some of the well-known NAT traversal protocols is presented below,

Session Traversal Utilities for NAT (STUN) is a client-server protocol that acts as a tool for other NAT traversal techniques. The protocol is used by an end host to learn the NAT binding associated with the host, and it requires a STUN server deployment in the public network. STUN by itself is not a NAT traversal solution and it is used to check the connectivity between two end points or to keep the bindings in the NAT alive [1].

Traversal Using Relay around NAT (TURN) proposes a relay based architecture to complement the limitations of the STUN. It enables two end points located behind different NAT devices to communicate with each other by relaying their information through a TURN server [2].

Interactive Connectivity Establishment (ICE) utilizes the capabilities of both STUN and TURN protocol. It provides each communicating peer with enough information about their topologies and presents them with different potential communication paths using STUN and TURN techniques [3]. ICE is known for successfully establishing a connection even under very challenging network conditions.

Despite solving the NAT reachability problem, the afore-mentioned schemes come with certain drawbacks. The STUN/TURN/ICE schemes require keep-alive signaling to prevent the NAT binding from expiring. The keep-alive signaling solves the NAT traversal issue, but puts some serious constraints on mobile usage and may deplete the mobile battery quickly, because of frequent signaling requirement. Moreover, the NAT traversal schemes require client code integration in the respective applications. The TURN and especially the ICE approach significantly increase the connection setup delay between the communicating hosts. The delay in ICE happens because it requires that the client should send up-to 7 mapping messages for a single candidate address using STUN and it waits until the timeout expires before giving up on the candidate address. ICE takes up-to 100 messages before picking an optimal connection path out of multiple candidate paths, after having gathered topological information using multiple techniques.

4. Customer Edge Switching

This chapter presents an overview of Customer Edge Switches (CES). The chapter first highlights the motivation behind customer edge switching and its main features. Next, a brief overview of the CES architecture is presented, followed by the protocol used for communication and a detailed description of packet forwarding in CES.

4.1 Motivation

As described in section 3.2.5, NAT deployment in the Internet has introduced the reachability problem. The reachability problem prevents a private host from being globally reachable and accepting connections from the hosts in the public network, as a public host is unable to address the destination univocally. While many NAT traversal proposals i.e. STUN, TURN and ICE etc. have attempted to solve the reachability problem, none of these solution is perfect for mobile devices. A long connection setup time or mandatory keep-alive signaling are general limitations of these schemes. The keep-alive signaling requirement makes NAT traversal protocol highly unsuitable for mobile devices, as it may drain mobile battery quickly [33]. Realizing the need for an efficient solution, a research was carried at the COMNET department of Aalto University led by Raimo Kantola and a prototype was implemented by Lauri Virtanen called "Customer Edge Switching". The prototype was extended by Jesus Llorente and Maryam Pahlevan in their MSc. Thesis, supervised by Raimo Kantola. The proposed CES architecture not only solves the reachability issue, but also improves the security in the Internet by making 'Trust' as a cornerstone of the design. This guides Internet from end-to-end model towards a trust-to-trust model, advocated by David Clark [34].

CES is a proposed replacement of NAT devices, which aims to solve the problems introduced by NAT without incurring any change in the end hosts or the protocols used in the Internet. The CES solution uses regular network capabilities to provide the end-to-end connectivity. It uses globally unique domain names for end host identification and then uses private/public address pools for addressing an end-host. The solution does not require any keep-alive signaling, which makes it suitable for mobile environments. CES also supports interworking with legacy networks through a component called Private Realm Gateway (PRGW).

4.2 Architecture

CES deployment proposes a split architecture, where network is divided into Customer Network (CN) and Service Provider Network (SPN). This separation of CN from SPN provides the benefits of isolation, independent deployment of technologies and clear definition of trust boundaries in the Internet. From the provider's perspective, this results in improved performance and services, as it can facilitate the deployment of new protocols and technologies in the SPN.

Figure 4.1 presents an overview of the CES architecture where end users are connected to the customer network. These end hosts are accessible to a remote host through their globally unique domain names. Next, the CES uses a non-unique and a reusable address called "proxy address", from a pool of available addresses, to represent the remote host in the local host's network technology i.e. IPv4 or IPv6. The CES device at the network edge also maintains connection state information in the translation table, which enables forwarding of data packets between trust domains when moving from the source to the destination, and vice versa.

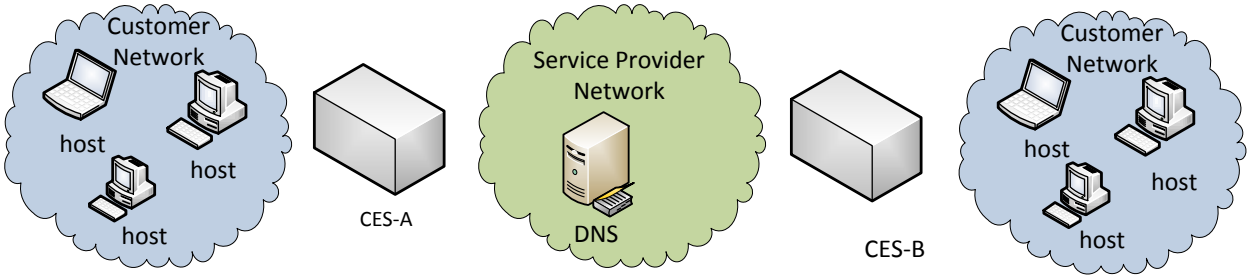


Figure 4.1 CES Architecture

The CES architecture proposes a Global Trust Operator (GTO) to rate different trust domains based on their trust services i.e. spam volume or attack traffic influx/outflux of the CES. This rating can be reflected in the admission policies used by customer networks and also in commercial aspects of the Internet i.e. interconnection prices of the ISP. The proposed techno-economic model aims to make "trust" an essence of future Internet design [6]. Such a global trust management can be used to bring down huge volume of spam and unwanted traffic in the current Internet [35].

4.3 Packet Forwarding Across CES

Packet forwarding in CES can be divided into three categories. First is the packet forwarding in an Intra-CES communication i.e. when both communicating hosts are behind the same CES. Second is packet forwarding in an inter-CES communication, when the communicating hosts are behind different CES devices. Third category involves the packet forwarding in PRGW, which deals with the Internet traffic from legacy IP networks. A brief overview of the second and the third category of packet forwarding is presented in the following sections, whereas intra-CES communication is not discussed for being out of the thesis scope.

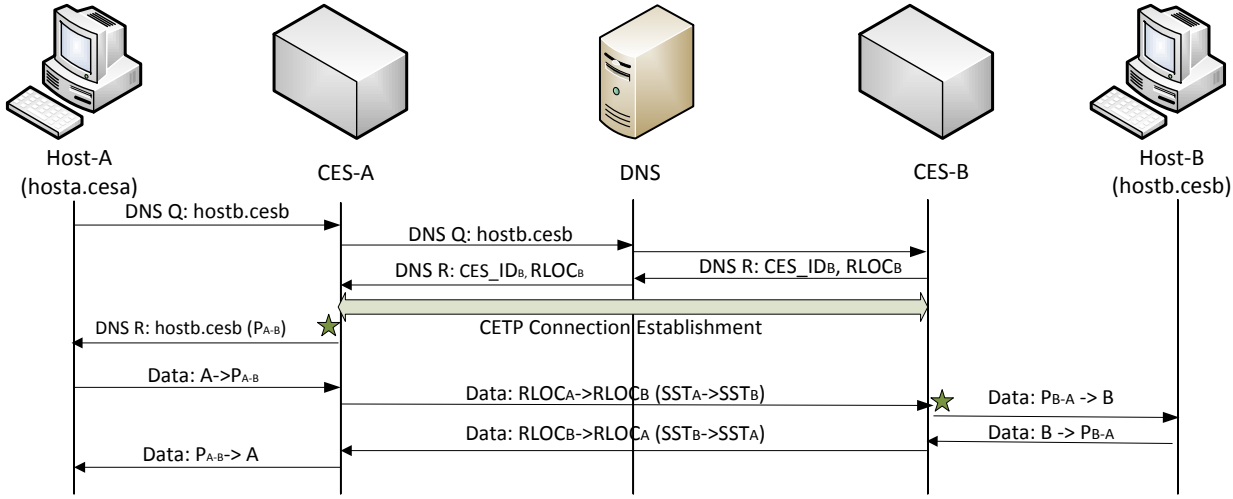
4.3.1 Packet Forwarding in Inter-CES Communication

Inter-CES communication refers to the case when the communicating hosts are behind different CES devices. The CES solution heavily relies on the Domain Name System for successful connection establishment between the hosts. The principle employed is that the source performs DNS resolution in order to discover the CES-ID that hosts the destination domain, after which the sender and the destination CES carry out a connection establishment procedure in accordance with the host admission policies. A successful connection establishment creates a connection state in each CES device, where the CES represents the remote host locally using a proxy address. Following this, the DNS response carrying the proxy address of the destination is forwarded to the source, and both hosts exchange the data packets using the states created in the CES devices.

Figure 4.2 depicts the scenario where Host-A behind CES-A tries to communicate with Host-B that resides behind CES-B. According to the principle, Host-A issues a DNS query for the Host-B. Since, Host-B does not reside in the network of the CES-A, so the CES-A forwards the DNS query to the DNS server, which in turn sends it to the DNS server located in CES-B, based on its NS resource record. The DNS response from CES-B conveys the routing locator (RLOC) and the CES-ID information corresponding to the destination host.

Based on the DNS response, CES-A sends the connection request to CES-B for subsequent data transfer between the source and the destination hosts. The connection establishment process between the outbound CES (oCES) and the inbound CES (iCES) is a policy controlled affair and is described in section 5.1. Upon successful connection establishment, CES creates a state in the connection table and reserves a proxy-address to represent the remote host locally. The state

stored in the connection table, among others, includes: Source Session Tag, Destination Session Tag, RLOCs, local address, proxy address etc.



A: Private IP of Host-A	CES_ID_A: CES Id of Host-A
B: Private IP of Host-B	CES_ID_B: CES Id of Host-B
$RLOC_A$: Routing Locator of CES-A	P_{A-B}: Proxy-address representing Host-B to Host-A
$RLOC_B$: Routing Locator of CES-B	P_{B-A}: Proxy-address representing Host-A to Host-B
hosta.cesa: Domain Name of Host- A	SST_A: Session Tag for session initiated in CES-A
hostb.cesb: Domain Name of Host- B	SST_B: Session Tag for session initiated in CES-B

Figure 4.2 Packet flow in CES-to-CES communication

CES-A modifies the received DNS response to carry a proxy-address, to represent the destination host locally, and forwards this response to Host-A. Next, Host-A sends the intended data packets towards the proxy-address received in the DNS response, assuming this as the address of the destination host. Upon receiving a data packet at this proxy-address, CES-A processes the packet according to the state stored in the connection table and forwards the packet to CES-B. The data packet will be forwarded to Host-B after undergoing changes according to the state information stored in CES-B. A similar processing is carried for the data packet traversing in the reverse direction, towards Host-A.

4.3.2 Packet Forwarding in PRGW

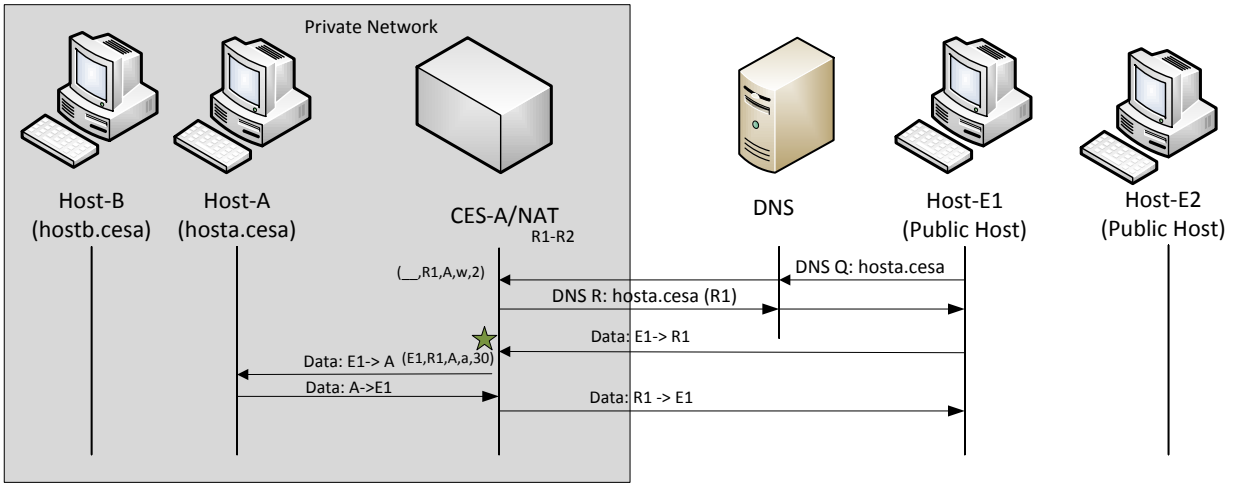
CES has a component called Private Realm Gateway (PRGW), which provides the backward compatibility when dealing with legacy IP sources. Jesus Llorente carried out his Master thesis on interworking with legacy networks, in Customer Edge Switching, and implemented the

concept of Private Realm Gateway (PRGW). While a detailed architecture and salient features of this implementation are available at [4], this section explains packet forwarding in PRGW.

PRGW also supports DNS functionality, and it acts as an authoritative name server for the domains hosted in the private network. The DNS support enables PRGW to resolve a DNS request received from a public host, for domains hosted in the network. Similarly, the PRGW acts as a DNS resolver for hosts in the private network and it performs DNS look-up for the destination hosts in the public Internet.

An outbound connection from a private host to a destination in the public Internet is handled in a similar manner to NAT. When an outbound packet is received, the PRGW creates a state in the connection table and performs outbound address translation on the packet. Similarly, upon the reception of response from the public host, the PRGW looks up for the corresponding connection state and forwards the response to the private host after performing inbound address translation. It is noteworthy that a private host does not necessarily need to perform name resolution for the destination, when establishing an outbound connection in PRGW.

But, for an incoming connection from a legacy source, PRGW is heavily dependent on DNS and the functionality of the Circular Pool of public IP addresses. For a legacy source to access a domain behind the CES, it needs to send the DNS query for the destination domain. Assuming that CES is also the authoritative name server for the domain, an address is reserved from the circular pool and the DNS response containing the reserved address is returned to the source of the DNS query. The returned address is reserved in a state addressed to an unknown sender in the connection table for subsequent data flow from the source. A subsequent data packet from a source whose destination address is the same as the destination address of the reserved connection state and for which there is no already ongoing connection is believed to be the source of the DNS query, and the data packet from this source is forwarded to the destination domain, behind the CES/PRGW.



A: Private IP of Host-A	R1-R2: Circular Pool addresses
E1: Public IP of Host-E1	E2: Public IP of Host-E2
hosta.cesa: Domain Name of Host- A	
Connection state: (IP source, Public IP in CES, internal host, status, timeout)	

Figure 4.3 Packet flow in PRGW for an inbound connection

The detailed operations of PRGW for an inbound connection is described using Figure 4.3, where Host-E1 sends a DNS query for the domain of Host-A, which is forwarded to CES-A. For the simplicity of explanation, we assume that CES also supports DNS functions for the private network. The CES reserves the next available address from the circular pool and returns the DNS response carrying the address reserved i.e. 'R1', to the source of the DNS query. A state with 'waiting' status, addressed to an unknown sender and the destination Host-A, is created in the connection table. A state in the CPOOL, among others, includes the source address, the allocated address, private address of the destination, status (waiting or active) and a timeout value.

Upon reception of the DNS response, Host-E1 believes that address R1 is the IP address of Host-A and it sends data packets addressed to R1. Since, the received packet at CES does not match an ongoing connection, but the destination address of the data packet matches with the destination address of a 'waiting' state, so the PRGW admits the data packet as a legitimate flow and changes the status of connection state from 'waiting' to 'active'. The data packet is forwarded to Host-A after performing public-to-private address translation. Similarly, the response from Host-A is sent to the legacy source after performing private-to-public address translation at PRGW. A detailed description of packet forwarding in PRGW is described in [4].

4.4 Customer Edge Traversal Protocol

Customer Edge Switching defines a tunneling protocol for communication between CES devices, called Customer Edge Traversal Protocol (CETP). Since, the scope of CETP protocol is limited between CES devices, so hosts or applications in a Customer Network (CN) do not need to be aware of CETP.

The CETP protocol has evolved many times since it was first developed by Pahlevan [5], and has experienced several changes based on the experiences acquired. Even now, CETP is a work under progress and may be subjected to changes in the future. While some of the protocol details are common with earlier versions of the protocol, we present an overview of the current CETP packet structure in this section, as a detailed explanation of CETP format is out of scope of this thesis. The details of the previous version of the CETP protocol can be found at [36].

CETP can be divided in two parts: Control plane and Data plane. The control plane carries the signaling information exchanged between two CES devices and the data plane carries the data packets received from the hosts behind the CES. For example, in Figure 4.2 CES-A receives a data packet from Host-A and tunnels it from CES-A to CES-B using the data plane of CETP. Whereas, the control plane of CETP carries signaling information i.e. IDs, RLOCs, Signature, payload encapsulation type etc. necessary for connection establishment in CES.

Figure 4.4 presents the packet structure of the control plane of CETP, where all control information elements are aligned with a 32-bit boundary. The structure consists of a fixed header, source and destination session tags and a set of control TLVs. The initial 4 bytes define the fixed header part, where Version field identifies the CETP protocol version, C and P flags indicate the presence of Control TLVs and Payload TLV in the CETP packet. Header length field of 11 bits informs of the CETP header size, and is calculated as a sum of fixed header (4 bytes) + length of Source and Destination session tags + length of all control TLVs. Reserved field of 8 bits is left for the future extensions of the protocol. SSTLen and DSTLen are 4 bit fields each, and their value indicates the length of Source Session Tag (SST) and Destination Session Tag (DST), respectively. The length is computed as two bytes multiple of the value contained in SSTLen and DSTLen. Even though values in SSTLen and DSTLen can range from 0 to 15, the current version only supports 0, 2, 4 and 8 byte length session tags.

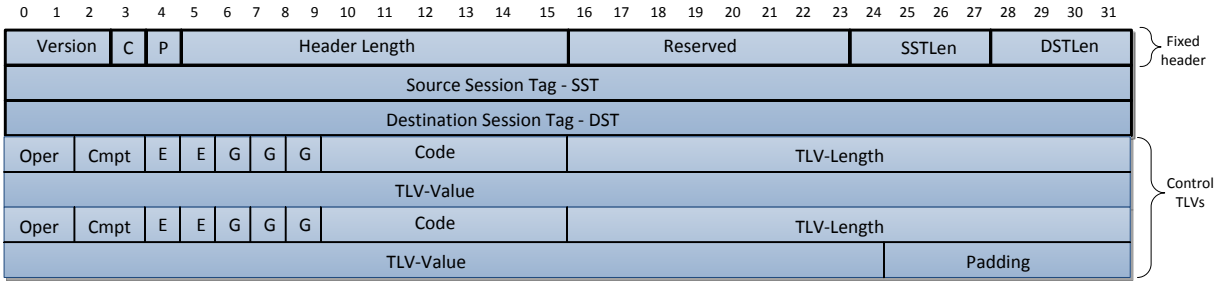


Figure 4.4 CETP Control plane structure

Next, the CETP header contains Source and Destination session tags that are used to uniquely identify a session, and consequently identify the binding between the private and public addresses used for communication, between two end points in the CES devices. Control TLVs in the CETP header follow a Type-Length-Value format and they are used to carry various flow related signaling information between CES devices. The control TLVs exchanged between CES devices are a policy controlled affair, and they are used for packet admission control, receiver policy enforcement as well as for connection establishment between CES devices. Padding is added to control TLVs to keep up with the CETP's requirement of 32-bit boundary.

A Control TLV contains a 3-bit 'group' field that defines the general type of the control TLV, whereas the 'code' field defines the specific level of control TLV within each group. A list of different control TLVs and their encodings used in the current CETP format can be found at [5], [36]. The 2-bit 'operation' field tells the remote end if the TLV carries a query, response or information operation. The two flags marked with 'E' are reserved for future extensions. The two bits of compatibility field indicate if a TLV is compulsory for connection establishment when used with the 'query' TLV operation OR to indicate unavailability of a TLV using 'Notavailable' option within a 'response' TLV.

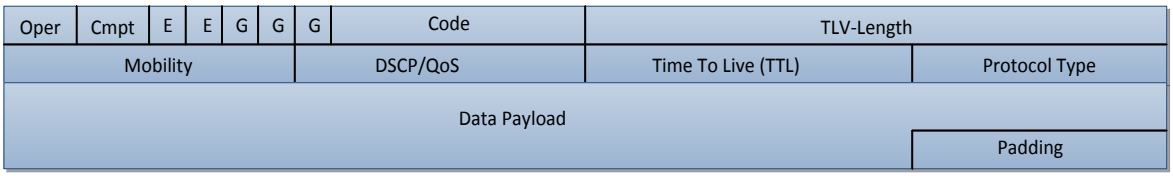


Figure 4.5 CETP payload TLV (for IPv4/IPv6 Encapsulation)

The data plane of CETP is represented by a payload TLV, and it carries the actual data received from the host. The host data undergoes either of IPv4, IPv6, Ethernet or Carrier Grade Ethernet based payload encapsulation. The choice of payload encapsulation is agreed upon during the

negotiations performed for connection establishment, using Control TLVs. The packet structure of the CETP payload TLV is presented in Figure 4.5 for IPv4/IPV6 encapsulation, where the first 4 bytes of payload TLV are the same as in a control TLV.

The Mobility field in the payload TLV provides support for mobile environments, whereas the DSCP/QoS field is used for transporting the traffic with special requirements. The TTL field in CETP serves the same purpose as the "TTL" field in the IPv4 protocol or as the "Hop Limit" field in the IPv6 protocol, i.e. to prevent the CETP packet from running forever in the Internet. Hence, the TTL field is used to calculate the TTL value of the IP packet that the remote CES sends to the endpoint of communication in the CES enabled customer network. The 8-bit "Protocol Type" field in the payload TLV indicates the upper layer protocol carried in the CETP payload i.e. ICMP, UDP or TCP etc.

Figure 4.6 presents the format for an Ethernet encapsulated payload TLV, which differs from Figure 4.5 in last two bytes before the Data Payload. The two byte 'Ethertype' field indicates the upper layer protocol carried in the CETP payload i.e. IPv4 or IPv6 protocol.

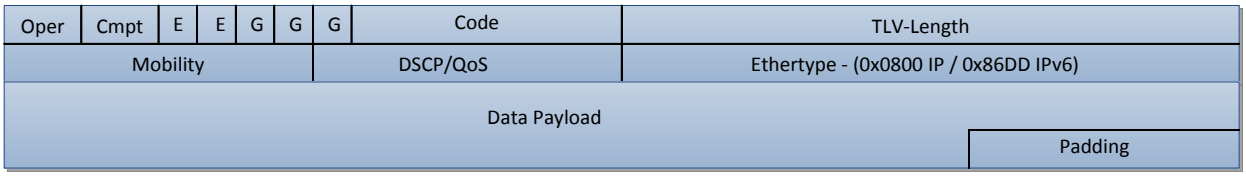


Figure 4.6 CETP payload TLV (for Ethernet encapsulation)

5. Security Vulnerabilities in Customer Edge Switching

The chapter describes security vulnerabilities present in the CES architecture. The vulnerabilities are divided into two categories: Circular Pool vulnerabilities and CES-to-CES communication vulnerabilities. The chapter presents these vulnerabilities in a structured way to facilitate application of security models to secure Customer Edge Switches.

5.1 CETP security vulnerabilities

This section aims at presenting the security vulnerabilities in the CES-to-CES communication model. However, the section begins by explaining the CETP connection establishment process in CES-to-CES communication.

5.1.1 CETP Connection Establishment

While the packet forwarding in a CES-to-CES communication is explained in section 4.3.1, we only describe the connection establishment part of inter-CES communication here. In an inter-CES communication, a connection can be established in either of the ways described below,

CETP Connection Establishment in 1 RTT

CETP connection establishment in one round trip time (1RTT) is explained using Figure 5.1 and the policies listed below. A detailed description of these policy elements is presented in [36]. However, control.cesid is a newly introduced policy element that identifies the CES node hosting the sender, carried in ‘Id.fqdn’ policy element.

***Outbound policy of Host-A*:**
Role: Outbound
Required: Id.fqdn, rloc.ipv4, payload.ipv4
Offer: Id.fqdn, rloc.ipv4, payload.ipv4, control.cesid
Available: Id.fqdn, rloc.ipv4, payload.ipv4, control.cesid

***Inbound policy of Host-B*:**
Role: Inbound
Required: Id.fqdn, rloc.ipv4, payload.ipv4
Available: Id.fqdn, rloc.ipv4, payload.ipv4, control.cesid, control.headersignature

Upon the reception of the DNS response at oCES, the oCES encodes a CETP packet with Host-A policy and sends it to the iCES (inbound CES), identified from the DNS response. The packet received by the iCES contains the query TLVs for the receiver host-ID, RLOC and Payload type alongside the sender's offer of the host-ID, RLOC, Payload and CESID. The received packet identifies the sending and the receiving ends of communication using 'ID' and 'Destep' TLVs. The 'Destep' TLV in the received packet identifies the destination host behind the iCES. The communication session between CES devices is uniquely identified using SST and DST values. The SST value is set to a locally selected number e.g. '33000' by the oCES, whereas the DST value is set to 0 in the outgoing packet.

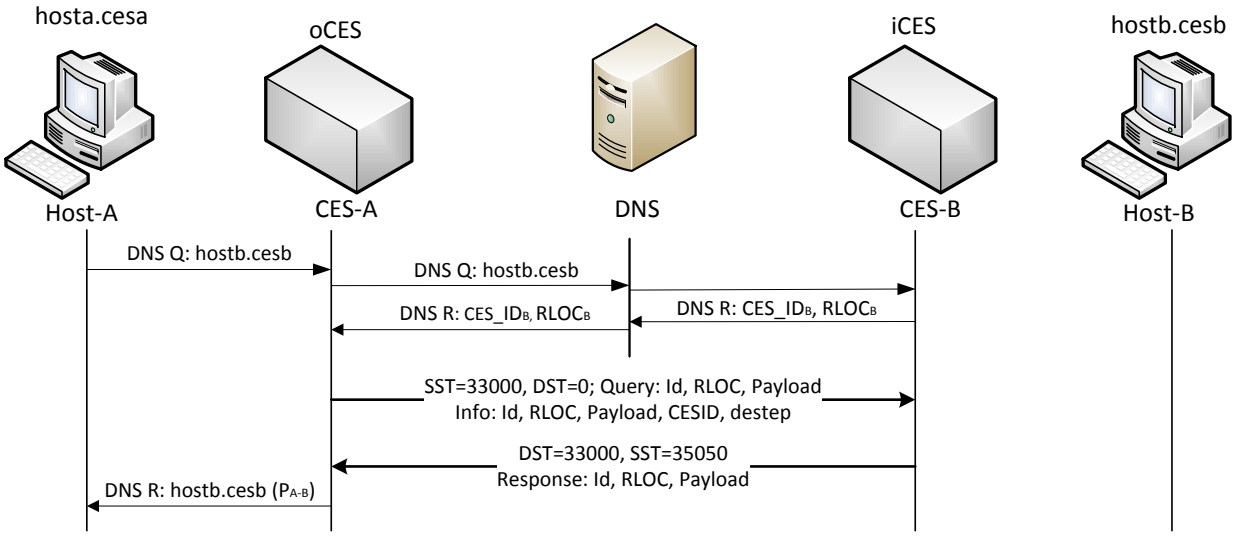


Figure 5.1 CETP Connection Establishment in 1 RTT

Since the "Info" TLVs in the received packet fulfill the policy requirements of Host-B, the next packet from the iCES carries the response TLVs for the queries received in this packet. The response packet bears the DST value of 33000, which is the same as SST of the received packet, and it also bears an SST value of '35050', which is a locally assigned value by the iCES upon the successful connection establishment in the iCES. The iCES marks a connection establishment "successful" when a received packet fulfills the policy requirements of the destination and the iCES can successfully respond to the sender's policy requirements carried in the CETP packet.

The response packet is then received by the oCES that looks up for a connection state whose SST value is the same as the DST value received in the packet. For a matching state, the oCES believes that the packet is a response to the connection request sent earlier and it records the SST value. Next, if the response packet carries the reply for all queried TLVs, the oCES considers the connection establishment as "successful". Hence, the connection is established in 1 RTT.

CETP Connection Establishment in 2 RTT

A CETP connection can also establish in two round trip times (2RTT), as described using Figure 5.2 and the policies listed below

Outbound policy of Host-A:

Role: Outbound
Required: Id.fqdn, rloc.ipv4, payload.ipv4
Offer: Id.fqdn, rloc.ipv4, payload.ipv4
Available: Id.fqdn, rloc.ipv4, payload.ipv4, control.cesid, control.headersignature

Inbound policy of Host-B:

Role: Inbound
Required: Id.fqdn, rloc.ipv4, payload.ipv4, control.cesid
Offer:
Available: Id.fqdn, rloc.ipv4, payload.ipv4, control.cesid, control.headersignature

Upon the reception of DNS response at oCES, the oCES encodes a CETP packet with Host-A policy and sends it to the iCES, identified from the DNS response. At the iCES, unlike the 1RTT case, the offered TLVs in the received packet fail to fulfill the policy requirements of the destination Host-B, i.e. as 'control.cesid' policy element is not carried in the inbound packet. So, the iCES responds with Full Query (FQ) message to the oCES, and informs the oCES of all policy elements required to successfully establish a connection with this destination. The FQ message contains the DST=33000, same as the received SST, and sets the SST as zero. The SST=0 value is assigned by the iCES as the connection is not yet established at the iCES.

Upon receiving the CETP packet, the oCES looks up for a connection state whose SST value matches the DST value received in the packet. For a matching state, the oCES accepts the incoming packet as a response to the connection establishment request sent earlier. The SST value of '0' in the received packet and the presence of query TLVs inform the oCES about policy mismatch at the iCES. The oCES then re-encodes a CETP packet in the light of the received policy requirements, only if these requirements are supported by the "Available" policy vector of Host-A. The newly encoded CETP packet bears the same SST and DST values as for the first packet sent to the iCES.

Upon receiving this CETP packet at the iCES, since the policy requirements of Host-B are fulfilled by the "info" TLVs in the packet and the query TLVs can be answered from Host-B policy, the iCES declares the connection establishment as "successful". Following this, the iCES assigns a session tag to this communication. A CETP response packet carrying the response to all

queried TLVs is encoded and sent in the direction of the oCES. The packet carries a locally generated SST value of '35050', and the DST is set to the SST value received in the packet.

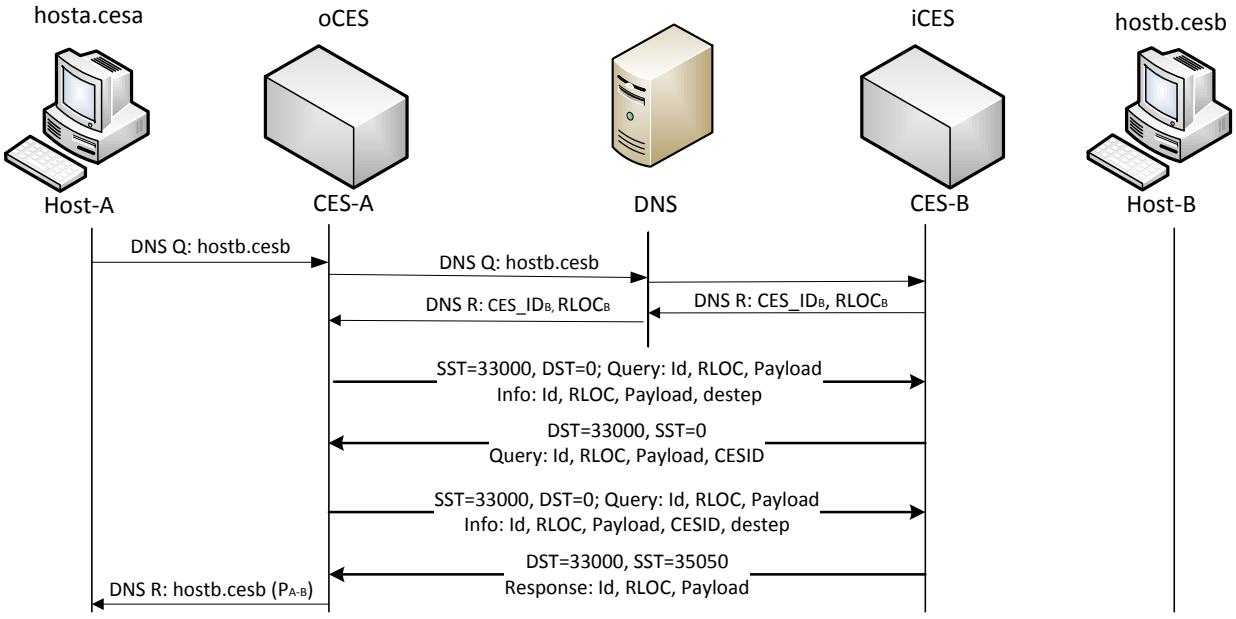


Figure 5.2 CETP Connection Establishment in 2RTT

The CETP response is received at the oCES, which performs a lookup operation for a connection state whose SST value is the same as the DST value received. For a matching state, the oCES records the corresponding SST value and if the received packet carries the response to all required policy elements of the sender, the connection establishment is declared as successful. Hence, the connection is established in 2 RTTs. The SST and DST values learned during the connection establishment phase are used to forward subsequent data packets between the CES devices.

5.1.2 CETP Attacks

CETP Attacks can be categorized into Legacy Host attacks and CES-based attacks. Legacy Host attacks refer to the attacks from legacy IP hosts, as they share the same VPN with CES devices. This situation is possible given the fact that the IPv4 address space is almost all in use and the CES RLOCs can belong to any address block. This raises a situation where legacy hosts, residing in the same VPN as CES, can generate CETP attack traffic after a Trojan has successfully installed the CETP attack module on them.

Whereas, the term CES-based attack refer to an attack possible on a CES device, when CES nodes are deployed in a separate VPN than legacy hosts. In such a VPN, a legacy host cannot send forged CETP packets towards a CES, to launch an attack. The attack scenarios presented for a Host-CES shared VPN can also be launched in a CES only VPN, if an attacker configures an attacking host in the same VPN as CES devices or if a legitimate CES has been taken over by a bot to launch spoofing attacks (or DoS attacks).

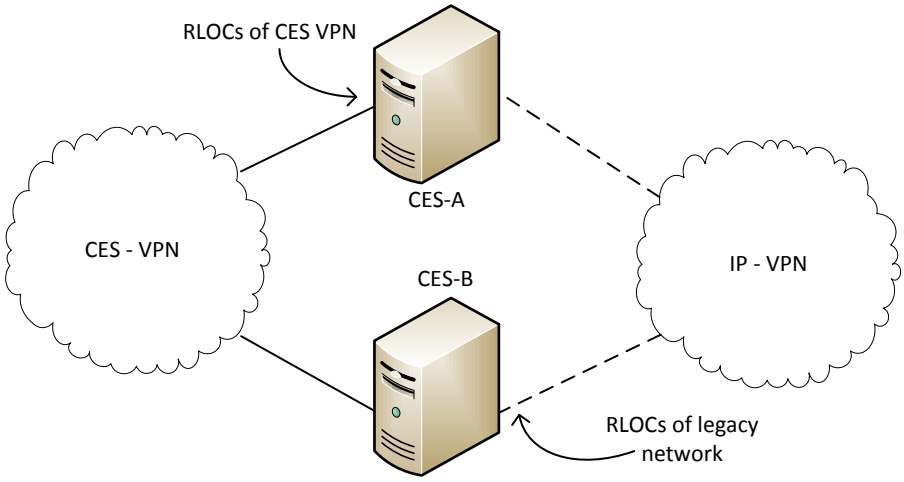


Figure 5.3 CES deployment model to prevent attacks

To prevent the attacks from legacy hosts, CES can benefit from a deployment model presented in Figure 5.3, where the traffic from a legacy host is expected over a separate interface than the CETP traffic from a legitimate CES. Hence, CES devices can ingress filter the traffic received over the legacy interface to drop CETP attack traffic. This protects the CES node against attacks described below. However, when the device with CES functionality is not large, e.g. an ADSL modem, the CETP traffic may be received over the same link as IP. This leaves a window of opportunity for an attacker controlling a legacy host that can send forged CETP packets towards CES to launch a DoS attack.

5.1.3 CETP Attack-1

Figure 5.4 presents a security vulnerability, where a legacy host with CETP attack module forwards spoofed CETP packets towards CES-B. Upon receiving the CETP packet fulfilling the destination policy, CES-B opens a connection with the sender without eliminating the source address spoofing in the received packet.

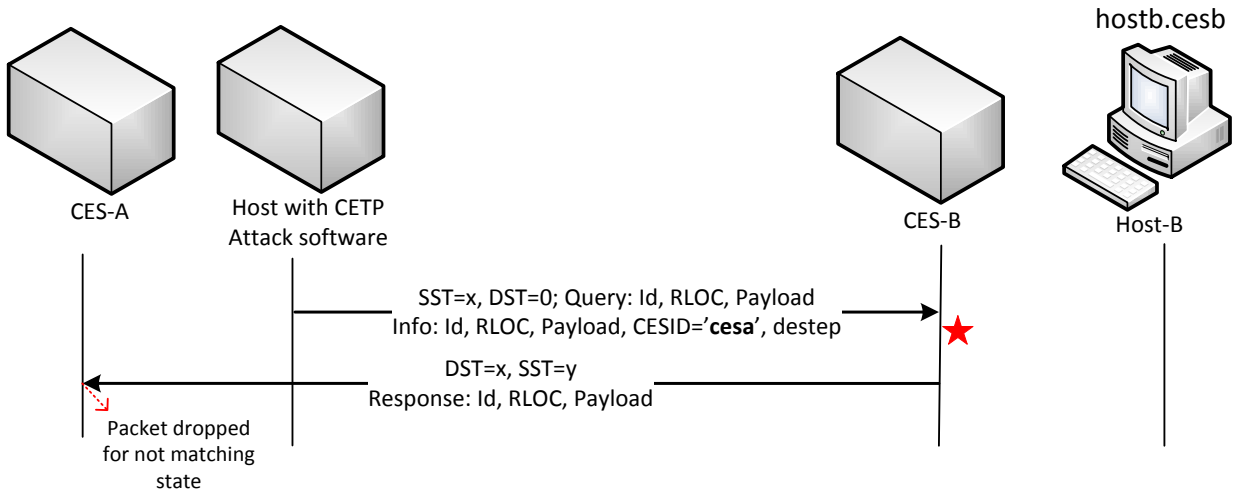


Figure 5.4 CETP Attack-1

Damage: The attacker opens a connection in the iCES by sending a spoofed CETP packet. For a bot-controlled legacy host, this can result in a DoS attack on the inbound CES. A large number of spoofed CETP packets received can open multiple connections in the iCES, and thereby increase the resource consumption and processing overload in the iCES.

Vulnerability: In a shared VPN of legacy hosts and CES, the CES architecture is vulnerable to this attack only if spoofing the source address is possible.

Counter-measures: The CES architecture can employ a cookie mechanism to authenticate the claimed sender of the CETP packet, and hence eliminate spoofing attacks. This prevents an iCES from creating connection upon receiving a spoofed CETP packet.

5.1.4 CETP Attack-2

Figure 5.5 describes a vulnerability, where the Attacker-CES replays an earlier communication between the source Host-A (hidden behind the CES-A) and the destination Host-B. Upon reception of the replayed CETP packet, a connection is created in the inbound CES after the destination policy requirements have been met. Such replay attacks succeed even in the presence of cookie mechanism, unless the cookie computed for each CETP transaction is unique.

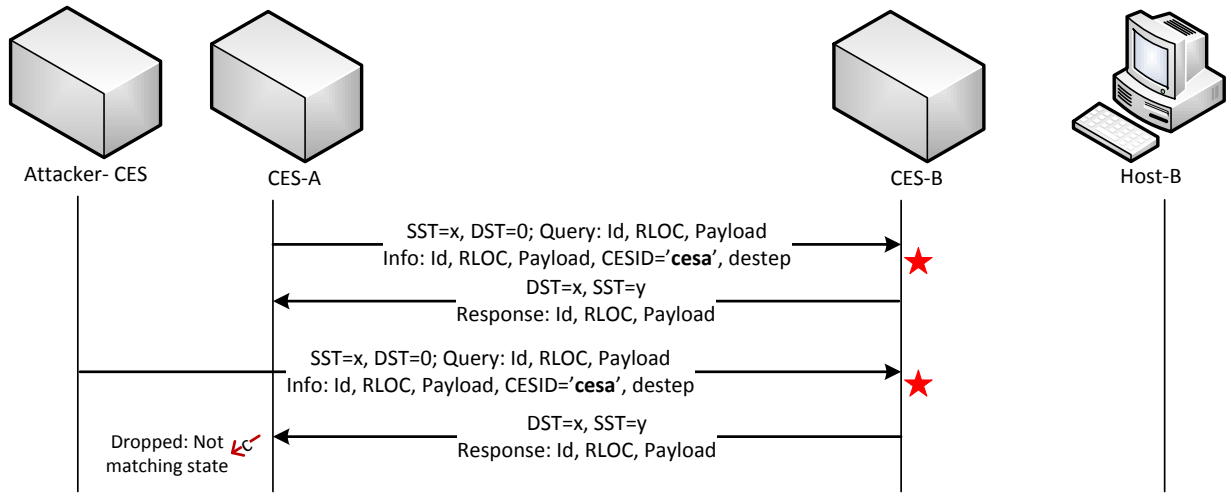


Figure 5.5 CETP Attack-2

Damage: The attacker opens a connection in the iCES by replaying CETP packets from an earlier communication. A large number of these replayed packets can result in a DoS attack, by increasing the resource consumption and processing overload in the inbound CES.

Vulnerability: The CES architecture is vulnerable to this attack only if spoofing source RLOCs is possible. The replay attack can result in spurious connection establishments in the iCES.

Counter-measures: The cookie mechanism proposed can also provide protection against replay attacks if the cookie computed for each transaction is unique, e.g. by introducing an expiration time in the cookie. This helps an inbound CES to detect a replayed packet and hence prevents the iCES from establishing the connection.

5.1.5 CETP Attack-3

Figure 5.6 presents a security vulnerability, where a legacy host with CETP attack module imitates as CES-A and sends CETP packets towards CES-B. Upon receiving the CETP packet fulfilling the destination policy, CES-B opens a connection with the sender without verifying the authenticity of the sender. Such an attack succeeds even in the presence of spoofing elimination techniques, as the attacking host uses its actual address to send the CETP packets.

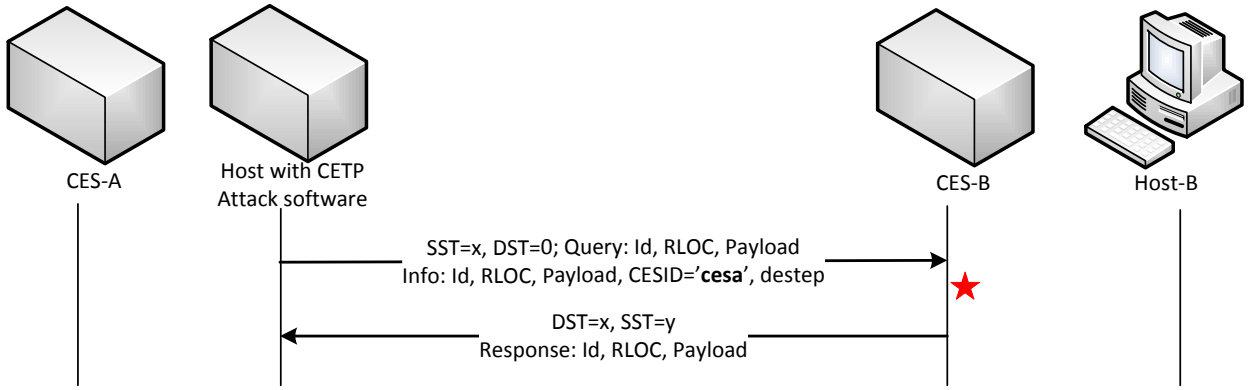


Figure 5.6 CETP Attack-3

Damage: The attacker successfully establishes a connection with the victim behind CES-B.

Vulnerability: The CES architecture is vulnerable to this attack, if CES does not determine the legitimacy of the CETP packet source.

Counter-measures: For a received packet, after eliminating the RLOC spoofing, the iCES can use a CES verification mechanism to determine if the CETP packet source is a legitimate CES. For a verification failure, the sender address is logged and subsequent CETP packets received from this address are dropped by CES. With spoofing eliminated, the iCES node can put the blame on the sender, in case a suspicious activity is detected later on.

5.1.6 CETP Attack-4

Figure 5.7 presents an attack, where "MITM-CES" between CES-A and CES-B launches a Man-in-the-middle attack after successfully compromising the routing infrastructure, e.g. DNS cache poisoning. The CETP packet originated from CES-A is received by MITM-CES and forwarded in the direction of the CES-B, after performing suitable changes.

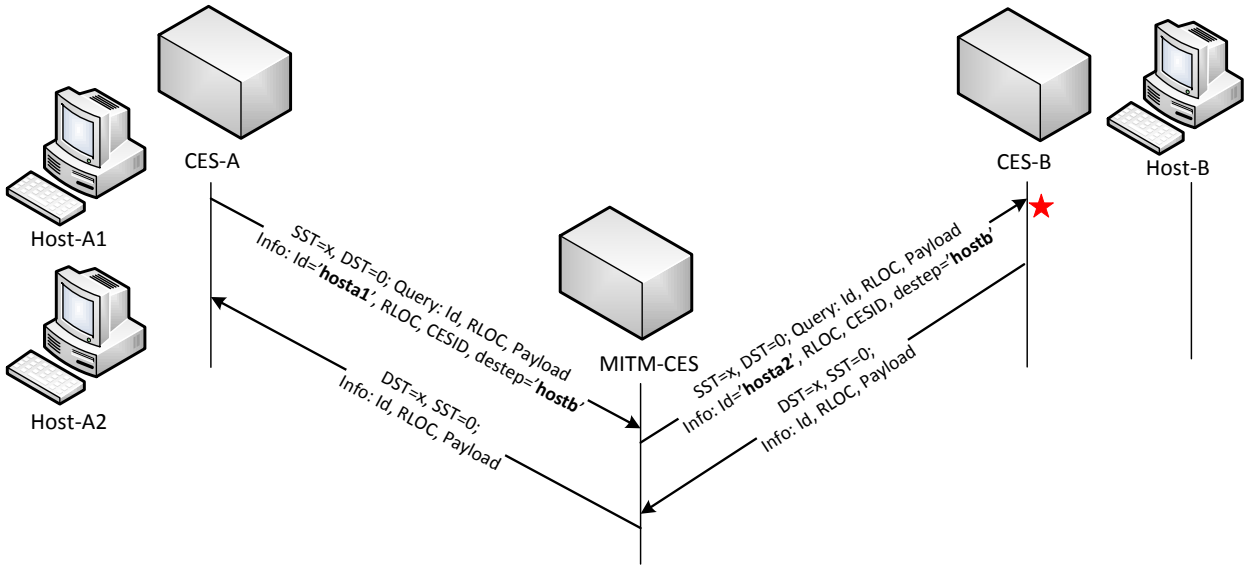


Figure 5.7 CETP Attack-4

Damage: A successful man-in-the-middle attack compromises the integrity of the messages exchanged between two CES devices. A MITM attacker can either passively eavesdrop on a communication or can masquerade as a legitimate source and steal the victim's information.

Vulnerability: The attack can affect all the communications with a remote CES for which the routing infrastructure has been compromised.

Counter-measures: The use of cryptographic signatures is well known to ensure the message integrity, even in the presence of a compromised infrastructure. Similarly, encryption techniques can be used to protect a communication against eavesdropping attempts.

Limitations: The use of cryptographic signatures and encryption techniques to prevent a MITM attack relies on Public Key Infrastructure (PKI), which increases the CES-to-CES delay because of processing involved in computing/verifying the signatures and encrypting/decrypting the communication flows.

5.2 Circular Pool Vulnerabilities

This section first describes the operations of the circular pool model. Next, it presents different attack scenarios that expose the vulnerabilities present in the circular pool model.

5.2.1 Operation

The circular pool model consists of a set of public IP addresses which are used to represent a host behind the CES to a remote host. The circular pool model allocates public IP addresses to incoming and outgoing connections following a circular mechanism, beginning from the start of the address pool, picking up the next available address and then back to the start upon reaching the end of the address pool.

For an outbound connection, a host can reserve an address from the circular pool regardless of a DNS query i.e. analogous to NAT behavior, described earlier. However, for incoming connections, the design of the circular pool is highly dependent upon Domain Name System (DNS). Following a DNS query received for a domain behind the CES, the circular pool model reserves a public IP address from its address pool and creates a connection state in ‘waiting’ mode.

The connection state consists of a tuple of information: sender IP address, allocated IP address, private IP of the destination, status (waiting or active) and a timeout value of the state. Following the ‘waiting’ state, an incoming data packet for which no active state exists, and whose destination address is the same as the destination address of the waiting state, is believed as the source of the DNS query and the data packet is forwarded to the destination behind the CES. Following which, the address reserved is released and it is returned to the circular pool, for future connection establishments [4].

5.2.2 Limiting Factor

The circular pool uses the addresses that are not in 'waiting' state to establish new connections. If all of the circular pool addresses are reserved when a DNS query is received, then the circular pool cannot serve this DNS request and the connection request is dropped. This state of circular pool is called *blocking state* and it is directly related to the number of public IP addresses in the circular pool. It is pertinent to mention here that this state is not permanent, and it does not affect ongoing connections in the CES [4].

Given the capacity of a server behind PRGW to serve the arrival of N connection requests per second, the required number of CPOOL addresses can be calculated such that the CPOOL does not become the bottleneck for capacity of the server. However, if the server capacity is higher than N, the number of public IP addresses required in the CPOOL is high and the CPOOL is not scalable for this scenario. It is pertinent to mention here that CPOOL does not handle the HTTP and HTTPs traffic, rather they are processed using the reverse proxy method by PRGW [4].

The presence of this bottleneck, i.e. blocking state, in the circular pool model offers a vulnerable spot to an attacker, and it can be a target of different Denial of Service (DoS) attacks. In the following sections, we take a look at different attack conditions, which exploit this vulnerability in the circular pool model.

5.2.3 Attack-1

Figure 5.8 presents a security vulnerability in the circular pool model, where Attacker-E2 continuously sends IP packets to circular pool address, R1. These packets are dropped by the CES for having no corresponding 'waiting' connection state. Amidst this, Host-E1 performs a DNS query to access the domain hosted by Host-A and reserves a connection state addressed to an unknown sender and CPOOL address 'R1'. Before Host-E1 responds, a data segment from Attacker-E2 arrives and hijacks the connection reserved for Host-E1.

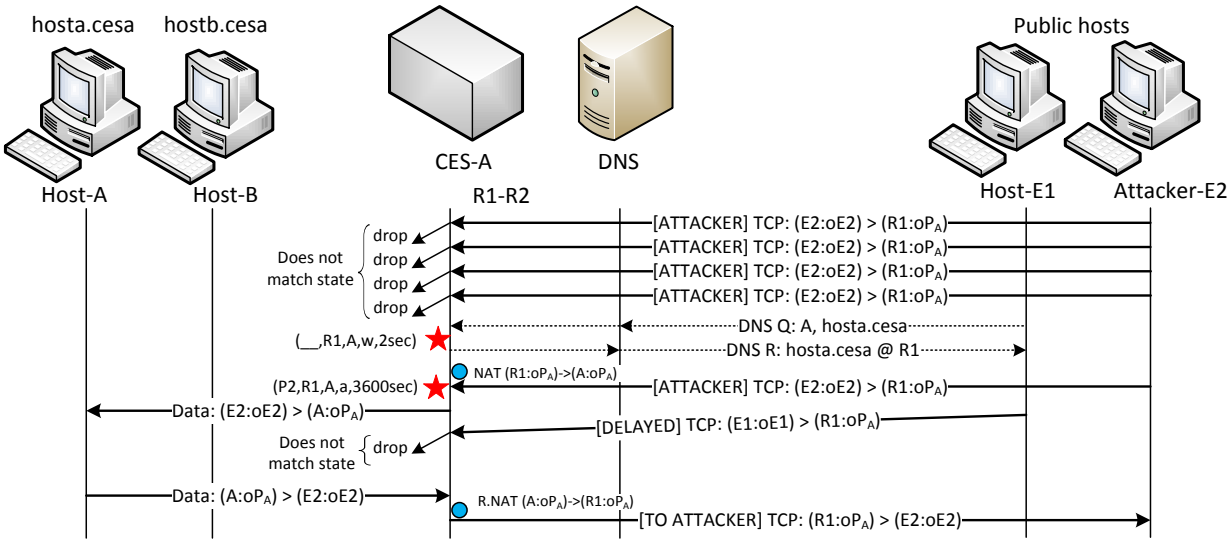


Figure 5.8 Attack-1: Hijacking a connection state in circular pool

Damage: Attacker-E2 hijacks the legitimate connection created by the Host-E1, which results in DoS to legitimate Host-E1 since its packets are dropped by CES.

Vulnerabilities: The vulnerability only affects the connections in the waiting state, whereas ongoing connections are never affected. A DDoS attack launched from different botnet machines can target and hence take over all the connections in the waiting state.

Countermeasures: CES policy should be to drop UDP, as allowing a UDP flow to a domain behind the CPOOL is not secure without some prior signaling e.g. SIP.

A TCP segment should be accepted only after the spoofing has been eliminated by CES, to determine the legitimacy of the sender. CES can generate a blacklist of non-spoofing hosts whose packets are constantly being dropped during time "T". During attack time, a packet from this source should not be accepted for claiming a 'waiting' state in the circular pool.

5.2.4 Attack-2

This section presents an attack in Figure 5.9, where the Attacker-E2 continuously sends DNS queries to different domains behind the CES. This reserves all the circular pool addresses from the address pool. With all the circular pool addresses reserved for connections that never complete, CES is forced to drop subsequent connection requests and this results in DoS to potential legitimate hosts.

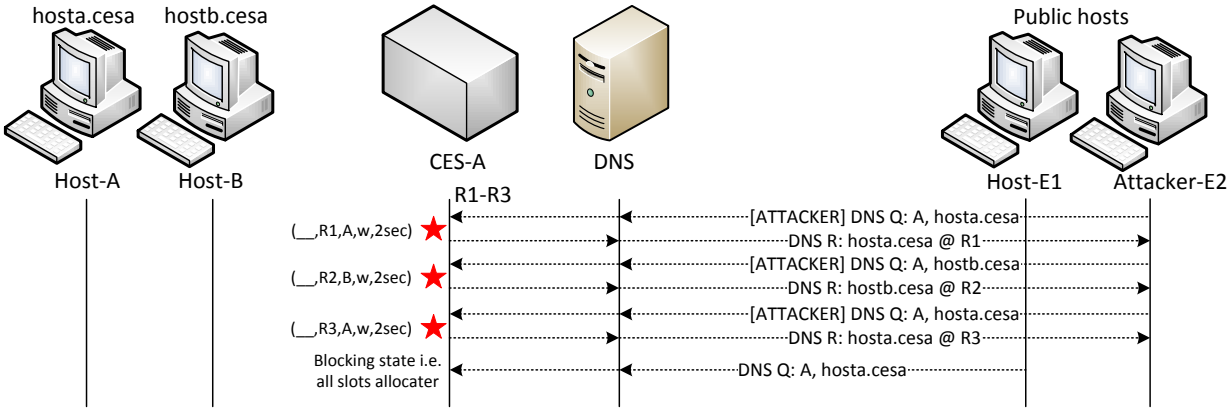


Figure 5.9 Attack-2: Achieving blocking state in the CES by targeting different domains [4]

Damage: CES is unable to accept any new incoming connection requests. However, this state is temporary and the attacker must keep sending DNS queries to retain the blocking state.

Vulnerability: Ongoing connections are not affected by this vulnerability. An attacker must know sufficient number of domains behind the CES in order to engage all the circular pool addresses, provided that limited numbers of connections per domain are allowed.

Countermeasures: Allow a limited number of connections in the waiting state per source of the DNS query. A DNS request from the source after this limit should be dropped. Also, if connection success rate from a DNS server falls below a threshold, blacklist the DNS server for time duration 'T_O' i.e. do not accept the DNS requests from this name server.

Limitations: It could be a risk to place a popular service behind the circular pool as it can draw a large number of connection requests and cause frequent blocking state in the circular pool [4].

5.2.5 Attack-3

Figure 5.10 presents an attack scenario, where an attacker sends multiple DNS queries through different DNS servers to domain ‘hosta.cesa’. Each DNS query reserves an address from the circular pool until there are no addresses left in the circular pool. The unavailability of an address in the circular pool results in a denial of service to connection requests from a legitimate sender.

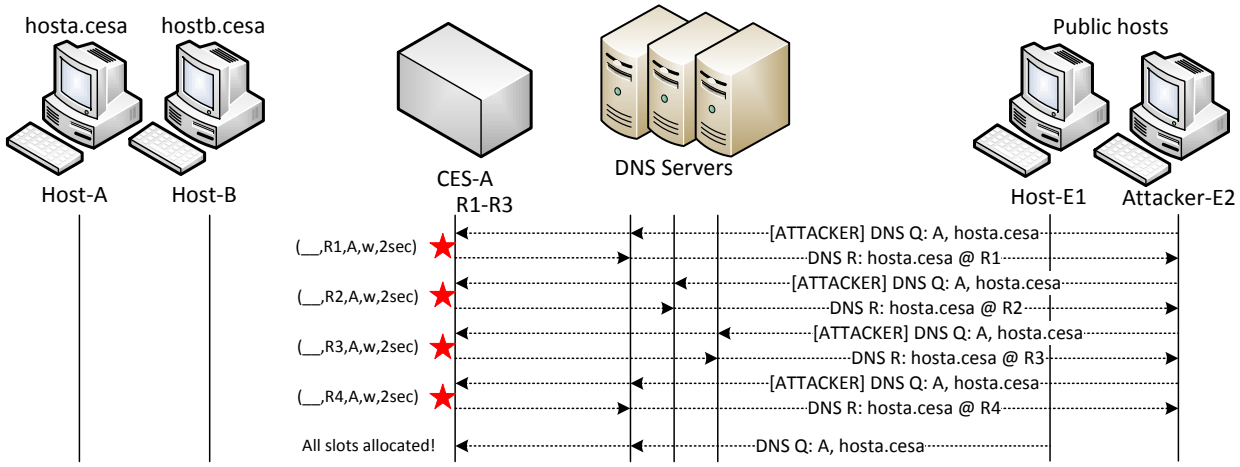


Figure 5.10 Attack-3: Attacking a single domain behind CES from different DNS servers

Damage: CES is unable to accept new inbound connection requests. Again, the state is temporary and the attacker must keep sending DNS queries to retain the blocking state.

Vulnerabilities: Ongoing connections are not affected by this attack. An attacker only needs to know a single domain behind the CES to launch this attack.

Countermeasures: Allow a limited number of connections in the waiting state per domain. A DNS request received for the domain after the maximum limit should be dropped.

Limitations: Again, a popular service can cause frequent blocking state in the circular pool, and therefore it could be a risk to place it behind the circular pool.

5.2.6 Attack-4

Figure 5.11 presents a distributed version of a denial of service attack, which is a hybrid of Attack-2 and Attack-3. Here, Attacker-E2 host sends DNS queries to different domains behind the CES through various DNS servers. Principally, this reserves all the circular pool addresses in the 'waiting' state and thereby forces a CES to drop subsequent incoming connection requests.

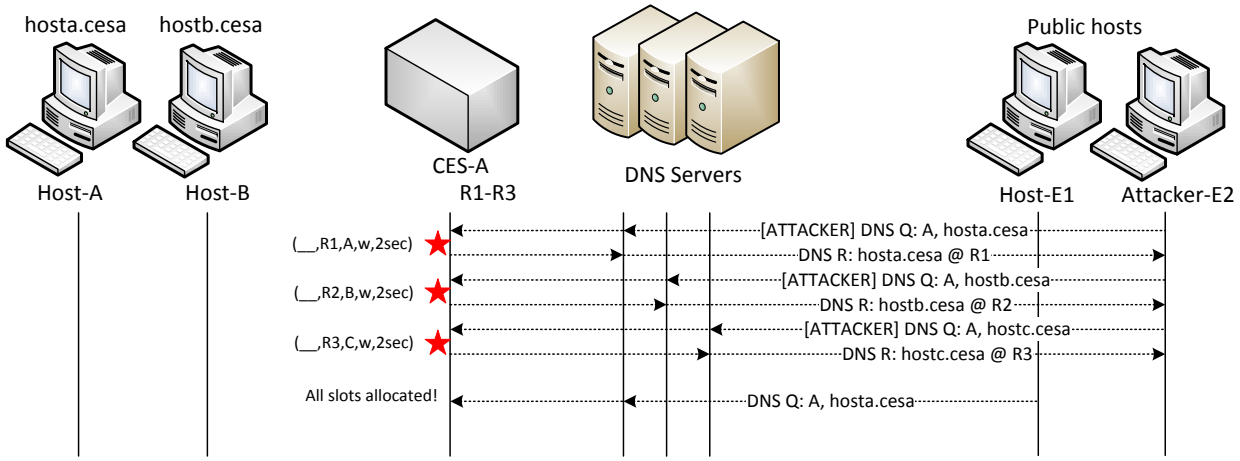


Figure 5.11 Attack-4: DDoS attack targeting different domains behind CES from multiple DNS servers

Damage: CES reaches the blocking state, and it cannot serve new incoming connection requests.

Vulnerabilities: Only connections in the waiting state are affected by this attack and ongoing connections remain unaffected. An attacker needs to know sufficient number of domains behind the CES and send DNS requests though sufficient number of name servers to reserve all the circular pool addresses.

Countermeasures: Besides limiting the number of connections in the waiting state per source of the DNS query and the domain queried, the circular pool model can benefit from different sets of interfaces for receiving connection requests from the whitelisted and the greylisted sources.

The circular pool model can also define an address allocation criterion, as presented in section 6.2.3, to limit the extent of a DDoS attack. Here, a set of circular pool resources is always available to a whitelisted name server, even under DDoS attack conditions. Naturally, a connection request that cannot fulfill the address allocation criteria is dropped.

6. Securing Customer Edge Switches

The chapter describes various security mechanisms added to secure CES against security vulnerabilities present in its architecture. The chapter concludes by presenting a security model to protect the CES-to-CES communication and the circular pool model against security vulnerabilities identified in Chapter 5.

6.1 Security of CES-to-CES Communication

This section presents the security mechanisms developed to secure the CES-to-CES communication against attacks. The section first explains each security mechanism individually and then concludes by presenting a CETP security model to protect CES-to-CES communication against all vulnerabilities.

6.1.1 Principles of Security Mechanisms

The security mechanisms designed to secure the CES architecture adhere to following principles:

1) A light weight attack should consume minimal processing at an inbound CES. Heavy verification mechanisms on attack packets generated with minimal processing give the attacker an advantage, where the attacker can flood the CES with huge attack volumes and force the CES into a denial-of-service state.

2) The response to light weight received packets should be small, to prevent network traffic amplification. The detailed response can be sent after spoofing has been eliminated.

3) The CES architecture must eliminate source address spoofing before admitting a packet for heavy verification checks. This prevents CES from carrying processing-heavy verification checks upon receiving a spoofed CETP packet.

4) Heavy verification mechanisms, executed after eliminating spoofing, must guarantee the legitimacy of the CETP packet source.

5) With spoofing eliminated, a failure in heavy verification mechanisms must enable the CES to attribute an attack to the packet source, and maybe present the attack evidence to a trust reputation system or GTO.

Adhering to these principles, security mechanisms have been developed for the security of CES-to-CES communication. CES defines a CETP cookie algorithm as a light weight protection mechanism in order to eliminate source address spoofing. Whereas, CETP header signature and HSS verification are defined as heavy verification mechanisms because of the processing involved. These mechanisms are described in the following sections,

6.1.2 CETP Cookie

Section 5.1.3 describes a vulnerability in the CES architecture, where an inbound CES opens a connection in the CES upon receiving connection requests from a spoofed sender. As a countermeasure, this thesis introduces a cookie mechanism in the CES architecture to eliminate spoofing in the CETP packets received for connection establishment.

For an incoming CETP packet without Cookie-TLV, the iCES extracts the SST, DST, Host-ID, Destep and CES-ID values from the received packet. These values are added with a locally unique SECRET, and a SHA-1 MD is computed. Next, a timeout value of 'T_O' seconds from the current time is encoded and appended to the last 4-bytes of the computed MD. This value is encrypted with a 64-bit DES-key to generate the cookie, which is inserted into the Cookie-TLV and sent towards the sender. The cookie computation process is explained in Figure 6.1, while the mathematical equation describing cookie computation is presented next,

Cookie = 64-bit DES Encryption (Last 4-bytes of MD {SST, DST, Host-ID, Destep, CES-ID, SECRET} + T_O)

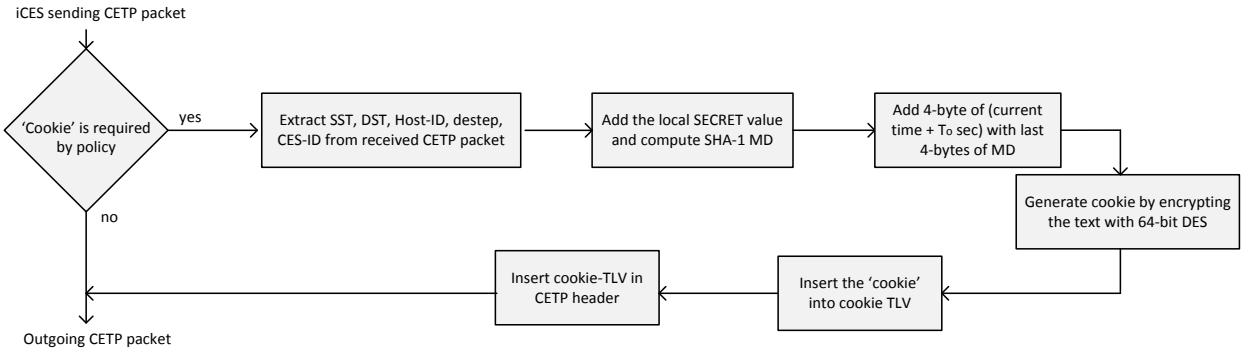


Figure 6.1 Cookie computation by inbound-CES

Cookie Verification

If a CETP packet received by the iCES contains a cookie-TLV, then the cookie is verified according to Figure 6.2. The receiver decrypts the cookie with the symmetric 64-bit DES key and the timeout value is extracted from the decrypted cookie. For a replayed attack packet, this timeout value would be smaller than the current time and this will lead to CETP packet drop by the iCES. However, for a valid timeout value, the receiver CES extracts SST, DST, Host-ID, Destep, CES-ID values from the received CETP packet and a SHA-1 MD is computed with locally unique SECRET.

Next, if the last 4-bytes of the computed MD are the same as the first 4 bytes of the decrypted cookie then cookie is considered as verified, and the CETP packet is believed to have come from a non-spoofing source. However, a cookie verification failure would result in connection denial to the sender, and the CETP packet is dropped as it is believed to be a spoofed packet.

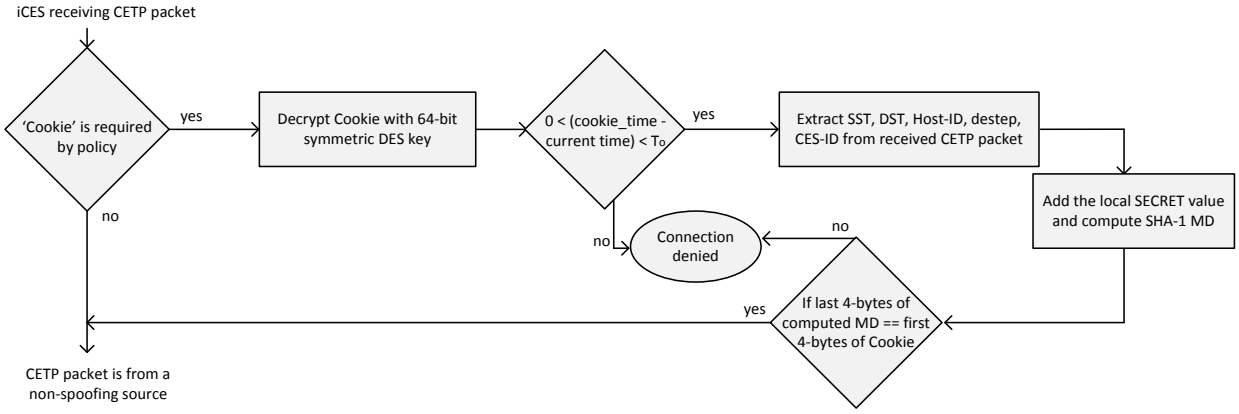


Figure 6.2 Cookie verification by an inbound-CES

Cookie-TLV processing in oCES:

For a CETP packet received with matching (SST, DST=0) state in an outbound CES, if the received packet carries a query Cookie-TLV then the outbound CES is expected to relay the received cookie back in the CETP response towards the iCES. The successful verification of cookie-TLV at the iCES followed by the destination policy fulfillment would lead to a successful connection establishment in the iCES.

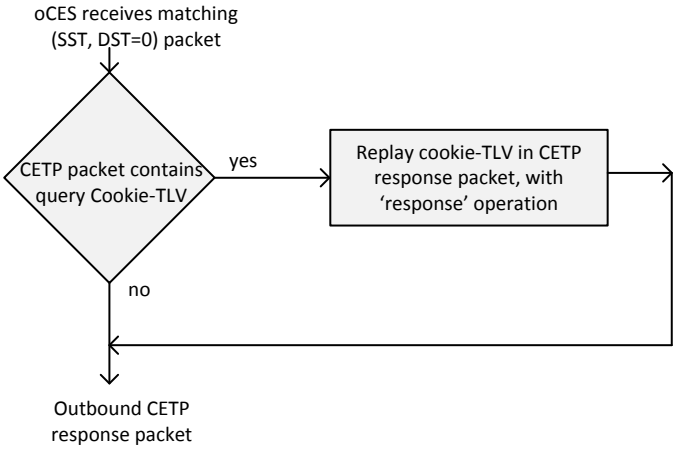


Figure 6.3 Cookie-TLV processing in oCES

Consequences of Cookie Deployment

As a consequence of introducing the cookie in CETP, an iCES only processes a connection request after the sender has been verified as a non-spoofing source. However, with cookie-TLV in place, the CETP connection establishment process always completes in 2RTT, as the 1st RTT CETP response packet carries the cookie-TLV from the iCES, which has to be relayed back by the oCES in the 2RTT CETP packet along with all the policy requirements listed by the iCES.

The cookie mechanism implemented in CETP provides protection against Spoofing attacks and Replay attacks identified in section 5.1.3 and section 5.1.4, respectively. An attacker cannot forge a CETP cookie because of three layers of protection: 1) the locally kept SECRET, 2) the unique 64-bit DES key and 3) the expiration timeout. The 64-bit DES-key and the SECRET are kept local, and are not known to any other entity. Moreover, the timeout value included in the cookie helps an iCES detect and thwart a replay attack from a sender. The algorithm used for cookie computation limits the cookie size to 8-bytes, and thereby avoids unnecessary CETP header overload, but it still keeps the cookie size long enough to prevent the spoofer from forging a valid cookie. The choice of small cookie size comes in agreement with the need to prevent amplification attacks, i.e. a long cookie generates lengthy packets for processing.

6.1.3 CETP Header Signature

The CES architecture uses digital signatures in order to prevent an unauthorized third party intrusion into a communication. The approach bears resemblance with how Transport Layer

Security (TLS) protocol [37] provides security against MITM attacks in many protocols e.g. in HTTPS, using certificates issued by trusted certificate authorities. However, unlike negotiating many security parameters in TLS for sender authentication and encryption of the subsequent packet flow, CETP only authenticates the sender based on its public-key certificate. The process involves computing the message-digest (MD) of the CETP header, excluding the signature-TLV. Next, the computed MD is signed with the private-key of the sender to generate the signature, which is inserted into the signature-TLV and sent towards the destination in a CETP packet.

This use of signatures not only ensures message integrity but also guarantees authentication and non-repudiation, since only the claimed sender could have generated the signature which will be successfully verified from the public-key certificate of the sender. With spoofing eliminated, a failure in signature verification process identifies the source of the attack. Following which, the subsequent CETP packets from the attacker are dropped.

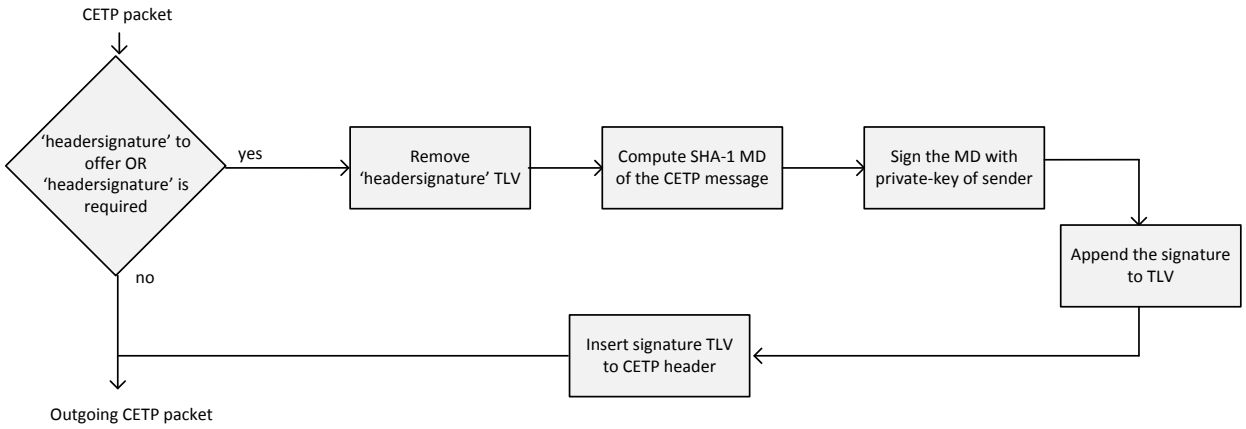


Figure 6.4 Signature computation

The CETP header signature is only computed if the host policy offers 'headersignature' policy element and if the receiver requires the 'headersignature' from the sender, to deter against MITM attack attempts.

Signature Verification

For an incoming CETP packet, the inbound CES carries out the signature verification process only if the signature TLV is present in the received CETP packet, and if required by the destination policy. The signature verification process involves computing the MD of the received CETP header after removing the signature TLV. Next, the signature extracted from the signature TLV is decrypted using the public-key of the sender. Finally, the decrypted MD is compared

with the computed MD, and a mismatch of both MDs confirms that the received packet has been subjected to alterations by an unauthorized third party. The CETP connection request is terminated following the signature verification failure.

The public key of the sender is accessed from the public-key certificate of the sender, which is downloaded from the HSS address carried in the CACES-TLV of the received CETP packet. The use of signatures to thwart MITM attack attempts mandates the presence of 'cesid' and 'caces' policy elements in the host policy, because a remote CES requires these policy elements in order to download the sender certificate from HSS and verify the CETP header signature.

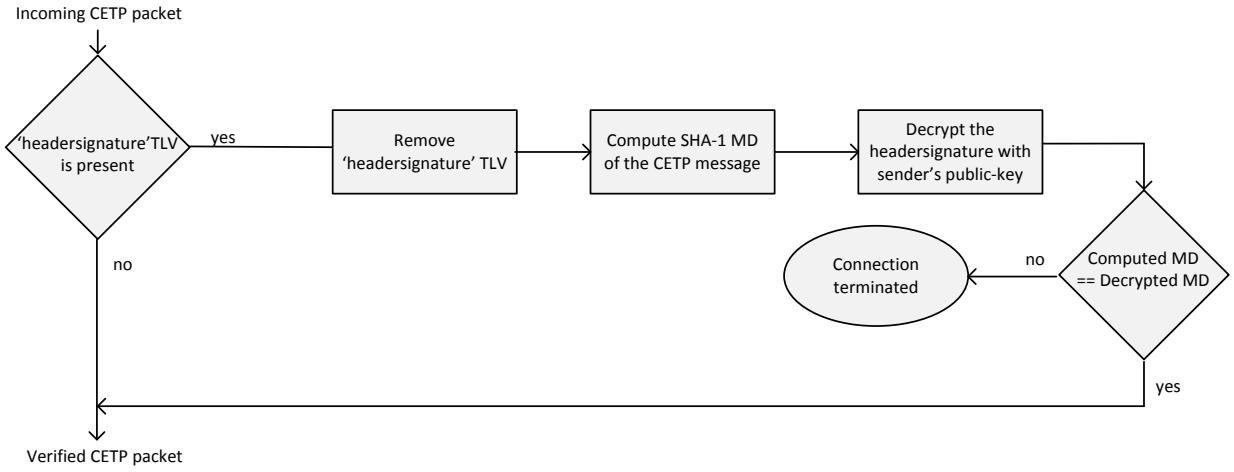


Figure 6.5 CETP signature verification process

6.1.4 Certificate Authority

The CES architecture uses public-key cryptography for computing and verifying the CETP header signature. A receiver accesses the public-key of the sender to verify the received header signature. The public-key is carried in a X.509 certificate issued by a certificate authority (CA), which vouches for the binding between the source-id and the corresponding public-key carried in the certificate. This thesis implements a CA to issue X.509 certificates to the CES devices for multiple purposes: to counter the MITM attack presented in Figure 5.7, and for CES registration process as explained in section 6.1.5.

Figure 6.6 presents the CA implementation in the current CES prototype. The grey-shaded area in the figure indicates the steps executed once per validity period of each certificate in the CES device. The process begins by sending a Certificate Signing Request (CSR) to the CA. The CA

returns a certificate in response to the received CSR message, after performing verification checks. The CES stores the certificate received from the CA and also uploads it to the HSS.

The step 5 and 6 in the Figure 6.6 are executed once by an iCES, for the first connection request received from a remote CES, if the iCES requires header signatures from the remote CES. The iCES downloads the sender's certificate from the HSS address carried in CACES-TLV of the received packet. For a CA issued certificate, the digital signature of the certificate can be verified using the public-key certificate of the CA. In case of verification, the certificate is stored locally and it is used to verify the header signature of the CETP packet received from the remote-CES, according to section 6.1.3. A new request for the sender's certificate is issued after the stored certificate has reached its expiration time.

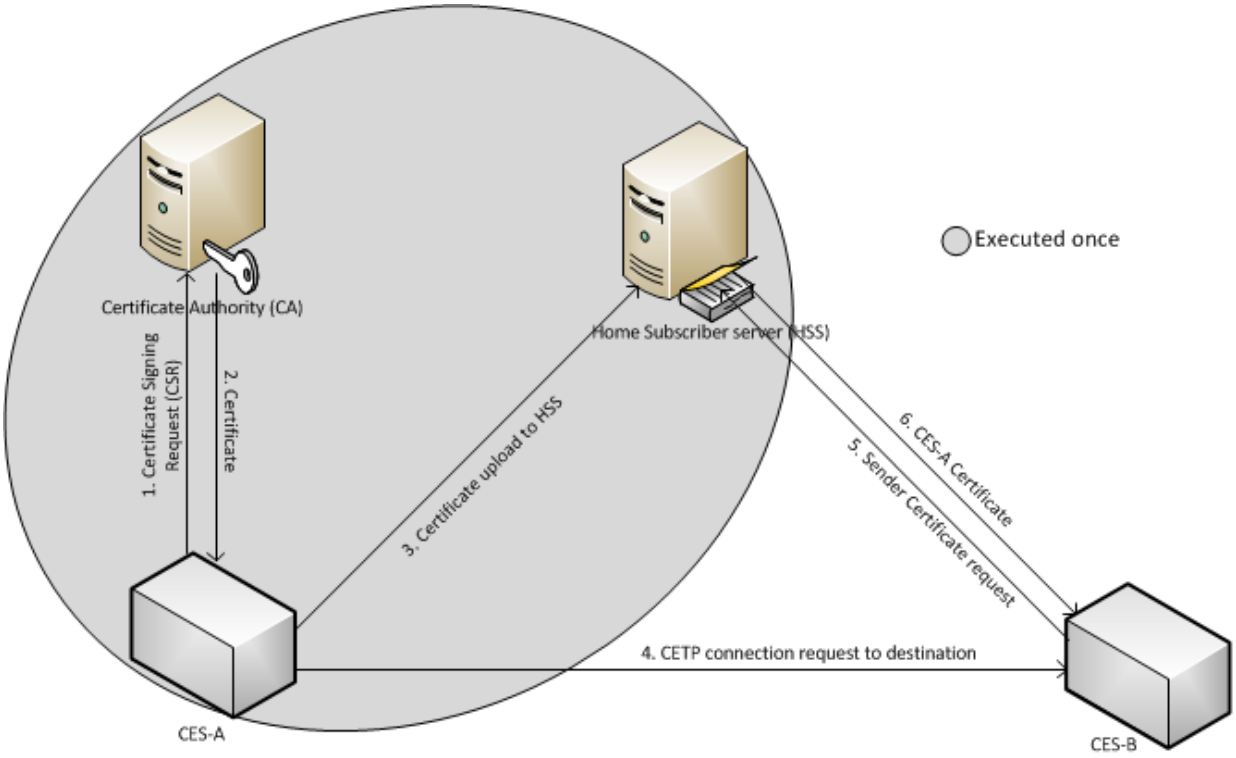


Figure 6.6 CES Certificate Authority

Difference from Real-world Certificate Authority

In real world, a CA carries out various detailed verification checks on a received CSR i.e. verifies the certificate requestor, claimed resource, public-key, the requestor contact information etc. before issuing a certificate. After successful verification, the certificate is issued to the requesting entity through e-mail, web-interface or to sender's contact information. However, the

CA implemented in this thesis performs minimal verification checks on a received CSR and returns the newly created certificate back to the requestor through a custom protocol.

In communication networks, during data-exchange, a sender signs the message with its private-key and sends the public-key encapsulated in a X.509-certificate to the receiver for signature verification. But, given the fact that CETP protocol does not support packet fragmentation, and that X.509 certificate sizes along with header signatures exceed the MTU of physical layer i.e. 1500 bytes, the CES devices exchange their certificates through HSS database. Therefore, in the current CES prototype, the sender conveys the HSS address to the receiver in CACES-TLV for downloading the certificate rather than sending the actual certificate in the CETP packet.

6.1.5 Home Subscriber Server (HSS)

Similar to HLR/HSS in mobile communication networks, the use of HSS as a central repository is aimed at securing the CES-to-CES connection establishment. An HSS server normally contains multiple databases which maintain user-related information such as location information, security information (certificates), user profile information etc. For a received CETP packet, a CES can verify the sender's credentials with HSS by issuing multiple queries to relevant databases in the HSS. The successful validation of these queries with HSS guarantees that the received CETP packet has come from a legitimate source. The implemented HSS verification process is presented below, where a connection is terminated if either of the received TLVs could not be verified with the HSS.

A CES performs the HSS verification process only if the destination requires CACES policy elements from the sender. The CACES-TLV contains the HSS address that a receiver must contact in order to verify the sender credentials received in the inbound packet. The HSS verification process requires Host-ID, CES-ID and RLOCs information in the received packet to verify the source of the CETP packet. The HSS verification mandates the presence of the CES-ID policy element in the host policy, when CACES is present, because the CES-ID is required to carry out the verification queries with the HSS.

To simplify the HSS verification process, two separate TLVs: CACES and CAEP, have been defined for CES-ID and Host-ID verification in CES. This enables a scenario where the CES and the host related data can be managed separately. For example, RLOCs associated with CES devices can be stored in one database (globally). Whereas, the domains registered within a CES

can be managed locally by the CES, in a separate database. A remote CES can contact this database at the address provided in the CAEP-TLV to verify the sending host's identity.

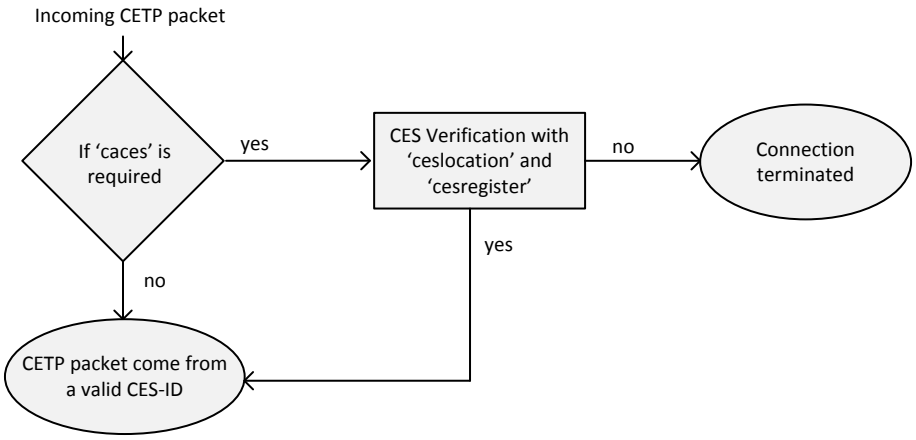


Figure 6.7 HSS based CES-ID verification process

For RLOC verification, the HSS provides '**ceslocation**' database which maintains the list of RLOC-types and RLOC-values corresponding to a CES-ID. The CES issues a query to the HSS for the rloc-value against the ces-id and the rloc-type received in the CETP packet. If the response from the HSS carries the same RLOCs as the one received in the CETP packet, the packet is believed to have come from a legitimate CES.

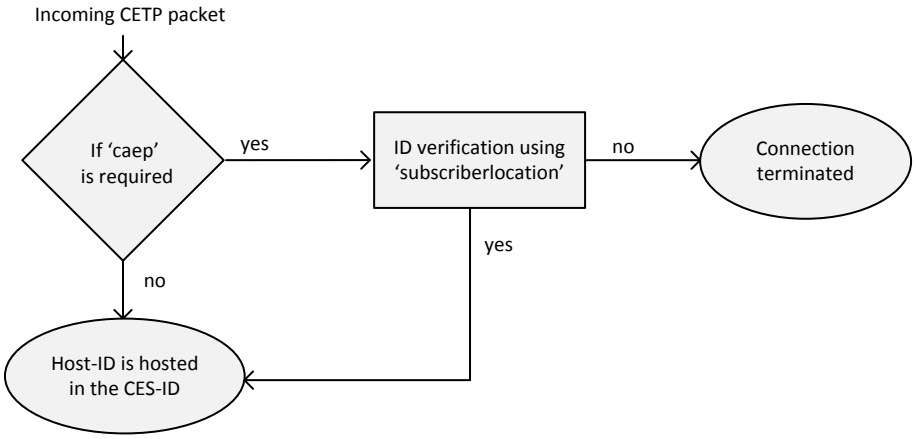


Figure 6.8 HSS based Host-ID verification

After performing CES verification, the CES verifies the host-ID by contacting the '**subscriberlocation**' database, which contains a list of Host-ID type and Host-IDs registered corresponding to a CES-ID. Upon receiving a CETP packet, the CES connects with the HSS at the address provided in the CAEP-TLV and issues a query for the host-id against the received

CES-ID and the host-id type. If the response from the HSS carries the same host-id as the one received in the CETP packet, the sender is determined as a legitimate host registered behind the CES-ID. Figure 6.8 describes how a received host-id is verified by the CES, if CAEP policy element is required by the destination policy.

6.1.6 CES Registration and Verification

CES devices deployed in the Internet or at mobile network boundaries must be identified/registered in order to differentiate them from the legacy elements present in the Internet infrastructure, i.e. NAT, servers, public-hosts etc. This identification/registration for the CES devices prevents a legacy host from sending forged CETP attack packets towards a CES. The registration/verification process enables a receiver to determine if the CETP packet has come from a legitimate CES device. This thesis proposes and implements two CES registration/verification mechanisms, described below,

Centralized CES Database: This mechanism involves registering all the CES devices in a centralized database, which maintains a list of CES-IDs and the associated RLOCs values. For a received CETP packet, the CES contacts the database and determines if the received CES-ID and RLOC values exist in the database. For a query match, the source of the CETP packet is believed as a legitimate CES device, after spoofing has been eliminated on the received packet. A remote CES only needs to be validated once by the local CES and subsequent CETP packets from the sender CES-ID and RLOC combination are accepted by the local CES without verification.

The current prototype employs a simplified version of this model, using '**cesregister**' and '**ceslocation**' database in the HSS. The 'cesregister' database maintains the list of legitimate CES-IDs. Whereas, the 'ceslocation' database contains RLOCs associated with the CES-ID. For a received CETP packet, the sending CES is accepted as a legitimate CES only after the CES-ID and RLOC values have been validated with these databases in the HSS.

Decentralized Registration: A centralized registration/verification mechanism, described above, requires additional infrastructure in the Internet and certain management activities to maintain such a database. Therefore, we propose a decentralized CES registration/verification mechanism that utilizes existing Internet infrastructure i.e. certificate authorities, to register/verify the CES devices. Hence, besides supporting digital signatures to thwart MITM attack attempts, the mechanism enables the receiver to determine if the source is a CES device.

The current version of X.509 certificate defines various extension fields that provide additional information about the certificate and put constraints to the certificate usage. These constraint extensions in X.509 certificate among others include: Basic Constraints, Key Usage and Extended Key Usage fields. The Basic constraint field defines if the certificate belongs to a Certificate Authority or to an End Entity e.g. client, server etc. In case the certificate belongs to an End Entity, the certificate contains Extended Key Usage field. This extension field, in the words of RFC 5280, "indicates one or more purposes for which the certified public key may be used, in addition to the basic purposes indicated in the Key Usage extension". We propose that a certificate issued to a CES device must carry the Extended Key Usage constraint that differentiates a certificate issued to a CES device from the rest of certificates. The absence of the "CES Verification" value in the Extended Key Usage field of the certificate prevents a (certificate bearing) legacy host from imitating the CES behavior, i.e. sending CETP packets.

For the first CETP packet received from a remote sender, the CES requests for the CETP header signature and the sender certificate from the sender. If the Extended Key Usage field of the received certificate carries the "CES Verification" value and if the CETP header signature can be verified with the received certificate, the sender is believed as a CES device. Otherwise, the CETP packets received from this sender are dropped by the CES, as they can be forged packets generated by a CETP App (or software) running on a bot controlled legacy host. Appendix-A presents the detailed description of CES registration/authentication mechanism based on the certificates issued by a CA.

6.1.7 CES Specific Policy Elements

Until now, the CES prototype has generated CETP packets according to the policy elements defined in the host policy. However, the security model introduces the concept of CES specific policy elements and suggests two such policy elements: Cookie and CACES.

An inbound CES node needs to request these policy elements from the sender before admitting the packet for connection establishment. The presence of cookie-TLV is mandatory to secure CES against spoofing attacks from the sender. Similarly, a CES node requires HSS address in CACES-TLV to determine if the sender is indeed a legitimate CES device. Therefore, from the security perspective, a CES needs to request these policy elements from a remote CES even if they are missing in the policy requirements of the destination host. The absence of these policy elements in the host's policy leaves an attack window for the attacker.

A CES node needs to define a minimum set of TLVs that a received CETP packet must have before it can respond to the packet. Otherwise, the packet should be dropped. The CES-to-CES security suggests that a minimum set of TLVs that a received packet must have should include: Host-ID, CES-ID, Destep and RLOCs. These TLVs constitute the minimum requirement because they are required for cookie generation/verification. Furthermore, a minimum set of required policy elements protects CES against spoofed attack packets with few TLVs, and amplification attacks that expect to generate more traffic by sending minimal TLVs. Therefore, the CES node needs to define a minimum set of required policy elements in the received packet and in the host's policy.

6.1.8 CETP Security Model

'CETP security model' refers to previously proposed security mechanisms put together in an orderly fashion to protect CES against security vulnerabilities in the architecture, identified in section 5.1. These security mechanisms affect the connection establishment process in inbound and outbound CES by introducing new processing modules, which are described next.

Inbound CES Security model

An inbound CES processes a received CETP packet according to the security model depicted in Figure 6.9. The CETP packet received over a legacy interface is dropped, in case the CES has separate interfaces for receiving traffic from legacy hosts and CES devices. However, if the CES shares the same interface to receive IP and CETP traffic, the security is assured by the procedures described below in this section.

The received packet is first checked for presence of a cookie-TLV. If the packet is not received with a cookie, the iCES responds to the sender with a cookie-TLV generated according to Figure 6.1. However, if the cookie-TLV is present, the cookie is verified according to process described in Figure 6.2. A failure in cookie verification process leads to the CETP packet drop, as the packet has come from an attacker or a spoofing source. But, if the cookie is successfully verified, this guarantees that the sender is a non-spoofing source. Fulfilling the security principles, the cookie mechanism protects the iCES from spoofing attacks. However, if the network does not allow spoofing at all, then the cookie mechanism to eliminate spoofing is not required in CES.

After spoofing has been eliminated, the iCES determines if all the required policy elements have been received in the CETP packet. For a missing required-TLV, the iCES generates a Full Query response packet and sends it in the direction of the outbound CES. In case all required policy elements are received, the iCES performs either of the verification checks listed in Figure 6.9.

Next, the iCES can use either CETP header signature or HSS verification to determine the legitimacy of the sender. CETP header signature is verified according to Figure 6.6, if the destination policy requires header signature from the sender. Whereas, the presence of CACES-TLV in the policy requirements of the destination triggers the HSS verification process, described in section 6.1.5. A failure in signature verification or HSS verification detects the reception of a forged packet, which would result in dropping the CETP packet received for connection establishment.

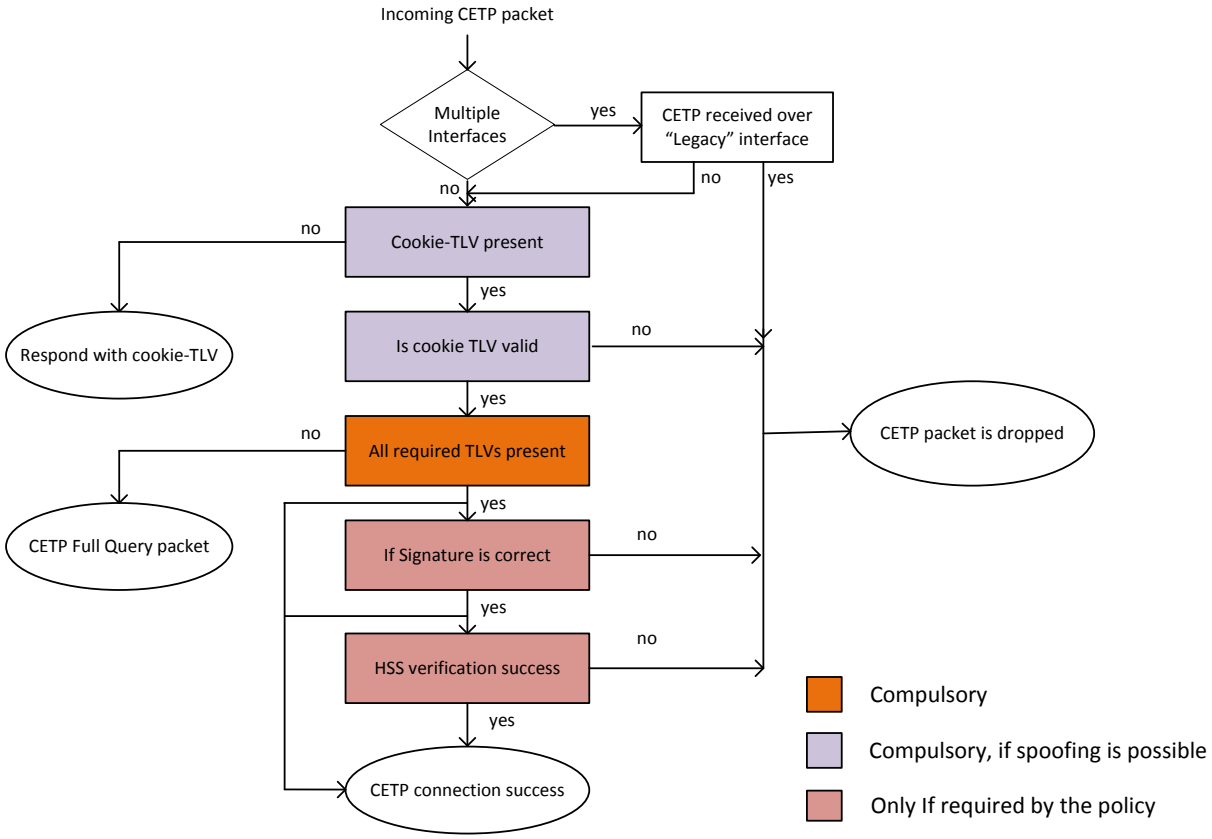


Figure 6.9 iCES security model

The CETP packet processing must follow the order presented in Figure 6.9. The cookie-TLV processing at the beginning of the security model prevents an iCES from carrying un-necessary

verifications on a spoofed CETP packet received, and hence saves processing time in the iCES. The ordering is also supported by the fact that symmetric-key algorithms used for cookie generation/verification take much less computation/verification time than corresponding public-key algorithms used for signature verification.

Outbound CES Security Model

For a received CETP packet matching with the connection state (SST, DST=0) in the oCES, if the packet contains a query TLV, the oCES needs to send a 2RTT CETP packet towards the inbound CES. However, if the CETP packet received for a matching (SST, DST=0) state contains no 'query' TLV, the oCES believes it as the last packet of the connection establishment and carries out either of HSS verification or CETP signature verification mechanisms to determine authenticity of the receiver, similar to the iCES security model.

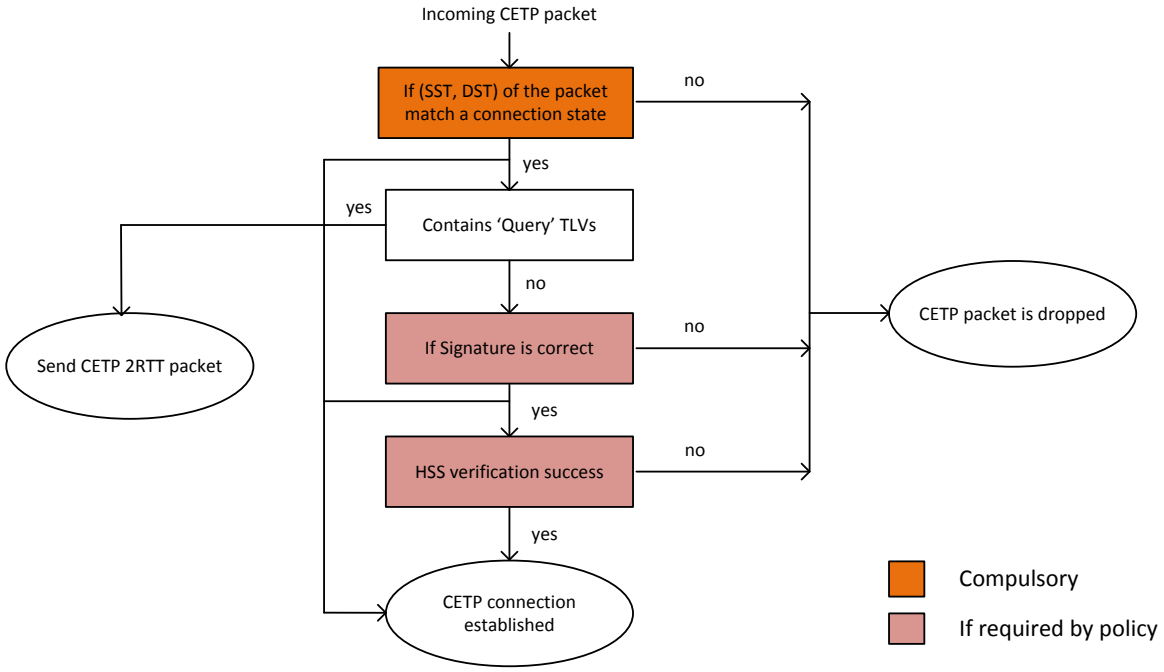


Figure 6.10 oCES security model

Consequently, a failure in either of the verification mechanisms leads to the CETP packet drop, whereas a verification success would lead to connection establishment with the iCES. Figure 6.10 presents the security related processing modules for a CETP packet received by an oCES.

6.2 Security of Circular Pool model

This section presents security mechanisms implemented in order to secure the circular pool against vulnerabilities present in its design. First, each security mechanism is described individually and finally the section concludes by presenting a security model to secure the circular pool against all vulnerabilities.

6.2.1 Blacklisting/Whitelisting DNS Servers

Circular pool divides DNS name servers into three categories Whitelisted, Greylisted and Blacklisted name servers. A whitelisted DNS server is a trusted name server, i.e. a mobile operator DNS or a paying network DNS, and hence it is not accessible to a user in the public Internet. A greylisted name server is a publicly available DNS server and can be accessed by any Internet user, and therefore it is offered comparatively fewer resources in the circular pool than corresponding whitelisted name server. Whereas, a name server is blacklisted if has recently generated attack traffic or malicious DNS queries. A blacklisted DNS is barred from accessing circular pool resources, i.e. all the DNS queries from this name server are dropped.

The circular pool model assumes that whitelisted DNS servers are specifically known to the circular pool. The circular pool expects to receive a DNS request from a whitelisted name server over a specific set of interfaces, in case multiple interfaces are available with the network. Any name server except for known whitelisted name servers is treated as a greylisted name server by the circular pool, and a DNS request from this name server is expected on a different set of interfaces. Naturally, an incoming DNS request needs to be ingress filtered at the network in order to eliminate spoofing in the DNS requests. This prevents an attacker from sending a forged DNS request with source address set to one of the whitelisted name servers. The circular pool deployment models are discussed in section 6.2.6.

A DNS server is moved between the name server categories following DNS connection failures or success rate in the circular pool. DNS connection failure refers to the case when a ‘waiting’ connection state expires in the circular pool because no data packet is received at the circular pool address, reserved in response to a received DNS query. If the connection success rate for a whitelisted name server falls below the threshold ‘R_w’, this indicates that a DDoS attack is in progress using DNS spoofing. This calls for strict (or aggressive) ingress filtering at the network

for the received DNS queries, i.e. drop all the DNS requests received with source address set to a name server residing within the network.

Similarly, for a greylisted name server, if the connection success rate falls below the threshold 'R_{g0}', the circular pool drops a certain percentage of the DNS requests from the name server. However, if the connection success rate falls beyond the threshold 'R_{g1}' the name server is blacklisted for duration 'T_o'.

A DNS server is demoted from its original category to a lower category only for a certain time duration 'T_o', after which it is restored to its original category. For example, a greylisted name server demoted to the blacklisted category is restored to greylisted status after it has served the time-penalty in the demoted category. While blacklisted, a DNS request from the name server is not processed by the circular pool.

6.2.2 System Load, Source Load and Domain Load

Source load: This parameter indicates the number of circular pool connections in the 'waiting' state for a particular name server.
Destination load: This parameter measures the number of circular pool connections in the 'waiting' state for a particular domain behind the circular pool.
System load: This parameter measures overall circular pool load by computing the percentage of CPOOL addresses in the 'waiting' state out of the whole pool.

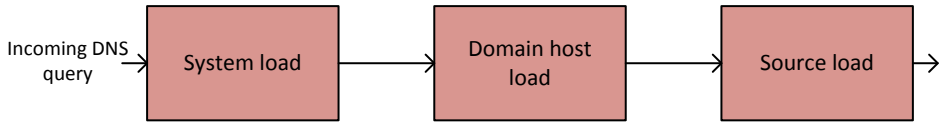


Figure 6.11 Processing a DNS query in Circular pool

Figure 6.11 describes how the circular pool utilizes these parameters to decide the fate of an incoming DNS request. The circular pool first determines the system load parameter to determine if the system can accept an incoming connection request. If the system is fully loaded i.e. all the CPOOL addresses are reserved, the DNS request is dropped. However, if the system load is below the maximum value, the circular pool determines the 'destination load' parameter. The DNS response is denied if the destination load is greater than the maximum load allowed for

the destination. The upper limit on the destination load parameter prevents the circular pool model from the DoS attack presented in Figure 5.10.

Next, a similar check is executed on the 'source load' parameter, and the DNS response is denied if the parameter value is greater than the maximum load allowed to the name server. The upper limit on the source load parameter prevents an attacker from launching the DoS attack presented in Figure 5.9, where an attacker floods the circular pool with DNS queries from a single name server.

6.2.3 CPOOL Address Allocation

To prevent an attacker from launching the DDoS attack, presented in section 5.2.6, the circular pool defines an address allocation criterion presented in Figure 6.12. Given that the source and destination load parameters are below their maximum values, a circular pool accepts an inbound DNS request from a name server only if it fulfills the criteria presented in the figure below,

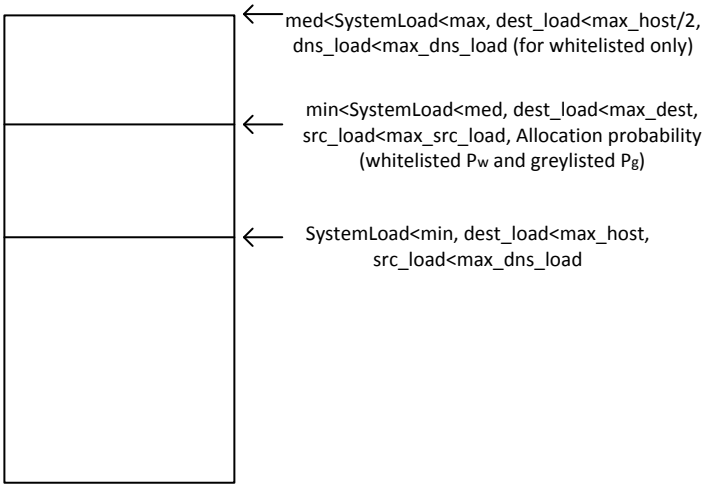


Figure 6.12 CPOOL address allocation policy

A connection request is accepted from either a whitelisted or a greylisted name server, if the system load is below the minimum load threshold of the circular pool. However, if the system load is between the 'minimum' and the 'medium' threshold values, the DNS request from a whitelisted name server is accepted with a probability of P_w (e.g. 0.7) for CPOOL address allocation, and a request from a greylisted name server is processed with a probability P_g (e.g. 0.3). But, if the system load exceeds the 'medium' threshold but lies below the 'maximum'

threshold, then only the DNS requests from a whitelisted name server are processed by the circular pool. A DNS response is denied for any other load conditions of the circular pool.

With this CPOOL address allocation, even under DoS conditions, a whitelisted name server can reserve the circular pool resources and the domains behind the circular pool can be accessed by a whitelisted source. However, a whitelisted name server has to compete with greylisted name servers for circular pool resources, when the system load is below the medium threshold. A DoS attack from hosts in the public Internet can only affect the CPOOL resources below this threshold, after which, the circular pool only accepts the connection requests from a whitelisted name server. Therefore, the allocation model guarantees that a paying or a trusted name server always has certain resources available in the circular pool.

6.2.4 Security Model

The obvious benefit of classifying DNS servers to white/grey/blacklisted name servers is that the circular pool is protected from the DDoS attack, presented in Figure 5.11. Even if an attacker employs multiple name servers from the public Internet to launch the DDoS attack, it will still not be able to reserve all the CPOOL addresses. The fact that the DNS servers in the public Internet are classified as ‘greylisted’ prevents the circular pool from reserving all the addresses, and hence limits the DoS attack to a portion of the circular pool.

With this security model deployed, a circular pool can be placed in the blocking state only if multiple whitelisted name servers are used to launch the DDoS attack. However, the probability of such an attack is far less given the fact that a whitelisted name server is not accessible to an attacker in the public Internet. But, for this model to work, the querying network (QN) needs to eliminate the DNS spoofing by ingress filtering the received DNS requests. A DoS attack originating from a whitelisted DNS can be reported to the network operator, and better security heuristics on the egress traffic are expected from the operator to reduce such attacks in the future.

Figure 6.13 presents the circular pool security model, where the upper half of model decides the fate of an incoming connection request based on System load, Source load and Destination load parameters. This protects the circular pool against the DoS attacks described in section 5.2.4 and section 5.2.5. A connection request from a blacklisted name server is dropped.

The lower half of the model represents the CPOOL address allocation criteria for a received DNS request, defined in section 6.2.3. The circular pool issues the DNS response only if the

received DNS request fulfills the CPOOL allocation criterion. A failure at any point in this model denies the response to the received DNS request.

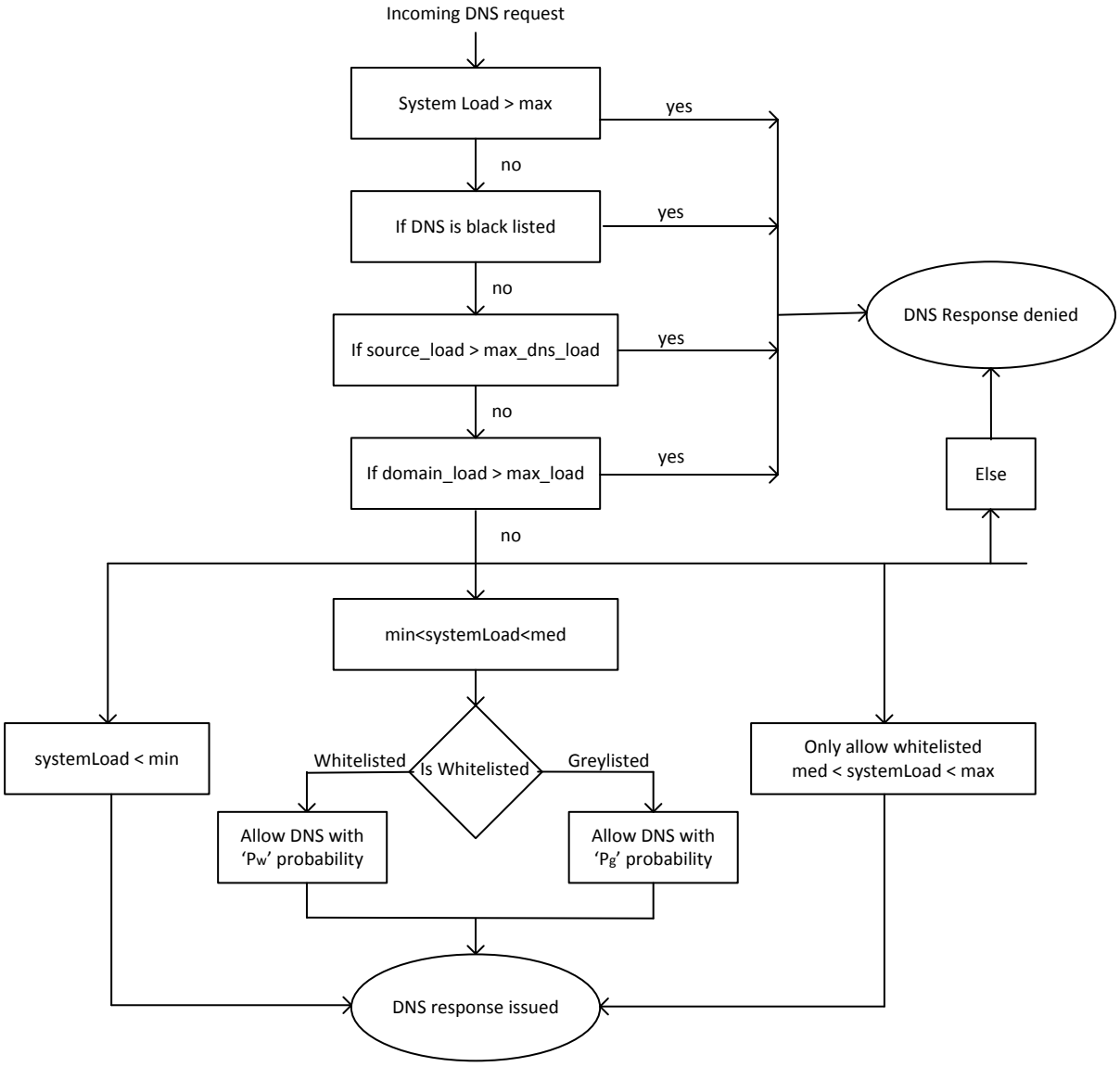


Figure 6.13 Circular Pool Security Model

6.2.5 Preventing Connection-Hijacking

Section 5.2.3 describes the vulnerability where an attacker host continuously sending IP packets to circular pool address space hijacks a ‘waiting’ connection state in the circular pool. This results in denial of service to a legitimate sender, as its IP packets are dropped by the CES for having no corresponding ‘waiting’ state in the circular pool.

To prevent connection-hijacking, the default policy of the circular pool model is to drop all the UDP flows initiating a communication with a server behind the circular pool. The setting up of a UDP or similar flow to a server behind the CPOOL without some prior signaling, i.e. SIP leaves room for an attacker who can take over states in the circular pool by sending spoofed UDP packets.

However, for a received TCP SYN segment, the circular pool can adopt either of three approaches to prevent connection hijacking: Bot-detection method, TCP-Relay method or Filtering/logging method.

Bot-detection Method: The circular pool model can attempt to verify the legitimacy of the sender host. An attacker aiming at hijacking a connection would send multiple spoofed TCP SYN segments towards the circular pool address space. The circular pool can detect such an attack by keeping a count of TCP SYN segments received without a prior mapping in the circular pool. For each TCP SYN packet received without a connection state, the circular pool drops the packet and logs a tuple of information: sender IP, destination IP, source Port, destination port, protocol used and the timestamp of packet arrival.

As a countermeasure, once a sufficient number of packets from the same source address have been received without a prior mapping, the circular pool can send a fake TCP SYN/ACK segment towards the sender with an ISN (Initial Sequence Number) computed according to TCP SYN-cookie mechanism [38]. If the response from the sender carries an ACK segment bearing the ISN+1 value, where ISN was sent in the SYN/ACK segment towards the sender, the sender is determined as a non-spoofing host. Based on the non-spoofing check and the history of the packets dropped from the host address, we classify the sender as a bot-controlled legacy host. Following which, the sender is blacklisted for time duration ‘T_O’, during which an IP packet from this source is not accepted for claiming a ‘waiting’ connection state in the circular pool.

However, if no TCP ACK packet with expected ISN is received for multiple fake SYN/ACK segments sent in the direction of the attacker, the circular pool identifies a spoofing attack in

action and generates the attack alarm in the circular pool. When a DoS attack is detected, the circular pool can benefit from the CES deployment model and accept the SYN segments coming from a whitelisted source only.

TCP-Relay Method: In this method, for each TCP SYN segment received, the circular pool should send a TCP SYN/ACK packet towards the sender with an ISN computed according to TCP SYN-cookie mechanism. For a non-spoofing host, an ACK segment with expected ISN shall be received, whereas a spoofing source never answers with the expected ISN in the ACK segment. The mechanism provides good defense against spoofing attacks, but fails to detect connection hijacking attempts from a bot-controlled non-spoofing legacy host. Another drawback of the mechanism is that it extends the blocking state duration in the circular pool, as TCP handshake has to be completed before the *waiting* state can be assigned to the sender.

Logging/Filtering Method: This is the simplest approach of all protection mechanism. It assumes that spoofing is eliminated in the inbound traffic because of underlying infrastructure. A suitable place to deploy this mechanism is in customer-CES which can assume that spoofing in the received segments has been eliminated by carrier grade CES, e.g. by TCP-Relay mechanism.

The method relies on a logging/filtering approach on the inbound traffic to spot a bot-controlled legacy host. In this approach, the circular pool logs the source IP address of the packets that are dropped by the circular pool for having no corresponding 'waiting' state, as they are deemed as connection hijacking packets from an attacker. If the number of such dropped packets from a source exceed a threshold of 'x' in time duration 'T_O', the sender is believed as bot-controlled host and the IP address is marked as blacklisted.

However, realizing the possibility of false positives and false negatives in attacker detection process, an IP source is only blacklisted for time duration 'T_1', and not permanently. While blacklisted, an IP packet from the source is not accepted for claiming a waiting state in the circular pool.

Implemented Solution

Because of the time constraints in this thesis, I implemented the Bot-detection method and Logging/Filtering method to prevent connection hijacking attempts in the circular pool. These schemes have been evaluated under different test cases and results are presented in section 7.4.

6.2.6 PRGW Deployment Model

Security mechanisms defined to protect the circular pool against attacks can benefit from a CES/PRGW deployment model. In this thesis, we define two deployment models to further strengthen the security of circular pool design: Multiple Interface model and Customer CES/ Carrier Grade CES (C3G-CES) model.

In the multiple interface model, PRGW defines a distinct set of interfaces to receive the whitelisted traffic, while the traffic from greylisted sources is expected over separate interfaces. The model assumes that whitelisted sources are specifically known to the PRGW and only connection requests from these sources are expected over the whitelisted interface. The network must aggressively filter the connection requests received over a greylisted interface, i.e. drop the packets received with a spoofed whitelisted address in order to eliminate spoofing in the received requests. Under DDoS attack conditions, the model may drop the connection requests received over a greylisted interface and hence guarantees that domains behind the CPOOL always remains accessible to a whitelisted source.

Figure 6.14 presents a multiple interface based CES/PRGW model. The model is suitable for a network like mobile operator network, where the network has a set of interfaces available for receiving inbound connection requests. The operator can choose to receive an inbound DNS request from another mobile operator (a whitelisted source) through a whitelisted name server, whereas the hosts in the public Internet can reach the operator's network via greylisted name servers in the Internet.

In such a model, a different set of protection mechanisms can be employed at each interface. This is also supported by the fact that mobile networks are much cleaner than the public Internet and therefore relatively lax filtering can be applied to connection requests from mobile networks. Whereas, the connection requests from the public Internet should go through strict filtering before they are accepted by PRGW.

The deployment model can be tailored by mobile operators according to their needs, or based on the profile of the Internet traffic received. And according to inter-operator agreements, mobile operators can also choose to cooperate and run a trust alliance much larger in size, to handle DDoS attacks targeted at PRGWs. In such a case, mobile operators can decide to receive connection requests from each other over a dedicated set of interfaces. Hence, under attack conditions, mobile operators can choose to accept connection requests from each other and drop the connection requests from other sources, to reduce the impact of a DDoS attack.

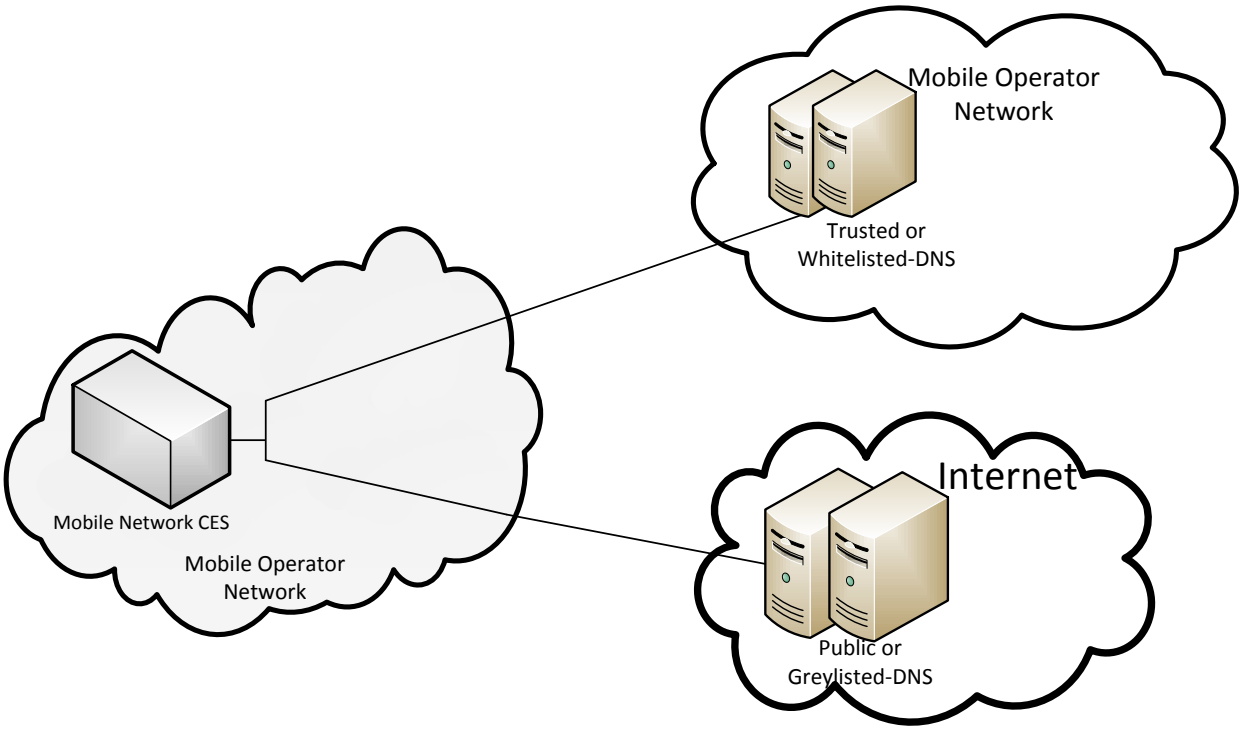


Figure 6.14 CES/PRGW deployment for a PRGW with multiple interfaces

The deployment model presented above ensures the security in the circular pool model because of multiple interfaces available with the network to receive the inbound traffic. However, for a CES device with a single interface, the model does not offer much security. In such a case, the CES/PRGW can be secured following a hierarchical approach, where the carrier CES box can attempt to protect the customer-CES against DDoS attacks and the illegitimate access. A carrier grade CES can have multiple interfaces for receiving the inbound traffic and can ensure high reliability and availability of the CES resources using a variety of security mechanisms, e.g. TCP-Relay method to eliminate spoofing in the received traffic, DNS tracer: to trace DNS requests back to the original sender, reporting a malicious source (attacker) to the trust reputation system etc. Hence, the carrier CES can reduce the attack volume received in the traffic forwarded towards a single interface customer CES device.

6.3 CES Security Semantics

This section explains CES security semantics in a mobile environment. Figure 6.15 presents a scenario where Host-A and Host-B move away from their home networks into visited networks, Visited CES-A' and Visited CES-B'.

When a host roams into a visited network, it issues an attachment request to register itself in the visited network. The attachment is complete, after MME has successfully authenticated the host using the subscriber information in the HSS of the home network of the host. Next, the subscriber information in the MME and the HSS is updated according to the current location of the host. Following the attachment, Host-A sets up a tunnel to the PDN gateway of the home network and thus to CES-A.

For a connection establishment between the roaming hosts, Host-A and Host-B, the DNS request for the destination returns the RLOCs of home network of the host i.e. CES-B. The connection is successfully established between home CES devices, after performing necessary verification queries with the respective HSS. After connection establishment, the data packets exchanged are tunneled from home CES to the visited network of the host, based on the location update information from the HSS.

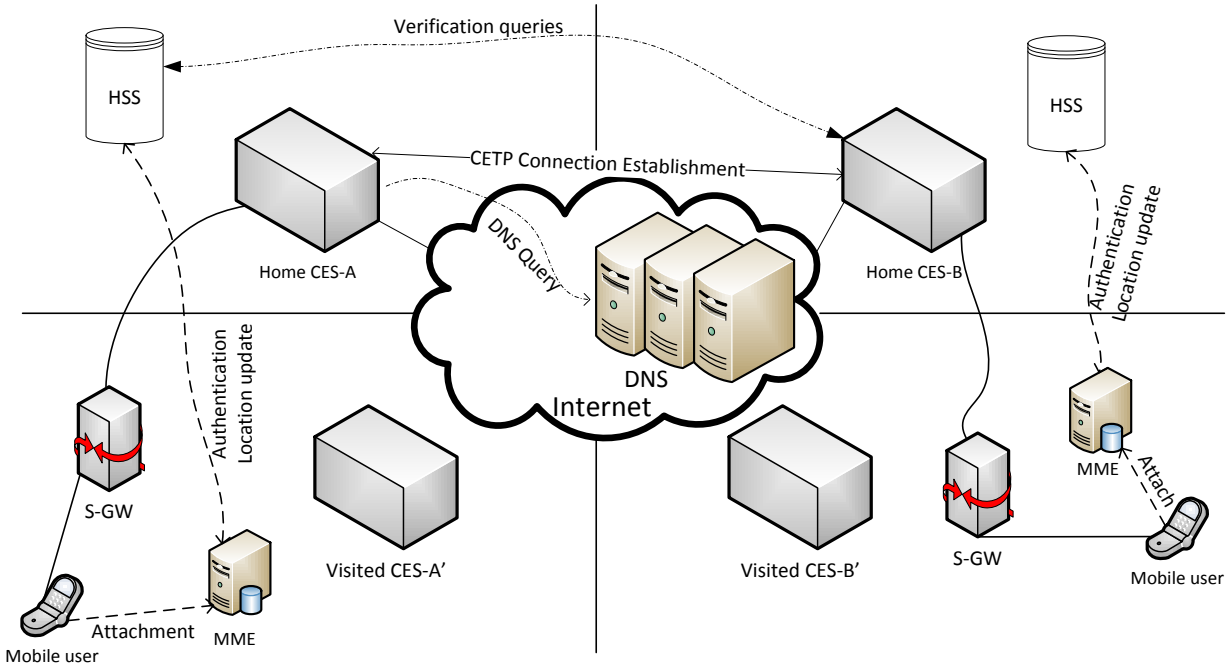


Figure 6.15 Mobility in CES enabled mobile networks

In the second model, presented in Figure 6.16, after the location information has been updated in the MME of the visited network and HSS of the home network, the location update is reflected in the DNS name space using Dynamic DNS. Hence, the DNS record against the domain of the host corresponds to the CES-ID and RLOC of the visited network, i.e. CES-B'. For a connection establishment between roaming hosts, the visited network performs the DNS query to determine the RLOC of the current CES hosting the destination. After which, the connection is established between the visited CES devices and the data is exchanged between the roaming hosts. Alternatively, instead of Dynamic DNS, reachability through visited CES nodes can be handled completely on an application layer, i.e. as done in IP Multimedia Subsystem (IMS).

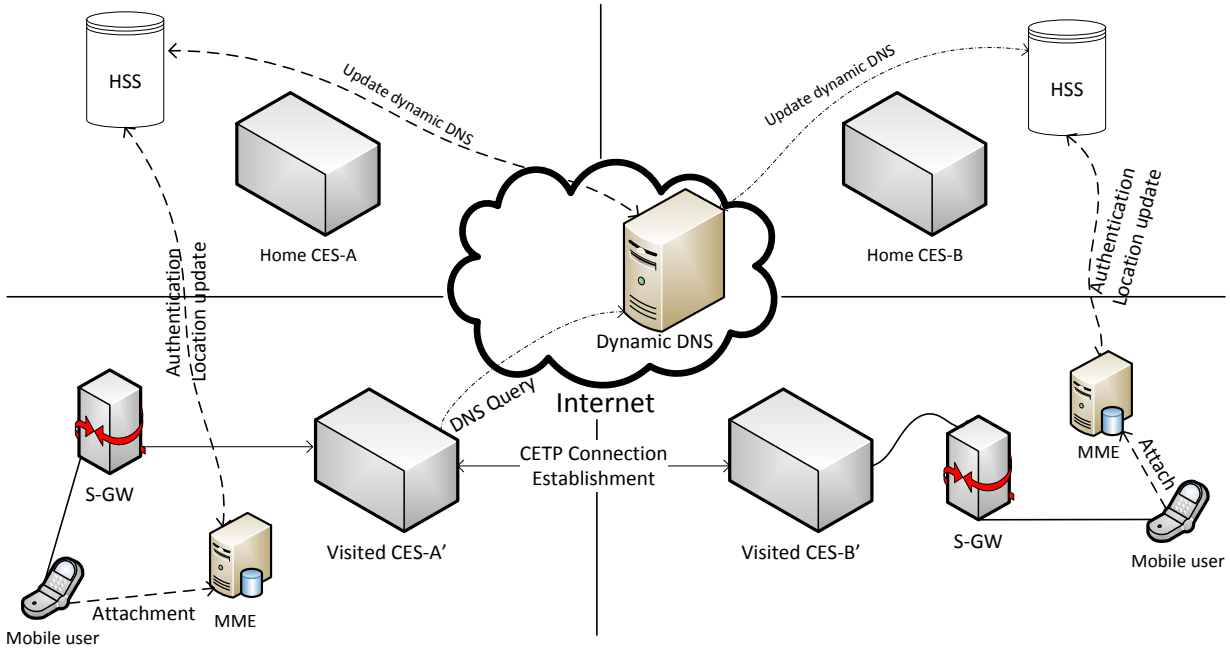


Figure 6.16 Mobility in CES enabled network (VOIP model)

The choice of a particular mobility model can depend on the service APN. For example, the first model suits the case when mobile devices require access to the Internet, because mobile operators charge volume based pricing, and therefore they need to count the traffic volume. However, for the case when billing is at application layer, e.g. IMS, the mobile networks can support roaming through visited CES device. This leads to shorter end-to-end delay, which is a required in VOIP services.

7. Evaluation

The chapter introduces the prototype network developed as CES proof-of-concept. Next, the chapter analyzes the security mechanisms added to secure the CES architecture, following a comprehensive set of test scenarios. A comparison of test results from before and after the security indicates the effectiveness of the security mechanisms. Finally, the security of the system is evaluated in terms of its efficiency and a performance analysis of the security is presented.

7.1 CES prototype Network

Figure 7.1 presents the CES prototype network, simulated on a PC running a Linux/Debian operating system. The setup uses a virtualization solution KVM (Kernel-based Virtual Machine) to run various virtual machines on a single PC such that each virtual machine runs a Linux operating system and has a virtualized hardware i.e. network card. To run source codes and test scripts on these machines, SSH connections are established with each virtual machine. The prototype network consists of two CES devices: CES-A and CES-B, to simulate the sending and the receiving ends of CES-to-CES communication. The network also contains a machine "Host-Public" to simulate a legacy IP source and to demonstrate the backward compatibility of CES with legacy networks.

The prototype network consists of two private networks behind the CES devices, and each network consists of a set of hosts. To facilitate communication between hosts, an authoritative name server is provided for the network. The machine running the authoritative DNS server also hosts a Certificate Authority (CA) and a Home Subscriber Server (HSS) to provide the security services for the CES architecture.

7.2 Libraries Used

CES prototype has been developed in Python with support from multiple python libraries. This thesis utilizes following python libraries to develop and test the security mechanisms of CES,

DNSPython is a Python based DNS toolkit, which supports various DNS messages and record types. The prototype uses this library for generating DNS queries, responses, and testing the circular pool security.

M2Crypto is a python library that provides wrapper functionality for OpenSSL. It supports a wide variety of cryptographic functions i.e. RSA, DSA, message digests, symmetric/asymmetric ciphers and SSL functionality. In this thesis, generating public/private keys, implementing a certification authority and CES registration process were accomplished using this library.

MySQLdb is a python library, which is used as an interface to a MySQL database server.

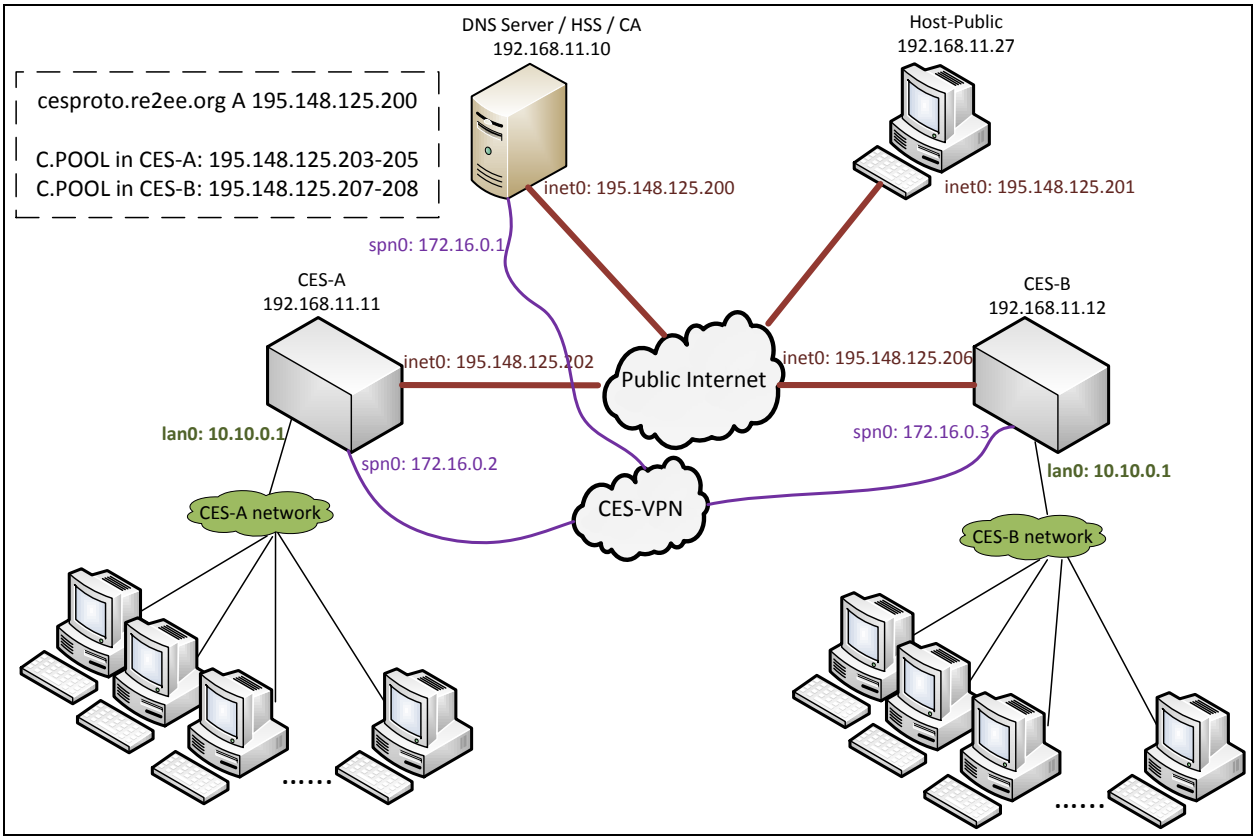


Figure 7.1 CES prototype network

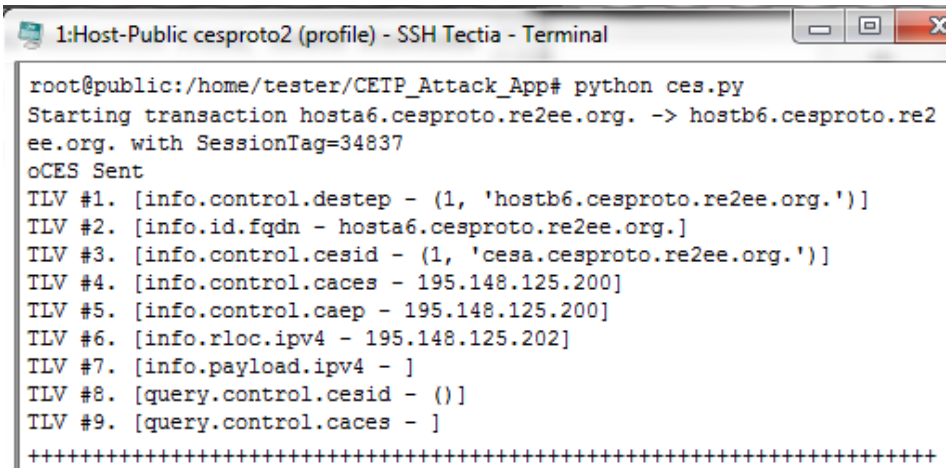
7.3 Testing the Security of CES-to-CES Communication

This section evaluates the security mechanisms added to secure the CES architecture against vulnerabilities present in the CES-to-CES communication. The section submits different test cases for analysis that explore the effectiveness of the security measures by comparing the results before and after deploying the security model.

7.3.1 Testing CES Security against Spoofing Sources

As discussed before, CES devices may share the same VPN as the legacy hosts in the wake of the depleting IPv4 address space. In such a case, the attack traffic generated from a legacy host, due to attack software or a bot, can affect a CES in different ways. This test case demonstrates a scenario where a legacy host spoofs the RLOCs of a legitimate CES to launch an attack.

Figure 7.2 shows that the legacy host 'Host-Public' spoofs the RLOCs of CES-A and sends a CETP packet towards CES-B (the iCES). Upon successfully fulfilling the destination policy, CES-B opens a connection for subsequent packets from CES-A. Meanwhile, the CETP response packet sent in the direction of CES-A fails to trigger a session and is therefore dropped by oCES.



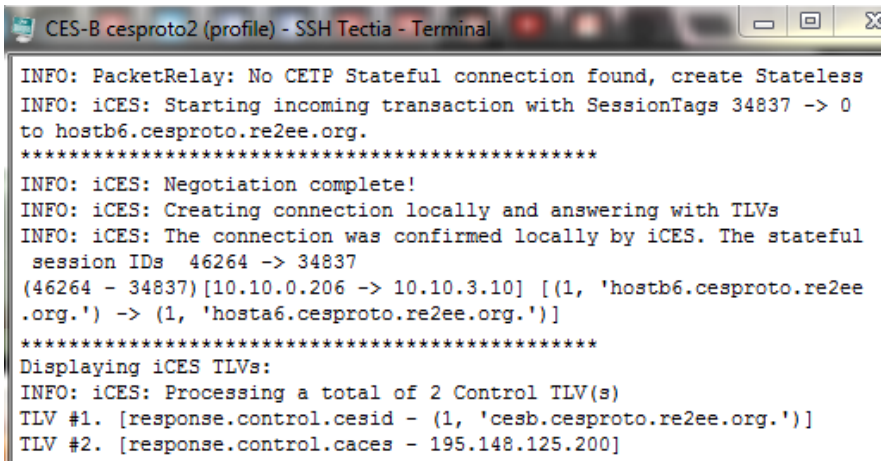


Figure 7.2 Legacy host spoofs CES-A RLOCs to establish a connection in iCES, before security

However, the cookie mechanism introduced in the CES architecture prevents an iCES from opening a connection upon receiving a CETP packet, unless the sender is validated as a non-spoofing source. Figure 7.3 shows the result of the same test case, when the cookie mechanism has been added to eliminate spoofing in the CES-to-CES communication model. In contrast to the earlier results, the spoofed CETP packet is unable to open a connection in CES-B. Rather, the iCES responds with a cookie-TLV along with other required policy elements to the sender. For the connection to establish, the same cookie should be received in the next inbound packet from the sender, which confirms that the sender is a non-spoofing source. Since such a cookie is never received in this test case, a spoofing attack is detected and the attack is stopped at the iCES.

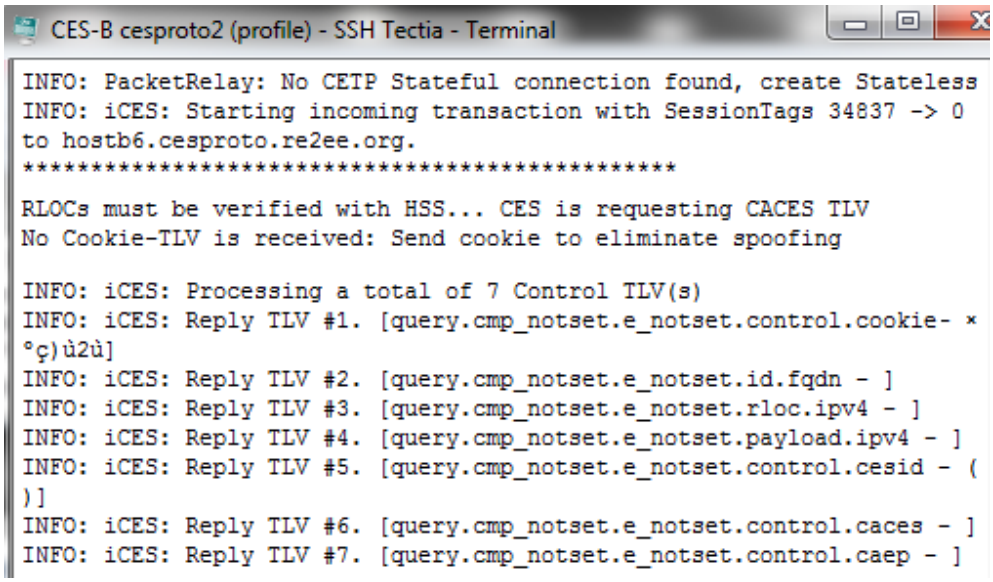


Figure 7.3 Legacy host spoofing CES-A RLOCs fails to reserve a connection in iCES, after security

7.3.2 Testing CES Security with Non-spoofing Legacy Host

Since the CES devices may share the same VPN as legacy hosts, the attack traffic generated from a non-spoofing legacy host can defeat the cookie based protection mechanism. Figure 7.4 presents the scenario where a non-spoofing legacy host imitates a legitimate CES, i.e. CES-A, and sends CETP packets towards CES-B. In the absence of a CES verification mechanism, CES-B processes the received CETP packet and establishes the connection with the legacy host, due to the destination policy fulfillment.

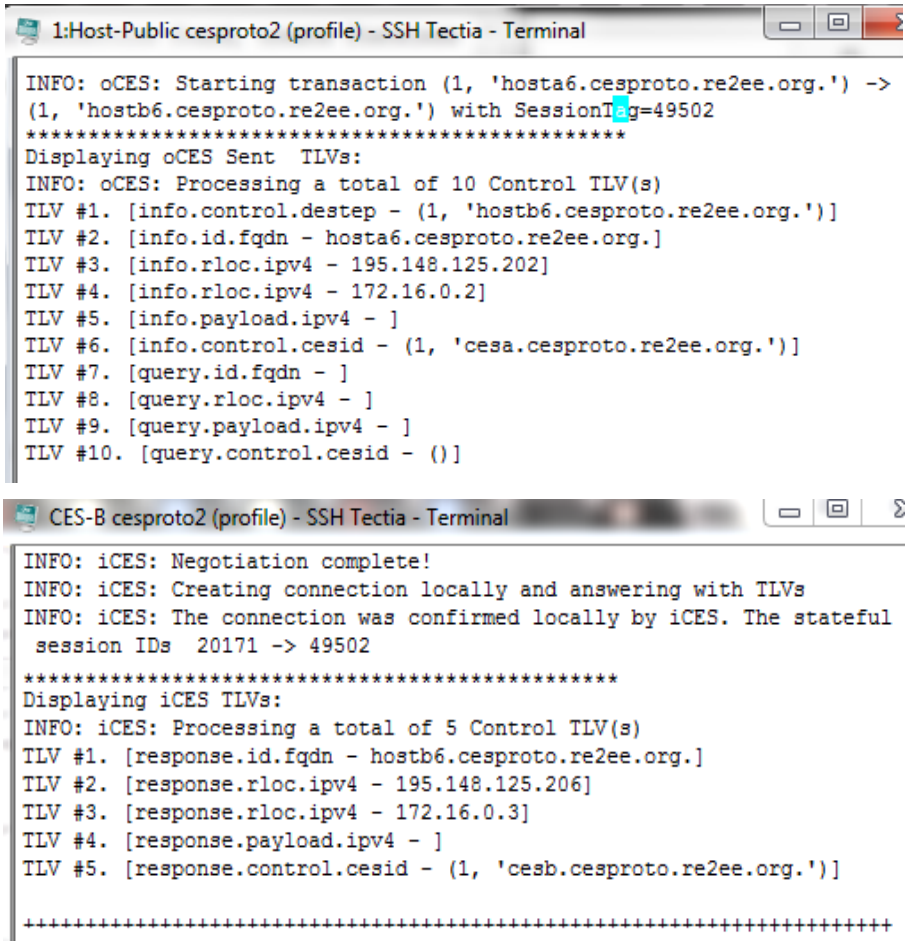


Figure 7.4 Connection establishment in iCES with a legacy host, prior to security

However, when a CES verification mechanism is in place, i.e. HSS, the CETP packet from the legacy host is dropped by CES-B (or iCES) as it fails the CES verification check. Figure 7.5 shows that the CETP packet received from the legacy host with SST=27974 is dropped for failing the CES verification process. Once RLOCs have failed the CES verification process, the RLOCs are blacklisted for time duration ‘To’ and subsequent packets from these RLOCs are dropped by CES, without re-performing the HSS verification.

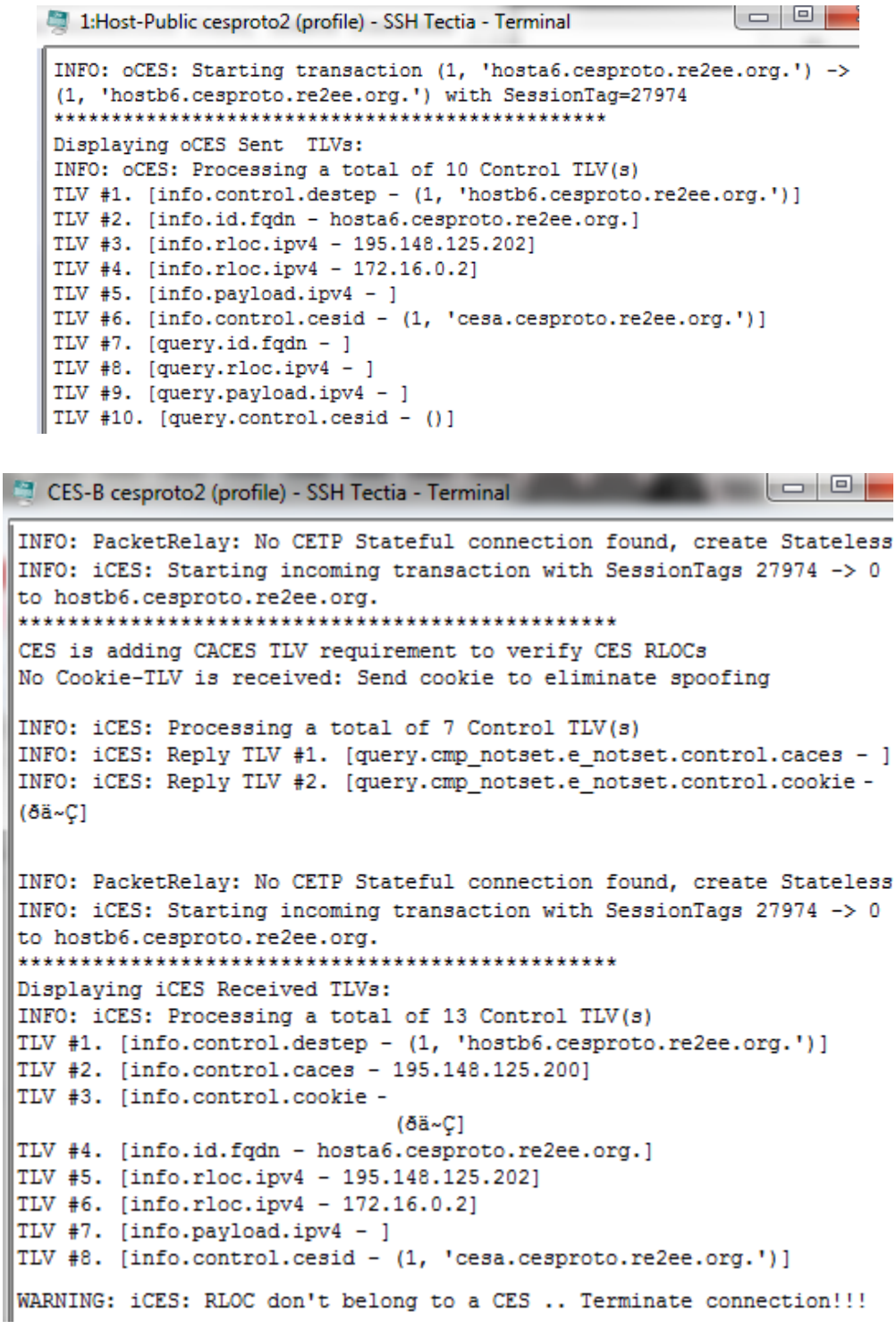


Figure 7.5 Legacy host initiating CETP connection establishment fails, after the security

Alternatively, we can make it impossible for hosts to send CETP traffic towards CES, e.g. by expecting the traffic from legacy hosts on a separate interface.

7.3.3 Testing the Cookie Mechanism

Following the cookie mechanism, a connection is established in the iCES only after the cookie sent towards the sender is received in the next inbound packet from the sender, which confirms that the sender is a non-spoofing source. An attacker can attempt to thwart the cookie mechanism by forging a cookie or by replaying a cookie received earlier from the victim. Figure 7.6 presents the test case where a legacy host imitating as a legitimate CES sends a forged cookie towards the iCES, i.e. CES-B. Since, the cookie received could not be verified by the cookie verification algorithm presented in section 6.1.2, the iCES drops the received packet.

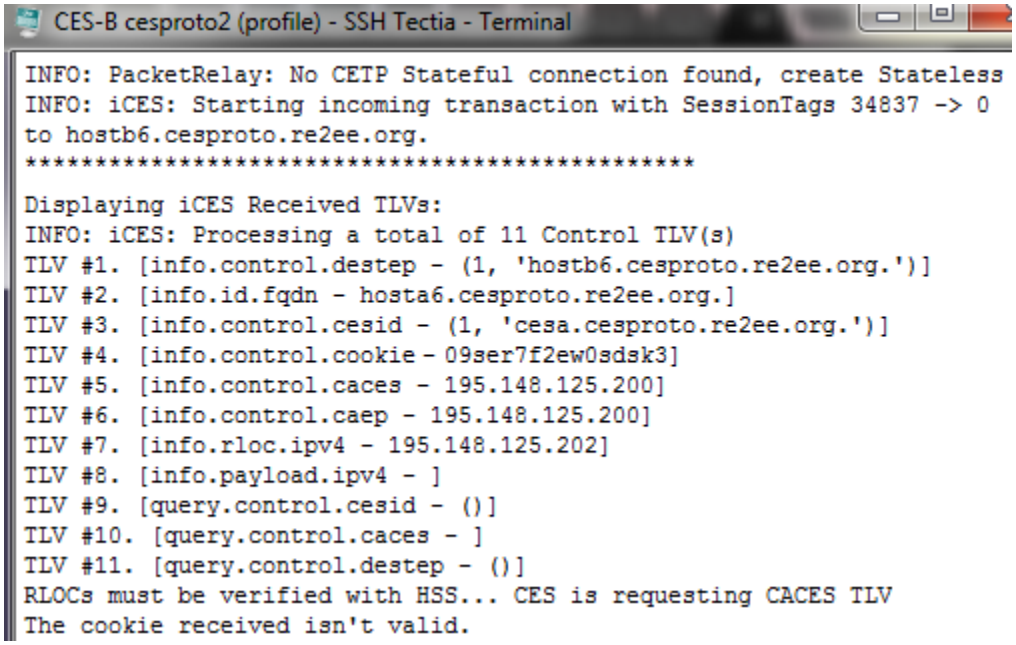


Figure 7.6 CES detects and drops the CETP packet with forged cookie

Figure 7.7 presents the case where the iCES drops a replayed CETP packet. A replayed packet is detected by the iCES because of the timeout value present in the cookie computation/verification algorithm.

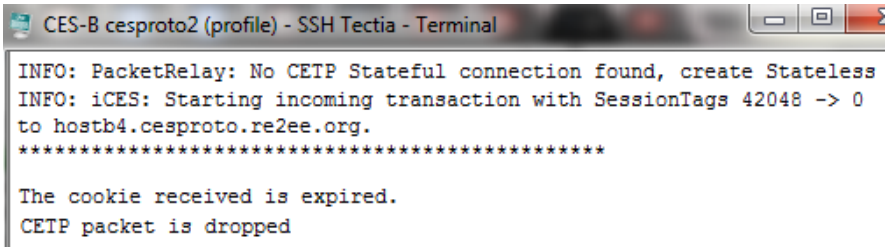
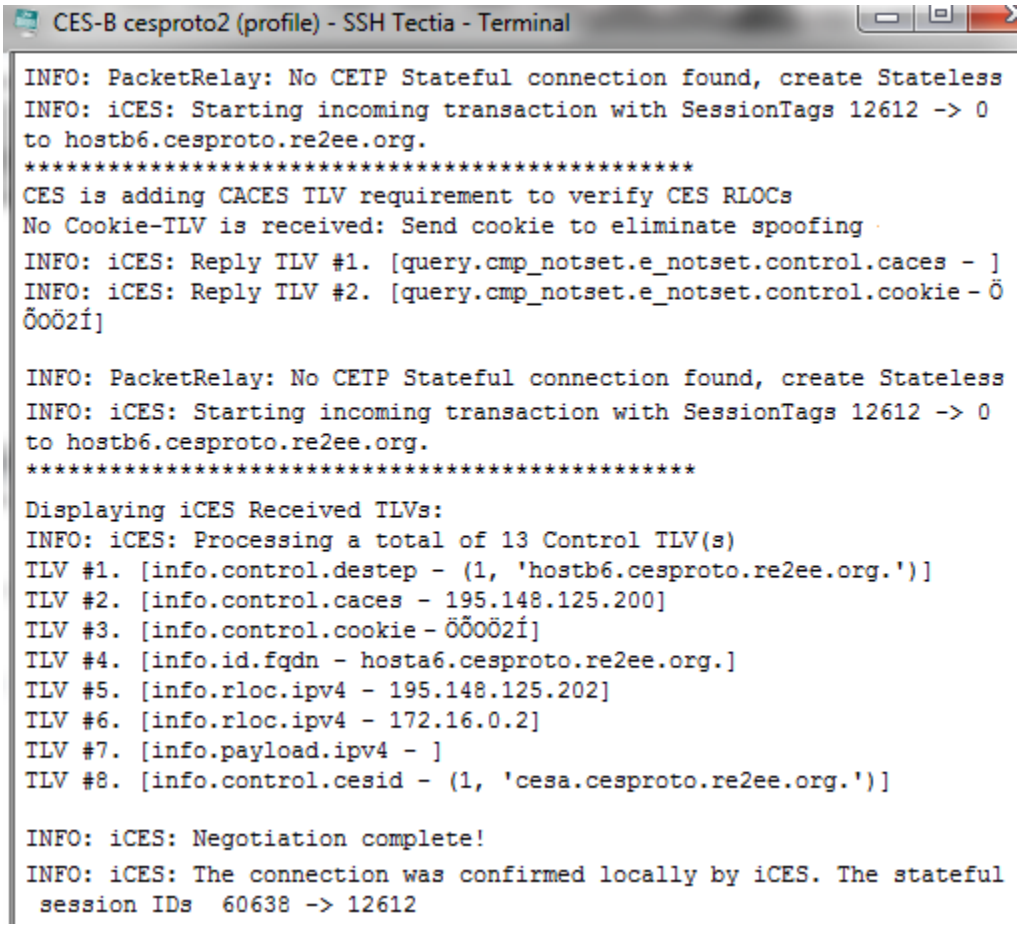


Figure 7.7 CES detects and drops a replayed CETP packet

7.3.4 Testing CES Registration/Verification Mechanism

The CES architecture employs a CES registration/verification mechanism to determine if the source of a received CETP packet is a legitimate CES. For the first packet received from a source, the iCES responds with a cookie to eliminate spoofing on the received packet and requests for CACES-TLV from the sender. Figure 7.8 presents the result of the CES verification mechanism, where the same cookie returned in the next inbound packet guarantees that the sender is not a spoofing source. Next, the iCES connects with HSS at the address received in the CACES-TLV to determine if the sender is indeed a registered CES device. After verification, the iCES responds with the requested policy elements and the connection is successfully completed with CES-A.



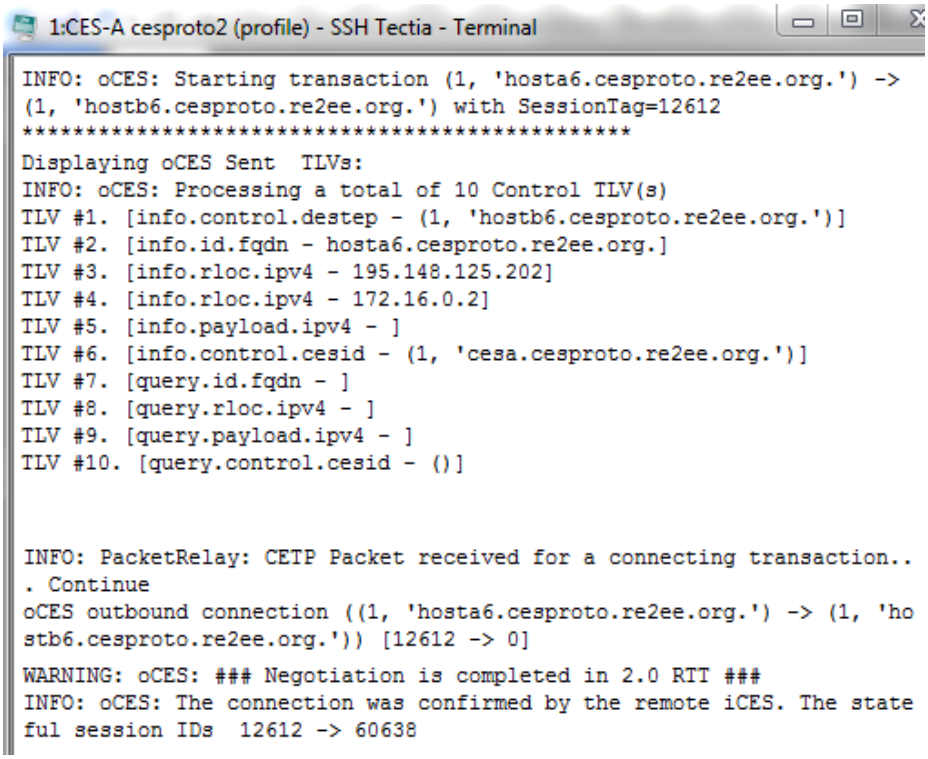


Figure 7.8 CES validation mechanism on 1st CETP packet received

Once verified, the sender is logged as a registered CES device and subsequent packets from the CES are accepted without re-performing the CES verification process, for time 'Tx'.

7.3.5 Updated CETP Connection Establishment

As a consequence of the CETP security model, a CETP connection is always established in 2RTT. The connection establishment in 1RTT is not possible, as the first CETP packet receives a cookie in the response, which must be relayed in the subsequent packet from the sender for the connection to establish in the iCES. Figure 7.9 presents the connection establishment process after security, using the same policies as listed for connection establishment in 1RTT, prior to the CETP security model. The policies used in this case are,

***Outbound policy of Host-A*:**

Required:	Id.fqdn,	rloc.ipv4,	payload.ipv4	control.cesid
Offer:	Id.fqdn,	rloc.ipv4,	payload.ipv4,	control.cesid
Available:	Id.fqdn,	rloc.ipv4,	payload.ipv4,	control.cesid

***Inbound policy of Host-B*:**
Required: Id.fqdn, rloc.ipv4, payload.ipv4, control.cesid
Offer:
Available: Id.fqdn, rloc.ipv4, payload.ipv4, control.cesid, control.headersignature

During the connection establishment, the oCES encodes a CETP packet with Host-A policy and sends it towards the iCES, identified from the DNS response of the destination host. At the iCES, since the packet is received without a cookie, the iCES computes a cookie using the information received in the offered TLVs and encodes the CETP response packet. The response packet carries the computed cookie along with other policy requirements and it is sent towards the oCES with SST=0 and DST, same as the received SST.

Upon receiving the CETP packet, the oCES looks up for a connection state whose SST value matches the DST value received in the packet. For a matching state, the oCES accepts the incoming packet as a response to the connection request sent earlier. Besides responding to the requested policy elements, the presence of cookie TLV requires that the oCES must relay the cookie back to the iCES. The oCES encodes the CETP packet according to the received policy requirements and assigns it the same SST and DST values as for the first packet sent towards the iCES.

Upon receiving the CETP packet, the iCES checks for the presence of a cookie. If present, the authenticity of the received cookie is verified. The successful verification guarantees that the sender is not a spoofing source. Since the policy requirements of Host-B are fulfilled by the received “info” TLVs and the queried TLVs can be answered from Host-B policy, the iCES declares the connection establishment as “successful”. Following which, the iCES encodes a CETP response packet with all requested policy elements and sends it towards the oCES. The response packet bears a locally generated SST value, and the DST is set to the SST value received in the packet.

The CETP response packet at the oCES goes through the same operation as for the previous response packet. If the received packet successfully responds to all requested policy elements of the sending host, the connection establishment is successfully complete. Hence, the connection is established in 2 RTTs. The SST and DST values learned during the connection establishment are used to forward subsequent data packets between CES devices.

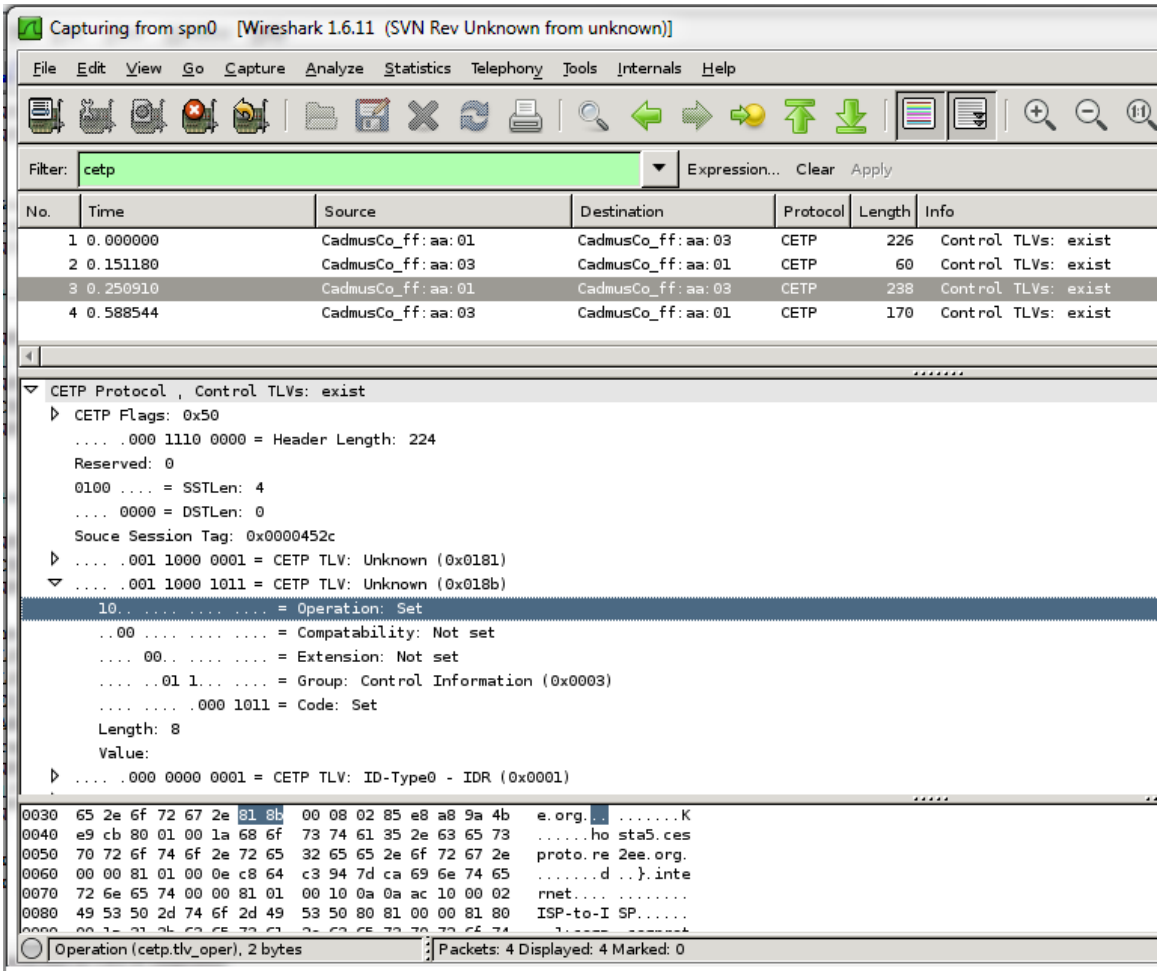


Figure 7.9 Wireshark capture of CETP connection establishment, after security

7.3.6 Performance Analysis

The CETP security model developed to secure CES against vulnerabilities present in its architecture lead to the addition of new processing modules in CES. This section analyzes the performance of the CES prototype after the deployment of the security model. The cost of the security is evaluated in terms of delay, by conducting approximately 80 CES-to-CES connection establishments and the results generated are described below.

Figure 7.10 presents a comparison of delay in CES-to-CES connection establishment prior to and after the deployment of CETP security model. The figure presents the connection setup delay when considering only the CETP part of connection establishment, i.e. without DNS. The connection establishment delay before the deployment of the security model is relatively less than the connection setup delay after the security. The rise and fall of connection setup delay before security indicates that a connection was successfully completed in either 1RTT or 2RTT.

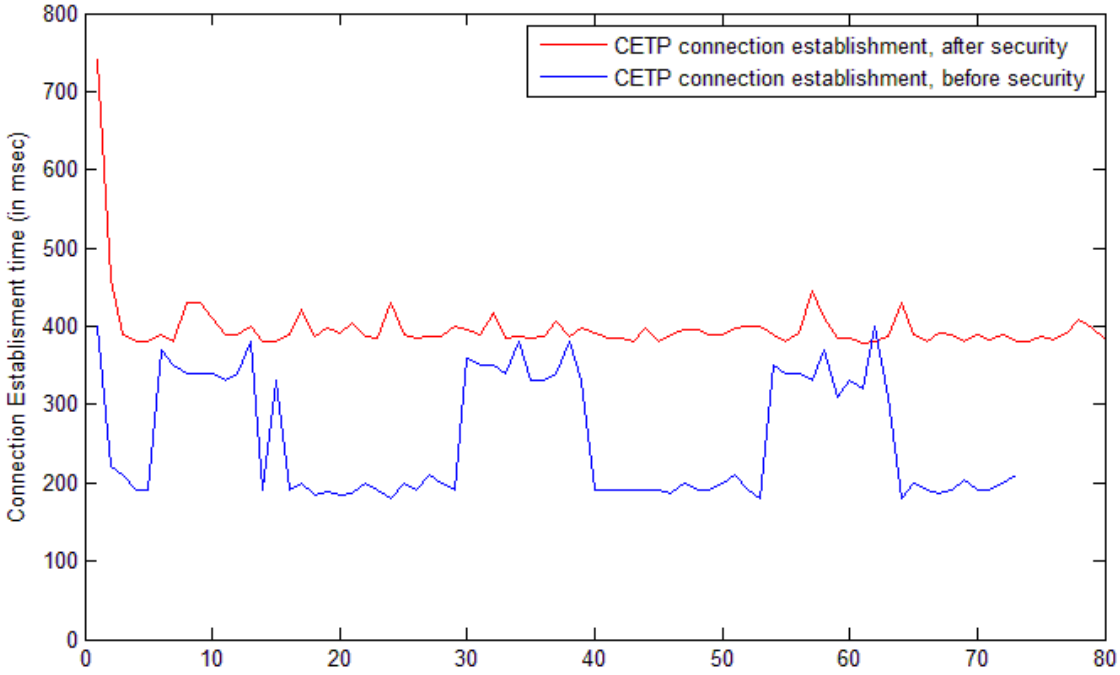


Figure 7.10 CETP connection establishment duration, before and after the security

The connection establishment delay after security is slightly higher than before, due to addition of new processing modules. However, the delay is relatively constant because all the connections are established in 2RTT, after the security. The first connection request from a CES suffers more

delay than the subsequent connections, as CES verification checks are carried out on the first inbound connection from the sender.

Figure 7.11 depicts the same phenomenon when including the DNS part in CES-to-CES connection setup delay. Table 7-1 presents the mean connection setup delay for a CES-to-CES communication before and after deploying the security, based upon the test results.

	Before Security (msec)	After Security (msec)
CETP connection delay – 1RTT	197.724	-
CETP connection delay – 2RTT	360.371	398.150
Connection delay (DNS included) – 1RTT	243.644	-
Connection delay (DNS included) – 2RTT	410.335	487.721

Table 7-1 Mean connection setup delay, before and after the security

The difference of about 40 milliseconds in connection establishment before and after the security is mainly because of the presence of HSS based ID verification process in oCES and iCES, for each connection establishment. The ID verification with HSS contributes around 16 milliseconds in delay, at each CES, since it involves opening a connection with HSS and validating the identity of the sender host. The rest of security modules introduce less than 3 milliseconds in terms of delay.

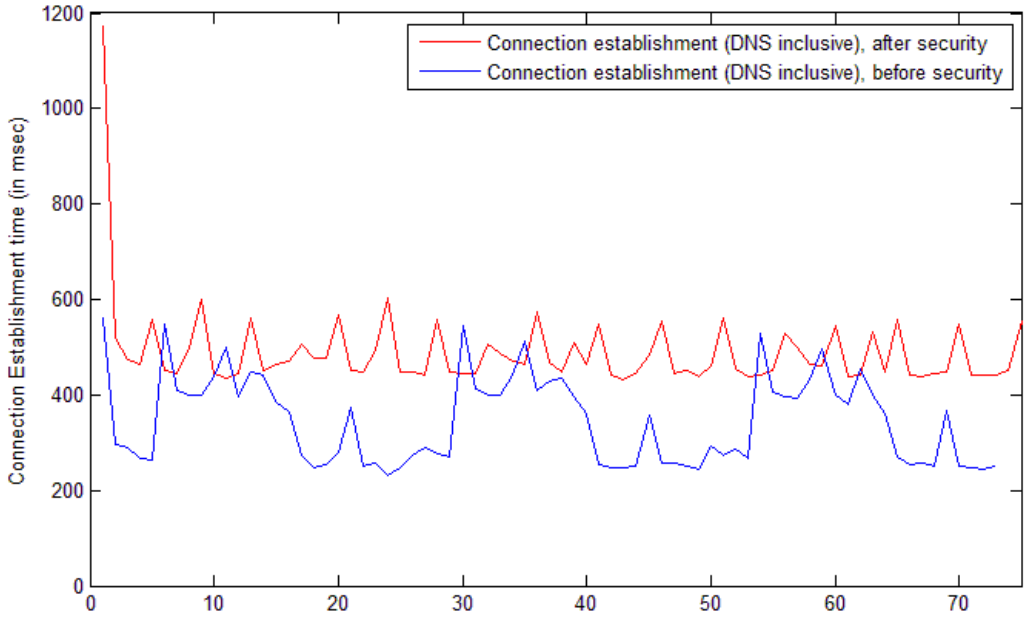


Figure 7.11 CES-to-CES connection establishment duration, before and after security

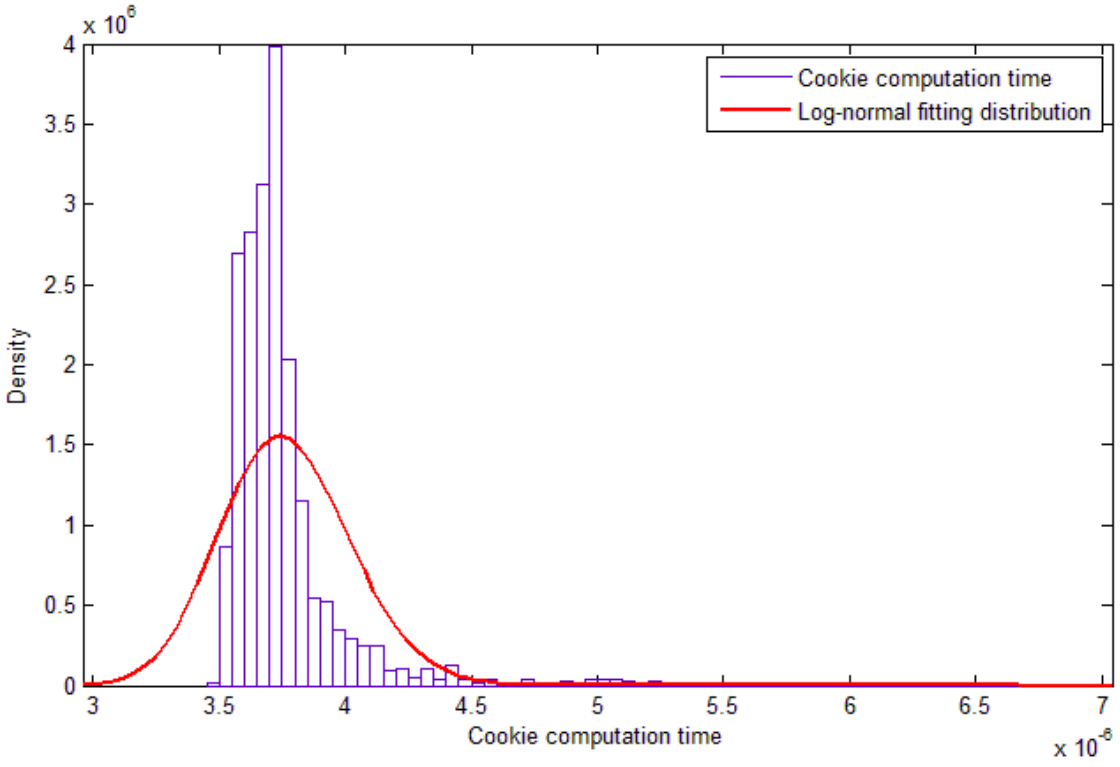


Figure 7.12 CETP cookie computation duration

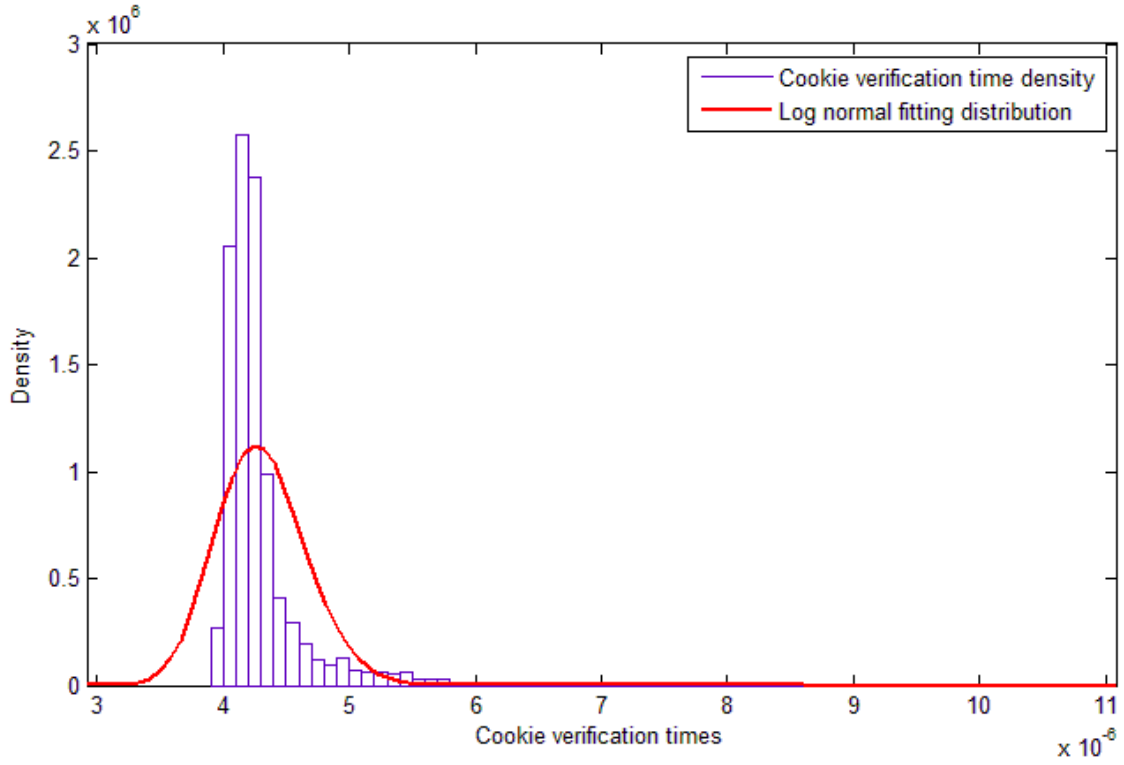


Figure 7.13 CETP cookie verification times

Figure 7.12 presents the summary of over a million trials conducted to generate a cookie using the CETP cookie computation algorithm. The time duration for cookie computation follows a log-normal distribution and has an average of 3.37 microseconds. This small delay makes CETP cookie an ideal mechanism to counter the spoofing attacks, or DoS attempts, without investing much into processing at the iCES. Figure 7.13 presents the time invested in verifying a received cookie using the CETP cookie verification algorithm. The mean verification time of 4.33 microseconds suggests that the cookie mechanism could provide a good light-weight defense against spoofing attacks.

As a further measure for performance analysis, Table 7-2 presents the time duration a CES node spends on a received attack packet before it is rejected by the CES. A CETP packet received with a forged cookie is detected and discarded by an iCES much earlier than other attack types. The first transaction from a legacy host imitating a legitimate CES takes the most time in the security model, 16 milliseconds with HSS based RLOC verification and 4 milliseconds with CETP header signature mechanism. However, once detected as a legacy host, the subsequent CETP packets originated from the source are discarded in less than a fraction of a millisecond. A (spoofed) CETP packet received without a cookie spends about 2 milliseconds in the iCES, which involves generating the cookie, encoding it and inserting it in the response packet.

	Processing at algorithm (msec)	Total duration in the security model (msec)
CETP packet with forged cookie	0.00433	1.83
CETP packet without cookie	0.00373	2.0
HSS - RLOC verification (first packet)	16	26
Signature - RLOC verification (1st packet)	4	15
RLOC verification(subsequent packets)	< 0.01	1

Table 7-2 iCES processing duration on a received attack packet

The table presents processing duration spent in the security model for an attack type at two levels: Algorithm level and security module level. The algorithm level value indicates the time consumed by the algorithm, whereas the time duration in security module python based pre-processing necessary to execute the algorithm, besides the algorithm level itself. Naturally, the total duration spent in the security model is larger than algorithm level, because of slower processing involved at python layer.

The performance of CES-to-CES communication can be evaluated in terms of delay budget for various host operating systems in use. Operating systems e.g. Unix or Windows, typically reattempt a DNS request 4 times, after each earlier sent request was not responded to within the

timeout duration. The DNS timeout duration defines the delay budget of the operating system for the CETP session setup to complete.

In CES-to-CES connection establishment, the DNS query timeout corresponds to the delay suffered by the DNS request plus the duration of 2RTT based CETP connection establishment. This can be modeled by the equation presented below

$$db \geq DNSrtt + 2 * CETPrtt$$

Since 1RTT equals two one-way edge-to-edge (e2e) delays.

$$db \geq 2 * DNSe2e + 4 * CETPe2e$$

Roughly assuming that in terms of delay, DNSe2e = CETP e2e

$$db \geq 6 * CETPe2e$$

The delay budget for a DNS request can range from 1 to 5 seconds depending on the operating system in use, Windows or Unix, respectively. This leaves us with one way edge-to-edge delay of 167 milliseconds for Windows, and 833 milliseconds for Unix. Table 7-1 indicates that a connection between CES devices is established in less than 490 milliseconds, when taking DNS into account. Because in our network of virtual machines hosted by a single PC, the edge-to-edge packet transfer delay is zero, we calculate the maximum allowable end-to-end packet transfer delay as following,

$$db \geq 6 * CETPe2e + 490\ msec$$

Taking delay budget values of different operating systems into account, this leaves N milliseconds for a packet to traverse on wire from CES to CES, where N corresponds to 85 milliseconds in Windows and 750 milliseconds in Unix. Pessimistically, we can adopt the end-to-end delay requirements by ITU-T in Recommendation G.114 i.e. 150 milliseconds one-way edge-to-edge delay should be met by our architecture. Since for a delay-budget of 1 sec, the maximal allowable edge-to-edge delay can be 85 msec, the outbound CES can absorb the first DNS re-attempt from the host operating system in order to support the larger edge-to-edge delay requirements. With this provision, CETP negotiation can tolerate an edge-to-edge delay of 250 milliseconds.

Furthermore, a high performance implementation of CETP and CES logic can reduce the processing delay of 490 ms and thus improve the tolerance to high edge to edge delays. The additional delay requirement does not disturb the normal communication pattern significantly, and most times the initiating host does not notice a difference compared to the current situation.

7.4 Testing Circular Pool Security

The network setup submitted for testing circular pool security is presented in Figure 7.14. The setup consists of a private network behind CES/PRGW and two external legacy networks. The CES/PRGW defines two distinct interfaces for connectivity with the networks, defined below.

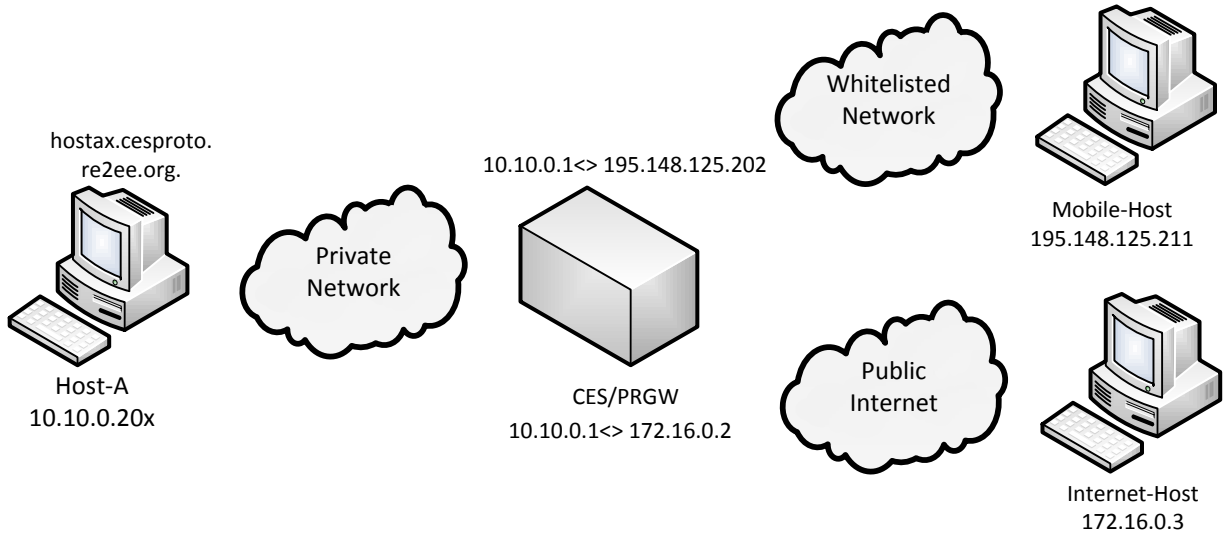


Figure 7.14 Testing Network for Circular Pool

Whitelisted Network: The traffic from the mobile operator network is received by the CES/PRGW over a dedicated interface i.e. 195.148.125.0/24 address.

Public Network: The public network is a legacy IP network with no CES device. The traffic from the public network is expected over the interface with 172.16.0.0/24 addressing.

CES/NAT: serves as a gateway between the private and the public networks, and it acts as an authoritative name server for the domains in the private network. The circular pool address range 195.148.125.20[3-10] serves a legacy host initiating a connection with Host-A behind the CES.

The whitelisted network interface is 195.148.125.0/24 interface, while the traffic from other networks is expected over the greylisted interface. A connection request over the whitelisted interface is offered more resources and better services in the circular pool than corresponding greylisted connection request.

7.4.1 Testing Security against DNS Spoofing

Circular pool security relies on categorizing name servers into whitelisted and greylisted name servers and upon the CPOOL address allocation model. However, by spoofing a whitelisted DNS, an attacker can reserve more resources (or addresses) from the circular pool and can launch a DoS attack. Therefore, the network must perform aggressive filtering to eliminate spoofing in the inbound DNS requests.

This section presents how filtering on the received DNS requests can protect the circular pool from DoS attack. But, because the DNS requests are received over the stateless UDP protocol, it leaves very few options to eliminate spoofing. However, the current prototype only provides the security against DNS spoofing in a multiple interface model deployment model of PRGW, e.g. when a DNS request from whitelisted server is received over a greylisted interface.

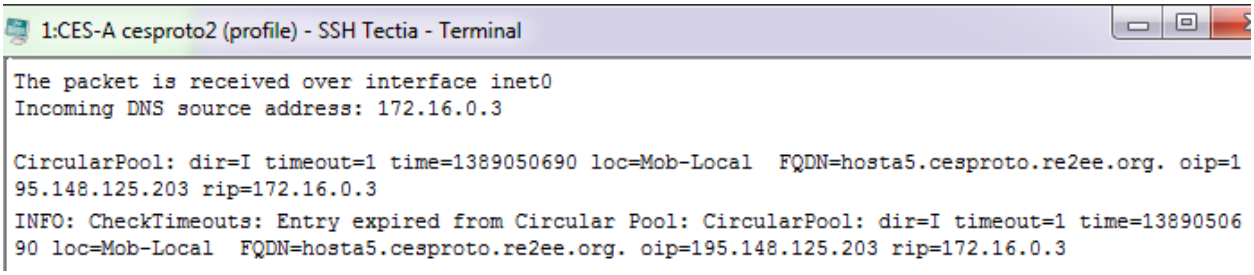


Figure 7.15 cross interface DNS spoofing, before security

In this test, the circular pool receives a spoofed DNS request with address 172.16.0.0/16 over the interface reserved for whitelisted traffic, i.e. 195.148.125.0/24 addressing. Figure 7.15 shows that the spoofed DNS request was able to reserve a connection in the circular pool. However, after applying filtering on the inbound DNS request the spoofing is detected and the DNS request is dropped, presented in Figure 7.16.

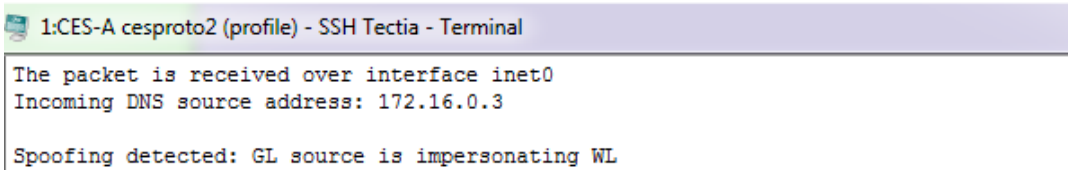


Figure 7.16 cross-interface DNS spoofing detected, after security

By filtering the inbound DNS requests at the network, spoofing attacks can be detected and the circular pool is protected from DoS attacks. However, realizing the limited possibilities in the

presence of UDP protocol, further research needs to be carried out to eliminate DNS spoofing. With existing algorithms to eliminate spoofing in TCP, i.e. cookie mechanism, the possibility of DNS over TCP protocol for whitelisted name servers needs to be explored in the future research. This could provide a security model where whitelisted name servers can access circular pool resources even under attack conditions, i.e. DNS spoofing attacks, without the need for having a separate interface for whitelisted traffic.

7.4.2 Testing security against DoS Attacks

This demonstrates the circular pool protection against denial of service attacks, based on the security model. Figure 7.17 shows that a DNS request from the name server is dropped by the circular pool, if the name server has already reached to the maximum number of waiting states allowed for a DNS source.

Figure 7.17 Limiting maximum number of connections from a DNS source

Figure 7.18 shows that the circular pool drops a DNS request for the destination, if the domain already has the maximum number of waiting states allowed per domain, i.e. 1 in this case.

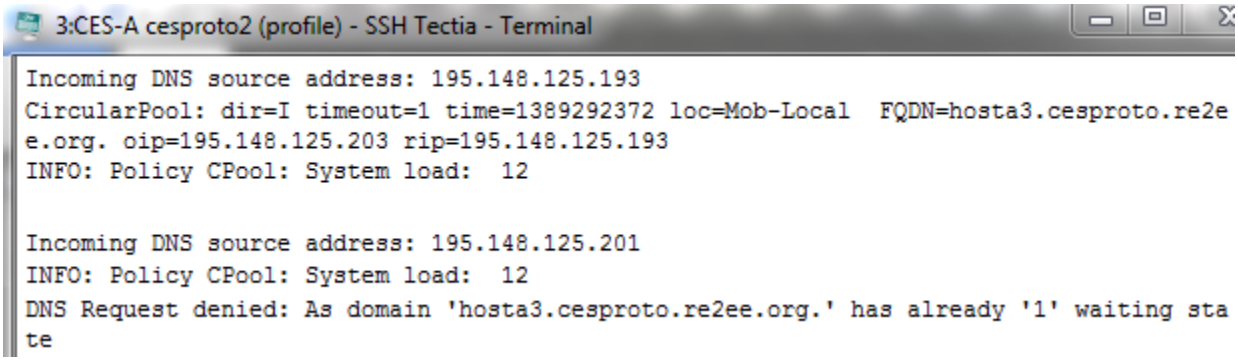


Figure 7.18 Limiting maximum number of connections to a destination

It is pertinent to mention that the DNS requests received in this test case are assumed to be non-spoofing. The network must ingress filter the received DNS requests to eliminate spoofing.

The next section tests the security protection of the CPOOL against a DDoS attack, where an attacker sends a DNS request through various name servers to multiple destinations behind the circular pool to reserve all the CPOOL addresses.

7.4.3 Testing CPOOL Address Allocation Model

This section not only verifies the conformance of CPOOL address allocation model, but also demonstrates the circular pool's protection against the DDoS attack, identified in section 5.2.6. The test is conducted by sending multiple DNS requests towards the circular pool from a set of whitelisted and greylisted name servers. The name servers with '195.148.125.19x' prefix are registered as whitelisted DNS sources, while the rest, i.e. '195.148.125.18x', are classified as greylisted DNS sources. The parameter system load, defined in section 6.2.2, is computed against a total of 8 circular pool addresses, for this test. The circular pool defines (minimum, medium, maximum) threshold values of the system load as (60, 80, 100) in this test case, which translate to (5,6,8) addresses in the circular pool of 8 addresses.

Figure 7.19 shows the address allocation behavior when the circular pool is subjected to DNS requests from greylisted DNS sources. Once the system load reaches its minimum threshold the subsequent DNS request from a greylisted source is dropped. Hence, it ensures that an attacker from the public Internet (greylisted) is unable to launch a DDoS attack on the circular pool.

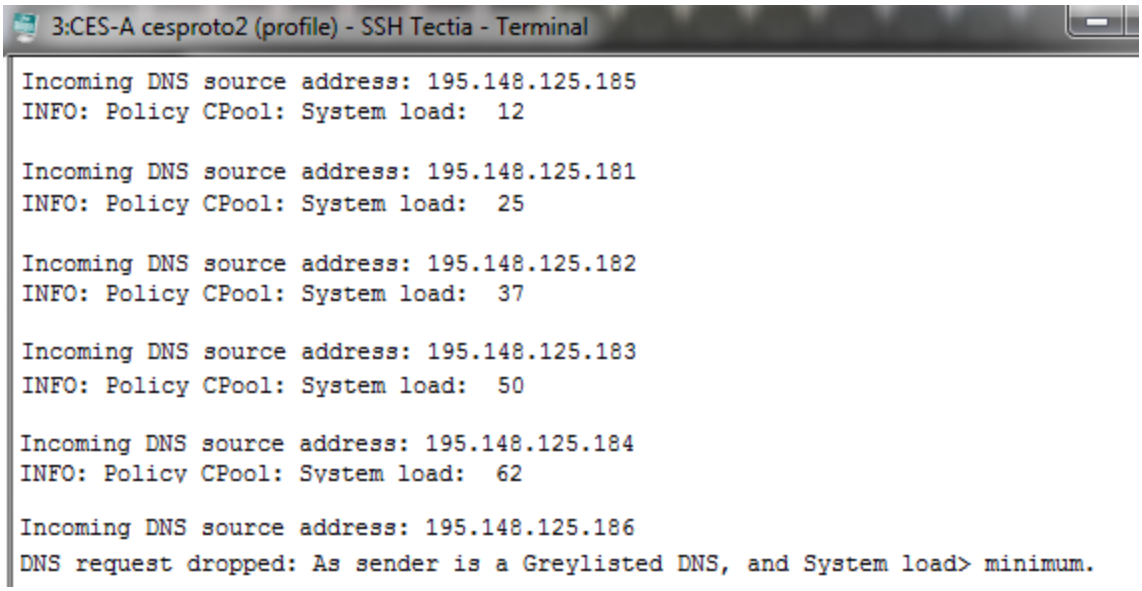


Figure 7.19 CPOOL address allocation for greylisted DNS servers

Figure 7.20 presents the address allocation behavior of the circular pool when subjected to DNS requests from a set of whitelisted and greylisted DNS sources, simultaneously. The whitelisted DNS sources correspond to a legitimate access, while the greylisted DNS sources provide an abstraction for the DDoS attack launched by an attacker, in this test case. The figure shows that CPOOL assigns an address to a whitelisted DNS request received after the system load has reached the minimum threshold, while a similar request from a greylisted source was dropped previously, in Figure 7.19. Hence, the model ensures that a whitelisted DNS source is preferred over corresponding greylisted name server, and it always has certain resources available in the circular pool, under the attack conditions.

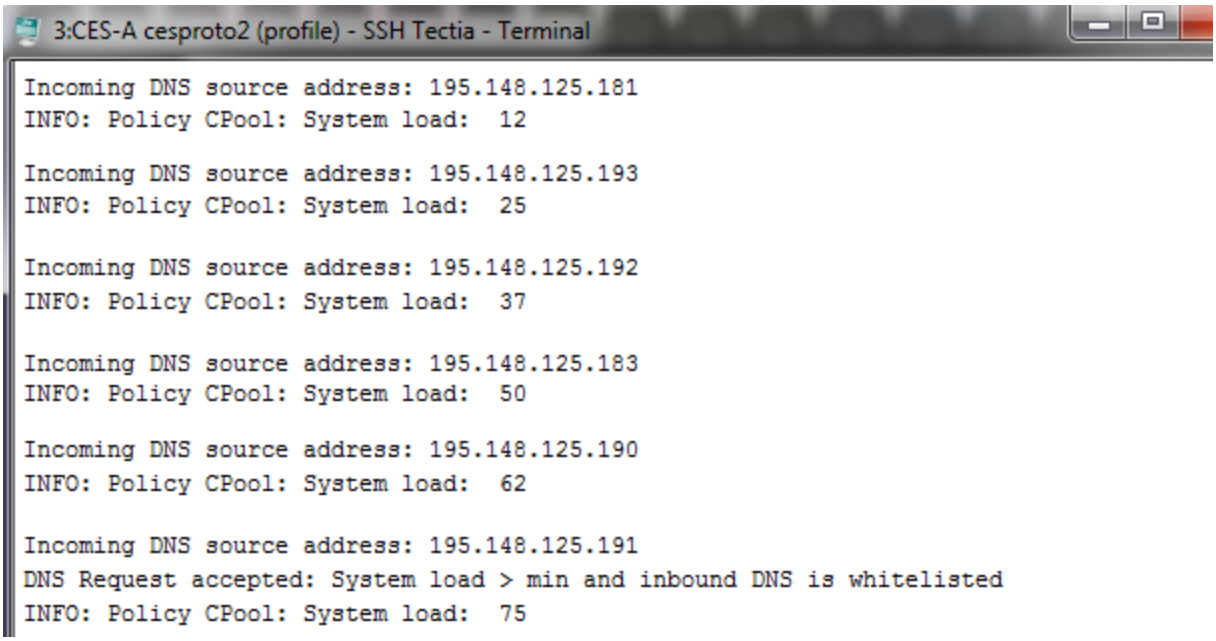


Figure 7.20 CPOOL address allocation when system load is above threshold

7.4.4 Testing Security against Connection Hijacking Attempts

The circular pool can prevent connection hijacking attempts in the CPOOL by employing a set of techniques, presented in section 6.2.5. The circular pool can either deny a hijacking attempt based on the waiting state or can rely on the transport protocol to prevent connection hijacking.

In the waiting state mechanism, the CES prototype ensures that a ‘waiting’ state created by a whitelisted DNS can only be reserved by a whitelisted IP source. This prevents a greylisted IP source from hijacking a waiting state created by a whitelisted name server. Figure 7.21 shows a greylisted IP source ‘172.16.0.3’ is denied of taking over the waiting state reserved by a whitelisted address ‘195.148.125.201’. Whereas, a whitelisted IP source can successfully claim the state reserved by a whitelisted name server, as shown in Figure 7.22

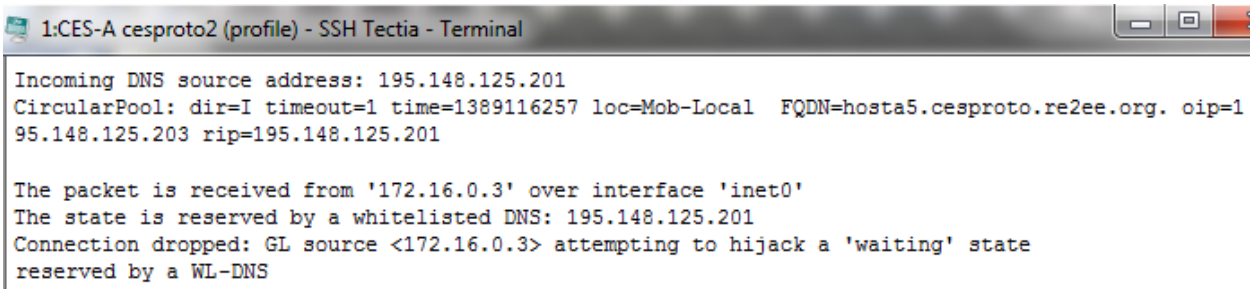


Figure 7.21 Preventing connection hijacking by a greylisted source, after security

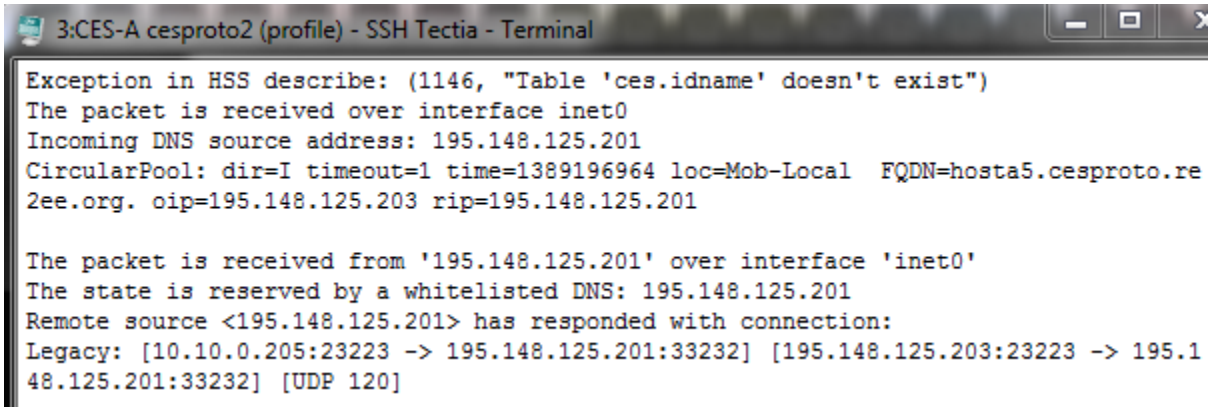


Figure 7.22 Whitelisted source taking the 'waiting' state by whitelisted DNS

The circular pool ensures that a waiting state is protected against connection hijacking attempts based on the transport protocol of the received packet. The circular pool drops an inbound UDP packet for claiming a 'waiting' connection state, as described in section 7.4.5. However, an IP packet received with TCP protocol can successfully claim a waiting state from the circular pool, because spoofing checks can be executed for the received TCP packet.

The current circular pool design supports two security mechanisms to prevent the CPOOL against TCP based connection hijacking attempts: Logging/filtering method or Bot detection method.

Figure 7.23 Preventing connection hijacking using Logging/filtering approach

Logging/filtering mechanism is applied in the networks where spoofing is eliminated on the inbound packets e.g. by a carrier CES. The elimination of spoofing guarantees that a packet has indeed come from a non-spoofing source. Therefore, generating a blacklist of hosts based on the dropped packets identifies the sources of connection hijacking attempts. Figure 7.23 presents this mechanism where the circular pool blacklists a sender after receiving 3 packets from the source without a corresponding waiting state. Once blacklisted, any future packets from the source are not accepted for claiming a waiting state in the circular pool, for duration 'T_O'. The threshold of 3 dropped packets is only selected for demonstration purposes.

Bot detection method aims at thwarting connection hijacking attempts by identifying the bot-controlled attacker host (or amplifier). Figure 7.24 shows that when repeated packet drops from the sender '195.148.125.201' reach a threshold 'x' within time duration 'T', the CPOOL sends a fake TCP SYN/ACK segment in response to the last received SYN packet. The fake SYN/ACK segment carries an initial sequence number (ISN), computed according to SYN cookie mechanism [38].

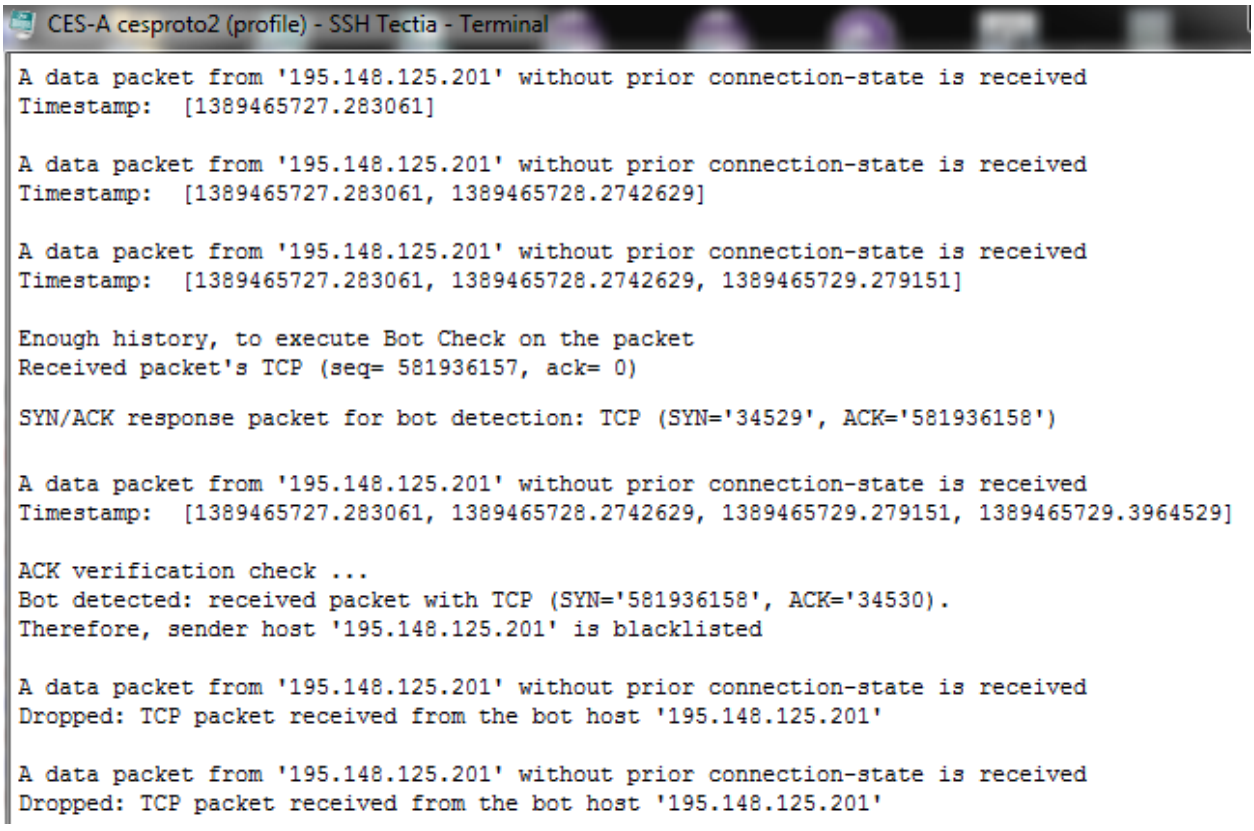


Figure 7.24 Bot-detection method to prevent connection hijacking

For a non-spoofing bot, upon receiving SYN/ACK segment, the host responds with ACK segment bearing the same ISN value. Next, because the received ACK segment carries the expected (ISN+1) value, the sender is verified as a non-spoofing source. Based on the history of packets dropped and the non-spoofing check, the CPOOL identifies the sender as a bot-controlled legacy host. Following this, the sender is marked as blacklisted and future connection requests from this source are not accepted for time duration 'T_O'.

7.4.5 Testing Protection against UDP Flow Initiations

Figure 7.25 demonstrates that the circular pool drops a UDP packet received for claiming a 'waiting' connection state, as per the security model.

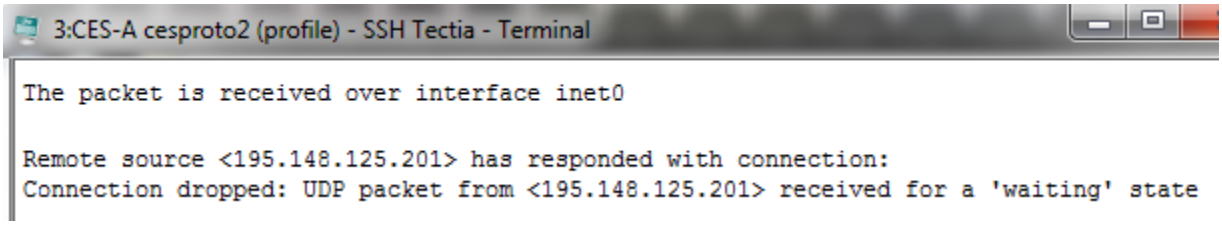


Figure 7.25 CPOOL drops UDP packet for a waiting state

7.4.6 Performance Analysis

The performance of the circular pool security model depends on the security of DNS and the protection against connection hijacking attacks.

This section presents the performance analysis of Bot-detection method against connection hijacking attempts at the circular pool. Figure 7.26 presents the timeline view of circular pool security against connection hijacking attempts. In this case, a circular pool with 1 address receives SYN segments from a legitimate sender and an attacker host, simultaneously. The attacker floods the circular pool with TCP SYN segments at an average of 20 connections per second. Essentially, the traffic received by the circular pool has the concentration of 20:1 in favor of the attacker.

Amidst the SYN segments from the attacker, the legitimate host issues a DNS query to reserve an address from the circular pool. The host then sends subsequent TCP SYN packet to claim the reserved address and to establish the connection with the destination. However, the attacker host

sending SYN segments at much higher rate than the legitimate host, i.e. 20 connections per second, hijacks the reserved state. As a result, the legitimate user suffers from the DoS attack.

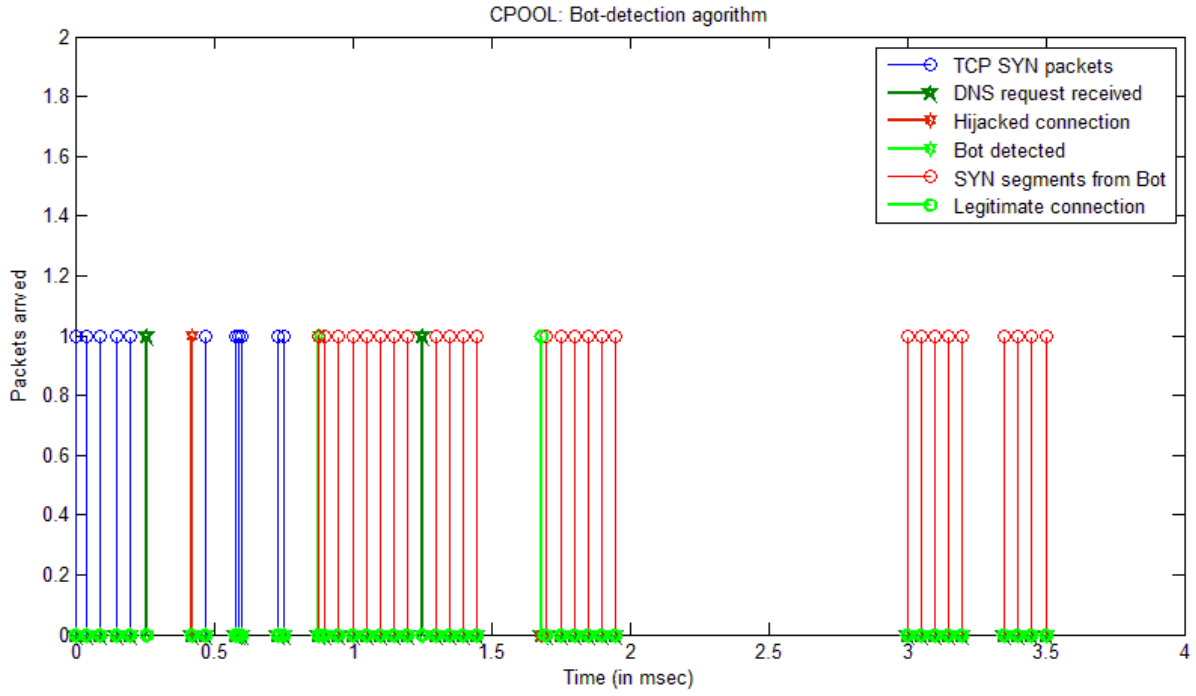


Figure 7.26 Bot detection method to prevent connection hijacking

When the number of dropped TCP SYN segments from the host exceeds a threshold 'x', i.e. 20 here, the circular pool issues a TCP SYN/ACK segment towards the sender with a specific ISN. If a TCP ACK segment is received with expected ISN from the sender, the sender is classified as a bot-controlled legacy host because of its history of dropped packets and non-spoofing check. Following this, the SYN segments from the bot are dropped to prevent connection hijacking in the CPOOL. Hence, next time, when the legitimate host reserves an address from the CPOOL, the connection doesn't suffer from hijacking as the bot host has been identified and the packets received from the bot are dropped by the CPOOL, as shown at time=1.6 millisecond. The traffic from the attacker, once identified as bot, is dropped and it is shown by the lines in red color.

The figure clearly shows that the method is not proofed against false negative case, as shown by the hijacked connection. However, the method clearly reduces the chances of a false positive by carefully choosing the threshold value for bot identification. Operating systems, Windows and Unix, establish a connection using multiple RTTs before giving up on the remote endpoint and the next RTT from a legitimate host is transmitted after retransmission timeout, i.e. 3 seconds for Unix. Therefore, the measurement duration for detecting a bot-controlled host should be less than this RTO value, to avoid false positives. This prevents the CPOOL from classifying a

legitimate host as an attacker, because of multiple SYN segments received for the already hijacked connection state.

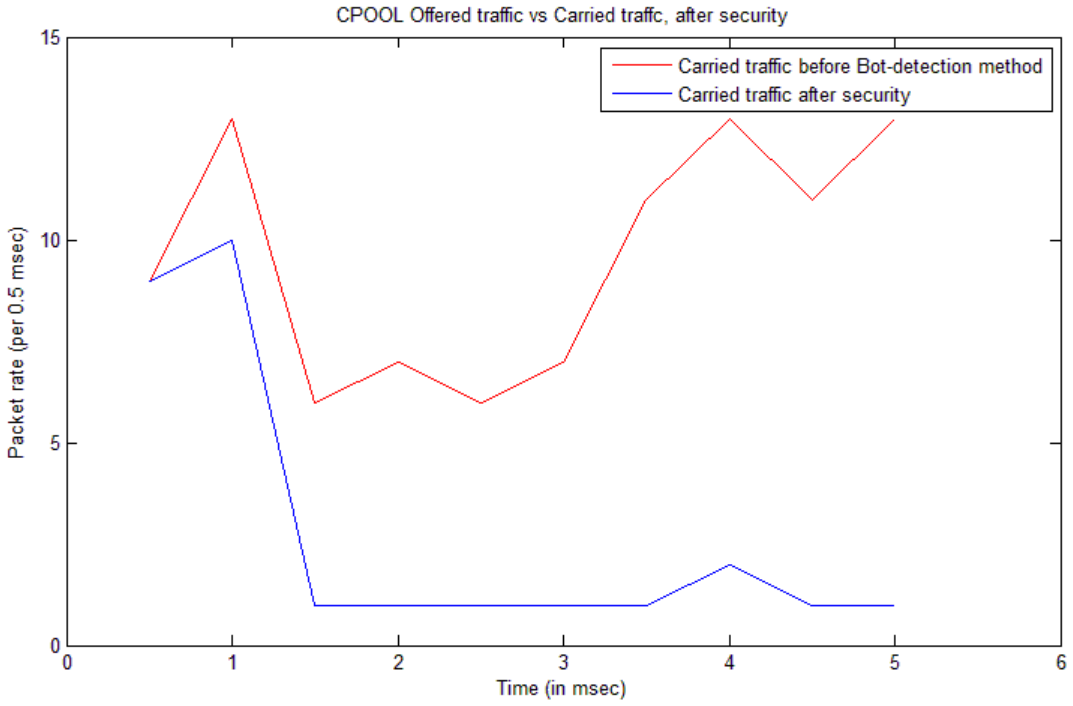


Figure 7.27 Traffic influx before and after security

Figure 7.27 shows another aspect of Bot-detection method for preventing connection hijacking. The figure demonstrates reduction in the volume of traffic accepted by the CPOOL for claiming a connection state, using a legitimate host and an attacker generating SYN segments at an average of 20 connections/sec. The figure shows that all the offered traffic was carried into the CPOOL and was processed for connection establishment, prior to the security. However after the security, the SYN segments accepted for connection establishment reduced drastically once the bot-controlled host is identified. The reduction in attack volume of the carried traffic versus offered traffic is an attribute of the circular pool security.

8. Conclusion

The Master thesis was aimed at identifying the security vulnerabilities present in the CES architecture. The research work included the development of the security mechanisms to protect CES against the attacks on these vulnerabilities. The thesis identifies, documents and secures the CES architecture against vulnerabilities present in the circular pool design and the CES-to-CES communication model.

In CES-to-CES communication, the CES prototype was vulnerable to denial of service attacks, because of spoofing in the received packets. The CETP security model employs an approach similar to TCP SYN cookie mechanism to eliminate spoofing on the received CETP packets. A detailed evaluation of the implemented security model suggests that CES is not vulnerable to spoofing attacks anymore. Besides patching the vulnerabilities, the thesis contributes towards security by proposing two CES registration models. These CES registration/verification models are necessary to distinguish a legitimate CES from a legacy host imitating to be a CES. This protects CES against various attack forms launched from legacy hosts or botnets.

The evaluation of CES after security shows that the CES architecture is secured against different network attacks, without introducing significant processing in CES. The performance analysis of the security model reveals that CES-to-CES communication suffers a mere delay of less than 3 milliseconds, if ID verification is not mandatory on the received CETP packet.

The circular pool design was vulnerable to denial of service attacks from malicious users or botnets. However, the circular pool security model employs a set of tools i.e. greylisted/whitelisted sources, circular pool address allocation, bot detection algorithm etc. to prevent DoS conditions at the circular pool. The CPOOL address allocation model guarantees that a whitelisted source is preferred over a greylisted sender and it always has resources available in the circular pool, for a legitimate access. The security model also aims at averting possible connection hijacking attempts in the circular pool through a set of techniques. A series of tests conducted to evaluate security of the circular pool have depicted promising result from security perspective.

Besides presenting the security model for the circular pool, the thesis discusses different circular pool deployment models to further strengthen security of the circular pool. These deployment models in conjunction with the security model guarantee a promising degree of assurance for a legitimate access.

It is pertinent to mention that because the thesis has been carried out as a part of a large project with existing python modules, the security has been implemented in Python and it is therefore not optimized in terms of performance. However, the performance of the CES prototype and its security can be enhanced considerably, if it is re-encoded with C/C++. Similarly, the security of CES can be further optimized if the packets received from a blacklisted source are discarded at the data plane, rather than forwarding them to the control plane of CES.

8.1 Future Work

This section presents a brief summary of some important topics related to security that were not covered by this thesis or were considered out of the scope.

Proof-of-work mechanism: The current prototype employs a cookie mechanism to eliminate spoofing in the received CETP packets. In this mechanism the receiver generates and verifies the cookie involved in the CETP connection establishment. However, a proof of work mechanism can shift the processing overload for the cookie computation towards the sender while the receiver only needs to verify the received cookie. The mechanism not only can further reduce the processing load at the receiver but it can increase the cost of launching spamming, spoofing and DoS attacks for an attacker.

Certificate based CES registration: The certificate based CES registration mechanism can be permanently added to the CES architecture, given its acceptance in a research conference. Moreover, the mechanism needs to be realized with a suitable library as the current implementation does not support modifications in X.509 standard for the certificates.

Sybil Attack: occurs when an attacker disguises itself behind multiple identities acquired for itself. The attacker represents itself with a new FQDN each time it performs a malicious activity. Such an attack significantly impacts the trust ratings of CES in a trust reputation system.

DNS/TCP: A DNS request received over UDP can be subjected to spoofing attacks, which affects the CPOOL address allocation model. Therefore, the possibility of receiving DNS requests over TCP for a whitelisted name server needs to be explored. With TCP, spoofing can be eliminated using the SYN cookie mechanism. Moreover, a TCP connection can remain open for many DNS queries from the same DNS source.

Secure signaling for UDP flows: Following the deployed security model, the circular pool must define a secure signaling for connection establishment, before accepting an inbound UDP packet from a legacy host. SIP functionality is supported in CES through ALGs [39].

Faster control plane and data plane: The connection establishment delay due to the deployed security mechanisms can be further reduced, if CES is re-encoded in a more efficient programming language i.e. C/C++.

Besides these topics, the Diameter protocol for verification queries with HSS, encryption to prevent eavesdropping on CETP signaling, definition of cooperative mechanisms between CES devices, and TCP-Relay approach to prevent connection hijacking in the circular pool offer further venues of research, to improve the security of CES.

9. References

[1] J. Rosenberg, R. Mahy, P. Matthews, and D. Wing, "Session Traversal Utilities for NAT (STUN)," RFC 5389, 2008.

[2] J. Rosenberg and E. S. Perreault,, "Traversal Using Relays around NAT (TURN) Extensions for TCP Allocations," RFC 6062, 2010.

[3] J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols," RFC 5245, 2010.

[4] J. S. Llorente, "Private Realm Gateway," Master Thesis, Aalto University, School of Electrical Engineering, 2012.

[5] M. Pahlevan, "Signalling and Policy Enforcement for Cooperative Firewalls," Aalto University, School of Electrical Engineering Thesis, 2013.

[6] R. Kantola, J. Manner, J. Luoma, and N. Beijar, "Towards Internet of Trust," Aalto University, School of Electrical Engineering, 2011.

[7] "2013 Cisco Annual Security Report," CISCO, 2013.

[8] "Security in Telecommunications and Information Technology," ITU-T, 2003.

[9] W. Stallings, *Cryptography and Network Security: Principles and Practice Pearson*, 5th ed. Prentice Hall, 2010.

[10] "The Basics of Computer Security," Aalto University, School of Electrical Engineering S-38.3153, Noppa Lecture , 2013.

[11] J. E. Canavan, *Fundamentals of Network Security*. Artech House Publishers, 2001.

[12] "Economic Impact of Network Security Threats," CISCO Systems White Paper.

[13] R. Needham and B. Lampson, "Network Attack and Defense," in *Security Engineering: A Guide to Building Dependable Distributed Systems*, 2008, ch. 18, pp. 383-390.

[14] H. Bertine, "ITU-T Security Standardization: Telecommunication Security," ITU-T Presentation, 2006.

[15] A. Boulanger, "Internet Infrastructure Attacks," in *Cybercrimes: A multidisciplinary Analysis*, ch. 13, pp. 209-213.

[16] A. Chakrabarti and G. Manimaran, "Internet Infrastructure Security: A Taxonomy," in *Network, IEEE*, 2002, pp. 13-21.

[17] T. Chen and P. J. Walsh, "Guarding Against Network Intrusions," in *NETWORK AND SYSTEM SECURITY*, ch. 4, pp. 87-88.

[18] B. A. Forouzan, *Data Communications and Networking*, 4th ed. Alan R. Apt, 2007.

[19] A. Shamir , R. L. Rivest, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," in *ACM*, NewYork, 1978, pp. 120-126.

[20] R. Rivest, "RFC1321, The MD5 Message-Digest Algorithm," RFC, 1992.

[21] D. Cooper, et al., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 5280, 2008.

[22] S. Cooper, Z. D. Elizabeth , and C. D. Brent, *Building Internet Firewalls*, Second Edition ed. 2000.

[23] J. R. Vacca, *Network and System security*, 1st ed. Syngress, 2010.

[24] "Logging and Monitoring to Detect Network Intrusions and Compliance Violations in the Environment," The SANS Instititute Certification Paper, 2012.

[25] M. Nakhjiri and M. Nakhjiri, *AAA and Network Security for Mobile Access: Radius, Diameter, EAP, PKI and IP Mobility*, 1st ed. USA: Wiley, 2005.

[26] C. Rigney, A. Rubens, W. Simpson, and S. Willens, "Remote Authentication Dial In User Service (RADIUS)," IETF RFC 2138, 1997.

[27] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko, "Diameter Base Protocol," IETF RFC 3588, 2003.

[28] G. Camarillo and M.-A. García-Martín, "Overview of IMS Archiecture," in *The 3G IP Multimedia Subsystem: Merging the Internet and the Cellular Worlds*. WILEY, 2008, pp. 30-32.

[29] G. Heine, "The Network Switching Subsystem," in *GSM Networks: Protocols, Terminology and Implementation*, 1998.

[30] W. Stallings, *Data and Computer Communications*. New Jersey, 2007, ch. 23, pp. 773-784.

[31] K. Egevang and P. Francis, "The IP Network Address Translator (NAT)," RFC 1631, 1994.

[32] E. F. Audet and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP," RFC 4787, 2007.

[33] R. Kantola, "Implementing Trust-to-Trust with Customer Edge Switching," *Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference*, pp. 1092-1099, Apr. 2010.

[34] D. D. Clark, "Application design and the end-to-end arguments," in *MIT Communications Futures Program, Bi-annual meeting, May 30-31*, Philadelphia, PA, May 2007.

[35] Z. Yan, R. Kantola, and Y. Shen, "Unwanted Traffic Control via Global Trust," in *IEEE TrustCom*, Changsha, China, 2011.

[36] N. Beijar, Z. Yan, M. Pahlavan, and R. Kantola. (2012, Mar.) Customer Edge Traversal Protocol. [Online]. http://www.re2ee.org/

[37] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246, 2008.

[38] W. Eddy, "TCP SYN Flooding Attacks and Common Mitigations," RFC 4987, 2007.

[39] P. Leppaaho, "Design of Application Layer Gateways for Collaborative Firewalls,," M.Sc. Thesis, Aalto University, Department of Communications and Networking, 2012.

Appendix A - Certificate based CES registration/authentication

This appendix presents a practical realization of the proposed decentralized mechanism for CES registration, discussed in section 6.1.6. The approach utilizes certificates issued by CA to determine if the CETP sender is indeed a legitimate CES device or a bot-controlled legacy host.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2 (0x2)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer:
        Validity
            Not Before: Jan 15 16:52:13 2014 GMT
            Not After : Feb 14 16:52:13 2014 GMT
        Subject: C=FI, CN=cesa.cesproto.re2ee.org., ST=CA, O=work, OU=testing
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:bf:c0:6c:a8:79:e9:73:dd:20:a4:f5:62:c6:f2:
                    34:7c:7f:dd:21:9a:f8:23:02:67:c5:88:1c:e2:d9:
                    98:3e:12:59:b2:e1:4f:42:4f:62:9e:a4:29:ec:71:
                    43:a4:40:72:55:2a:90:8b:5b:d4:75:bb:cd:1c:64:
                    c6:9d:81:48:c9:80:93:e6:64:0f:74:ad:9f:ef:17:
                    66:c6:15:ff:07:ac:12:ac:c6:72:e9:e4:20:85:e9:
                    2b:45:18:d9:bf:40:04:3a:04:68:8b:bf:82:2d:7c:
                    03:f3:b3:f4:68:12:a1:12:48:fc:9f:53:19:1f:a1:
                    bb:91:5c:de:8a:3c:ba:e0:b5
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage: critical
                Digital Signature
            Netscape Comment:
                CES Verification
    Signature Algorithm: sha1WithRSAEncryption
        33:37:12:93:68:0a:6e:d4:3a:a3:4c:cf:42:a6:29:87:16:a9:
        73:87:03:e3:9f:78:aa:70:ea:ca:ce:4a:e8:a8:18:be:97:fe:
        01:da:98:37:73:b4:5b:80:7b:32:3b:5c:22:75:16:cd:85:c4:
        eb:77:89:6d:8e:e9:c5:e3:9e:91:8d:73:78:71:98:31:f2:69:
        15:10:34:31:0d:b5:3f:50:a0:6d:fd:47:39:4a:ad:cd:ff:56:
        ec:76:d3:65:74:25:b3:0d:d3:d1:ac:65:4b:46:8c:b1:39:72:
        e5:48:0c:0f:7e:f3:5f:41:c0:db:e4:03:67:7f:95:eb:02:c1:
        59:23
```

Figure A.1 Certificate issued by CA to a CES device

Figure A.1 presents the certificate issued by a CA to the legitimate CES device of id ‘cesa.cesproto.re2ee.org.’ The extensions present in this X.509v3 certificate comply with our

suggested CES specific certificate. The Basic constraint field indicates that the certificate does not belong to a CA. Rather it is an End-Entity certificate i.e. host, webserver etc. The Key Usage field, marked as critical, limits the certificate usage to Digital Signatures only. Next, the certificate utilizes Netscape Comment field, instead of Extended Key Usage field, to describe the purpose of the certificate. The Netscape Comment field is used as an equivalent to Extended Key Usage field in this demonstration, because the M2Crypto library doesn't support changes (or additions) in the already defined values for Extended Key Usage field.

The "CES Verification" value of *Extended Key Usage* field states that the certificate belongs to a legitimate CES device. And, a receiver shall verify the received digital signature with the public-key in the certificate to determine if the sender is a legitimate CES device. The absence of "CES Verification" value in *Extended Key Usage* field prevents a (certificate bearing) legacy host from imitating as a legitimate CES device.

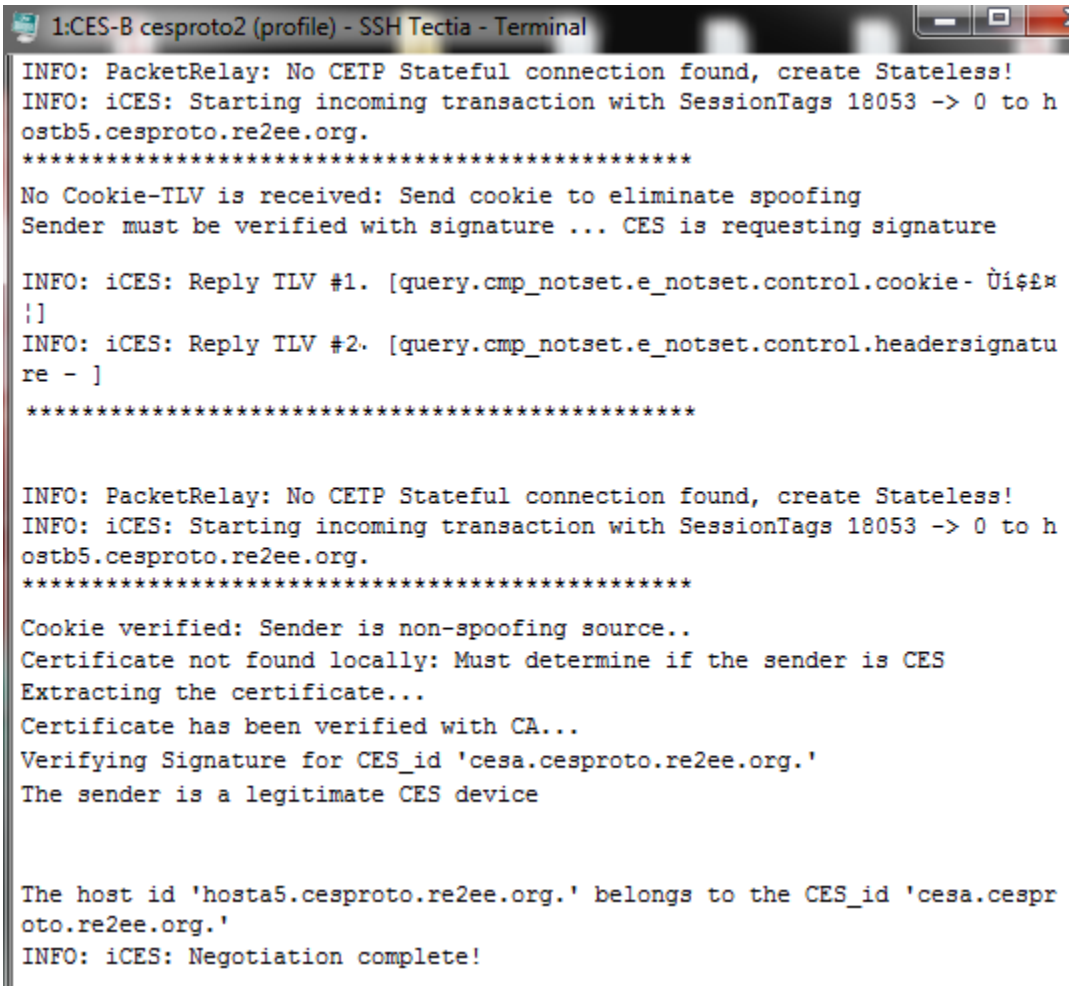


Figure A.2 Certificate based CES verification for the first CETP packet received

Figure A.2 presents the CES verification process carried by the iCES upon receiving the first CETP packet from the sender. The iCES responds with a cookie-TLV to eliminate the spoofing on the received packets and requests the sender of CETP header signature in the next inbound CETP packet along with the sender's certificate. The next inbound packet with the same SST=18052 and DST=0 is received from the sender with a valid cookie, header signature and the requested certificate.

If the Extended Key Usage field of the received certificate carries "CES Verification" value and if the CETP header signature can be verified with the received certificate, the sender is accepted as a legitimate CES device. Following this, subsequent packets from the sender are accepted without performing the CES verification process again.