jamk.fi

Design and implementation of Telia firewall laboratory

Samuli Varjoluoto

Bachelor’s Thesis
Fall 2018
Information and communication technology
Bachelor’s degree (AMK), Information and communication technology
Data Network Techonology

Jyväskylän ammattikorkeakoulu
JAMK University of Applied Sciences

jamk.fi

Description

Author(s)	Type of publication	Date
Varjoluoto, Samuli	Bachelor's thesis	April 2019
		Language of publication: English
	Number of pages 85	Permission for web publication: yes

Title of publication

Design and implementation of Telia firewall laboratory

Degree programme

Information and Communication Technology, Data network technology

Supervisor(s)

Saarisilta Juha, Kotikoski Sampo

Assigned by

Telia Finland Oyj

Abstract

The thesis was assigned by Telia Finland Oy, a large telecommunications operator that provides cellphone and internet connection subscriptions to private and business customers, as well as datacenter services and network design, implementation and maintenance for corporations. The objective was to design and implement a Check Point laboratory environment to replace an existing obsolete laboratory environment. The laboratory was to be used for testing firewall appliances and designing new firewall services that could be provided for business-to-business customers.

The initial topology design of the firewall laboratory was conducted with firewall specialists from Telia by interviewing to them and gathering information about what they wanted the end product to be like. The implementation of the laboratory environment initially consisted of clearing out old firewall appliances, switches and routers. The new firewall appliances and switches were manually installed in racks and cabled to create the physical topology of the laboratory. Remote management to the laboratory system was built over the internet and access was provided for the firewall specialists' VPN network allowing the operation of the laboratory environment from anywhere.

The final product of the assignment was a functioning firewall laboratory that could be easily expanded and modified to match any needs. The provided laboratory could also be used as a midway point when migrating old customer firewall configurations from legacy systems to newer ones.

Keywords/tags (subjectshttp://vesa.lib.helsinki.fi/)

Firewall Laboratory, Network design, VPN

Miscellaneous

jamk.fi

Kuvailulehti

Tekijä(t)	Julkaisun laji	Päivämäärä
Varjoluoto, Samuli	Opinnäytetyö, AMK	Huhtikuu 2019
		Julkaisun kieli Suomi
	Sivumäärä 85	Verkkojulkaisulupa myönnetty: kyllä

Työn nimi

Telian palomuurilaboratorioympäristön suunnittelu ja toteutus

Tutkinto-ohjelma

Tieto- ja viestintätekniikka

Työn ohjaaja(t)

Juha Saarisilta, Sampo Kotikoski

Toimeksiantaja(t)

Telia Finland Oyj

Tiivistelmä

Opinnäytetyönaiheen toimeksiantajana toimi Telia Finland Oy, suuri tietoliikenneoperaattori, joka tarjoaa puhelin- ja internetliittymiä yksityis- sekä yritysasiakkaille. Lisäksi yrityksille tarjotaan konesali- sekä verkon suunnittelu-, toteutus- ja ylläpitopalveluita. Tavoitteena oli suunnitella ja toteuttaa Check Point -palomuurilaboratorioympäristö korvaamaan olemassa oleva vanhentunut laboratorioympäristö. Laboratorioympäristöä käytettäisiin uusien palomuurituotteiden ominaisuuksien sekä uusien palveluiden testaamiseen.

Laboratorioympäristön alkuperäinen topologiasuunnitelma tehtiin yhdessä Telian palomuuriasiantuntijoiden kanssa haastattelemalla heitä ja kartoittamalla heidän tarpeensa laboratorion suhteen. Laboratorion toteutus alkoi vanhentuneiden palomuurilaitteiden, kytkimien ja reitittimien siivoamisella laboratoriotilasta. Uudet palomuurilaitteet ja kytkimet asennettiin laboratoriossa sijaitseviin räkkeihin ja kaapeloitiin. Näin luotiin laboratorion fyysinen topologia. Etähallinta laboratoriolaitteisiin luotiin internetin yli ja pääsy sallittiin palomuuriasiantuntijoiden VPN-verkosta, mikä mahdollistaa laboratorion hallinnan mistä tahansa.

Opinnäytetyön lopputulokseksi muodostui toimiva palomuurilaboratorio, jota on mahdollista laajentaa tarpeen tullen. Laboratoriota voitaisiin myös käyttää välietappina asiakastoteutuksia siirrettäessä vanhasta ympäristöstä uuteen.

Avainsanat/tags (subjectshttp://vesa.lib.helsinki.fi/)

Palomuurilaboratorio, Verkonsuunnittelu, VPN

Muita tietoja

1 Contents

1 Telia Company 4
2 Introduction and research question 4
3 Theory 5
3.1 Firewalls 5
3.2 Check Point R80.10 6
3.3 Network Design 9
3.4 VPN 10
3.5 IKE 10
3.5.1 ISAKMP 11
3.5.2 Oakley 13
3.5.3 SKEME 13
3.6 IPsec 14
3.6.1 Protocol mode 14
3.6.2 ESP 14
3.6.3 Authentication Header 15
3.7 Virtual Local Area Network (VLAN) 16
4 Designing the environment 17
5 Implementing the environment 22
5.1 Objective and first steps 22
5.2 Check Point firewall gateway configuration 23
5.3 Check Point management device configuration 25
5.4 VPN tunnel configuration for customer traffic 29
5.5 Switch management network configuration 35
6 Conclusion 38
References 40
Appendices 42

Figures

Figure 1. Checkpoint Database Domain types 7
Figure 2. System performance 8
Figure 3. Check Point R80.10 status monitoring 8
Figure 4. ISAKMP Header 11
Figure 5. ESP Header 15
Figure 6. AH Header 16
Figure 7. First draft of the topology 18
Figure 8. Second draft of the topology 19
Figure 9. Third draft of the topology 20
Figure 10. Final draft of the topology 21
Figure 11. Old laboratory 23
Figure 12. CheckPoint R80.10 first time wizard 24
Figure 13. Check Point R80.10 management installation 26
Figure 14. Creating a cluster on the management device 27
Figure 15. Cluster states 27
Figure 16. Firewall instances 28
Figure 17. Firewall policy 28
Figure 18. Adding route to Juniper router 29
Figure 19. VPN configuration 1 30
Figure 20. VPN configuration 2 31
Figure 21. VPN configuration 3 32
Figure 22. Firewall VPN policy 32
Figure 23. VPN tunnel up 33
Figure 24. Ping through VPN tunnel 34
Figure 25. SSH from ubuntu jump machine to switch 36
Figure 26. Final laboratory Check Point side 37

Acronyms

AD	Active Directory
AES	Advanced Encryption Standard
AH	Authentication Header
CP	Check Point
DOI	Domain of Interpretation
ESP	Encapsulating Security Payload
GUI	Graphical User Interface
HA	High Availability
IKE	Internet Key Exchange
IPS	Intrusion Prevention System
IPsec	Internet Protocol Security
ISAKMP	Internet Security Association and Key Management Protocol
MPLS	Multiprotocol Label Switching
NGFW	Next-Generation Firewall
PA	Palo Alto
PFS	Perfect Forward Secrecy
PSK	Pre-Shared Key
QoS	Quality of Service
SA	Security Association
SHA	Secure Hash Algorithm
SPI	Security Parameter Index
SSH	Secure Shell
TCP	Transmission Control Protocol
VLAN	Virtual Local Area Network
VPN	Virtual Private Network

1 Telia Company

Telia Company is a multinational operator founded in 1853 in Sweden as a telecommunications company; however, it found its current form after a merger with Sonera a Finnish telecommunications company in 2002. Nowadays Telia provides a wide range of services including mobile, internet, and pay television subscriptions for both private and corporate customers. Telia also provides design, implementation, and management of corporate networks and network components for companies of all sizes. Telia employs approximately 20,000 people and its headquarters is in Stockholm Sweden. Telia also has side offices and subsidiaries in Afghanistan, Denmark, Estonia, Finland, Kazakhstan, Latvia, Lithuania, Moldova, Norway, Turkey and Uzbekistan. Telia does, however, provide services for companies all around the world from North and South America all the way to Eastern Asia. (About the company 2017.)

Telia Finland

Telia Finland was formerly known as Sonera, a telecommunications company first established in 1917 as Finland's national telegraphy company, which was owned by the government till 1997. In 1998 following the privatization the company was named Sonera. In 2002 Sonera merged with Telia and obtained the name TeliaSonera Finland. Today, Telia Finland is functionally a separate company from Telia Company and operates independently. (Edwardsson N.d.)

2 Introduction and research question

The idea for this thesis originated from Secure Firewall team's need for a revamp and cleanup of their old firewall laboratory. The old laboratory environment had initially been implemented over 15 years ago and been upgraded and improved over the years without any documentation or removal of old hardware or configuration. This had lead to a great amount of unnecessary cabling and hard to comprehend configuration that did not really provide an effective platform to test new products or new features introduced by software updates.

The author and his coworker were tasked with designing and implementing a new firewall laboratory that would act as a platform for test configuration and features before implementing them in production environment for customers. This new laboratory would also work as a base to build and expand future applications. The laboratory would comprise separate compartments for Palo Alto and Check Point firewall products, and the author's task was the design and implementation of the Check Point side. In this thesis the theory behind all the products and technologies used will first be discussed followed by a detailed description about the implementation and its stages. The Public IP addresses in this document are hidden because they could be used to connect to the devices from across the public internet. These addresses could be used by a nefarious actor for an attempt to penetrate the network.

Presented as a research question the objective of this thesis was figuring out how to create a firewall laboratory environment for testing customer implementations. To achieve this goal the constructive research method was applied. Constructive research method is in its essence a way to reach a goal that is known without knowing how to get there. It is based around gathering background knowledge that is relevant to solving the problem. Then based upon that knowledge creating a course of action about how to reach the assigned goal. After creating a course of action, it must be tested for validity in the real world and corrected in the process to achieve the assigned goal. Once the goal is reached through executing the plan and applying trial and error, the method of reaching the goal must be examined for its applicability to other problems. (Crnkovic 2010.)

3 Theory

3.1 Firewalls

Traditional firewalls are a crucial part of any network providing an outer layer of protection for an internal network positioned behind the firewall. In a traditional firewall one creates a rule base that filters transport layer network traffic based on source

and destination IP addresses and the port used for that traffic. This type of firewalling relies upon the idea that all the network devices on the internal network are to be trusted; there is no protection against network attacks operating on the application layer. There are, for example, many types of attacks that can be carried out over HTTP port 80. When one allows HTTP traffic to pass through the firewall between two networks or a network and a host, one simultaneously allows any network attacks operating on port 80.

Next Generation Firewall

NGFW (Next Generation Firewall) means a new type of firewall that can inspect the payload carried inside transport layer segments. This is an important feature as attacks on a network can be invisible to a traditional firewall if hidden inside valid IP packets. The basic idea is that there is a database of known applications, vulnerabilities, exploits and malware that can be compared against incoming network traffic. This comparison can then be used to ensure, for example, that traffic to a server is indeed Office 365 and any other traffic will be dropped and marked as potentially harmful, or similarly web applications such as Facebook games can be blocked by a next generation firewall without blocking HTTP traffic all together. NGFW also allows for user authentication through AD (Active Directory) integration, so traffic to a server or to a network can be allowed for certain users but denied for others even though they are using the same IP address and Transport layer port.

3.2 Check Point R80.10

Check Point R80.10 is a part of the Check Point Infinity security architecture that combines Security Policy Management, System Health Monitoring, Log Analysis, and Multi Domain Management into one single solution. R80.10 provides zone and sub-policy based firewalling, which allows for segregation of policy, for example, to different parts of a network. R80.10 also includes session based policy changes which allow for multiple administrators to be working on the policy at the same time, and installing the policy only applies the changes made in a single session instead of

changes made by all the administrators. Database Domains in R80.10 are implemented in four distinct entities and Data Domains in three as depicted in Figure 1. Global domain is only used in Multi Domain Systems and stores the global configuration and objects. This domain is linked to every user domain as a peer domain, which means that the information stored in the global domain does not need to be installed in the individual user domains. User domains store the configuration of individual systems and they are not linked to each other allowing for complete separation of data. (Security Management Architecture Overview N.d.)

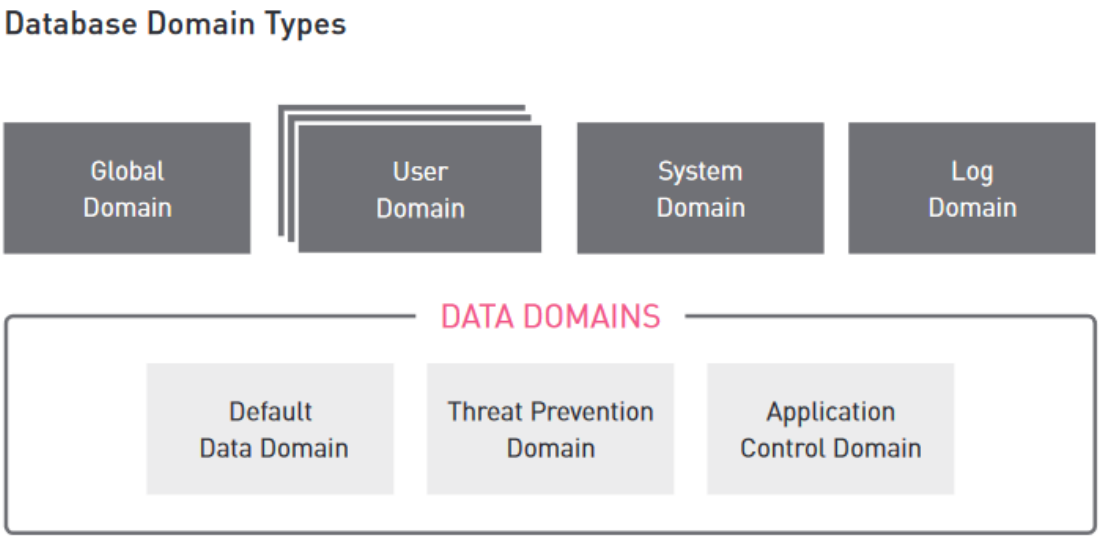


Figure 1. Checkpoint Database Domain types (Security Management Architecture Overview N.d., 7)

Check Point R80.10 provides easy inspection of the system performance status from the graphical user interface as seen in Figure 2. This status information can also be queried from the database using custom parameters to view all the necessary statistics on the same graph.

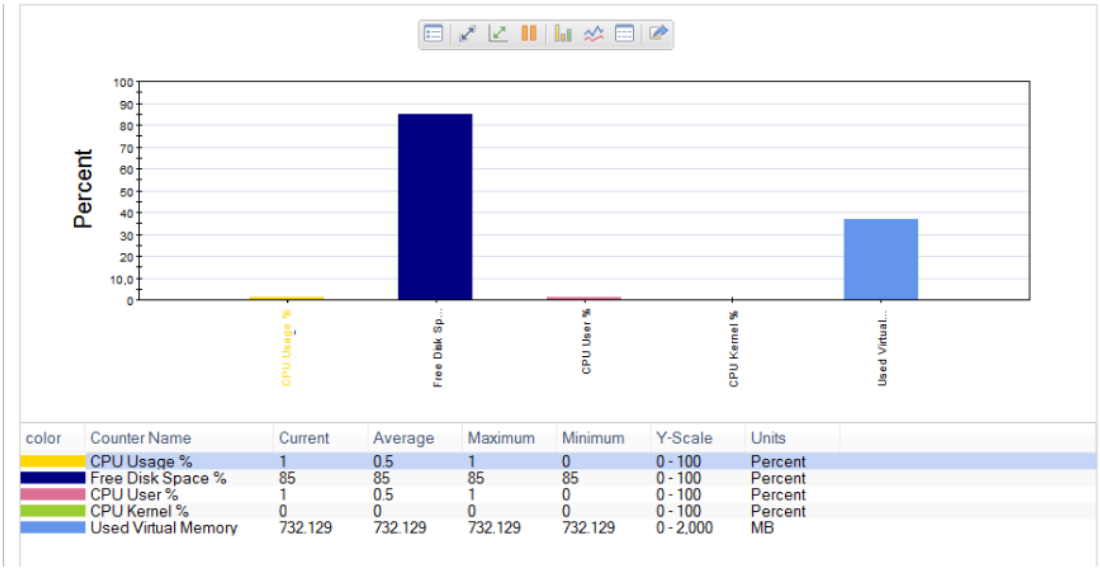


Figure 2. System performance

Status monitoring on R80.10 can provide the user a comprehensive look into high level information about their system as can be seen in Figure 3. These statistics could be used in determining whether or not a customer firewall needs to be upgraded or in assigning QoS (Quality of Service) markings for different types of traffic in the customer network.

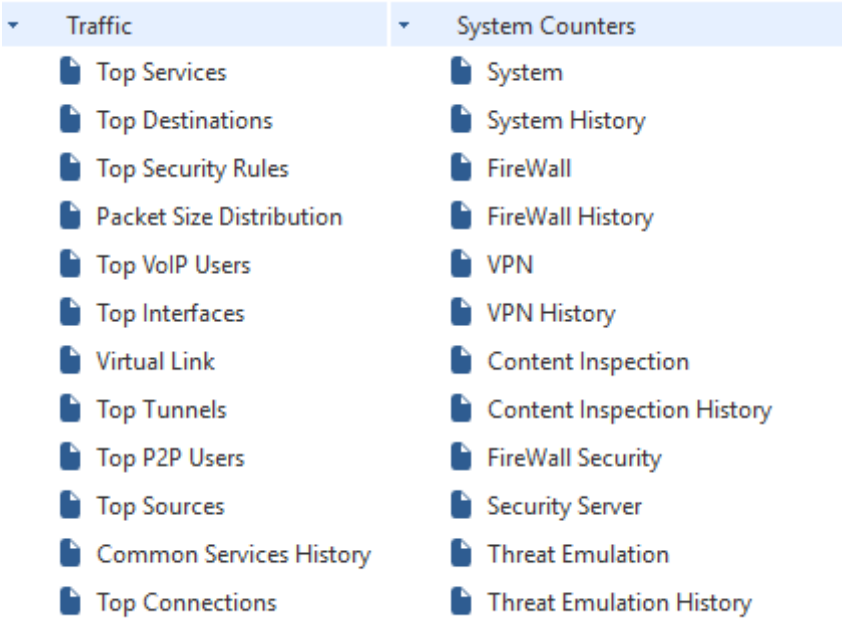


Figure 3. Check Point R80.10 status monitoring

3.3 Network Design

Basic building blocks of network design consists of four individually definable entities. First of which is scalability that could be understood as the ability of the network to grow without impeding on quality of service for the end users or applications. Scalability can be achieved by using appliances that are powerful enough to allow for at least doubling of current traffic levels without creating an issue and applying protocols and technologies which can handle growth of the network. Example of using protocols that allow the network to grow would be using static routing versus a dynamic routing protocol. Static routing can be used to manage routing in a small network but comes unmanageable very quickly as device numbers on the network increase.

Second building block is availability which entails that the network must be operational for a defined percentage of time depending on the use case. In a smaller office network where people only work during the day availability requirements are lower than in a large company network in a manufacturing plant for example where machines are running 24 hours a day and every second they are not running money is being lost. Availability can be achieved using a multitude of techniques including duplicate devices on parts of the network where a single point of failure is existent and correctly gauging the amount of throughput needed that even during times of maximum traffic, there will not be any congestion.

Third part is security which simply means securing the network in a way that no one cannot access devices or segments of the network they are not supposed to. This point is especially important on the perimeter of the network as most nefarious attempts at penetrating a network come from the outside. Securing the network does not only mean securing it on the logical level but also making sure that devices cannot be accessed physically by any one other that people with permission to do so. It is necessary that the security of a network is a major point beginning from the first parts of the design and not an afterthought, as adding security to a completed network can be very difficult. Position of security devices on the network and using technologies that are on par with the times is paramount in creating a safe network.

Lastly the administrators must be able to manage the network in an efficient way for it to be functional. Most of the time this means creating the means for remote management of all the devices on the network or having a centralized management portal. Networks often span large geographical areas which means that local administration of devices would cause massive overhead on network changes, this could have catastrophic consequences in the event of a security issue emerging in the network. (Introducing Network Design Concepts N.d.)

3.4 VPN

VPN or Virtual private network is an encrypted virtual connection between two networks that allows traffic to flow over public networks without anyone being able to see what the traffic contains inside. VPN can be thought of as a tunnel or a pipe that is created between two or more endpoints and is used to provide security for inter network traffic. After a VPN tunnel is formed between the endpoints, it functions how an actual connecting cable between two devices would work. All the devices between the endpoints disappear from their point of view and they communicate straight to each other. In more technical terms, tunneling in a VPN means adding new headers to the IP traffic which hide the traffic inside and can only be opened by the endpoint with which the tunnel is formed. Traffic does still go through the same route to the other endpoint of the tunnel as it normally would without a VPN tunnel; however, now the devices in between that are handling the routing and switching cannot see what is inside the packets. (Makati 2013.)

3.5 IKE

IKE or Internet Key Exchange is a protocol used to negotiate the protocols, encryption, algorithms and keys for a VPN protocol. IKE consists of two phases, first of which uses ISAKMP to establish a secure connection between the two devices to allow for secure communication. Phase two negotiates the parameters for the actual security association for securing the data stream. (Harkins & Carrel 1998.)

3.5.1 ISAKMP

ISAKMP or Internet Security Association and Key Management Protocol is a protocol for negotiating security associations for different types of security services on the network, transport or application layers. ISAKMP can be used to establish, modify or delete security associations. ISAKMP is different from a key exchange protocol as it defines the payloads used to negotiate key authentication and generation; however, it does not care about the algorithms used to generate the keys. ISAKMP can therefore be used as common grounds upon which different devices from different manufacturers can establish a security association. Five values are most commonly negotiated during the creation of an ISAKMP Security Association. These are authentication algorithm, Diffie-Hellman group, encryption algorithm, HASH algorithm and the lifetime for the security association. ISAKMP provides cookies to prevent DoS as seen in Figure 4**Virhe. Viitteen lähdettä ei löytynyt.**. Cookies are a hash over the values of the source and destination IP addresses and ports and a secret value that is created locally on each device. If the cookies do not match, the traffic gets dropped. (Maughan, Schertler, Schneider, & Turner 1998.)

<table>
<tr><td colspan="5">Initiator Cookie(64bit)</td></tr>
<tr><td colspan="5">Responder Cookie(64bit)</td></tr>
<tr><td>Next Payload(8bit)</td><td>MjVer(4bit)</td><td>MnVer(4bit)</td><td>Exchange Type(8bit)</td><td>Flags(8bit)</td></tr>
<tr><td colspan="5">Message ID(32bit)</td></tr>
<tr><td colspan="5">Length(32bit)</td></tr>
</table>

Figure 4. ISAKMP Header (Wen, Meng, Fend & Tang 2017)

Next Payload field informs the protocol about the type of data contained in the next payload. Security Association, proposal and transform payloads are used in negotiat-

ing the SA parameters. Identification, Hash and Signature payloads are used to authentication and integrity checks. There are sixteen types of payloads in total that can be transmitted, and they are as follows.

Next Payload Type	Value
NONE	0
Security Association (SA)	1
Proposal (P)	2
Transform (T)	3
Key Exchange (KE)	4
Identification (ID)	5
Certificate (CERT)	6
Certificate Request (CR)	7
Hash (HASH)	8
Signature (SIG)	9
Nonce (NONCE)	10
Notification (N)	11
Delete (D)	12
Vendor ID (VID)	13
RESERVED	14 - 127
Private USE	128 - 255

Major and minor version fields contain the information about the version of ISAKMP being used. Exchange type field dictates what type of ISAKMP exchange is being conducted. There are nine types of exchanges that can be done by ISAKMP and they are as follows:

Exchange Type	Value
NONE	0
Base	1
Identity Protection	2
Authentication Only	3
Aggressive	4
Informational	5

ISAKMP Future Use	6 - 31
DOI Specific Use	32 - 239
Private Use	240 - 255

Flags field can hold three types of additional options specified for an ISAKMP exchange. These are E for encryption, C for commit, or A for authentication only. Encryption flag means that the algorithm specified by the ISAKMP security association encrypts all the traffic. Commit flag indicates to the device that key exchange synchronization is ensuing. Authentication flag is used in conjunction with informational exchanges (5) containing notification payloads (11).
(Maughan, Schertler, Schneider, & Turner 1998.)

3.5.2 Oakley

Oakley is a Key Determination Protocol for deciding keying material to be used in creating a secure connection between devices. Oakley uses Diffie-Hellman to allow for two devices to concur on a shared secret across a non-secure communications channel that can be used to encrypt authentication traffic or data. Oakley enables secure communications to be established with perfect forward secrecy, which ensures that even if a key for a session were to be cracked by a nefarious actor, traffic from other sessions would not be compromised. (Orman 1998.)

3.5.3 SKEME

SKEME works in three phases that are SHARE, EXCH, and AUTH. The first phase SHARE is used in creating a shared key between peers by utilizing their public and private keys. The devices do not share their private keys but use them in conjunction with the peer devices public key in order to create a shared secret key. This is accomplished by using a long modular exponentiation of the private and public keys. The second phase EXCH is used to facilitate the change of Diffie-Hellman exponent values. The final phase AUTH authenticates the exponents shared in the EXCH phase by

using the shared key negotiated in the SHARE phase. This process provides SKEME with strong perfect forward secrecy. (Doraswamy & Harkins 2002; Krawczyk 1996.)

3.6 IPsec

3.6.1 Protocol mode

IPsec operates in two distinct modes, tunnel mode and transport mode. Tunnel mode is usually configured between two IPsec peers that are most often routers or firewalls. These devices form a tunnel between each other and encrypt the IP packets coming from a network behind them by encapsulating the traffic in ESP header and trailer or AH header and adding a new IP header. Transport mode is best utilized for end-to-end traffic coming from a client going towards a server, for example. The main difference between the two is that in transport mode the original IP header remains intact and is used for traversing through the network instead of using a new added IP header. (Frankel & Krishnan 2011; IPsec VPN Overview 2018.)

3.6.2 ESP

ESP or Encapsulating Security Payload allows for complete encryption of IP traffic encapsulating it in ESP packet. ESP adds a new IP header for the traffic to be used in navigating through public internet or whatever network is in between the sending and receiving end of the VPN tunnel. ESP uses Security Parameter Index a 32-bit random value to identify the security association that the incoming packet is intended for. The SPI field in an ESP header is mandatory for the operation of ESP and must be included. ESP also uses 32-bit Sequence Numbers or 64-bit Extended Sequence Numbers that are used as counters for the traffic passed on that security association. The payload field in an ESP packet carries the entire original IP packet in encrypted format. Padding for ESP is only necessary when the encryption algorithm requires the data to be of a specific length. Pad Length field indicates the number of padding bits used before it. Next header field informs the receiver of what type of traffic is encrypted inside the ESP packet. Common values for the next header field are 4 which

stands for IPv4, 41 which stands for IPv6 and 6 which stands for TCP (Transmission Control Protocol). In Figure 5 one can see an ESP header. (Kent 2005.)

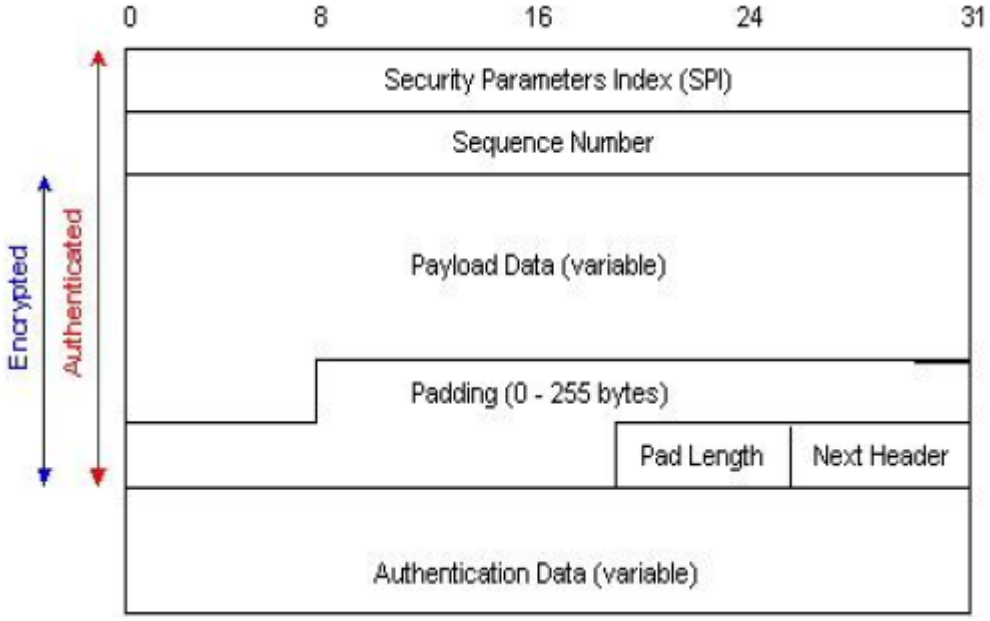


Figure 5. ESP Header (Leaf 2015)

3.6.3 Authentication Header

AH (Authentication Header) is a way to achieve peer authentication and data integrity in IPsec communication. AH provides this by utilizing a shared secret value and applying that to a hash function alongside the IP addresses and data being sent. AH also allows for protection against replay attacks similarly to ESP by using sequence numbers in the header. In Figure 6. AH Header (Doraswamy & Harkins 2002, 38) is the header format of AH. What AH does not do, however, is to encrypt the data while it is passed across the network. This means that AH is really only useful in cases where the actual data being sent is only valuable to the receiver, such as sensor data being sent to a control unit. Especially when small sensor units most likely will not have a strong microprocessor on them and doing encryption would add unnecessary latency to the connection. This data on its own might not provide anything to a nefarious actor so them getting their hands on it is not a priority. This type of data needs to be authenticated and checked for integrity; however, as faulty or intentionally modified data could cause major harm, e.g. the data coming from a heat sensor used to control the cooling system of a manufacturing plant. (Frankel et al. 2011.)

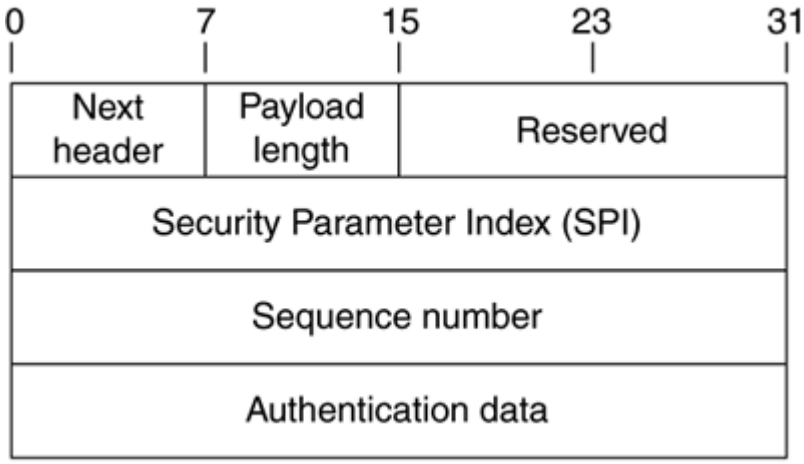


Figure 6. AH Header (Doraswamy & Harkins 2002, 38)

3.7 Virtual Local Area Network (VLAN)

Virtual Local Area Networks are virtual representations of Local Area Networks operating on the OSI layer 2 whereas an actual Local Area Network operates on the OSI layer 1. This allows for virtual segregation of networks existing in shared physical media. VLANs function by adding tags to layer 2 Ethernet frames traversing the network. These tags are applied on network switches or routers and then stripped off when they reach their destination. Where the frames then travel on the network is dictated by the tag associated with it. For every VLAN there is a broadcast domain where all the frames tagged with that VLAN number are sent. VLANs are often used to divide a network between two groups, for example, students and faculty. Which is useful as students most likely should not have access to the same resources as teachers do, but it would not be very cost effective to construct different physical networks for students and teachers to use. Using VLANs allows for teachers and faculty members to use the same physical network as students but be virtually separated. VLANs also provide protection against problems caused by broadcast traffic. As broadcast traffic is sent out of every interface on a switch except the one it came from and replicated by every subsequent switch it can create much traffic on a network, or in worse case scenarios even broadcast storms, which are caused by loops in the network. VLANs can be used to break up large broadcast domains as traffic is only sent to interfaces belonging to that VLAN.

4 Designing the environment

The design started from a conversation with the representatives of the security services team and discussions about what they wanted to be created. From the initial conversations, the topology was simple with one Palo Alto firewall protecting the laboratory from traffic coming from the internet, and then having two independent firewall environments for Check Point and Palo Alto. As stated in the theory section 3.3 security of a network was one of the first approaches taken in designing the topology. For a laboratory environment a protecting firewall on the perimeter of the network was sufficient, as the only way to enter the network was through that firewall. Physical security of the laboratory environment was established by the location of the laboratory, which was located behind at least two keycard locked reinforced doors no matter which way a person would approach the laboratory. The first draft of the laboratory topology was hand drawn on a piece of paper which can be seen in appendix 7. The hand drawn topology was translated to a digital version which can be seen in Figure 7. At that stage, there the idea was that the laboratory environment would have two internet access points primary and a backup, BGP configuration between the Check Point and Palo Alto environments and two VMware platforms.

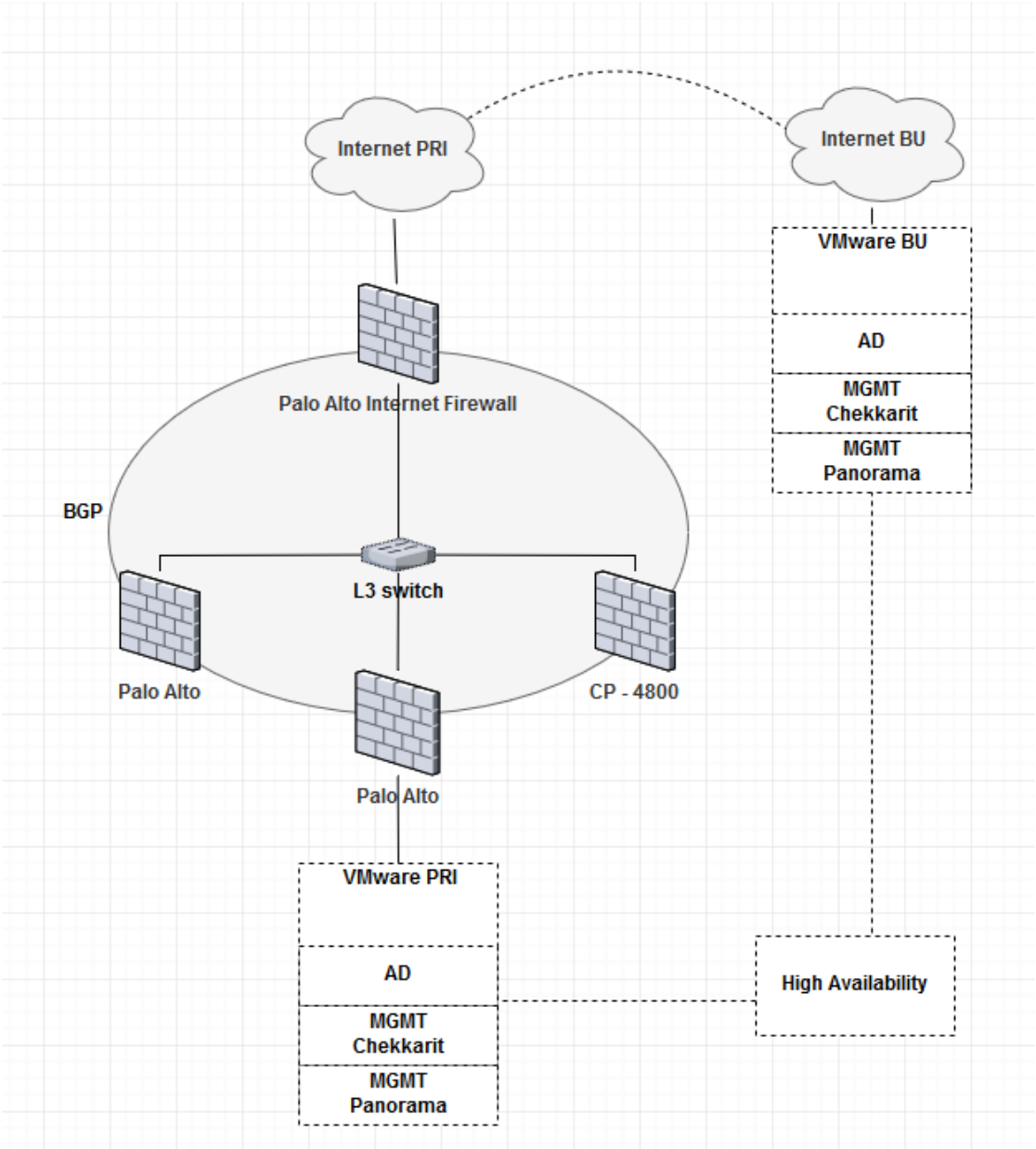


Figure 7. First draft of the topology

This type of topology was chosen as it mimicked actual customer environments and could be used to test the compatibility of Check Point and Palo Alto devices. In theory section availability was named as one of the four defining parts of designing a network and in actual production environments many of the devices designed for the laboratory network would be duplicated to provide required availability to the network. However, as the environment would be running as a testing laboratory availability was not a high priority. The network could be down for a week without any major consequences. The management of the firewall devices was to be located on a

VMware virtualization platform device that was already configured in the old laboratory.

The idea for two internet access points was discarded as the other connection was hidden behind an existing old virtual firewall which was not going to be a part of the new laboratory environment. The design was altered to discard the other internet access point as illustrated in Figure 8. BGP was still a part of the design and was supposed to be handling the routing for the environment.

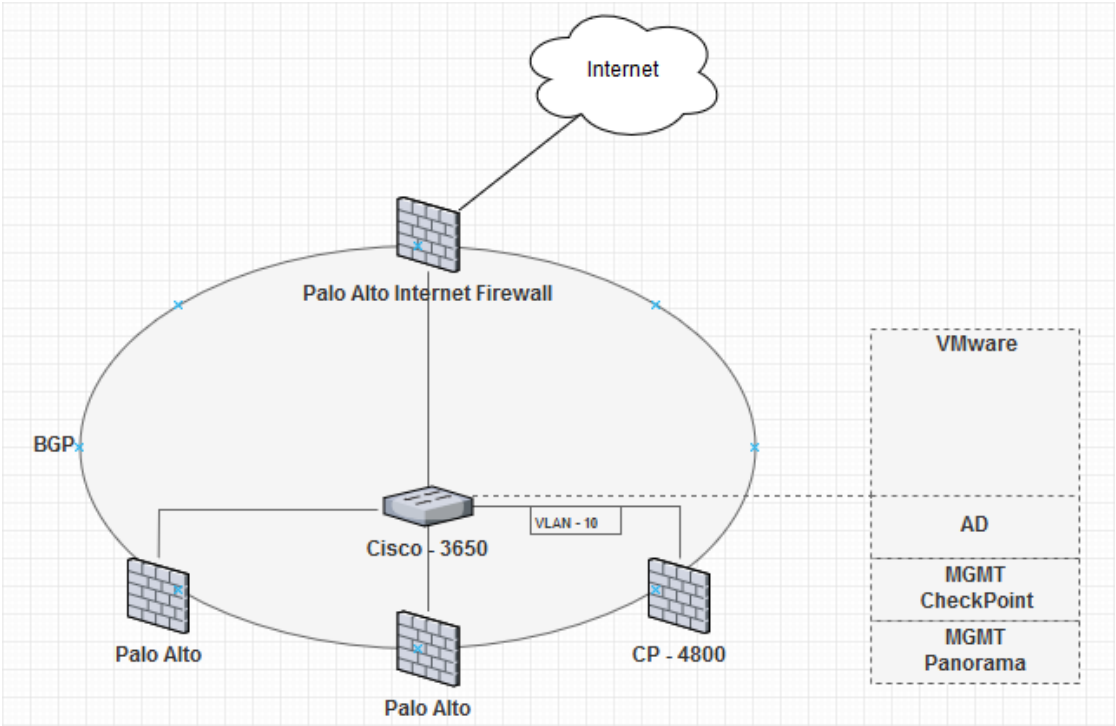


Figure 8. Second draft of the topology

The laboratory would be divided between the Check Point and Palo Alto environments using virtual local area networks. Management traffic and simulated customer traffic would also be divided into different virtual local area networks as this would provide segregation of network segments that were not supposed to have direct contact with each other, which would also improve security of the network. Connection between the Check Point and Palo Alto customer networks was going to achieved using a VPN tunnel in a same way two companies from two different telecommunications operators would connect their networks. After this point in the design it was decided that BGP was not an integral part of the laboratory and as the environment

was not going to be that large, routing could easily be done using static routes. This decision was based on the scalability section of the network design theory, the amount of network segments and devices that were going to be used now or in the future in the laboratory environment could be handled using static routing. Dedicated switches were also to be added for both firewall environments as seen in Figure 9. In addition, a new Check Point cluster was added for the Check Point environment to test high availability and other clustering functions.

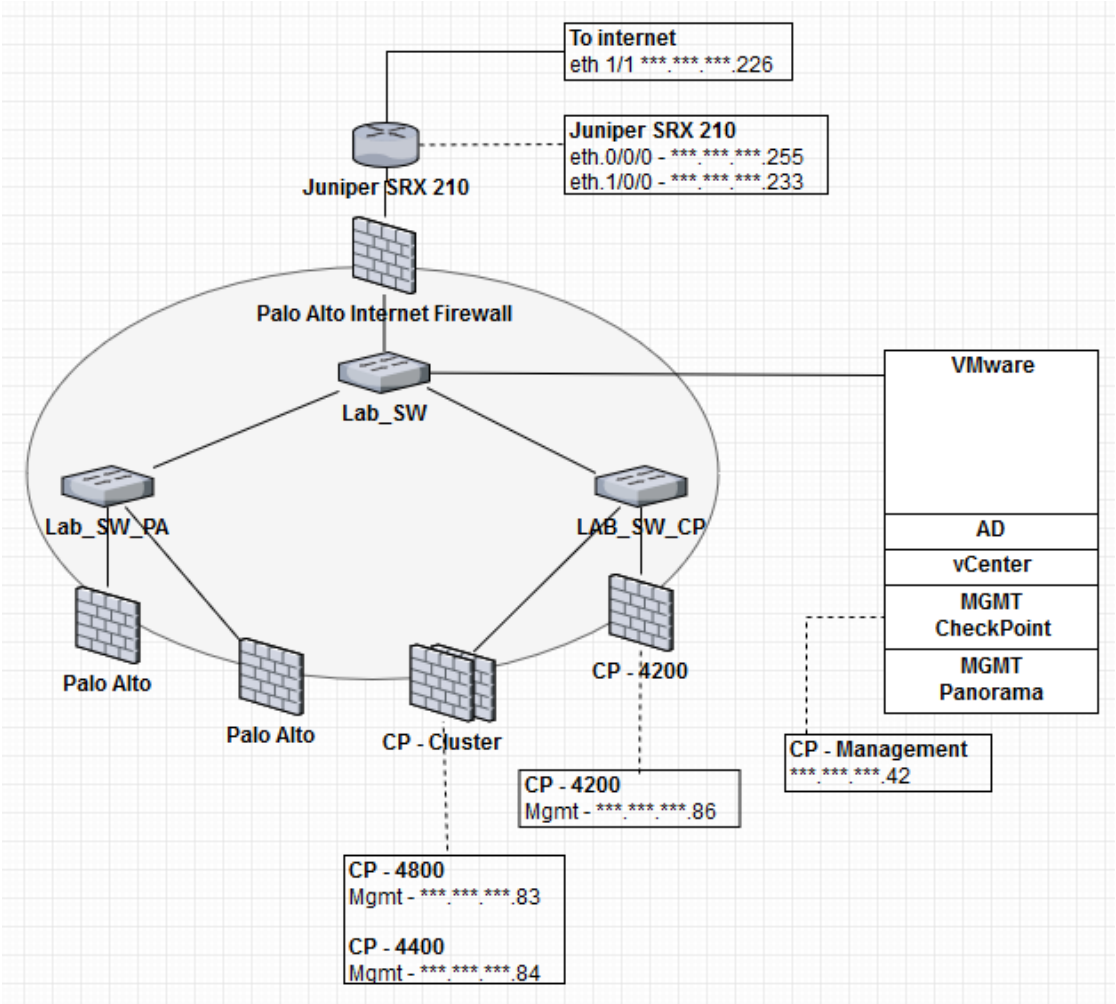


Figure 9. Third draft of the topology

At this point as the number of switches in the laboratory design had increased and to allow remote management by allocating public IP addresses for all these would be a waste of addresses and a private switch management network was decided to provide management. This required a dedicated jump machine for the switch management network. Public IP address would be allocated to it and private IP addressing

would be used for the switch management network as illustrated in Figure 10. Both decisions were based on the theory of network design. Increasing the number of switches allowed for the network to be expanded to a much larger degree than is ever likely to happen. Configuring a management network with a dedicated management device for the switches provided remote manageability that is critical for the usability of the network. The jump machine was going to be a Linux machine running on the VMware virtualization platform. This Linux was going to have a public IP address and could be connected to over the internet, and an SSH session could be established to connect to switches beyond that.

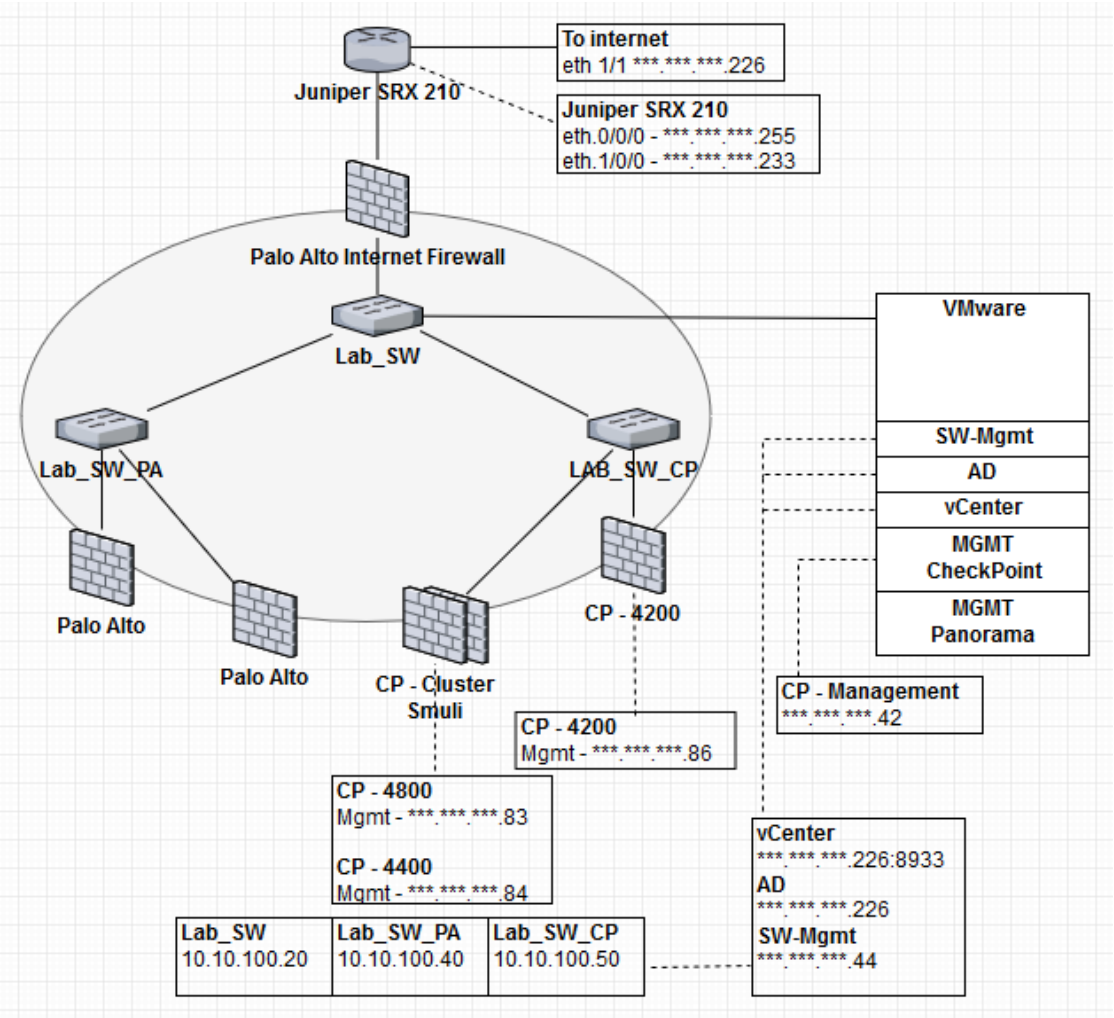


Figure 10. Final draft of the topology

5 Implementing the environment

5.1 Objective and first steps

The implementation of the firewall laboratory consisted of three major steps which were configuring the firewall devices, configuring and creating the physical and logical topology of the switch network, and configuring the management devices. Completing these steps created a functioning firewall laboratory that could be remotely managed and used for testing. Configuration for all the switches and Check Point devices described in the implementation can be found in appendices 1-6.

The first part of the implementation was clearing some of the obsolete network devices, cabling and boxes from the laboratory as seen in Figure 9 (Figure 11. Old laboratory). Most of the devices left in the racks were either not working anymore or were obsolete and would be thrown away. Most of the devices, however, were still left in place after some initial cleanup as it was not clear at that point which of them would be saved and which would be dumped. As this was the case, the new laboratory environment was built in an empty rack and the old equipment left in place.

Figure 11. Old laboratory

5.2 Check Point firewall gateway configuration

Designing a network is beneficial to start from the outside in and first think of a secure perimeter and designing the rest of the pieces inward, as can be seen done in the Network Design section of the thesis, where one of the first stable design elements that did not get altered was the perimeter firewall. However, implementing the network is better to start from the inside and work towards the edge of the network, because first parts of the implementation often require the use of console connections. Building from the perimeter inwards towards a device that does not yet exist could cause problem if the topology or configuration need to be changed when installing the final devices. Therefore, the next step for the implementation was the installation of the Check Point R80.10 software on Check Point 4800 and 4400 appliances. This was done via a console connection after booting the appliance. After that, the first time wizard for the device was run locally through a browser. The IP address and gateway addresses of the appliance were set up for the device as seen in Figure

12, and the connection could be tested through the management interface using the Check Point R80.10 GUI. To allow for clustering of the devices a Synchronization interface was configured using 1.1.1.0/30 network. This allows for the devices to communicate about their current state during clustering. Sync interfaces were connected through the switch and a dedicated vlan 30 was chosen for that traffic.

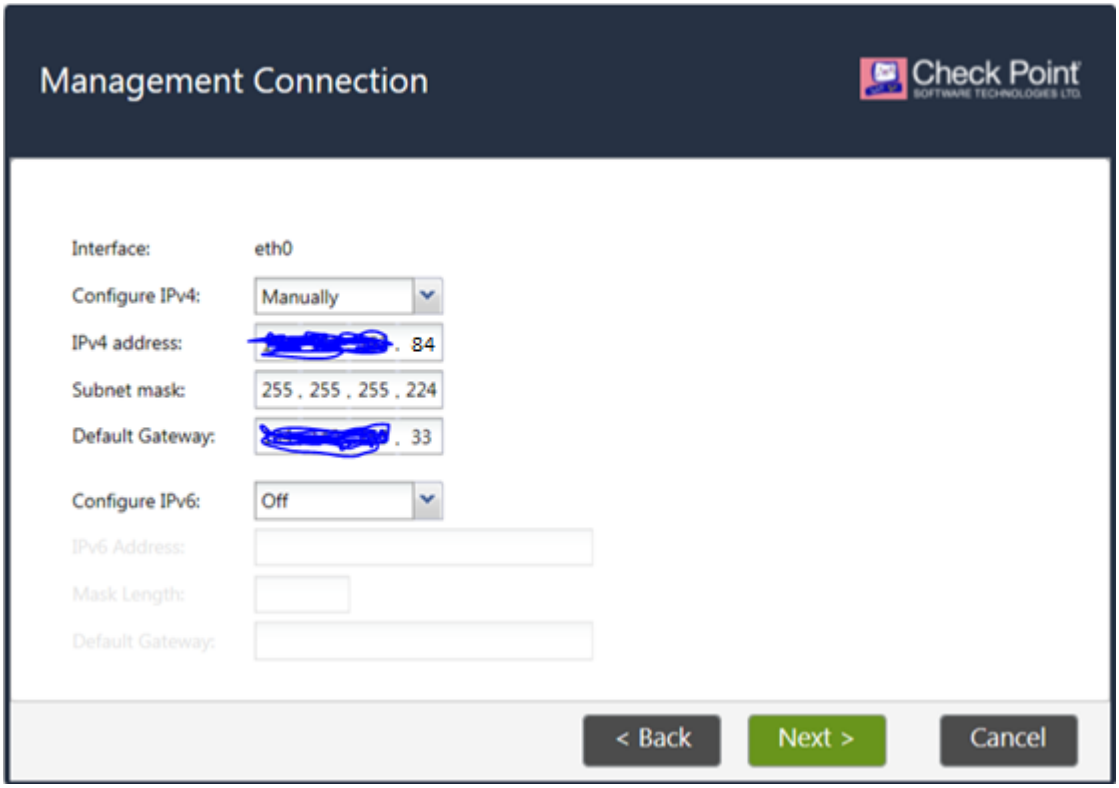


Figure 12. CheckPoint R80.10 first time wizard

The next step in building the laboratory environment was creating a connection outward from the gateway appliances toward the edge of the network, continuing with the implementation trajectory of building from the inside out. Connection from the gateway appliances after they were up and running to the Palo Alto perimeter firewall was established using a Cisco 3650 switch and configuring vlan 10 for the Check Point management traffic. Some public IP address blocks were already routed towards the laboratory over the internet so addresses for the gateway appliances were chosen from these blocks to allow for remote SSH connections to them after configuring the management device configuration.

5.3 Check Point management device configuration

To achieve the manageability required for a network defined in the Network Design theory section a management server was installed on the VMware platform as described in the network design section on pages 18-19. This was going to be used to manage the gateway. First, an ISO image of the management software was downloaded from Check Point website and the connection to the VMware platform was established via remote desktop connection. The computer's local hard drive was mounted in the remote desktop session to allow the usage of the downloaded ISO image. VMware virtualization platform was connected to the Lab_SW switch using vlan 50 and connected to the Palo Alto internet firewall from the Lab_SW switch. Routing the traffic between the gateways and management device was done by the Palo Alto internet firewall.

The installation of the device in VMware works the same way it does elsewhere. The only difference between installing a gateway and installing a management device comes in selecting the product type as seen in Figure 13. For a standalone installation both the gateway and management would be chosen for the device. For a device that is going to be a part of a cluster the option "Unit is a part of a cluster, type:" would be chosen. Otherwise the device cannot be added to a cluster.

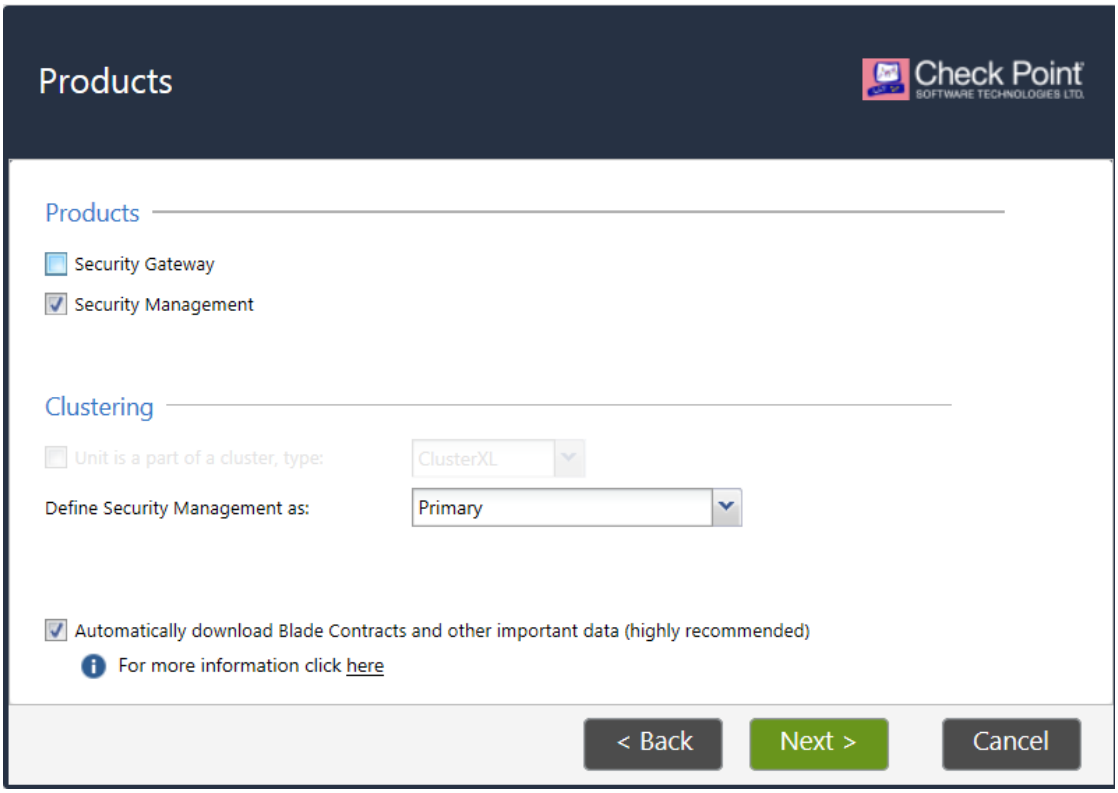


Figure 13. Check Point R80.10 management installation

After the management device was installed on the virtual machine the gateways were used to create a cluster in the management device. Both gateways were connected to the management server and secure communication was established using a onetime activation key seen in Figure 14 . This connection confirmed that the devices are able to communicate, and routing is working between them.

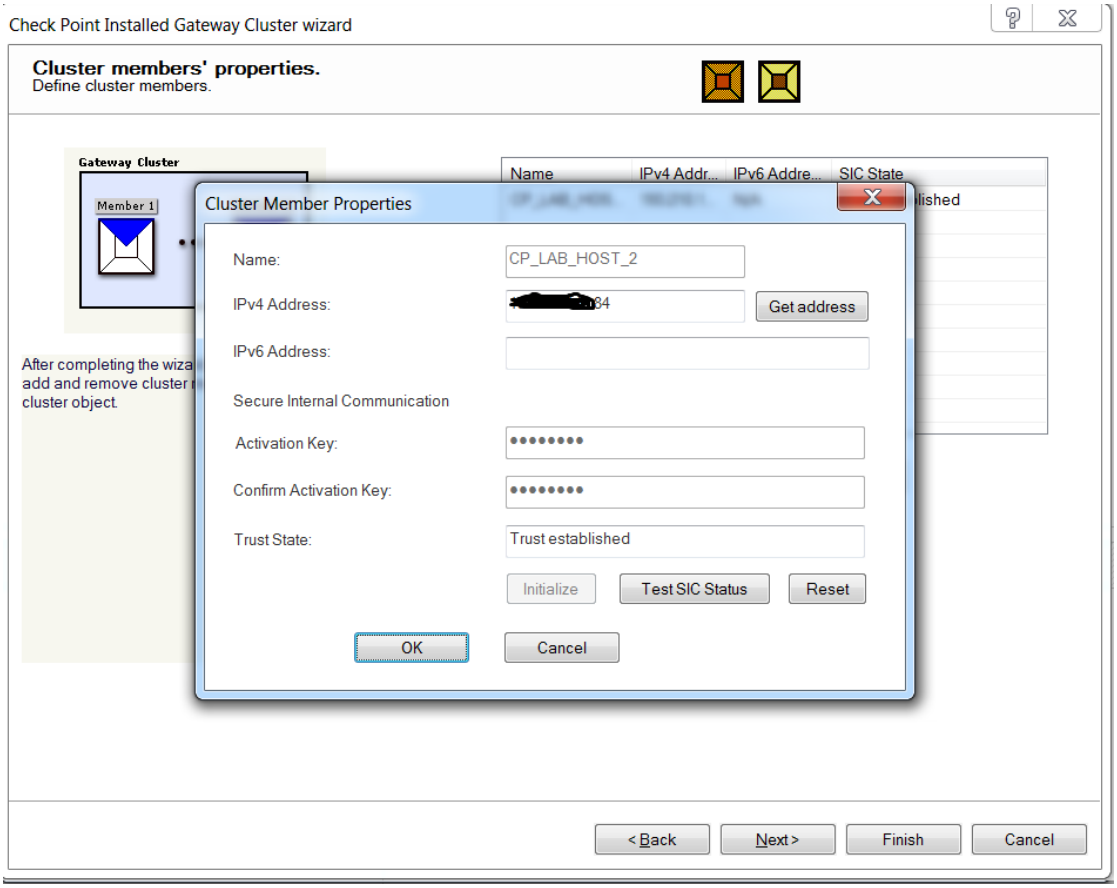


Figure 14. Creating a cluster on the management device

Initially after creating the cluster on the management device the cluster members were seeing each other in the down state and therefore clustering was not working as seen in Figure 15.

```
CP_LAB_HOST_2> cphaprob state

Cluster Mode:   High Availability (Active Up) with IGMP Membership

Number      Unique Address  Assigned Load   State

1           1.1.1.1         0%              Down
2 (local)   1.1.1.2         100%            Active Attention

Local member is in current state since Tue Aug 14 15:14:25 2018
```

Figure 15. Cluster states

After extensive searching the problem was determined to be differentiating configuration of firewall instances on the 4800 and 4400 appliances. The 4400 was configured with 2 firewall instances and the 4800 appliance was configured with 3 firewall

instances as depicted in Figure 16. After correcting this error, the cluster went to active / standby state and was working correctly.

```
Configuring Check Point CoreXL...
=================================


CoreXL is currently enabled with 3 IPv4 firewall instances.


(1) Change the number of firewall instances
(2) Disable Check Point CoreXL

(3) Exit
Enter your choice (1-3) : 1

This machine has 4 CPUs.

Note: All cluster members must have the same number of firewall instances
enabled.

How many IPv4 firewall instances would you like to enable (2 to 4) [3] ? 2

CoreXL was enabled successfully with 2 firewall instances.
Important: This change will take effect after reboot.
```

Figure 16. Firewall instances

Once the firewall cluster was created on the management server a firewall policy was created an pushed on the clustered devices. As depicted in Figure 17 traffic between the cluster members, management, and sync interfaces was allowed and traffic from SU4IPM which contained the VPN networks for the firewall experts was allowed to connect to the networks from which IP addresses for the devices in the laboratory were chosen from.

No.	Name	Source	Destination	VPN	Services & Applications	Action
1		CP_CLU_HOST_1 CP_CLU_HOST_2 CP_LAB_CLU CP_LAB_MGMT sync_1.1.1.0m30	CP_CLU_HOST_1 CP_CLU_HOST_2 CP_LAB_CLU CP_LAB_MGMT sync_1.1.1.0m30	* Any	* Any	Accept
2		SU4IPM	net_ m28 net_ m27 net_ m26	* Any	* Any	Accept
3		net_ m28 net_ m27 net_ m26	SU4IPM	* Any	* Any	Accept
5	Cleanup rule	* Any	* Any	* Any	* Any	Drop

Figure 17. Firewall policy

After that point, the address for the appliances had to be made accessible over the internet to allow remote management of the device. This was achieved by adding a static route to the internet router behind the internet firewall as seen in Figure 18.

```
[edit]
kbs5456@jklsoneral4902# set routing-options static route [redacted].84/29 nexthop [redacted].226
```

Figure 18. Adding route to Juniper router

Creating the simulated customer network behind the firewall gateway devices was achieved by configuring interfaces on the gateways and creating a new customer traffic vlan on the switches in between. Virtual local area network for the Check Point customer traffic was chosen to be vlan 20. Network for the customer were chosen from private network segments, this meant they could not be routed over the internet if implemented on an actual development platform and no routing for these networks was added on the Palo Alto internet firewall.

5.4 VPN tunnel configuration for customer traffic

To allow for the traffic to traverse over the simulated internet a VPN tunnel was constructed between the Palo Alto and Check Point environments, as described in the network design section on page 19. First part in creating a VPN tunnel was creating a remote gateway object that represented the Palo Alto firewall. Once the gateway object was created, an encryption domain group was created and linked to that gateway object. The customer network 192.168.30.0/24 located behind the Palo Alto firewall was defined in the encryption domain. An encryption domain was also created for the Check Point firewall and customer network 192.168.20.0/24 located behind the Check Point was defined in there. In the VPN tunnel configuration, the Check Point cluster object was defined as the center gateway and the Palo Alto gateway object was added as a satellite gateway as seen in Figure 19.

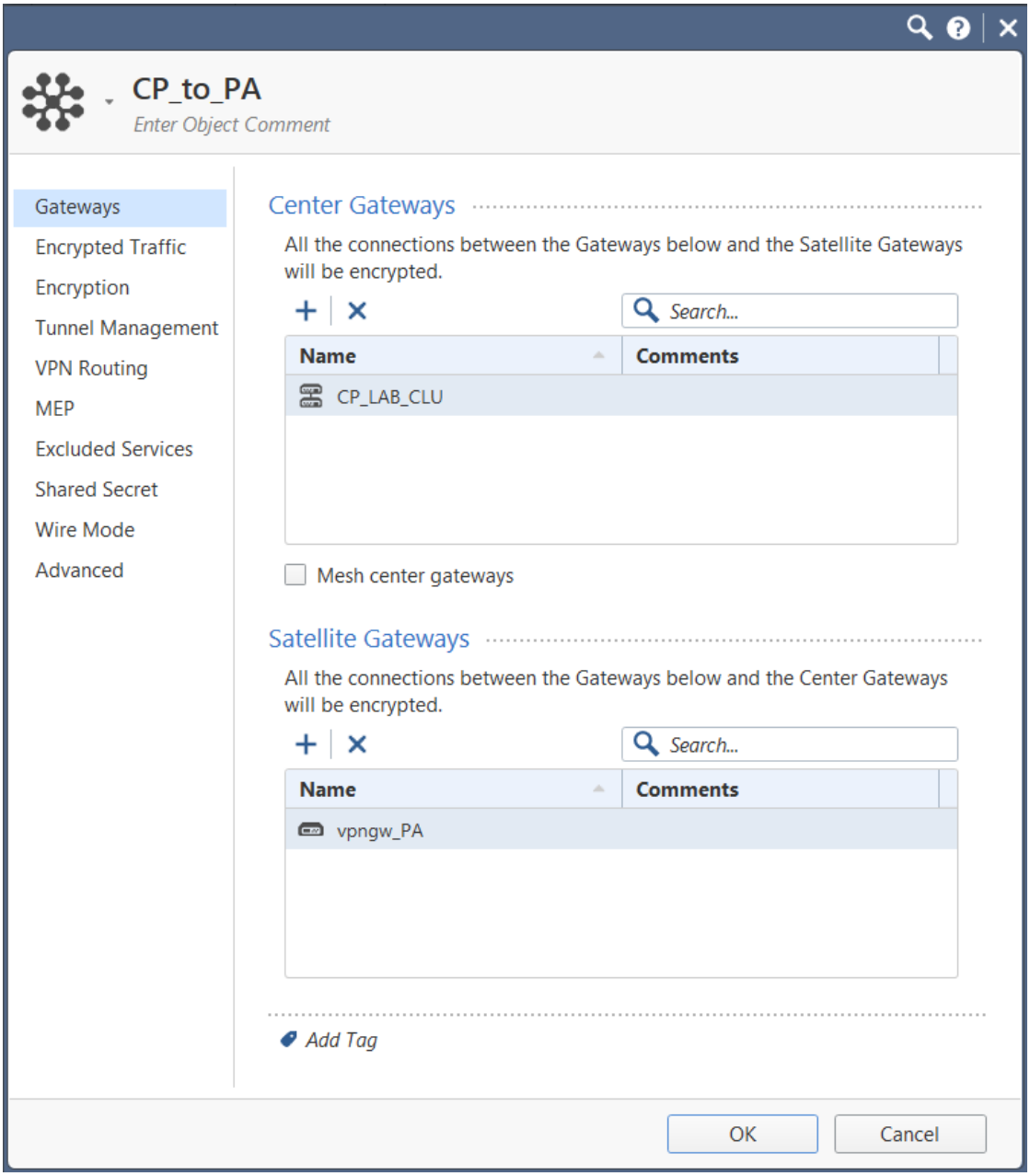


Figure 19. VPN configuration 1

Next step was to specify the encryption and authentication algorithms to be used and Diffie Helman groups to be associated with those. Figure 20 illustrates the encryption and authentication configuration for the VPN tunnel. For both encryption and authentication AES256 (Advanced Encryption Standard 256) was chosen as that is an encryption algorithm sanctioned by the Finnish Communications Agency (Kryptografiset vahvuusvaatimukset luottamuksellisuuden suojaamiseen – kansalliset suojaustasot 2018). Diffie Hellman groups were chose as DH 5 for IKE phase 1 and DH 14 for phase 2 as those were found to be compatible between Check Point and Palo

Alto. PFS or Perfect Forward Secrecy was used for the VPN connection as this increases the security of the connection substantially as described in the theory section 3.5.2 and testing whether this option worked between Palo Alto and Check Point devices was therefore essential.

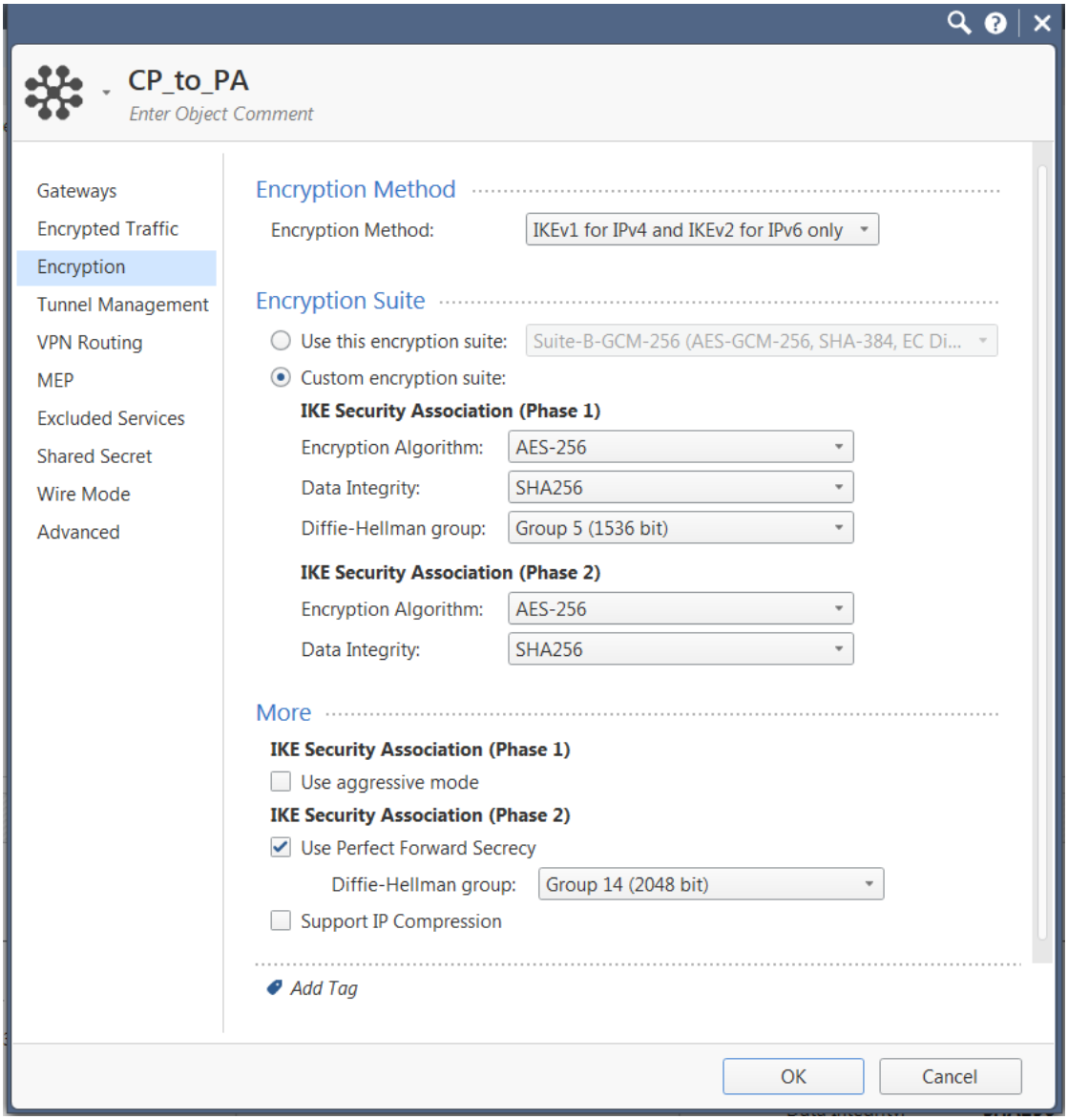


Figure 20. VPN configuration 2

Final part in configuring the VPN tunnel was to designate IKE phase 1 and phase 2 timeouts after which the tunnel needed to be renegotiated. For phase 1 the timeout was set as 1440 minutes or 24 hours as this was a laboratory environment there is no real need to negotiate the IKE security association very often. For the phase 2 or IP-

sec security association timeout was chosen as 3600 seconds or 1 hour as seen in Figure 21. IPsec security association is usually negotiated more frequently than IKE security association as it takes less resources to negotiate the phase 2, and since PFS is used for this VPN connection if a key was cracked by an attacker only 1 hour of traffic would be compromised.

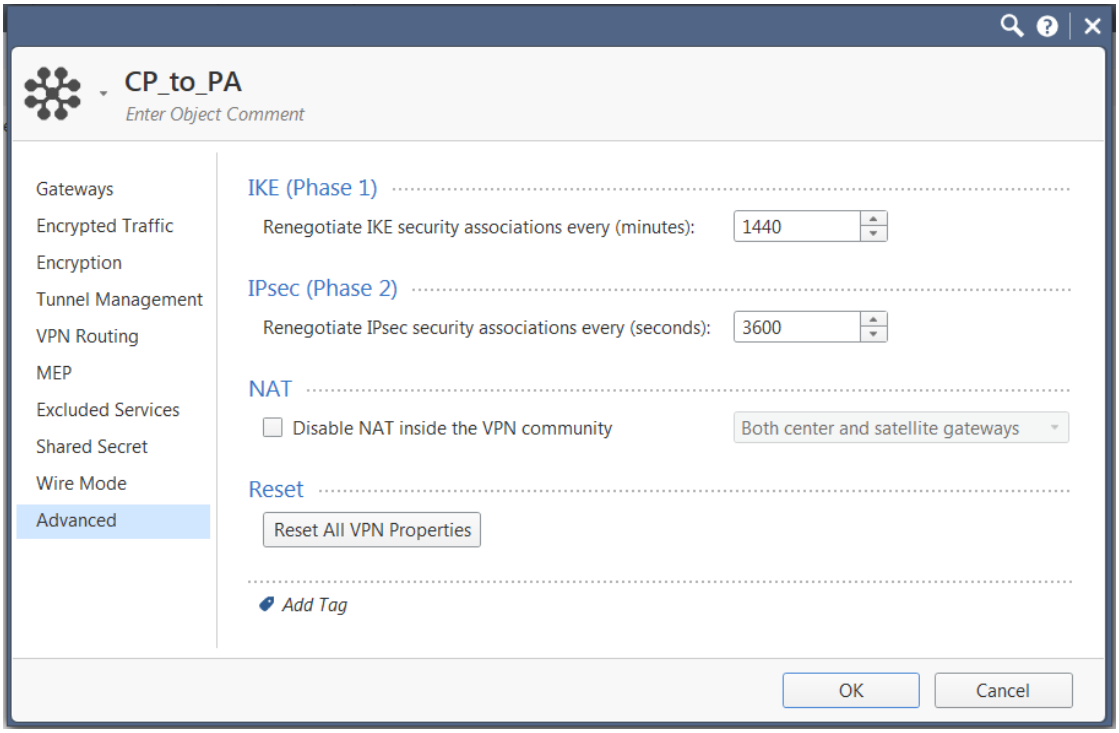


Figure 21. VPN configuration 3

After configuring the VPN tunnel traffic needed to be allowed through in the firewall policy. As depicted in Figure 22 traffic was allowed through the firewall between the gateway objects and the encryption domains using any service and in both directions.

4	PA-L2L	vpngw_PA CP_LAB_CLU encdom_CP encdom_PA	vpngw_PA CP_LAB_CLU encdom_PA encdom_CP	CP_to_PA	* Any	Accept

Figure 22. Firewall VPN policy

As Figure 23 illustrates the VPN tunnel was successfully established between the Check Point and Palo Alto environments.

```
[Expert@CP_LAB_HOST_1:0]# vpn tu

**********     Select Option     **********

(1)                List all IKE SAs
(2)              * List all IPsec SAs
(3)                List all IKE SAs for a given peer (GW) or user (Client)
(4)              * List all IPsec SAs for a given peer (GW) or user (Client)
(5)                Delete all IPsec SAs for a given peer (GW)
(6)                Delete all IPsec SAs for a given User (Client)
(7)                Delete all IPsec+IKE SAs for a given peer (GW)
(8)                Delete all IPsec+IKE SAs for a given User (Client)
(9)                Delete all IPsec SAs for ALL peers and users
(0)                Delete all IPsec+IKE SAs for ALL peers and users

* To list data for a specific CoreXL instance, append "-i <instance number>" to your selection.

(Q)                Quit

*****************************************

2

SAs of all instances:

Peer [redacted].231 , vpngw_PA SAs:

        IKE SA <813742a4239b252e,f972606a2f9aa70d>
                INBOUND:
                        1. 0xac2676a3   (i: 1)
                OUTBOUND:
                        1. 0xd57d830a   (i: 1)
```

Figure 23. VPN tunnel up

As seen in Figure 24 interface eth2 was the customer interface assigned with IP address from the Check Point customer network 192.168.20.0/24. And from the routing table you can see there is no direct route towards the Palo Alto customer network 192.168.30.0/24. Ping using the customer interface eth2 towards the Palo Alto firewalls customer interface 192.168.30.1 traffic was able to reach the destination. But using the management interface to ping traffic is not able to reach the destination as only the network 192.168.20.0/24 network is defined in the local encryption domain on the firewall.

```
[Expert@CP_LAB_HOST_1:0]# ifconfig
Mgmt      Link encap:Ethernet  HWaddr 00:1C:7F:33:2F:68
          inet addr:          .83  Bcast:          .95  Mask:255.255.255.240
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:157604644 errors:0 dropped:0 overruns:0 frame:0
          TX packets:156836804 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13108289872 (12.2 GiB)  TX bytes:21004740205 (19.5 GiB)
          Interrupt:201 Memory:feae0000-feb00000

eth1      Link encap:Ethernet  HWaddr 00:1C:7F:33:2F:63
          inet addr:1.1.1.1  Bcast:1.1.1.3  Mask:255.255.255.252
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:499731961 errors:0 dropped:0 overruns:0 frame:0
          TX packets:500014308 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:42276536444 (39.3 GiB)  TX bytes:42080737071 (39.1 GiB)
          Interrupt:185 Memory:fe3e0000-fe400000

eth2      Link encap:Ethernet  HWaddr 00:1C:7F:33:2F:65
          inet addr:192.168.20.2  Bcast:192.168.20.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:138175211 errors:0 dropped:0 overruns:0 frame:0
          TX packets:138412084 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:11195715836 (10.4 GiB)  TX bytes:11197543498 (10.4 GiB)
          Interrupt:201 Memory:fe5e0000-fe600000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:6420664 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6420664 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1646352198 (1.5 GiB)  TX bytes:1646352198 (1.5 GiB)

[Expert@CP_LAB_HOST_1:0]# netstat -rnv
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
1.1.1.0         0.0.0.0         255.255.255.252 U         0 0          0 eth1
           .80  0.0.0.0         255.255.255.240 U         0 0          0 Mgmt
192.168.20.0    0.0.0.0         255.255.255.0   U         0 0          0 eth2
172.16.200.0    -               255.255.255.0   !D        - -          - -
0.0.0.0                    .81  0.0.0.0         UGD       0 0          0 Mgmt
[Expert@CP_LAB_HOST_1:0]# ping -I eth2 192.168.30.1
PING 192.168.30.1 (192.168.30.1) from 192.168.20.2 eth2: 56(84) bytes of data.
64 bytes from 192.168.30.1: icmp_seq=1 ttl=64 time=1.28 ms
64 bytes from 192.168.30.1: icmp_seq=2 ttl=64 time=0.870 ms

--- 192.168.30.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.870/1.079/1.288/0.209 ms
[Expert@CP_LAB_HOST_1:0]# ping 192.168.30.1
PING 192.168.30.1 (192.168.30.1) 56(84) bytes of data.

--- 192.168.30.1 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2000ms
```

Figure 24. Ping through VPN tunnel

5.5 Switch management network configuration

To access the switches remotely an Ubuntu machine was deployed on the VMware platform. The address for this machine was chosen from the same public address block as the addresses for the management device, as routing for this network had already been done which allowed the Ubuntu machine to be reached via SSH immediately after configuring the public address to it. Vlan 60 was chosen for the switch management network and private IP addresses from network 10.10.100.0/24 were used for the vlan interfaces. IP addresses were set for the vlan interfaces on the switches and cabling was added to have the switch management network entirely separated from other traffic, in case the other interfaces would get congested by malfunctioning traffic. This was not likely to happen in a laboratory environment but is crucial in a production environment, so the decision was made to apply the same design for the laboratory as the extra cost of adding a few cables was not high so there really were no downsides to it. On the Ubuntu jump machine the IP addresses of the switches were added to the hosts file to allow SSH connections to be established using the hostnames of the switches instead of the IP addresses. As seen in Figure 25 from the ubuntu machine SSH connection could be established to the switches in the laboratory using their hostnames which meant that the manageability for the switch network was aligned with the requirements detailed in the Network design theory section of the thesis.

```
Last login: Tue Mar 12 12:59:13 2019
hyppykone@ubuntu_jumpmachine:~$
hyppykone@ubuntu_jumpmachine:~$
hyppykone@ubuntu_jumpmachine:~$
hyppykone@ubuntu_jumpmachine:~$
hyppykone@ubuntu_jumpmachine:~$
hyppykone@ubuntu_jumpmachine:~$
hyppykone@ubuntu_jumpmachine:~$
hyppykone@ubuntu_jumpmachine:~$ ssh -l admin Lab_SW
Password:


Lab_SW>
Lab_SW>
Lab_SW>enable
Password:
Lab_SW#
Lab_SW#
Lab_SW#exit
Connection to lab_sw closed by remote host.
Connection to lab_sw closed.
hyppykone@ubuntu_jumpmachine:~$ ssh -l admin Lab_SW_CP
Password:


Lab_SW_CP>enable
Password:
Lab_SW_CP#
Lab_SW_CP#
```

Figure 25. SSH from ubuntu jump machine to switch

After this point the implementation of the laboratory was completed and cables for the existing topology were cleared out of the way using zip ties and cable hooks mounted on the sides of the racks. In Figure 26 one can see the Lab_SW_CP cisco 3650 switch and below that the Check Point 4200 single mode gateway appliance

with the Check Point 4400 and 4800 appliances which were used to create the fire-wall cluster.

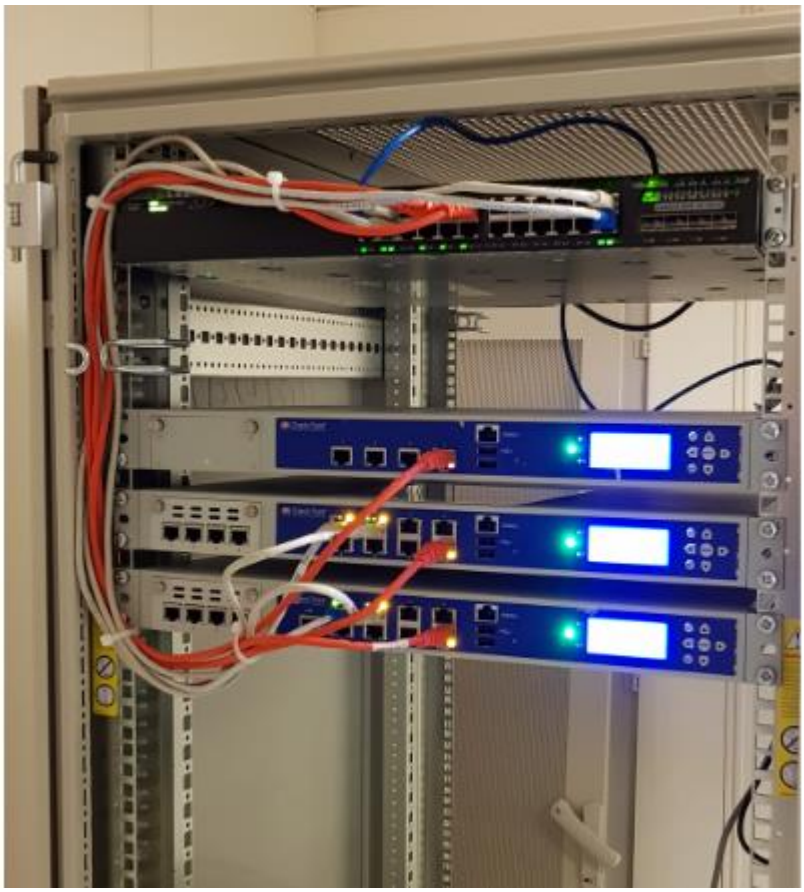


Figure 26. Final laboratory Check Point side

6 Conclusion

In totality the work done for the thesis included creating a simulated public network (internet) using only virtual local area networks, where private IP addresses were not routed. Configuring dedicated customer firewalls and LAN networks behind them and building remote management to all the devices included in the laboratory. The result of the thesis was a functioning laboratory environment that could be expanded to resemble an actual production environment or to test products from more manufacturers. Which meant that the research question proposed in the beginning was successfully completed. Using the constructive research method was the right choice as gathering knowledge about technologies to use and then implementing them through a process of trial and error was very successful in getting results. This however meant that once a working solution was found not much inquiry went into finding if the same result could have been reached by other means. Finding sources for the thesis was quite easy as most of the protocols and technologies used are extensively documented in official publications like RFC documents. To ensure that the new laboratory will not end up like the first one a procedure should be implemented dictating that all changes must be documented, and redundant configuration should be deleted in the process of making changes. As the environment is simply a testing platform no official orders to change configuration or documentation should be necessary but a simple topology picture and a text file to go along with it containing text based explanations about what has been done by who and when.

The initial design of the network which created the foundation for the entire design was provided by the firewall specialists at Telia. This meant that not much research went into what type of design would be most optimal and alternatives were not really investigated, so the design of the laboratory is more of a specific design to suit the needs of this particular organization but are not necessarily directly applicable to different environments. However, the design methods used to refine the original topology are valid and generalizable to any environment. If the laboratory was going to be used for more than testing firewall appliances and other network security devices,

then the initial design would have been different, and a more accurate version of a production environment would have been built. This could have included things like dedicated routers for both Check Point and Palo Alto environments and possibly even a small MPLS (Multiprotocol Label Switching) network, but that was outside of the scope of this thesis and most likely will not be the future of this environment, as teams who work with MPLS network have their own dedicated laboratory for testing. To improve the laboratory an SSL VPN gateway could be installed on the network which would then be used to connect end devices to the local area networks behind either the Palo Alto or Check Point firewalls. This would make the environment more realistic as most companies provide the possibility for out of the office work for their employees and having working SSL VPN connections through a firewall is crucial.

The entire process of creating the laboratory faced most of its difficulties from having to work around the obsolete devices and configuration. This was especially pointless as in the end the laboratory environment was built entirely from the ground up. If the existing laboratory was stripped down in its entirety from the beginning, building the new environment would have been easier and smoother. Working on the laboratory provided work on multitude of levels from installing devices and cabling to the racks to testing traffic through a VPN tunnel created on the devices. This gave good perspective on the process from beginning to end which is often not done by any single person in a large corporation like Telia. Usually physical installation, configuration of routers and switches, and configuration of the firewall are all done by different teams and organizations.

The laboratory has already seen use testing features of the new Check Point R80.10 platform. Migration of existing customer implementations to new appliances has also been done through the laboratory devices, as this allows for policies from Juniper devices to be completely converted to Check Point before the deployment of the new device. This is particularly useful as most of the time the conversion from one manufacturers device to another does not work perfectly and policies need to be gone over to make sure everything makes sense. Especially if moving from a zone-based policy to an interface-based policy.

References

About the company. 2017. Article posted on Telia Company website. Accessed on 1.12.2018. Retrieved from https://www.teliacompany.com/en/about-the-company/.

Crnkovic, G. 2010. Model-Based Reasoning in Science and Technology Pages 359-380 Constructive Research and Info-computational Knowledge Generation. Germany: Springer.

Doraswamy, N. & Harkins, D. 2002. IPSec: the new security standard for the internet, intranets and virtual private networks. New Jersey: Prentice Hall.

Edwardsson, A. N.d. Telia + Sonera = Teliasonera. Accessed on 1.12.2018. Retrieved from https://www.teliacompany.com/en/about-the-company/history/telia-sonera-and-teliasonera/.

Frankel, S. & Krishnan, S. 2011. IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap. RFC document published on IETF website. Accessed on 18.12.2018. Retrieved from https://tools.ietf.org/html/rfc6071.

Harkins, D. & Carrel, D. 1998. The Internet Key Exchange (IKE). RFC document published on IETF website. Accessed on 11.12.2018. Retrieved from https://tools.ietf.org/html/rfc2409.

Introducing Network Design Concepts. N.d. Cisco CCNA document published on Society of Cable Telecomminications Engineers website. Accessed on 19.3.2019. Retrieved from https://www.scte.org/documents/pdf/CCNA4%20Sample.pdf.

IPsec VPN Overview. 2018. Article published in Juniper TechLibrary. Accessed on 18.12.2018. Retrieved from https://www.juniper.net/documentation/en_US/junos/topics/topic-map/security-ipsec-vpn-overview.html.

Kent, S. 2005. IP Encapsulating Security Payload (ESP). RFC document published on IETF website. Accessed on 17.12.2018. Retrieved from https://tools.ietf.org/html/rfc4303.

Krawczyk, H. 1996. SKEME: A Versatile Secure Key Exchange Mechanism for Internet. Research paper published on IEEE website. Accessed on 15.12.2018. Retrieved from https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=492418.

Kryptografiset vahvuusvaatimukset luottamuksellisuuden suojaamiseen – kansalliset suojaustasot. 2018. A document published by the Finnish Communications Agency. Accessed on 18.1.2019. Retrieved from

https://legacy.viestintavirasto.fi/attachments/tietoturva/Kryptografiset_vahvuusvaatimukset_-_kansalliset_suojaustasot.pdf.

Leaf. 2015. Internet Protocol Security (IPSec). Blogpost on Anadril website. Accessed on 18.12.2018. Retrieved from https://www.anardil.net/2015/internet-protocol-security-ipsec.html.

Makati, M. 2013. Virtual Private Networks – Indepth technical details. Blogpost on rawbytes blog site. Accessed on 10.12.2018. Retrieved from http://www.rawbytes.com/virtual-private-networks-in-depth-technical-details.

Maughan, D., Schertler, M., Schneider, M. & Turner, J. 1998. Internet Security Association and Key Management Protocol (ISAKMP). RFC document published on IETF website. Accessed on 11.12.2018. Retrieved from https://tools.ietf.org/html/rfc2408.

Orman, H. 1998. The OAKLEY Key Determination Protocol. RFC document published on IETF website. Accessed on 12.12.2018. Retrieved from https://tools.ietf.org/html/rfc2412.

Security Management Architecture Overview. N.d. Document on Check Points website. Accessed on 05.12.2018. Retrieved from https://www.checkpoint.com/downloads/products/r80.10-security-management-architecture-overview.pdf.

VMware vSphere 6.5 Documentation Center. 2017. Document collection on VMware website. Accessed on 22.12.2018. Retrieved from https://pubs.vmware.com/vsphere-6-5/index.jsp.

Wen, S., Meng, Q., Fend, C. & Tang, C. 2017. A model-guided symbolic execution approach for network protocol implementations and vulnerability detection. Article published on ResearchGate website. Accessed on 12.12.2018. Retrieved from https://www.researchgate.net/publication/321119545_A_model-guided_symbolic_execution_approach_for_network_protocol_implementations_and_vulnerability_detection.

Appendices

Appendix 1. Check Point Cluster host 1 configuration

```
CP_LAB_HOST_1> show configuration
#
# Configuration of CP_LAB_HOST_1
# Language version: 13.1v1
#
# Exported by admin on Mon Dec 10 13:36:20 2018
#
set installer policy check-for-updates-period 3
set installer policy periodically-self-update on
set installer policy send-cpuse-data off
set installer policy self-test install-policy off
set installer policy self-test network-link-up off
set installer policy self-test start-processes on
set arp table cache-size 4096
set arp table validity-timeout 60
set arp announce 2
set message banner on

set message motd off

set message caption off
set core-dump enable
set core-dump total 1000
set core-dump per_process 2
set clienv debug 0
set clienv echo-cmd off
set clienv output pretty
set clienv prompt "%M"
set clienv rows 24
```

```
set clienv syntax-check off
set dns primary 8.8.8.8
set edition 64-bit
set expert-password-hash **********
set format date dd-mmm-yyyy
set format time 24-hour
set format netmask Dotted
set hostname CP_LAB_HOST_1
add allowed-client host any-host
set web table-refresh-rate 15
set web session-timeout 10
set web ssl-port 443
set web ssl3-enabled off
set web daemon-enable on
set inactivity-timeout 10
set ipv6-state off
add command api path /bin/api_wrap description "Start, stop, or check status of
API server"
add command tecli path /bin/tecli_start description "Threat Emulation Blade shel
l"
set lcd screensaver mode model
set lcd screensaver timeout 30
set net-access telnet off
set ntp active on
set ntp server primary 0.fi.pool.ntp.org version 4
set ntp server secondary 1.fi.pool.ntp.org version 4
set user admin shell /etc/cli.sh
set user admin password-hash **********
set user monitor shell /etc/cli.sh
set user monitor password-hash *
set password-controls min-password-length 6
set password-controls complexity 2
set password-controls palindrome-check true
```

```
set password-controls history-checking true
set password-controls history-length 10
set password-controls password-expiration never
set password-controls expiration-warning-days 7
set password-controls expiration-lockout-days never
set password-controls force-change-when no
set password-controls deny-on-nonuse enable false
set password-controls deny-on-nonuse allowed-days 365
set password-controls deny-on-fail enable false
set password-controls deny-on-fail failures-allowed 10
set password-controls deny-on-fail allow-after 1200
set aaa tacacs-servers state off
set aaa radius-servers super-user-uid 96
set max-path-splits 8
set tracefile maxnum 10
set tracefile size 1
set syslog filename /var/log/messages
set syslog cplogs off
set syslog mgmtauditlogs on
set syslog auditlog permanent
set timezone Europe / Helsinki
set interface Mgmt link-speed 1000M/full
set interface Mgmt state on
set interface Mgmt auto-negotiation on
set interface Mgmt ipv4-address ***.***.***.83 mask-length 28
set interface eth1 state on
set interface eth1 ipv4-address 1.1.1.1 mask-length 30
set interface eth1-01 state off
set interface eth1-02 state off
set interface eth1-03 state off
set interface eth1-04 state off
set interface eth2 state on
set interface eth2 ipv4-address 192.168.20.2 mask-length 24
```

```
set interface eth3 state off
set interface eth4 state off
set interface eth5 state off
set interface eth6 state off
set interface eth7 state off
set interface lo state on
set interface lo ipv4-address 127.0.0.1 mask-length 8
set inbound-route-filter ospf2 accept-all-ipv4
set inbound-route-filter rip accept-all-ipv4
set management interface Mgmt
set ospf area backbone on
set rip update-interval default
set rip expire-interval default
set snmp mode default
set snmp agent off
set snmp agent-version v3-Only
set snmp traps trap authorizationError disable
set snmp traps trap biosFailure disable
set snmp traps trap coldStart disable
set snmp traps trap configurationChange disable
set snmp traps trap configurationSave disable
set snmp traps trap fanFailure disable
set snmp traps trap highVoltage disable
set snmp traps trap linkUpLinkDown disable
set snmp traps trap lowDiskSpace disable
set snmp traps trap lowVoltage disable
set snmp traps trap overTemperature disable
set snmp traps trap powerSupplyFailure disable
set snmp traps trap raidVolumeState disable
set snmp traps trap vrrpv2AuthFailure disable
set snmp traps trap vrrpv2NewMaster disable
set snmp traps trap vrrpv3NewMaster disable
set snmp traps trap vrrpv3ProtoError disable
```

set static-route default nexthop gateway address ***.***.***.81 on

Appendix 2. Check Point Cluster host 2 configuration

```
CP_LAB_HOST_2> show configuration
#
# Configuration of CP_LAB_HOST_2
# Language version: 13.1v1
#
# Exported by admin on Mon Dec 10 13:38:19 2018
#
set installer policy check-for-updates-period 3
set installer policy periodically-self-update on
set installer policy send-cpuse-data off
set installer policy self-test install-policy off
set installer policy self-test network-link-up off
set installer policy self-test start-processes on
set arp table cache-size 4096
set arp table validity-timeout 60
set arp announce 2
set message banner on

set message motd off

set message caption off
set core-dump enable
set core-dump total 1000
set core-dump per_process 2
set clienv debug 0
set clienv echo-cmd off
set clienv output pretty
set clienv prompt "%M"
set clienv rows 43
```

```
set clienv syntax-check off
set dns primary 8.8.8.8
set edition 64-bit
set expert-password-hash *********
set format date dd-mmm-yyyy
set format time 24-hour
set format netmask Dotted
set hostname CP_LAB_HOST_2
add allowed-client host any-host
set web table-refresh-rate 15
set web session-timeout 10
set web ssl-port 443
set web ssl3-enabled off
set web daemon-enable on
set inactivity-timeout 10
set ipv6-state off
add command api path /bin/api_wrap description "Start, stop, or check status of API
server"
add command tecli path /bin/tecli_start description "Threat Emulation Blade shell"
set lcd screensaver mode model
set lcd screensaver timeout 30
set net-access telnet off
set ntp active on
set ntp server primary 0.fi.pool.ntp.org version 4
set ntp server secondary 1.fi.pool.ntp.org version 4
set user admin shell /etc/cli.sh
set user admin password-hash **********
set user monitor shell /etc/cli.sh
set user monitor password-hash *
set password-controls min-password-length 6
set password-controls complexity 2
set password-controls palindrome-check true
set password-controls history-checking true
```

```
set password-controls history-length 10
set password-controls password-expiration never
set password-controls expiration-warning-days 7
set password-controls expiration-lockout-days never
set password-controls force-change-when no
set password-controls deny-on-nonuse enable false
set password-controls deny-on-nonuse allowed-days 365
set password-controls deny-on-fail enable false
set password-controls deny-on-fail failures-allowed 10
set password-controls deny-on-fail allow-after 1200
set aaa tacacs-servers state off
set aaa radius-servers super-user-uid 96
set max-path-splits 8
set tracefile maxnum 10
set tracefile size 1
set syslog filename /var/log/messages
set syslog cplogs off
set syslog mgmtauditlogs on
set syslog auditlog permanent
set timezone Europe / Helsinki
set interface Mgmt link-speed 1000M/full
set interface Mgmt state on
set interface Mgmt auto-negotiation on
 set interface Mgmt ipv4-address ***.***.***.84 mask-length 28
set interface eth1 state on
set interface eth1 ipv4-address 1.1.1.2 mask-length 30
set interface eth1-01 state off
set interface eth1-02 state off
set interface eth1-03 state off
set interface eth1-04 state off
set interface eth2 state on
set interface eth2 ipv4-address 192.168.20.3 mask-length 24
set interface eth3 state off
```

```
set interface eth4 state off
set interface eth5 state off
set interface eth6 state off
set interface eth7 state off
set interface lo state on
set interface lo ipv4-address 127.0.0.1 mask-length 8
set inbound-route-filter ospf2 accept-all-ipv4
set inbound-route-filter rip accept-all-ipv4
set management interface Mgmt
set ospf area backbone on
set rip update-interval default
set rip expire-interval default
set snmp mode default
set snmp agent off
set snmp agent-version v3-Only
set snmp traps trap authorizationError disable
set snmp traps trap biosFailure disable
set snmp traps trap coldStart disable
set snmp traps trap configurationChange disable
set snmp traps trap configurationSave disable
set snmp traps trap fanFailure disable
set snmp traps trap highVoltage disable
set snmp traps trap linkUpLinkDown disable
set snmp traps trap lowDiskSpace disable
set snmp traps trap lowVoltage disable
set snmp traps trap overTemperature disable
set snmp traps trap powerSupplyFailure disable
set snmp traps trap raidVolumeState disable
set snmp traps trap vrrpv2AuthFailure disable
set snmp traps trap vrrpv2NewMaster disable
set snmp traps trap vrrpv3NewMaster disable
set snmp traps trap vrrpv3ProtoError disable
set static-route default nexthop gateway address ***.***.***.81 on
```

Appendix 3. Check Point single gateway configuration

```
CP_LAB_4200> show configuration
#
# Configuration of CP_LAB_4200
# Language version: 13.1v1
#
# Exported by admin on Mon Dec 10 16:21:33 2018
#
set installer policy check-for-updates-period 3
set installer policy periodically-self-update on
set installer policy send-cpuse-data off
set installer policy self-test install-policy off
set installer policy self-test network-link-up off
set installer policy self-test start-processes on
set arp table cache-size 4096
set arp table validity-timeout 60
set arp announce 2
set message banner on

set message motd off

set message caption off
set core-dump enable
set core-dump total 1000
set core-dump per_process 2
set clienv debug 0
set clienv echo-cmd off
set clienv output pretty
set clienv prompt "%M"
set clienv rows 42
```

```
set clienv syntax-check off
set edition 32-bit
set expert-password-hash *********
set format date dd-mmm-yyyy
set format time 24-hour
set format netmask Dotted
set hostname CP_LAB_4200
add allowed-client host any-host
set web table-refresh-rate 15
set web session-timeout 10
set web ssl-port 443
set web ssl3-enabled off
set web daemon-enable on
set inactivity-timeout 10
set ipv6-state off
add command api path /bin/api_wrap description "Start, stop, or check status of API
server"
add command tecli path /bin/tecli_start description "Threat Emulation Blade shell"
set lcd screensaver mode model
set lcd screensaver timeout 30
set net-access telnet off
set ntp active off
set user admin shell /etc/cli.sh
set user admin password-hash ***********
set user monitor shell /etc/cli.sh
set user monitor password-hash *
set password-controls min-password-length 6
set password-controls complexity 2
set password-controls palindrome-check true
set password-controls history-checking true
set password-controls history-length 10
set password-controls password-expiration never
set password-controls expiration-warning-days 7
```

```
set password-controls expiration-lockout-days never
set password-controls force-change-when no
set password-controls deny-on-nonuse enable false
set password-controls deny-on-nonuse allowed-days 365
set password-controls deny-on-fail enable false
set password-controls deny-on-fail failures-allowed 10
set password-controls deny-on-fail allow-after 1200
set aaa tacacs-servers state off
set aaa radius-servers super-user-uid 96
set max-path-splits 8
set tracefile maxnum 10
set tracefile size 1
set syslog filename /var/log/messages
set syslog cplogs off
set syslog mgmtauditlogs on
set syslog auditlog permanent
set timezone Europe / Helsinki
set interface Mgmt link-speed 100M/full
set interface Mgmt state on
set interface Mgmt auto-negotiation on
set interface Mgmt ipv4-address ***.***.***.86 mask-length 28
set interface eth1 state on
set interface eth1 ipv4-address 192.168.10.1 mask-length 24
set interface eth2 state off
set interface eth3 state off
set interface lo state on
set interface lo ipv4-address 127.0.0.1 mask-length 8
set inbound-route-filter ospf2 accept-all-ipv4
set inbound-route-filter rip accept-all-ipv4
set management interface Mgmt
set ospf area backbone on
set rip update-interval default
set rip expire-interval default
```

```
set snmp mode default
set snmp agent off
set snmp agent-version v3-Only
set snmp traps trap authorizationError disable
set snmp traps trap biosFailure disable
set snmp traps trap coldStart disable
set snmp traps trap configurationChange disable
set snmp traps trap configurationSave disable
set snmp traps trap fanFailure disable
set snmp traps trap highVoltage disable
set snmp traps trap linkUpLinkDown disable
set snmp traps trap lowDiskSpace disable
set snmp traps trap lowVoltage disable
set snmp traps trap overTemperature disable
set snmp traps trap powerSupplyFailure disable
set snmp traps trap raidVolumeState disable
set snmp traps trap vrrpv2AuthFailure disable
set snmp traps trap vrrpv2NewMaster disable
set snmp traps trap vrrpv3NewMaster disable
set snmp traps trap vrrpv3ProtoError disable
set static-route default nexthop gateway address ***.***.***.81 on
```

Appendix 4. Check Point Management configuration

```
CP_LAB_MGMT> show configuration
#
# Configuration of CP_LAB_MGMT
# Language version: 13.1v1
#
# Exported by admin on Mon Dec 10 14:36:23 2018
#
set installer policy check-for-updates-period 3
```

```
set installer policy periodically-self-update on
set installer policy send-cpuse-data off
set installer policy self-test install-policy off
set installer policy self-test network-link-up off
set installer policy self-test start-processes on
set arp table cache-size 4096
set arp table validity-timeout 60
set arp announce 2
set message banner on

set message motd off

set message caption off
set core-dump enable
set core-dump total 1000
set core-dump per_process 2
set clienv debug 0
set clienv echo-cmd off
set clienv output pretty
set clienv prompt "%M"
set clienv rows 24
set clienv syntax-check off
set edition 64-bit
set expert-password-hash *********
set format date dd-mmm-yyyy
set format time 24-hour
set format netmask Dotted
set hostname CP_LAB_MGMT
add allowed-client host any-host
set web table-refresh-rate 15
set web session-timeout 10
set web ssl-port 443
set web ssl3-enabled off
```

```
set web daemon-enable on
set inactivity-timeout 10
set ipv6-state off
add command api path /bin/api_wrap description "Start, stop, or check status of API server"
add command tecli path /bin/tecli_start description "Threat Emulation Blade shell"
add command vsec_central_license path /bin/vsec_central_license description "virtual edition licenses distribution utility"
set net-access telnet off
set ntp active off
set user admin shell /etc/cli.sh
set user admin password-hash **********
set user monitor shell /etc/cli.sh
set user monitor password-hash *
set password-controls min-password-length 6
set password-controls complexity 2
set password-controls palindrome-check true
set password-controls history-checking true
set password-controls history-length 10
set password-controls password-expiration never
set password-controls expiration-warning-days 7
set password-controls expiration-lockout-days never
set password-controls force-change-when no
set password-controls deny-on-nonuse enable false
set password-controls deny-on-nonuse allowed-days 365
set password-controls deny-on-fail enable false
set password-controls deny-on-fail failures-allowed 10
set password-controls deny-on-fail allow-after 1200
set aaa tacacs-servers state off
set aaa radius-servers super-user-uid 96
set max-path-splits 8
set tracefile maxnum 10
set tracefile size 1
```

```
set syslog filename /var/log/messages
set syslog cplogs off
set syslog mgmtauditlogs on
set syslog auditlog permanent
set timezone Europe / Helsinki
set interface eth0 link-speed 1000M/full
set interface eth0 state on
set interface eth0 auto-negotiation on
set interface eth0 ipv4-address ***.***.***.42 mask-length 27
set interface lo state on
set interface lo ipv4-address 127.0.0.1 mask-length 8
set inbound-route-filter ospf2 accept-all-ipv4
set inbound-route-filter rip accept-all-ipv4
set management interface eth0
set ospf area backbone on
set rip update-interval default
set rip expire-interval default
set snmp mode default
set snmp agent off
set snmp agent-version v3-Only
set snmp traps trap authorizationError disable
set snmp traps trap biosFailure disable
set snmp traps trap coldStart disable
set snmp traps trap configurationChange disable
set snmp traps trap configurationSave disable
set snmp traps trap fanFailure disable
set snmp traps trap highVoltage disable
set snmp traps trap linkUpLinkDown disable
set snmp traps trap lowDiskSpace disable
set snmp traps trap lowVoltage disable
set snmp traps trap overTemperature disable
set snmp traps trap powerSupplyFailure disable
set snmp traps trap raidVolumeState disable
```

```
set snmp traps trap vrrpv2AuthFailure disable
set snmp traps trap vrrpv2NewMaster disable
set snmp traps trap vrrpv3NewMaster disable
set snmp traps trap vrrpv3ProtoError disable
set static-route default nexthop gateway address ***.***.***.33 on
```

Appendix 5. Lab_SW config

```
Lab_SW#show running-config
Building configuration...

Current configuration : 10775 bytes
!
! Last configuration change at 11:16:17 UTC Mon Aug 20 2018 by admin
!
version 16.3
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no platform punt-keepalive disable-kernel-core
!
hostname Lab_SW
!
!
vrf definition Mgmt-vrf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
```

```
enable secret 5 **********
enable password *******
!
no aaa new-model
switch 1 provision ws-c3650-24ts
!
!
!
!
!
!
!
ip domain name jkllab.zz
!
!
!
!
!
!
!
!
!
crypto pki trustpoint TP-self-signed-1544775874
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-1544775874
 revocation-check none
 rsakeypair TP-self-signed-1544775874
!
!
crypto pki certificate chain TP-self-signed-1544775874
 certificate self-signed 01
  30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101
05050030
```

31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
69666963 6174652D 31353434 37373538 3734301E 170D3138 30373131 31313033
30385A17 0D323030 31303130 30303030 305A3031 312F302D 06035504 03132649
4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D31 35343437
37353837 34308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201
0A028201 01008BA5 B3E150F4 8F7759EF 85839914 C4DD0BA7 32ADA7EC A0D8DF2D
5213A702 59B84615 0A589984 78A8F3B5 C94E7F58 5B8CA929 8008269F 560129D1
8DBA66D9 3AFAA43A 9C279C20 BF6DF3B6 E46DA6E9 04DC3B11 ACE17A50 05B9AE04
9D6D0589 EB315D5C 70A36AC8 D1D07CAE 90EAAEBF B1829223 1039DAFD B802ADF9
43E438AB 2529EB1C 1FD00A5D AA9E6CCE 387683AF 015D151B BB6BB5F4 7AB3A4FC
02557E03 3341019E B8B82603 13790717 BC5C5047 DB5B37AA 14E6A090 C3628B96
DF038FDA 105A61B9 C108C46A ABE99C1B B2A885CD E8BE8636 18B39DF8 9D2BFD36
4B64DA98 438CB2FA 428E69BB 46EA1811 AADD588B A73B1FA3 F1C776E8 5296EF7A
1FEC48C5 94610203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF
301F0603 551D2304 18301680 14828C9C 6F8A1D31 EC15F859 A7196872 2224F552
6E301D06 03551D0E 04160414 828C9C6F 8A1D31EC 15F859A7 19687222 24F5526E
300D0609 2A864886 F70D0101 05050003 82010100 558C10D4 50ADB926 F59E4A93
67FEA4C9 9A86B190 03B28558 329C4A6D 20E089CA 0BCFDE22 581E2FA5 052A07EC

```
  9C2D46E9 BB9AEB81 96A4749B FAE4545F 8C8D2FEC 8FA9DC77 B4B37F08
66865293
  8E2A3587 18FED8EE BFBDE52B 0F73D9E0 538B169B D883504A 34B3249B
8236DC31
  60A3B586 554DEABA 6F13C98E 8DE8876C D724A30F EE9C8809 E27CD43B
D8F14C9C
  BFE038B9 868284D4 3B57F3F6 6BE1174C 2C0F45C6 E332DECC D9119A10
AE0C1494
  B20C39A5 A4D5B219 0014261E CC9F48F9 41482A19 10139EA9 BD2A0D9E
578272E2
  1837011D C189ED56 6FDD31E9 50CC5D09 210DC840 6F7854C5 18C697F2
220A9F56
  B84BD778 1496B482 A6D4F53B 42CA0B11 094E31C6
       quit
!
license boot level ipbasek9
diagnostic bootup level minimal
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
!
username administrator privilege 15 secret 5 *********
username admin secret 5 *********
!
redundancy
 mode sso
!
!
vlan 2048,3010
!
vlan 10
 name CP_mgmt_vlan
!
```

```
vlan 20
 name CP-Customer-vlan
!
vlan 60
 name SW_mgmt
!
vlan 30
 name PA-Customer-vlan
!
!
class-map match-any system-cpp-police-topology-control
  description Topology control
class-map match-any system-cpp-police-sw-forward
  description Sw forwarding, SGT Cache Full, LOGGING
class-map match-any system-cpp-default
  description DHCP snooping, show forward and rest of traffic
class-map match-any system-cpp-police-sys-data
  description Learning cache ovfl, Crypto Control, Exception, EGR Exception, NFL
SAMPLED DATA, Gold Pkt, RPF Failed
class-map match-any system-cpp-police-punt-webauth
  description Punt Webauth
class-map match-any system-cpp-police-forus
  description Forus Address resolution and Forus traffic
class-map match-any system-cpp-police-multicast-end-station
  description MCAST END STATION
class-map match-any system-cpp-police-multicast
  description Transit Traffic and MCAST Data
class-map match-any system-cpp-police-l2-control
  description L2 control
class-map match-any system-cpp-police-dot1x-auth
  description DOT1X Auth
class-map match-any system-cpp-police-data
  description ICMP_GEN and BROADCAST
```

```
class-map match-any system-cpp-police-control-low-priority
  description ICMP redirect and general punt
class-map match-any system-cpp-police-wireless-priority1
  description Wireless priority 1
class-map match-any system-cpp-police-wireless-priority2
  description Wireless priority 2
class-map match-any system-cpp-police-wireless-priority3-4-5
  description Wireless priority 3,4 and 5
class-map match-any non-client-nrt-class
class-map match-any system-cpp-police-routing-control
  description Routing control
class-map match-any system-cpp-police-protocol-snooping
  description Protocol snooping
!
policy-map port_child_policy
 class non-client-nrt-class
  bandwidth remaining ratio 10
policy-map system-cpp-policy
 class system-cpp-police-data
  police rate 200 pps
 class system-cpp-police-sys-data
  police rate 100 pps
 class system-cpp-police-sw-forward
  police rate 1000 pps
 class system-cpp-police-multicast
  police rate 500 pps
 class system-cpp-police-multicast-end-station
  police rate 2000 pps
 class system-cpp-police-punt-webauth
 class system-cpp-police-l2-control
 class system-cpp-police-routing-control
  police rate 1800 pps
 class system-cpp-police-control-low-priority
```

```
 class system-cpp-police-wireless-priority1
 class system-cpp-police-wireless-priority2
 class system-cpp-police-wireless-priority3-4-5
 class system-cpp-police-topology-control
 class system-cpp-police-dot1x-auth
 class system-cpp-police-protocol-snooping
 class system-cpp-police-forus
 class system-cpp-default
!
!
!
!
!
!
!
!
!
!
!
!
!
!
interface GigabitEthernet0/0
 vrf forwarding Mgmt-vrf
 no ip address
 negotiation auto
!
interface GigabitEthernet1/0/1
 description To PA.228
 switchport trunk allowed vlan 1,60,101,200,1001-1005
 switchport mode trunk
!
interface GigabitEthernet1/0/2
```

```
 description Trunk to Lab_SW_CP interface eth1
 switchport trunk allowed vlan 10,20,60
 switchport mode trunk
!
interface GigabitEthernet1/0/3
 description To PA.228 interface 1 from internet SRX210
!
interface GigabitEthernet1/0/4
 description To PA.228 interface 11 from CP Firewalls
 switchport access vlan 10
 switchport mode access
!
interface GigabitEthernet1/0/5
 description To SRX210(internet) from PA.228
!
interface GigabitEthernet1/0/6
 description IN-liittyma  Fa1
 switchport access vlan 1001
 switchport mode access
!
interface GigabitEthernet1/0/7
 shutdown
!
interface GigabitEthernet1/0/8
 shutdown
!
interface GigabitEthernet1/0/9
 shutdown
!
interface GigabitEthernet1/0/10
 description vmware host 10.142.253.93 port 4 vswitch3
 switchport mode trunk
!
```

```
interface GigabitEthernet1/0/11
 shutdown
!
interface GigabitEthernet1/0/12
 description vwmare 10.142.253.92 portti 1
 switchport trunk allowed vlan 1,101
 switchport mode trunk
!
interface GigabitEthernet1/0/13
 description vwmare 10.142.253.93 portti 1
 switchport trunk allowed vlan 1,101
 switchport mode trunk
!
interface GigabitEthernet1/0/14
 description vcenter 10.142.253.91
 switchport access vlan 101
!
interface GigabitEthernet1/0/15
 description vmware 10.142.253.92 iSCS  1.1.1.2
 switchport access vlan 400
 switchport mode access
!
interface GigabitEthernet1/0/16
 description vmware node 10.142.253.92 vSwitch2
 switchport trunk allowed vlan 60
 switchport mode trunk
!
interface GigabitEthernet1/0/17
 shutdown
!
interface GigabitEthernet1/0/18
 shutdown
!
```

```
interface GigabitEthernet1/0/19
 description vmware 10.142.253.92 port4
 switchport trunk allowed vlan 50,60,101,1001
 switchport mode trunk
!
interface GigabitEthernet1/0/20
 shutdown
!
interface GigabitEthernet1/0/21
 description vmware node 10.142.253.93 port2
 switchport access vlan 400
 switchport mode access
!
interface GigabitEthernet1/0/22
 description vmware node 10.142.253.93 port3
 switchport trunk allowed vlan 60,20,30
 switchport mode trunk
!
interface GigabitEthernet1/0/23
 description SW_mgmt
 switchport trunk allowed vlan 60
 switchport mode trunk
!
interface GigabitEthernet1/0/24
 description PA-Customer-Network jumpcable behind PA.231
 switchport access vlan 3630
 switchport mode access
!
interface GigabitEthernet1/1/1
!
interface GigabitEthernet1/1/2
!
interface GigabitEthernet1/1/3
```

```
!
interface GigabitEthernet1/1/4
!
interface Vlan1
 no ip address
 shutdown
!
interface Vlan333
 no ip address
!
interface Vlan1001
 no ip address
!
interface Vlan60
 ip address 10.10.100.20 255.255.255.0
!
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
!
ip access-list extended AutoQos-4.0-wlan-Acl-Bulk-Data
 permit tcp any any eq 22
 permit tcp any any eq 465
 permit tcp any any eq 143
 permit tcp any any eq 993
 permit tcp any any eq 995
 permit tcp any any eq 1914
 permit tcp any any eq ftp
 permit tcp any any eq ftp-data
 permit tcp any any eq smtp
 permit tcp any any eq pop3
ip access-list extended AutoQos-4.0-wlan-Acl-MultiEnhanced-Conf
```

```
 permit udp any any range 16384 32767
 permit tcp any any range 50000 59999
ip access-list extended AutoQos-4.0-wlan-Acl-Scavanger
 permit tcp any any range 2300 2400
 permit udp any any range 2300 2400
 permit tcp any any range 6881 6999
 permit tcp any any range 28800 29100
 permit tcp any any eq 1214
 permit udp any any eq 1214
 permit tcp any any eq 3689
 permit udp any any eq 3689
 permit tcp any any eq 11999
ip access-list extended AutoQos-4.0-wlan-Acl-Signaling
 permit tcp any any range 2000 2002
 permit tcp any any range 5060 5061
 permit udp any any range 5060 5061
ip access-list extended AutoQos-4.0-wlan-Acl-Transactional-Data
 permit tcp any any eq 443
 permit tcp any any eq 1521
 permit udp any any eq 1521
 permit tcp any any eq 1526
 permit udp any any eq 1526
 permit tcp any any eq 1575
 permit udp any any eq 1575
 permit tcp any any eq 1630
 permit udp any any eq 1630
 permit tcp any any eq 1527
 permit tcp any any eq 6200
 permit tcp any any eq 3389
 permit tcp any any eq 5985
 permit tcp any any eq 8080
!
ip sla enable reaction-alerts
```

```
!
!
control-plane
 service-policy input system-cpp-policy
!
!
no vstack
!
line con 0
 logging synchronous
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 password *******
 login local
 transport input ssh
line vty 5 15
 password *******
 login
!
!
wsma agent exec
!
wsma agent config
!
wsma agent filesys
!
wsma agent notify
!
!
ap dot11 airtime-fairness policy-name Default 0
ap group default-group
```

```
ap hyperlocation ble-beacon 0
ap hyperlocation ble-beacon 1
ap hyperlocation ble-beacon 2
ap hyperlocation ble-beacon 3
ap hyperlocation ble-beacon 4
end
```

Appendix 6. Lab_SW_CP config

```
Lab_SW_CP#show running-config
Building configuration...

Current configuration : 9440 bytes
!
! Last configuration change at 06:40:25 UTC Tue Aug 21 2018 by admin
!
version 16.3
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no platform punt-keepalive disable-kernel-core
!
hostname Lab_SW_CP
!
!
vrf definition Mgmt-vrf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
```

```
!
enable secret 5 *********
!
no aaa new-model
switch 1 provision ws-c3650-24ts
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
crypto pki trustpoint TP-self-signed-2231370201
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-2231370201
 revocation-check none
 rsakeypair TP-self-signed-2231370201
!
!
crypto pki certificate chain TP-self-signed-2231370201
 certificate self-signed 01
  30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101
05050030
```

31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
69666963 6174652D 32323331 33373032 3031301E 170D3138 30363235 31303038
30395A17 0D323030 31303130 30303030 305A3031 312F302D 06035504 03132649
4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 32333133
37303230 31308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201
0A028201 0100A233 F53596CD E3DC5C24 F2A0041B EA2F2E20 B7722B35 13E7C15C
2FE7301D 44EF2C23 A3B24BE8 A8467F02 C8F80DED 3E2D0443 DE5A2827 A791638F
A85933B4 9C354679 7BFE776D DEB8A3FA 586CF412 65D052F1 E4D7BC94 96DFEE0E
B6EF5973 205485DC F677AD94 97F3FD79 728D158A 478A9515 3D22DDEA F8EC6138
B74C319E E6FDA8A4 BF3BD1ED 08B62696 31AAEBF4 BE1675E1 8128FDDF C4445C3C
B745E49D F0287A7A C3D7BCE1 2F637547 D4669DED 642F3826 5EC5F4A8 5602B59E
1FC5AAED 4886C9C2 C6A6D441 2913A2CD C08B3ECF 76390F1E 17AEFAD2 E3DCC229
F1075336 C1035801 AB41921F 01AACB4E 21EF8EE4 AADB0E6A 504FE977 AE273985
CD2DF11B 27D30203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF
301F0603 551D2304 18301680 14AB7B95 B7F5BA65 85AB3641 716F2813 7B74A819
2B301D06 03551D0E 04160414 AB7B95B7 F5BA6585 AB364171 6F28137B 74A8192B
300D0609 2A864886 F70D0101 05050003 82010100 8972F028 A4AF9537 69FF806E
6F59E4E7 DB7F9BA7 0A592F39 7D9AF911 B907A274 C1C3E0C2 7C87696A B6D5BBF3

```
  38EF68DB FC1EB6D2 BE0BB069 0FB945BC 69F1D7CE 1ED7C79D 94FB6357
1B894A3D
  03F6CD9E 67CCFD98 C0C41CD8 2C7576A0 BDE90996 842A642D 21CE28C1
A6C76EDC
  C056D25A 4DB07376 D5FCBB73 D85E2D01 7168EF2C 5844C564 966AD0AF
716C3343
  BCF926D8 A8273642 971CCED3 90D45C15 8590CF83 42CA6530 FC91B15A
9AA8FA02
  939C3034 B8FE3975 403554A2 48A89381 A632C387 D2DFB675 6BE06133
C0E5D578
  B2031190 C886E96D 58D6C828 B6D084D9 1D653438 931D0EAC 67B7A8F4
371F4E63
  FDC07EFB 25BA2E65 AE253CF9 7919A9FD 54BC2F83
      quit
!
license boot level ipbasek9
diagnostic bootup level minimal
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
!
username admin secret 5 ********
!
redundancy
 mode sso
!
!
vlan 10
 name CP_mgmt
!
vlan 20
 name CP-Customer-vlan
!
```

```
vlan 30
 name CP_Sync
!
vlan 60
 name SW_mgmt
!
!
class-map match-any system-cpp-police-topology-control
  description Topology control
class-map match-any system-cpp-police-sw-forward
  description Sw forwarding, SGT Cache Full, LOGGING
class-map match-any system-cpp-default
  description DHCP snooping, show forward and rest of traffic
class-map match-any system-cpp-police-sys-data
  description Learning cache ovfl, Crypto Control, Exception, EGR Exception, NFL
SAMPLED DATA, Gold Pkt, RPF Failed
class-map match-any system-cpp-police-punt-webauth
  description Punt Webauth
class-map match-any system-cpp-police-forus
  description Forus Address resolution and Forus traffic
class-map match-any system-cpp-police-multicast-end-station
  description MCAST END STATION
class-map match-any system-cpp-police-multicast
  description Transit Traffic and MCAST Data
class-map match-any system-cpp-police-l2-control
  description L2 control
class-map match-any system-cpp-police-dot1x-auth
  description DOT1X Auth
class-map match-any system-cpp-police-data
  description ICMP_GEN and BROADCAST
class-map match-any system-cpp-police-control-low-priority
  description ICMP redirect and general punt
class-map match-any system-cpp-police-wireless-priority1
```

```
  description Wireless priority 1
class-map match-any system-cpp-police-wireless-priority2
  description Wireless priority 2
class-map match-any system-cpp-police-wireless-priority3-4-5
  description Wireless priority 3,4 and 5
class-map match-any non-client-nrt-class
class-map match-any system-cpp-police-routing-control
  description Routing control
class-map match-any system-cpp-police-protocol-snooping
  description Protocol snooping
!
policy-map port_child_policy
 class non-client-nrt-class
  bandwidth remaining ratio 10
policy-map system-cpp-policy
 class system-cpp-police-data
  police rate 200 pps
 class system-cpp-police-sys-data
  police rate 100 pps
 class system-cpp-police-sw-forward
  police rate 1000 pps
 class system-cpp-police-multicast
  police rate 500 pps
 class system-cpp-police-multicast-end-station
  police rate 2000 pps
 class system-cpp-police-punt-webauth
 class system-cpp-police-l2-control
 class system-cpp-police-routing-control
  police rate 1800 pps
 class system-cpp-police-control-low-priority
 class system-cpp-police-wireless-priority1
 class system-cpp-police-wireless-priority2
 class system-cpp-police-wireless-priority3-4-5
```

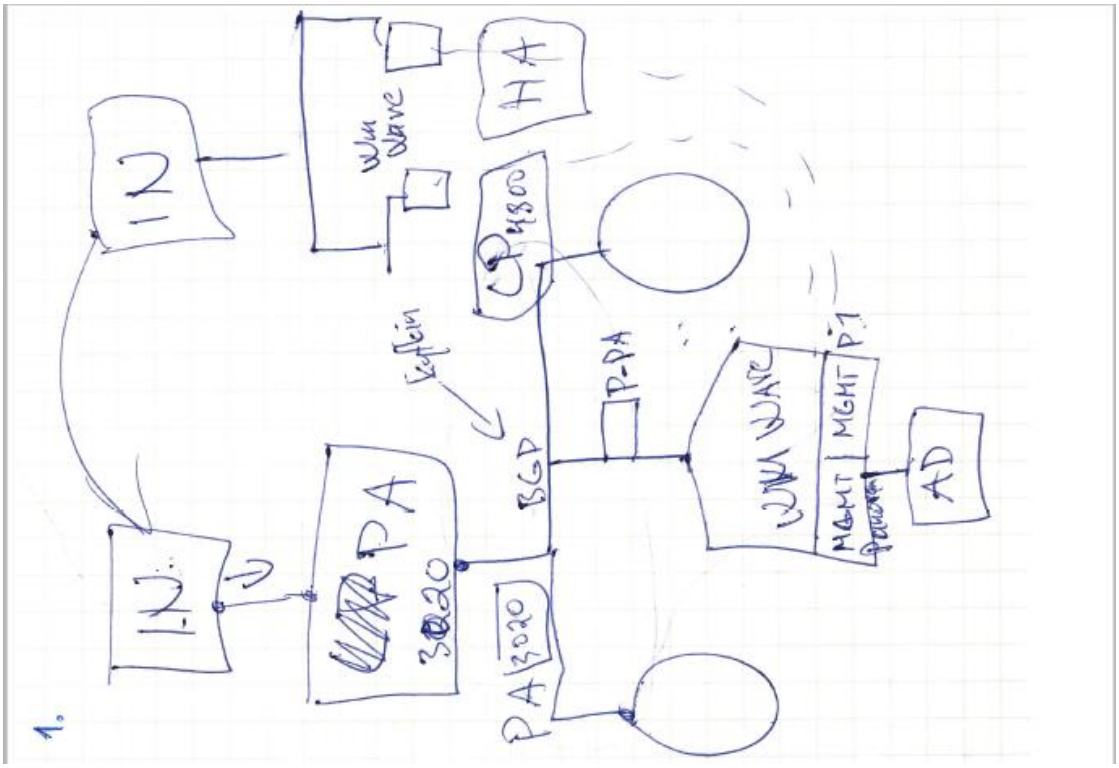
```
 class system-cpp-police-topology-control
 class system-cpp-police-dot1x-auth
 class system-cpp-police-protocol-snooping
 class system-cpp-police-forus
 class system-cpp-default
!
!
!
!
!
!
!
!
!
!
!
!
!
!
interface GigabitEthernet0/0
 vrf forwarding Mgmt-vrf
 no ip address
 negotiation auto
!
interface GigabitEthernet1/0/1
 description Trunk to Lab_SW interface eth1/0/2
 switchport trunk allowed vlan 10,20,60
 switchport mode trunk
!
interface GigabitEthernet1/0/2
!
interface GigabitEthernet1/0/3
 description CP-Customer-Network
```

```
 switchport access vlan 20
 switchport mode access
!
interface GigabitEthernet1/0/4
 description CP-Customer-Network
 switchport access vlan 20
 switchport mode access
!
interface GigabitEthernet1/0/5
!
interface GigabitEthernet1/0/6
!
interface GigabitEthernet1/0/7
 description CP mgmt 4200
 switchport access vlan 10
 switchport mode access
!
interface GigabitEthernet1/0/8
!
interface GigabitEthernet1/0/9
 description CP mgmt 4800
 switchport access vlan 10
 switchport mode access
!
interface GigabitEthernet1/0/10
!
interface GigabitEthernet1/0/11
 description CP mgmt 4400
 switchport access vlan 10
 switchport mode access
!
interface GigabitEthernet1/0/12
!
```

```
interface GigabitEthernet1/0/13
!
interface GigabitEthernet1/0/14
!
interface GigabitEthernet1/0/15
!
interface GigabitEthernet1/0/16
!
interface GigabitEthernet1/0/17
!
interface GigabitEthernet1/0/18
!
interface GigabitEthernet1/0/19
!
interface GigabitEthernet1/0/20
!
interface GigabitEthernet1/0/21
!
interface GigabitEthernet1/0/22
!
interface GigabitEthernet1/0/23
 switchport access vlan 30
 switchport mode access
!
interface GigabitEthernet1/0/24
 switchport access vlan 30
 switchport mode access
!
interface GigabitEthernet1/1/1
!
interface GigabitEthernet1/1/2
!
interface GigabitEthernet1/1/3
```

```
!
interface GigabitEthernet1/1/4
!
interface Vlan1
 no ip address
 shutdown
!
interface Vlan60
 ip address 10.10.100.50 255.255.255.0
!
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
!
ip access-list extended AutoQos-4.0-wlan-Acl-Bulk-Data
 permit tcp any any eq 22
 permit tcp any any eq 465
 permit tcp any any eq 143
 permit tcp any any eq 993
 permit tcp any any eq 995
 permit tcp any any eq 1914
 permit tcp any any eq ftp
 permit tcp any any eq ftp-data
 permit tcp any any eq smtp
 permit tcp any any eq pop3
ip access-list extended AutoQos-4.0-wlan-Acl-MultiEnhanced-Conf
 permit udp any any range 16384 32767
 permit tcp any any range 50000 59999
ip access-list extended AutoQos-4.0-wlan-Acl-Scavanger
 permit tcp any any range 2300 2400
 permit udp any any range 2300 2400
 permit tcp any any range 6881 6999
```

```
 permit tcp any any range 28800 29100
 permit tcp any any eq 1214
 permit udp any any eq 1214
 permit tcp any any eq 3689
 permit udp any any eq 3689
 permit tcp any any eq 11999
ip access-list extended AutoQos-4.0-wlan-Acl-Signaling
 permit tcp any any range 2000 2002
 permit tcp any any range 5060 5061
 permit udp any any range 5060 5061
ip access-list extended AutoQos-4.0-wlan-Acl-Transactional-Data
 permit tcp any any eq 443
 permit tcp any any eq 1521
 permit udp any any eq 1521
 permit tcp any any eq 1526
 permit udp any any eq 1526
 permit tcp any any eq 1575
 permit udp any any eq 1575
 permit tcp any any eq 1630
 permit udp any any eq 1630
 permit tcp any any eq 1527
 permit tcp any any eq 6200
 permit tcp any any eq 3389
 permit tcp any any eq 5985
 permit tcp any any eq 8080
!
!
!
control-plane
 service-policy input system-cpp-policy
!
!
no vstack
```

```
!
line con 0
 logging synchronous
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login local
 transport input ssh
line vty 5 15
 login
 transport input ssh
!
!
wsma agent exec
!
wsma agent config
!
wsma agent filesys
!
wsma agent notify
!
!
ap dot11 airtime-fairness policy-name Default 0
ap group default-group
ap hyperlocation ble-beacon 0
ap hyperlocation ble-beacon 1
ap hyperlocation ble-beacon 2
ap hyperlocation ble-beacon 3
ap hyperlocation ble-beacon 4
end
```

Appendix 7. First draft of the firewall laboratory