Tommi Lundell

# Multi-vendor VPN Troubleshooting

## Study of IPsec VPN Troubleshooting in a multi-vendor environment

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Thesis

9 May 2017

| | |
|---|---|
| Author<br>Title<br><br>Number of Pages<br>Date | Tommi Lundell<br>Multi-vendor VPN Troubleshooting<br><br>57 pages + 2 appendices<br>9.5.2017 |
| Degree | Bachelor of Engineering |
| Degree Programme | Information Technology |
| Specialisation option | Networking |
| Instructor | Marko Uusitalo, Senior Lecturer |

With the vast reliance on the Internet, corporations end up managing not only their intranet but extranets as well to allow resource sharing with their partner corporations. In order to connect users or sites to these intra- and extranets over various internets, VPN tunnels are typically constructed to create such a logical connection over physical infrastructure. In such environments different networking devices are found with varying degrees of support for vendor interoperability. This will set constraints and considerations to the VPN tunnel creation as well in multivendor environments.

This study was fueled by the desire to explore differences in VPN tunnel building and troubleshooting processes between two devices designed by different vendors. As such this project aims to construct an IPsec site-to-site VPN tunnel between two such gateways and systematically introduce issues to the connection for troubleshooting analysis. The results then would be compiled into a guide to aid future troubleshooting. Additionally, any differences and similarities in the devices' protocol handling will be documented.

The available hardware for the project was Cisco's ASA 5505 and Palo Alto Networks' PA-200 security gateways. The physical network setup was done in the campus' laboratory environment the third-party network included.

The experimentation results were fruitful, providing two points of view to the protocol exchanges. While the standardized protocols guided the negotiations along a certain path, the debugging output certainly differed between the vendors; Palo Alto focused more on printing out each stage transitioning, while Cisco was more focused on user-friendliness.

In the future when site-to-site IPsec VPNs are implemented, corporations could be seen gradually switching to IKEv2 protocol as the latest stable software versions have confirmed the support. As the authentication method is more of a question of scalability, both PSK and Certificate- based methods will certainly remain in use. Other than the upgrade of security levels, future studies will more often implement support for IPv6 tunnelling parallel to IPv4.

| | |
|---|---|
| Keywords | VPN, IPsec, IKEv1, site-to-site, troubleshooting |

| | |
|---|---|
| Tekijä<br>Otsikko<br><br>Sivumäärä<br>Päivämäärä | Tommi Lundell<br>VPN:n vianselvitys<br><br>57 sivua + 2 liitettä<br>9.5.2017 |
| Tutkinto | Insinööri (AMK) |
| Koulutusohjelma | Tietotekniikka |
| Suuntautumisvaihtoehto | Tietoverkot |
| Ohjaaja | Lehtori Marko Uusitalo |

Internetistä on kasvanut niin suuri tekijä yrityksille, että intranetin lisäksi ekstranetien hallinnointi tulee olla huomioitu resurssienjakoa varten. Etäkäyttäjien ja toimipisteiden yhteenliittämiseksi julkisien verkkojen kautta, VPN-tunneleita yleisesti käytetään näiden loogisien yhteyksien muodostamisessa. Tällaisissä ympäristöissä tulee vastaan monien eri laitevalmistajien laitteita, jotka vaihtelevissa määrin ovat yhteensopivia. Eri laitevalmistajien laitteiden tukeminen vaatii tiettyjen ominaisuuksien huomioimista VPN-tunneleidenkin kohdalla.

Insinöörityön taustalla oli kiinnostus tutkia, kuinka VPN-tunnelien rakentaminen ja vianselvitys eroavat eri laitevalmistajien laitteiden välillä. Työssä rakennettiin kuviteltujen toimipisteiden välille VPN-tunneli, jota systemaattisesti koeteltiin eri ennaltamääritellyillä ongelmatilanteilla. Nämä ongelmatilanteet puolestaan dokumentoitiin analyysia varten, ja niiden pohjalta tehtiin kooste olennaisista viesteistä, jotka osoittivat vikojen alkuperät. Lisäksi laitteiden protokollien soveltamista tarkkailtiin ja vertailtiin.

Työssä käytettiin Ciscon ASA 5505- ja Palo Alto Networksin PA-200-palomuureja. Käytännön työ suoritettiin ammattikorkeakoulun kampuksen laboratorioympäristössä kokonaisuudessaan.

Testitulokset olivat värikkäitä ja ilmaisivat keskustelun kahdesta selvästi eri näkökulmasta, vaikka protokollien standardit ohjasivatkin protokollaneuvottelujen etenemistä. Palo Alton PA-200 kuvaili tarkemmin keskustelun vaiheet ja taustaoperaatiot, kun taas Ciscon ASA 5505 keskittyi käyttäjäystävällisyyteen ja luettavuuteen.

Tulevaisuudessa tällaisissa implementaatioissa siirrytään varmasti IKEv2-protokollan käyttöön sitä mukaa, kun käyttöjärjestelmät sitä laajemmin tukevat. Sertifikaattiautentikoinnin käyttö lisääntyy ekstranetien lisääntyessä, mutta PSK-autentikointimenetelmä pysynee pienempien yrityksen käytössä. IPv6-protokollan käyttöönotto ja tuki lisääntyy myös jatkuvasti – VPN-tunneleiden osalla erityisesti.

| | |
|---|---|
| Avainsanat | VPN, IPsec, IKEv1, vianselvitys |

## Contents

Appendices

## Abbreviations

| | |
|---|---|
| 3DES | Triple Digital Encryption Standard |
| ACC | Application Command Center |
| ACL | Access Control List |
| AES | Advanced Encryption Standard |
| AH | Authentication Header |
| API | Application Programming Interface |
| AM | Aggressive Mode |
| ASA | Adaptive Security Appliance |
| ASDM | Adaptive Security Device Manager |
| CA | Certificate Authority |
| CLI | Command Line Interface |
| DH | Diffie-Hellman |
| DHCP | Dynamic Host Configuration Protocol |
| DES | Digital Encryption Standard |
| DPD | Dead Peer Detection |
| ESP | Encapsulating Security Payload |
| GUI | Graphical User Interface |
| HMAC | Hashed Message Authentication Code |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hypertext Transfer Protocol Secure |
| ICV | Integrity Check Value |
| IETF | Internet Engineering Task Force |
| IKE | Internet Key Exchange |
| IOS | Internetwork Operating System |
| IPv4 | Internet Protocol version 4 |
| IPS | Intrusion Prevention System |
| IPsec | Internet Protocol Security |
| ISAKMP | Internet Security Association and Key Management Protocol |
| ISP | Internet Service Provider |
| LAN | Local Area Network |
| MD5 | Message Digest 5 |
| MM | Main Mode |
| NAT | Network Address Translation |
| NAT-T | Network Address Translation-Traversal |
| NIC | Network Interface Card |

| | |
|---|---|
| OS | Operating System |
| PAT | Port Address Translation |
| PBF | Policy-Based Forwarding |
| PFS | Perfect Forward Secrecy |
| PSK | Pre-Shared Key |
| QM | Quick Mode |
| QoS | Quality of Service |
| RFC | Request For Comments |
| RSA | Rivest, Shamir, Adelman |
| SA | Security Association |
| SHA | Secure Hash Algorithm |
| SPI | Security Parameter Index |
| SSH | Secure Shell |
| SSL | Secure Sockets Layer |
| TCP | Transmission Control Protocol |
| UDP | User Datagram Protocol |
| URL | Uniform Resource Locator |
| VLAN | Virtual Local Area Network |
| VPN | Virtual Private Network |
| Xauth | Extended Authentication |
| XML | Extensive Markup Language |

# 1 Introduction

This paper aims to serve as a guide for troubleshooting Virtual Private Network (VPN) issues between two networking devices: Cisco's ASA 5505 and Palo Alto Networks' PA-200. In this study a Site-to-Site VPN connection is set up using the Internet Protocol Security (IPsec) protocol suite.

Starting with the theory section the concept of VPN and the IPsec protocol suite is presented and the structure of the necessary modifications to Internet Protocol (IP) packets covered. Next the devices and vendors are briefly introduced following step-by-step device configurations. Once the VPN tunnels have been established, problem situations are introduced to the VPN connectivity via systematic misconfigurations. These issues are documented and identified from both devices' perspectives via various methods introduced by their respective vendors. After the issues have been identified, a summary is drawn in conclusion to report the findings.

The motivation for the study comes from having worked in a multi-vendor environment where access to only one tunnel endpoint is available. In order to save time and be able to draw conclusions with proof to show in such cases, I chose this topic.

While this study covers Site-to-Site IPsec VPN connectivity troubleshooting methods, it is limited to the devices Cisco's ASA 5505 and Palo Alto Networks' PA-200. Changes in software versions and peering with any other network devices are likely to introduce issues not covered by this troubleshooting guide.

# 2 Virtual Private Networks (VPNs)

The Internet, being a composite of various devices and mediums in-between, is typically treated as an obscure, virtual cloud. The ownership of the Internet is divided between numerous Internet Service Providers (ISPs) and other organizations each responsible for their own infrastructure as part of the Internet. End-users utilizing their ISP's services trust that they receive connectivity to all resources connected to the Internet. However, in case of confidential data a layer of security should be added to the data travelling across the Internet. Here we introduce the concept of VPN.

The purpose of a VPN, as its name implies, is the creation of a virtual private network connection also called a VPN tunnel. The VPN tunnel establishes a logical connection between two endpoints over a third-party network, such as the Internet, for security and management purposes. As a security measure data can be delivered over third-party networks encrypted and then authenticated and integrity checked at the end of the tunnel. For management purposes the whole third-party infrastructure can be viewed as a flexible pipe spanning from one end to another without caring what there is physically in-between.

## 2.1 Site-to-Site VPN

VPN types are commonly divided into Site-to-Site and Remote Access VPNs. A Site-to-Site VPN is established between two gateways connecting for instance a branch site's network to the headquarters' network. A Remote Access VPN on the other hand connects a host to a remote gateway for instance a telecommuter to corporate site. A less commonly known third type, host-to-host, also exists, though it can be classified as a restricted Remote Access VPN [1, 245–250]. This study uses Site-to-Site VPN to connect two sites together, allowing their private networks to communicate securely.

## 2.2 IPsec VPN

IPsec is what commonly provides VPNs with the security they are known for – at least in Site-to-Site implementations. IPsec is a protocol suite operating on the Network layer, thus it complements VPN well when transmitting data between trusted networks over an untrusted network. [2, 256–257].

## 3 Internet Protocol Security (IPsec)

Rather than being an actual tunnelling protocol, IPsec, defined by Internet Engineering Task Force's (IETF) RFC6071 document among others, is a modular collection of protocols to provide VPN tunnelling with data confidentiality, message integrity, data origin authentication and anti-replay protection [3, 3]. IPsec's ability to support various open standards also allows for vendor interoperability [2, 256].

The IPsec framework comprises two main protocols: Authentication Header (AH) and Encapsulating Security Payload (ESP). These two are responsible for providing the security services of the IPsec framework. Figure 1 below breaks the framework down and presents its components.

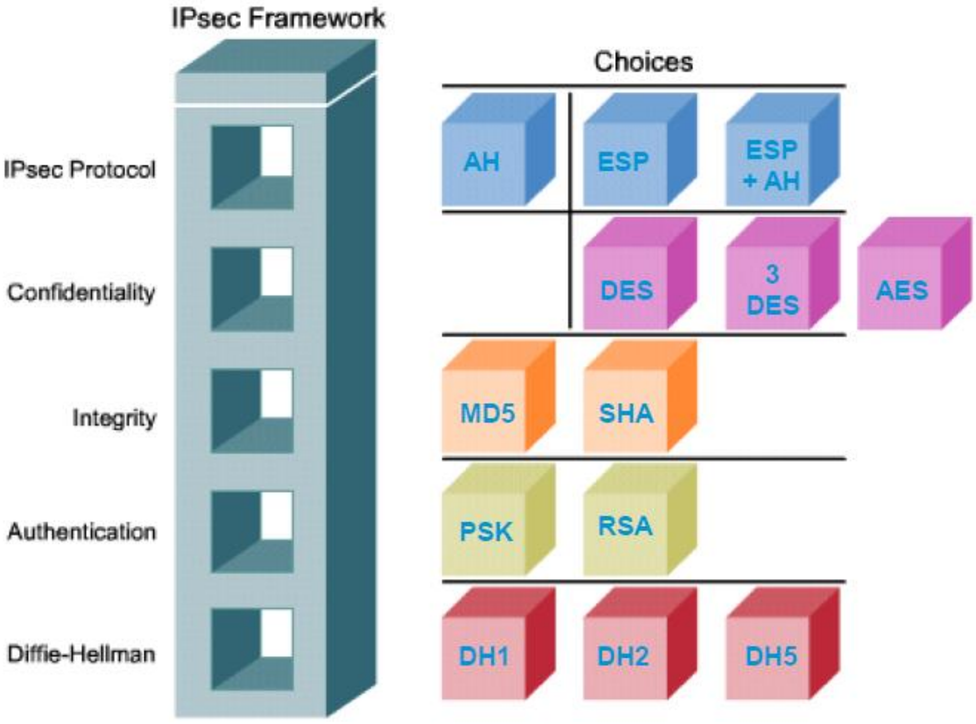


Figure 1. IPsec Framework overview (modified) [4].

In IPsec VPNs IP packets can be encapsulated either using Authentication Header (AH) or Encapsulating Security Payload (ESP). The primary difference between the two is that AH does not encrypt the IP packets and thus does not provide data confidentiality [2, 261]. Both AH and ESP can be used together, but such practical examples are non-existent. Nowadays ESP is the standard choice.

The confidentiality of  data is guaranteed via cryptography; the contents of IP packets are encrypted before entering the tunnel and then decrypted at the end of the tunnel. In the encrypted form the otherwise plaintext data is temporarily transformed into unreadable format, so the packet's contents could not be deciphered even if captured in transmit over a third-party network. For the process of encryption and decryption a symmetric encryption algorithm is chosen according to the level of security required. The strength of standardized encryption algorithms can be generally estimated by the length of the keys (in bits) used in the algorithm's computations; the longer the more secure against brute-force attacks. The most common encryption algorithms as pictured in Figure 1 are as follows:

- Digital Encryption Standard (DES)
- Triple Digital Encryption Standard (3DES)
- Advanced Encryption Standard (AES). [2, 256–267.]

The data integrity ensures that data does not get lost or modified in transmit. This is done by taking a packet's content and running it through a cryptographic hash function. This will result in a theoretically unique, irreversible and compressed bit stream known as a hash. The hash generated at tunnel's transmitting end is concatenated to the packet to be used at the receiving end in verifying the packet's integrity; if the hash included in the packet matches with the hash generated at the receiving end the contents can be verified to be unharmed. Cryptographic hash functions as shown in Figure 1 include:

- Message Digest 5 (MD5)
- Secure Hash Algorithm (SHA). [2, 256–259.]

Additionally utilizing Hashed Message Authentication Code (HMAC) with the chosen hash function, the message authenticity  can be verified along with its integrity as a shared secret is used in producing the hash [5, 1–2]. Implemented together with a cryptographic hash function such as SHA-1 results in a variant called HMAC-SHA-1 [6].

VPN peer authentication ensures that the VPN tunnel is established between the right devices and is thus an essential step in forming VPN connectivity. The primary methods are usage of Pre-Shared Key (PSK), RSA signatures and RSA-encrypted nonces. PSKs are manually configured and shared between the engineers responsible for the VPN peers over a different communication channel. RSA signatures are digital certificates signed by a trusted, third-party Certificate Authority (CA). RSA-encrypted nonces utilize public key cryptography making it more complex than PSK. RSA-encrypted nonces as an authentication method is also limited to the Cisco Internetwork Operating System (IOS) software. While RSA signatures possess superior scalability over PSKs due to saving the trouble of configuring it separately for each individual peer, this study will use the more easily deployed PSK as the peer authentication method.

Finally the Diffie-Hellman (DH) group selection will determine the security level of the secure key exchange. DH, being a public-key cryptography protocol, allows the secure generation of shared secret keys for encryption algorithms over third-party networks. The higher the DH group number the more secure it is against brute-force attacks.

## 3.1 IPsec Modes

IPsec can operate either in Transport mode or Tunnel mode. The Transport mode protects packets up to the Transport layer and the Tunnel mode protects up to the Network layer.

In the Transport mode IPsec places the header (AH, ESP or AH+ESP) between the original IP header and the IP payload (data). This will leave the original IP header exposed to traffic analysis when transmitted over public networks. [7, 457–459.]

In the Tunnel mode the IPsec header (AH, ESP or AH+ESP) encapsulates both the original IP header and IP payload. Additionally a new IP header is created on top of the IPsec header. This new IP header specifies the VPN peers as the source and destination addresses, thus concealing the actual message sender and recipient. While the Tunnel mode increases the packet overhead, it provides additional security. [7, 457–459.]

## 3.2 IPsec Security Protocols

In practice IPsec security services are applied to traffic via AH and/or ESP header encapsulation. The AH and ESP headers contain the information necessary to validate the IPsec Security Association (SA) and the data within. These headers are not applied to IKE/IPsec negotiation exchanges and only for the data to be secured once the IKE and IPsec SAs have been established.

AH is an IPsec security protocol with the following features:

- data integrity
- data origin authentication
- anti-replay protection.

Figure 2 below displays the fields within an AH header.

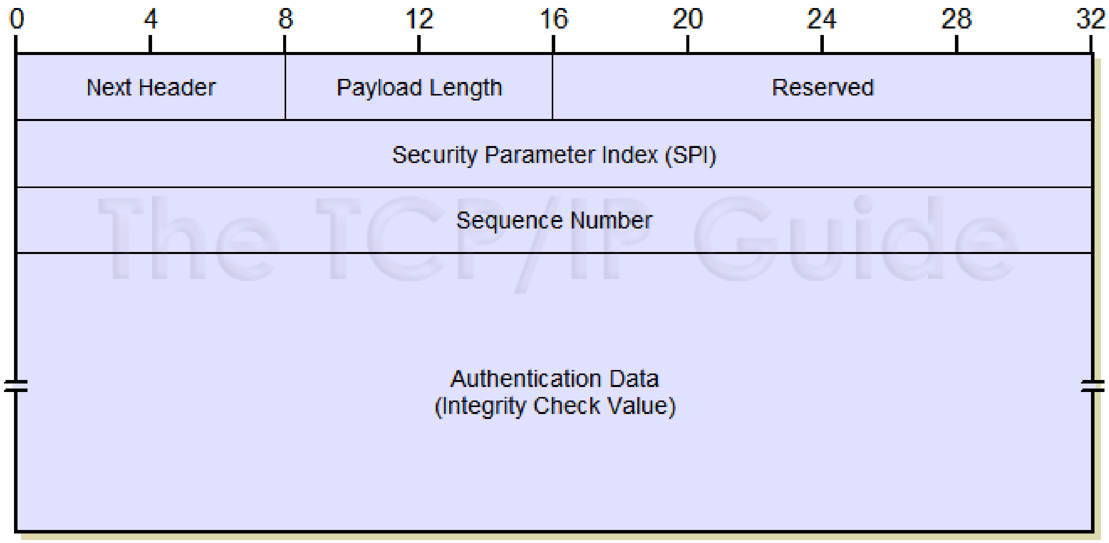


Figure 2. IPsec AH format [7, 465].

The Next Header field contains the protocol number of the next header such as 4 for IPv4 header. The Payload length specifies the length of the AH without the following payload. The Reserved functions as padding and is set to zero. The Security Parameter Index (SPI), combined with destination address and protocol type, acts as an identifier for IPsec SAs – receiver should have a matching inbound SA. Sequence Number is a running number used to protect against replay attacks by preventing retransmission. Integrity Check Value (ICV) is a signed hash which the recipient uses to verify data integrity and to authenticate the sender. [7, 465.]

ESP as the other IPsec security protocol provides the following features:

- data confidentiality
- data integrity
- data origin authentication
- anti-replay protection.

Figure 3 below displays the fields within an ESP header.

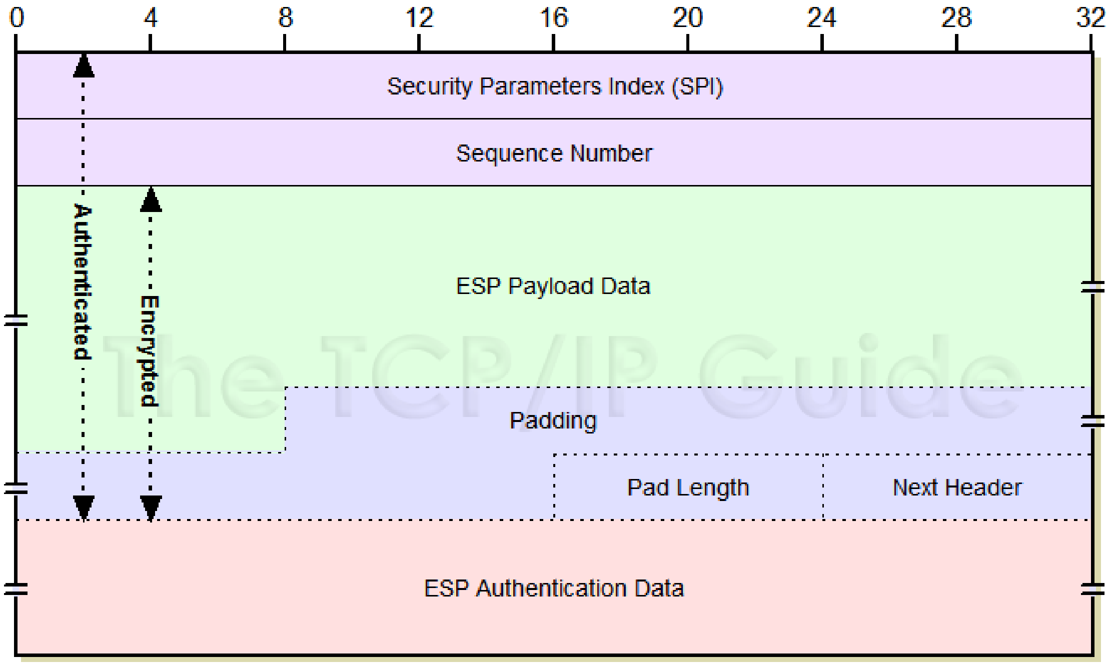


Figure 3. IPsec ESP format [7, 471; 8, 4].

Compared to AH and it’s distinct header, ESP is scattered around the payload due to the encryption scheme. ESP header comprises SPI and Sequence Number fields acting as IPsec SA identifier and anti-replay protection just as with AH header. Following ESP header is ESP Payload, the encrypted data. Following the ESP Payload is ESP Trailer which contains Padding, Pad Length and Next Header. ESP Trailer is encrypted along with ESP Payload and provides necessary padding to adjust the input for the encryption algorithm. Next Header field in IPsec Tunnel mode has the number 4 for IP Header inside the ESP Payload. Finally, like AH, ESP contains an ICV, ESP Authentication Data, to verify data integrity and authenticate the sender. [7, 466–470.]

Figure 4 portraits the AH and ESP encapsulations for both Transport and Tunnel mode.

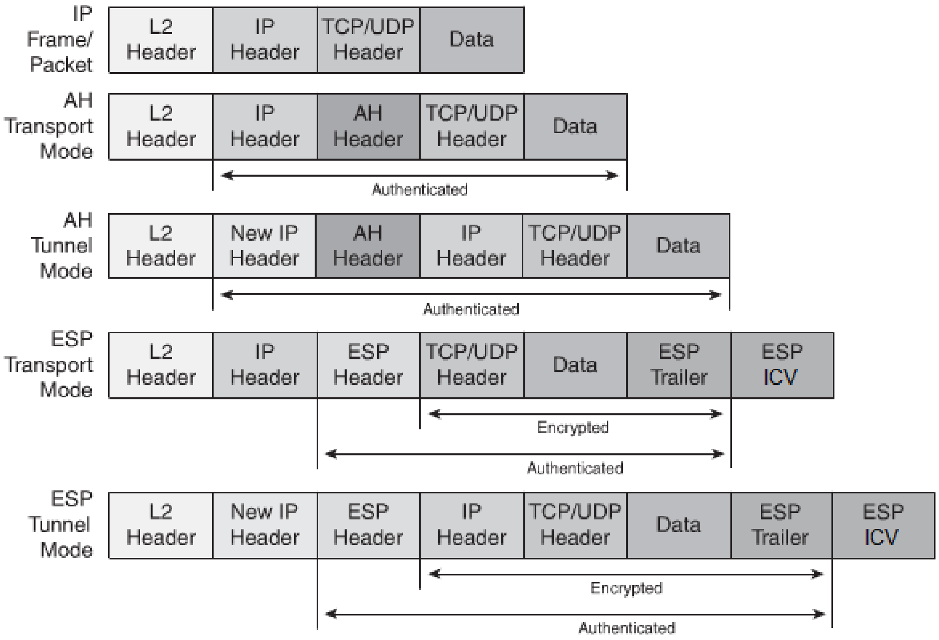


Figure 4. IPsec encapsulation in Transport and Tunnel mode (modified) [2, 261; 8, 17–19.]

As shown in Figure 4 Transport mode leaves the original IP header exposed, whereas Tunnel mode encapsulates it inside AH or ESP header and creates a new header for transmission until tunnel endpoint. Figure 4 also highlights which headers are included in ESP's encryption and authentication (ICV). It is noteworthy to point out that AH additionally includes the IP header preceding the AH header to its ICV computation. As a result AH requires that the topmost IP header remains immutable during transmit. [8, 17–19]. This renders Network Address Translation (NAT) as well as IPsec's optional feature Network Address Translation-Traversal (NAT-T) impossible to combine with AH. Also, the ESP header does not specify Transmission Control Protocol (TCP) / User Datagram Protocol (UDP) header information. This is only inside the inner, encapsulated IP header. In case NAT-T is enabled and discovered during IKE Phase 1 MM messages 3–4, a new UDP header will precede the ESP header with UDP port 4500 as the source and destination port.

## 3.3 Internet Key Exchange (IKE)

Internet Key Exchange (IKE) is the protocol responsible for the policy negotiations between peers and establishment of Security Associations (SA) in the IPsec framework. IKE is in fact a hybrid protocol utilizing the functions of three protocols:

- Internet Security Association and Key Management Protocol (ISAKMP)

- Oakley
- SKEME. [9, 2–3.]

ISAKMP provides IKE with the framework for SA negotiation, establishment, updating and termination. Oakley is a key determination protocol and manages the key exchanges using a DH algorithm. Oakley can also provide Perfect Forward Secrecy (PFS) for keys and peer identities. Similar to Oakley, SKEME defines public key encryption methods for authenticated key exchange and swift re-keying via nonce exchange. [9, 2–3.]

IKE currently has two different implementations: IKE version 1 (IKEv1) and IKE version 2 (IKEv2). IKEv2, being the successor for IKEv1, runs most of the same features as IKEv1 does with improved stability, more efficient exchanges and various other improvements. [10, 135–136]. However, as not nearly all networking devices yet fully support IKEv2, IKEv1 is still widely implemented. This study will only cover configuration steps for IKEv1.

IKE negotiations are divided into two phases: Phase 1 and Phase 2. IKE Phase 1 negotiates IKE policies, performs DH key exchange and peer authentication in order to establish a bidirectional IKE SA – also known as ISAKMP SA. IKE Phase 2, utilizing the existing IKE SA, negotiates IPsec security parameters for tunnelled traffic and establishes a pair of unidirectional IPsec SAs. [2, 284–288.] An optional IKE Phase 1.5 also exists to provide user authentication via Extended Authentication (Xauth) [2, 263–266].

IKE Phase 1 can operate in two modes: Main Mode (MM) and Aggressive Mode (AM). Both MM and AM perform the same tasks; MM does this in 6 messages while AM in 3 messages. MM stages are as follows:

1. The Initiator sends the receiver its IKE policy sets as proposals
2. Receiver compares the policy sets to its own and sends the agreed policy set back
3. Initiator sends its DH public key and nonce to initiate DH key exchange
4. Receiver sends its DH public key and nonce completing DH key exchange
   - With the exchanged keys, they both compute a shared secret
5. Initiator sends its identity payload encrypted and hashed

6. Receiver decrypts and validates the hashed identity payload in order to authenticate the peer then sends its identity payload encrypted and hashed in return.

In comparison AM stages are as follows:

1. Initiator sends its IKE policy set proposal, DH key materials and identity payload
2. Receiver replies with the selected IKE policy set, its own DH key materials and identity payload with authentication hash for Initiator to authenticate.
3. Initiator sends its encrypted authentication hash for receiver to authenticate.

While AM employs more efficiency in its exchanges, it does not provide protection for identity payload and receiver has to agree on initiator's DH group selection. [9, 7–15.]

IKE Phase 2 utilizes the established IKE SA applying the negotiated security parameters to protect its exchanges via encryption and authentication. IKE Phase 2 operation mode is called Quick Mode (QM). QM exchanges the following three messages:

1. Initiator sends the IPsec policy set proposal, a nonce and a hash payload
   - If PFS is enabled, new DH key materials are exchanged for IPsec SAs
2. Receiver sends the agreed IPsec policy set, its own nonce and a hash payload
   - If PFS is enabled, new DH key materials are exchanged for IPsec SAs
3. Initiator sends a hash payload generated with the nonces completing the exchange [9, 17.]

## 3.4 Security Association (SA)

SA represents an established secure connection between two parties. When an SA is active, the agreed security parameters are applied to the exchanges. IKE SA, as established during IKE Phase 1, acts as a bidirectional control channel for management of IPsec SAs. IPsec SA, as established during IKE Phase 2, is a unidirectional data channel, thus two IPsec SAs are required for inbound and outbound trafficking between two tunnelled networks. Should VPN peers negotiate multiple networks to be allowed pass through the VPN, an IPsec SA pair will be generated for each pair. [7, 460–461.]

SA information is stored in a Security Association Database (SAD). SAD records Destination IP address, SPI number and IPsec security protocol number for each SA. All three pieces of information are present in IPsec-encapsulated packets to identify which SA the packet belongs to. Among the negotiated security parameters is SA lifetime, which is tracked by each peer to determine when the SA is to be terminated and new one(s) established. [2, 291–292.]

# 4 Device Setup

For the study two firewalls capable of acting as gateways for a small to mid-size site were chosen. Among the feature requirements were support for IPsec Site-to-Site VPN and IPv6 deployment. The devices chosen were Palo Alto Networks' PA-200 and Cisco's Adaptive Security Appliance (ASA) 5505. The two are the most affordable models among their series of firewalls.

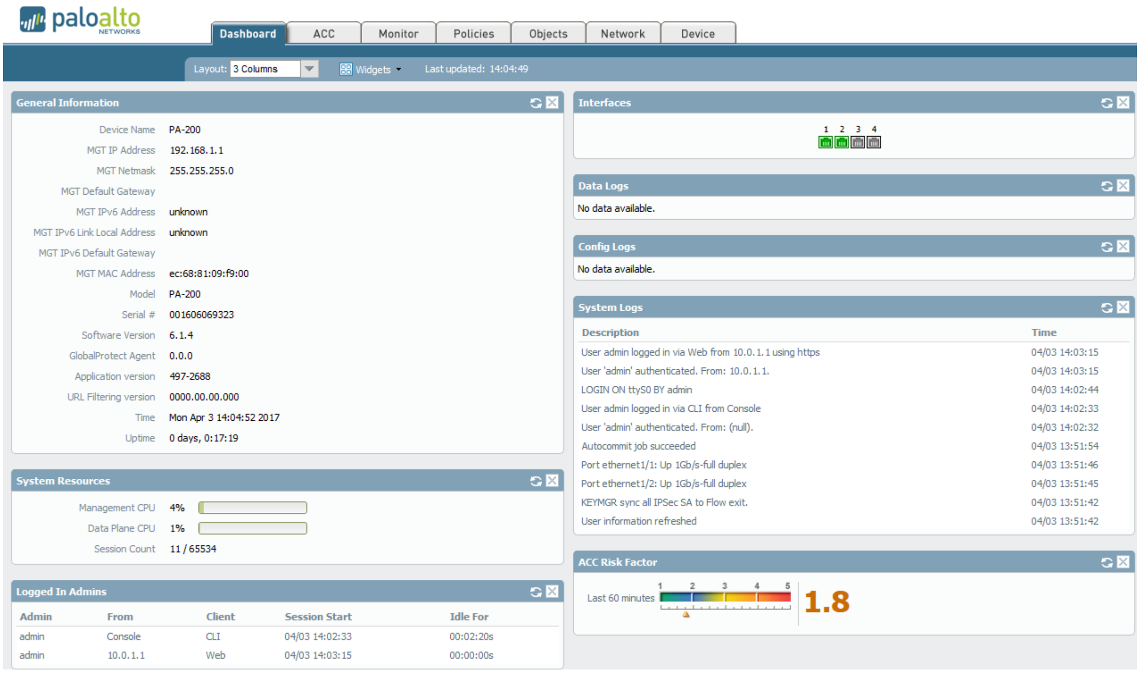
## 4.1 Palo Alto Networks PA-200

Palo Alto Networks' PA-200 is a next-generation security appliance running Palo Alto's PAN-OS operating system. The device comes with a 4-port Ethernet switch supporting Layer 3, Layer 2 and transparent Virtual-wire interface modes as well as separate management, console and USB ports. PA-200 is capable of applying security policies not only based on the IP address, zone, port or protocol but also based on User-ID, App-ID and Content-ID. In addition, the firewall provides Intrusion Prevention System (IPS) engine for traffic scanning, Secure Sockets Layer (SSL) and IPsec VPN tunnelling and virtualization of system or routing instances for scalability. [11.]

PA-200 of Palo Alto Networks has the following management interfaces:

- Web-based Graphical User Interface (GUI) over Hypertext Transfer Protocol (HTTP) or HTTP Secure (HTTPS) via an Ethernet port
- Command Line Interface (CLI) either over Telnet or Secure Shell (SSH) via an Ethernet port or via the console port
- Extensive Markup Language (XML) based Application Programming Interface (API) over HTTPS via an Ethernet port. [11].

For initial configuration either the console port or the pre-configured Management port (MGT) should be used. This study will handle device configurations via GUI and use both GUI and CLI for troubleshooting. The Web-based GUI is illustrated in Figure 5 below.



General Information

| | |
|---|---|
| Device Name | PA-200 |
| MGT IP Address | 192.168.1.1 |
| MGT Netmask | 255.255.255.0 |
| MGT Default Gateway | |
| MGT IPv6 Address | unknown |
| MGT IPv6 Link Local Address | unknown |
| MGT IPv6 Default Gateway | |
| MGT MAC Address | ec:68:81:09:f9:00 |
| Model | PA-200 |
| Serial # | 001606069323 |
| Software Version | 6.1.4 |
| GlobalProtect Agent | 0.0.0 |
| Application version | 497-2688 |
| URL Filtering version | 0000.00.00.000 |
| Time | Mon Apr 3 14:04:52 2017 |
| Uptime | 0 days, 0:17:19 |

System Resources

| | |
|---|---|
| Management CPU | 4% |
| Data Plane CPU | 1% |
| Session Count | 11 / 65534 |

Logged In Admins

| Admin | From | Client | Session Start | Idle For |
|---|---|---|---|---|
| admin | Console | CLI | 04/03 14:02:33 | 00:02:20s |
| admin | 10.0.1.1 | Web | 04/03 14:03:15 | 00:00:00s |

Interfaces

Data Logs

No data available.

Config Logs

No data available.

System Logs

| Description | Time |
|---|---|
| User admin logged in via Web from 10.0.1.1 using https | 04/03 14:03:15 |
| User 'admin' authenticated. From: 10.0.1.1. | 04/03 14:03:15 |
| LOGIN ON ttyS0 BY admin | 04/03 14:02:44 |
| User admin logged in via CLI from Console | 04/03 14:02:33 |
| User 'admin' authenticated. From: (null). | 04/03 14:02:32 |
| Autocommit job succeeded | 04/03 13:51:54 |
| Port ethernet1/1: Up 1Gb/s-full duplex | 04/03 13:51:46 |
| Port ethernet1/2: Up 1Gb/s-full duplex | 04/03 13:51:45 |
| KEYMGR sync all IPSec SA to Flow exit. | 04/03 13:51:42 |
| User information refreshed | 04/03 13:51:42 |

ACC Risk Factor

Last 60 minutes 1.8

Figure 5. PA-200 Web GUI Dashboard

The PA-200 Web GUI contains the following tabs:

- Dashboard
- Application Command Center (ACC)
- Monitor
- Policies
- Objects
- Network
- Device.

The Dashboard contains selectable widgets to display the overall operating status and recent events of the device. The ACC constructs illustrious application usage, Uniform Resource Locator (URL) filtering, data filtering and threat prevention statistics based on traffic logs. The Monitor allows the viewing and filtering of traffic, threat, URL, data, configuration, system and alarm logs. Monitor also allows packet capture and reporting. Policies section enforces the firewall’s security policies, NAT rules, Policy-Based Forwarding (PBF) and Quality of Service (QoS) policies. Objects tab contains all definitions pertaining to entities or entity groups given custom identifiers to be referenced in places such as security policies. Network contains all configurations regarding interfaces, virtual routers, routing protocols, zones, Virtual Local Area Networks (VLANs), IPsec tunnels, Dynamic Host Configuration Protocol (DHCP) and QoS. Finally all device management settings such as access rights, MGT port settings, service routes, log management as well as software and content updates are located under the Device tab.

## 4.2 Cisco Adaptive Security Appliance (ASA) 5505

The Cisco ASA 5505 is a network security appliance running Cisco Systems' IOS operating system. The device has a built-in 8-port Layer 2 Ethernet switch, requiring logical VLAN interfaces for Layer 3 connectivity, as well as a console port and two USB ports. ASA 5505 integrates a stateful, zone-based firewall with IPsec/SSL VPN concentrator and IPS engine.

Cisco ASA 5505 has the following management interfaces:

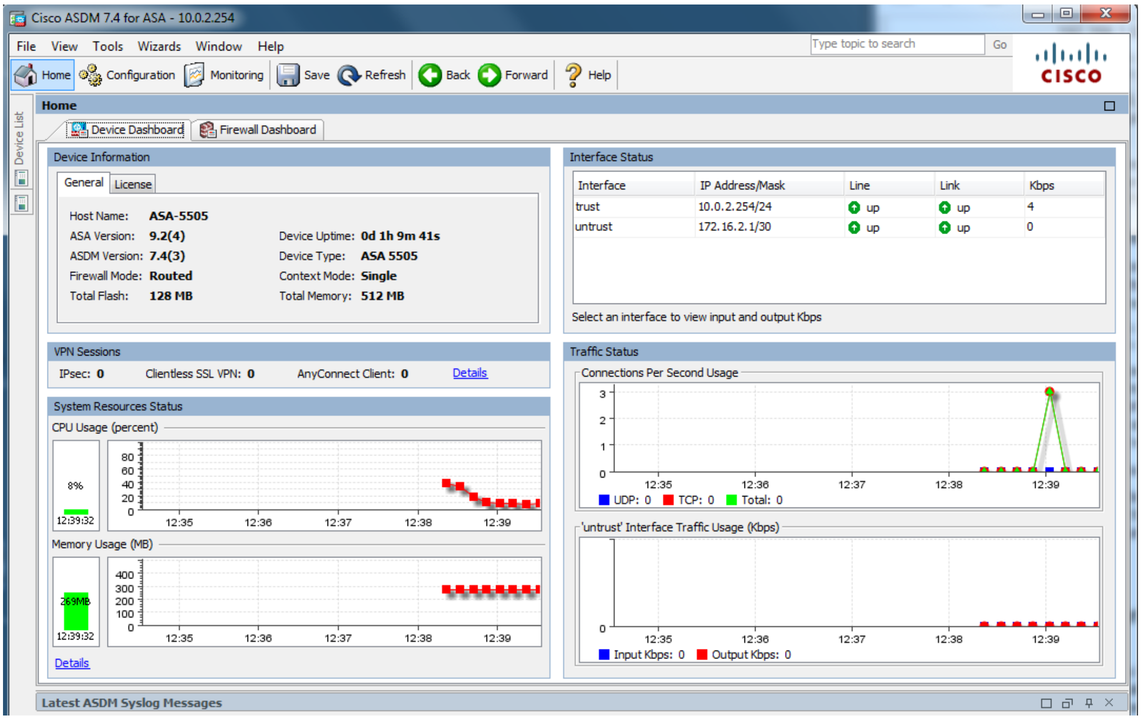
- Web-based GUI best known as the Adaptive Security Device Manager (ASDM) over HTTP or HTTPS via an Ethernet port
- CLI either over Telnet or SSH via an Ethernet port or via the console port.

For initial configuration the console port should be used. While the factory default configuration comes with pre-configured VLAN1 and VLAN2 interfaces, we will start with a clean start-up configuration. In this case access to the ASDM needs to be explicitly configured. To allow an internal host 10.0.2.1, connected to Ethernet0/0 port, access the ASDM GUI, the interface, HTTP server service and an access filter need to be configured to allow HTTP/HTTPS requests. Listing 1 below shows how to allow a host to access the ASDM.

```
interface Vlan10
 nameif trust
 security-level 100
 ip address 10.0.2.254 255.255.255.0
!
interface Ethernet0/0
 switchport access vlan 10
 no shutdown
!
http server enable
http 10.0.2.1 255.255.255.255 trust
```

Listing 1. ASA ASDM Management Configuration.

This study goes through the ASA device configurations via CLI and uses both ASDM and CLI for troubleshooting. The ASDM is illustrated in Figure 6 below.


Cisco ASDM 7.4 for ASA - 10.0.2.254
File View Tools Wizards Window Help
Type topic to search Go
Home Configuration Monitoring Save Refresh Back Forward Help
Home
Device List
Device Dashboard
Firewall Dashboard
General License
CPU Usage (percent)
8%
12:39:32
80 60 40 20 0
12:35 12:36 12:37 12:38 12:39
Memory Usage (MB)
269MB
12:39:32
400 300 200 100 0
Details
Connections Per Second Usage
3 2 1 0
12:35 12:36 12:37 12:38 12:39
🟦 UDP: 0 🟥 TCP: 0 🟩 Total: 0
'untrust' Interface Traffic Usage (Kbps)
0
🟦 Input Kbps: 0 🟥 Output Kbps: 0
Latest ASDM Syslog Messages

Device Information

Host Name: ASA-5505
ASA Version: 9.2(4)
ASDM Version: 7.4(3)
Firewall Mode: Routed
Total Flash: 128 MB

Device Uptime: 0d 1h 9m 41s
Device Type: ASA 5505
Context Mode: Single
Total Memory: 512 MB

Interface Status

| Interface | IP Address/Mask | Line | Link | Kbps |
|---|---|---|---|---|
| trust | 10.0.2.254/24 | up | up | 4 |
| untrust | 172.16.2.1/30 | up | up | 0 |

Select an interface to view input and output Kbps

VPN Sessions

IPsec: 0 Clientless SSL VPN: 0 AnyConnect Client: 0 Details

System Resources Status

Traffic Status

Figure 6. ASA 5505 ASDM Device Dashboard

The Device Dashboard under Home view displays general device information, system resource and interface statuses, and traffic levels. The adjacent Firewall Dashboard draws graphs for session and packet filtering statistics. The Configuration tab allows the configuration of interfaces, routing protocols, firewall policies, SSL/IPsec VPNs and Management services. The Monitoring tab displays interface statuses, routing tables, VPN session details and statistics, and logging data.

The ASDM top panel additionally holds useful features under File, Tools and Wizards menus. The File drop-down menu lists configuration management functionalities. Tools menu contains troubleshooting functionalities such as Packet Tracer, Ping and Traceroute tools, updating services and even allows usage of CLI via ASDM. Wizards menu contains step-by-step setups for device start-up configuration, VPN configuration as well as a Packet Capture Wizard for capturing specified traffic passing the ASA firewall.

## 4.3 Network Topology and Addressing

The network topology comprises the two firewalls, PA-200 and ASA 5505, a router representing a third-party hop in-between the firewalls and a host behind each firewall.

The firewalls act as the gateway for the two sites allowing the sites' private networks to connect with the use of a VPN. Figure 7 below illustrates the network topology.

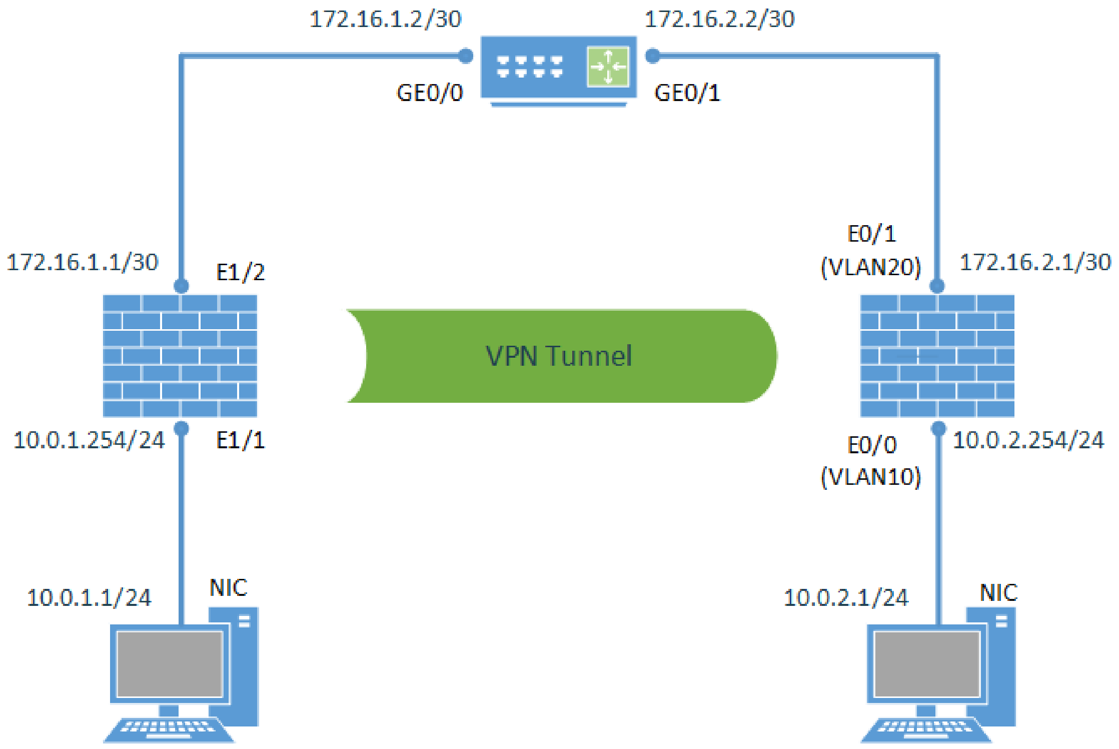
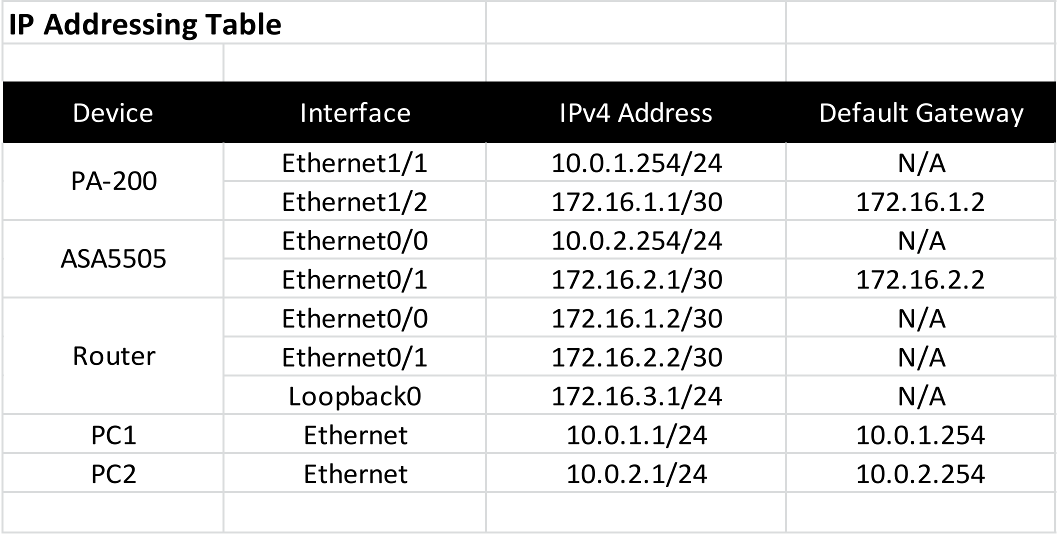


Figure 7. Network topology diagram.

The site behind PA-200 will have 10.0.1.0/24 as its internal Local Area Network (LAN). PA-200 site's internal host's Network Interface Card (NIC) is configured with the IPv4 address 10.0.1.1, subnet mask 255.255.255.0 and default gateway 10.0.1.254. Similarly the site behind ASA 5505 uses 10.0.2.0/24 as its LAN while the site's host is configured with IPv4 address 10.0.2.1, subnet mask 255.255.255.0 and default gateway. Although third-party networks, the public network facing interfaces on PA-200 and ASA 5505 have 172.16.1.0/30 and 172.16.2.0/30 respectively as their upstream networks. All device connections use standard Ethernet cables (RJ-45). Table 1 presents the whole IP addressing scheme.

Table 1. IP Addressing scheme



**IP Addressing Table**

| Device | Interface | IPv4 Address | Default Gateway |
|---|---|---|---|
| PA-200 | Ethernet1/1 | 10.0.1.254/24 | N/A |
| | Ethernet1/2 | 172.16.1.1/30 | 172.16.1.2 |
| ASA5505 | Ethernet0/0 | 10.0.2.254/24 | N/A |
| | Ethernet0/1 | 172.16.2.1/30 | 172.16.2.2 |
| Router | Ethernet0/0 | 172.16.1.2/30 | N/A |
| | Ethernet0/1 | 172.16.2.2/30 | N/A |
| | Loopback0 | 172.16.3.1/24 | N/A |
| PC1 | Ethernet | 10.0.1.1/24 | 10.0.1.254 |
| PC2 | Ethernet | 10.0.2.1/24 | 10.0.2.254 |

In addition to the topology view in Figure 7, the router will get its Loopback0 interface configured when NAT-T is covered later in the Experimenting and Troubleshooting section. This will give the router a distinct pool of addresses for NAT.

## 5 Device Configuration

This section will cover the steps and commands required to configure the devices in the topology. The host computers are configured with static IPv4 addresses and default gateways according to Table 1. The third-party router only has its interfaces configured, thus only routing between its directly connected networks 172.16.1.0/30 and 172.16.2.0/30.

The configuration steps or commands may vary with different software versions. The software versions used in this study are as follows:

- PA-200 software version 6.1.4
- ASA 5505 software version 9.2(4)
- ASA 5505 ASDM version 7.4(3).

### 5.1 Configuring PA-200

In the case of PA-200 the GUI is recommended over CLI when applying configurations. As the configuration file is structured in a hierarchical manner, dependencies caused by virtual routers and virtual systems can be confusing. GUI covers the required dependencies to alleviate confusion. In case all necessary dependencies are not handled before committing the configuration changes, the commit prompt will issue a warning.

Before beginning, due to choosing a Layer 3 deployment, the default virtual-wire configuration should be purged as the Ethernet1/1 and Ethernet1/2 interfaces are initially bound and allow traffic pass the PA-200 firewall transparently without switching or routing. Via CLI we execute the following commands under the configuration mode as shown in Listing 2 below.

```
delete network virtual-wire default-vwire
delete zone trust
delete zone untrust
set network interface ethernet ethernet1/1 layer3
set network interface ethernet ethernet1/2 layer3
```

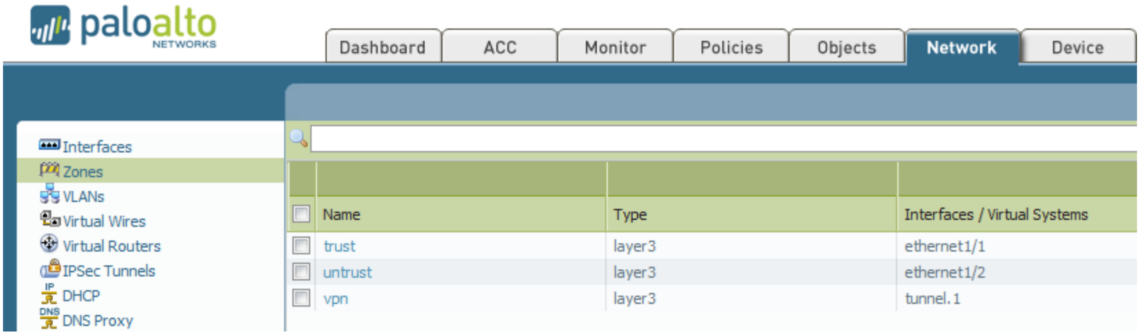
Listing 2. Preparing PA-200 for Layer 3 deployment.

While the MGT port can be used access the GUI with its default settings, in this study the internal LAN interface will be configured to allow access to the device GUI as shown in Listing 3 below.

```
set network profiles interface-management-profile MGMT https yes
set network profiles interface-management-profile MGMT ssh yes
set network profiles interface-management-profile MGMT ping yes
set network interface ethernet ethernet1/1 layer3 ip 10.0.1.254/24
set network interface ethernet ethernet1/1 layer3 interface-
management-profile MGMT
```

Listing 3. PA-200 GUI management configuration.

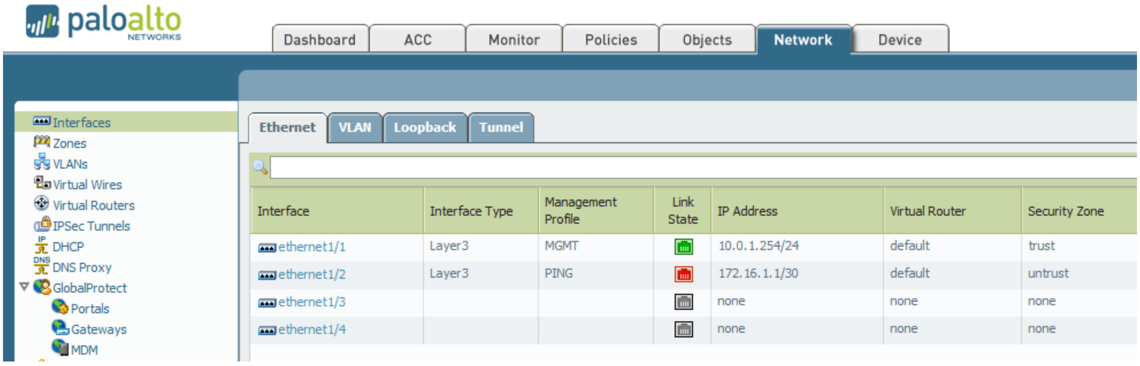
With management access allowed through the data port Ethernet1/1, the host configured with IPv4 address 10.0.1.1/24 and connected to the aforementioned port can now access https://10.0.1.254 via its web browser. On the GUI Network tab we will create three Layer 3 zones named ‘trust’, ‘untrust’ and ‘vpn’ for policies and traffic monitoring. By default zones named ‘trust’ and ‘untrust’ existed as virtual-wire type zones.



| □ | Name | Type | Interfaces / Virtual Systems |
|---|---|---|---|
| □ | trust | layer3 | ethernet1/1 |
| □ | untrust | layer3 | ethernet1/2 |
| □ | vpn | layer3 | tunnel.1 |

Figure 8. PA-200 Network Zones

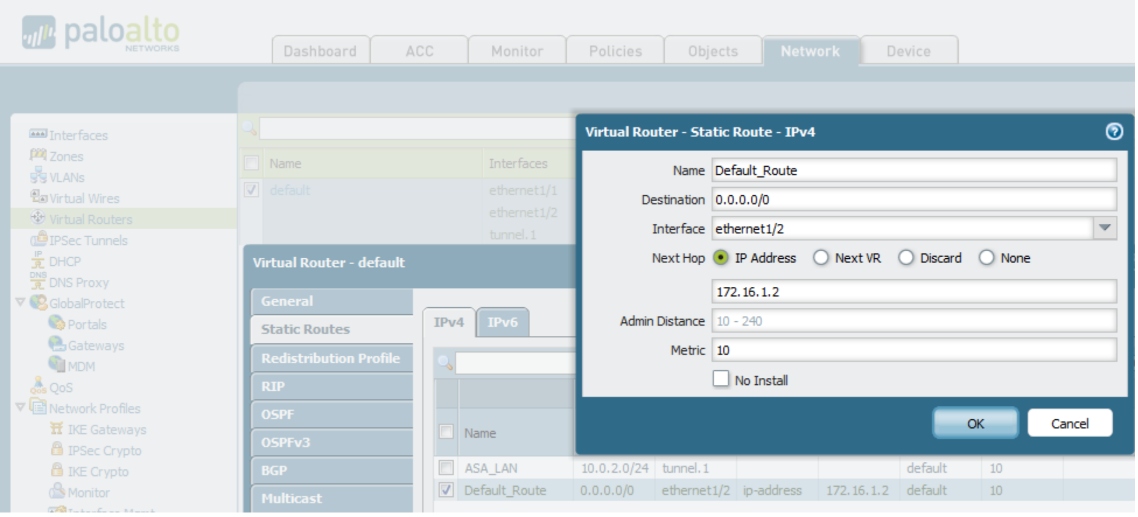
Next we will finish the interface configurations. Interfaces are configured as Layer 3 interfaces with IP addressing according to Table 1. Choose ‘default’ as the virtual router and assign Ethernet1/1 to ‘trust’ zone and Ethernet1/2 to ‘untrust’ zone.



| Interface | Interface Type | Management Profile | Link State | IP Address | Virtual Router | Security Zone |
|---|---|---|---|---|---|---|
| ethernet1/1 | Layer3 | MGMT | 🟩 | 10.0.1.254/24 | default | trust |
| ethernet1/2 | Layer3 | PING | 🟥 | 172.16.1.1/30 | default | untrust |
| ethernet1/3 | | | ▨ | none | none | none |
| ethernet1/4 | | | ▨ | none | none | none |

Figure 9. PA-200 Network Interfaces

To finalize the basic configurations, a default static route towards the third-party Router will be configured to establish routing between the devices. This is set under Network > Virtual Routers.



Virtual Router - Static Route - IPv4

Name: Default_Route
Destination: 0.0.0.0/0
Interface: ethernet1/2
Next Hop: ● IP Address ○ Next VR ○ Discard ○ None
172.16.1.2
Admin Distance: 10 - 240
Metric: 10
□ No Install

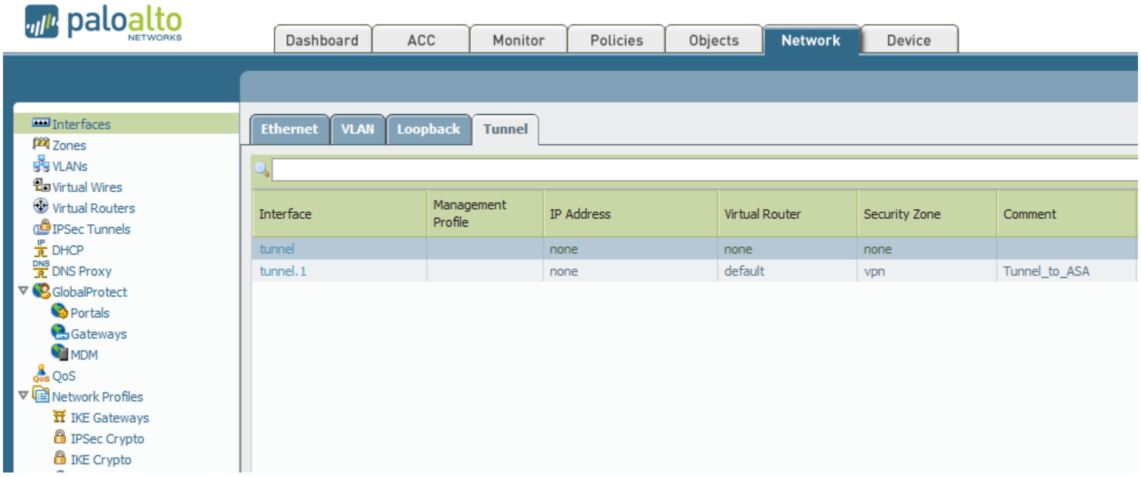
OK Cancel

| | Name | Interfaces |
|---|---|---|
| ☑ | default | ethernet1/1<br>ethernet1/2<br>tunnel.1 |

| | Name | | | | | | |
|---|---|---|---|---|---|---|---|
| □ | ASA_LAN | 10.0.2.0/24 | tunnel.1 | | | default | 10 |
| ☑ | Default_Route | 0.0.0.0/0 | ethernet1/2 | ip-address | 172.16.1.2 | default | 10 |

Figure 10. PA-200 Default IPv4 Routing

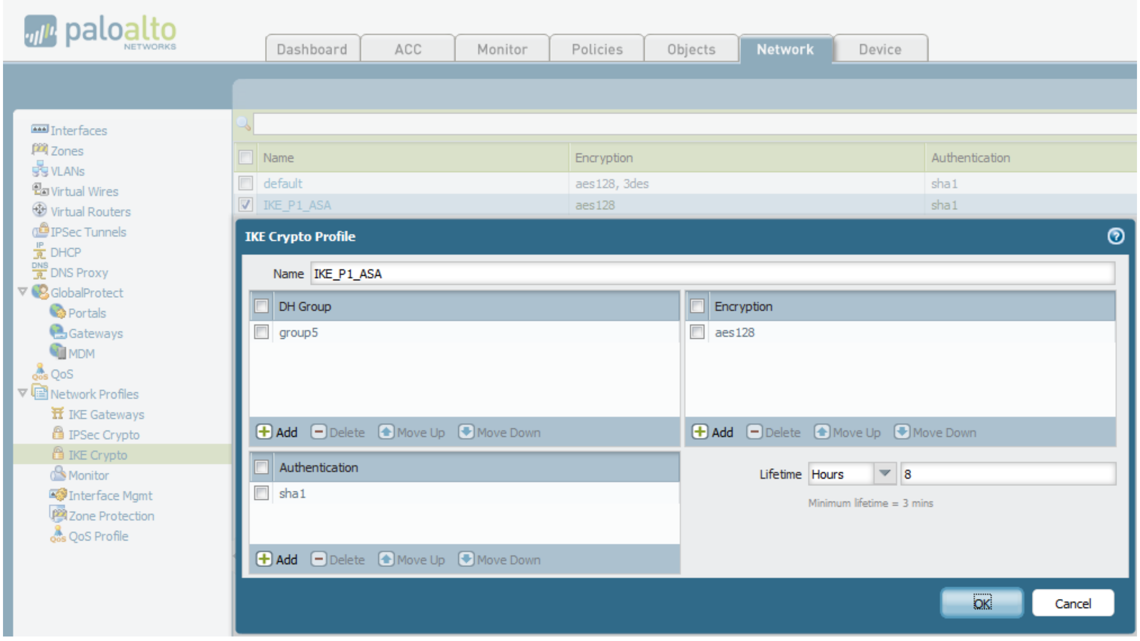
To establish a tunnel, a tunnel interface needs to be configured. The Tunnel interface 'tunnel.1' will be applied to the same 'default' Virtual Router instance as the other interfaces, but it will be assigned to the 'vpn' zone to more conveniently apply security policies. An IPv4 address is not required, since ASA will not be using them either. It would, however, allow the usage of the Tunnel Monitor feature under IPsec Tunnel settings, which can be used to poll the other side.



| Interface | Management Profile | IP Address | Virtual Router | Security Zone | Comment |
|---|---|---|---|---|---|
| tunnel | | none | none | none | |
| tunnel.1 | | none | default | vpn | Tunnel_to_ASA |

Figure 11. PA-200 Tunnel Interfaces

To start off the policy negotiations, IKE parameters will be configured. Much like with other features, a default IKE Crypto Profile already exists. Let us create a new one with a comprehensive name 'IKE_P1_ASA' identifying that we are configuring IKE Phase 1 settings to be used in negotiations with ASA. As displayed in Figure 12 below, AES128 is used for encryption, SHA-1 for authentication and DH group 5 for key exchange. IKE SA lifetime will be set to 8 hours.



| ☐ Name | Encryption | Authentication |
|---|---|---|
| ☐ default | aes128, 3des | sha1 |
| ☑ IKE_P1_ASA | aes128 | sha1 |

IKE Crypto Profile

Name IKE_P1_ASA

| ☐ DH Group |
|---|
| ☐ group5 |

| ☐ Encryption |
|---|
| ☐ aes128 |

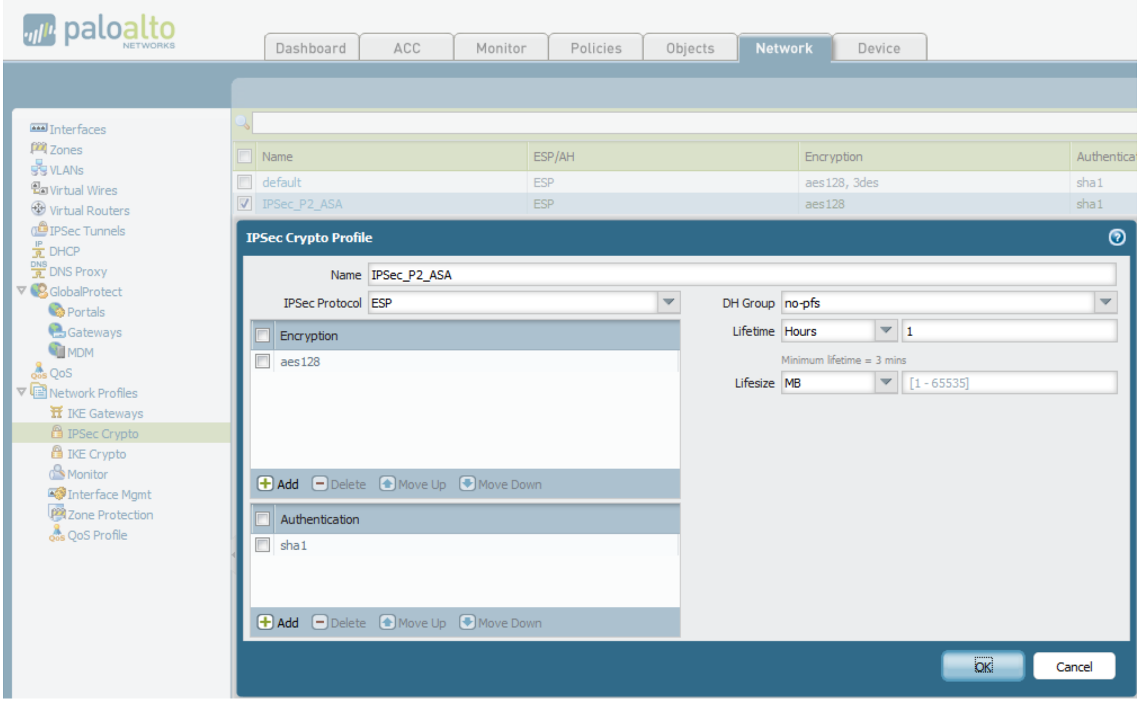
| ☐ Authentication |
|---|
| ☐ sha1 |

Lifetime Hours 8

Minimum lifetime = 3 mins

Figure 12. PA-200 IKE Crypto Profile

Continuing onto Phase 2, again a default set of parameters exists. This time the IPsec Crypto Profile will be named ‘IPsec_P2_ASA’ to distinguish it as Phase 2 parameter set. Here as well AES128 and SHA-1 will be selected. ESP will be selected as the IP-sec protocol. IPsec SA lifetime will be set to 1h. No PFS will be required due to the relatively low SA lifetimes.



| ☐ | Name | ESP/AH | Encryption | Authentica |
|---|---|---|---|---|
| ☐ | default | ESP | aes128, 3des | sha1 |
| ☑ | IPSec_P2_ASA | ESP | aes128 | sha1 |

IPSec Crypto Profile

Name IPSec_P2_ASA

IPSec Protocol ESP

DH Group no-pfs

Lifetime Hours 1

Minimum lifetime = 3 mins

Lifesize MB [1 - 65535]

☐ Encryption

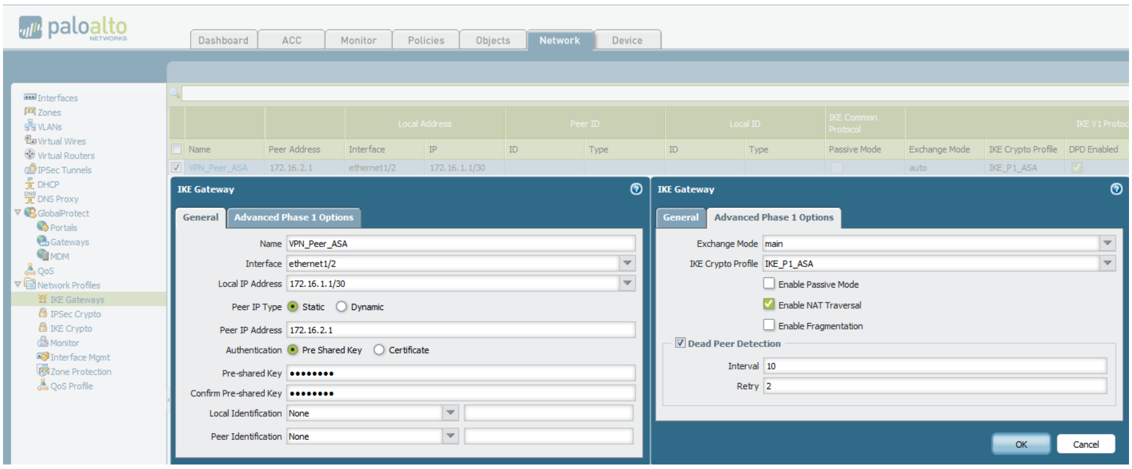
☐ aes128

☐ Authentication

☐ sha1

Figure 13. PA-200 IPsec Crypto Profile

In order to bind all details about the VPN peer together an IKE Gateway will be defined. Here we set the physical interface to reach the VPN peer, VPN peer identity and authentication method as PSK. The peer has a static IPv4 address 172.16.2.1 and it will also be used to identify the peer. In the Advanced Options tab the Exchange Mode is by default 'auto' which allows both MM and AM. Here we also map the IKE Crypto Profile 'IKE_P1_ASA' to this IKE Gateway.



| | | | Local Address | | Peer ID | | Local ID | | IKE Common Protocol | | | IKE V1 Protoc |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ☐ | Name | Peer Address | Interface | IP | ID | Type | ID | Type | Passive Mode | Exchange Mode | IKE Crypto Profile | DPD Enabled |
| ☑ | VPN_Peer_ASA | 172.16.2.1 | ethernet1/2 | 172.16.1.1/30 | | | | | ☐ | auto | IKE_P1_ASA | ✓ |

IKE Gateway

General | Advanced Phase 1 Options

Name VPN_Peer_ASA

Interface ethernet1/2

Local IP Address 172.16.1.1/30

Peer IP Type ● Static ○ Dynamic

Peer IP Address 172.16.2.1

Authentication ● Pre Shared Key ○ Certificate

Pre-shared Key ••••••••

Confirm Pre-shared Key ••••••••

Local Identification None

Peer Identification None

IKE Gateway

General | Advanced Phase 1 Options

Exchange Mode main

IKE Crypto Profile IKE_P1_ASA

☐ Enable Passive Mode

☑ Enable NAT Traversal

☐ Enable Fragmentation

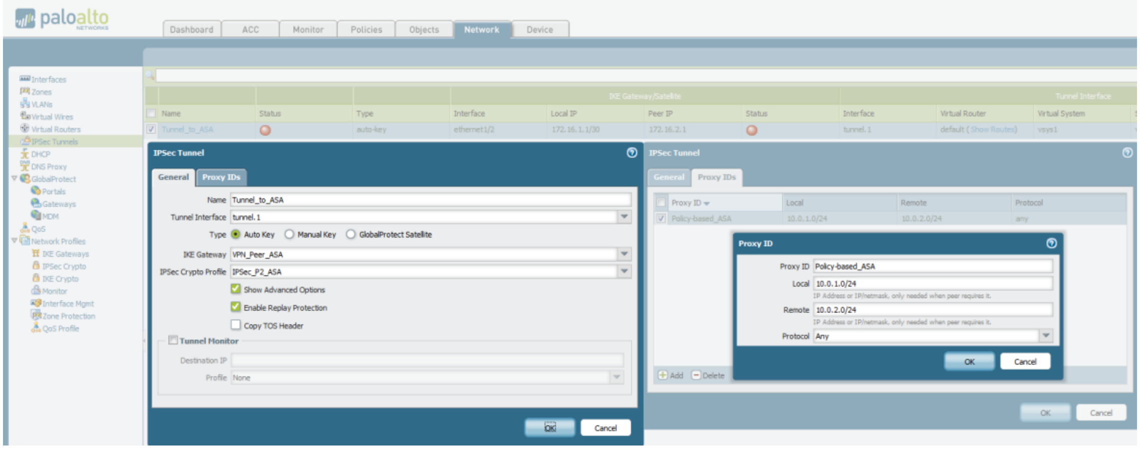
☑ Dead Peer Detection

Interval 10

Retry 2

Figure 14. PA-200 IKE Gateway

The Last step in configuring the VPN tunnel is creating an IPsec Tunnel. This IPsec Tunnel named ‘Tunnel_to_ASA’ will be bound to the Tunnel Interface ‘tunnel.1’. This is also where the IKE Gateway and IPsec Crypto Profile are linked together. If the Tunnel Interfaces had IP addresses assigned, we can enable the Tunnel Monitor option here. The QM Proxy-IDs or Encryption domains are also defined here, which is an imperative step when support for policy-based VPNs is required, like when peering with an ASA.



|  | Name | Status | Type | Interface | Local IP | Peer IP | Status | Interface | Virtual Router | Virtual System |
|---|---|---|---|---|---|---|---|---|---|---|
| ☑ | Tunnel_to_ASA | ● | auto-key | ethernet1/2 | 172.16.1.1/30 | 172.16.2.1 | ● | tunnel.1 | default ( Show Routes) | vsys1 |

IPSec Tunnel — General

Name: Tunnel_to_ASA
Tunnel Interface: tunnel.1
Type: ● Auto Key ○ Manual Key ○ GlobalProtect Satellite
IKE Gateway: VPN_Peer_ASA
IPSec Crypto Profile: IPSec_P2_ASA
☑ Show Advanced Options
☑ Enable Replay Protection
□ Copy TOS Header
□ Tunnel Monitor
Destination IP:
Profile: None

IPSec Tunnel — Proxy IDs

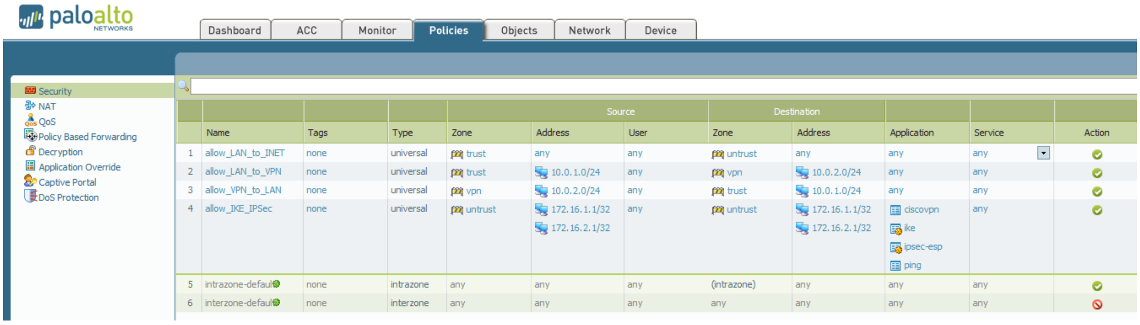
|  | Proxy ID | Local | Remote | Protocol |
|---|---|---|---|---|
| ☑ | Policy-based_ASA | 10.0.1.0/24 | 10.0.2.0/24 | any |

Proxy ID

Proxy ID: Policy-based_ASA
Local: 10.0.1.0/24
IP Address or IP/netmask, only needed when peer requires it.
Remote: 10.0.2.0/24
IP Address or IP/netmask, only needed when peer requires it.
Protocol: Any

Figure 15. PA-200 IPsec Tunnel

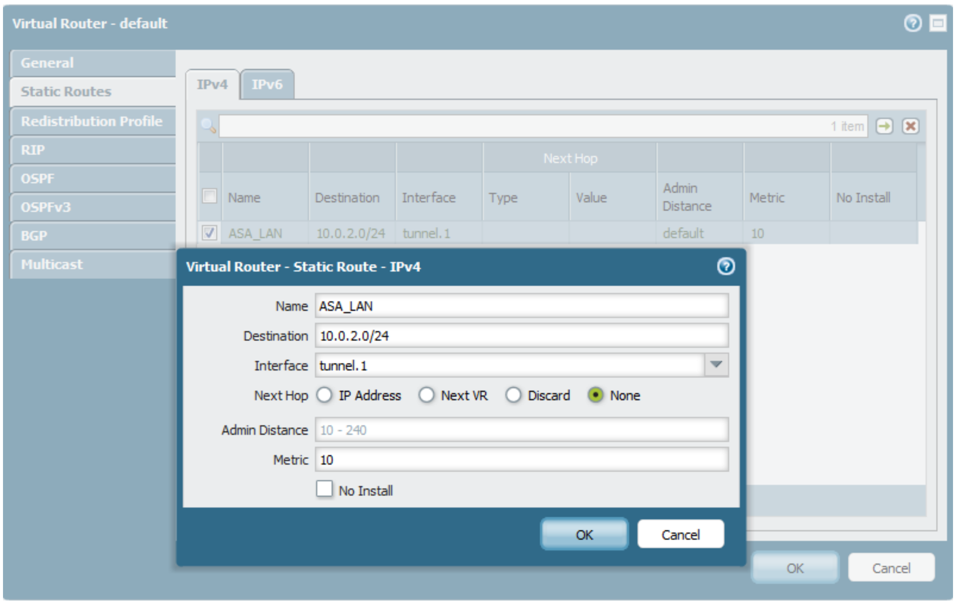
In order to allow and monitor the traffic flow related to the VPN, we create policies to allow bidirectional traffic between the local network 10.0.1.0/24 and remote network 10.0.2.0/24. We also explicitly allow incoming IKE and IPsec traffic for traffic logging.



|  |  |  |  | Source |  |  | Destination |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|
|  | Name | Tags | Type | Zone | Address | User | Zone | Address | Application | Service | Action |
| 1 | allow_LAN_to_INET | none | universal | trust | any | any | untrust | any | any | any | ✓ |
| 2 | allow_LAN_to_VPN | none | universal | trust | 10.0.1.0/24 | any | vpn | 10.0.2.0/24 | any | any | ✓ |
| 3 | allow_VPN_to_LAN | none | universal | vpn | 10.0.2.0/24 | any | trust | 10.0.1.0/24 | any | any | ✓ |
| 4 | allow_IKE_IPSec | none | universal | untrust | 172.16.1.1/32<br>172.16.2.1/32 | any | untrust | 172.16.1.1/32<br>172.16.2.1/32 | ciscovpn<br>ike<br>ipsec-esp<br>ping | any | ✓ |
| 5 | intrazone-defaul | none | intrazone | any | any | any | (intrazone) | any | any | any | ✓ |
| 6 | interzone-defaul | none | interzone | any | any | any | any | any | any | any | ⦸ |

Figure 16. PA-200 Security Policies

Lastly, a static route to the remote network 10.0.2.0/24 is required for traffic flow. These packets will be routed to the Tunnel Interface. No next-hop IPv4 address is configured, since the Interface does not use IP addresses.



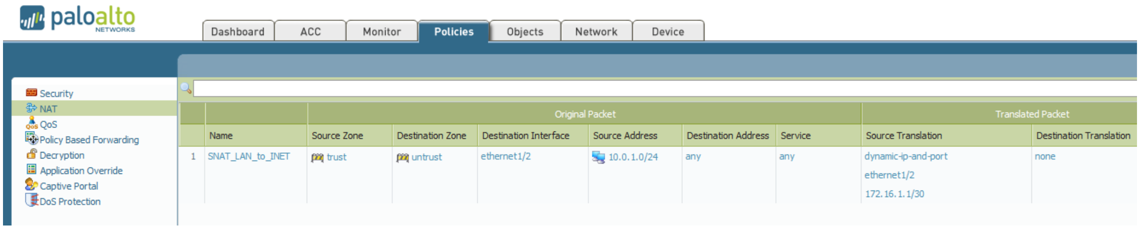
|  |  |  | Next Hop |  |  |  |  |
|---|---|---|---|---|---|---|---|
| □ Name | Destination | Interface | Type | Value | Admin Distance | Metric | No Install |
| ☑ ASA_LAN | 10.0.2.0/24 | tunnel.1 |  |  | default | 10 |  |

Virtual Router - Static Route - IPv4

|  |  |
|---|---|
| Name | ASA_LAN |
| Destination | 10.0.2.0/24 |
| Interface | tunnel.1 |
| Next Hop | ○ IP Address ○ Next VR ○ Discard ● None |
| Admin Distance | 10 - 240 |
| Metric | 10 |
|  | □ No Install |

Figure 17. PA-200 Static Route to remote network

Optionally, we can implement NAT or Port Address Translation (PAT) to allow the internal LAN to access public networks. NAT operation is performed before IPsec, so it must be accounted for. However, as we have assigned a separate zone ‘vpn’ for the VPN tunnel interface, there will be no need to exempt tunnel traffic from address translation. Figure 18 displays the configured NAT rule on PA-200.



|  |  | Original Packet |  |  |  |  |  | Translated Packet |  |
|---|---|---|---|---|---|---|---|---|---|
|  | Name | Source Zone | Destination Zone | Destination Interface | Source Address | Destination Address | Service | Source Translation | Destination Translation |
| 1 | SNAT_LAN_to_INET | trust | untrust | ethernet1/2 | 10.0.1.0/24 | any | any | dynamic-ip-and-port<br>ethernet1/2<br>172.16.1.1/30 | none |

Figure 18. PA-200 NAT Rules

According to Figure 18 the NAT (PAT) rule will use the ‘untrust’ interface’s IP address for traffic sourced from internal network 10.0.1.0/24 and destined to any destination network belonging to ‘untrust’ zone.

## 5.2 Configuring ASA 5505

The configuration wizards in ASA 5505's CLI and ASDM tend to create an unnecessary number of default objects; thus the configurations will be done manually via CLI.

Before the IPsec VPN tunnel can be configured the interfaces and default routing need to be configured. As ASA's switch ports operate at Layer 2, we also configure VLAN interfaces to which the switch ports are assigned. For the site's local network 10.0.2.0/24 we assign the highest security level and call it our 'trust' zone. The host computer is connected to the Ethernet0/0 switch port. For the site's third-party-connected network 172.16.2.0/30 we assign the lowest security level and call it our 'untrust' zone. The third-party router is connected to the Ethernet0/1 switch port. Finally, we tell ASA to route packets towards the third-party upstream router by default. Listing 4 below shows how to set up this preliminary configuration.

```
interface Vlan10
 nameif trust
 security-level 100
 ip address 10.0.2.254 255.255.255.0
!
interface Vlan20
 nameif untrust
 security-level 0
 ip address 172.16.2.1 255.255.255.252
!
interface Ethernet0/0
 switchport access vlan 10
 no shutdown
!
interface Ethernet0/1
 switchport access vlan 20
 no shutdown
!
route untrust 0.0.0.0 0.0.0.0 172.16.2.2
```

Listing 4. ASA's preliminary configuration.

Next we will define the traffic we wish to have traverse the VPN tunnel. For this we define an ACL (Access Control List) stating the source (10.0.2.0/24) and destination (10.0.1.0/24) networks. Let us call this ACL 'PROXY-ID-ACL'. For future referencing, we create network objects for local and remote networks. Listing 5 below shows the equivalent configuration.

```
object network 10.0.1.0_24
 subnet 10.0.1.0 255.255.255.0
object network 10.0.2.0_24
 subnet 10.0.2.0 255.255.255.0
access-list  PROXY-ID-ACL  extended  permit  ip  object  10.0.2.0_24
object 10.0.1.0_24
```

Listing 5. Configuring ASA's network objects and ACLs.

The configuration for the actual VPN tunnel and peer is shown in Listing 6 below. Exclamation marks are used to separate the commands into logical sections.

```
crypto ikev1 policy 10
 authentication pre-share
 encryption aes
 hash sha
 group 5
 lifetime 28800
crypto ikev1 enable untrust
!
tunnel-group 172.16.1.1 type IPsec-l2l
tunnel-group 172.16.1.1 IPsec-attributes
 ikev1 pre-shared-key palocisco
!
crypto IPsec ikev1 transform-set ESP-AES1-SHA esp-aes esp-sha-hmac
!
crypto map VPN_MAP 1 match address PROXY-ID-ACL
crypto map VPN_MAP 1 set peer 172.16.1.1
crypto map VPN_MAP 1 set ikev1 transform-set ESP-AES1-SHA
crypto  map  VPN_MAP  1  set  security-association  lifetime  seconds
3600
crypto map VPN_MAP interface untrust
```

Listing 6. ASA's VPN tunnel and peer configuration.

For the IKE negotiations we first need at least one IKEv1 policy – also known as the ISAKMP policy. We define all the security parameters for the IKE policy set and enable the negotiations on the 'untrust' interface which terminates the VPN. Since PSK is used as the authentication method, let us next configure a tunnel group for the VPN peer. The tunnel is configured as Site-to-Site – or LAN-to-LAN (L2L) as ASA understands it. In the tunnel group we define the PSK (palocisco) for this IPsec VPN peer.

With IKE settings configured, next we define our IPsec parameters. For this we create a transform-set 'ESP-AES1-SHA'. By default the tunnel mode is used. Finally we link the VPN peer, transform-set and ACL together with the use of crypto map 'VPN_MAP'. The crypto map is then applied to the same interface as the IKE policies. PFS is not configured.

As NAT is a common reason why traffic does not enter a VPN tunnel, in addition to incorrect routing and security policies denying the traffic, let us additionally implement it for the upstream traffic that will not pass the VPN tunnel. For this purpose we will need an extra exempt NAT rule in addition to the basic NAT according to Listing 7 below.

```
nat   (trust,untrust)   source   static   10.0.2.0_24   10.0.2.0_24
destination  static  10.0.1.0_24  10.0.1.0_24  no-proxy-arp  route-
lookup
nat (trust,untrust) after-auto source dynamic any interface
```

Listing 7. ASA's NAT rules.

The first NAT rule, an exempt NAT rule, specifies that traffic sourced from network 10.0.2.0/24 (zone 'trust') and destined to network 10.0.1.0/24 (zone 'untrust') is not to be translated. The second NAT rule, a dynamic PAT rule, states that any traffic sourced from the 'trust' zone and destined to the 'untrust' zone is to be dynamically translated to use the outgoing interface's IP address as a new source and get a new custom source port assigned to this session. The keyword `after-auto` will guarantee that the more specific NAT rules will have a higher priority during a NAT lookup.

# 6 VPN Experimenting and Troubleshooting

In this chapter an issue at a time will be introduced to the VPN tunnel, reversing the troubleshooting process of using commands and tools to determine the cause of failure then deriving the proper corrective measure. In this case the cause of failure is known beforehand, and unfolding the methods, to determine that this very incident is the cause of failure, is the objective. For the test traffic simple ICMP Requests (Ping) sourced by the hosts are used. At first the available troubleshooting tools will be covered.

On PA-200's GUI the following troubleshooting tools are available:

- Monitor > Logs > System
- Monitor > Logs > Traffic
- Monitor > Session Browser
- Monitor > Packet Capture.

System Log helpfully logs IKE/IPsec-related events among other events. Traffic Log lists allowed and denied sessions matching the Security Policies that have logging enabled. While Traffic Log concentrates more on session history and traffic statistics, Session Browser offers details regarding the ongoing sessions and their flows. Packet Capture can capture packets at different stages of the firewall; received, inspected, transmitted and dropped. The output can then be viewed in the Wireshark format.

On PA-200's CLI for convenience's sake, we will first modify the console/terminal output such that it will use the whole screen worth of space to print the output.

```
set cli terminal height 500
set cli terminal width 500
```

For debugging IKE negotiations we determine our logging level: normal or debug. The normal level, which is the default, will generally report about successes and failures, but typically the debug level is required to find the cause. For real-time CLI logging we specify `tail` command to follow the output of the management log called 'ikemgr.log' in real-time. The log can also be read later by replacing `tail` with `less`, which will print the whole log file's contents.

```
debug ike global on debug
tail follow yes mp-log ikemgr.log
```

To verify the tunnel's parameters, SPI, lifetime and statistics, we use the `show vpn flow` command as such:

```
show vpn flow tunnel-id 1
```

On ASA 5505's ASDM the following troubleshooting tools are available:

- Monitor > VPN
- Monitor > Logging
- Wizards > Packet Capture Wizard.

ASDM's VPN Monitor is a neat utility to quickly verify or terminate establishment of VPN sessions, SA parameters and statistics. ASDM's Logging feature, when properly filtered, is capable of recording the essential events in IKE/IPsec negotiations such as phase completions, SA establishment and termination and any blatant issues during negotiations. The logging level used in this study is the default informational, just less detailed than the debugging level. Packet Capture Wizard can be used to confirm traffic flow through ASA.

For VPN debugging ASA 5505 via CLI the following commands should prove useful:

```
debug crypto condition peer 172.16.1.1
debug crypto ikev1 127
debug crypto ipsec 127
```

The command `debug crypto condition` is used to narrow the debugging down to only the interesting output, in this case the tunnel with the peer 172.16.1.1. The command `debug crypto ikev1 127` will trigger a debug output via the console terminal. This will cover both IKEv1 phases, while the `debug crypto ipsec` variant will provide additional data regarding the IKEv1 Phase 2. The following integer can vary in range 1–255 determining the debug level. Level 127 is generally informative and compact enough. In comparison the debug level 254 will also include ISAKMP header information and is required in order to view the contents of the message payloads such as the SA payload.

To verify the state of an IPsec tunnel and the related SAs via CLI we type:

```
show crypto ikev1 sa
show crypto isakmp sa detail
show crypto IPsec sa
show vpn-sessiondb detail l2l
```

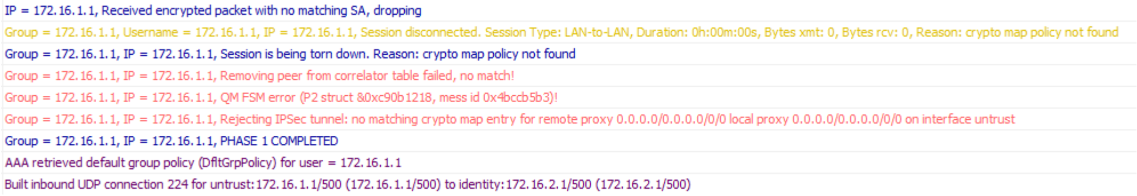
The first displays the type, role and state for each active IKEv1 peer. While IKE SA state is typically of interest to determine how far the negotiations have progressed, the device in the role of a responder generally has easier time to determine the cause(s) of failure. The second covers both IKEv1 and IKEv2 SAs as well as the chosen policy set parameters. The third command displays extended information regarding all IPsec SAs and their statistics. The fourth provides detailed IKEv1, IKEv2 and IPsec session information with all policy parameters, lifetimes and Proxy-IDs present.

Due to the abundant amount of available troubleshooting tools, priority will be given to the concise information provided by PA-200’s GUI System Log and ASA’s ASDM Logging features. Should these fail to catch onto the issue, the secondary measure will be CLI debugging followed by the rest of the methods. The troubleshooting subjects will be divided into CASEs each covering multiple scenarios depending on the experiment arrangements. Any changes done in a scenario are rolled back for the following scenario.

## 6.1 CASE 01: Proxy-ID Mismatch

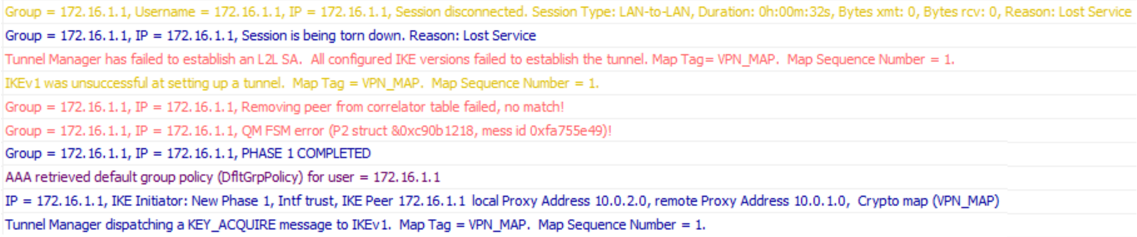
Proxy-ID mismatch indicates that the tunnel endpoints have disagreement regarding which source and destination networks are allowed to traverse the tunnel. Proxy-IDs are negotiated during QM messages 1–2. While the IPsec policy proposal is contained in the SA payload, Proxy-IDs are located in a separate Identification payload. During the configuration phase ASA 5505 also refers to them as ACLs assigned to a crypto map. Three different scenarios were used in experimentation.

In the first scenario Proxy-ID definitions were erased on PA-200, causing IKE Phase 2 to fail. Figures 19 and 20 below portray the issue on ASA 5505 ASDM's end.



```
IP = 172.16.1.1, Received encrypted packet with no matching SA, dropping
Group = 172.16.1.1, Username = 172.16.1.1, IP = 172.16.1.1, Session disconnected. Session Type: LAN-to-LAN, Duration: 0h:00m:00s, Bytes xmt: 0, Bytes rcv: 0, Reason: crypto map policy not found
Group = 172.16.1.1, IP = 172.16.1.1, Session is being torn down. Reason: crypto map policy not found
Group = 172.16.1.1, IP = 172.16.1.1, Removing peer from correlator table failed, no match!
Group = 172.16.1.1, IP = 172.16.1.1, QM FSM error (P2 struct &0xc90b1218, mess id 0x4bccb5b3)!
Group = 172.16.1.1, IP = 172.16.1.1, Rejecting IPSec tunnel: no matching crypto map entry for remote proxy 0.0.0.0/0.0.0.0/0/0 local proxy 0.0.0.0/0.0.0.0/0/0 on interface untrust
Group = 172.16.1.1, IP = 172.16.1.1, PHASE 1 COMPLETED
AAA retrieved default group policy (DfltGrpPolicy) for user = 172.16.1.1
Built inbound UDP connection 224 for untrust:172.16.1.1/500 (172.16.1.1/500) to identity:172.16.2.1/500 (172.16.2.1/500)
```

Figure 19. Proxy-ID mismatch 01: ASA 5505 as receiver



```
Group = 172.16.1.1, Username = 172.16.1.1, IP = 172.16.1.1, Session disconnected. Session Type: LAN-to-LAN, Duration: 0h:00m:32s, Bytes xmt: 0, Bytes rcv: 0, Reason: Lost Service
Group = 172.16.1.1, IP = 172.16.1.1, Session is being torn down. Reason: Lost Service
Tunnel Manager has failed to establish an L2L SA.  All configured IKE versions failed to establish the tunnel. Map Tag= VPN_MAP.  Map Sequence Number = 1.
IKEv1 was unsuccessful at setting up a tunnel.  Map Tag = VPN_MAP.  Map Sequence Number = 1.
Group = 172.16.1.1, IP = 172.16.1.1, Removing peer from correlator table failed, no match!
Group = 172.16.1.1, IP = 172.16.1.1, QM FSM error (P2 struct &0xc90b1218, mess id 0xfa755e49)!
Group = 172.16.1.1, IP = 172.16.1.1, PHASE 1 COMPLETED
AAA retrieved default group policy (DfltGrpPolicy) for user = 172.16.1.1
IP = 172.16.1.1, IKE Initiator: New Phase 1, Intf trust, IKE Peer 172.16.1.1  local Proxy Address 10.0.2.0, remote Proxy Address 10.0.1.0,  Crypto map (VPN_MAP)
Tunnel Manager dispatching a KEY_ACQUIRE message to IKEv1.  Map Tag = VPN_MAP.  Map Sequence Number = 1.
```

Figure 20. Proxy-ID mismatch 01: ASA 5505 as initiator

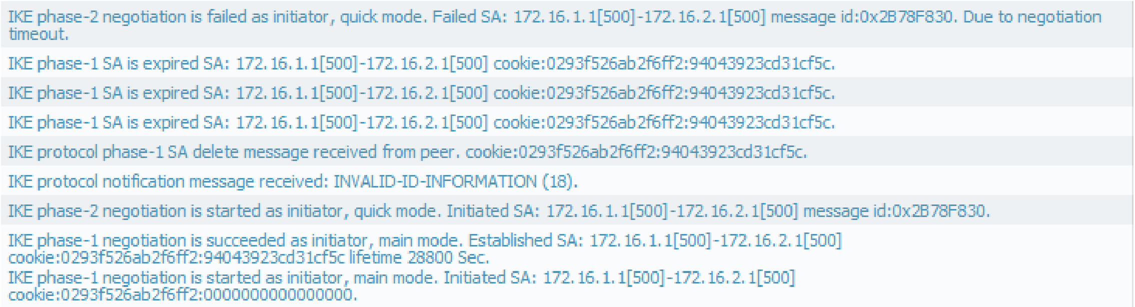
In both the role of a receiver and initiator ASDM Logging triggers errors during IKE Phase 2 QM negotiations. However, only as a receiver is ASDM able to deduce that this is due to a Proxy-ID mismatch. Additionally running `debug crypto ikev1 127` will bring the details shown in the following Listing 8 when ASA is acting as the initiator.

```
[IKEv1 DEBUG]Group = 172.16.1.1, IP = 172.16.1.1,
IKE QM Initiator FSM error history (struct &0xc90b1218)  <state>,
<event>:       QM_DONE,    EV_ERROR-->QM_WAIT_MSG2,    EV_TIMEOUT--
>QM_WAIT_MSG2, NullEvent-->QM_SND_MSG1, EV_SND_MSG-->QM_SND_MSG1,
EV_START_TMR-->QM_SND_MSG1,          EV_RESEND_MSG-->QM_WAIT_MSG2,
EV_TIMEOUT-->QM_WAIT_MSG2, NullEvent
```

Listing 8. ASA stuck at ‘QM_WAIT_MSG2’ stage.

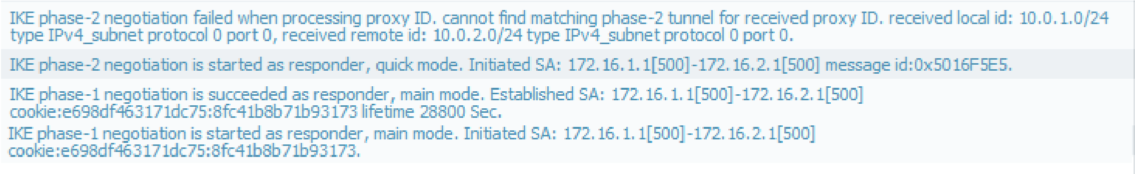
This output essentially does not reveal the issue either, but tells us that the peer does not  send the continuation message or any kind of notification. Thus ASA’s end hangs at the ‘QM_WAIT_MSG2’ state, being left waiting for the second message of QM which never comes.

On PA-200’s end the logging output reflected similar results. Figures 21 and 22 below depict the System Log output on PA-200:



```
IKE phase-2 negotiation is failed as initiator, quick mode. Failed SA: 172.16.1.1[500]-172.16.2.1[500] message id:0x2B78F830. Due to negotiation
timeout.
IKE phase-1 SA is expired SA: 172.16.1.1[500]-172.16.2.1[500] cookie:0293f526ab2f6ff2:94043923cd31cf5c.
IKE phase-1 SA is expired SA: 172.16.1.1[500]-172.16.2.1[500] cookie:0293f526ab2f6ff2:94043923cd31cf5c.
IKE phase-1 SA is expired SA: 172.16.1.1[500]-172.16.2.1[500] cookie:0293f526ab2f6ff2:94043923cd31cf5c.
IKE protocol phase-1 SA delete message received from peer. cookie:0293f526ab2f6ff2:94043923cd31cf5c.
IKE protocol notification message received: INVALID-ID-INFORMATION (18).
IKE phase-2 negotiation is started as initiator, quick mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500] message id:0x2B78F830.
IKE phase-1 negotiation is succeeded as initiator, main mode. Established SA: 172.16.1.1[500]-172.16.2.1[500]
cookie:0293f526ab2f6ff2:94043923cd31cf5c lifetime 28800 Sec.
IKE phase-1 negotiation is started as initiator, main mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500]
cookie:0293f526ab2f6ff2:0000000000000000.
```

Figure 21. Proxy-ID mismatch 01: PA-200 as initiator



```
IKE phase-2 negotiation failed when processing proxy ID. cannot find matching phase-2 tunnel for received proxy ID. received local id: 10.0.1.0/24
type IPv4_subnet protocol 0 port 0, received remote id: 10.0.2.0/24 type IPv4_subnet protocol 0 port 0.
IKE phase-2 negotiation is started as responder, quick mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500] message id:0x5016F5E5.
IKE phase-1 negotiation is succeeded as responder, main mode. Established SA: 172.16.1.1[500]-172.16.2.1[500]
cookie:e698df463171dc75:8fc41b8b71b93173 lifetime 28800 Sec.
IKE phase-1 negotiation is started as responder, main mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500]
cookie:e698df463171dc75:8fc41b8b71b93173.
```

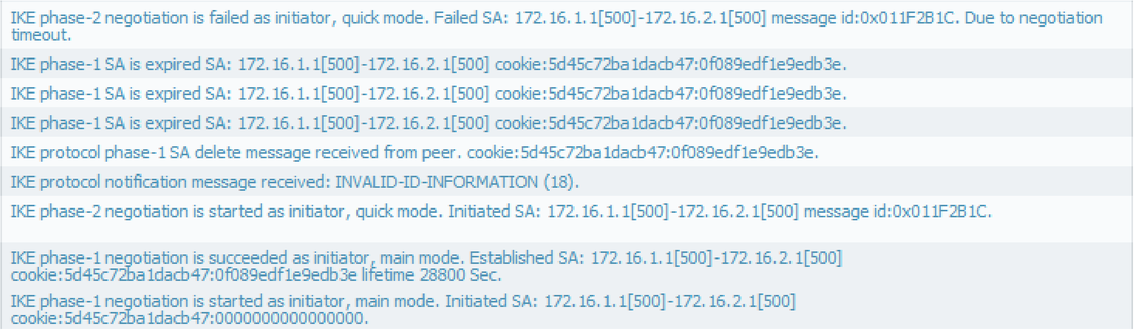
Figure 22. Proxy-ID mismatch 01: PA-200 as receiver

As seen from Figure 21, the notification message ‘INVALID-ID-INFORMATION’ indicates that the contents of the ID payload, containing the Proxy-ID data that PA was offering to ASA, were unacceptable. As a receiver PA-200 realizes that the issue is the peer’s Proxy-ID proposal not matching its own ones when comparing the two.

In the second scenario ASA’s ACL mapping was removed as follows:

```
no crypto map VPN_MAP 1 match address PROXY-ID-ACL
WARNING: The crypto map entry will be incomplete!
```

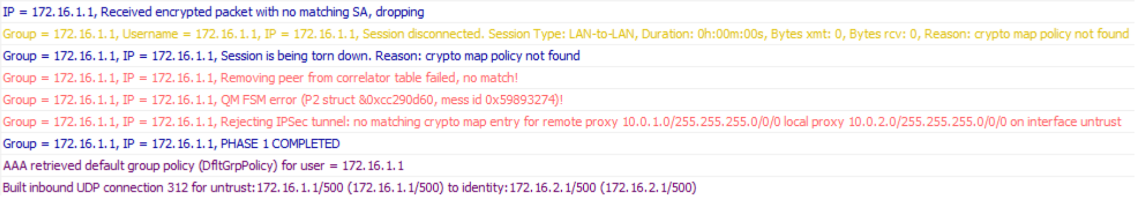
Conveniently, without an ACL mapped to the crypto map, the ASA will not even attempt to establish the tunnel. It will, however, reply to PA-200’s attempts to do so. Figures 23 and 24 show the log entries from both perspectives when PA initiates the negotiations:



```
IKE phase-2 negotiation is failed as initiator, quick mode. Failed SA: 172.16.1.1[500]-172.16.2.1[500] message id:0x011F2B1C. Due to negotiation
timeout.
IKE phase-1 SA is expired SA: 172.16.1.1[500]-172.16.2.1[500] cookie:5d45c72ba1dacb47:0f089edf1e9edb3e.
IKE phase-1 SA is expired SA: 172.16.1.1[500]-172.16.2.1[500] cookie:5d45c72ba1dacb47:0f089edf1e9edb3e.
IKE phase-1 SA is expired SA: 172.16.1.1[500]-172.16.2.1[500] cookie:5d45c72ba1dacb47:0f089edf1e9edb3e.
IKE protocol phase-1 SA delete message received from peer. cookie:5d45c72ba1dacb47:0f089edf1e9edb3e.
IKE protocol notification message received: INVALID-ID-INFORMATION (18).
IKE phase-2 negotiation is started as initiator, quick mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500] message id:0x011F2B1C.

IKE phase-1 negotiation is succeeded as initiator, main mode. Established SA: 172.16.1.1[500]-172.16.2.1[500]
cookie:5d45c72ba1dacb47:0f089edf1e9edb3e lifetime 28800 Sec.
IKE phase-1 negotiation is started as initiator, main mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500]
cookie:5d45c72ba1dacb47:0000000000000000.
```

Figure 23. Proxy-ID mismatch 02: PA-200 as initiator



```
IP = 172.16.1.1, Received encrypted packet with no matching SA, dropping
Group = 172.16.1.1, Username = 172.16.1.1, IP = 172.16.1.1, Session disconnected. Session Type: LAN-to-LAN, Duration: 0h:00m:00s, Bytes xmt: 0, Bytes rcv: 0, Reason: crypto map policy not found
Group = 172.16.1.1, IP = 172.16.1.1, Session is being torn down. Reason: crypto map policy not found
Group = 172.16.1.1, IP = 172.16.1.1, Removing peer from correlator table failed, no match!
Group = 172.16.1.1, IP = 172.16.1.1, QM FSM error (P2 struct &0xcc290d60, mess id 0x59893274)!
Group = 172.16.1.1, IP = 172.16.1.1, Rejecting IPSec tunnel: no matching crypto map entry for remote proxy 10.0.1.0/255.255.255.0/0/0 local proxy 10.0.2.0/255.255.255.0/0/0 on interface untrust
Group = 172.16.1.1, IP = 172.16.1.1, PHASE 1 COMPLETED
AAA retrieved default group policy (DfltGrpPolicy) for user = 172.16.1.1
Built inbound UDP connection 312 for untrust:172.16.1.1/500 (172.16.1.1/500) to identity:172.16.2.1/500 (172.16.2.1/500)
```

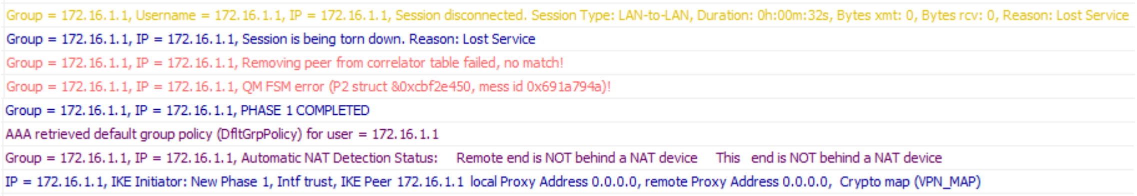
Figure 24. Proxy-ID mismatch 02: ASA 5505 as receiver

Figures 23 and 24 reflect essentially the same output as Figures 21 and 19; the receiver (ASA) rejects the initiator’s (PA) Proxy-ID proposal and just sends a notification that the contents of the Identification payloads were no good.

In the third scenario the ACL on ASA’s side was switched to one allowing all traffic to the tunnel. Configuration changes on ASA were as follows:

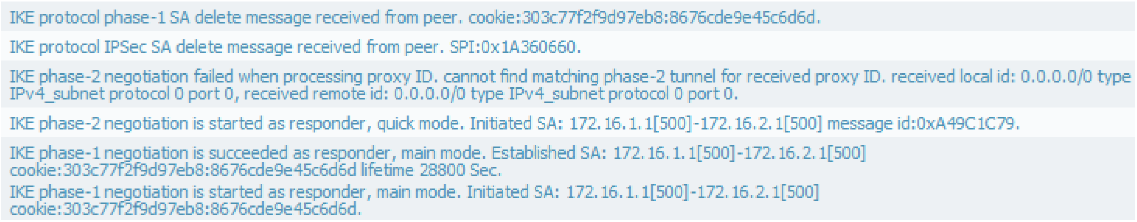
```
access-list FALSE-ACL extended permit ip any any
crypto map VPN_MAP 1 match address FALSE-ACL
```

Having ASA initiate the negotiations will cause negotiations to break down with slight delay. Figures 25 and 26 show how PA will not accept ASA’s proposal for Proxy-IDs, and how ASA after determining that negotiations do not progress terminates the IPsec negotiations and the IKE SA, as no IPsec SAs were successfully established.



```
Group = 172.16.1.1, Username = 172.16.1.1, IP = 172.16.1.1, Session disconnected. Session Type: LAN-to-LAN, Duration: 0h:00m:32s, Bytes xmt: 0, Bytes rcv: 0, Reason: Lost Service
Group = 172.16.1.1, IP = 172.16.1.1, Session is being torn down. Reason: Lost Service
Group = 172.16.1.1, IP = 172.16.1.1, Removing peer from correlator table failed, no match!
Group = 172.16.1.1, IP = 172.16.1.1, QM FSM error (P2 struct &0xcbf2e450, mess id 0x691a794a)!
Group = 172.16.1.1, IP = 172.16.1.1, PHASE 1 COMPLETED
AAA retrieved default group policy (DfltGrpPolicy) for user = 172.16.1.1
Group = 172.16.1.1, IP = 172.16.1.1, Automatic NAT Detection Status:     Remote end is NOT behind a NAT device     This   end is NOT behind a NAT device
IP = 172.16.1.1, IKE Initiator: New Phase 1, Intf trust, IKE Peer 172.16.1.1  local Proxy Address 0.0.0.0, remote Proxy Address 0.0.0.0,  Crypto map (VPN_MAP)
```

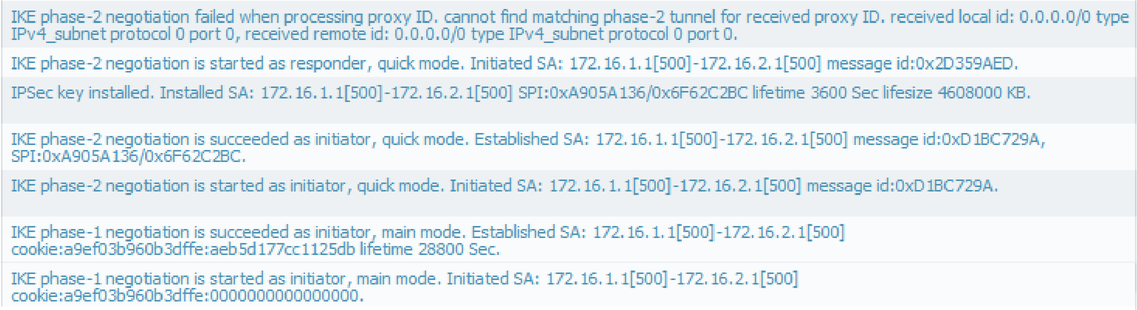
Figure 25. Proxy-ID mismatch 03: ASA 5505 as initiator



```
IKE protocol phase-1 SA delete message received from peer. cookie:303c77f2f9d97eb8:8676cde9e45c6d6d.
IKE protocol IPSec SA delete message received from peer. SPI:0x1A360660.
IKE phase-2 negotiation failed when processing proxy ID. cannot find matching phase-2 tunnel for received proxy ID. received local id: 0.0.0.0/0 type
IPv4_subnet protocol 0 port 0, received remote id: 0.0.0.0/0 type IPv4_subnet protocol 0 port 0.
IKE phase-2 negotiation is started as responder, quick mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500] message id:0xA49C1C79.
IKE phase-1 negotiation is succeeded as responder, main mode. Established SA: 172.16.1.1[500]-172.16.2.1[500]
cookie:303c77f2f9d97eb8:8676cde9e45c6d6d lifetime 28800 Sec.
IKE phase-1 negotiation is started as responder, main mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500]
cookie:303c77f2f9d97eb8:8676cde9e45c6d6d.
```

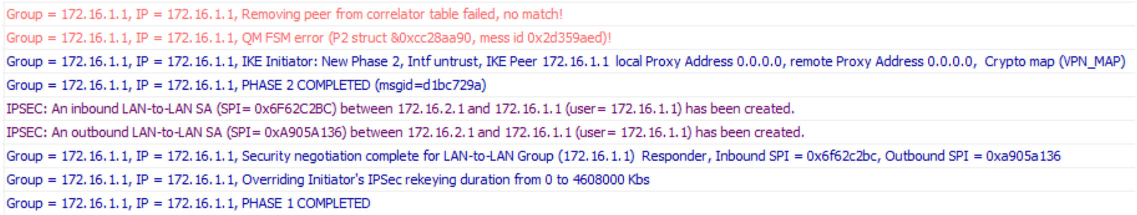
Figure 26. Proxy-ID mismatch 03: PA-200 as receiver

Figures 27 and 28 below show PA-200 assuming the initiator's role in turn.



```
IKE phase-2 negotiation failed when processing proxy ID. cannot find matching phase-2 tunnel for received proxy ID. received local id: 0.0.0.0/0 type
IPv4_subnet protocol 0 port 0, received remote id: 0.0.0.0/0 type IPv4_subnet protocol 0 port 0.
IKE phase-2 negotiation is started as responder, quick mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500] message id:0x2D359AED.
IPSec key installed. Installed SA: 172.16.1.1[500]-172.16.2.1[500] SPI:0xA905A136/0x6F62C2BC lifetime 3600 Sec lifesize 4608000 KB.
IKE phase-2 negotiation is succeeded as initiator, quick mode. Established SA: 172.16.1.1[500]-172.16.2.1[500] message id:0xD1BC729A,
SPI:0xA905A136/0x6F62C2BC.
IKE phase-2 negotiation is started as initiator, quick mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500] message id:0xD1BC729A.
IKE phase-1 negotiation is succeeded as initiator, main mode. Established SA: 172.16.1.1[500]-172.16.2.1[500]
cookie:a9ef03b960b3dffe:aeb5d177cc1125db lifetime 28800 Sec.
IKE phase-1 negotiation is started as initiator, main mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500]
cookie:a9ef03b960b3dffe:0000000000000000.
```

Figure 27. Proxy-ID mismatch 03: PA-200 as initiator



```
Group = 172.16.1.1, IP = 172.16.1.1, Removing peer from correlator table failed, no match!
Group = 172.16.1.1, IP = 172.16.1.1, QM FSM error (P2 struct &0xcc28aa90, mess id 0x2d359aed)!
Group = 172.16.1.1, IP = 172.16.1.1, IKE Initiator: New Phase 2, Intf untrust, IKE Peer 172.16.1.1  local Proxy Address 0.0.0.0, remote Proxy Address 0.0.0.0,  Crypto map (VPN_MAP)
Group = 172.16.1.1, IP = 172.16.1.1, PHASE 2 COMPLETED (msgid=d1bc729a)
IPSEC: An inbound LAN-to-LAN SA (SPI= 0x6F62C2BC) between 172.16.2.1 and 172.16.1.1 (user= 172.16.1.1) has been created.
IPSEC: An outbound LAN-to-LAN SA (SPI= 0xA905A136) between 172.16.2.1 and 172.16.1.1 (user= 172.16.1.1) has been created.
Group = 172.16.1.1, IP = 172.16.1.1, Security negotiation complete for LAN-to-LAN Group (172.16.1.1)  Responder, Inbound SPI = 0x6f62c2bc, Outbound SPI = 0xa905a136
Group = 172.16.1.1, IP = 172.16.1.1, Overriding Initiator's IPSec rekeying duration from 0 to 4608000 Kbs
Group = 172.16.1.1, IP = 172.16.1.1, PHASE 1 COMPLETED
```

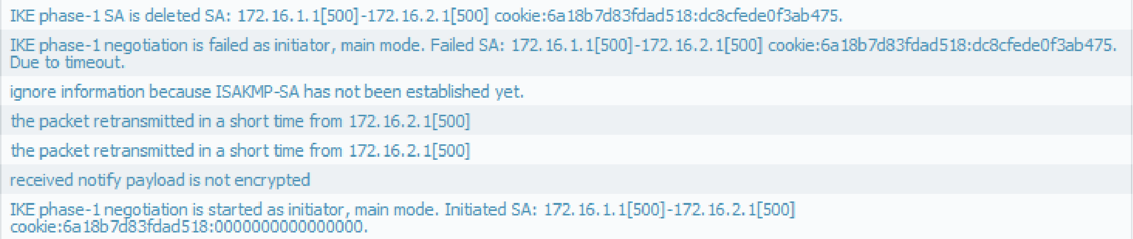
Figure 28. Proxy-ID mismatch 03: ASA 5505 as receiver

As a receiver ASA agrees to establish the IPsec SA with the proposed Proxy-IDs partially matching its own. Soon enough ASA realizes that its crypto map ACL does not have matching IPsec SAs even though an IKE SA has been established. Thus ASA continues offering its 0.0.0.0/0 network only to get rejected by PA time and time again. ASA then continues to attempt creating a new IPsec SA pair.

## 6.2 CASE 02: Pre-Shared Key (PSK) Mismatch

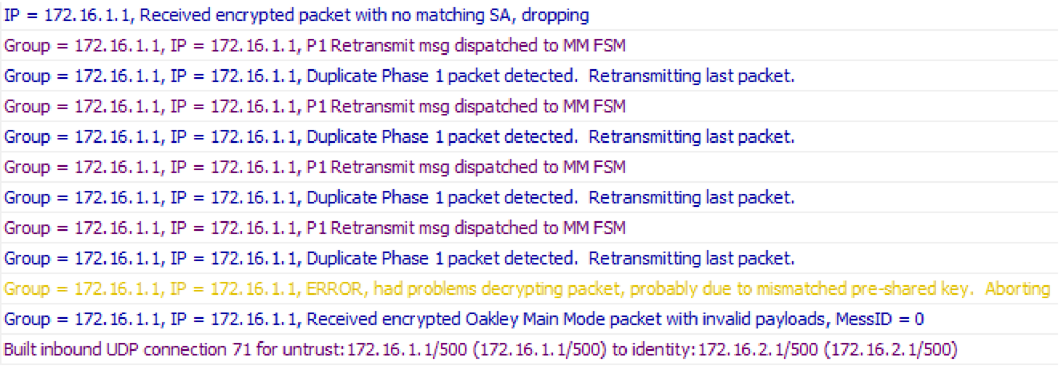
PSK mismatch during IKE Phase 1 occurs when the receiver is unable to process the MM message 5, which directly follows the DH key exchange. The Receiver’s inability to process the packet will prompt it to resend the previous MM message 4. As a result The Initiator will receive a duplicate message and end up resending MM message 5 and will expect MM message 6 in return.

To start off, Figures 29 and 30 below play out the scenario when PA initiates the negotiations:



```
IKE phase-1 SA is deleted SA: 172.16.1.1[500]-172.16.2.1[500] cookie:6a18b7d83fdad518:dc8cfede0f3ab475.
IKE phase-1 negotiation is failed as initiator, main mode. Failed SA: 172.16.1.1[500]-172.16.2.1[500] cookie:6a18b7d83fdad518:dc8cfede0f3ab475.
Due to timeout.
ignore information because ISAKMP-SA has not been established yet.
the packet retransmitted in a short time from 172.16.2.1[500]
the packet retransmitted in a short time from 172.16.2.1[500]
received notify payload is not encrypted
IKE phase-1 negotiation is started as initiator, main mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500]
cookie:6a18b7d83fdad518:0000000000000000.
```

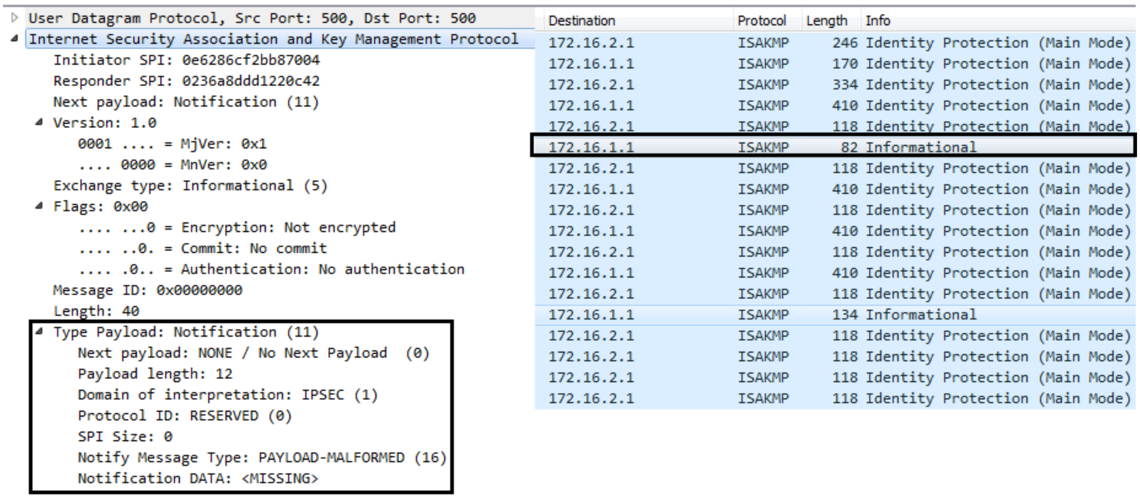
Figure 29. PSK mismatch: PA-200 as initiator



```
IP = 172.16.1.1, Received encrypted packet with no matching SA, dropping
Group = 172.16.1.1, IP = 172.16.1.1, P1 Retransmit msg dispatched to MM FSM
Group = 172.16.1.1, IP = 172.16.1.1, Duplicate Phase 1 packet detected.  Retransmitting last packet.
Group = 172.16.1.1, IP = 172.16.1.1, P1 Retransmit msg dispatched to MM FSM
Group = 172.16.1.1, IP = 172.16.1.1, Duplicate Phase 1 packet detected.  Retransmitting last packet.
Group = 172.16.1.1, IP = 172.16.1.1, P1 Retransmit msg dispatched to MM FSM
Group = 172.16.1.1, IP = 172.16.1.1, Duplicate Phase 1 packet detected.  Retransmitting last packet.
Group = 172.16.1.1, IP = 172.16.1.1, P1 Retransmit msg dispatched to MM FSM
Group = 172.16.1.1, IP = 172.16.1.1, Duplicate Phase 1 packet detected.  Retransmitting last packet.
Group = 172.16.1.1, IP = 172.16.1.1, ERROR, had problems decrypting packet, probably due to mismatched pre-shared key.  Aborting
Group = 172.16.1.1, IP = 172.16.1.1, Received encrypted Oakley Main Mode packet with invalid payloads, MessID = 0
Built inbound UDP connection 71 for untrust:172.16.1.1/500 (172.16.1.1/500) to identity:172.16.2.1/500 (172.16.2.1/500)
```

Figure 30. PSK mismatch: ASA 5505 as receiver

As seen in Figure 29, PA-200 ends up hanging and eventually time-outing the negotiations. ASA 5505, being on the receiving end, realizes an error processing the encrypted packet and even suspects a mismatched PSK as proven by Figure 30. This deduction is based on the decryption resulting in invalid payloads. ASA sends this information to PA as a separate message with Notification payload of type ‘PAYLOAD-MALFORMED’. PA’s debugging does not indicate to trigger any actions on PA’s side regarding this piece of information. Such details can be spotted by running Packet Capture on PA filtered by the peer addresses. Figure 31 shows the Packet Capture output.

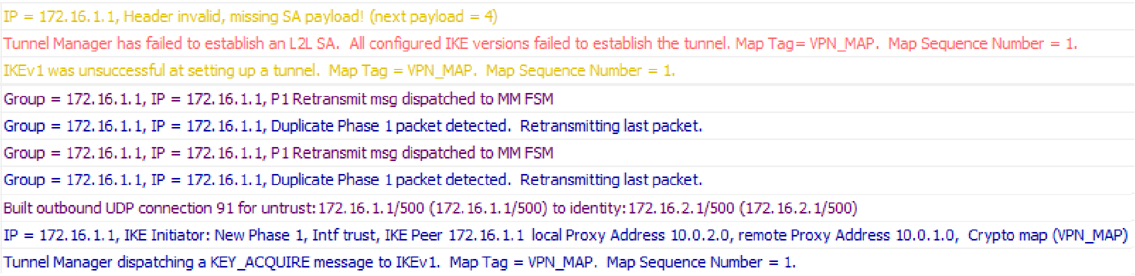


```
▷ User Datagram Protocol, Src Port: 500, Dst Port: 500
◢ Internet Security Association and Key Management Protocol
    Initiator SPI: 0e6286cf2bb87004
    Responder SPI: 0236a8ddd1220c42
    Next payload: Notification (11)
  ◢ Version: 1.0
      0001 .... = MjVer: 0x1
      .... 0000 = MnVer: 0x0
    Exchange type: Informational (5)
  ◢ Flags: 0x00
      .... ...0 = Encryption: Not encrypted
      .... ..0. = Commit: No commit
      .... .0.. = Authentication: No authentication
    Message ID: 0x00000000
    Length: 40
  ◢ Type Payload: Notification (11)
      Next payload: NONE / No Next Payload  (0)
      Payload length: 12
      Domain of interpretation: IPSEC (1)
      Protocol ID: RESERVED (0)
      SPI Size: 0
      Notify Message Type: PAYLOAD-MALFORMED (16)
      Notification DATA: <MISSING>
```

```
Destination      Protocol  Length Info
172.16.2.1       ISAKMP     246 Identity Protection (Main Mode)
172.16.1.1       ISAKMP     170 Identity Protection (Main Mode)
172.16.2.1       ISAKMP     334 Identity Protection (Main Mode)
172.16.1.1       ISAKMP     410 Identity Protection (Main Mode)
172.16.2.1       ISAKMP     118 Identity Protection (Main Mode)
172.16.1.1       ISAKMP      82 Informational
172.16.2.1       ISAKMP     118 Identity Protection (Main Mode)
172.16.1.1       ISAKMP     410 Identity Protection (Main Mode)
172.16.2.1       ISAKMP     118 Identity Protection (Main Mode)
172.16.1.1       ISAKMP     410 Identity Protection (Main Mode)
172.16.2.1       ISAKMP     118 Identity Protection (Main Mode)
172.16.1.1       ISAKMP     410 Identity Protection (Main Mode)
172.16.2.1       ISAKMP     118 Identity Protection (Main Mode)
172.16.1.1       ISAKMP     134 Informational
172.16.2.1       ISAKMP     118 Identity Protection (Main Mode)
172.16.2.1       ISAKMP     118 Identity Protection (Main Mode)
172.16.2.1       ISAKMP     118 Identity Protection (Main Mode)
172.16.2.1       ISAKMP     118 Identity Protection (Main Mode)
```

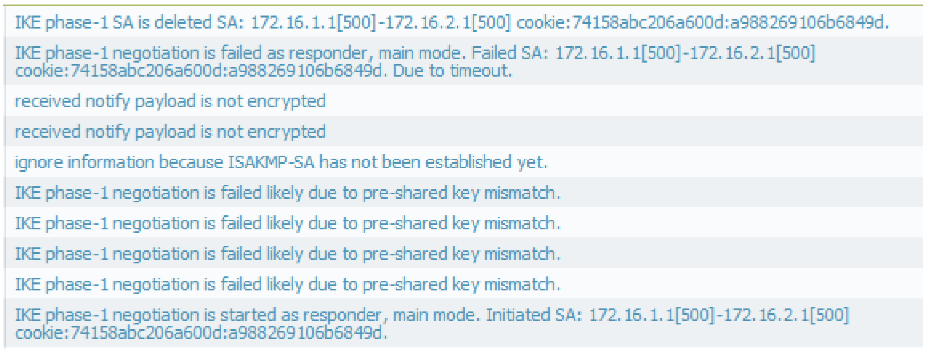
Figure 31. PSK mismatch: PA-200’s Packet Capture as initiator

Reverting the negotiator roles, the outcome appears as follows:



```
IP = 172.16.1.1, Header invalid, missing SA payload! (next payload = 4)
Tunnel Manager has failed to establish an L2L SA.  All configured IKE versions failed to establish the tunnel. Map Tag= VPN_MAP.  Map Sequence Number = 1.
IKEv1 was unsuccessful at setting up a tunnel.  Map Tag = VPN_MAP.  Map Sequence Number = 1.
Group = 172.16.1.1, IP = 172.16.1.1, P1 Retransmit msg dispatched to MM FSM
Group = 172.16.1.1, IP = 172.16.1.1, Duplicate Phase 1 packet detected.  Retransmitting last packet.
Group = 172.16.1.1, IP = 172.16.1.1, P1 Retransmit msg dispatched to MM FSM
Group = 172.16.1.1, IP = 172.16.1.1, Duplicate Phase 1 packet detected.  Retransmitting last packet.
Built outbound UDP connection 91 for untrust:172.16.1.1/500 (172.16.1.1/500) to identity:172.16.2.1/500 (172.16.2.1/500)
IP = 172.16.1.1, IKE Initiator: New Phase 1, Intf trust, IKE Peer 172.16.1.1  local Proxy Address 10.0.2.0, remote Proxy Address 10.0.1.0,  Crypto map (VPN_MAP)
Tunnel Manager dispatching a KEY_ACQUIRE message to IKEv1.  Map Tag = VPN_MAP.  Map Sequence Number = 1.
```

Figure 32. PSK mismatch: ASA 5505 as initiator



```
IKE phase-1 SA is deleted SA: 172.16.1.1[500]-172.16.2.1[500] cookie:74158abc206a600d:a988269106b6849d.
IKE phase-1 negotiation is failed as responder, main mode. Failed SA: 172.16.1.1[500]-172.16.2.1[500]
cookie:74158abc206a600d:a988269106b6849d. Due to timeout.
received notify payload is not encrypted
received notify payload is not encrypted
ignore information because ISAKMP-SA has not been established yet.
IKE phase-1 negotiation is failed likely due to pre-shared key mismatch.
IKE phase-1 negotiation is failed likely due to pre-shared key mismatch.
IKE phase-1 negotiation is failed likely due to pre-shared key mismatch.
IKE phase-1 negotiation is failed likely due to pre-shared key mismatch.
IKE phase-1 negotiation is started as responder, main mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500]
cookie:74158abc206a600d:a988269106b6849d.
```

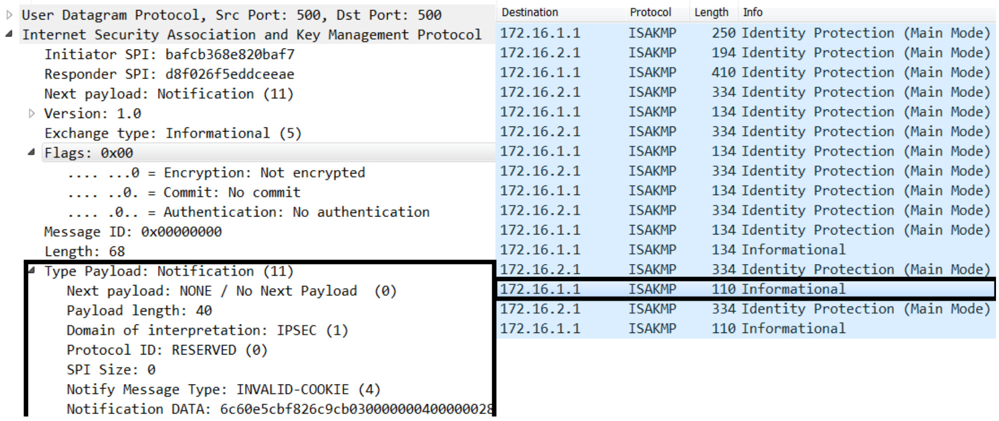
Figure 33. PSK mismatch: PA-200 as receiver

Figures 32 and 33 show that as in the former case, the initiator (ASA) cannot determine the issue by itself, while the receiver (PA) suspects a PSK mismatch, which is indeed correct. The MM messages 5–6 are encrypted, so with decrypting them with incorrect key(s), the payloads will end up invalid. However, PA withholds the information and the MM messages 4–5 continue to be retransmitted. ASA as the initiator soon ends the loop and informs PA via an encrypted message carrying a Delete payload specifying the SPIs – or better known as initiator and responder cookies during Phase 1. Listing 9 presents how the error loop occurs.

```
[IKEv1 DEBUG] IP = 172.16.1.1, IKE MM Initiator FSM error history
(struct &0xcbf3f4c0)  <state>, <event>:  MM_DONE, EV_ERROR-->MM_WAIT_MSG6,  EV_PROB_AUTH_FAIL-->MM_WAIT_MSG6,  EV_TIMEOUT-->MM_WAIT_MSG6, NullEvent-->MM_SND_MSG5, EV_SND_MSG-->MM_SND_MSG5, EV_START_TMR-->MM_SND_MSG5,       EV_RESEND_MSG-->MM_WAIT_MSG6, EV_RESEND_MSG
[IKEv1  DEBUG]Group  =  172.16.1.1,  IP  =  172.16.1.1,  IKE  SA MM:ef8f5050 terminating:  flags 0x01000022, refcnt 0, tuncnt 0
[IKEv1 DEBUG]Group = 172.16.1.1, IP = 172.16.1.1,
sending delete/delete with reason message
```

Listing 9. ASA stuck at ‘MM_WAIT_MSG6’ stage.

Upon receiving and decrypting the message, PA chooses to ignore it, since the SA has not been established yet, continuing to resend MM message 4. ASA, having already deleted the SAD entry for this incomplete ISAKMP (IKE) SA, now sends PA a notification of type ‘INVALID-COOKIE’ since the cookie (SPI) pair identifying the SA that was being negotiated no longer exist in its database. Running Packet Capture on PA again reveals how this exchange looks like as printed on Figure 34.



```
▷ User Datagram Protocol, Src Port: 500, Dst Port: 500
◢ Internet Security Association and Key Management Protocol
    Initiator SPI: bafcb368e820baf7
    Responder SPI: d8f026f5eddceeae
    Next payload: Notification (11)
  ▷ Version: 1.0
    Exchange type: Informational (5)
  ◢ Flags: 0x00
      .... ...0 = Encryption: Not encrypted
      .... ..0. = Commit: No commit
      .... .0.. = Authentication: No authentication
    Message ID: 0x00000000
    Length: 68
  ◢ Type Payload: Notification (11)
      Next payload: NONE / No Next Payload  (0)
      Payload length: 40
      Domain of interpretation: IPSEC (1)
      Protocol ID: RESERVED (0)
      SPI Size: 0
      Notify Message Type: INVALID-COOKIE (4)
      Notification DATA: 6c60e5cbf826c9cb030000000400000028
```

| Destination | Protocol | Length | Info |
|---|---|---|---|
| 172.16.1.1 | ISAKMP | 250 | Identity Protection (Main Mode) |
| 172.16.2.1 | ISAKMP | 194 | Identity Protection (Main Mode) |
| 172.16.1.1 | ISAKMP | 410 | Identity Protection (Main Mode) |
| 172.16.2.1 | ISAKMP | 334 | Identity Protection (Main Mode) |
| 172.16.1.1 | ISAKMP | 134 | Identity Protection (Main Mode) |
| 172.16.2.1 | ISAKMP | 334 | Identity Protection (Main Mode) |
| 172.16.1.1 | ISAKMP | 134 | Identity Protection (Main Mode) |
| 172.16.2.1 | ISAKMP | 334 | Identity Protection (Main Mode) |
| 172.16.1.1 | ISAKMP | 134 | Identity Protection (Main Mode) |
| 172.16.2.1 | ISAKMP | 334 | Identity Protection (Main Mode) |
| 172.16.1.1 | ISAKMP | 134 | Identity Protection (Main Mode) |
| 172.16.1.1 | ISAKMP | 134 | Informational |
| 172.16.2.1 | ISAKMP | 334 | Identity Protection (Main Mode) |
| 172.16.1.1 | ISAKMP | 110 | Informational |
| 172.16.2.1 | ISAKMP | 334 | Identity Protection (Main Mode) |
| 172.16.1.1 | ISAKMP | 110 | Informational |

Figure 34. PSK mismatch: PA-200’s Packet Capture as receiver

PA gets to receive two of such notifications before its negotiations timeout. It becomes apparent that PA’s support for ISAKMP notifications is at a lower level than ASA.

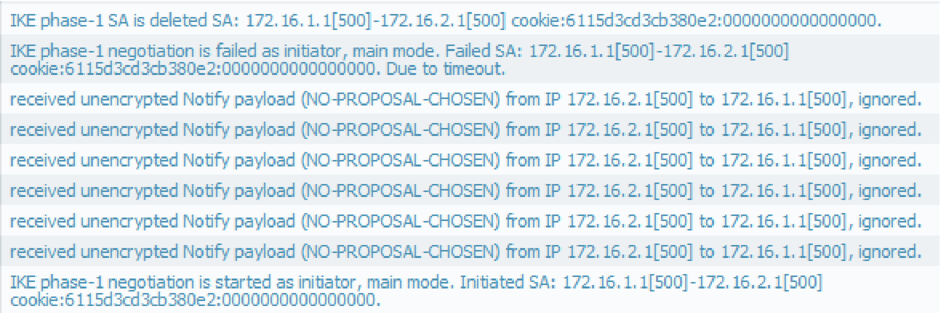
## 6.3 CASE 03: IKE Policy Set Mismatch

IKE policy negotiation takes place during MM messages 1–2 and is used to agree on a policy set comprising an encryption algorithm and key length, hash algorithm, DH group, authentication method and key lifetime. Mismatches in the encryption algorithm, hash algorithm, DH group and authentication method result in clear error messages, whereas lifetime is more flexible and may thus result in difficult-to-spot misbehaviours when a session reaches the rekeying stage.

In order to view IKE policy proposals as the receiver for the CLI debugging we will use debug level on PA and 254 level on ASA as follows:

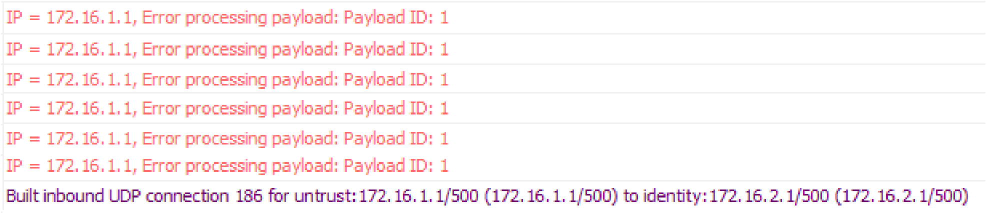
```
PA-200> debug ike global on debug
ASA-5505# debug crypto ikev1 254
```

In the experiments the IKE policy parameters were changed on PA-200 one by one and the output recorded; AES128 was changed to 3DES, SHA-1 to MD5 and DH5 to DH14. Encryption and hash algorithm mismatches had the exact same error messages, so these will be covered together. This will be followed by the DH group mismatch with a more distinct reaction. Figures 35 and 36 picture the IKEv1 encryption and/or hash mismatch scenario when PA-200 initiates the negotiations.



```
IKE phase-1 SA is deleted SA: 172.16.1.1[500]-172.16.2.1[500] cookie:6115d3cd3cb380e2:0000000000000000.
IKE phase-1 negotiation is failed as initiator, main mode. Failed SA: 172.16.1.1[500]-172.16.2.1[500]
cookie:6115d3cd3cb380e2:0000000000000000. Due to timeout.
received unencrypted Notify payload (NO-PROPOSAL-CHOSEN) from IP 172.16.2.1[500] to 172.16.1.1[500], ignored.
received unencrypted Notify payload (NO-PROPOSAL-CHOSEN) from IP 172.16.2.1[500] to 172.16.1.1[500], ignored.
received unencrypted Notify payload (NO-PROPOSAL-CHOSEN) from IP 172.16.2.1[500] to 172.16.1.1[500], ignored.
received unencrypted Notify payload (NO-PROPOSAL-CHOSEN) from IP 172.16.2.1[500] to 172.16.1.1[500], ignored.
received unencrypted Notify payload (NO-PROPOSAL-CHOSEN) from IP 172.16.2.1[500] to 172.16.1.1[500], ignored.
received unencrypted Notify payload (NO-PROPOSAL-CHOSEN) from IP 172.16.2.1[500] to 172.16.1.1[500], ignored.
IKE phase-1 negotiation is started as initiator, main mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500]
cookie:6115d3cd3cb380e2:0000000000000000.
```

Figure 35. IKEv1 policy mismatch (encryption or hash algorithm): PA-200 as initiator



```
IP = 172.16.1.1, Error processing payload: Payload ID: 1
IP = 172.16.1.1, Error processing payload: Payload ID: 1
IP = 172.16.1.1, Error processing payload: Payload ID: 1
IP = 172.16.1.1, Error processing payload: Payload ID: 1
IP = 172.16.1.1, Error processing payload: Payload ID: 1
IP = 172.16.1.1, Error processing payload: Payload ID: 1
Built inbound UDP connection 186 for untrust:172.16.1.1/500 (172.16.1.1/500) to identity:172.16.2.1/500 (172.16.2.1/500)
```

Figure 36. IKEv1 policy mismatch (encryption or hash algorithm): ASA 5505 as receiver

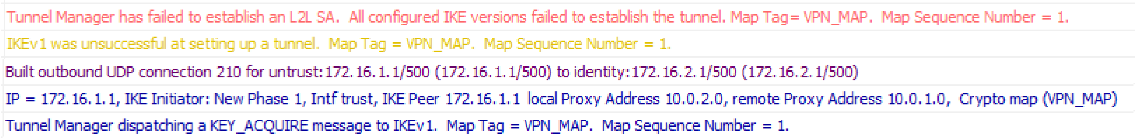
On ASA’s CLI debugging this is indicated as shown in Listing 10 below.

```
[IKEv1]IP = 172.16.1.1, IKE_DECODE SENDING Message (msgid=0) with
payloads : HDR + NOTIFY (11) + NONE (0) total length : 92
[IKEv1 DEBUG]IP = 172.16.1.1, All SA proposals found unacceptable
[IKEv1]IP = 172.16.1.1, Error processing payload: Payload ID: 1
[IKEv1 DEBUG]IP = 172.16.1.1, IKE MM Responder FSM error history
(struct &0xcbf2e110)  <state>, <event>:  MM_DONE, EV_ERROR--
>MM_START, EV_RCV_MSG-->MM_START, EV_START_MM-->MM_START, …
[IKEv1 DEBUG]IP = 172.16.1.1, IKE SA MM:00710c6c terminating:
flags 0x01000002, refcnt 0, tuncnt 0
[IKEv1 DEBUG]IP = 172.16.1.1, sending delete/delete with reason
message
```

Listing 10. ASA rejecting all IKEv1 proposals.

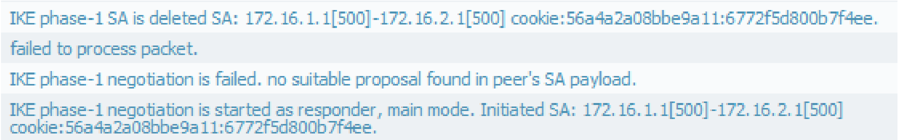
Here we may observe that PA’s proposal is rejected and the notification ‘NO-PROPOSAL-CHOSEN’ is send. PA chooses to ignore this and simply resend the proposal until negotiations timeout.

Swapping the roles we get an output equivalent to Figures 37 and 38.



```
Tunnel Manager has failed to establish an L2L SA.  All configured IKE versions failed to establish the tunnel. Map Tag= VPN_MAP.  Map Sequence Number = 1.
IKEv1 was unsuccessful at setting up a tunnel.  Map Tag = VPN_MAP.  Map Sequence Number = 1.
Built outbound UDP connection 210 for untrust:172.16.1.1/500 (172.16.1.1/500) to identity:172.16.2.1/500 (172.16.2.1/500)
IP = 172.16.1.1, IKE Initiator: New Phase 1, Intf trust, IKE Peer 172.16.1.1  local Proxy Address 10.0.2.0, remote Proxy Address 10.0.1.0,  Crypto map (VPN_MAP)
Tunnel Manager dispatching a KEY_ACQUIRE message to IKEv1.  Map Tag = VPN_MAP.  Map Sequence Number = 1.
```

Figure 37. IKEv1 policy mismatch (encryption or hash algorithm): ASA 5505 as initiator



```
IKE phase-1 SA is deleted SA: 172.16.1.1[500]-172.16.2.1[500] cookie:56a4a2a08bbe9a11:6772f5d800b7f4ee.
failed to process packet.
IKE phase-1 negotiation is failed. no suitable proposal found in peer's SA payload.
IKE phase-1 negotiation is started as responder, main mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500]
cookie:56a4a2a08bbe9a11:6772f5d800b7f4ee.
```

Figure 38. IKEv1 policy mismatch (encryption or hash algorithm): PA-200 as receiver

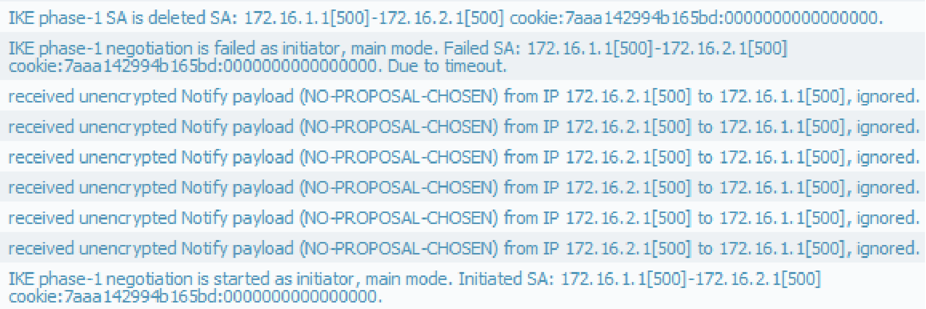
The scarce amount of output data can be explained by looking at PA’s CLI output as shown in Listing 11 below.

```
[PROTO_ERR]:    IPsec_doi.c:514:print_ph1mismatched():    rejected
enctype: DB(prop#1:trns#1):Peer(prop#1:trns#1) = 3DES:AES
[PROTO_ERR]:   IPsec_doi.c:278:get_ph1approval():   no   suitable
proposal found
[PROTO_ERR]:    isakmp_ident.c:1030:ident_r1recv():    0:?    -
172.16.2.1[500]:(nil):failed to get valid proposal.
[PROTO_ERR]:  ikev1.c:1415:isakmp_ph1begin_r():  failed  to  process
packet.
[INFO]:  ikev1.c:2485:log_ph1deleted():  ====>  PHASE-1  SA  DELETED
<====
```

Listing 11. PA rejecting all IKE proposals.

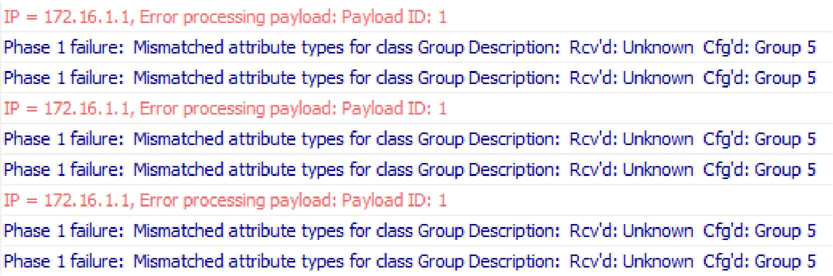
While ASA did send a notification message, PA did no such thing. PA compares the policy sets until the last one, determines the reason for rejection and ends the negotiations there while ASA retries until timeout.

Next comes the DH group mismatch scenario. Having PA to initiate the negotiations, the output shown in Figures 39 and 40 will be generated.



IKE phase-1 SA is deleted SA: 172.16.1.1[500]-172.16.2.1[500] cookie:7aaa142994b165bd:0000000000000000.
IKE phase-1 negotiation is failed as initiator, main mode. Failed SA: 172.16.1.1[500]-172.16.2.1[500] cookie:7aaa142994b165bd:0000000000000000. Due to timeout.
received unencrypted Notify payload (NO-PROPOSAL-CHOSEN) from IP 172.16.2.1[500] to 172.16.1.1[500], ignored.
received unencrypted Notify payload (NO-PROPOSAL-CHOSEN) from IP 172.16.2.1[500] to 172.16.1.1[500], ignored.
received unencrypted Notify payload (NO-PROPOSAL-CHOSEN) from IP 172.16.2.1[500] to 172.16.1.1[500], ignored.
received unencrypted Notify payload (NO-PROPOSAL-CHOSEN) from IP 172.16.2.1[500] to 172.16.1.1[500], ignored.
received unencrypted Notify payload (NO-PROPOSAL-CHOSEN) from IP 172.16.2.1[500] to 172.16.1.1[500], ignored.
received unencrypted Notify payload (NO-PROPOSAL-CHOSEN) from IP 172.16.2.1[500] to 172.16.1.1[500], ignored.
IKE phase-1 negotiation is started as initiator, main mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500] cookie:7aaa142994b165bd:0000000000000000.

Figure 39. IKEv1 policy mismatch (DH group): PA-200 as initiator



IP = 172.16.1.1, Error processing payload: Payload ID: 1
Phase 1 failure: Mismatched attribute types for class Group Description: Rcv'd: Unknown Cfg'd: Group 5
Phase 1 failure: Mismatched attribute types for class Group Description: Rcv'd: Unknown Cfg'd: Group 5
IP = 172.16.1.1, Error processing payload: Payload ID: 1
Phase 1 failure: Mismatched attribute types for class Group Description: Rcv'd: Unknown Cfg'd: Group 5
Phase 1 failure: Mismatched attribute types for class Group Description: Rcv'd: Unknown Cfg'd: Group 5
IP = 172.16.1.1, Error processing payload: Payload ID: 1
Phase 1 failure: Mismatched attribute types for class Group Description: Rcv'd: Unknown Cfg'd: Group 5
Phase 1 failure: Mismatched attribute types for class Group Description: Rcv'd: Unknown Cfg'd: Group 5

Figure 40. IKEv1 policy mismatch (DH group): ASA 5505 as receiver

The error messages and behavior are about the same for DH group mismatch as in the previous scenario of encryption or hash algorithm mismatch. Only notable difference is that ASA reports the DH group mismatch. Apparently for this software version DH group 14 and above appears as ‘Unknown’ whereas group 2 would appear properly. PA receives ASA’s mismatch notifications like seen in Figure 39.

In the case of ASA initiating the negotiations, PA would react in the same manner as with encryption or hash mismatch; realizing DH group mismatch has occurred and ending the negotiations without a word.

Finally we have the IKEv1 SA lifetime mismatch. This will be broken down into the following four scenarios:

1. PA initiates negotiations with a higher lifetime value (86400s > 28800s)
2. ASA initiates negotiations with a lower lifetime value (28800s < 86400s)
3. PA initiates negotiations with a lower lifetime value (7200s < 28800s)
4. ASA initiates negotiations with a higher lifetime value (28800s > 7200s).

In scenario 1 when PA proposes 86400 second lifetime value, ASA replies without changing the lifetime value in the policy set. However, ASA has chosen to utilize its lower lifetime of 28800 seconds. ASA will convey this information to PA at the end of MM negotiations via a notification message ‘RESPONDER-LIFETIME’. The parties will continue with mismatched IKE SA lifetimes until ASA prompts a rekeying process.

In scenario 2 when ASA proposes 28800 second lifetime value, PA agrees to the lower key lifetime value over its higher 86400 second lifetime. ASA receives confirmation of this in the MM message 2.

In scenario 3 when PA proposes 7200 second lifetime value, ASA will agree on the proposed lower lifetime value over its higher 28800 second lifetime. No changes occur to the following message exchanges.

In scenario 4 when ASA proposes 28800 second lifetime value, PA replies without changing the lifetime value in the policy set. However, PA has chosen to utilize its lower lifetime of 7200 seconds. PA abstains from sending the notification message to ASA. The parties will continue with mismatched IKE SA lifetimes until PA prompts a rekeying process.

IKE SA lifetime mismatch thus does not generate blatant warning or error messages, at most notification messages informing the peer. In the end vendors are given leeway in deciding how the peers will react to these lifetime mismatch situations. [12, 22–23].

## 6.4 CASE 04: IPsec Policy Set Mismatch

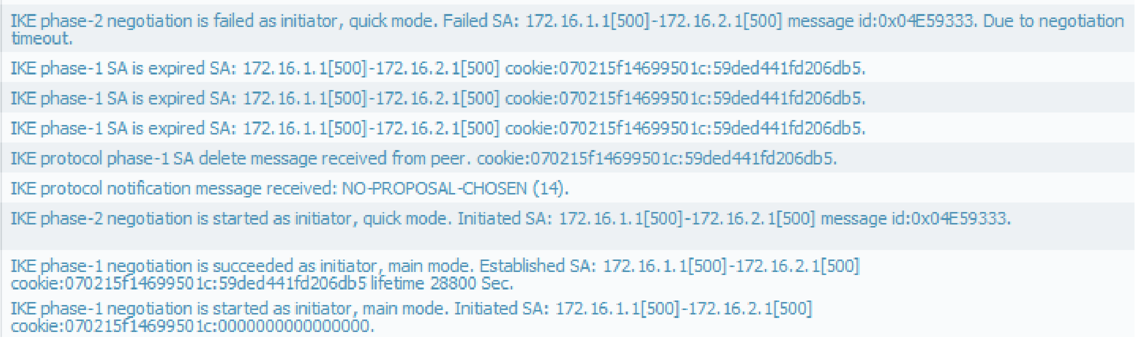
The IPsec policy negotiation takes place during QM messages 1–2 and is used to agree on a policy set comprising an encapsulation protocol, encryption algorithm and key length, hash algorithm, key lifetime in seconds and key lifesize in kilobytes. An optional DH group is also negotiated if PFS is enabled. Tunnelling mode will be the solely used encryption mode. Very much like during IKE policy set negotiation, IPsec policy set mismatches in encapsulation protocol, encryption algorithm, hash algorithm or DH group (PFS) result in clear error messages, whereas lifetime is more flexible.

In order to view IPsec policy proposals as the receiver for the CLI debugging we will use debug level on PA and 254 level on ASA as follows:

```
PA-200> debug ike global on debug
ASA-5505# debug crypto ikev1 254
ASA-5505# debug crypto ipsec 254
```

In the experiments the IPsec policy parameters were changed on PA-200 one by one and the output recorded; AES128 was changed to AES256, SHA-1 to MD5, PFS group to DH2 when enabled and ESP encapsulation to AH. The more varied IPsec SA lifetime talks will be covered at the end of the section.

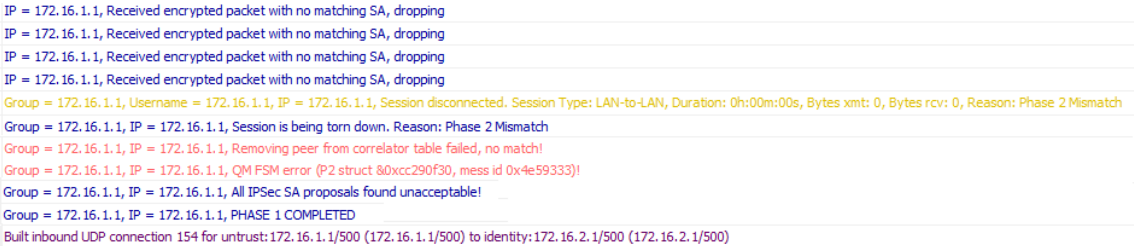
As with during IKE policy negotiations, IPsec encryption and hash algorithm mismatches had the exact same error messages, so these will be covered together. This will be followed by the DH group (PFS) mismatch and encapsulation protocol mismatch. Figures 41 and 42 depict the IPsec encryption and/or hash mismatch scenario when PA-200 initiates the negotiations.



```
IKE phase-2 negotiation is failed as initiator, quick mode. Failed SA: 172.16.1.1[500]-172.16.2.1[500] message id:0x04E59333. Due to negotiation timeout.
IKE phase-1 SA is expired SA: 172.16.1.1[500]-172.16.2.1[500] cookie:070215f14699501c:59ded441fd206db5.
IKE phase-1 SA is expired SA: 172.16.1.1[500]-172.16.2.1[500] cookie:070215f14699501c:59ded441fd206db5.
IKE phase-1 SA is expired SA: 172.16.1.1[500]-172.16.2.1[500] cookie:070215f14699501c:59ded441fd206db5.
IKE protocol phase-1 SA delete message received from peer. cookie:070215f14699501c:59ded441fd206db5.
IKE protocol notification message received: NO-PROPOSAL-CHOSEN (14).
IKE phase-2 negotiation is started as initiator, quick mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500] message id:0x04E59333.

IKE phase-1 negotiation is succeeded as initiator, main mode. Established SA: 172.16.1.1[500]-172.16.2.1[500]
cookie:070215f14699501c:59ded441fd206db5 lifetime 28800 Sec.
IKE phase-1 negotiation is started as initiator, main mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500]
cookie:070215f14699501c:0000000000000000.
```

Figure 41. IPsec policy mismatch (encryption or hash algorithm): PA-200 as initiator

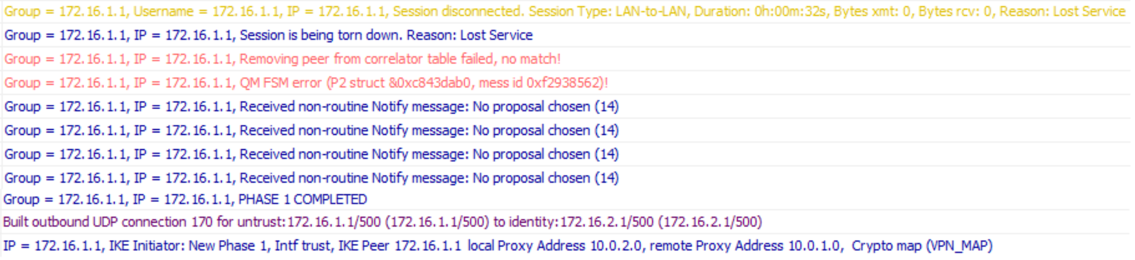


```
IP = 172.16.1.1, Received encrypted packet with no matching SA, dropping
IP = 172.16.1.1, Received encrypted packet with no matching SA, dropping
IP = 172.16.1.1, Received encrypted packet with no matching SA, dropping
IP = 172.16.1.1, Received encrypted packet with no matching SA, dropping
Group = 172.16.1.1, Username = 172.16.1.1, IP = 172.16.1.1, Session disconnected. Session Type: LAN-to-LAN, Duration: 0h:00m:00s, Bytes xmt: 0, Bytes rcv: 0, Reason: Phase 2 Mismatch
Group = 172.16.1.1, IP = 172.16.1.1, Session is being torn down. Reason: Phase 2 Mismatch
Group = 172.16.1.1, IP = 172.16.1.1, Removing peer from correlator table failed, no match!
Group = 172.16.1.1, IP = 172.16.1.1, QM FSM error (P2 struct &0xcc290f30, mess id 0x4e59333)!
Group = 172.16.1.1, IP = 172.16.1.1, All IPSec SA proposals found unacceptable!
Group = 172.16.1.1, IP = 172.16.1.1, PHASE 1 COMPLETED
Built inbound UDP connection 154 for untrust:172.16.1.1/500 (172.16.1.1/500) to identity:172.16.2.1/500 (172.16.2.1/500)
```

Figure 42. IPsec policy mismatch (encryption or hash algorithm): ASA 5505 as receiver

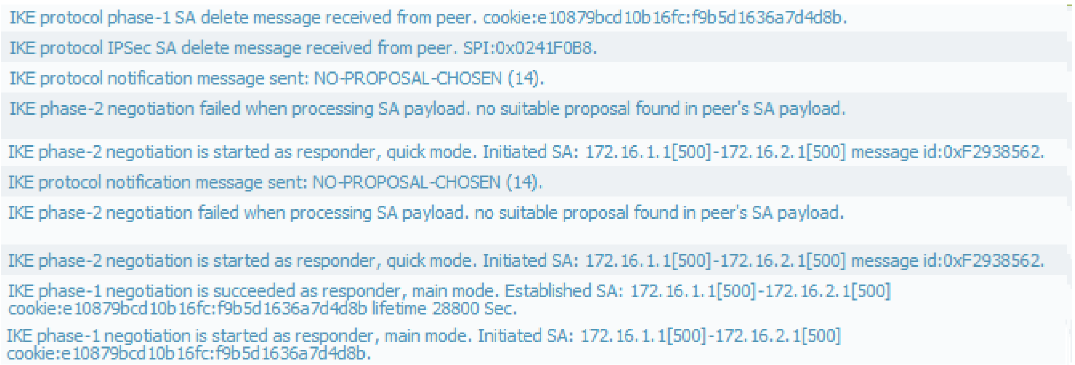
Having received PA's IPsec policy set proposal, ASA rejects the mismatched set and sends PA the notification message 'NO-PROPOSAL-CHOSEN'. ASA then informs PA of IKE SA deletion. PA still continues to offer its IPsec proposal until negotiation timeout. Meanwhile ASA drops the following proposals which no longer have an associated SA.

For the reverse interaction, Figures 43 and 44 reflect the occurring negotiations.



```
Group = 172.16.1.1, Username = 172.16.1.1, IP = 172.16.1.1, Session disconnected. Session Type: LAN-to-LAN, Duration: 0h:00m:32s, Bytes xmt: 0, Bytes rcv: 0, Reason: Lost Service
Group = 172.16.1.1, IP = 172.16.1.1, Session is being torn down. Reason: Lost Service
Group = 172.16.1.1, IP = 172.16.1.1, Removing peer from correlator table failed, no match!
Group = 172.16.1.1, IP = 172.16.1.1, QM FSM error (P2 struct &0xc843dab0, mess id 0xf2938562)!
Group = 172.16.1.1, IP = 172.16.1.1, Received non-routine Notify message: No proposal chosen (14)
Group = 172.16.1.1, IP = 172.16.1.1, Received non-routine Notify message: No proposal chosen (14)
Group = 172.16.1.1, IP = 172.16.1.1, Received non-routine Notify message: No proposal chosen (14)
Group = 172.16.1.1, IP = 172.16.1.1, Received non-routine Notify message: No proposal chosen (14)
Group = 172.16.1.1, IP = 172.16.1.1, PHASE 1 COMPLETED
Built outbound UDP connection 170 for untrust:172.16.1.1/500 (172.16.1.1/500) to identity:172.16.2.1/500 (172.16.2.1/500)
IP = 172.16.1.1, IKE Initiator: New Phase 1, Intf trust, IKE Peer 172.16.1.1  local Proxy Address 10.0.2.0, remote Proxy Address 10.0.1.0,  Crypto map (VPN_MAP)
```

Figure 43. IPsec policy mismatch (encryption or hash algorithm): ASA 5505 as initiator



```
IKE protocol phase-1 SA delete message received from peer. cookie:e10879bcd10b16fc:f9b5d1636a7d4d8b.
IKE protocol IPSec SA delete message received from peer. SPI:0x0241F0B8.
IKE protocol notification message sent: NO-PROPOSAL-CHOSEN (14).
IKE phase-2 negotiation failed when processing SA payload. no suitable proposal found in peer's SA payload.
IKE phase-2 negotiation is started as responder, quick mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500] message id:0xF2938562.
IKE protocol notification message sent: NO-PROPOSAL-CHOSEN (14).
IKE phase-2 negotiation failed when processing SA payload. no suitable proposal found in peer's SA payload.
IKE phase-2 negotiation is started as responder, quick mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500] message id:0xF2938562.
IKE phase-1 negotiation is succeeded as responder, main mode. Established SA: 172.16.1.1[500]-172.16.2.1[500]
cookie:e10879bcd10b16fc:f9b5d1636a7d4d8b lifetime 28800 Sec.
IKE phase-1 negotiation is started as responder, main mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500]
cookie:e10879bcd10b16fc:f9b5d1636a7d4d8b.
```

Figure 44. IPsec policy mismatch (encryption or hash algorithm): PA-200 as receiver

Just as above, once ASA's IPsec proposal gets rejected by PA, PA sends the notification message 'NO-PROPOSAL-CHOSEN'. Listing 12 shows PA's CLI output.

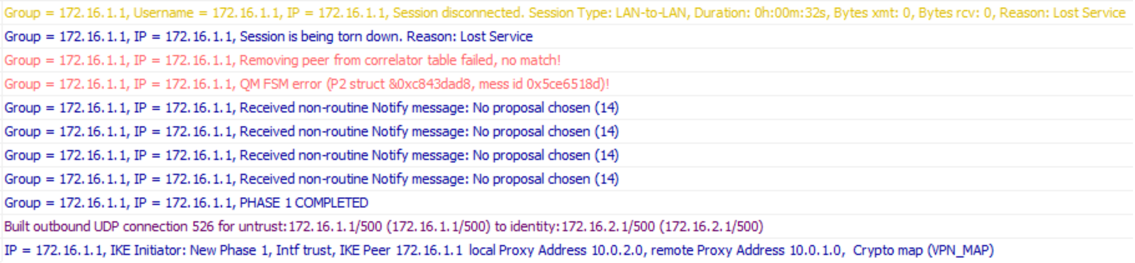
```
[DEBUG]:   IPsec_doi.c:1172:get_ph2approvalx():   peer's   single
bundle:
[DEBUG]: proposal.c:1057:printsaproto():  (proto_id=ESP spisize=4
spi=de172888 spi_p=00000000 encmode=Tunnel reqid=0:0)
[DEBUG]: proposal.c:1091:printsatrns():  (trns_id=AES enclen=128
authtype=hmac-sha)
[DEBUG]: IPsec_doi.c:1175:get_ph2approvalx(): my single bundle:
[DEBUG]: proposal.c:1057:printsaproto():  (proto_id=ESP spisize=4
spi=00000000 spi_p=00000000 encmode=Tunnel reqid=0:0)
[DEBUG]: proposal.c:1091:printsatrns():  (trns_id=AES enclen=128
authtype=hmac-md5)
[DEBUG]: proposal.c:564:cmpsatrns(): authtype mismatched: my:hmac-
md5 peer:hmac-sha
[PROTO_ERR]: IPsec_doi.c:1183:get_ph2approvalx(): not matched
[PROTO_ERR]:  IPsec_doi.c:1146:get_ph2approval():  no  suitable
policy found.
[INTERNAL_ERR]: ikev1.c:1621:isakmp_ph2begin_r(): failed to pre-
process packet.
```

Listing 12. PA rejecting all IPsec proposals.

Having compared ASA's proposal to its own, PA compiles the notification message to be sent to ASA. Each consecutive time ASA's proposal arrives, it performs the same IPsec policy comparison and replies with the same notification message until ASA's negotiations timeout and notification for IPSEC and IKE SA deletion arrives.

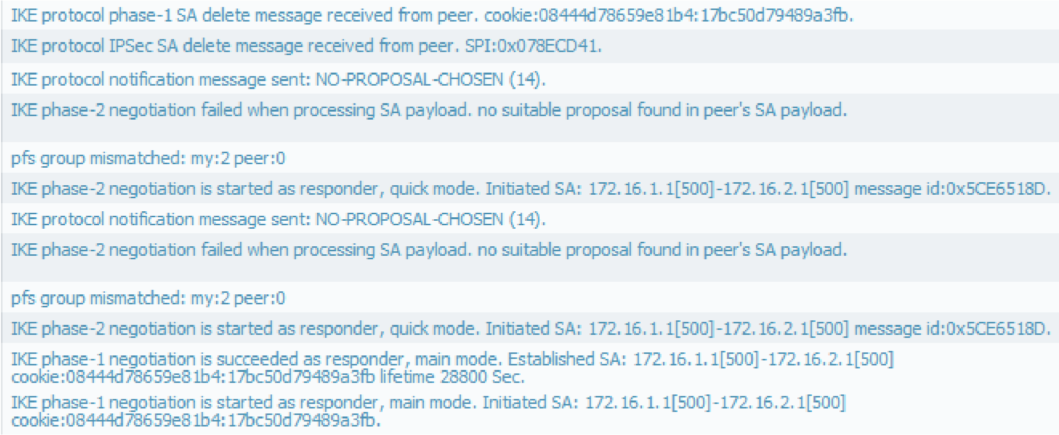
When PFS is enabled on PA, but disabled on ASA, the System Log output will look exactly the same when PA is in the initiator's role as if the mismatch's cause was encryption or hash algorithm. This occurs since ASA does not have a PFS group configured to compare the proposal's group to. When PA, who has PFS enabled, is in the receiving end, the PFS group mismatch is apparent on the GUI's System Log alone.

PA’s PFS has been set to DH group 2 while ASA does not have it enabled, which shows up as DH group 0. Figures 45 and 46 portray this exchange scenario.



```
Group = 172.16.1.1, Username = 172.16.1.1, IP = 172.16.1.1, Session disconnected. Session Type: LAN-to-LAN, Duration: 0h:00m:32s, Bytes xmt: 0, Bytes rcv: 0, Reason: Lost Service
Group = 172.16.1.1, IP = 172.16.1.1, Session is being torn down. Reason: Lost Service
Group = 172.16.1.1, IP = 172.16.1.1, Removing peer from correlator table failed, no match!
Group = 172.16.1.1, IP = 172.16.1.1, QM FSM error (P2 struct &0xc843dad8, mess id 0x5ce6518d)!
Group = 172.16.1.1, IP = 172.16.1.1, Received non-routine Notify message: No proposal chosen (14)
Group = 172.16.1.1, IP = 172.16.1.1, Received non-routine Notify message: No proposal chosen (14)
Group = 172.16.1.1, IP = 172.16.1.1, Received non-routine Notify message: No proposal chosen (14)
Group = 172.16.1.1, IP = 172.16.1.1, Received non-routine Notify message: No proposal chosen (14)
Group = 172.16.1.1, IP = 172.16.1.1, PHASE 1 COMPLETED
Built outbound UDP connection 526 for untrust:172.16.1.1/500 (172.16.1.1/500) to identity:172.16.2.1/500 (172.16.2.1/500)
IP = 172.16.1.1, IKE Initiator: New Phase 1, Intf trust, IKE Peer 172.16.1.1  local Proxy Address 10.0.2.0, remote Proxy Address 10.0.1.0,  Crypto map (VPN_MAP)
```

Figure 45. IPsec policy mismatch (PFS): ASA 5505 as initiator



```
IKE protocol phase-1 SA delete message received from peer. cookie:08444d78659e81b4:17bc50d79489a3fb.
IKE protocol IPSec SA delete message received from peer. SPI:0x078ECD41.
IKE protocol notification message sent: NO-PROPOSAL-CHOSEN (14).
IKE phase-2 negotiation failed when processing SA payload. no suitable proposal found in peer's SA payload.
pfs group mismatched: my:2 peer:0
IKE phase-2 negotiation is started as responder, quick mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500] message id:0x5CE6518D.
IKE protocol notification message sent: NO-PROPOSAL-CHOSEN (14).
IKE phase-2 negotiation failed when processing SA payload. no suitable proposal found in peer's SA payload.
pfs group mismatched: my:2 peer:0
IKE phase-2 negotiation is started as responder, quick mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500] message id:0x5CE6518D.
IKE phase-1 negotiation is succeeded as responder, main mode. Established SA: 172.16.1.1[500]-172.16.2.1[500]
cookie:08444d78659e81b4:17bc50d79489a3fb lifetime 28800 Sec.
IKE phase-1 negotiation is started as responder, main mode. Initiated SA: 172.16.1.1[500]-172.16.2.1[500]
cookie:08444d78659e81b4:17bc50d79489a3fb.
```

Figure 46. IPsec policy mismatch (PFS): PA-200 as receiver

In the event that PA proposes the usage of AH encapsulation to ASA who uses ESP – or vice versa, the system log output will not change; same ‘NO-PROPOSAL-CHOSEN’ notification message is sent once again. The CLI debug output in the receiver’s point of view changes slightly for both parties. ASA’s reaction is as shown in Listing 13.

```
[IKEv1 DEBUG]Group = 172.16.1.1, IP = 172.16.1.1,
processing IPsec SA payload
[IKEv1 DEBUG]Group = 172.16.1.1, IP = 172.16.1.1,
AH proposal not supported
[IKEv1]Group = 172.16.1.1, IP = 172.16.1.1,
All IPsec SA proposals found unacceptable!
[IKEv1 DEBUG]Group = 172.16.1.1, IP = 172.16.1.1,
sending notification message
[IKEv1 DEBUG]Group = 172.16.1.1, IP = 172.16.1.1,
sending delete/delete with reason message
```

Listing 13. ASA encounters an encapsulation mismatch.

When ASA in turn initiates the negotiations, PA’s debug output is as shown in Listing 14 below.

```
[DEBUG]:   IPsec_doi.c:1172:get_ph2approvalx():   peer's   single
bundle:
[DEBUG]: proposal.c:1057:printsaproto():  (proto_id=ESP spisize=4
spi=c1c4d87d spi_p=00000000 encmode=Tunnel reqid=0:0)
[DEBUG]: proposal.c:1091:printsatrns():  (trns_id=AES encklen=128
authtype=hmac-sha)
[DEBUG]: IPsec_doi.c:1175:get_ph2approvalx(): my single bundle:
[DEBUG]: proposal.c:1057:printsaproto():  (proto_id=AH spisize=4
spi=00000000 spi_p=00000000 encmode=Tunnel reqid=0:0)
[DEBUG]:   proposal.c:1085:printsatrns():          (trns_id=SHA1
authtype=hmac-sha)
[PROTO_ERR]: IPsec_doi.c:1183:get_ph2approvalx(): not matched
[PROTO_ERR]:  IPsec_doi.c:1146:get_ph2approval():  no  suitable
policy found.
[INTERNAL_ERR]: ikev1.c:1621:isakmp_ph2begin_r(): failed to pre-
process packet.
<output omitted>
[DEBUG]: isakmp_inf.c:807:isakmp_info_send_common():
sendto Information notify.
<output omitted>
[INFO]: isakmp_inf.c:1437:isakmp_info_recv_d():
IKE IPSEC KEY_DELETE recvd:  SPI:0xC1C4D87D.
```

Listing 14. PA encounters an encapsulation mismatch.

From PA’s CLI debugging output we can also note that AH indeed does not exercise encryption as this field is filled with its hash algorithm (SHA1) instead. Interestingly enough ASA has been confirmed to be the more aggressive peer with regards to SA terminations be it as initiator or receiver.

Last but not least comes IPsec SA lifetime and lifesize mismatch. PA’s lifesize has been left at the default zero value, which practically makes it unlimited, thus making ASA’s specified lifesize value being preferred. Like with IKEv1 SA lifetime troubleshooting, this will be broken down into the following four scenarios:

1. PA initiates negotiations with a higher lifetime value (7200s > 3600s)
2. ASA initiates negotiations with a lower lifetime value (3600s < 7200s)
3. PA initiates negotiations with a lower lifetime value (1800s < 3600s)
4. ASA initiates negotiations with a higher lifetime value (3600s > 1800s).

In scenario 1 when PA proposes 7200 second lifetime value, ASA returns the proposal and adds a notification payload specifying that it has chosen the lower lifetime value of 3600 seconds and lifesize of 4608000 KB. PA updates its IPsec SA lifetime and lifesize to the preferred values in the notification payload according to ASA’s notification as shown in Listing 15 below.

```
[IKEv1]Group = 172.16.1.1, IP = 172.16.1.1,
Overriding Initiator's IPsec rekeying duration from 7200 to 3600
seconds
[IKEv1]Group = 172.16.1.1, IP = 172.16.1.1,
Overriding Initiator's IPsec rekeying duration from 0 to 4608000
Kbs
[IKEv1 DEBUG]Group = 172.16.1.1, IP = 172.16.1.1,
Sending RESPONDER LIFETIME notification to Initiator
```

Listing 15. ASA forcing its lower IPsec SA lifetime and lifesize onto PA.

In scenario 2 when ASA proposes 3600 second lifetime value, PA agrees to the lower SA lifetime proposal and the preferred lifesize of 4608000 KB.

In scenario 3 when PA proposes 1800 second lifetime value, ASA agrees to the lower SA lifetime proposal, but notifies PA of choosing its preferred lifesize as shown in Listing 16 below.

```
[IKEv1]Group = 172.16.1.1, IP = 172.16.1.1,
Overriding Initiator's IPsec rekeying duration from 0 to 4608000
Kbs
[IKEv1 DEBUG]Group = 172.16.1.1, IP = 172.16.1.1,
Sending RESPONDER LIFETIME notification to Initiator
```

Listing 16. ASA choosing its preferred IPsec SA lifesize.

In scenario 4 when ASA proposes 3600 second lifetime value, PA decides to keep its lower lifetime of 1800 seconds and includes a notification payload in its response. PA also unconditionally agrees to the proposed IPsec SA lifesize of 4608000 KB. ASA in turn chooses to update its lifetime value to PA’s lower 1800 seconds as shown in Listing 17 below.

```
[IKEv1 DECODE] Responder Lifetime decode follows (outb
SPI[4]|attributes):
[IKEv1 DECODE] 0000: B9823F3F 80010001 00020004 00000708
[IKEv1]Group = 172.16.1.1, IP = 172.16.1.1, Responder forcing
change of IPsec rekeying duration from 3600 to 1800 seconds
```

Listing 17. PA forcing its lower IPsec SA lifetime onto ASA.

In conclusion, IPsec SA lifetime negotiations between these two peers will always agree on the lower lifetime value regardless of the negotiation role. While PA’s debug output shows that ASA’s notification arrives, it does not explicitly state whether the notification is ignored or agreed upon until the lifetime is updated at the end of the negotiations.

### 6.5 CASE 05: IKE Version Mismatch

While the theoretical part of this study does not cover IKEv2 protocol in detail, introducing a scenario where one peer speaks IKEv1 and another IKEv2 is definitely worthwhile to find out how it actually affects the negotiations. To implement the configurations for IKEv2, we will use Cisco’s IKE migration feature, which requires an existing IPsec VPN configuration for IKEv1 with PSK authentication and will create a matching configuration for IKEv2. Issuing the CLI command `migrate l2l` on ASA will result in the following configuration changes shown in Listing 18.

```
crypto IPsec ikev2 IPsec-proposal ESP-AES1-SHA
 protocol esp encryption aes
 protocol esp integrity sha-1
!
crypto map VPN_MAP 1 set ikev2 IPsec-proposal ESP-AES1-SHA
!
crypto ikev2 policy 10
 encryption aes
 integrity sha
 group 5
 prf sha
 lifetime seconds 28800
crypto ikev2 enable untrust
!
tunnel-group 172.16.1.1 IPsec-attributes
 ikev2 remote-authentication pre-shared-key *****
 ikev2 local-authentication pre-shared-key *****
```

Listing 18. ASA migrating from IKEv1 to IKEv2.

IKEv2 now functions parallel to IKEv1 sharing the same transform set and crypto map. IKEv2 will automatically use the same PSK as was configured for IKEv1. In order to debug IKEv2 the following additional CLI debugging commands are introduced:

```
debug crypto ike-common 5
debug crypto ikev2 platform 4
debug crypto ikev2 protocol 4
```

The command `debug crypto ike-common` will print general tunnel management output for both IKEv1 and IKEv2, thus making it useful to determine which protocol is used to build a tunnel. The commands `debug crypto ikev2 platform` and `de-`

`bug crypto ikev2 protocol` generate messages for IKEv2 negotiation exchanges and protocol’s processing. The debugging level can be set within the range 1–255. A lower debugging level is used here to just debug the version mismatch scenarios.

For the first scenario, we will temporarily disable IKEv1 forcing ASA to only negotiate with IKEv2. To disable IKEv1 negotiations on the peer-facing interface, we issue:

```
    no crypto ikev1 enable untrust
```

Initiating the negotiations from PA’s side will only prompt connection building and teardown on the informational syslog level. PA will keep on sending its IKEv1 proposal until the SA negotiation times out. ASA’s IKEv1 CLI debugging prints the output shown in Listing 19.

```
[IKEv1]IKE Receiver: Discarding IKEv1 packet, disabled on
<untrust>
[IKEv1]IKE Receiver: Packet received on 172.16.2.1:500 from
172.16.1.1:500
```

Listing 19. ASA rejects unsupported IKEv1 packet.

After reversing the roles and initiating the negotiations from ASA, PA generates output matching Listing 20.

```
[IKE COMMON DEBUG]Tunnel Manager dispatching a KEY_ACQUIRE
message to IKEv2.  Map Tag = VPN_MAP.  Map Sequence Number = 1.
IKEv2-PLAT-2: mapped to tunnel group 172.16.1.1 using peer IP
IKEv2-PROTO-2: (1): Sending Packet
[To 172.16.1.1:500/From 172.16.2.1:500/VRF i0:f0]
IKEv2-PLAT-3: (1): SENT PKT [IKE_SA_INIT]
[172.16.2.1]:500->[172.16.1.1]:500      InitSPI=0x53c7d1aa6c4267bd
RespSPI=0x0000000000000000 MID=00000000
<output omitted>
IKEv2-PROTO-1: (1): Maximum number of retransmissions reached
IKEv2-PROTO-2: (1): Deleting SA
IKEv2-PLAT-1: Failed to remove peer correlation entry from
cikePeerCorrTable.  Local Type = 0.  Local Address = 0.0.0.0.
Remote Type = 0.  Remote Address = 0.0.0.0.  Correlation Peer
Index = 0. IPSEC Tunnel Index = 0.
[IKE COMMON DEBUG]IKEv2 was unsuccessful at setting up a tunnel.
Map Tag = VPN_MAP.  Map Sequence Number = 1.
[IKE COMMON DEBUG]Tunnel Manager has failed to establish an L2L
SA.  All configured IKE versions failed to establish the tunnel.
Map Tag= VPN_MAP.  Map Sequence Number = 1.
[IKE COMMON DEBUG]Tunnel Manager Removed entry.
Map Tag = VPN_MAP.  Map Sequence Number = 1.
```

Listing 20. ASA’s IKEv2 proposals are rejected until timeout.

On PA’s end this generates the output shown in Listing 21.

```
[DEBUG]: isakmp.c:1043:isakmp_handler():
521 bytes message received from 172.16.2.1[500]
[PROTO_WARN]: ikev2.c:262:ikev2_input():
0:172.16.1.1[500] - 172.16.2.1[500]:0x103cc268:unknown ikev2 peer
```

Listing 21. PA rejects unsupported IKEv2 proposals.

So, in both cases the IKE proposals are silently discarded, and the initiating peer will continue offering their proposal until negotiation timeout.

For the next scenario IKEv1 will be re-enabled on ASA to support both versions in parallel. When ASA begins the IKE SA negotiations it begins with the preferred IKEv2, but after timing out it will proceed to attempt IKEv1 negotiations, which then succeed. PA will as before discard the IKEv2 proposals and only reply when IKEv1 proposals are received. Listing 22 shows the common IKE exchange occurring on ASA’s CLI.

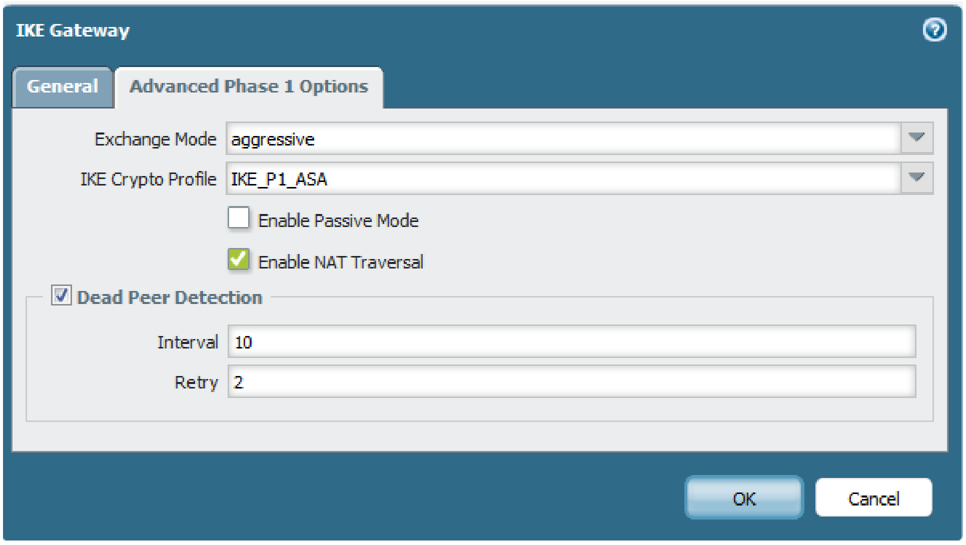
```
[IKE COMMON DEBUG]Tunnel Manager dispatching a KEY_ACQUIRE
message to IKEv2.  Map Tag = VPN_MAP.  Map Sequence Number = 1.
<output omitted>
[IKE COMMON DEBUG]IKEv2 was unsuccessful at setting up a tunnel.
Map Tag = VPN_MAP.  Map Sequence Number = 1.
[IKE COMMON DEBUG]Tunnel Manager dispatching a KEY_ACQUIRE
message to IKEv1 after a failed attempt..  Map Tag = VPN_MAP.
Map Sequence Number = 1.
<output omitted>
[IKE COMMON DEBUG]IKEv1 was successful at setting up a tunnel.
Map Tag = VPN_MAP. Map Sequence Number = 1.
```

Listing 22. ASA falls back to IKEv1 after failed IKEv2 negotiations.

For reference, it took 7 proposals and 2 minutes for IKEv2 to timeout before IKEv1 was given the baton. Should PA initiate the negotiations, ASA would automatically revert to IKEv1 as soon as the first proposal comes, since the ISAKMP (IKE) header comes with a Version field.

## 6.6 CASE 06: IKE Mode Mismatch

IKE mode or exchange type is determined by the initiator in its first message. The receiver will then either agree or disagree on the proposed exchange type. By default both ASA 5505 and PA-200 will use auto mode, which will initiate negotiations using MM, but will agree to both MM and AM proposals. That being the case, the only scenario to be covered is having PA configured for AM and ASA for MM. Figure 47 shows PA-200 being configured for AM.


IKE Gateway

Exchange Mode: aggressive
IKE Crypto Profile: IKE_P1_ASA
☐ Enable Passive Mode
☑ Enable NAT Traversal
☑ Dead Peer Detection
Interval: 10
Retry: 2

Figure 47. Reconfiguring PA-200 IKE Gateway for AM.

Now, on the ASA’s CLI we can disable AM and only allow MM with the command:

```
crypto ikev1 am-disable
```

When ASA receives PA’s IKE policy proposal, it only needs to look at the ISAKMP header’s Exchange Type field to determine the following action:

```
[IKEv1]IP = 172.16.1.1, Aggressive Mode connections disabled on
interface ... dropping pkt
```

When PA receives ASA’s IKE policy proposal, the reaction is just as anticlimactic:

```
[PROTO_ERR]: ikev1.c:1341:isakmp_ph1begin_r(): IKE phase-1 request
for gateway VPN_Peer_ASA is rejected: main mode is not allowed by
configuration.
```

In both cases the receiver promptly discards the proposals. In these scenarios the initiator will not be able to figure out why the negotiations fail.

### 6.7 CASE 07: NAT Traversal (NAT-T)

NAT during transit introduces more design changes to the IPsec VPN configuration than NAT-T alone can fix. It will also require changes to peer IP address and Local and Remote Identifiers depending on which kind of address translation takes place. Once NAT is confirmed to exist in transit, we are concerned about whether it is a static or dynamic NAT. Any peer behind a dynamic NAT should only take on the role of an initiator, since the other peer would not be able to reach the peer behind dynamic NAT when the NAT entry is not in place. [13, 6–7].

While any NAT/PAT device in transit is known to modify the IP addresses and/or TCP/UDP ports, yet another consideration is that IPsec traffic may end up dropped altogether. The issue lies in ESP header not specifying port numbers. Some NAT/PAT devices discard ESP encapsulated traffic on sight, since address translation cannot be performed. [13, 7].

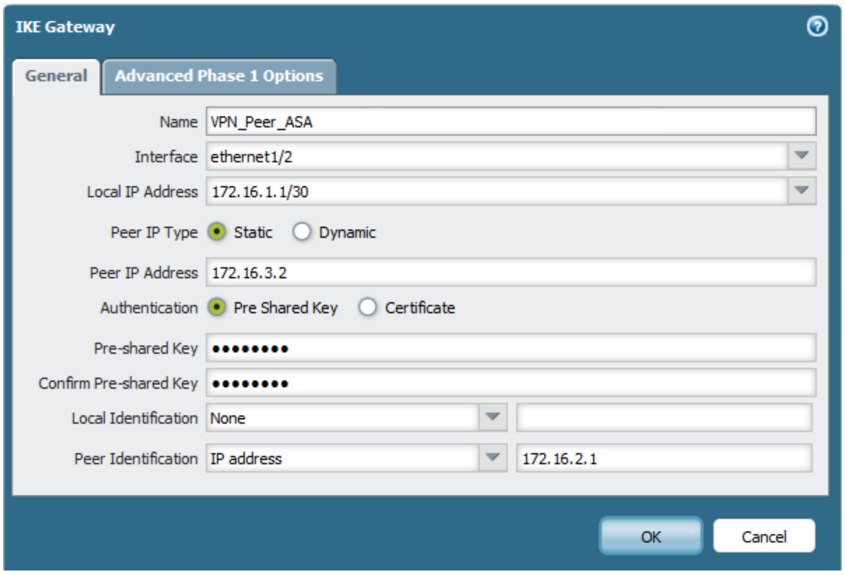
In this study we will enforce NAT on the third-party router such that ASA will be inside NAT. The router configuration is as shown in Listing 23.

```
interface GigabitEthernet0/0
 description LINK_to_PA200
 ip address 172.16.1.2 255.255.255.252
 ip nat outside
!
interface GigabitEthernet0/1
 description LINK_to_ASA5505
 ip address 172.16.2.2 255.255.255.252
 ip nat inside
!
interface Loopback0
 ip address 172.16.3.1 255.255.255.0
!
ip nat inside source static 172.16.2.1 172.16.3.2
```

Listing 23. Third-party router’s interface and NAT configurations.

With such a static NAT the NAT-designated inside local address 172.16.2.1 will be known as the global address 172.16.3.2 behind the NAT-designated outside interface and vice versa. The single NAT rule will create bi-directional address translations. While addresses solely used for NAT, such as the 172.16.3.0/24 here, are not required to be of a network connected to the NAT/PAT device in order to be available, Loopback0 was configured to clearly show where this network exists in the topology.

Configuration changes on PA are then necessary, since ASA will no longer be reachable via the previously known peer IP address 172.16.2.1.

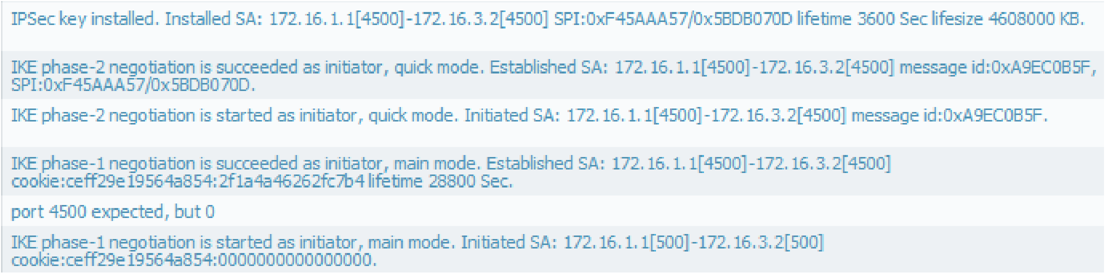


Name: VPN_Peer_ASA
Interface: ethernet1/2
Local IP Address: 172.16.1.1/30
Peer IP Type: ● Static ○ Dynamic
Peer IP Address: 172.16.3.2
Authentication: ● Pre Shared Key ○ Certificate
Pre-shared Key: ••••••••
Confirm Pre-shared Key: ••••••••
Local Identification: None
Peer Identification: IP address 172.16.2.1

Figure 48. Reconfiguring PA-200 IKE Gateway in preparation for NAT.

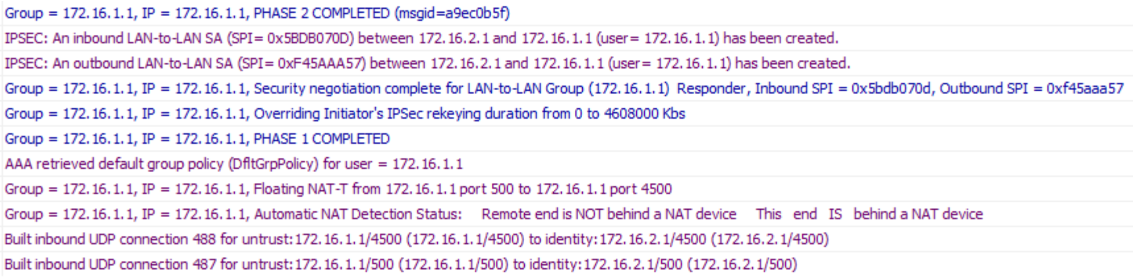
ASA's identity will remain as 172.16.2.1, but the packets destined to the peer will be sent to 172.16.3.2 instead. Since ASA is inside NAT, it will still reach PA at the address 172.16.1.1.

NAT-T has already been enabled on PA-200 during the configuration stage. This software version of ASA 5505 also seems to construct NAT-T payloads without explicitly needing to configure the feature. Without further ado, let us look at how a successful IKEv1 negotiation with NAT-T looks like, as shown in Figures 49 and 50.



```
IPSec key installed. Installed SA: 172.16.1.1[4500]-172.16.3.2[4500] SPI:0xF45AAA57/0x5BDB070D lifetime 3600 Sec lifesize 4608000 KB.
IKE phase-2 negotiation is succeeded as initiator, quick mode. Established SA: 172.16.1.1[4500]-172.16.3.2[4500] message id:0xA9EC0B5F,
SPI:0xF45AAA57/0x5BDB070D.
IKE phase-2 negotiation is started as initiator, quick mode. Initiated SA: 172.16.1.1[4500]-172.16.3.2[4500] message id:0xA9EC0B5F.
IKE phase-1 negotiation is succeeded as initiator, main mode. Established SA: 172.16.1.1[4500]-172.16.3.2[4500]
cookie:ceff29e19564a854:2f1a4a46262fc7b4 lifetime 28800 Sec.
port 4500 expected, but 0
IKE phase-1 negotiation is started as initiator, main mode. Initiated SA: 172.16.1.1[500]-172.16.3.2[500]
cookie:ceff29e19564a854:0000000000000000.
```

Figure 49. NAT-T with ASA inside NAT; PA-200 as initiator



```
Group = 172.16.1.1, IP = 172.16.1.1, PHASE 2 COMPLETED (msgid=a9ec0b5f)
IPSEC: An inbound LAN-to-LAN SA (SPI= 0x5BDB070D) between 172.16.2.1 and 172.16.1.1 (user= 172.16.1.1) has been created.
IPSEC: An outbound LAN-to-LAN SA (SPI= 0xF45AAA57) between 172.16.2.1 and 172.16.1.1 (user= 172.16.1.1) has been created.
Group = 172.16.1.1, IP = 172.16.1.1, Security negotiation complete for LAN-to-LAN Group (172.16.1.1)  Responder, Inbound SPI = 0x5bdb070d, Outbound SPI = 0xf45aaa57
Group = 172.16.1.1, IP = 172.16.1.1, Overriding Initiator's IPSec rekeying duration from 0 to 4608000 Kbs
Group = 172.16.1.1, IP = 172.16.1.1, PHASE 1 COMPLETED
AAA retrieved default group policy (DfltGrpPolicy) for user = 172.16.1.1
Group = 172.16.1.1, IP = 172.16.1.1, Floating NAT-T from 172.16.1.1 port 500 to 172.16.1.1 port 4500
Group = 172.16.1.1, IP = 172.16.1.1, Automatic NAT Detection Status:    Remote end is NOT behind a NAT device    This   end   IS   behind a NAT device
Built inbound UDP connection 488 for untrust:172.16.1.1/4500 (172.16.1.1/4500) to identity:172.16.2.1/4500 (172.16.2.1/4500)
Built inbound UDP connection 487 for untrust:172.16.1.1/500 (172.16.1.1/500) to identity:172.16.2.1/500 (172.16.2.1/500)
```

Figure 50. NAT-T with ASA inside NAT; ASA 5505 as receiver

Once NAT Discovery has been confirmed during MM messages 3–4, IKEv1 starts using UDP port 4500 instead of 500. UDP encapsulation with source and destination port 4500 continues during IKE SA negotiations, IPsec SA negotiations and then the ESP encapsulated data exchanges, which it was originally setup for. We may also verify the address translations on the NAT Router as shown in Listing 24.

```
Router# show  ip nat translations
Pro  Inside  global          Inside  local           Outside  local
Outside global
udp  172.16.3.2:500          172.16.2.1:500          172.16.1.1:500
172.16.1.1:500
udp  172.16.3.2:4500        172.16.2.1:4500        172.16.1.1:4500
172.16.1.1:4500
--- 172.16.3.2        172.16.2.1         ---               ---
```

Listing 24. Third-party router’s NAT translations with NAT-T.

In another scenario ASA was placed behind dynamic PAT instead of static NAT. The purpose of this was to see the PAT Router drop the ESP packets devoid of port numbers. However, this caused slightly unexpected results as after the successful negotiations ESP traffic was not obstructed at all. The NAT translations appeared according to the following Listing 25.

```
Router#show  ip nat translations
Pro  Inside  global          Inside  local           Outside  local
Outside global
esp      172.16.3.2:0      172.16.2.1:0      172.16.1.1:2117378768
172.16.1.1:7E34A2D0
udp  172.16.3.2:500          172.16.2.1:500          172.16.1.1:500
172.16.1.1:500
esp    172.16.3.2:3677576430    172.16.2.1:DB335CEE    172.16.1.1:0
172.16.1.1:0
```

Listing 25. Third-party router’s NAT translations without NAT-T.

The hexadecimal value ‘0x7E34A2D0’ is the SPI for ASA’s inbound IPsec SA and PA’s outbound IPsec SA. The value ‘2117378768’ on the other hand is the decimal value matching that hexadecimal value. Likewise the hexadecimal value ‘0xDB335CEE’ is the SPI for PA’s inbound IPsec SA and ASA’s outbound IPsec SA, and ‘3677576430’ its equivalent decimal number. In other words, due to a feature in Cisco IOS (Version 15.6 in this case), the ESP packets can be subjected to not only NAT but PAT as well.

In the final scenario PA’s IKE Gateway configurations were reverted to previously configured without local identifiers and the address 172.16.2.1 as ASA’s peer address. ASA was placed behind dynamic PAT instead of static NAT. In such a situation IKEv1 initiations from PA passed Router without translations, but the return traffic got translated as sourced by Router’s PA-facing address. PA views this as follows:

```
ikev1.c:456:ikev1_main(): malformed cookie received or the spi expired.
```

Since the peer IP address and cookie, equivalent to an SPI, are used to identify an IKE SA, with the peer IP address changed, the message cannot be connected to any existing SA negotiations. PA will as usual continue offering its proposal to 172.16.2.1 until timeout. In case ASA initiates the negotiations, PA will know straight away to discard the packets from this unknown peer address (Router exercising PAT) as seen from the CLI output:

```
[PROTO_ERR]:   ikev1.c:1317:isakmp_ph1begin_r():   Couldn't   find
configuration for IKE phase-1 request for peer IP 172.16.1.2[500].
```

## 6.8 Summary: Internet Security Association and Key Management Protocol (ISAKMP) Notification Messages

This section will compile the relevant ISAKMP Notification messages that were discovered during the troubleshooting section in this study. Since the errors and warnings during debugging ASA and PA were varying and changed ever-so-slightly according to each scenario, it seems more productive to gather the standardized notification messages that described the mismatch situations for future reference. Table 2 lists and explains the notification message types appearing in the Notification Payload of ISAKMP (IKE) headers.

Table 2. ISAKMP Notification messages

| Code | Notification | Explanation |
|---|---|---|
| 4 | INVALID-COOKIE | The IKE SA being referenced by the cookie pair no longer exists |
| 14 | NO-PROPOSAL-CHOSEN | IKE/IPsec policies do not match (non-zero SPIs for IPsec) |
| 16 | PAYLOAD-MALFORMED | Decrypted contents unreadable; Likely incorrect PSK |
| 18 | INVALID-ID-INFORMATION | Proxy-IDs do not match |
| 24576 | RESPONDER-LIFETIME | Responder has chosen to enforce lower SA lifetime |

The notification type ‘INVALID-ID-INFORMATION’ occurs during the IPsec Phase 2 negotiations indicating a Proxy-ID mismatch. Notifications ‘NO-PROPOSAL-CHOSEN’ and ‘RESPONDER-LIFETIME’ may be sent either during Phase 1 or Phase 2. The specifics will be found in the SPI and Data fields of the Notification Payload. In the case of attempting to decrypt a MM message 5 with an incorrect PSK, ‘PAYLOAD-MALFORMED’ was generated, since the contents could not be read. Notifications of type ‘INVALID-COOKIE’ appeared only during IKE Phase 1, when ASA received MM message 4 without recognizing the initiator and responder cookies (SPIs), since it had just a while ago deleted the association entry.

Since these are messages specified in the protocol standards they are used by both ASA and PA. However, during the testing it became obvious that PA does not support the ‘INVALID-COOKIE’, ‘PAYLOAD-MALFORMED’ or ‘INVALID-ID-INFORMATION’ notification types in the scenarios covered in this study. Thus, we may conclude that ASA implements a wider range of  ISAKMP notification messages, thus making troubleshooting easier for the peer.

# 7 Device Interoperability and Limitations

IPsec, being by design a modular protocol framework, was easily implemented on and supported by both PA-200 and ASA 5505. The terminology differed slightly between the vendors such as ‘crypto map’ or transform-set’. The appendices will serve as references for the differences between CLI commands, but going through them is outside of this study.

Among the configuration steps, the most obvious variation between the two was ASA’s requirement for a policy-based VPN arrangement. Another noteworthy difference was support for DH groups; this software version of IOS for ASA would not support beyond group 5, while PA could go for the more secure group 14. The focus on IKEv1 over IKEv2 was also solidified by the fact that this PAN-OS version did not have IKEv2 support yet implemented. Thus, in this study it is no overstatement to consider software versions playing at least as big a role as the vendor difference.

With regard to troubleshooting means, both devices have features for packet capture, system logging, CLI debugging and status observations. ASA’s debugging capabilities appear cleaner and less cryptic compared to PA which tends to leave the majority of the message contents in hexadecimal form, though PA provides more insight on the operations taking place in the background such as hashing. Then again, Cisco and

Cisco’s ASA have more documentation available than Palo Alto, so this study may not have covered the most optimal troubleshooting tools – especially for PA-200.

## 8 Conclusion

The goal of this thesis was to configure and systematically troubleshoot an IPsec Site-to-Site VPN. The results were to be used to aid future troubleshooting and to compare the devices’ VPN troubleshooting capabilities.

The initial device setup for the practical part of the study went without notable issues. The Configuration stage was based on various other studies which came with a lessons learned input. As a result the configuring was complete as soon as the policy-based VPN constraint on ASA’s end was addressed.

VPN Experimenting and Troubleshooting section attempted to cover as many different cases and scenarios as possible, but still ended up limiting the number of subjects quite considerably such as omitting Transport mode or other authentication methods. As more scenarios were covered my desire for a deeper level of understanding grew. This is evident from the fact that ASA’s debugging was done at level 127 in the beginning and at 254 from halfway through until the end. My personal understanding by the end of experimenting was quite satisfactory, which alone made the project worthwhile.

As for lessons learned: with regard to information gathering, regardless of the vendor, the receiver will have a clearer understanding of the underlying issue. The initiator will hopefully have means to receive and decipher any notification messages sent by the receiver to understand why the negotiations are not progressing as they should. In this study, due to ASA 5505’s wider support for ISAKMP notifications, PA-200 had access to considerably more information.

## References

1 Umesha N., Umesh H.R. The InfoSec Handbook: An Introduction to Information Security. Apress; 2014.

2 Morgan B., Lovering N. CCNP ISCW Official Exam Certification Guide. USA, Cisco Press; 2007.

3 Frankel S., Krishnan S. IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap [online]. Internet Engineering Task Force (IETF); 2011.
URL: https://tools.ietf.org/html/rfc6071.
Accessed: 17 March 2017.

4 CCNA Security: Implementing Virtual Private Networks [offline slideshow]. Cisco Systems; 2012.

5 Krawczyk H., Bellare M., Canetti R. HMAC: Keyed-Hashing for Message Authentication [online]. Network Working Group; 1997.
URL: https://tools.ietf.org/html/rfc2104
Accessed: 20 March 2017.

6 Configuring Security for VPNs with IPsec [online]. Cisco Systems; 2012.
URL: http://www.cisco.com/en/US/docs/ios-xml/ios/sec_conn_vpnips/configuration/15-2mt/sec-cfg-vpn-IPsec.html
Accessed: 20 March 2017.

7 Kozierok C.M. The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference. USA, No Starch Press; 2005.

8 Kent S. IP Encapsulating Security Payload (ESP) [online]. Network Working Group; 2005.
URL: https://tools.ietf.org/html/rfc4303.
Accessed: 21 March 2017.

9 Harkins D., Carrel D. The Internet Key Exchange (IKE) [online]. Network Working Group; 1998.
URL: https://tools.ietf.org/html/rfc2409
Accessed: 22 March 2017.

10 Kaufman C., Hoffman P., Nir Y., Eronen P., Kivinen T. Internet Key Exchange Protocol Version 2 (IKEv2) [online]. Internet Engineering Task Force; 2014.
URL: https://tools.ietf.org/html/rfc7296
Accessed: 25 March 2017.

11 Palo Alto Networks Enterprise Firewall PA-200 [online]. Virtual Graffiti Inc.; 2014.
URL: http://www.paloguard.com/Firewall-PA-200.asp
Accessed: 28 March 2017.

12 Piper D. The Internet IP Security Domain of Interpretation for ISAKMP [online]. Network Working Group; 1998.
URL: https://tools.ietf.org/html/rfc2407
Accessed: 15 April 2017.

13 Aboba B., Dixon W. IPsec-Network Address Translation (NAT) Compatibility Requirements [online]. Network Working Group; 2004.
URL: https://tools.ietf.org/html/rfc3715
Accessed: 16 April 2017.

## Appendix 1: ASA 5505 Running Configuration

```
ASA-5505# show running-config
: Saved
:
: Serial Number: JMX16304066
: Hardware:   ASA5505, 512 MB RAM, CPU Geode 500 MHz
:
ASA Version 9.2(4)
!
hostname ASA-5505
enable password uklS0vKXrJp/cCYm encrypted
xlate per-session deny tcp any4 any4
xlate per-session deny tcp any4 any6
xlate per-session deny tcp any6 any4
xlate per-session deny tcp any6 any6
xlate per-session deny udp any4 any4 eq domain
xlate per-session deny udp any4 any6 eq domain
xlate per-session deny udp any6 any4 eq domain
xlate per-session deny udp any6 any6 eq domain
names
!
interface Ethernet0/0
 switchport access vlan 10
!
interface Ethernet0/1
 switchport access vlan 20
!
interface Ethernet0/2
 shutdown
!
interface Ethernet0/3
 shutdown
!
interface Ethernet0/4
 shutdown
!
interface Ethernet0/5
 shutdown
!
interface Ethernet0/6
 shutdown
!
interface Ethernet0/7
 shutdown
!
interface Vlan1
 no nameif
 no security-level
 no ip address
!
```

```
interface Vlan10
 nameif trust
 security-level 100
 ip address 10.0.2.254 255.255.255.0
!
interface Vlan20
 nameif untrust
 security-level 0
 ip address 172.16.2.1 255.255.255.252
!
ftp mode passive
clock timezone GMT 0
object network 172.16.2.2
 host 172.16.2.2
 description Default Gateway
object network 10.0.1.0_24
 subnet 10.0.1.0 255.255.255.0
 description PA_LAN
object network 10.0.2.0_24
 subnet 10.0.2.0 255.255.255.0
 description ASA_LAN
access-list PROXY-ID-ACL extended permit ip object 10.0.2.0_24 object 10.0.1.0_24
access-list FALSE-ACL extended permit ip any any
pager lines 100
logging enable
logging asdm informational
mtu trust 1500
mtu untrust 1500
icmp unreachable rate-limit 1 burst-size 1
no asdm history enable
arp timeout 14400
no arp permit-nonconnected
nat (trust,untrust) source static 10.0.2.0_24 10.0.2.0_24 destination static 10.0.1.0_24
10.0.1.0_24 no-proxy-arp route-lookup
!
nat (trust,untrust) after-auto source dynamic any interface
route untrust 0.0.0.0 0.0.0.0 172.16.2.2 1
timeout xlate 3:00:00
timeout pat-xlate 0:00:30
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02
timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00 mgcp-pat 0:05:00
timeout sip 0:30:00 sip_media 0:02:00 sip-invite 0:03:00 sip-disconnect 0:02:00
timeout sip-provisional-media 0:02:00 uauth 0:05:00 absolute
timeout tcp-proxy-reassembly 0:01:00
timeout floating-conn 0:00:00
dynamic-access-policy-record DfltAccessPolicy
user-identity default-domain LOCAL
http server enable
http 10.0.2.1 255.255.255.255 trust
no snmp-server location
no snmp-server contact
crypto ipsec ikev1 transform-set ESP-AES1-SHA esp-aes esp-sha-hmac
crypto ipsec security-association pmtu-aging infinite
crypto map VPN_MAP 1 match address PROXY-ID-ACL

crypto map VPN_MAP 1 set peer 172.16.1.1
crypto map VPN_MAP 1 set ikev1 transform-set ESP-AES1-SHA
crypto map VPN_MAP 1 set security-association lifetime seconds 3600
crypto map VPN_MAP interface untrust
crypto ca trustpool policy
crypto ikev1 enable untrust
crypto ikev1 am-disable
crypto ikev1 policy 10
 authentication pre-share
 encryption aes
 hash sha
 group 5
 lifetime 28800
telnet timeout 5
ssh stricthostkeycheck
ssh timeout 5
ssh key-exchange group dh-group1-sha1
console timeout 0

threat-detection basic-threat
threat-detection statistics access-list
no threat-detection statistics tcp-intercept
tunnel-group 172.16.1.1 type ipsec-l2l
tunnel-group 172.16.1.1 ipsec-attributes
 ikev1 pre-shared-key *****
!
class-map inspection_default
 match default-inspection-traffic
!
!
policy-map type inspect dns preset_dns_map
 parameters
  message-length maximum client auto
  message-length maximum 512
policy-map global_policy
 class inspection_default
  inspect dns preset_dns_map
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect ip-options
  inspect netbios
  inspect rsh
  inspect rtsp
  inspect skinny
  inspect esmtp
  inspect sqlnet
  inspect sunrpc
  inspect tftp
  inspect sip
  inspect xdmcp
```

```
!
service-policy global_policy global
prompt hostname context
no call-home reporting anonymous
call-home
 profile CiscoTAC-1
  no active
  destination address http
https://tools.cisco.com/its/service/oddce/services/DDCEService
  destination address email callhome@cisco.com
  destination transport-method http
  subscribe-to-alert-group diagnostic
  subscribe-to-alert-group environment
  subscribe-to-alert-group inventory periodic monthly
  subscribe-to-alert-group configuration periodic monthly
  subscribe-to-alert-group telemetry periodic daily
Cryptochecksum:b6c701a6071a2091643304f06c8e6fe6
: end.
```

## Appendix 2: PA-200 Configuration File

```
admin@PA-200> set cli config-output-format set
admin@PA-200> configure
Entering configuration mode
[edit]
admin@PA-200# show
set deviceconfig system ip-address 192.168.1.1
set deviceconfig system netmask 255.255.255.0
set deviceconfig system update-server updates.paloaltonetworks.com
set deviceconfig system update-schedule threats recurring weekly day-of-week wednesday
set deviceconfig system update-schedule threats recurring weekly at 01:02
set deviceconfig system update-schedule threats recurring weekly action download-only
set deviceconfig system timezone US/Pacific
set deviceconfig system service disable-telnet yes
set deviceconfig system service disable-http yes
set deviceconfig system hostname PA-200
set deviceconfig system route service
set deviceconfig setting config rematch yes
set deviceconfig setting management hostname-type-in-syslog FQDN
set network interface ethernet ethernet1/1 layer3 ipv6 neighbor-discovery router-advertisement enable no
set network interface ethernet ethernet1/1 layer3 ipv6 address fd00:1:1::254/64 advertise enable no
set network interface ethernet ethernet1/1 layer3 ipv6 address fd00:1:1::254/64 advertise valid-lifetime 2592000
set network interface ethernet ethernet1/1 layer3 ipv6 address fd00:1:1::254/64 advertise preferred-lifetime 604800
set network interface ethernet ethernet1/1 layer3 ipv6 address fd00:1:1::254/64 advertise onlink-flag yes
set network interface ethernet ethernet1/1 layer3 ipv6 address fd00:1:1::254/64 advertise auto-config-flag yes
set network interface ethernet ethernet1/1 layer3 ipv6 address fd00:1:1::254/64 enable-on-interface yes
set network interface ethernet ethernet1/1 layer3 ipv6 enabled yes
set network interface ethernet ethernet1/1 layer3 ip 10.0.1.254/24
set network interface ethernet ethernet1/1 layer3 interface-management-profile MGMT
set network interface ethernet ethernet1/2 layer3 ipv6 neighbor-discovery router-advertisement enable no
set network interface ethernet ethernet1/2 layer3 ip 172.16.1.1/30
set network interface ethernet ethernet1/2 layer3 interface-management-profile PING
set network interface ethernet ethernet1/3
set network interface ethernet ethernet1/4
set network interface loopback units
set network interface vlan units
set network interface tunnel units tunnel.1 comment Tunnel_to_ASA
set network interface tunnel units tunnel.1 ipv6 enabled yes
set network vlan
set network virtual-wire
set network profiles monitor-profile default interval 3
set network profiles monitor-profile default threshold 5
set network profiles monitor-profile default action wait-recover
set network profiles interface-management-profile MGMT https yes
set network profiles interface-management-profile MGMT ssh yes
set network profiles interface-management-profile MGMT ping yes
set network profiles interface-management-profile PING ping yes
set network ike crypto-profiles ike-crypto-profiles default encryption [ aes128 3des ]
set network ike crypto-profiles ike-crypto-profiles default hash sha1
set network ike crypto-profiles ike-crypto-profiles default dh-group group2
set network ike crypto-profiles ike-crypto-profiles default lifetime hours 8
set network ike crypto-profiles ike-crypto-profiles IKE_P1_ASA hash sha1
set network ike crypto-profiles ike-crypto-profiles IKE_P1_ASA dh-group group5
set network ike crypto-profiles ike-crypto-profiles IKE_P1_ASA encryption aes128
set network ike crypto-profiles ike-crypto-profiles IKE_P1_ASA lifetime hours 8
set network ike crypto-profiles ipsec-crypto-profiles default esp encryption [ aes128
3des ]
set network ike crypto-profiles ipsec-crypto-profiles default esp authentication sha1
set network ike crypto-profiles ipsec-crypto-profiles default dh-group group2
set network ike crypto-profiles ipsec-crypto-profiles default lifetime hours 1
set network ike crypto-profiles ipsec-crypto-profiles IPSec_P2_ASA lifetime hours 1
set network ike crypto-profiles ipsec-crypto-profiles IPSec_P2_ASA dh-group no-pfs
set network ike crypto-profiles ipsec-crypto-profiles IPSec_P2_ASA esp authentication
sha1
set network ike crypto-profiles ipsec-crypto-profiles IPSec_P2_ASA esp encryption
aes128
set network ike gateway VPN_Peer_ASA protocol ikev1 dpd enable yes
set network ike gateway VPN_Peer_ASA protocol ikev1 dpd interval 10
set network ike gateway VPN_Peer_ASA protocol ikev1 dpd retry 2
set network ike gateway VPN_Peer_ASA protocol ikev1 ike-crypto-profile IKE_P1_ASA
set network ike gateway VPN_Peer_ASA protocol ikev1 exchange-mode main
set network ike gateway VPN_Peer_ASA local-address ip 172.16.1.1/30
set network ike gateway VPN_Peer_ASA local-address interface ethernet1/2
set network ike gateway VPN_Peer_ASA authentication pre-shared-key key -
AQ==sr/IwvWjbANysxvwl7+9z7N/fs0=Sc7KRgDWgT68mpaktWdlJg==
set network ike gateway VPN_Peer_ASA protocol-common nat-traversal enable yes
set network ike gateway VPN_Peer_ASA protocol-common fragmentation enable no
set network ike gateway VPN_Peer_ASA peer-address ip 172.16.3.2
set network ike gateway VPN_Peer_ASA peer-id id 172.16.2.1
set network ike gateway VPN_Peer_ASA peer-id type ipaddr
set network qos profile default class class1 priority real-time
set network qos profile default class class2 priority high
set network qos profile default class class3 priority high
set network qos profile default class class4 priority medium
set network qos profile default class class5 priority medium
set network qos profile default class class6 priority low
set network qos profile default class class7 priority low
set network qos profile default class class8 priority low
set network virtual-router default protocol bgp enable no
set network virtual-router default protocol bgp dampening-profile default cutoff 1.25
set network virtual-router default protocol bgp dampening-profile default reuse 0.5
set network virtual-router default protocol bgp dampening-profile default max-hold-time
900
```

```
set network virtual-router default protocol bgp dampening-profile default decay-half-life-reachable 300
set network virtual-router default protocol bgp dampening-profile default decay-half-life-unreachable 900
set network virtual-router default protocol bgp dampening-profile default enable yes
set network virtual-router default protocol bgp routing-options graceful-restart enable yes
set network virtual-router default protocol bgp routing-options as-format 2-byte
set network virtual-router default interface [ ethernet1/1 ethernet1/2 tunnel.1 ]
set network virtual-router default routing-table ip static-route ASA_LAN interface tunnel.1
set network virtual-router default routing-table ip static-route ASA_LAN metric 10
set network virtual-router default routing-table ip static-route ASA_LAN destination 10.0.2.0/24
set network virtual-router default routing-table ip static-route Default_Route nexthop ip-address 172.16.1.2
set network virtual-router default routing-table ip static-route Default_Route interface ethernet1/2
set network virtual-router default routing-table ip static-route Default_Route metric 10
set network virtual-router default routing-table ip static-route Default_Route destination 0.0.0.0/0
set network virtual-router default routing-table ipv6 static-route ASA_LAN_IPv6 interface tunnel.1
set network virtual-router default routing-table ipv6 static-route ASA_LAN_IPv6 metric 10
set network virtual-router default routing-table ipv6 static-route ASA_LAN_IPv6 destination fd00:2:2::/64
set network tunnel ipsec Tunnel_to_ASA auto-key ike-gateway VPN_Peer_ASA
set network tunnel ipsec Tunnel_to_ASA auto-key ipsec-crypto-profile IPSec_P2_ASA
set network tunnel ipsec Tunnel_to_ASA auto-key proxy-id Policy-based_ASA protocol any
set network tunnel ipsec Tunnel_to_ASA auto-key proxy-id Policy-based_ASA local 10.0.1.0/24
set network tunnel ipsec Tunnel_to_ASA auto-key proxy-id Policy-based_ASA remote 10.0.2.0/24
set network tunnel ipsec Tunnel_to_ASA tunnel-monitor enable no
set network tunnel ipsec Tunnel_to_ASA tunnel-monitor destination-ip 10.0.2.254
set network tunnel ipsec Tunnel_to_ASA tunnel-interface tunnel.1
set shared application
set shared application-group
set shared service
set shared service-group
set shared botnet configuration http dynamic-dns enabled yes
set shared botnet configuration http dynamic-dns threshold 5
set shared botnet configuration http malware-sites enabled yes
set shared botnet configuration http malware-sites threshold 5
set shared botnet configuration http recent-domains enabled yes
set shared botnet configuration http recent-domains threshold 5
set shared botnet configuration http ip-domains enabled yes
set shared botnet configuration http ip-domains threshold 10
set shared botnet configuration http executables-from-unknown-sites enabled yes
set shared botnet configuration http executables-from-unknown-sites threshold 5
set shared botnet configuration other-applications irc yes
set shared botnet configuration unknown-applications unknown-tcp destinations-per-hour 10
set shared botnet configuration unknown-applications unknown-tcp sessions-per-hour 10
set shared botnet configuration unknown-applications unknown-tcp session-length maximum-bytes 100
set shared botnet configuration unknown-applications unknown-tcp session-length minimum-bytes 50
set shared botnet configuration unknown-applications unknown-udp destinations-per-hour 10
set shared botnet configuration unknown-applications unknown-udp sessions-per-hour 10
set shared botnet configuration unknown-applications unknown-udp session-length maximum-bytes 100
set shared botnet configuration unknown-applications unknown-udp session-length minimum-bytes 50
set shared botnet report topn 100
set shared botnet report scheduled yes
set rulebase security rules allow_LAN_to_INET from trust
set rulebase security rules allow_LAN_to_INET to untrust
set rulebase security rules allow_LAN_to_INET source any
set rulebase security rules allow_LAN_to_INET destination any
set rulebase security rules allow_LAN_to_INET service any
set rulebase security rules allow_LAN_to_INET application any
set rulebase security rules allow_LAN_to_INET action allow
set rulebase security rules allow_LAN_to_INET log-end yes
set rulebase security rules allow_LAN_to_INET source-user any
set rulebase security rules allow_LAN_to_INET category any
set rulebase security rules allow_LAN_to_INET hip-profiles any
set rulebase security rules allow_LAN_to_VPN to vpn
set rulebase security rules allow_LAN_to_VPN from trust
set rulebase security rules allow_LAN_to_VPN source 10.0.1.0/24
set rulebase security rules allow_LAN_to_VPN destination 10.0.2.0/24
set rulebase security rules allow_LAN_to_VPN source-user any
set rulebase security rules allow_LAN_to_VPN category any
set rulebase security rules allow_LAN_to_VPN application any
set rulebase security rules allow_LAN_to_VPN service any
set rulebase security rules allow_LAN_to_VPN hip-profiles any
set rulebase security rules allow_LAN_to_VPN action allow
set rulebase security rules allow_LAN_to_VPN log-end yes
set rulebase security rules allow_VPN_to_LAN to trust
set rulebase security rules allow_VPN_to_LAN from vpn
set rulebase security rules allow_VPN_to_LAN source 10.0.2.0/24
set rulebase security rules allow_VPN_to_LAN destination 10.0.1.0/24
set rulebase security rules allow_VPN_to_LAN source-user any
set rulebase security rules allow_VPN_to_LAN category any
set rulebase security rules allow_VPN_to_LAN application any
set rulebase security rules allow_VPN_to_LAN service any
set rulebase security rules allow_VPN_to_LAN hip-profiles any
set rulebase security rules allow_VPN_to_LAN action allow
set rulebase security rules allow_VPN_to_LAN log-end yes
set rulebase security rules allow_ICMP_IKE_IPSec to untrust
set rulebase security rules allow_ICMP_IKE_IPSec from untrust
set rulebase security rules allow_ICMP_IKE_IPSec source [ 172.16.1.1 172.16.2.1
172.16.3.2_NAT-T ]
set rulebase security rules allow_ICMP_IKE_IPSec destination [ 172.16.1.1 172.16.2.1
172.16.3.2_NAT-T ]
set rulebase security rules allow_ICMP_IKE_IPSec source-user any
set rulebase security rules allow_ICMP_IKE_IPSec category any
set rulebase security rules allow_ICMP_IKE_IPSec application [ ciscovpn icmp ike ip-
sec-esp ipsec-esp-udp ]
set rulebase security rules allow_ICMP_IKE_IPSec service any
set rulebase security rules allow_ICMP_IKE_IPSec hip-profiles any
set rulebase security rules allow_ICMP_IKE_IPSec action allow
set rulebase security rules allow_ICMP_IKE_IPSec rule-type universal
set rulebase security rules allow_ICMP_IKE_IPSec log-start yes
set rulebase security rules allow_ICMP_IKE_IPSec log-end yes
set rulebase security rules allow_VPN_ESP to untrust
set rulebase security rules allow_VPN_ESP from vpn
set rulebase security rules allow_VPN_ESP source 172.16.1.1
set rulebase security rules allow_VPN_ESP destination [ 172.16.2.1 172.16.3.2_NAT-T
]
set rulebase security rules allow_VPN_ESP source-user any
set rulebase security rules allow_VPN_ESP category any
set rulebase security rules allow_VPN_ESP application [ ipsec-esp ipsec-esp-udp ]
set rulebase security rules allow_VPN_ESP service any
set rulebase security rules allow_VPN_ESP hip-profiles any
set rulebase security rules allow_VPN_ESP action allow
set rulebase security rules allow_VPN_ESP log-start yes
set rulebase nat rules SNAT_LAN_to_INET to untrust
set rulebase nat rules SNAT_LAN_to_INET from trust
set rulebase nat rules SNAT_LAN_to_INET source 10.0.1.0/24
set rulebase nat rules SNAT_LAN_to_INET destination any
set rulebase nat rules SNAT_LAN_to_INET service any
set rulebase nat rules SNAT_LAN_to_INET to-interface ethernet1/2
set rulebase nat rules SNAT_LAN_to_INET nat-type ipv4
set rulebase nat rules SNAT_LAN_to_INET source-translation dynamic-ip-and-port
interface-address ip 172.16.1.1/30
set rulebase nat rules SNAT_LAN_to_INET source-translation dynamic-ip-and-port
interface-address interface ethernet1/2
set application-group
set application
set schedule
set address 172.16.1.1 ip-netmask 172.16.1.1/32
set address 172.16.2.1 ip-netmask 172.16.2.1/32
set address 172.16.3.2_NAT-T ip-netmask 172.16.3.2/32
set service-group
set service
set zone trust network layer3 ethernet1/1
set zone untrust network layer3 ethernet1/2
set zone vpn network layer3 tunnel.1
set import network interface [ ethernet1/1 ethernet1/2 tunnel.1 ]
set mgt-config users admin phash $1$agvfpfbm$WsgVKqkB0JEpBoeDyG1A4/
set mgt-config users admin permissions role-based superuser yes
```