The Subtle Art of Chaining Headers

IKEv2 Attack Surface Case Study

@AntoniosAtlasis

December 2020

Disclaimer

- **The content of this presentation is personal work of its author. It is not related by any means with his current or past employers, and it does not constitute any kind of recommendation or official endorsement.**

[localhost] $ whoami

Cyber Security Engineer at European Space Agency during day.

IT Security Researcher for fun at night :-)

Presenter at various Security Cons

- BlackHat, Troopers, Hack in the Box, Brucon, Deepsec, etc.

Main are of interest: Security Analysis of Network Protocols.

- IPv6 has been my favourite :-)

You can follow on twitter @AntoniosAtlasis

Personal blog post: www.secfu.net.

Outline

- **Introduction: Motivation and Objective**
- **Basic IKEv2 Background**
- **IKEv2 Attack Surface & Attacking Possibilities**
- **‘*yIKEs*’: An open-source tool for IKEv2 security assessment**
  - Released today for first time
- **Conclusions**

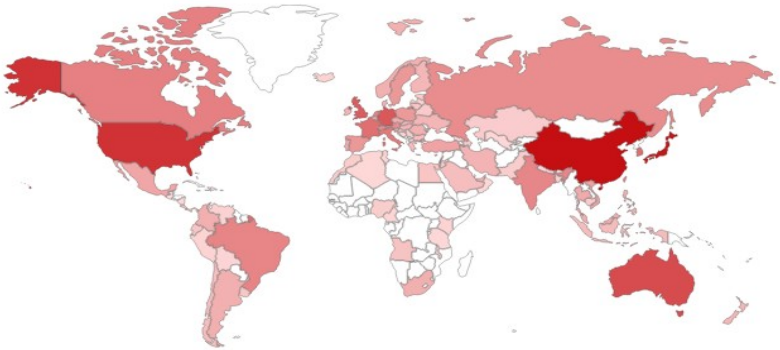
Why is the IKEv2 Analysis Important?

SHODAN

TOTAL RESULTS

4,048,715

TOP COUNTRIES



China	978,528
Japan	874,490
United States	484,790
Australia	261,341
Germany	190,352

- **IKE is the key-negotiation mechanism for IPSec, one of the main solutions for establishing VPNs.**
  - Some of these solutions are even accredited for the exchange of classified information.

TOP SERVICES

IKE	3,958,817
IKE-NAT-T	89,461

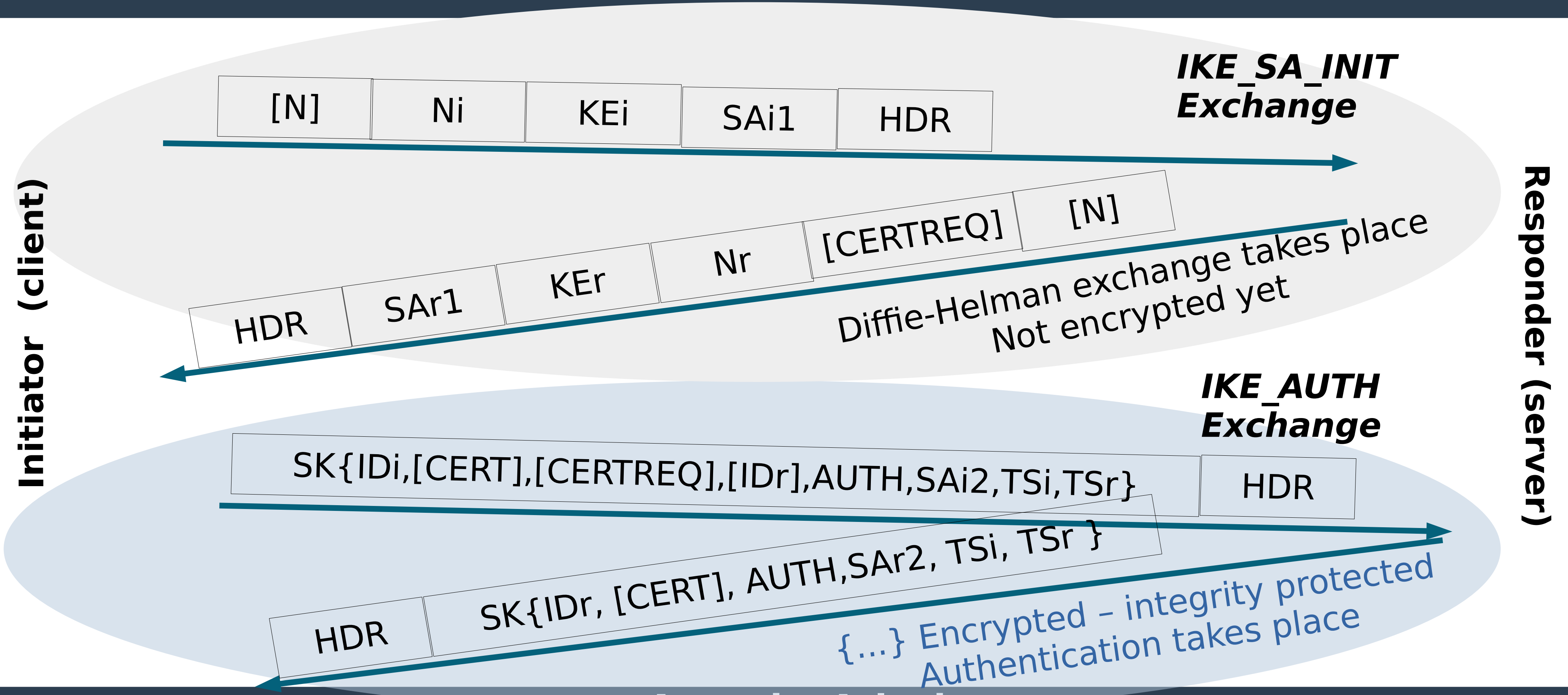
Objective

black hat EUROPE 2020

- **Examine the IKEv2 attack surface**
  - from an unauthenticated attacker's perspective.
  - By analysing the specifications (RFCs).
  - By testing specific implementations.
- **This talk will not reveal any new vulnerability.**
- **But it will help you understand areas of potential exploitation.**
- **An open-source tool is released today capable of implementing the described attacks.**
- **This is not a cryptographic talk**
  - We will not discuss potential crypto weaknesses – I am not a cryptographer after all.

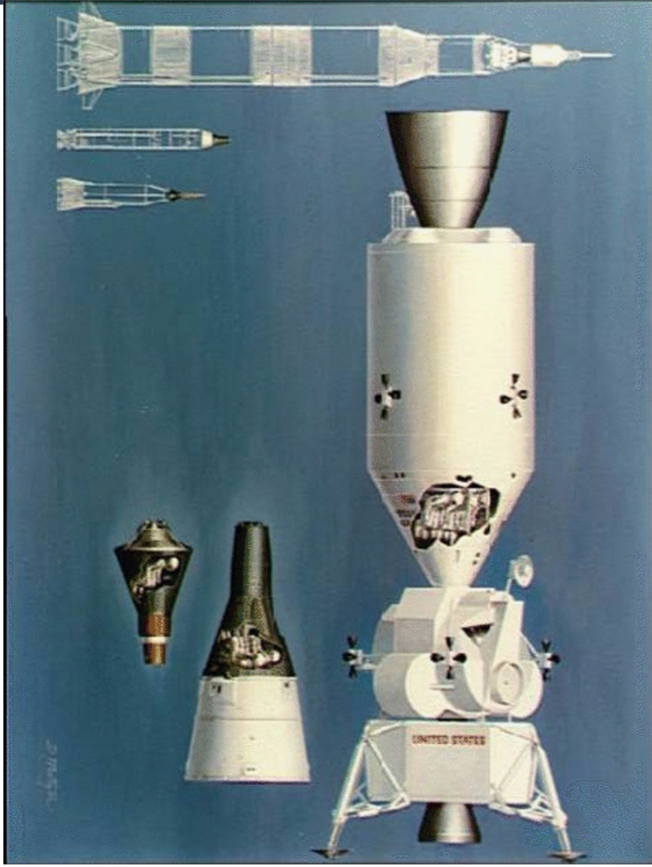
Basic IKEv2 Background

The IKEv2 SA Establishment



IKEv2 Payloads

Value	Next Payload Type	Notation	Reference
0	No Next Payload		[RFC7296]
1-32	Reserved		[RFC7296]
33	Security Association	SA	[RFC7296]
34	Key Exchange	KE	[RFC7296]
35	Identification - Initiator	IDi	[RFC7296]
36	Identification - Responder	IDr	[RFC7296]
37	Certificate	CERT	[RFC7296]
38	Certificate Request	CERTREQ	[RFC7296]
39	Authentication	AUTH	[RFC7296]
40	Nonce	Ni, Nr	[RFC7296]
41	Notify	N	[RFC7296]
42	Delete	D	[RFC7296]
43	Vendor ID	V	[RFC7296]
44	Traffic Selector - Initiator	TSi	[RFC7296]
45	Traffic Selector - Responder	TSr	[RFC7296]
46	Encrypted and Authenticated	SK	[RFC7296]
47	Configuration	CP	[RFC7296]
48	Extensible Authentication	EAP	[RFC7296]
49	Generic Secure Password Method	GSPM	[RFC6467]
50	Group Identification	IDg	[draft-yeung-g-ikev2]
51	Group Security Association	GSA	[draft-yeung-g-ikev2]
52	Key Download	KD	[draft-yeung-g-ikev2]
53	Encrypted and Authenticated Fragment	SKF	[RFC7383]
54	Puzzle Solution	PS	[RFC8019]
55-127	Unassigned		
128-255	Private use		[RFC7296]



https://www.iana.org/assignments/ikev2-parameters/ikev2-parameters.xhtml

and IKEv2 Payload Chaining

- **Each IKE payload starts with the following generic payload header:**

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Refers to the next Payload

Source: IETF RFC 7296

Security Association (SA) Payload

```
SA Payload
   |
   +--- Proposal #1 ( Proto ID = ESP(3), SPI size = 4,
   |        |              7 transforms,      SPI = 0x052357bb )
   |        |
   |        +-- Transform ENCR ( Name = ENCR_AES_CBC )
   |        |     +-- Attribute ( Key Length = 128 )
   |        |
   |        +-- Transform ENCR ( Name = ENCR_AES_CBC )
   |        |     +-- Attribute ( Key Length = 192 )
   |        |
   |        +-- Transform ENCR ( Name = ENCR_AES_CBC )
   |        |     +-- Attribute ( Key Length = 256 )
   |        |
   |        +-- Transform INTEG ( Name = AUTH_HMAC_SHA1_96 )
   |        +-- Transform INTEG ( Name = AUTH_AES_XCBC_96 )
   |        +-- Transform ESN ( Name = ESNs )
   |        +-- Transform ESN ( Name = No ESNs )
   |
   +--- Proposal #2 ( Proto ID = ESP(3), SPI size = 4,
            |              4 transforms,      SPI = 0x35a1d6f2 )
            |
            +-- Transform ENCR ( Name = AES-GCM with a 8 octet ICV )
            |     +-- Attribute ( Key Length = 128 )
            |
            +-- Transform ENCR ( Name = AES-GCM with a 8 octet ICV )
            |     +-- Attribute ( Key Length = 256 )
            |
            +-- Transform ESN ( Name = ESNs )
            +-- Transform ESN ( Name = No ESNs )
```

- **An SA can have one or more Proposals.**
- **Each Proposal can have one or more Transforms.**
- **Each Transform can have one or more Attributes.**

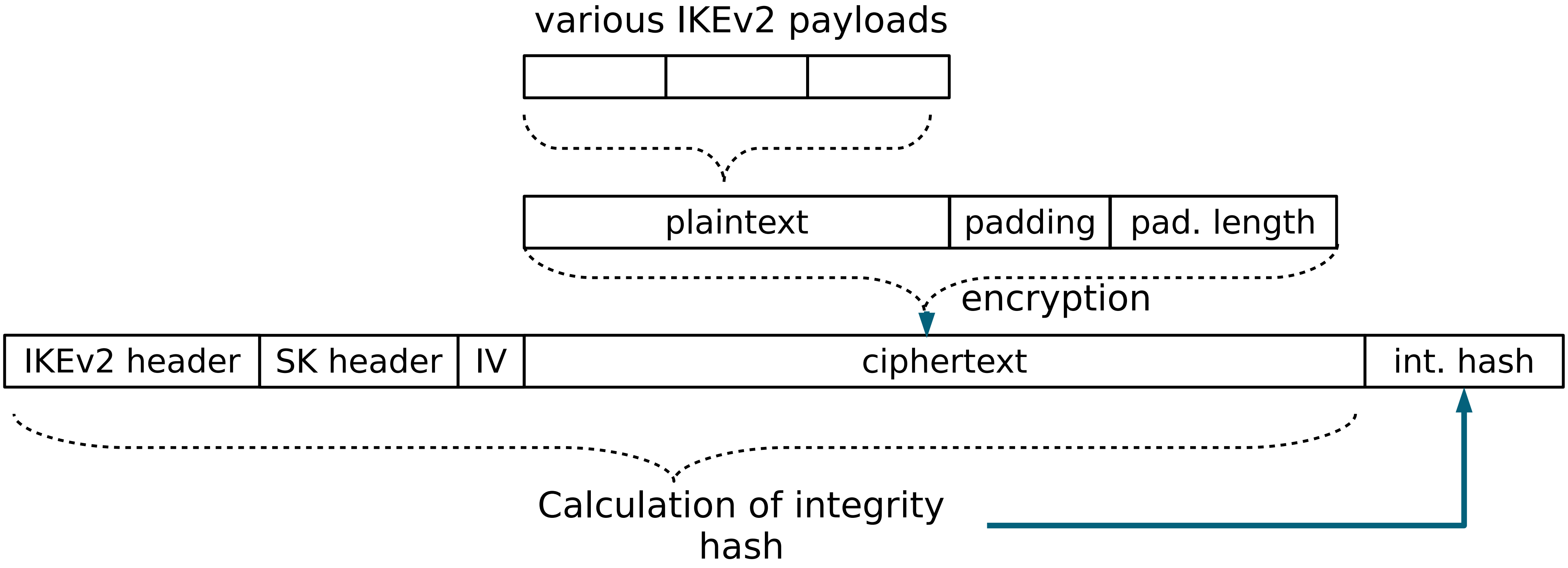
Notify Payload

- **Potentially almost unlimited length!**
- **Potentially unlimited different types (> 70 already defined).**

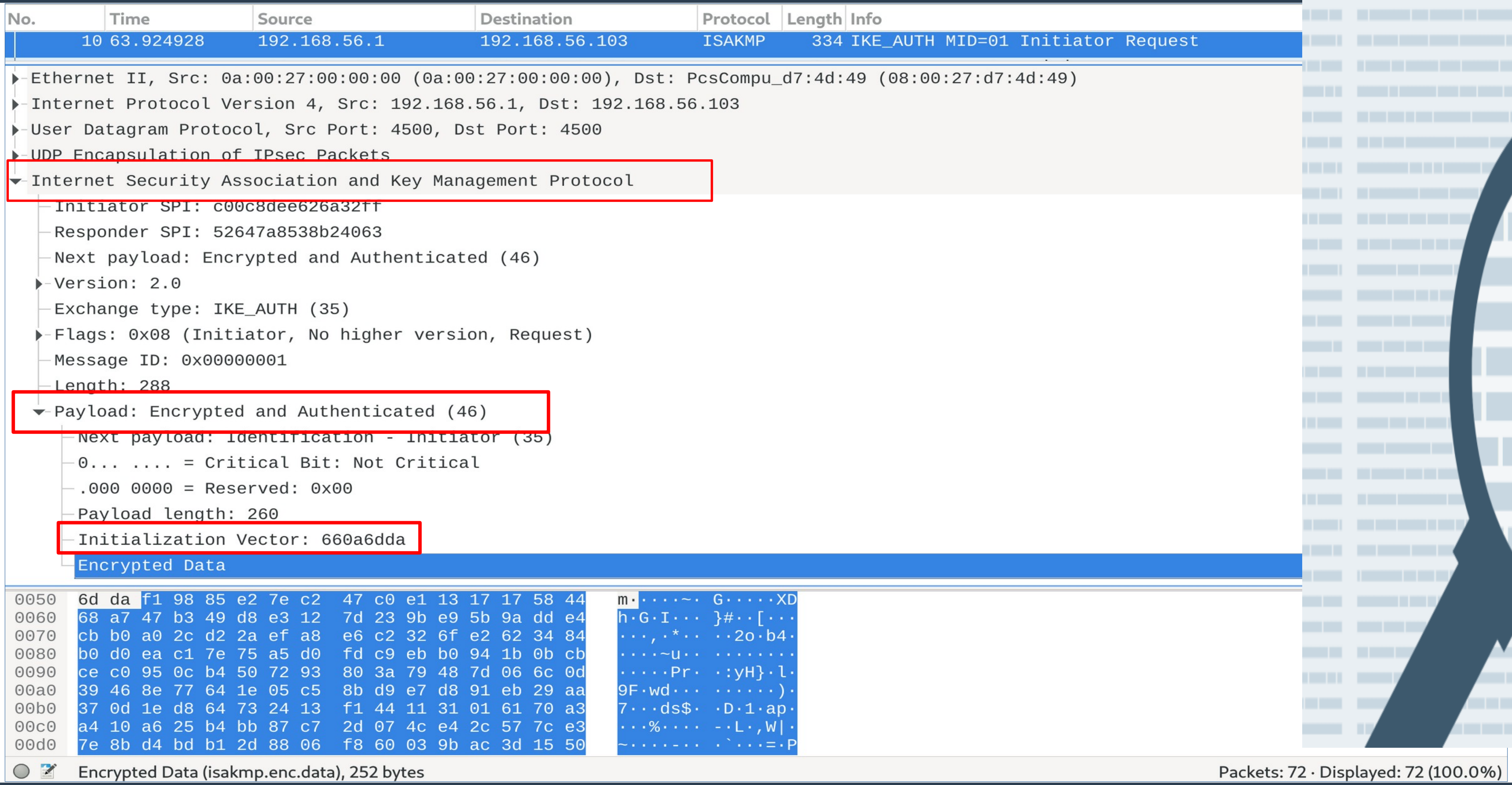
```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Protocol ID  |   SPI Size    |      Notify Message Type      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                Security Parameter Index (SPI)                 ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                       Notification Data                       ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

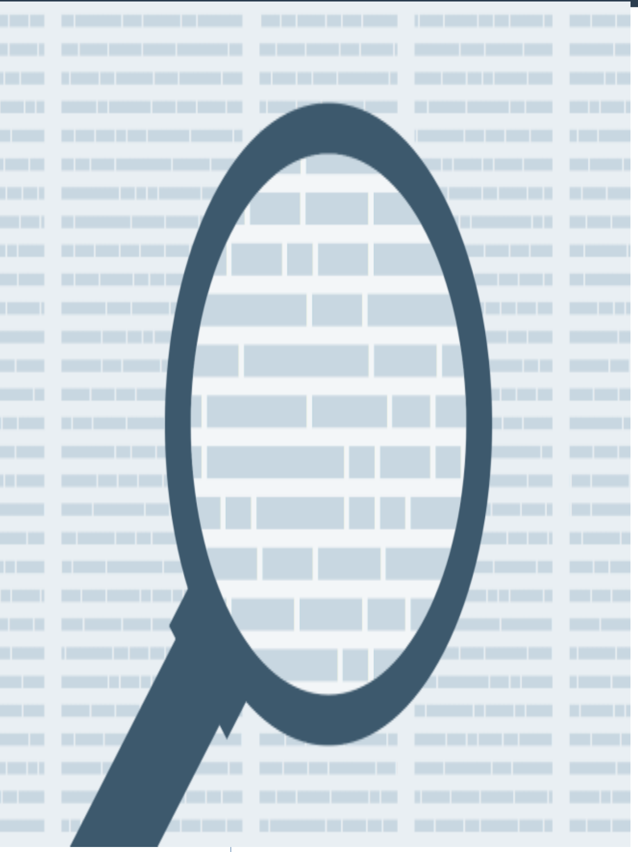
Source:
IETF RFC 7296

How “Encrypted and Authenticated” {SK} Payload is constructed



How an IKE_AUTH looks like





IKEv2 Attack Surface & Attacking Possibilities

Attack Opportunities for an Unauthenticated Attacker

- **At a first glance, IKEv2 is simple.**
- **Room for potential abuse:**
  - **IKE_INIT** Exchange
    - Not Encrypted/integrity protected, not authenticated
    - Open for MITM (authentication challenge follows)
  - **IKE_AUTH** Exchange (?)
    - Encrypted, integrity protected, authentication takes place at the end of it.
    - An initiator still have some changes for abuse, until authentication step.
  - **Child SAs** is not an option
    - The other end has already been authenticated.

yIKEs – an open source IKEv2 Security Assessment Tool

- ***Python3*** **tool, requires Scapy library; you need to be** ***root*** **:-)**
- **Auto-configures (bocks) iptables** ***ICMP Destination Unreachable***
- **To perform <u>successful</u> Diffie-Helman Exchange and IKE_AUTH Encryption/Decryption, currently only the following are supported:**
  - Diffie-Helman Group: 2
  - Encryption Key length: 256
  - Encryption algorithm: AES-CBC
  - Integrity protection algorithm: SHA2-256-128
  - PRF: PRF_HMAC_SHA2_256

Triggered IKE_AUTH Responses

- *yIKEs* does not implement successful authentication (due to its testing objective).

- It does perform though successful Diffie-Helman exchange (to trigger the "Authentication Failed" Notification and all potential attacks up to this point).

```
[root@linux IKEv2]# ./yIKEs.py -d 192.168.56.101 -i vboxnet0 -recon -listen
Number of Proposals per Security Associations = 1
.
Sent 1 packets.
packet IKEv2 INIT sent
IKEv2 packet sent from the Responder was received:
192.168.56.101,SA,KE,Nonce,Notify(MULTIPLE_AUTH_SUPPORTED),VendorID(strongSwan 4.3.6)
Initiator's/my SPI= 6b4961721a23cfc4
Responder's/peer SPI= 270b3ae70426f587
Number of Proposals per Security Associations = 1
.
Sent 1 packets.
IKE_AUTH packet as Initiator was sent
IKE_AUTH pachet sent from the Responder was received

Response received:
###[ IKEv2 Notify ]###
  next_payload= None
  res         = 0
  length      = 8
  proto       = Reserved
  SPIsize     = 0
  type        = AUTHENTICATION_FAILED
  SPI         = ''
  load        = ''

None
Reconfigure iptables to the old state
DONE
```

Reconnaissance?

- *VendorID* **Payload is your friend**
  - If enabled in the configuration.
  - Nothing new here, of course (just a reminder).

```
Payload: Vendor ID (43) : strongSwan
    Next payload: NONE / No Next Payload  (0)
    0... .... = Critical Bit: Not Critical
    .000 0000 = Reserved: 0x00
    Payload length: 20
    Vendor ID: 882fe56d6fd20dbc2251613b2ebe5beb
    Vendor ID: strongSwan
```

```
Payload: Vendor ID (43) : MS NT5 ISAKMPOAKLEY
Payload: Vendor ID (43) : MS-Negotiation Discovery Capable
Payload: Vendor ID (43) : Microsoft Vid-Initial-Contact
Payload: Vendor ID (43) : Unknown Vendor ID
```

Check *ike-scan* and
https://github.com/royhills/ike-scan/blob/master/ike-vendor-ids for more info.

Fingerprinting

- **Different responses in “weird” or not so weird combinations can help a remote attacker to identify its target**
  - What is the limit (if any) on lengthy (e.g. more than 10000 bytes) IKEv2 messages?
  - What is the response in “malformed” packets?
    - “Invalid Syntax”?
    - “No Proposal Chosen”?
    - “Invalid IKE SPI”?
    - “Private Use – Errors”?
    - No response at all?

- **It has been found out that different implementations respond differently.**
- **More on “malformed” or rather unusual IKEv2 chains, later.**

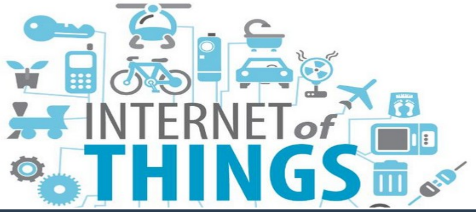
Denial of Service Attack Possibilities & Protections

- **Initiate many *Half-Open IKE-INIT* using different <u>spoofed</u> addresses (IKE is transmitted over UDP):**
  - Responder will have to reserve resources for an amount of time. Legitimate users, one way or another, may not be able to reach the VPN server.
  - Suitable for DDoS attacks.

- **RFC solutions:**
  - "*Cookies*": A simple mechanism introduced to prevent spoofed DoS attacks. The attacker has just to return the Cookie (sent via Notify payloads). [RFC 7296]
  - "*Puzzles*": Make more computationally expensive for an attacker (typically Initiator) to create these half-open IKE-INIT SAs than for the defender to address them. [RFC 8019]

IKEv2 and IoTs

- **Sometimes RFCs solve a problem, and then create a problem for the solution they have provided.**
- **[RFC 7815]: “*The Internet Protocol Suite is increasingly used on small devices with severe constraints on power, memory, and processing resources*”.**
  - Therefore, [RFC 7815] provides a minimal IKEv2 implementation for such devices.



Minimal IKEv2 Initiator Implementation [RFC 7815]

```
Kivinen                       Informational                     [Page 3]

RFC 7815          Minimal IKEv2 Initiator Implementation      March 2016
```

1. Introduction

The Internet Protocol Suite is increasingly used on small devices with severe constraints on power, memory, and processing resources. This document describes a minimal IKEv2 implementation designed for use on such constrained nodes that is interoperable with "Internet Key Exchange Protocol Version 2 (IKEv2)" [RFC7296].

A minimal IKEv2 implementation only supports the initiator end of the protocol. It only supports the initial IKE_SA_INIT and IKE_AUTH exchanges and does not initiate any other exchanges. It also replies with an empty (or error) message to all incoming requests.

This means that most of the optional features of IKEv2 are left out: NAT traversal, IKE SA rekey, Child SA rekey, multiple Child SAs, deleting Child / IKE SAs, Configuration payloads, Extensible Authentication Protocol (EAP) authentication, COOKIEs, etc.

So, if you want to DoS IoT Devices that use IPSec/IKEv2

- **You flood a Responder with half IKEv2-INIT requests by spoofing the address of the devices you want to spoof.**
- **Responder responds with a Cookie, or even worst, with a Puzzle.**
- **IoT devices cannot complete the IKEv2 SA Establishment due to the lack of support of Cookies / Puzzles.**

Other Denial of Service Attack Possibilities

- **When source spoofing is not an option, the best possibility for an attacker is to complete the IKE_INIT and submit an *fake IKE_AUTH* request.**
  - Fake IKE_AUTH = dummy load on the “Encrypted and Authenticate” Payload
    - Cheap for the attacker (i.e. no computations are required)
    - Recipient still has to calculate the Integrity hash to verify the message.

Half-Init & Half-Auth Attacks using yIKEs

- ***./yIKEs.py -d <dst ip addr> -i <iface> -half-init***

  It does not spoof source address

  => Typically "blocked" after few attempts

- ***./yIKEs.py -d <dst ip addr> -i <iface> -half-init -sub <subnet> -rand***

  => Randomise (spoofs) source address from a given subnet and triggers "Cookies"

  => If in the same LAN, it responds to ARP and performs Half-Auth attack (with "dummy" encrypted payload) – proof of concept.

IKE Half-Auth Attack- PoC

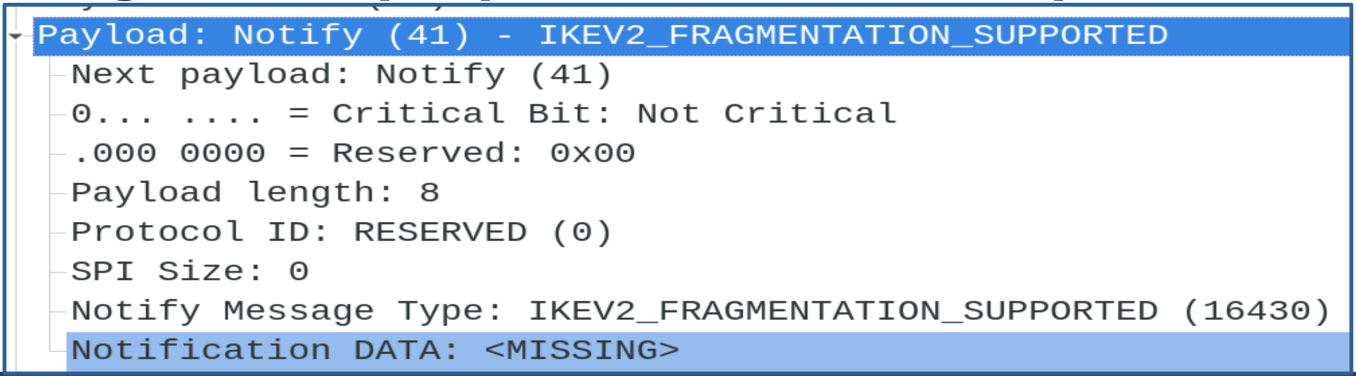

```
./yIKEs.py -d 192.168.56.101 -i vboxnet0 -half-init -sub 192.168.56.0/24 -rand -stimeout 1000
```

```
top - 18:08:11 up  8:14,  1 user,  load average: 0.88, 0.73, 0.37
Tasks:  93 total,   2 running,  91 sleeping,   0 stopped,   0 zombie
%Cpu(s): 25.6 us, 24.4 sy,  0.0 ni, 45.5 id,  0.0 wa,  0.0 hi,  4.5 si,  0.0 st
KiB Mem :   498728 total,     6924 free,   124024 used,   367780 buff/cache
KiB Swap:   839676 total,   836852 free,     2824 used.   343228 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 1087 root      20   0  955504   8732   1652 S 63.0  1.8   2:17.03 charon
```

IKEv2 Fragmentation [RFC 7383]

- **Fragmentation at IKEv2 level**
  - To avoid IP fragmentation (due to potential dropping of IP fragments)
- **Only IKE_SA_AUTH messages can be IKE-fragmented**
- **A *Notify type=16430* message denotes IKEv2 fragmentation capabilities**
- **Combination of IP and IKEv2 fragments may not make sense for legitimate purposes, but it is not prevented.**



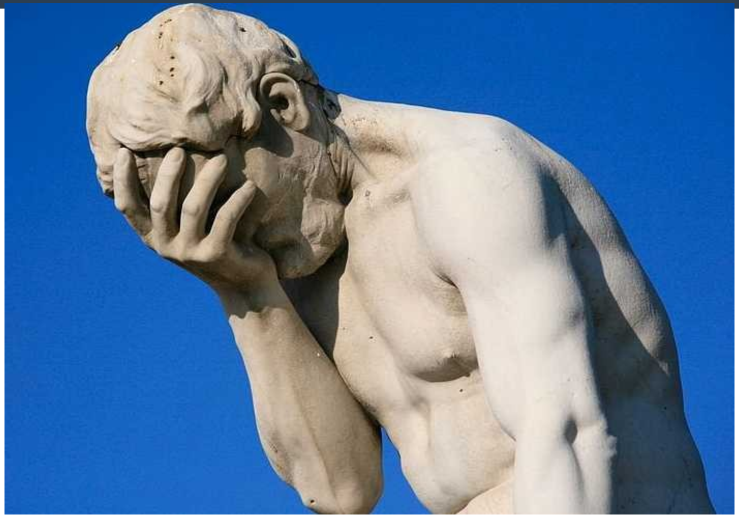
```
Payload: Notify (41) - IKEV2_FRAGMENTATION_SUPPORTED
  Next payload: Notify (41)
  0... .... = Critical Bit: Not Critical
  .000 0000 = Reserved: 0x00
  Payload length: 8
  Protocol ID: RESERVED (0)
  SPI Size: 0
  Notify Message Type: IKEV2_FRAGMENTATION_SUPPORTED (16430)
  Notification DATA: <MISSING>
```

IKEv2 Encrypted Fragment Payload

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Fragment Number        |        Total Fragments        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Initialization Vector                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                      Encrypted content                        ~
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               |             Padding (0-255 octets)            |
+-+-+-+-+-+-+-+-+                               +-+-+-+-+-+-+-+-+
|                                               |  Pad Length   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Integrity Checksum Data                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Source: IETF RFC 7383

IKEv2 Fragmentation Attacks



- **Incomplete fragments**
  - Fill-up target's memory
- **Create chains > 65535 bytes**
  - Theoretically unlimited
- **Fragmentation overlapping?**
  - Not partially (i.e. no offset in fragments)
  - Only duplicated fragments
    - Still rather not an option.

yIKEs - Fragmentation

-fr **The number of IKEv2 fragments > 0 to be used for IKEv2 fragmentation (in IKE_AUTH messages).**

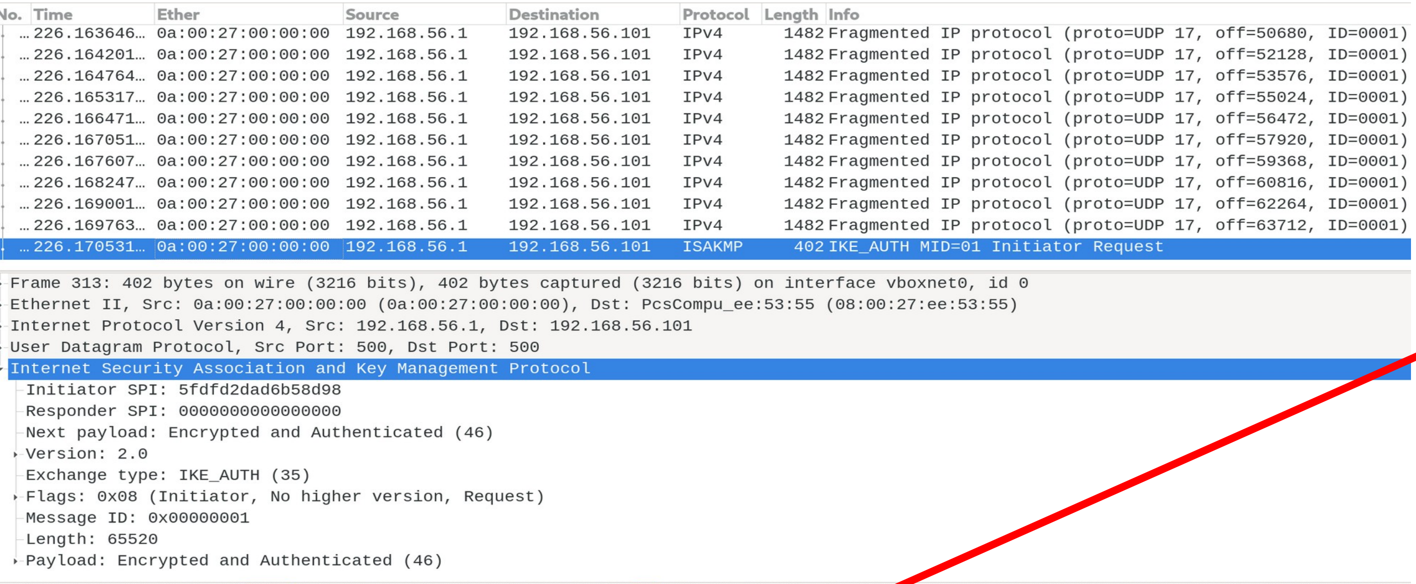
-ifr **The last fragment is not sent**

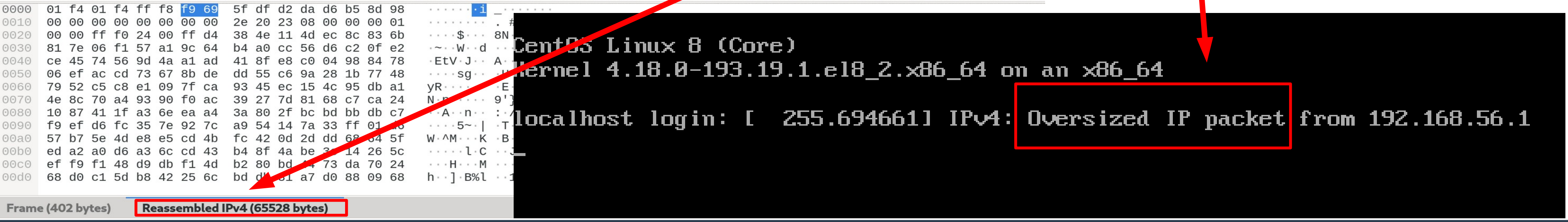
Try to fill-up target’s memory with many huge but incomplete fragments.

==> imagine multiple (spoofed) senders.

What is the Difference between IP fragmentation and IKEv2 fragmentation?

Oversized (i.e. near the limit of an IP datagram).





What is the Limit of IKEv2 Fragmentation?

- You can have IKEv2 65535 fragments
- With an Ethernet MTU (1480 bytes) you can have an IKE AUTH packet bigger than 91 million (!) bytes – if you can construct it.
- To make matter worst, you can combine it with IP fragmentation.

Any IKEv2 Official Length Limitations?

- IETF RFC 7296:
  - “All IKEv2 implementations MUST be able to send, receive, and process IKE messages that are up to 1280 octets long, and they SHOULD be able to send, receive, and process messages that are up to 3000 octets long”.
- In practice:

– Several implementations allow IKEv2 packets much bigger than these.

IKEv2 Fragmentation – Oversized Example

No.	Time	Ether	Source	Destination	Protocol	Length	Info
…	11.7492542…	0a:00:27:00:00:00	192.168.56.1	192.168.56.101	ISAKMP	926	IKE_AUTH MID=01 Initiator Request (fragment 152/161)
…	11.7587843…	0a:00:27:00:00:00	192.168.56.1	192.168.56.101	ISAKMP	926	IKE_AUTH MID=01 Initiator Request (fragment 153/161)
…	11.7784850…	0a:00:27:00:00:00	192.168.56.1	192.168.56.101	ISAKMP	926	IKE_AUTH MID=01 Initiator Request (fragment 154/161)
…	11.7953822…	0a:00:27:00:00:00	192.168.56.1	192.168.56.101	ISAKMP	926	IKE_AUTH MID=01 Initiator Request (fragment 155/161)
…	11.8111603…	0a:00:27:00:00:00	192.168.56.1	192.168.56.101	ISAKMP	926	IKE_AUTH MID=01 Initiator Request (fragment 156/161)
…	11.8314586…	0a:00:27:00:00:00	192.168.56.1	192.168.56.101	ISAKMP	926	IKE_AUTH MID=01 Initiator Request (fragment 157/161)
…	11.8460920…	0a:00:27:00:00:00	192.168.56.1	192.168.56.101	ISAKMP	926	IKE_AUTH MID=01 Initiator Request (fragment 158/161)
…	11.8662706…	0a:00:27:00:00:00	192.168.56.1	192.168.56.101	ISAKMP	926	IKE_AUTH MID=01 Initiator Request (fragment 159/161)
…	11.8852426…	0a:00:27:00:00:00	192.168.56.1	192.168.56.101	ISAKMP	926	IKE_AUTH MID=01 Initiator Request (fragment 160/161)
…	11.8975779…	0a:00:27:00:00:00	192.168.56.1	192.168.56.101	ISAKMP	270	IKE_AUTH MID=01 Initiator Request (fragment 161/161)

- **926 bytes x 160 fragments + 250 = 148340 bytes >> 65535 bytes**
- **No protection or alert from underlying Operation system, since the IP datagram itself never reaches the limit.**
- **It is on the specific implementation only.**

Examples of IKEv2 Fragmentation Related CVEs

NIST | NVD MENU

Information Technology Laboratory

NATIONAL VULNERABILITY DATABASE

NVD

VULNERABILITIES | SEARCH AND STATISTICS

Vuln ID	Summary	CVSS Severity
CVE-2016-6381	Cisco IOS 12.4 and 15.0 through 15.6 and IOS XE 3.1 through 3.18 and 16.1 allow remote attackers to cause a denial of service (memory consumption or device reload) via fragmented IKEv1 packets, aka Bug ID CSCuy47382. **Published:** October 05, 2016; 1:59:00 PM -0400	*V3.0:* 7.5 HIGH *V2.0:* 7.1 HIGH
CVE-2016-1344	The IKEv2 implementation in Cisco IOS 15.0 through 15.6 and IOS XE 3.3 through 3.17 allows remote attackers to cause a denial of service (device reload) via fragmented packets, aka Bug ID CSCux38417. **Published:** March 25, 2016; 9:59:01 PM -0400	*V3.0:* 5.9 MEDIUM *V2.0:* 7.1 HIGH
CVE-2013-6076	strongSwan 5.0.2 through 5.1.0 allows remote attackers to cause a denial of service (NULL pointer dereference and charon daemon crash) via a crafted IKEv1 fragmentation packet. **Published:** November 02, 2013; 2:55:03 PM -0400	*V3.x:*(not available) *V2.0:* 5.0 MEDIUM

What (else) Could Someone Potentially Abuse?

- **Each exchange may have:**
  - Several different types of payloads
  - Many payloads of one type (not necessarily acceptable for all types).
  - Different sizes for some types of payloads
    - Some of them can become <u>extremely</u> big (e.g. Notify or Certificate related payloads).
  - Some Payloads (eg SA) have have their internal, potentially unlimited, chain.
    - Many Proposals, each one having many Transforms, each one having some Attributes.
- **There is no pre-defined order, or strict number of occurrences (even if e.g. having two SAs may not make sense).**
  - Payloads may be repeated

Supported IKEv2 Payload Identifiers

• SA	Security Association
• KE	Key Exchange
• Nonce	Nonce Payload
• CERTREQ	Certificate Request Payload
• CERT	Certificate Payload
• IDi	Identification Payload (Initiator)
• IDr	Identification Payload (Responder)
• TSi	Traffic Selector (Initiator)
• TSr	Traffic Selector (Responder)
• AUTH	Authentication Payload
• Notify	Notify Payload

Constructing Arbitrary Payload Chains

black hat EUROPE 2020

- **In IKE_INIT**

./yIKEs.py -i <iface> ***-d <IP address of destination> -recon***

-ip <comma separated list of IKEv2 identifier payloads>

Example:

./yIKEs.py -i vboxet0 ***-d 192.168.56.101 -recon   -ip SA,KE,Notify.16380,Nonce,Notify.16388-1639***

comma-separated list of Identifier Payloads

Notify Types separated with dots (.)

Range of Notify Types

A special case: SA Payload

- **For IKE_INIT**
  - Defining Transforms in a Proposal

    *-**pr** 1.12,3.12,2.5,4.2*
  - Many Transforms in a Proposal:

    *-pr 1.12**-**14,3.12,2.5,4.2*
  - Many Proposals

    -pr 1.12-14,3.12,2.5,4.2/1.16,3.14,2.5,4.2
- **For IKE_AUTH: Just use -*pr2* (same syntax)**

TYPES OF TRANSFORMS (EXAMPLES)

Encryption: 1		**Integrity**: 3	
AES-CBC	12	HMAC-SHA1-96	2
AES-CTR	13	SHA2-256-128	12
Camellia-CBC	23	SHA2-256-128	14
PRF: 2		**GroupDesc**: 4	
PRF_HMAC_SHA2_256	5	1024MODPgr	2
PRF_HMAC_SHA2_384	6	2048MODPgr	14
Extended Sequence Number: 5			
No ESN	0		
ESN	1		

Example: 1.12 => AEC-CBC,
3.14 => SHA2-256-128
etc.

For a complete list, check:
https://www.iana.org/assignments/ikev2-parameters/ikev2-parameters.xhtml#ikev2-parameters-3

Potentially Malformed Payload Chains


```
[root@linux IKEv2]# ./yIKEs.py -d 192.168.56.101 -i vboxnet0 -recon -ip SA,KE,Notify.16450,Nonce,SA
Number of Proposals per Security Associations = 1
.
Sent 1 packets.
Response received:
192.168.56.101,Notify(INVALID_SYNTAX)
packet IKEv2 INIT sent
Reconfigure iptables to the old state
DONE
```

2 SAs in one IKE_INIT => “Invalid Syntax” for StrongSwan

Same for KE, Nonce for StrongSwan

- **This is not the case for Windows 2019 Servers**

  => happily respond to messages with several SA, KE, and Nonce payloads (> 140).

Search Results (Refine Search)

Sort results by: Publish Date Descending Sort

Search Parameters:

- Results Type: Overview
- Keyword (text search): IKEv2

There are **40** matching records.
Displaying matches **1** through **20**.

1 2 > >>

Vuln ID	Summary	CVSS Severity
CVE-2020-3230	A vulnerability in the Internet Key Exchange Version 2 (IKEv2) implementation in Cisco IOS Software and Cisco IOS XE Software could allow an unauthenticated, remote attacker to prevent IKEv2 from establishing new security associations. The vulnerability is due to incorrect handling of crafted IKEv2 SA-Init packets. An attacker could exploit this vulnerability by sending crafted IKEv2 SA-Init packets to the affected device. An exploit could allow the attacker to cause the affected device to reach the maximum incoming negotiation limits and prevent further IKEv2 security associations from being formed. **Published:** June 03, 2020; 2:15:20 PM -0400	*V3.1:* 7.5 HIGH *V2.0:* 5.0 MEDIUM
CVE-2019-12312	In Libreswan 3.27 an assertion failure can lead to a pluto IKE daemon restart. An attacker can trigger a NULL pointer dereference by initiating an IKEv2 IKE_SA_INIT exchange, followed by a bogus INFORMATIONAL exchange instead of the normallly expected IKE_AUTH exchange. This affects send_v2N_spi_response_from_state() in programs/pluto/ikev2_send.c that will then trigger a NULL pointer dereference leading to a restart of libreswan. **Published:** May 24, 2019; 10:29:00 AM -0400	*V3.0:* 7.5 HIGH *V2.0:* 5.0 MEDIUM
CVE-2017-17157	IKEv2 in Huawei IPS Module V500R001C00, V500R001C00SPC200, V500R001C00SPC300, V500R001C00SPC500, V500R001C00SPH303, V500R001C00SPH508, V500R001C20, V500R001C20SPC100, V500R001C20SPC100PWE, V500R001C20SPC200, V500R001C20SPC200B062, V500R001C20SPC300B078, V500R001C20SPC300PWE has an out-of-bounds memory access vulnerability due to insufficient input validation. An attacker could exploit it to craft special packets to trigger out-of-bounds memory access, which may further lead to system exceptions. **Published:** February 15, 2018; 11:29:01 AM -0500	*V3.0:* 7.5 HIGH *V2.0:* 5.0 MEDIUM

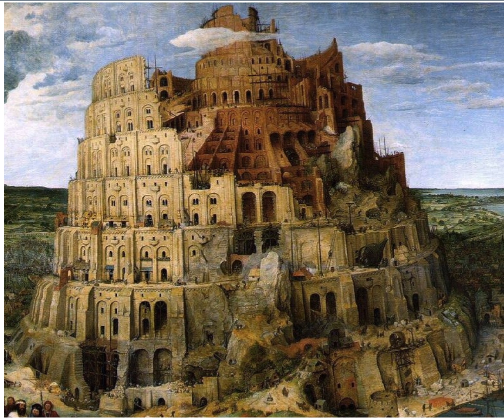
In Summary

- **IKEv2 has been simplified significantly, which leaves less room for potential exploitation.**
- **However, the chains that can be constructed using the various payloads and literally endless combinations still leave room for potential abuse.**
- **RFCs do not always help in the prevention of such attacks because:**
  - They do not enforce strict measures and behaviours in rather unnecessary for real world cases, hence leaving this to vendors' understanding of the various attacking scenarios.
  - Some times they "contradict" each other.

Now you can perform your own assessments using yIKEs:

- yIKEs is released today as open-source at https://github.com/aatlasis

Some Final Thoughts



- **RFCs still written following the "Robustness principle" philosophy:**

  "Be conservative in what you send, be liberal in what you accept from others"

  ==> Good for interoperability purposes, but for security?

  ==> same story is repeated in several protocols (e.g. see IPv6)

- **There have been efforts for a change in IETF community.**
- **Time for a change?**

The Way is Shown by:

INFORMATIONAL
Errata Exist

Network Working Group — R. Callon, Editor
Request for Comments: 1925 — IOOF
Category: Informational — 1 April 1996

The Twelve Networking Truths

(12) In protocol design, perfection has been reached not when there is nothing left to add, but when there is nothing left to take away.

Questions?

@AntoniosAtlasis

References

- Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <https://tools.ietf.org/html/rfc7296>.
- Smyslov, V., "Internet Key Exchange Protocol Version 2 (IKEv2) Message Fragmentation", RFC 7383, DOI 10.17487/RFC7383, November 2014, <https://www.rfc-editor.org/info/rfc7383>.
- Nir, Y. and V. Smyslov, "Protecting Internet Key Exchange Protocol Version 2 (IKEv2) Implementations from Distributed Denial-of-Service Attacks", RFC 8019, DOI 10.17487/RFC8019, November 2016, <https://www.rfc-editor.org/info/rfc8019>.
- Kivinen, T., "Minimal Internet Key Exchange Version 2 (IKEv2) Initiator Implementation", RFC 7815, DOI 10.17487/RFC7815, March 2016, < https://www.rfc-editor.org/info/rfc7815>.

yIKEs

An open-source tool for
IKEv2 security assessment

yIKEs - Introduction

black hat EUROPE 2020

- **Basic parameters:**

  -i <INTERFACE> The interface to use

  -d <IP> The IPv4 address of the target.

  -p <port> The UDP port of the target (default: 500).

  -sp <port> The source UDP port (default 500).

  -stimeout The time to sniff when in listen mode, in seconds (default: 10).

yIKEs – Modes of Operation

-recon Send an INIT packet only and print results of the Respone only => Initiator.

-listen Listen for INIT packets, print results and respond with AUTH => Responder.

-recon -listen Send INIT packet, listens for INIT response, send AUTH packet => Initiator.

(AUTH as responder not supported yet).

-half-init Initiates a half-open INIT attack; responds to cookies; it also responds with fake AUTH.

Constructing Arbitrary Payload Chains (2)

- **In IKE_AUTH**

./yIKEs.py -i <iface> ***-d <IP address> -recon -listen***

-ip2 <comma separated list of IKEv2 identifier payloads>

Example:

./yIKEs.py -i vboxnet0 ***-d 192.168.56.101 -recon -ip2 IDi,Notify.16384,IDr,AUTH,TSi,TSr***

Different comma-separated list of Identifier Payloads

Automatically Encrypted and put in “Encrypted” Payload