Internet Security

Tutorato

Davide Carnemolla

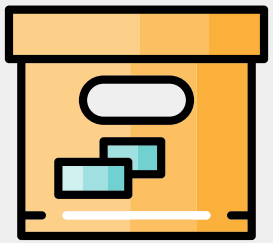
Dipartimento di Matematica e Informatica
Università degli Studi di Catania

A.A. 2021/2022

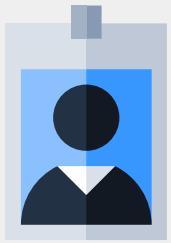
Proprietà di sicurezza

Proprietà di 1° livello

Confidenzialità
(Segretezza)



Integrità



Autenticazione

Proprietà di 1° livello: Confidenzialità

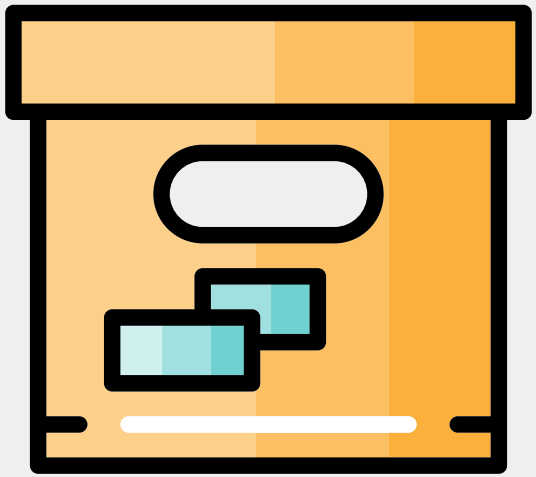
Definizione

L'informazione non sia rilasciata ad entità non autorizzate a conoscerla

Garantita da

- **Crittografia**
- **Steganografia**

Proprietà di 1° livello: Integrità



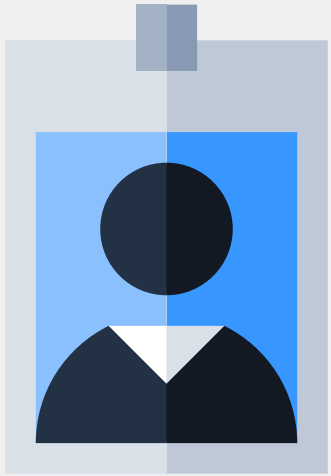
Definizione

L'informazione non sia modificata da entità non autorizzate

Garantita da

- **Firma elettronica**

Proprietà di 1° livello: Autenticazione



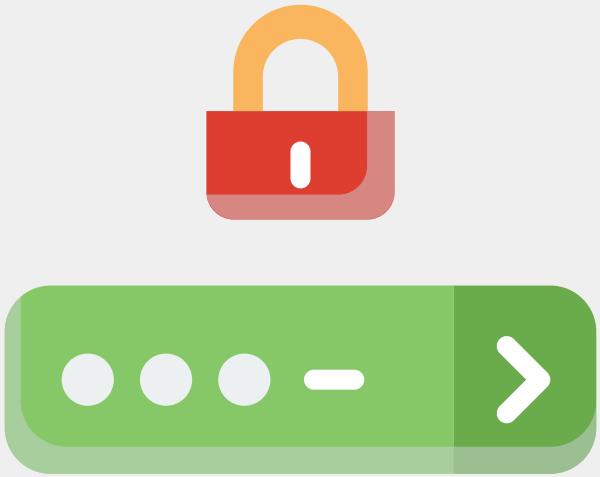
Definizione

Le entità siano esattamente chi dichiarano di essere

Garantità da

- **Conoscenza** (password, PIN)
- **Possesso** (Smart card, Smart Token)
- **Biometria** (impronte, iride)

Autenticazione basata su conoscenza



Come funziona

La conoscenza di un segreto (password, PIN) comprova l'identità dell'utente

Problemi

Sensibile ad attacchi di **guessing**, **snooping**, **spoofing**, **sniffing**

Come viene salvata una password?

- Memorizzate **in chiaro** su un file di sistema (CTS, 1960)
- Uso di funzioni hash crittografiche
- GNU/Linux

Password: GNU/Linux

cat /etc/shadow | grep root

root:1Etg2ExUZ$F9NTP7omafhKIlqaBMqng1:15651:0:99999:7:::

Analisi dell'output

- **1** indica l'utilizzo di una funzione hash MD5
- **Etg2ExUZ** è il sale
- **F9NTP7omafhKIlqaBMqng1** è il segreto
- **15651** è la data in cui è stata impostata la password
- **0** sono i giorni che devono trascorrere prima di poter modificare la password
- **99999** sono i giorni dopo cui bisogna modificare la password
- **7** sono i giorni dopo l'utente viene disabilitato

Autenticazione basata su possesso

Come funziona

Il possesso di un dispositivo fisico (Smart Card, Smart Token, YubiKey) comprova l'identità dell'utente

Caratteristiche

- Interamente leggibile nel caso di carte magnetiche
- Estrazione dei segreti gestita da un'interfaccia funzionale nel caso di carte elettroniche

Autenticazione basata su biometria

Come funziona

Il possesso di caratteristiche biometriche (impronte digitali, impronta della retina, viso etc.) comprova l'identità

Caratteristiche

- Meno accurato ma più affidabile
- Utilizzato in combinazione con autenticazione basata su conoscenza

Proprietà di 2° livello

Non ripudio

Disponibilità

Non ripudio

Definizione

L'entità non possa negare la propria partecipazione ad una transazione con uno specifico ruolo

Garantita da

- Protocolli di sicurezza appositi (ad esempio **PEC**)

Disponibilità

Definizione

Il sistema sia operante e funzionante in ogni momento

Garantita da

- Autenticazione
- Accesso complicato impegnando il chiamante computazionalmente

Crittografia

Crittologia

Il termine **Crittologia** deriva dal greco *kryptòs* (nascosto) e *logos* (discorso) ed è la scienza che si occupa delle scritture nascoste. Comprende la **Crittografia** e la **Crittoanalisi**.

Crittografia

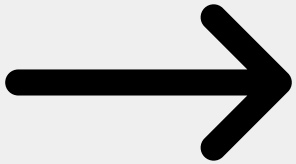
La **Crittografia** è la scienza che si occupa di costruire dei metodi per rendere un messaggio intelligibile.

Crittoanalisi

La **Crittoanalisi** è la scienza che si occupa di costruire metodi per "rompere" gli schemi crittografici

Crittografia Simmetrica

Bob



Alice

$c = \mathcal{E}(m, k)$

$m = \mathcal{D}(c, k)$

Perfetta sicurezza

Definizione

Sia $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ uno schema di cifratura simmetrico. Diremo che $\mathcal{SE}$ è *perfettamente sicuro* se

$$\forall M_1, M_2 \in \mathcal{M} \text{ e } \forall c \in \mathcal{C} \qquad Pr[\mathcal{E}_{\mathcal{K}}(M_1) = c] = Pr[\mathcal{E}_{\mathcal{K}}(M_2) = c]$$

One-Time Pad (OTP)

Specifiche

1. $\mathcal{M} = \{0,1\}^m,\ m \in \mathbb{N} : m > 0$
2. $\mathcal{K} \xleftarrow{\$} \{0,1\}^m$
3. $\mathcal{E}_{\mathcal{K}}(M) = \mathcal{K} \oplus M$
4. $\mathcal{D}_{\mathcal{K}}(C) = C \oplus \mathcal{K}$

Condizione per la perfetta sicurezza

OTP è perfettamente sicuro sotto l'ipotesi che la chiave venga utilizzata per cifrare un singolo messaggio.

One-Time Pad: implementazione (1)

```
import random

def generate_key(m):
    return bytes(
        random.randrange(0,256) for i in range(m)
    )

def xor_bytes(key, message):
    m = min(len(key), len(message))
    return bytes(
        [key[i] ^ message[i] for i in range(m)]
    )
```

One-Time Pad: implementazione (2)

```
message = "OTP is perfect for a single message"
message = message.encode()
key = generate_key(len(message))
cipher = xor_bytes(key, message)
print(key)
print(cipher)
print(xor_bytes(key, cipher))
```

One-Time Pad: attacco

```
message2 = "I don't need another key"
message2 = message.encode()
cipher2 = xor_bytes(key, message2)
print(cipher2)
```

Attacco

- $c_1 \oplus c_2 = m_1 \oplus k \oplus m_2 \oplus k = m_1 \oplus m_2$
- Se m_1 è noto posso ottenere $m_2 = c_1 \oplus c_2 \oplus m_1$

Funzione Hash

Una *funzione hash* è una funzione matematica con le seguenti proprietà:

- prende in input una stringa (di bit) di qualsiasi dimensione
- restituisce in output una stringa (di bit) di dimensione fissata
- è efficiente dal punto di vista computazionale

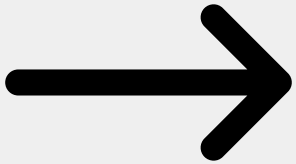
Funzione Hash Crittografica

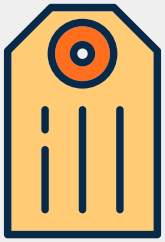
Una *funzione hash crittografica* è una funzione hash con le seguenti proprietà:

- **Collision Resistance**
- **Hiding**
- **Puzzle friendliness**

Message Authentication Code (MAC)

Bob





Alice

$m, T = MAC_k(m)$

$VF_k(m, T')$

https://cryptohack.org

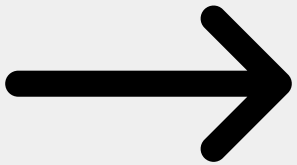
Come possiamo effettuare lo scambio di una **chiave**?

Crittografia Asimmetrica

Bob

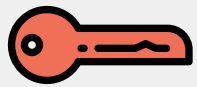
$c = \mathcal{E}(m, pk)$

pk



Alice

$m = \mathcal{D}(c, sk)$



sk

Firme digitali

"La firma digitale è
l'inverso della cifratura"

Anonimo

Firme digitali

"La firma digitale è
l'inverso della cifratura"

Anonimo

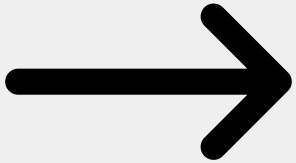
Firme digitale

Bob

pk

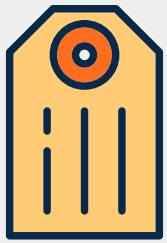
Alice

$m, \sigma = Sign(m, sk)$



$VF(pk, m, \sigma)$

sk

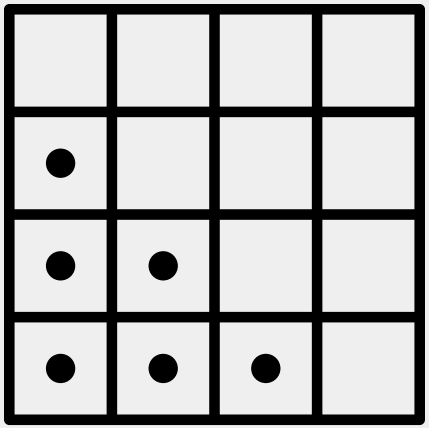


Steganografia

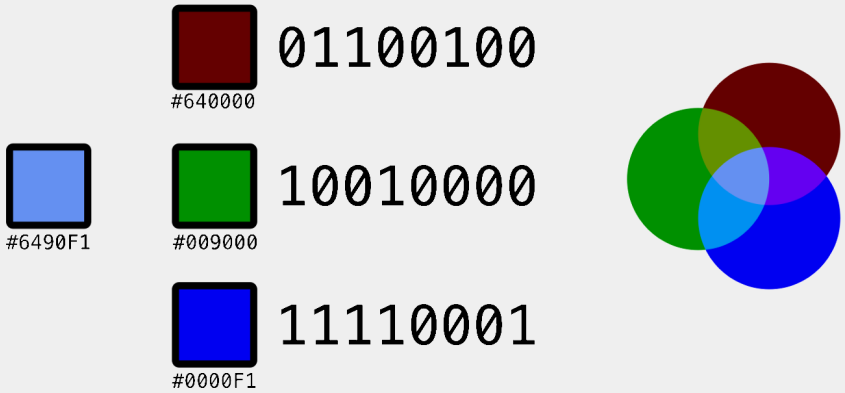
Steganografia

La steganografia è una tecnica che si prefigge di nascondere la comunicazione tra due interlocutori.

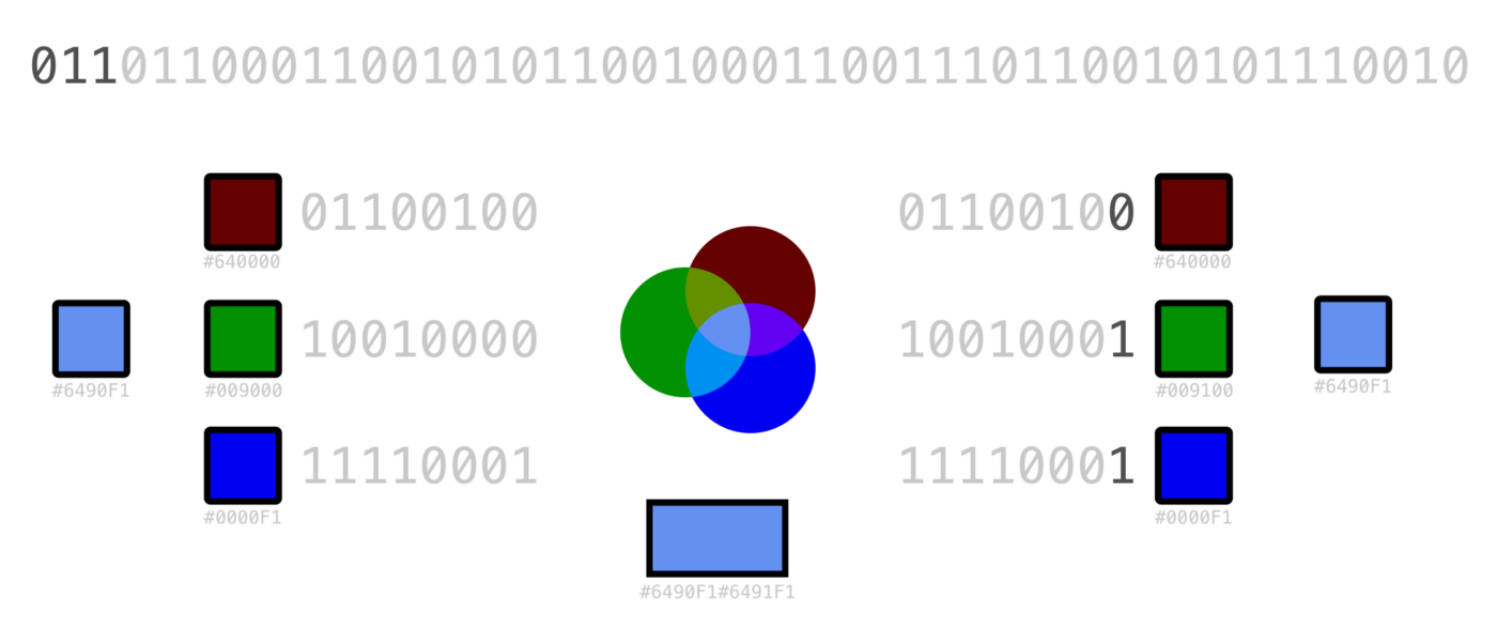
Least Significant Bit (LSB)



Immagine

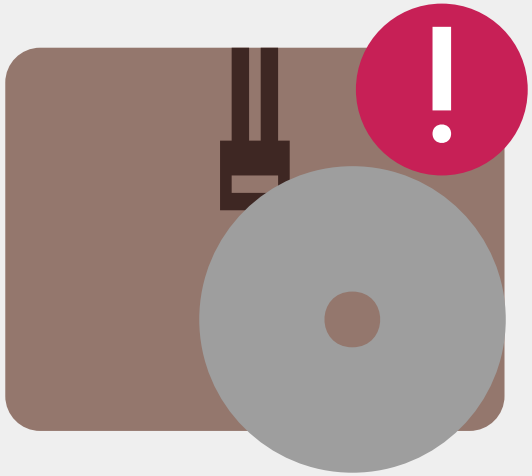


Least Significant Bit (LSB)



Classificazione software nocivi

Software nocivo



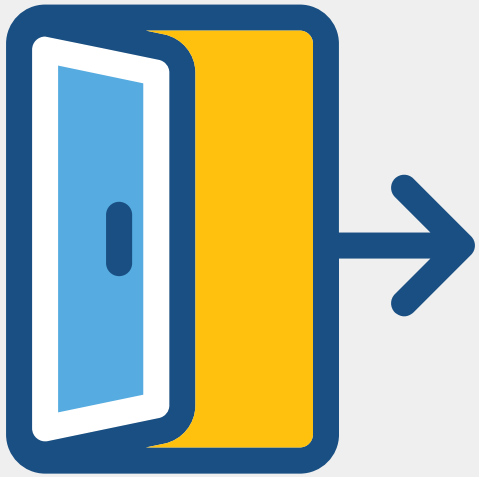
Definizione

Software scritto con l'esplicito scopo di violare alcune proprietà di sicurezza di un sistema.

Caratteristiche

- **Carico**
- **Propagazione**

Trapdoor



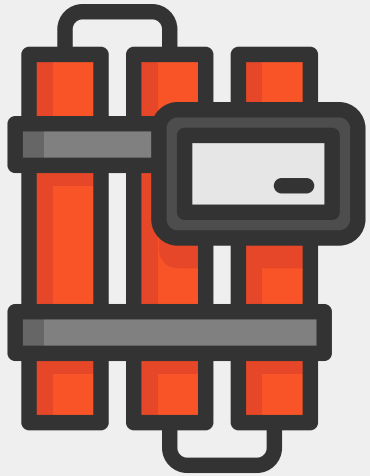
Definizione

Punto d'accesso segreto per bypassare l'autenticazione in un sistema.

Nota

Tipicamente le trapdoor vengono inserite dagli sviluppatori per testare il software.

Bomba logica



Definizione

Porzione di codice di un software nocivo apparentemente innocua fino al verificarsi di particolari condizioni.

Cavallo di Troia

Definizione

Software utile che in fase di esecuzione compie violazioni di sicurezza.

Zombie

Definizione

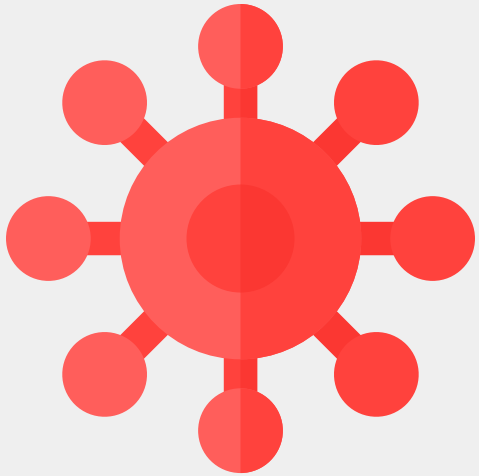
Software che sfrutta una macchina remota già violata per lanciare nuovi attacchi.

Worm

Definizione

Software nocivo che infetta macchine remote, ciascuna delle quali a loro volta infetta altre macchine remote.

Virus



Definizione

Software nocivo che viola altri programmi non nocivi, sfruttandoli per propagarsi.

Break: Hack The Box

https://app.hackthebox.com

Firewall

Firewall

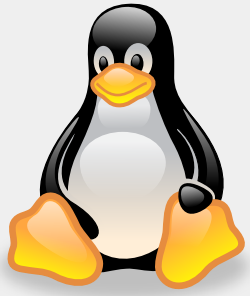
Definizione

Un firewall è un componente software o hardware di difesa perimetrale di una rete.

Funzionalità

- Protegge le risorse interne
- Monitora il traffico
- Filtra i dati

Firewall - esempi



GNU/Linux

iptables
shorewall
FirewallD
ufw

Windows

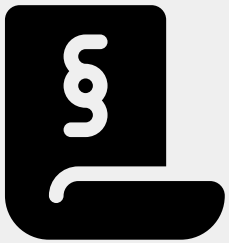
Windows Defender
GlassWire
Norton
Comodo

Mac Os

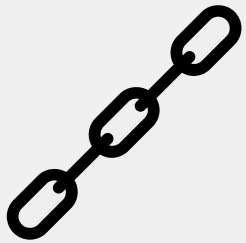
Apple Firewall
Total AV
Avira
Bitdefender

Informazioni

Iptables è un firewall per i sistemi GNU/Linux implementato a livello kernel (Netfilter).



Regole

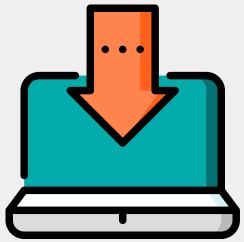


Catene

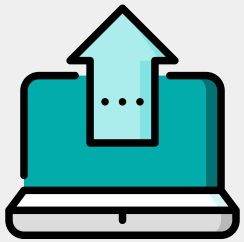
Nota

iptables è stato sostituito da nftables in Debian a partire dalla versione 11 (Buster).

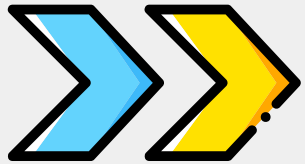
IPTABLES: CATENE DI DEFAULT



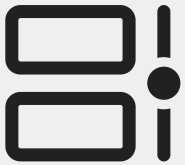
INPUT



OUTPUT



FORWARD



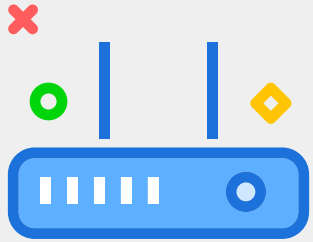
PREROUTING

POSTROUTING

IPTABLES: TABELLE

filter

INPUT
OUTPUT
FORWARD



nat

OUTPUT
PREROUTING
POSTROUTING

mangle

IPTABLES: COMANDI UTILI

Visualizzare le regole

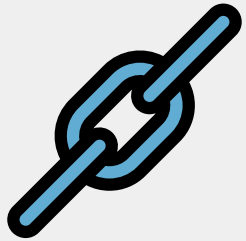
- iptables -t <table> -L

Un po' di pulizia

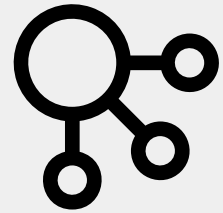
- iptables -F # elimina tutte le regole
- iptables -X # elimina tutte le catene personalizzate
- iptables -t nat -F # elimina tutte le regole di nat

IPTABLES: STATI DELLE CONNESSIONI

NEW



ESTABLISHED



RELATED

Accettiamo le connessioni ESTABLISHED e RELATED

iptables -A INPUT -m state –state ESTABLISHED,RELATED -j ACCEPT

IPTABLES: SSH (1)

Accettiamo le connessioni SSH

iptables -A INPUT -p tcp –dport 22 -m state –state NEW -j ACCEPT

Analisi

- **-A** aggiunge la regola in coda
- **INPUT** la catena a cui la regola fa riferimento
- **-p** indica il protocollo
- **-dport** indica la porta di destinazione
- **-m state -state NEW** indica di accettare le connessioni esterne in ingresso
- **-j** indica il target

IPTABLES: SSH (2)

Policy (default)

```
iptables -P INPUT DROP
```

Analisi

- **-P**: Policy
- **INPUT**: Chain
- **DROP**: Target

IPTABLES: SSH (3)

Una regola più precisa (address source)

iptables -A INPUT -p tcp -s 192.168.1.2 –dport 22 -j ACCEPT

Una regola più precisa (interface)

iptables -A INPUT -p tcp -i etho –dport 22 -j ACCEPT

Get your hands dirty!

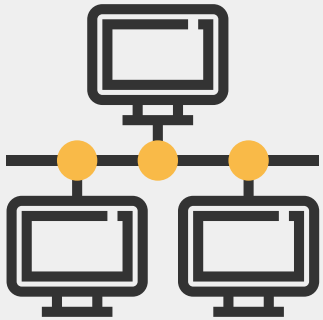
Intrusion Detection System

Intrusion Detection System

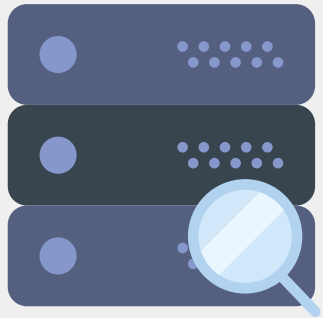
Definizione

Un Intrusion Detection System (IDS) è un dispositivo software o hardware per identificare accessi non autorizzati alla rete locale o alle macchine host.

IDS: Classificazione



NIDS



HIDS

Hybrid IDS

IDS: Tecniche di rilevamento

Signature based

Statistical anomaly-based

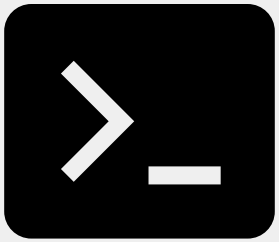
Stateful protocol analysis

IDS: da cosa è composto?

Sensors/Agents

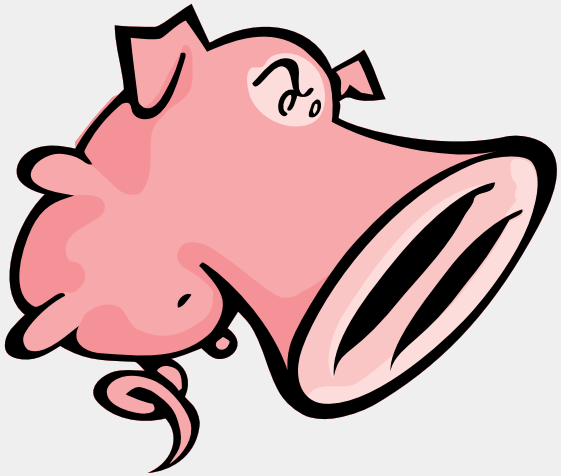
Management Server

Database Server



Console

IDS: esempi



Snort

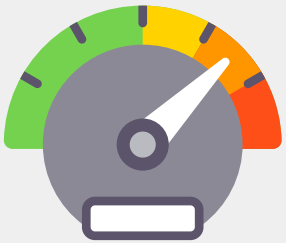
Suricata

Suricata

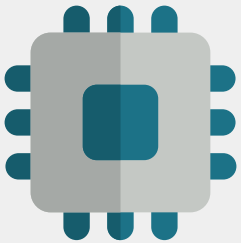
Definizione

Suricata è un intrusion prevention/detection system sviluppato dalla Open Information Security Foundation sotto licenza open source.

Suricata: caratteristiche

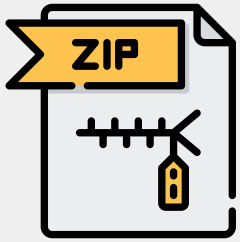


Multi-Threaded

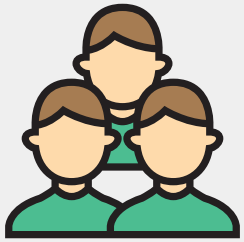


Built in Hardware Acceleration

LuaJIT



File Extraction



Great Community

Multipurpose Engine

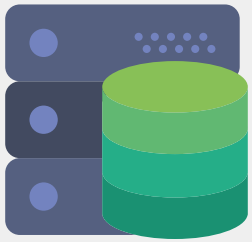
OAuth 2.0

Oauth 2.0

OAuth 2.0 è un protocollo standard aperto per l'autorizzazione.

OAuth 2.0: Roles

Client

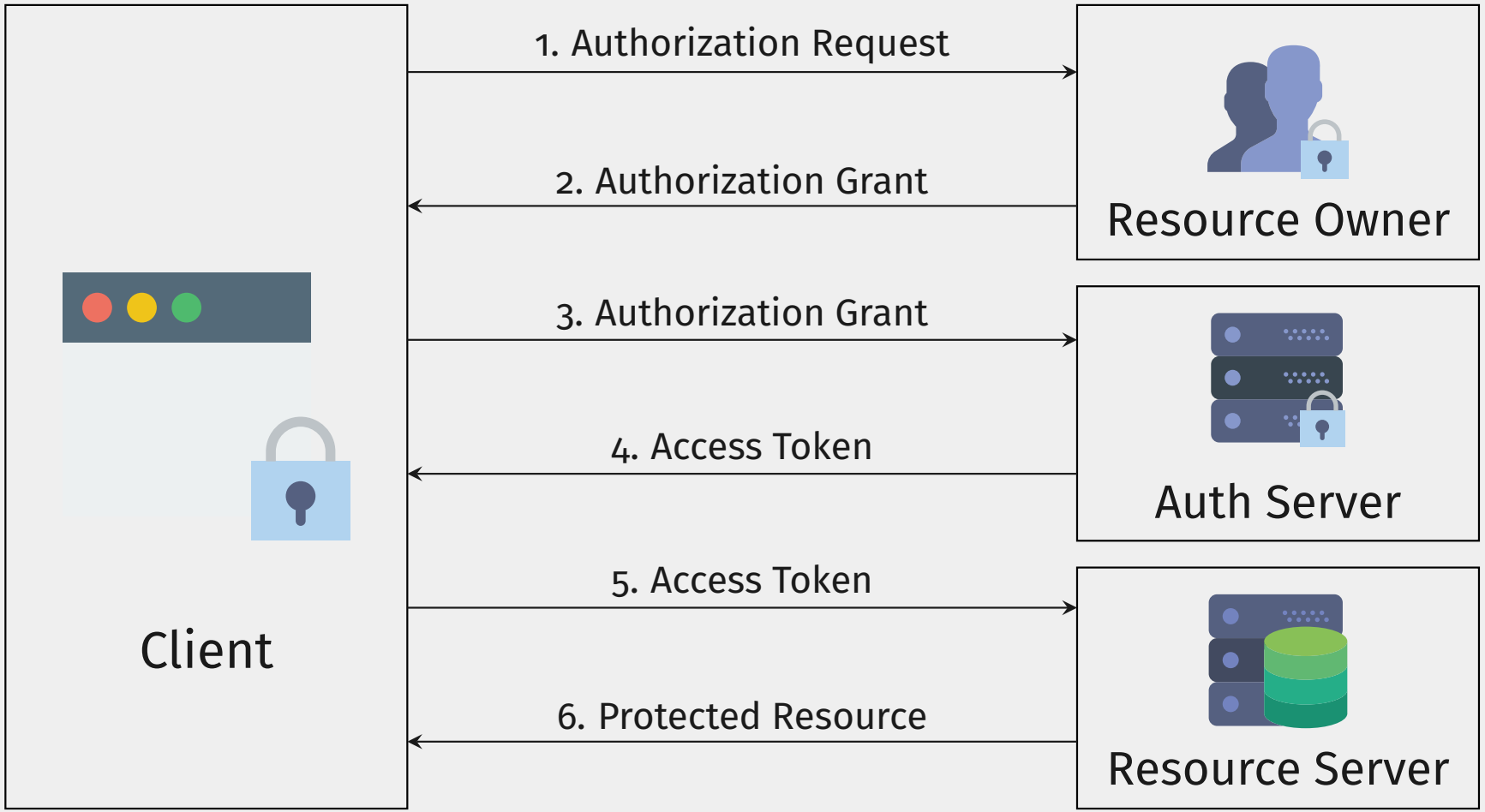


Resource Server

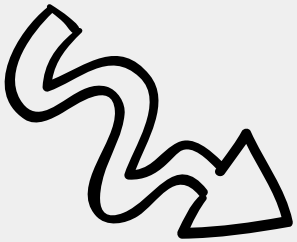
Authorization Server

Resource Owner

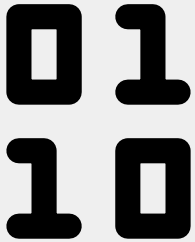
OAuth 2.0: Protocol Flow



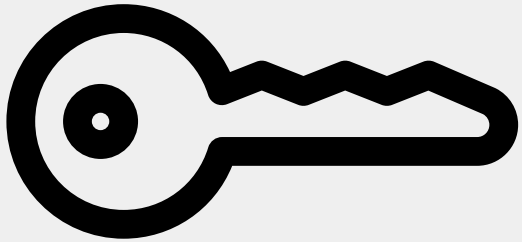
OAuth 2.0: Creating an App



Redirect URIs



Client ID

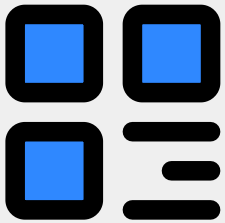


Secret

OAuth 2.0: Grant Type

Authorization Code

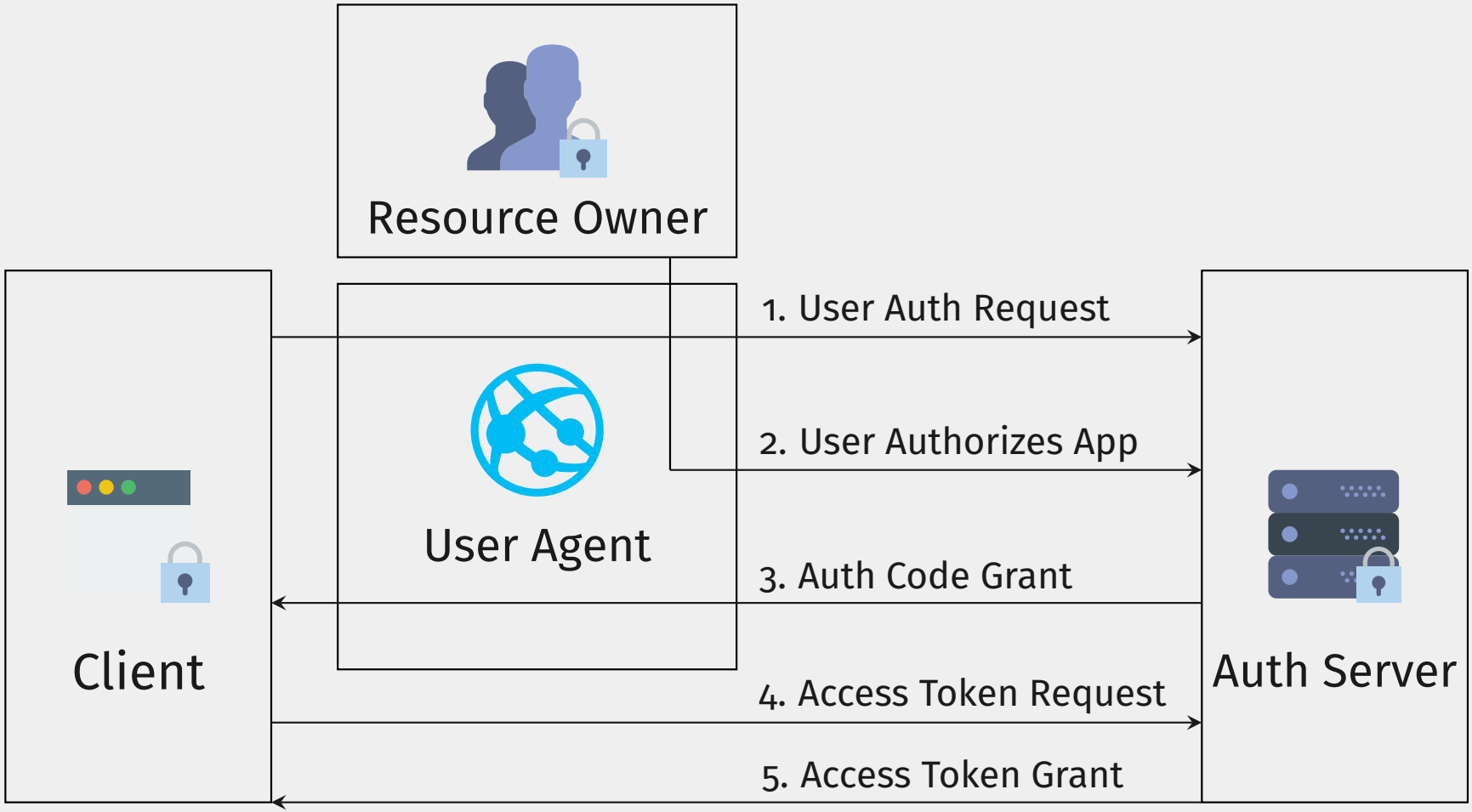
Password



Client Credential

Implicit

OAuth 2.0: Authorization Code Flow



OAuth 2.0: Authorization Code Flow Example

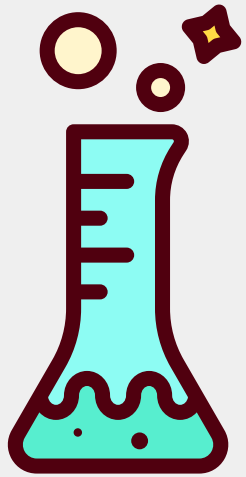
Request to the authorization server's token endpoint

```
POST https://api.authorization-server.com/token
grant_type=authorization_code&
code=AUTH_CODE_HERE&
redirect_uri=REDIRECT_URI&
client_id=CLIENT_ID&
client_secret=CLIENT_SECRET
```

Server's reply

```
{ "access_token":"RsT5OjbzRn430zqMLgV3Ia", "expires_in":3600 }
```

OAuth 2.0: Access Token Usage



curl Request

```
curl -X POST -H
"Authorization: Bearer ACCESS_TOKEN"
"https://api.app.com/v2/$OBJECT"
```

IP Security

IP Security

Definizione

IP Security, è uno standard per reti a pacchetto che si prefigge di ottenere connessioni sicure su reti IP.

Proprietà

- Segretezza
- Autenticazione
- Integrità

IPSec: sicurezza a livello di rete

PRO

- Le applicazioni “delegano” la sicurezza al livello sottostante
- Non vi è la necessità di insegnare agli utenti i meccanismi di sicurezza

CONTRO

- Comunicazione più pesante
- Necessità di supporto da parte del Sistema Operativo

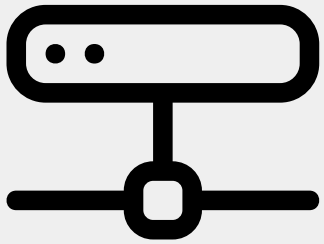
IPSec: protocols

Authentication Header

Encapsulating Security Protocol

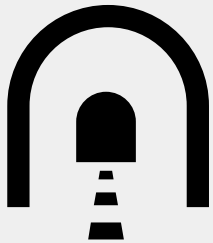
Internet Key Exchange

IPSec: Modes (1)



Transport Mode

- Payload
- End-to-end

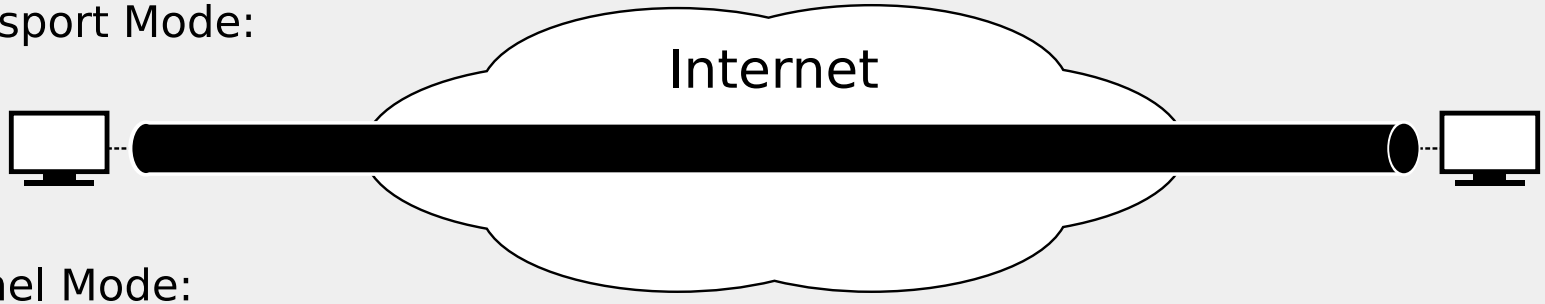


Tunnel Mode

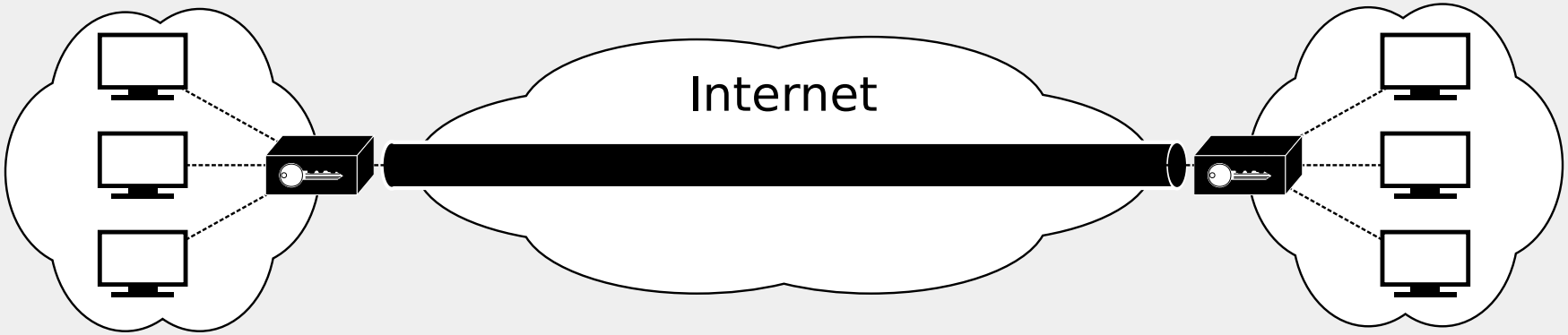
- Entire IP Packet
- Multiple hosts

IPSec: Modes (2)

Transport Mode:



Tunnel Mode:



IPSec: Transport Mode

AH

- Autentica payload
- Autentica porzioni dell'IP Header
- Autentica gli extension header di IPv6

ESP

- Cifra payload
- Cifra porzioni dell'IP Header
- Cifra gli extension header di IPv6

ESP (with auth)

- Cifra payload
- Cifra porzioni dell'IP Header
- Cifra gli extension header di IPv6
- Autentica il payload

IPSec: Tunnel Mode

AH

- Autentica l'intero pacchetto IP
- Autentica porzioni dell'IP Header
- Autentica gli extension header di IPv6

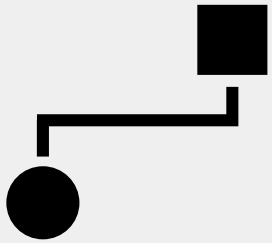
ESP

- Cifra l'intero pacchetto IP
- Cifra porzioni dell'IP Header
- Cifra gli extension header di IPv6

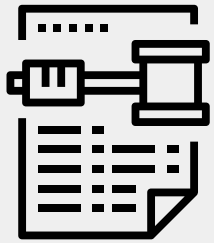
ESP (with auth)

- Cifra l'intero pacchetto IP
- Cifra porzioni dell'IP Header
- Cifra gli extension header di IPv6
- Autentica l'intero pacchetto IP

IP Security Policy

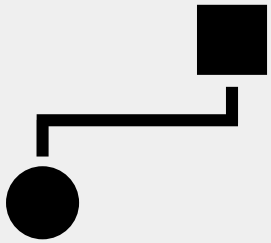


Security Associations Database



Security Policy Database

IPSec: Security Associations



Definizione

Un'association è una connessione logica tra un mittente e un destinatario che offre servizi di sicurezza.

Security Parameters Index

IP Destination Address

Security Protocol Identifier

IPSec: Security Associations Database

Security Parameter Index

Sequence Number Counter

Sequence Counter Overflow

Anti-Replay Window

AH Information

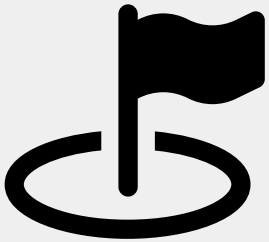
ESP Information

Lifetime

IPSec Protocol Mode

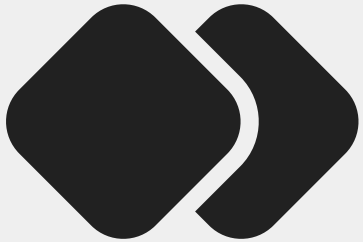
Path MTU

IPSec: Security Policy Database



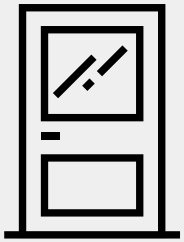
Remote IP Address

Local IP Address



Next Layer Protocol

Name

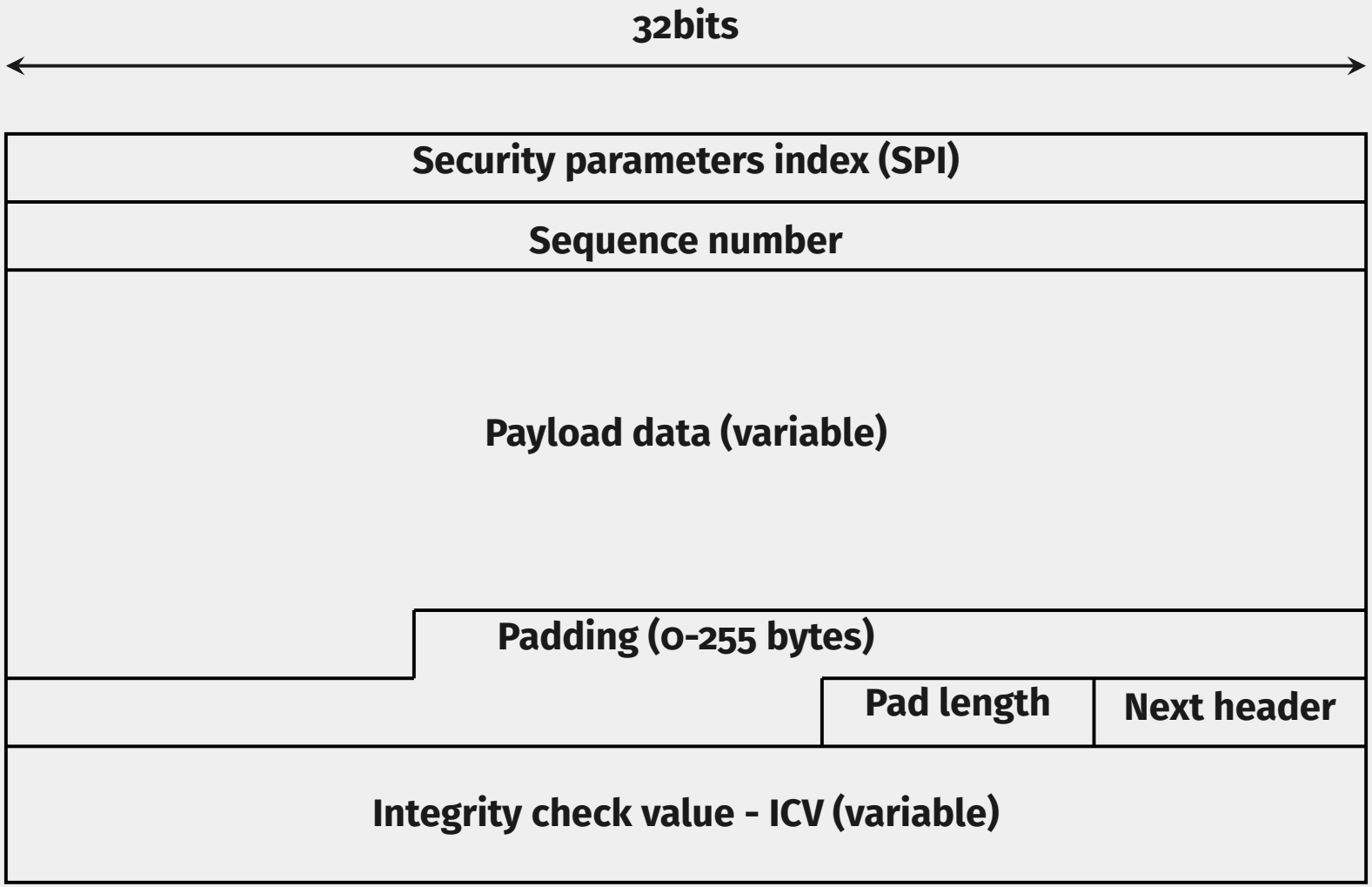


Local and Remote Ports

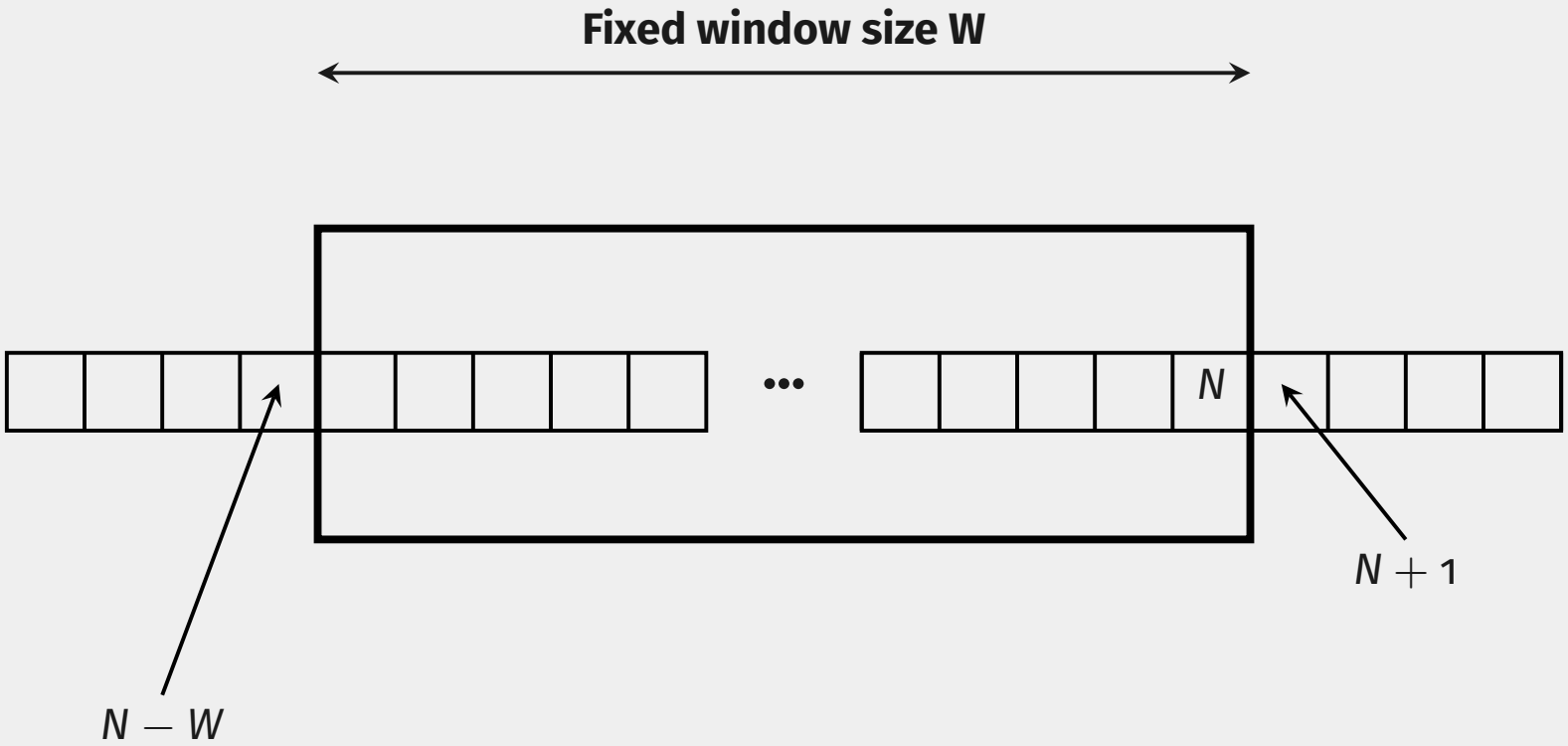
IPSec: Host SPD example

Protocol	Local IP	Port	Remote IP	Port	Action
UDP	1.2.3.101	500	*	500	BYPASS
ICMP	1.2.3.101	*	*	*	BYPASS
*	1.2.3.101	*	1.2.3.0/24	*	PROTECT: ESP
TCP	1.2.3.101	*	1.2.4.10	80	PROTECT: ESP
TCP	1.2.3.101	*	1.2.4.10	443	BYPASS
*	1.2.3.101	*	1.2.4.0/24	*	DISCARD
*	1.2.3.101	*	*	*	BYPASS

IPSec: ESP Format



IPSec: Anti-replay Mechanism



IPSec: Internet Key Exchange

Definizione

IKE rappresenta il protocollo che si occupa della gestione e distribuzione delle chiavi.

- Oakley (Diffie-Hellman based)
- ISAKMP (session keys)

> $ WHOAMI

- Davide Carnemolla
- Herbrant (Telegram, Github, Discord, ...)
- herbrant@protonmail.org