The Zero Trust Security Model

Vincenzo Calabrò

Modulo n. 1

Standard NIST

Motivazioni

Definizioni

Visione d’insieme

Agenda modulo n. 1

- Criticità
- Minacce
- Cambio di paradigma

- Nascita e storia di Zero Trust

- Standard NIST SP 800-207 “Zero Trust Architecture”
  - Definizioni
  - Concetti base
  - Principi
  - Requisiti

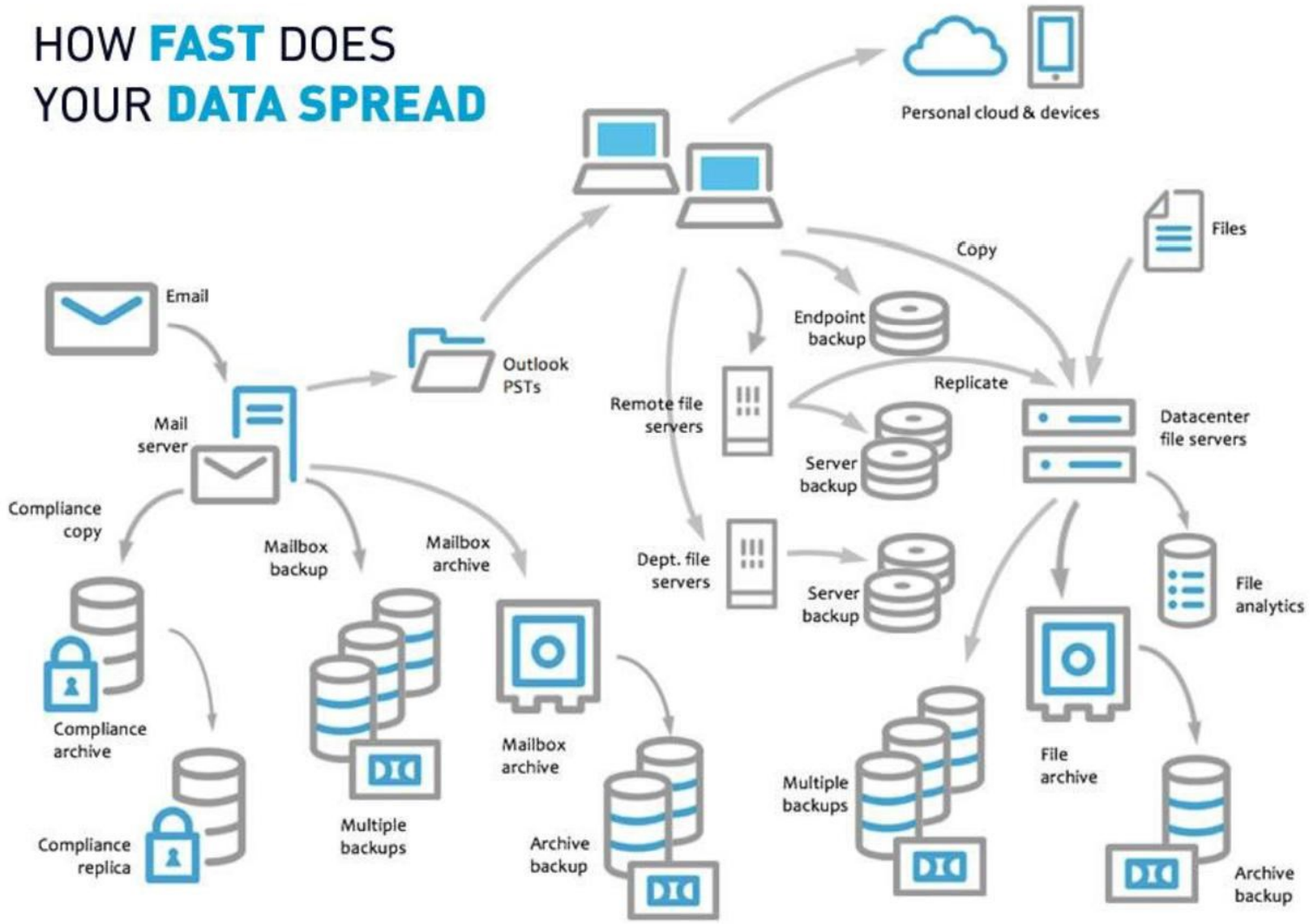
Criticità

- Strutture di dati sempre più complesse
- Applicativi di gestione sempre più complessi
- Hardware, container, virtualizzazioni sempre più complessi
- Normative sempre più stringenti per garantire confidenzialità, integrità, disponibilità dei dati
- Minacce esterne e interne

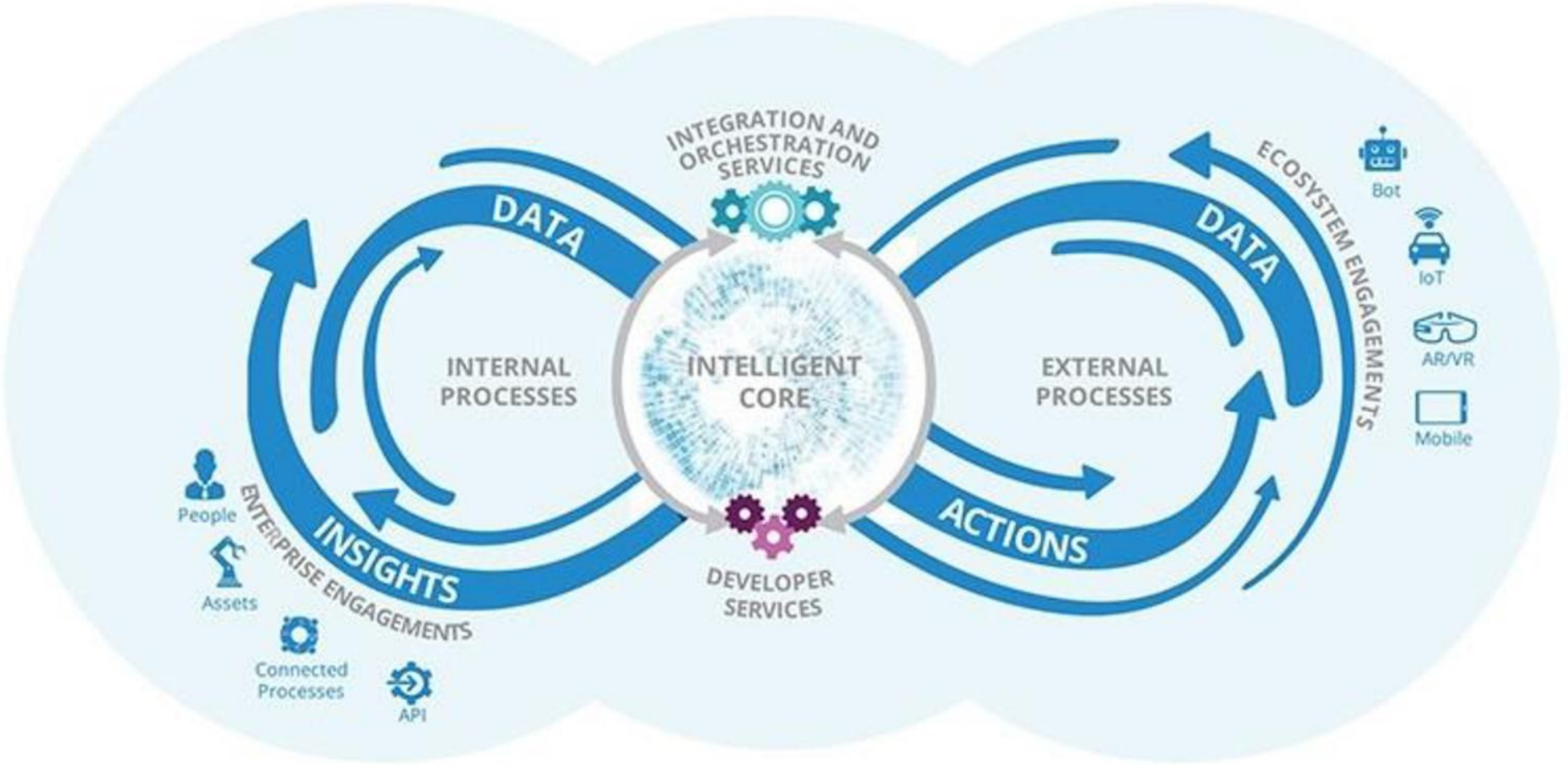
Data Complexity

- Dati immagazzinati in posti sempre più sparpagliati
- Dati fuori dal perimetro (cloud, fornitori esterni, SmartWorking)
- Mancanza di un perimetro definito della propria Organizzazione
- Dati vengono processati da più attori (~80%)
- Dati sempre più «corposi»

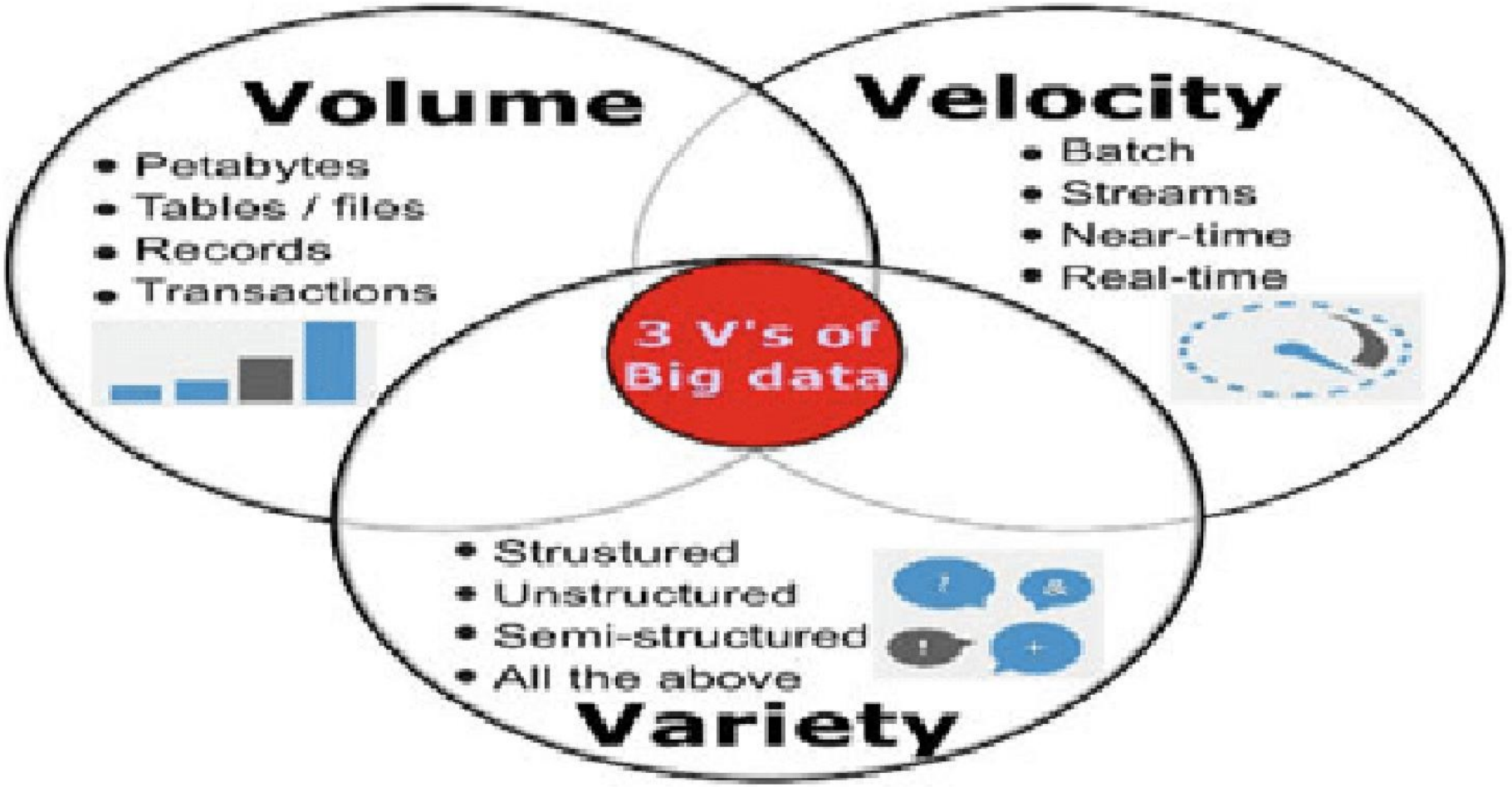
Dati sparpagliati – mancanza di perimetro



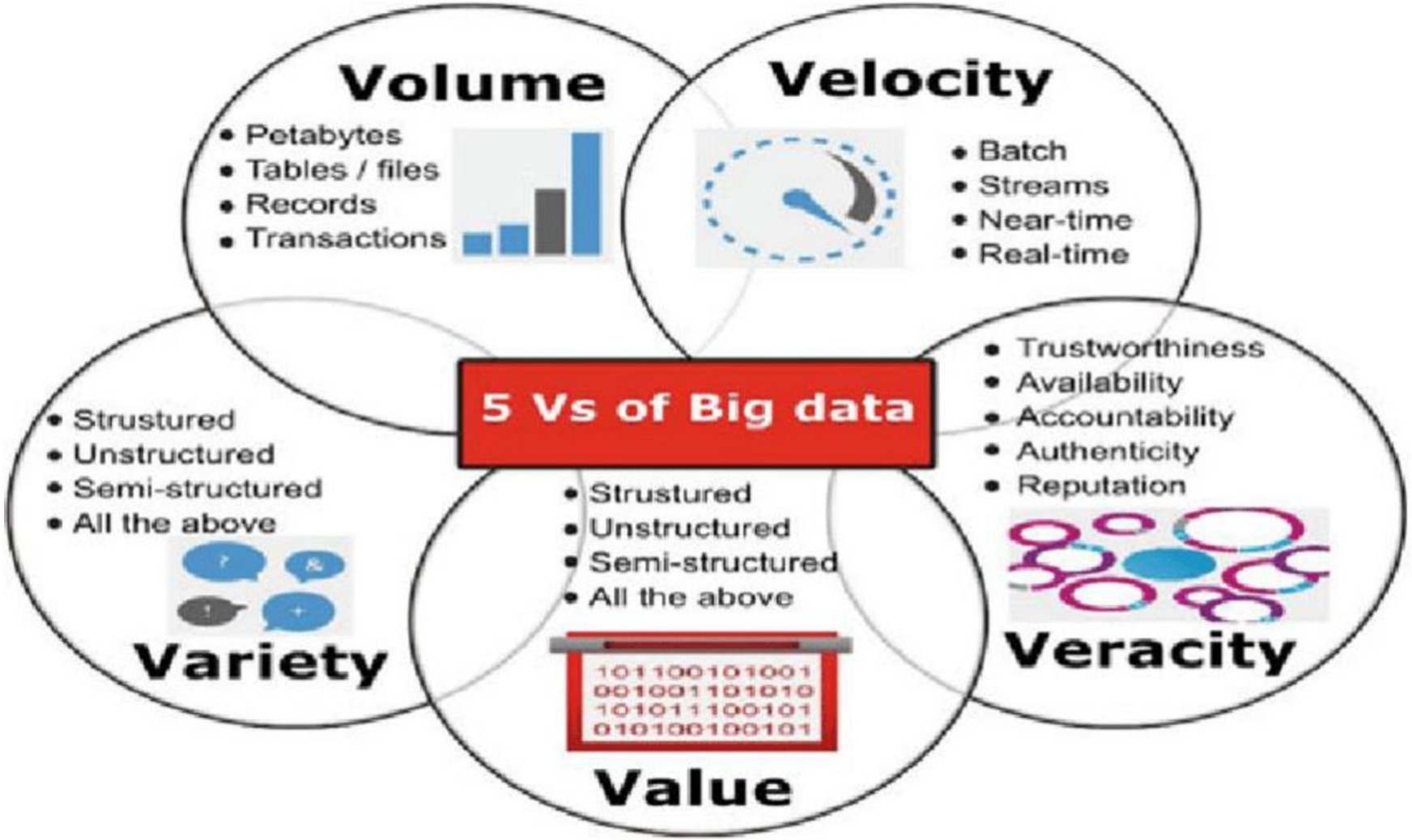
Dati gestiti da più organizzazioni



Verso il Big Data



Verso il Big Data

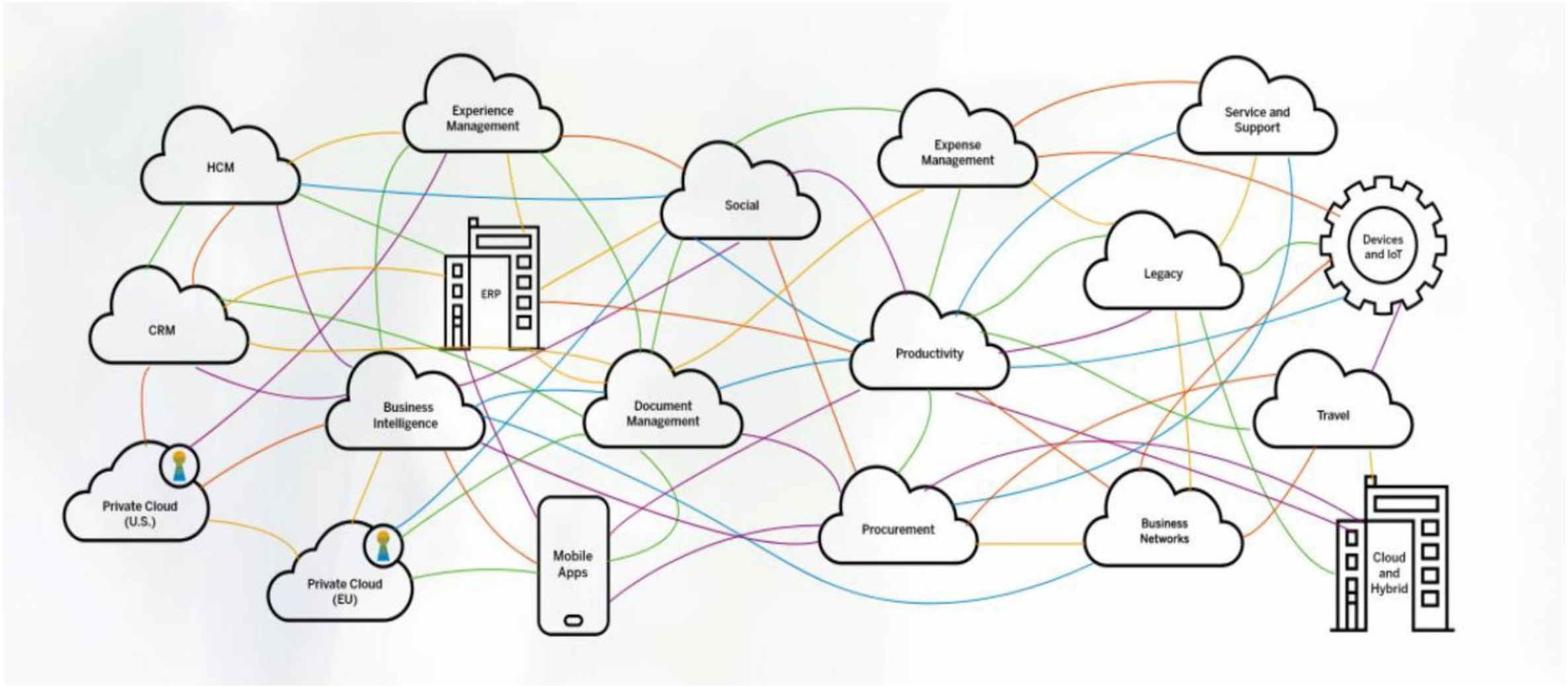


Application and Infrastructure Complexity

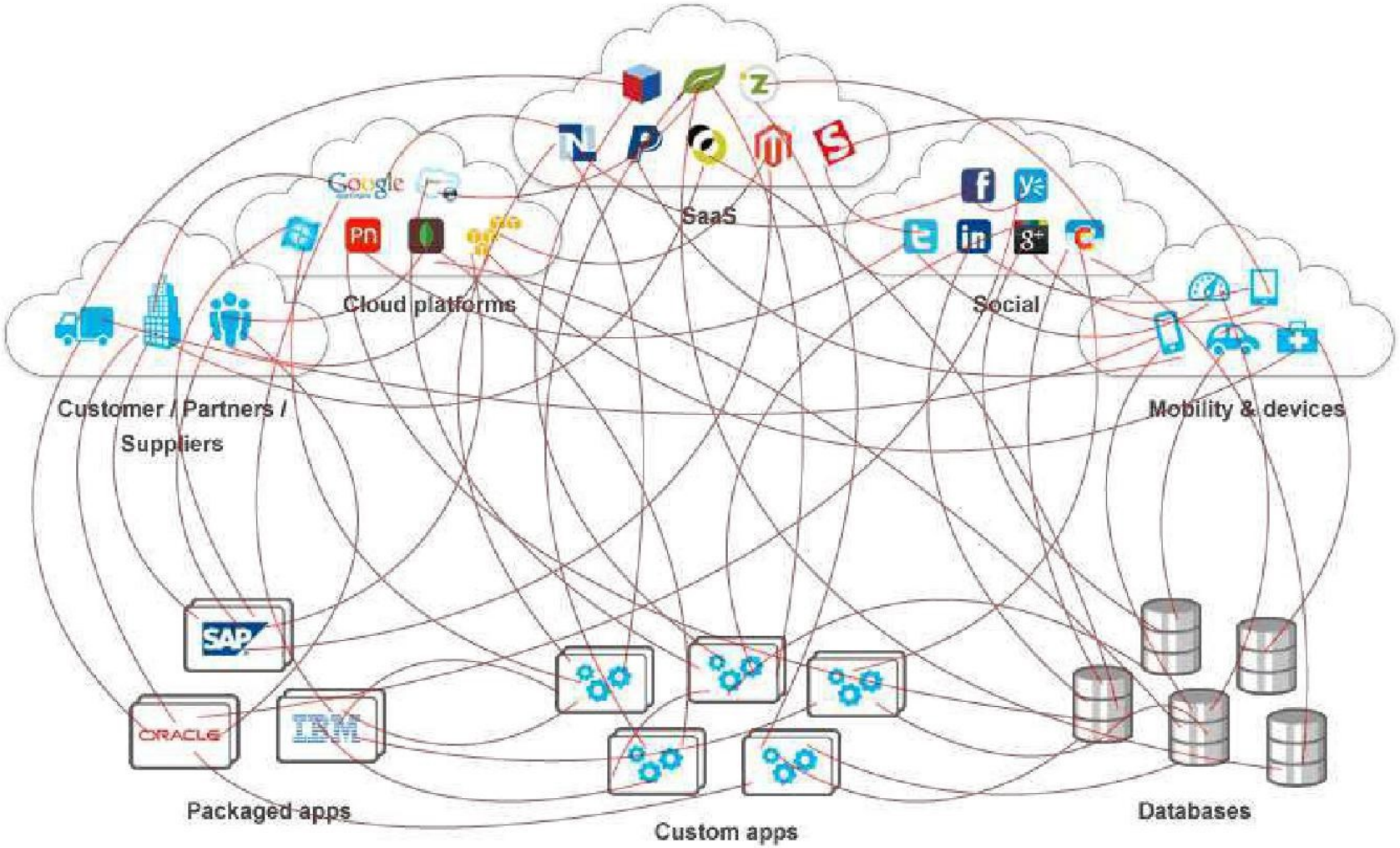
- Complessità nel numero e nel tipo di servizi erogati
- Complessità nell'integrazione delle diverse applicazioni
- Complessità nell'utilizzo dei dati per sviluppo software «in casa»
- Complessità nei layer di virtualizzazione, applicativi, hardware
- Complessità nel fornire servizi agli utenti

Impossibile controllare puntualmente ogni macchina e «rispondere» a tutti i possibili allarmi

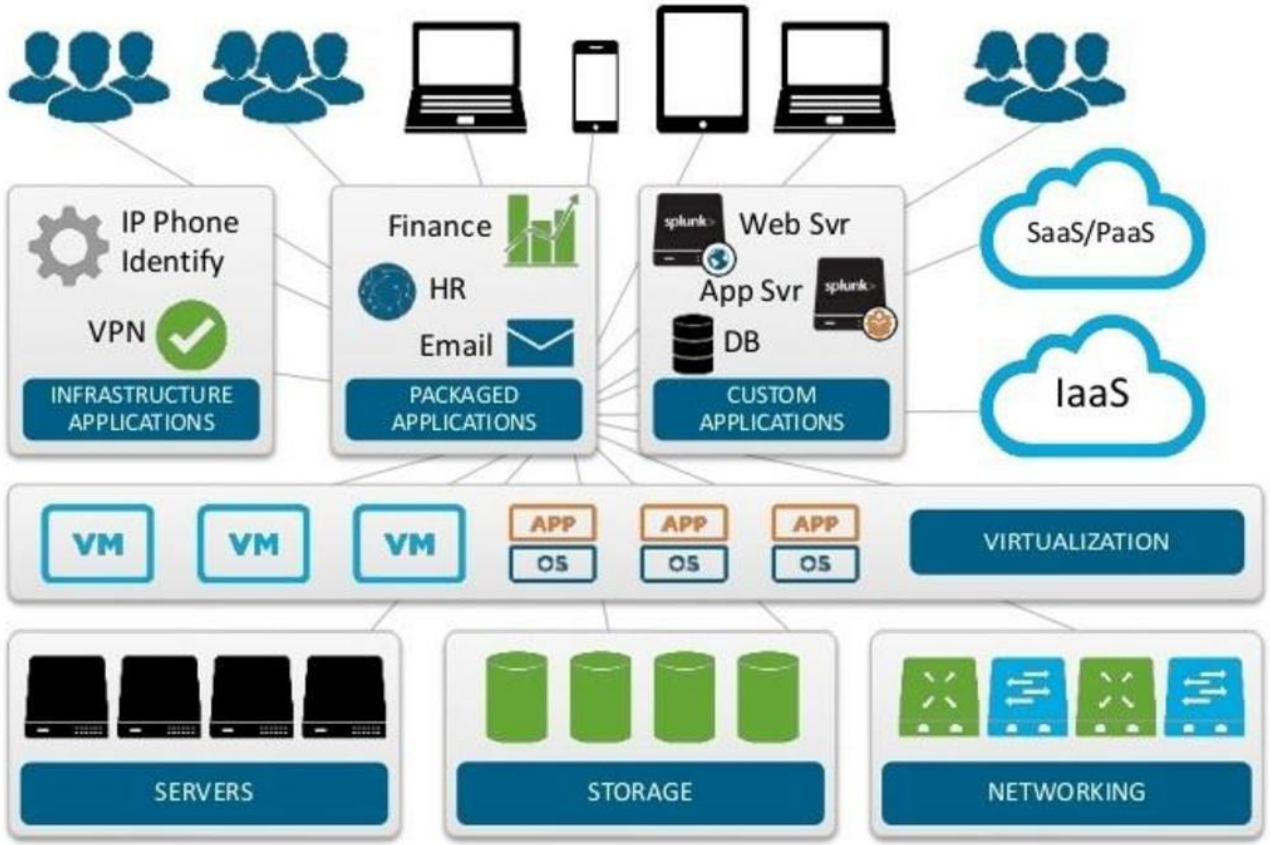
Numero e tipo di servizi erogati

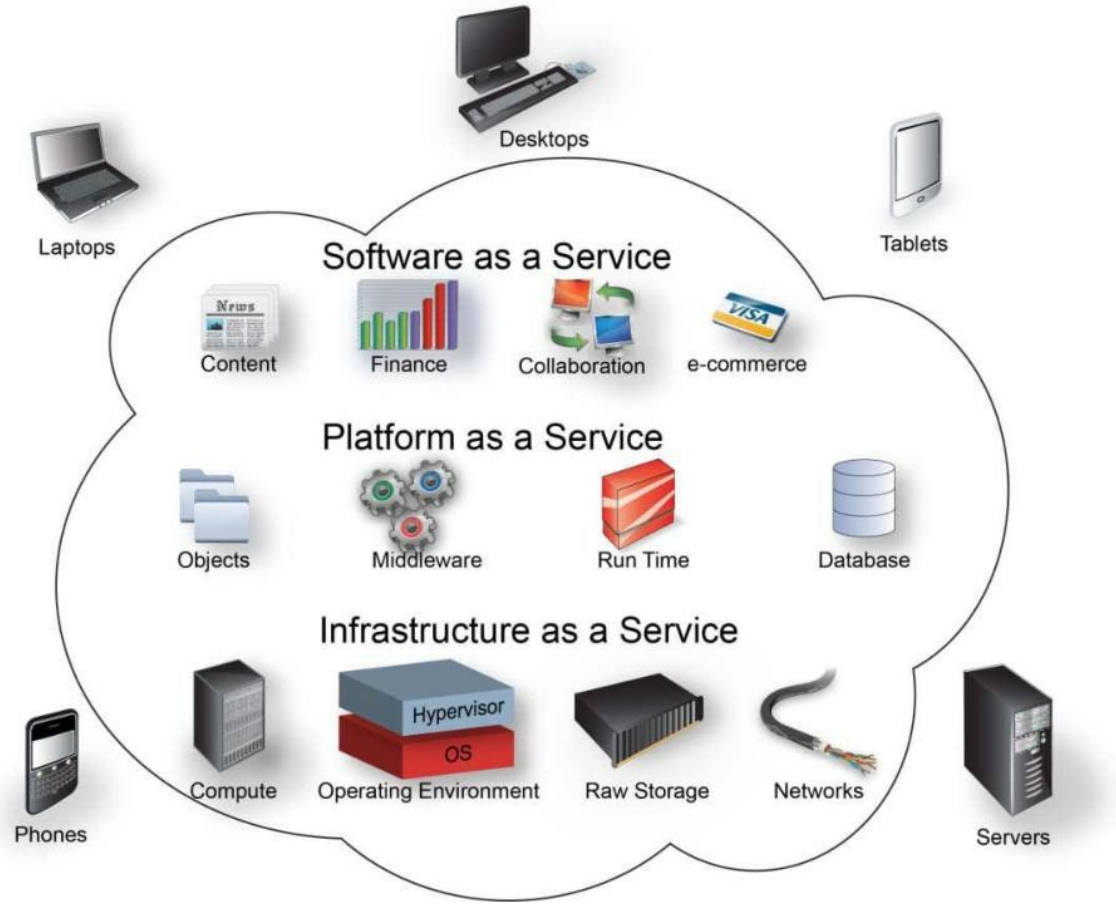


Integrazione fra applicazioni



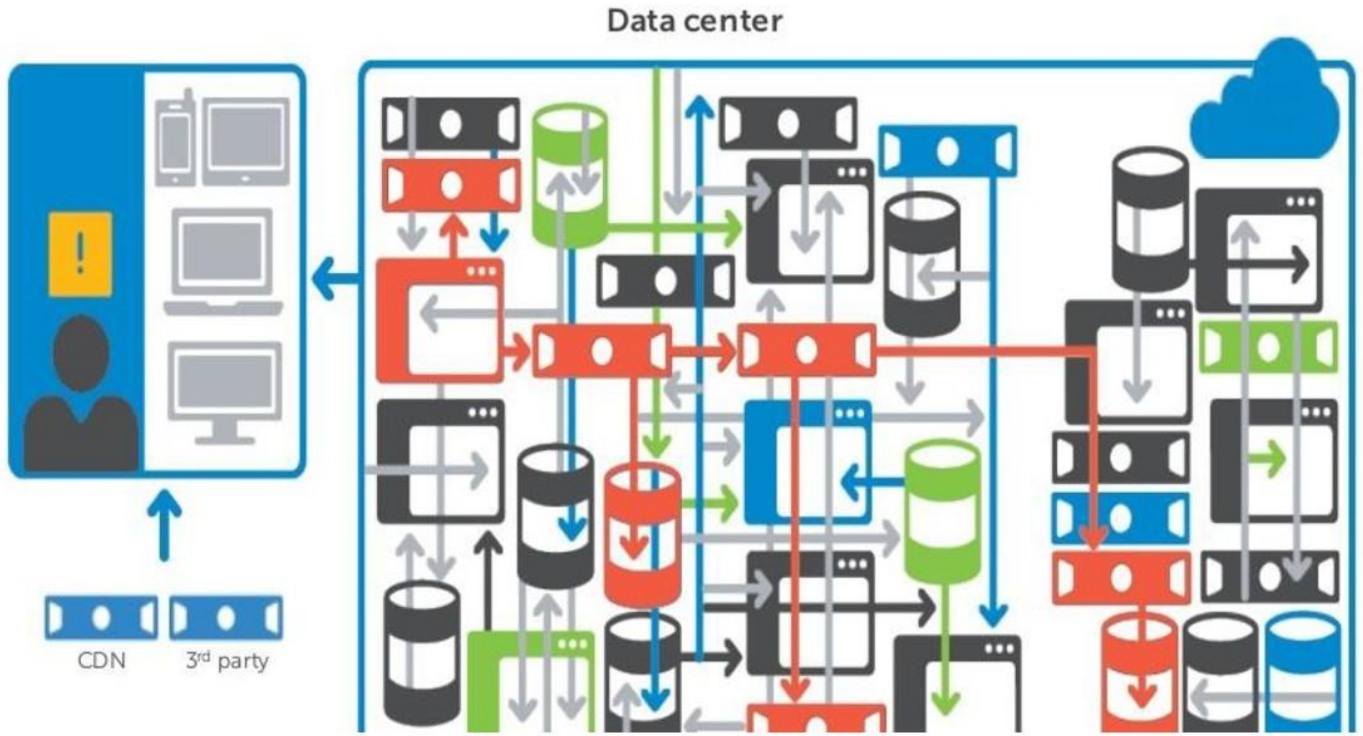
Infrastructure – virtualizzazione container - SaaS PaaS IaaS

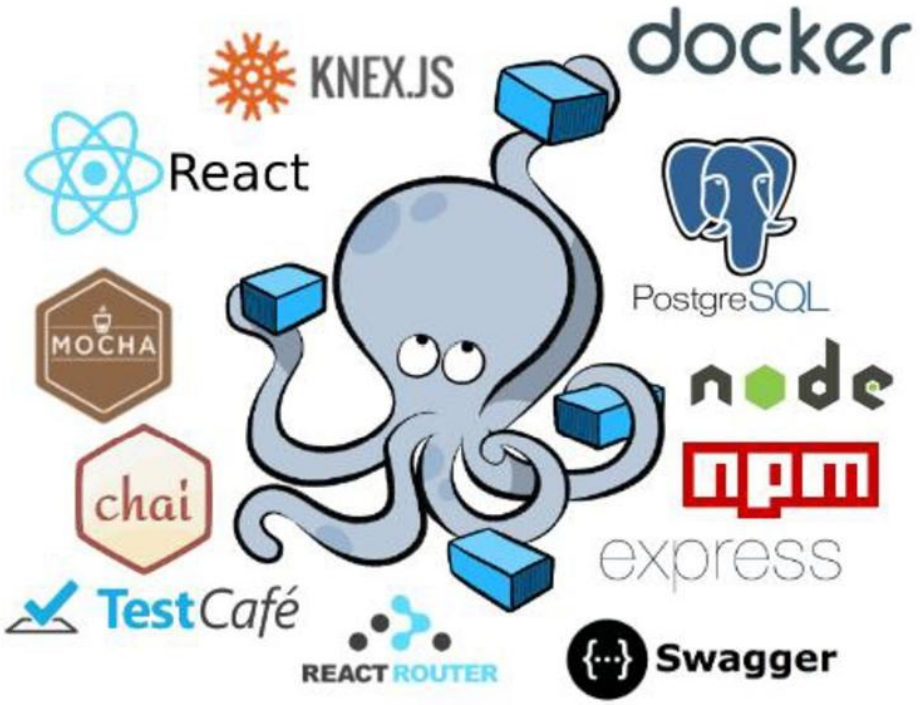




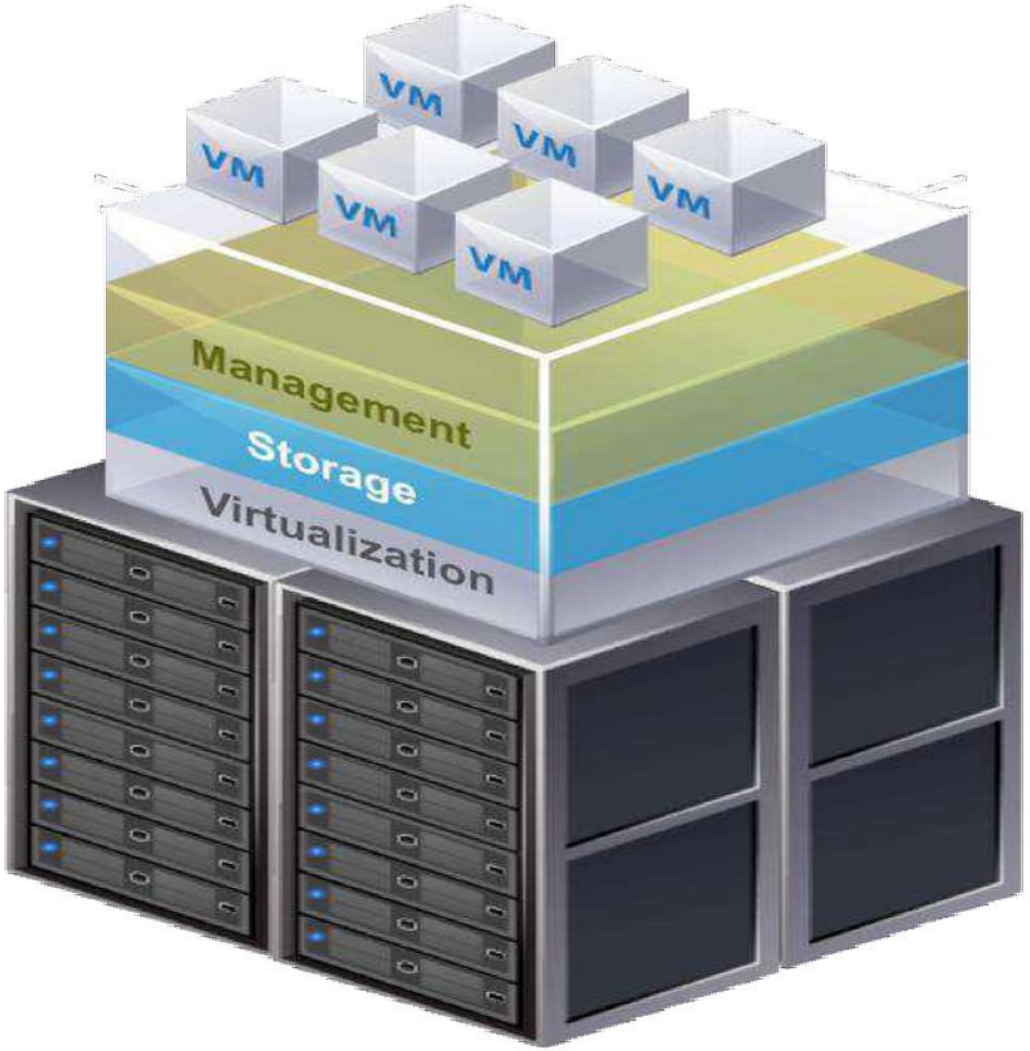
Infrastructure Software

Infrastructure Complexity Affects Visibility





HyperConvergence



Hyper-Converged Software

Industry-Standard Hardware

Complessità di normative e contratti

- Leggi a tutela dei dati disomogenee o non presenti
- Non solo in Asia o USA, ma anche in Europa!
- *«consenso informato»:* cosa significa per una organizzazione e per un utente?

Privacy Law Complexity

WFA Global Privacy Map

An overview of data protection and privacy regulation in key markets

Consenso informato?

Reading Time for a «Terms of Service»

https://www.visualcapitalist.com/terms-of-service-visualizing-the-length-of-internet-agreements/

Terms of Service: TL;DR

https://tosdr.org/

Lettera aperta ad un Preside
https://www.agendadigitale.eu/scuola-digitale/liberiamo-la-scuola-dai-servizi-cloud-usa-lettera-aperta-ai-presidi/

Minacce

- Malware, Virus, BotNet, DDoS
- Social Engineering, Phishing, Business Email Compromise (BEC)
- IoT, SmartWorking, Bring Your Own Device (BYOD) … etc etc
- Databreach sempre più numerosi (ma forse sempre meno divulgati?)

Mid Year 2020 At A Glance

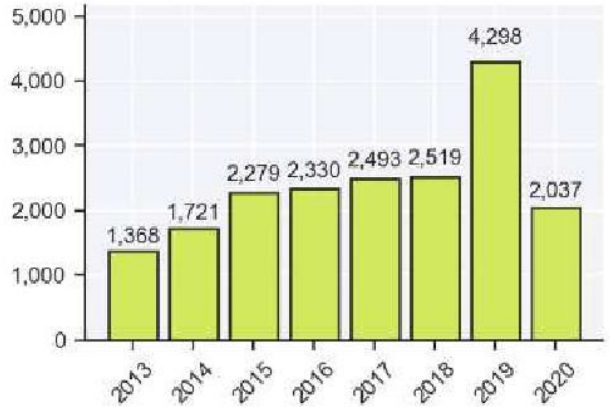


Figure 1: Number of breaches reported by Q2 each year

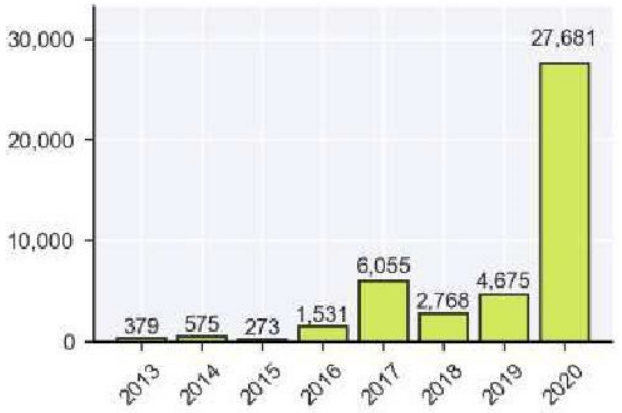


Figure 2: Number of records lost (in millions) reported by Q2 each year

Fonte: Risk Based Security Inc.

Mid 2020 Year Report

Data Breach Quick View

https://pages.riskbasedsecurity.com/en/2020-mid-year-data-breach-quickview-report

Accelerata in tempo di COVID-19

- Cambio degli attacchi
- Più numerosi
- Più verso lato client
- Più verso gli utenti e meno verso i sistemi
- Aumento dell'uso di applicazioni (Zoom), con nuove possibilità di attacco
- Organizzazioni più esposte ai rischi (accessi remoti)

Le minacce

- Sia esterne che interne – deperimetralizzate
- No attacchi tecnici sofisticati
- Soprattutto credenziali rubate
  - Password deboli
  - Attacchi (stavolta sì) anche molto sofisticati di Social Engineering

Quello di cui ci fidiamo per dimostrare che un utente ha diritto ad utilizzare un dato

Utente/Password

non è più sufficiente
si ritorce contro di noi

è il tallone di Achille della nostra struttura

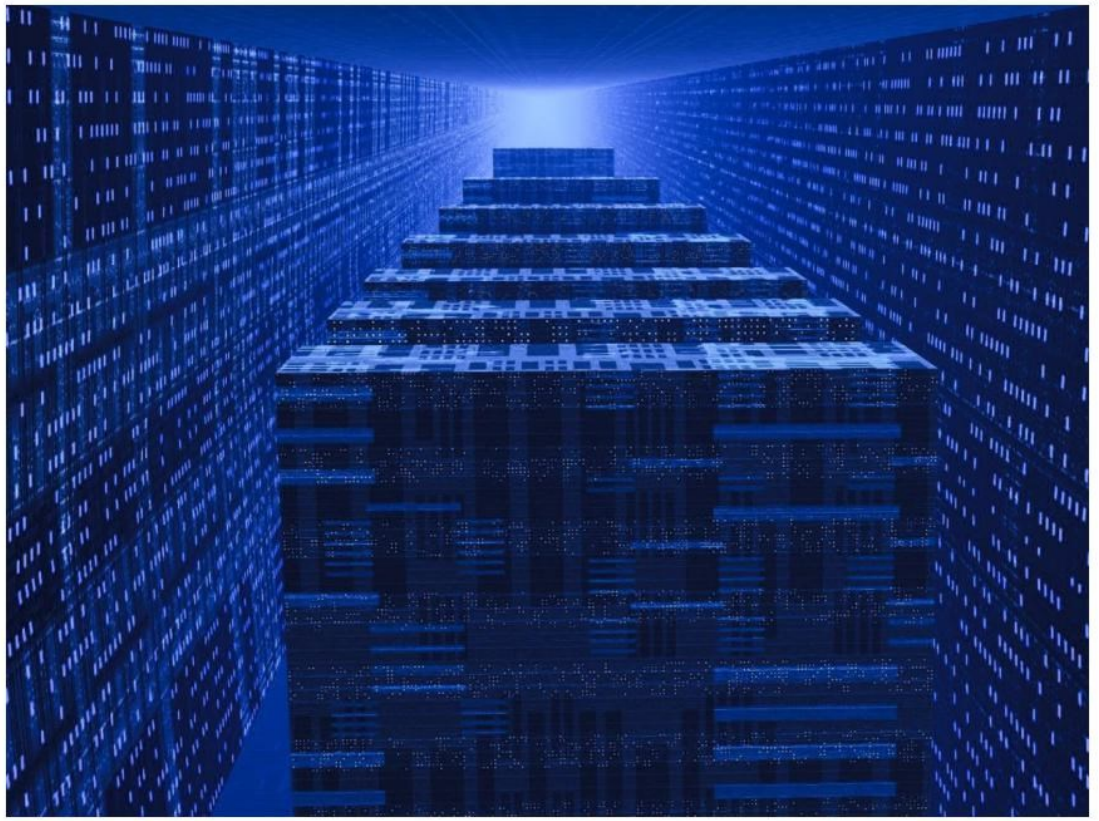
La Storia - oggi

Cambio di paradigma

- Servizi più pervasivi
- Minacce molto più critiche

Impattano non più sui sist. informatici

- **Sui dispositivi personali**
- **Sulla privacy**
- **Sull'identità delle persone**



Cambio di paradigma

- Consapevolezza nella Leadership
- Piano di sicurezza sviluppato dall'alto
- Analisi dei rischi
- Policy e procedure

- Approccio non puntuale ma secondo **l'analisi dei rischi**
- Approccio non perimetrale ma **Zero Trust**

La storia di Zero Trust

Concetto di assenza di perimetro «Black Core» - ~1994

Introdotto nel 2010 da Forrester Research in partnership con NIST

- Propagare la fiducia in situazioni tradizionali aumenta le vulnerabilità

Ripresa dei lavori su **Zero Trust** a seguito di minacce sempre più pervasive

- 09/23/19: SP 800-207 (Draft)
- 02/13/20: SP 800-207 (Draft)

Il 20 Agosto 2020 pubblicato lo STANDARD NIST SP 800-207 "Zero Trust Architecture"

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf

Panoramica

Zero Trust pone fiducia ZERO a qualsiasi elemento di una rete

Zero Trust assume che le organizzazioni dovrebbero verificare tutte le richieste di utilizzo di un dato prima di concederne l'accesso

Incentrato solamente sui dati

- Il dato diventa la parte più importante da tutelare
- Indipendentemente da dove sia immagazzinato
- Dovrebbe essere acceduto nella maniera più sicura possibile
- Controlli continui, dinamici e granulari sull'autenticazione e l'autorizzazione

Si passa dalla difesa dei sistemi alla difesa del dato

Cosa è, cosa non è

Zero trust è una serie di

- Principi
- Strategie
- Best practices di sicurezza

- Non è un prodotto
- Non è una architettura di rete
- Non è un disegno/deployment di sistemi

Le basi

La fiducia non è mai data per scontata ma va valutata continuamente

Rapporto diretto **End-to-End (1:1)** fra utente e risorsa

Tiene conto di

- Identità e Identity Management
- Credenziali
- Access Management
- Operations, procedure
- Infrastruttura di connessione

Per ogni singola sessione di accesso al dato

Zero Trust

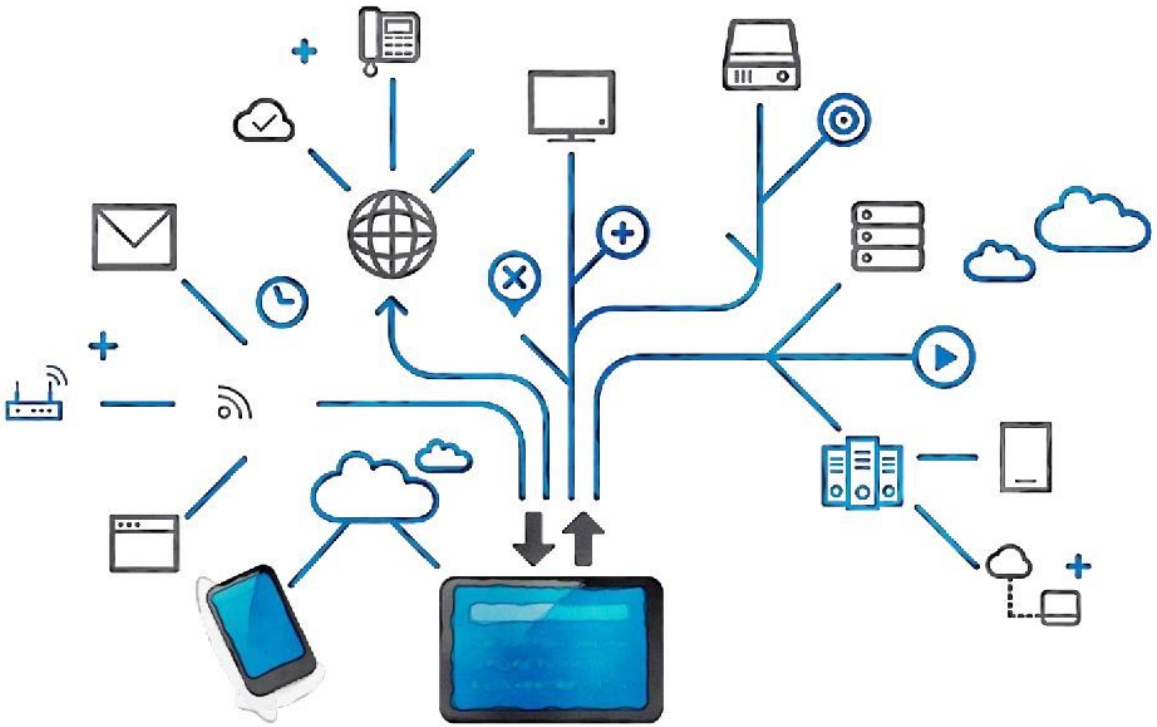
E’ l’nsieme di idee e concetti
con lo scopo di decidere se
autorizzare l’accesso ad una risorsa
sulla base di ogni singola sessione e
degli attributi della singola sessione

Zero Trust Architecture

E’ il piano sviluppato secondo i principi Zero Trust che tiene conto delle relazioni fra risorse (accedenti ed accedute), le procedure e le policy di accesso dell’organizzazione

Zero Trust Enterprise

E’ l’infrastruttura di rete e
le procedure/politiche operative
implementate nella Organizzazione
secondo la ZTA
(che è pianificata secondo lo ZT)



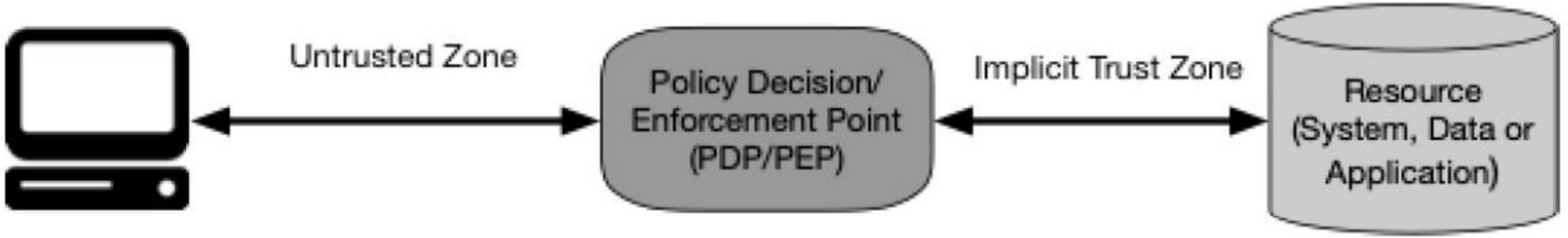
Principi fondamentali

- Prevenire accessi non autorizzati alle risorse
- Controlli di accessi più granulari possibili

Focus

- Autenticazione
- Autorizzazione
- Limitare al massimo le zone «trusted» - nel tempo e nella postura
- Principio del minimo privilegio (granularità)

Architettura di base per una richiesta



Studente – docente
Missione – ufficio - SmartWorking
Applicazione

Autenticità del soggetto && Validità della richiesta

Soggetto:

- Come si è autenticato
- Con quale client e postura del client
- Orari, geolocalizzazione, utilizzo reti pubbliche

PDP/PEP: E' autentico e valido? – E' TRUSTED?

Accede alla risorsa – SOLO QUELLA!

Lo **Zero Trust** rende la trusted zone più piccola possibile
(il perimetro diventa un frattale la cui dimensione tende a ZERO!)

Principi/Fondamenti 1

Per l'implementazione dei principi deve essere valutato all'interno dell'Organizzazione «il cosa» e «il come» (risk analisys)

Comunicazioni sicure utente/risorsa indipendentemente dal luogo

- Ogni connessione, anche proveniente dalla «intranet» deve essere valutata come una connessione esterna e valutata volta per volta
- Ogni sessione autorizzata si stabilisce con un tunnel criptato fra utente e risorsa

L'accesso è garantito solo per la singola sessione

- L'accesso è valutato prima di stabilire la sessione
- Se si vuole accedere ad altro serve una nuova autorizzazione

Principi/Fondamenti 2

L'accesso è valutato da **policy dinamiche** che considerino un pool di attributi (dipendenti dai rischi che ci possiamo assumere o no)

- Chi è, a quale federazione/ruolo appartiene
- Che device usa, quali versioni del sistema operativo/app
- Geolocalizzazione, data/orario di connessione
- Anomalia rispetto a «normale comportamento»
- Principio del minimo privilegio

L’autenticazione e autorizzazione sono dinamiche e applicate prima di dare accesso

- Valutare il livello di rischio ogni volta prima di autorizzare
- Basata fortemente su ICAM (Identity, Credential, Access Management) e asset management
  - Autenticazione semplice, con certificato, MFA, OneTimePassword etc
- Monitoraggio continuo della postura dell’utente e delle risorse

Ogni device che utilizza le risorse dell'organizzazione è «not trusted»

- Assicurarsi che i device dell'organizzazione siano nel miglior stato di sicurezza possibile
- Monitorare e controllare lo stato dell'asset di proprietà (Es. Vulnerability Management e Patch Level Analysis)
- Stabilire policy diverse per asset personali o di terze parti

Monitoraggio continuo della rete e dell'infrastruttura di rete per migliorare le policy in atto

- Traffico di rete
- Accesso ai sistemi
- Può essere utilizzato anche per stabilire il «comportamento normale» di un utente nell'accedere alle varie risorse

Presupposti sulla Network

La intranet non è degna di fiducia, e deve essere trattata come se ci fossero aggressori anche all'interno

- Autenticazione e crittazione del traffico

I device sulla nostra rete possono anche non essere di proprietà dell'organizzazione

- BYOD, telefonini
- Esperimenti o laboratori con partner

Nessuna risorsa è intrinsecamente trusted

- La postura di ogni soggetto deve essere valutata via PEP
- I device dell'organizzazione dovrebbero avere strumenti per forzare maggior fiducia – patch, livelli di autenticazione

Presupposti sulla Network 2

Non tutte le risorse della struttura risiedono fisicamente nella struttura. Queste risorse utilizzano reti e servizi esterni (Es. DNS)

- Utenti remoti
- Servizi cloud

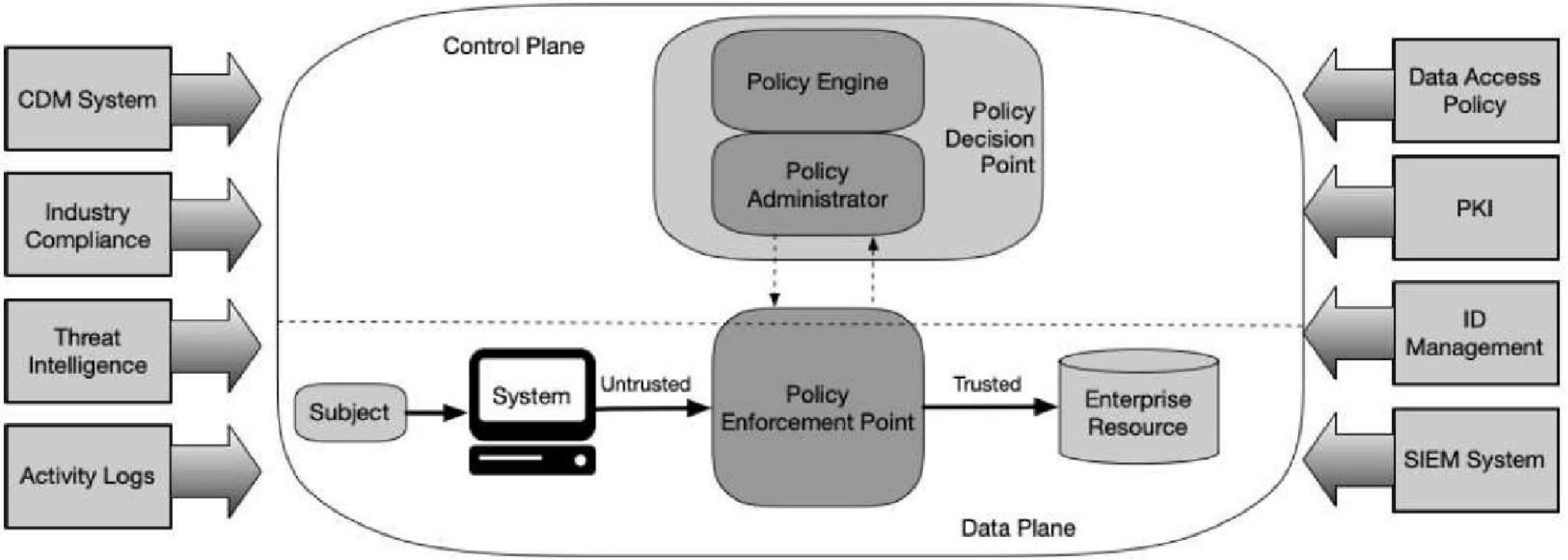
Gli utenti remoti stanno in reti ostili

- Monitorati continuamente
- Autenticati fortemente

L'asset che passa dall'Organizzazione all'esterno e viceversa deve mantenere la politica e posizione di sicurezza coerenti

- Es. PC e utenti che passano da smart working a ufficio
- Cloud che passano da organizzazione a fuori

Componenti logiche di Zero Trust



Policy Engine / Policy Administrator

PDP – Policy Decision Point

Sta sul control plane, elemento fondamentale di Zero Trust

E' formato da PE (Policy Engine) e PA (Policy Administrator)

PE: Decide se il soggetto può accedere alla risorsa

- Attraverso la configurazione delle policy dinamiche
- Mediante dati e attributi relativi alla sessione (tipo di autent., ruolo)
- Mediante dati e attributi relativi al soggetto e alla postura del soggetto

PA: esegue la policy decisa dal PE

- Fornisce un token, o un certificato a vita breve, per la sessione
- Configura il PEP per stabilire un tunnel criptato fra soggetto e risorsa
- Nel caso non sia autorizzato indica al PEP di chiudere la connessione

Policy Enforcement Point

PEP – Sta con una interfaccia sul Control Plane e una interfaccia sul Data Plane

Sta più vicino possibile alla risorsa acceduta

Colui che abilita le connessioni

- Monitorizza la connessione
- Raccoglie informazioni sul client

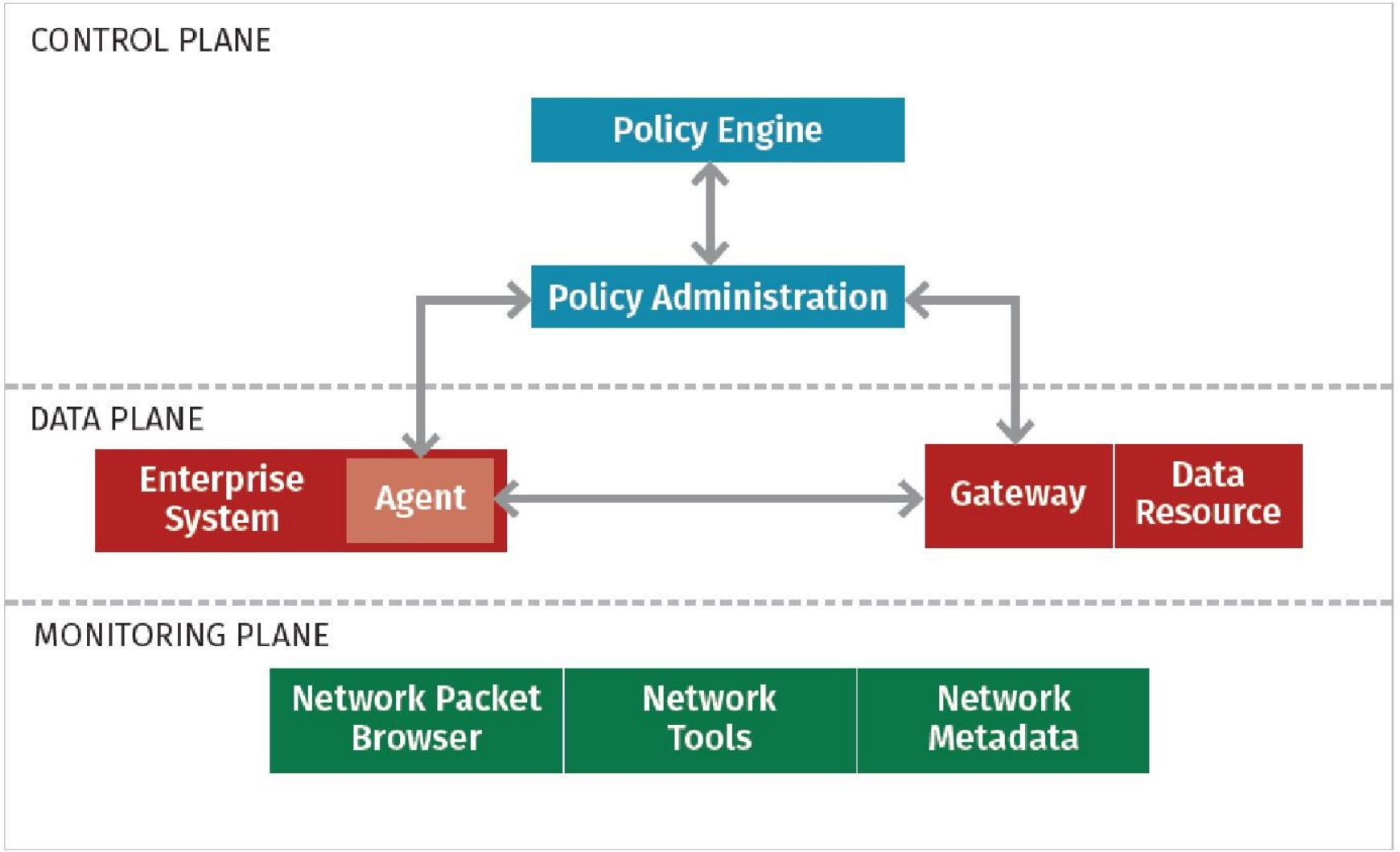
Può essere un componente singolo oppure diviso in due componenti

- Un portale/gatekeeper a cui ci si connette e da cui possiamo scegliere la destinazione/servizio
- Un agent sul client che si connette ad un gateway che smista la sessione sul servizio richiesto

Dietro al PEP c'è la implicit trusted zone, dove sta la risorsa richiesta

Nella maggior parte dei casi è necessario un PEP per ogni risorsa

«The missing plane» - Monitoring Plane



Monitoring Plan

Continuous diagnostics and mitigation (CDM)

Fornisce i feed per stabilire il livello di fiducia di un soggetto

- Tipo di client usato, versione, livello di patch, anche del sistema operativo
- A seconda del prodotto usato può anche applicare le patch o modificare le configurazioni sul client

Industry Compliant System (se c'è): policy dell'organizzazione (ISO, mission, normative)

Threat Intelligence feed: Informazioni su IoC, vulnerabilità, tipi di attacchi, blocklist, incidenti di sicurezza

Data access policies (politiche statiche di accesso)

- Chi/cosa può accedere alla risorsa X
- Attributi, ruoli, regole – sia statiche che generate dinamicamente
- E' la tabella di privilegi in cui l'Organizzazione, in base alla propria missione e regolamenti, stabilisce le regole per l'accesso alle risorse

Enterprise Public Key Infrastructure (PKI)

- Generazione di certificati per risorse, soggetti, applicazioni
- Può essere esterna o interna
- Può non essere basata soltanto su certificati X.509

Monitoring Plan

ID Management System

Creazione, gestione e custodia degli account utente e record di dati connessi (es. attributi LDAP)

- Dati degli utenti, attributi, certificati, ruoli, dispositivi assegnati
- Spesso contiene anche informazioni su PKI
- Può far parte di Federazioni di identità (per includere anche identità di terze parti che accedono al nostro asset)

Network and Activity logs: traffico di rete, log accessi, log eventi

Security Information and event management system (SIEM)

Considerazioni

E’ un viaggio, per farlo tutto ci vogliono anni

Si parte dal piccolo, piccoli pezzetti o servizi (Es. i più sensibili)

L’importante è l’approccio con cui si fa

- Basato sull’identità (autenticazione/autorizzazione)
- Micro segmentazione attraverso NGFW
- Segmentazione fisica e logica della rete

- Analisi dei rischi
- La propria mission
- Le proprie policy, propri dati e servizi
- Il prezzo da pagare per il cambiamento
- il prezzo da pagare per l’implementazione

Deployment

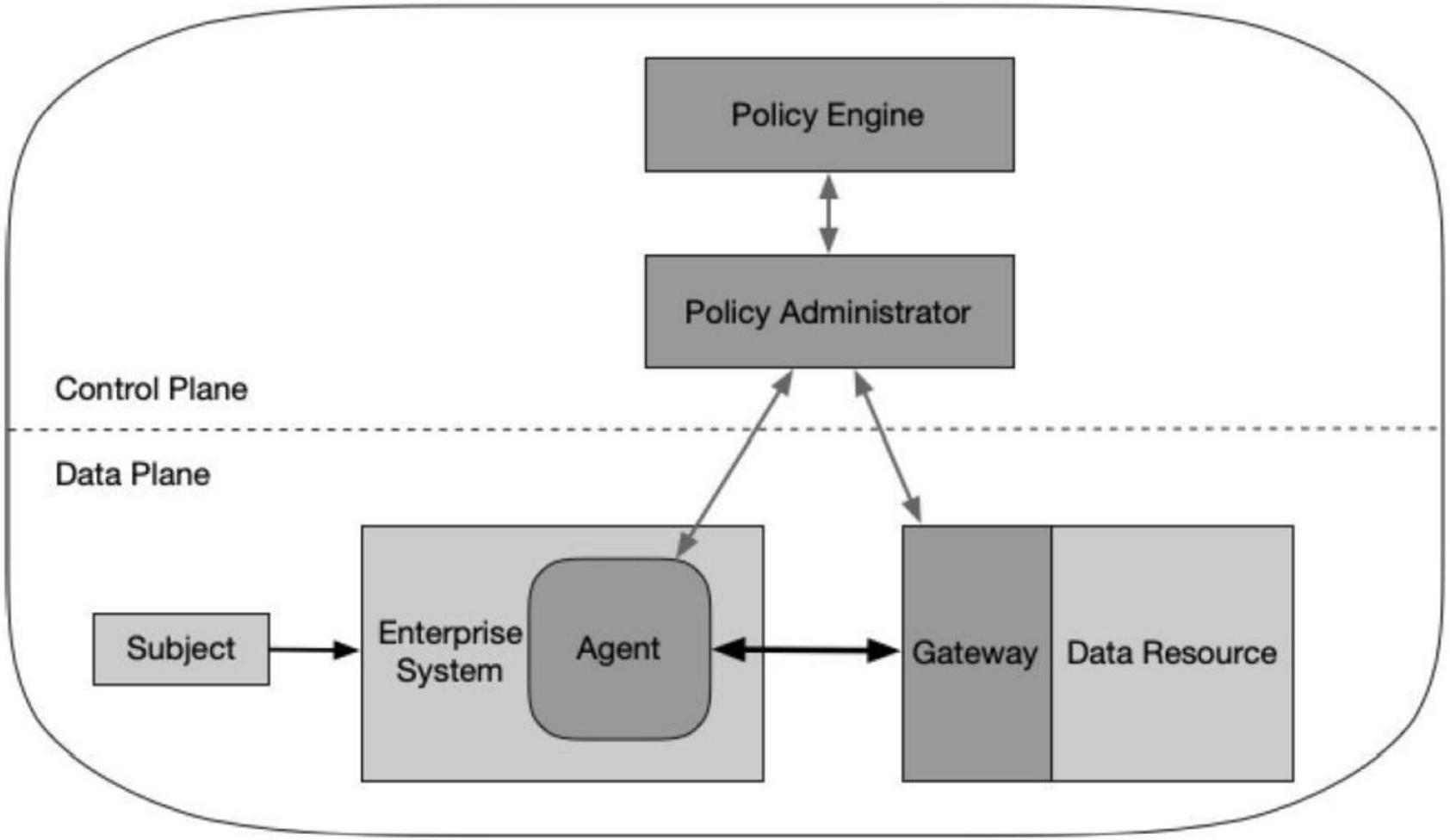
Per il deployment del PEP ci sono varie soluzioni, già pronte anche in ambito commerciale

Sostanzialmente si basano su due tipi diversi di sviluppo

La scelta dell'uno o dell'altro dipende da quello che già c'è nell'organizzazione e dall'architettura che si vuole raggiungere

- Un agent sul client che si connette ad un gateway che smista la sessione sul servizio richiesto
- Un portale/gatekeeper a cui ci si connette e da cui possiamo scegliere la destinazione/servizio

Device Agent/Gateway



Device Agent/Gateway - Casi d'uso

- Quando abbiamo un buon livello di discovery e management di tutti i device dell'organizzazione
- Quando abbiamo già un NGFW vicini alla risorsa (uno per risorsa)
- Non c'è BYOD

Nel caso di implementazione per risorse cloud questo è lo standard per Cloud Security Alliance and Software Defined Perimeter 1.0 (CSA-SDP, 2015)

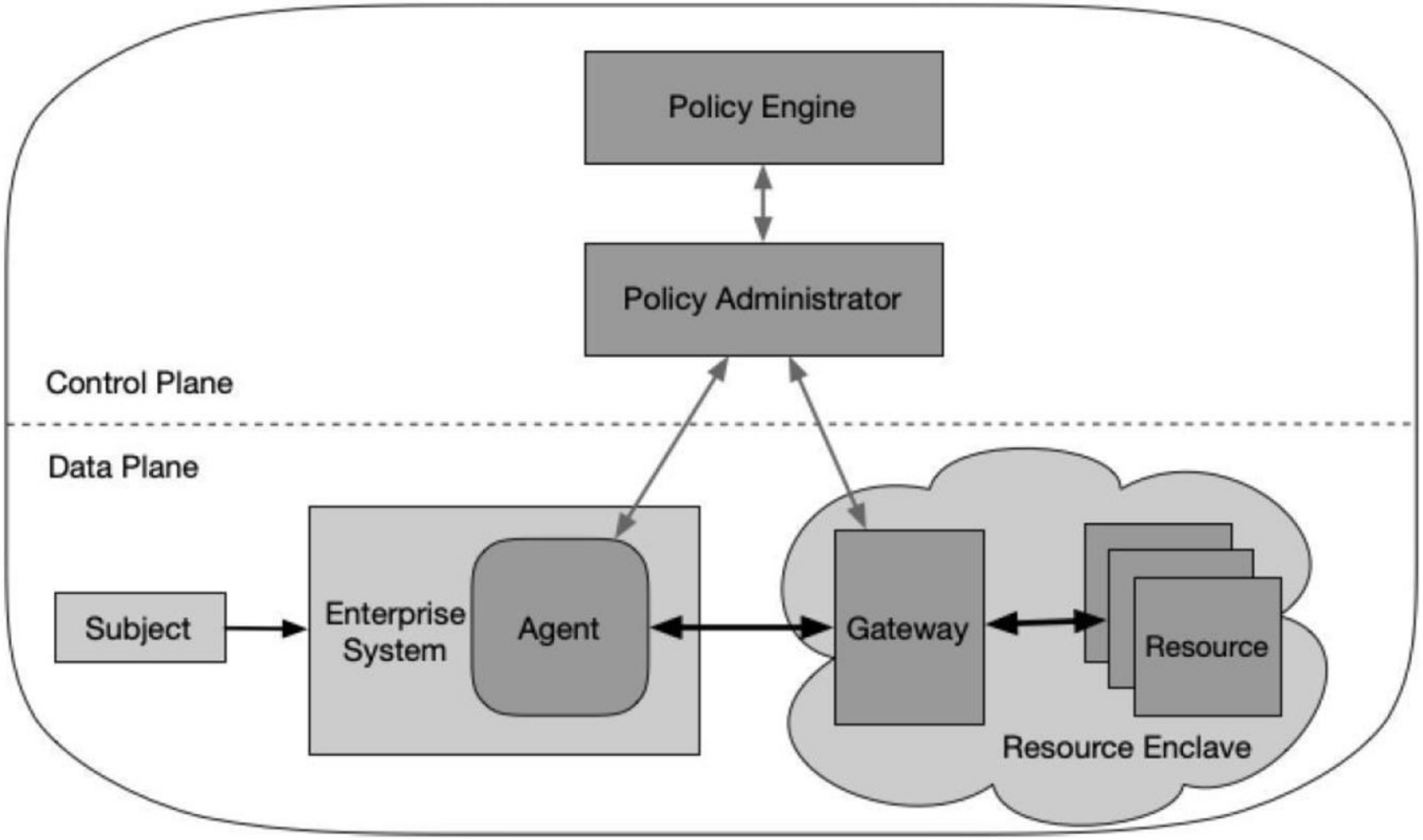
Device Agent/Gateway - esempio

PC dell'Organizzazione che si connette alla procedura stipendi
La richiesta è presa in carico dall'Agent che sta sul PC
L'agent comunica al Policy Administrator la richiesta
La richiesta viene inoltrata al PE affinché dia il verdetto

Se autorizzato: si configura un canale criptato end-to-end fra Agent e Gateway e il flusso dei dati passa da lì

La connessione viene terminata quando viene chiusa dall'utente o va in timeout o se per caso dal PA arrivano alert di sicurezza

Device Agent/Enclave



Device Agent/Enclave – Casi d'uso

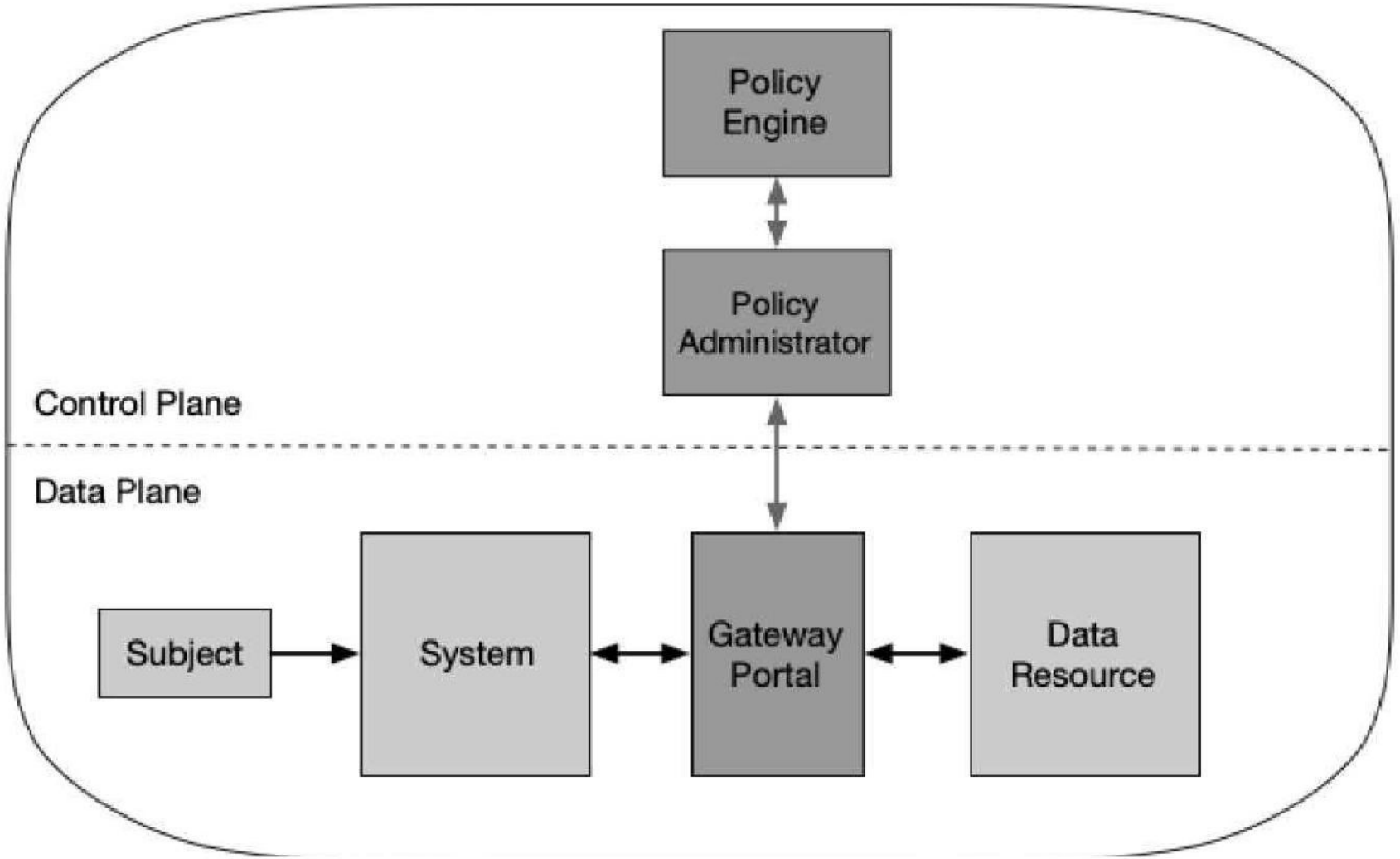
E' lo stesso del Gateway, solo che dietro ci sono più risorse
Quando non è possibile mettere un gateway/NGFW per ogni risorsa

Si mette davanti ad un pool di risorse (Es. un datacenter)
Si usa quando abbiamo servizi in cloud locali o remoti o micro-service

Importante: ogni enclave deve servire ad un'unica funzione (es. posta)

Possibili rischi: si possono fare movimenti laterali fra le risorse nella zona dietro al gateway

Resource Portal



Resurce portal – casi d'uso

In questo caso l'utente si connette ad un portale che fa da gateway per le risorse, che può essere singola o un'enclave

- Un portale per accedere a servizi cloud o datacenter o servizi legacy
- Per casi di BYOD
- Non necessita di software sul client/endpoint

Importante:

- *Non si hanno molte informazioni sulla postura del client: aggiungere protezioni tipo remote browser isolation*
- *Il portale è prono ad attacchi, anche DoS, deve essere CORAZZATO*

Resource Portal - esempio

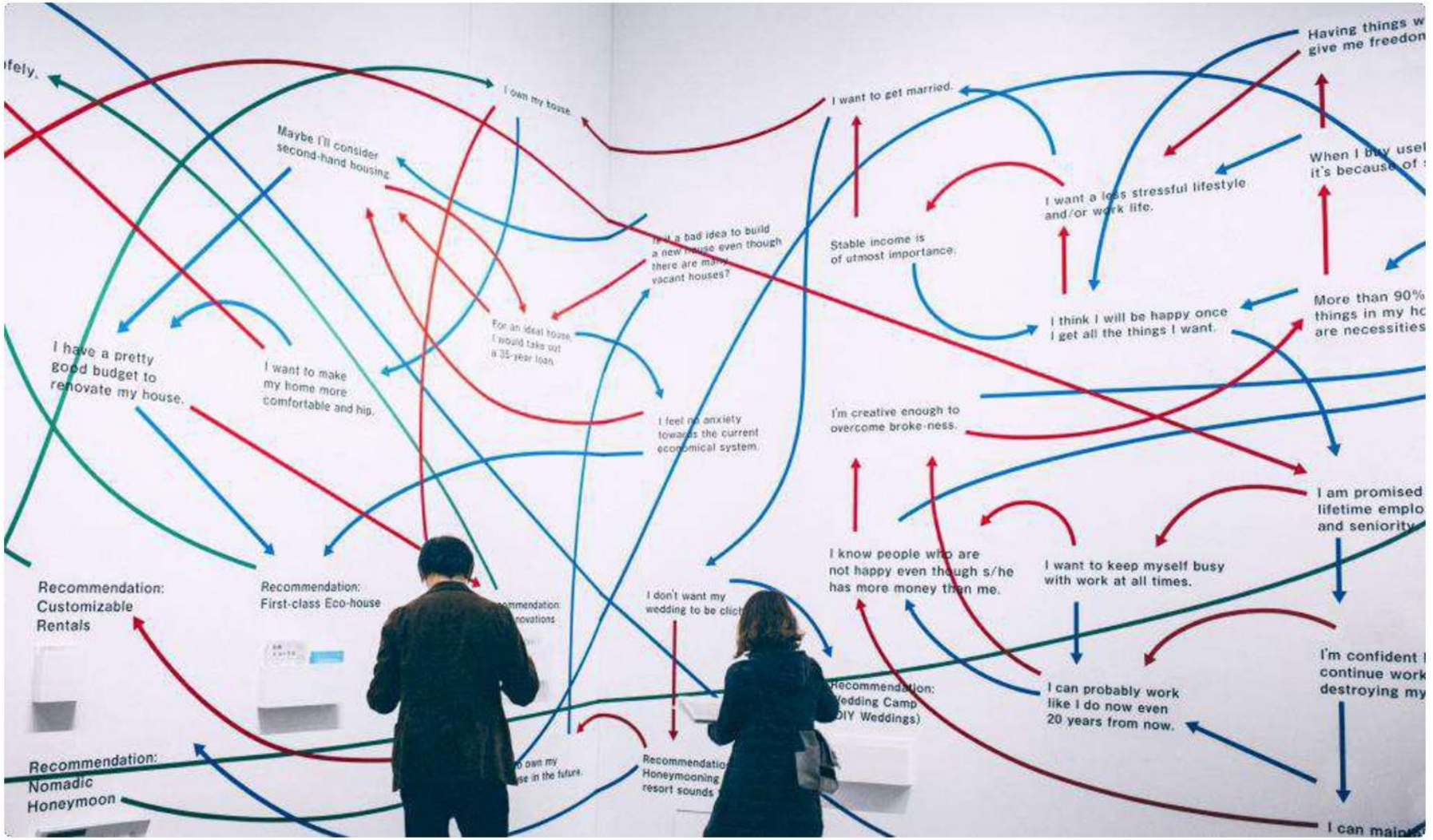
PC di un dipendente in SmartWorking che vuole accedere a stipendi

Si autentica tramite un portale, e a seconda della postura il PEP gli presenta i servizi ai quali può accedere (possono anche cambiare)

Non c'è un controllo diretto sul dispositivo perché non appartiene alla struttura (vale per tutto il BYOD)

- Riceviamo informazioni del client tramite la richiesta (HTTPS o altro)
- Per contenere problemi si può pensare ad implementare un remote browser isolation

Scenari


I own my house.
I want to get married.
Having things w
give me freedon
Maybe I'll consider
second-hand housing.
When I buy usel
it's because of
I want a less stressful lifestyle
and/or work life.
Stable income is
of utmost importance.
I have a pretty
good budget to
renovate my house.
I want to make
my home more
comfortable and hip.
For an ideal house,
I would take out
a 35-year loan.
I think I will be happy once
I get all the things I want.
More than 90%
things in my ho
are necessities
I feel no anxiety
towards the current
economical system.
I'm creative enough to
overcome broke-ness.
I am promised
lifetime emplo
and seniority
I know people who are
not happy even though s/he
has more money than me.
I want to keep myself busy
with work at all times.
Recommendation:
Customizable
Rentals
Recommendation:
First-class Eco-house
I don't want my
wedding to be
I'm confident
continue work
destroying my
I can probably work
like I do now even
20 years from now.
Recommendation:
Nomadic
Honeymoon
Recommendation:
Honeymooning
resort sounds
I can main

Es. 1 – Struttura con IDM molto sviluppato

Le policy di accesso sono basate sugli attributi (molti) dell'IDM

- Facilmente implementabili
- Solitamente un IDM avanzato prevede già anche più meccanismi di autenticazione
- Poi sono affinate dagli attributi aggiuntivi: dispositivi assegnati, mac address, sedi

In questo caso è più semplice il deployment «portal»

Es. 2 – Micro-segmentation

Supponiamo che ci sia una rete, o parti di essa, segmentate o micro-segmentate e protette da NGFW

In questo caso NGFW può fare da Policy Enforcement (PEP), cioè da quel componente che autorizza/nega sulla base delle policy

Alcuni di questi sistemi sono anche in grado di verificare il client asset: non hanno bisogno di un agent sul client

Sono configurabili sia staticamente che dinamicamente

In questo caso il deployment può essere Client Agent/gateway senza agent

Si possono usare FW più scrausi o addirittura FW stateless: difficoltà di configurazione dinamica, difficoltà di adattamento

Es. 3 – Software Defined Perimeter

Rete SDP (Software Defined Perimeter) – Nascondere la rete hardware attraverso software

Rete SDN (Software Defined Network) – Separare la gestione della rete dall'infrastruttura sotto

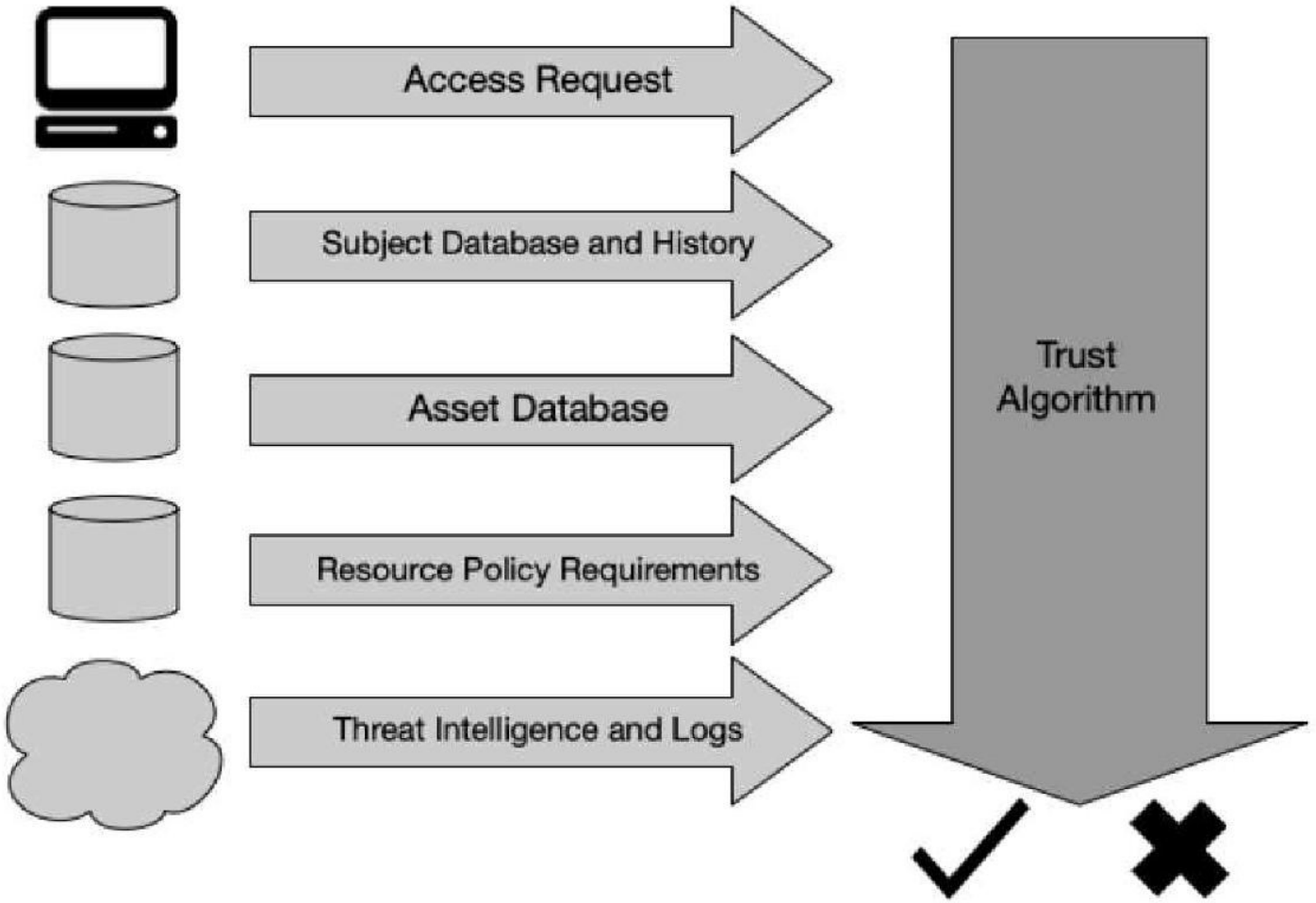
Rete IBN (Intent-Based Network) - Rete che si configura con software di auto-apprendimento

In questo caso il PA è il network controller che riconfigura le rete sulle direttive del PE

Il client si connette al PEP

Essendo una rete principalmente basata sul livello 7, il modello più facile di deployment è quello Client Agent/gateway 7layer

Algoritmi assegnazione TRUST



Algoritmi assegnazione TRUST

Fattori su cui basare il livello di trust di un agent/client/utente

- Richiesta di accesso: Versione OS, applicazione usata, patch level
- DB utenti/attributi/permessi: username, password, geolocation, identità (IDM)
- DB di tutti gli asset dell'Organizzazione: se l'asset usato è nel DB
- Requirement delle risorse: DB dei minimi requisiti per accedere ad una risorsa (es. MFA, IP in blocklist)
- DB di monitoring: qualsiasi informazione sul client che viene da scansioni interne, vulnerability assesment, log di sistema, eventi SIEM

Assegnazione del TRUST

Criteri Statici: per ogni risorsa e per ogni client si stabiliscono i criteri, se durante una sessione fanno match la sessione è autorizzata

A punteggio (score-based): si stabilisce il grado di fiducia assegnando un punteggio ad ogni singola voce e sommando. Si stabilisce una soglia sopra la quale la sessione è autorizzata

Singolari: viene valutata la postura singolarmente di volta in volta che si collega

Contestuali: Viene registrato in un DB lo storico delle sessioni di un utente, per poter segnare un «comportamento base» e distinguere più facilmente le anomalie di comportamento

Requisiti di rete

Conoscere gli asset di proprietà, soggetto, risorsa, e la postura (asset inventory)

L'organizzazione può monitorare tutto il traffico
- Non è possibile ispezionare tutto il traffico L7
- Carpire metadati di connessione (utente, orario, client) per dare feed alle policy

Le risorse non devono essere accessibili se non accedendo ad un PEP
- Il PEP stabilisce il canale (criptato) col client
- Nessun pacchetto escluso quelli dal PEP, neanche i ping (no discovery)

Data plane e control plane (e monitoring plane) sono su reti differenti
- Il PEP sta su data e control: comunica fra il client, il PA e connette alla risorsa
- Il PE+PA sta sul control e ricevono feed di monitoraggio (monitoring panel)

Requisiti 2

I client devono poter raggiungere i PEP
- Via portale web
- Via client/agent
- Device di rete

Ogni PEP è l'unico componente che si collega al policy administrator

I componenti logici (PE, PA, PEP) devono essere previsti «scalabili»
- Aggiunta di pezzi nell'architettura **Zero Trust** dell'organizzazione
- Alta affidabilità, disaster recovery – sono elementi fondamentali della rete

Alcuni PEP devono essere inaccessibili ad alcune classi di utenti (Es. telefonino personale all'estero su procedura stipendi)

Riferimenti

NIST SP 800-207 “Zero Trust Architecture”

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf

Modulo n. 2

Autorizzazione e Autenticazione

Device

Utenti

Agenda modulo n. 2

Possibili scenari di deployment degli elementi di Zero Trust

- Automazione
- Dettaglio di PE – PA – PEP
- Modalità possibili di autenticazione e autorizzazione in Zero Trust
  - Ciclo della fiducia
  - PKI e Certification Authority
  - Gestione dinamica della fiducia
  - Trust dei device

Automazione

Zero Trust non è un nuovo protocollo ne’ nuove librerie o software
Utilizza quello che c’è in maniera diversa

Considera ogni dispositivo come direttamente affacciato su internet
Tutti quanti dentro un ambiente ostile
Dove tutto è forse già compromesso o compromissibile a breve

Siamo bravi a fare sicurezza su un host
Ma richiede impegno e sforzo continuo di monitoraggio
Per estenderlo a tutta la rete abbiamo bisogno di automazione

Automazione

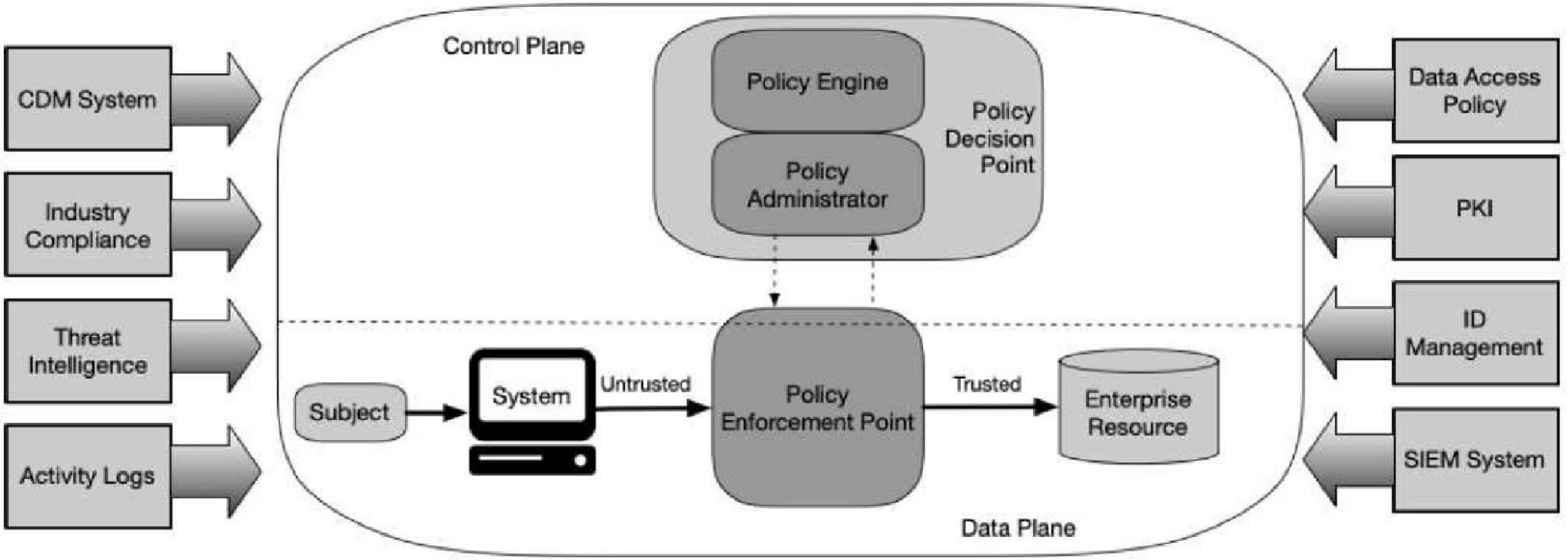
Il punto centrale è la dinamicità del processo di assegnazione del trust

- Acquisizione di dati
- Controllo sul control plane
- Risposta al soggetto
- Creazione del tunnel
- Verifica continua dell'asset esposto
- Eventuale modifica della policy

In questo ciclo continuo è necessaria l'automazione qualsiasi

(Es. configuration management)

Componenti Logiche di Zero Trust



Componenti logiche

PDP – Policy Decision Point

- PE (Policy Engine): assegna il punteggio di trust sulla base delle policy e lo comunica al PA
- PA (Policy Administrator): sulla base delle policy della risorsa autorizza (o no), crea i token per la connessione e configura il PEP col percorso

PEP – Policy Enforcement Point

- Abilita le connessioni
- Crea tunnel criptato fra soggetto e oggetto
- Sta davanti alla risorsa
- Può essere un portale
- Oppure può essere un agent sul client e gateway verso la risorsa

Requirements

Utenti/applicazioni autenticate e autorizzate

- sia «umani» e applicazioni (es. dentro datacenter) devono essere autenticati ed autorizzati per connessione

Device autorizzati

- in modo da creare connessioni end-2-end criptate fra utente/applicazione && device e servizio
- Nella versione perimetrale si faceva con NAC (es. 802.1X e VPN)

Livello di Trust

- Sulla base della postura di utenti/applicazioni e device viene calcolato il punteggio di trust
- Sulla base del punteggio di trust viene applicata una policy pre-definita
- Se autorizzato il Control p. segnala al Data p. di iniziare la connessione
- Il Control p. fornisce gli strumenti di crittazione necessari per creare il tunnel (a livello di utente/applicazione o device o entrambi

Ogni flusso che passa dalla nostra rete è autenticato, autorizzato e criptato

Nota Bene

Tutto quanto sopra favorisce il **monitoraggio**

- Passa molto meno traffico perché tutto quello che non è autorizzato dal Control p. è DROP
- Quello che è autorizzato è molto più semplice da analizzare

In tutto questo ciclo non abbiamo mai parlato di **IP, network, routing**

- Sia il soggetto che la risorsa possono risiedere ovunque in internet
  - Nell'Organizzazione
  - In un access point pubblico
  - In cloud «aziendale»
  - Con il PC dell'ufficio o il portatile aziendale o il portatile mio o il telefonino o chissà cos'altro
  - In cloud remote (i bucket AWS non saranno più «crossabili», neanche da NSA [**])

Quindi tutto questo rende le reti private, i NAT, ma SOPRATTUTTO le VPN obsolete e inutili!

Ci possiamo sbarazzare delle VPN!

Threat Model per Zero Trust

Attacchi occasionali: Script kiddies, automatici, senza target

Attacchi mirati: attacchi mirati verso un target preciso: spare phishing, spionaggio industriale

Insider threats: credenziali rubate di utenti non privilegiati

Trusted insider: credenziali rubate di account privilegiati

State-level: attacchi supportati da governi nazionali (il proprio o esteri)[non è contemplato da ZT]

Il threat model è quello di **Internet:** sia soggetto che risorsa è esposto ovunque su internet

RFC 3552 – Internet Threat Model – https://tools.ietf.org/html/rfc3552

NB: Zero Trust non si applica neanche alla DMZ perché per definizione deve essere acceduta ad tutti senza autorizzazione

Cosa stiamo dando per scontato

1. I vecchi sistemi di controllo delle reti non sono eventualmente da buttare
   - ACL su router di bordo, firewall vari, NAT, reti private – TENERE L'ANTISPOOFING
2. Gli endpoint (client e server) possono essere compromessi (ZERO TRUST!)
   - Se sono dell'organizzazione è necessario che siano censiti e patchati
   - Abbiano antivirus e soprattutto EDR – oppure sistemi di hardening e controllo sui server
   - Le credenziali devono essere lunghe e ruotate con cadenza opportuna
   - Devono essere monitorati per distinguere il comportamento anomalo
3. Ogni entità che accede alla nostra rete deve essere monitorata
   - Deployment del Monitorng plane
   - Su rete separata e inaccessibile
   - Sistemi di detection, alerting
   - SIEM, SOC, Incident Handling etc etc

Ciclo del Trust – Comunicazioni criptate e PKI

Zero Trust si basa fortemente sullo strato TLS/SSL e PKI nell'autenticazione e autorizzazione

I device si autenticano prevalentemente con un certificato

Le applicazioni si autenticano al sistema tramite certificato (semplice gestire e automatizzare)

Gli utenti si possono autenticare in maniera forte con certificati personali

I certificati possono contenere dati e informazioni aggiuntive utili alla valutazione del trust

Il PEP una volta ottenuta l'autorizzazione dal PA stabilisce connessione criptata fra client e server mediante token o certificati a vita breve prodotti dal PA

La creazione e gestione dei certificati digitali diventa cruciale nell'organizzazione

Si consiglia come best practice di avere una propria CA interna automatizzata e governabile via API dal Control Panel per la firma e il provisioning dei certificati

Ciclo del Trust – Principio del Minimo Privilegio

- Ad un soggetto devono essere garantiti soltanto i privilegi che servono per svolgere il lavoro. Esempio: solo il personale DEVOPS può accedere al codice di sviluppo
- Nel caso di una applicazione deve essere eseguita da un service account non root, in container, jail, e tutto quello che si può fare per limitare le possibilità soltanto al lavoro che deve svolgere
- Anche nel caso di un device si devono assegnare i privilegi che competono al device. Nel caso di un PC per esempio i privilegi di accesso del device sono gli stessi dell'utente

- L'accesso privilegiato spesso serve per un tempo limitato, non sempre – si consiglia di prevedere dei meccanismi che chiedono ulteriore autenticazione
  - `sudo` in Linux – anche in versione GUI

Ciclo del Trust – Principio del Minimo Privilegio

Zero trust prevede l'interdipendenza di device ed utente per calcolare il suo livello di trust.

- Mario Rossi accede normalmente al wiki interno

  Per accedere agli stipendi viene richiesta autenticazione aggiuntiva – più forte
- Mario Rossi accede con autenticazione forte agli stipendi ma solo quando è al PC aziendale

  Se prova ad accedere agli stipendi dal telefonino gli viene negato il privilegio – oppure autenticazione ancora più forte
- Mario Rossi accede alla procedura stipendi durante i giorni lavorativi 8-18 ma viene richiesta autenticazione aggiuntiva per connessioni fuori dall'orario d'ufficio
- Mario Rossi accede adesso dall'Italia al wiki aziendale, nello stesso momento cerca di accedere dal Giappone – DROP e allarme! Credenziali rubate

Ciclo del Trust – Principio del Minimo Privilegio

Si può essere molto granulari nell'assegnazione del minimo privilegio

L' autorizzazione ad usufruire di un servizio non è binaria ma è un ventaglio (trust score) dato da diversi fattori

La situazione di Mario, il suo device, altri attributi (spaziali e temporali), la «criticità» del servizio

Ad ogni attributo di Mario e del suo device assegniamo un punteggio (es. da 0 a 10)

- SmartPhone di Mario 2
- PC ufficio di Mario 9 (questo dipende dallo stato del PC, se e' aggiornato etc etc)
- Mario in ufficio 10
- Mario in asia 0
- Mario dalle 8-16 10
- Mario domenica 1

Il PE sulla base della richiesta (che dispositivo, dove sta, che ora e'), gli da' un punteggio (es. 63)

Il PA, sulla base della risorsa, sa che il punteggio deve essere superiore a 70 per passare

A questo punto chiede autorizzazione aggiuntiva (es. token) oppure nega

Gestione del trust e policy «vecchia maniera»

Scegliere chi può accedere ad una risorsa in base all'identità e al device a mano è impensabile

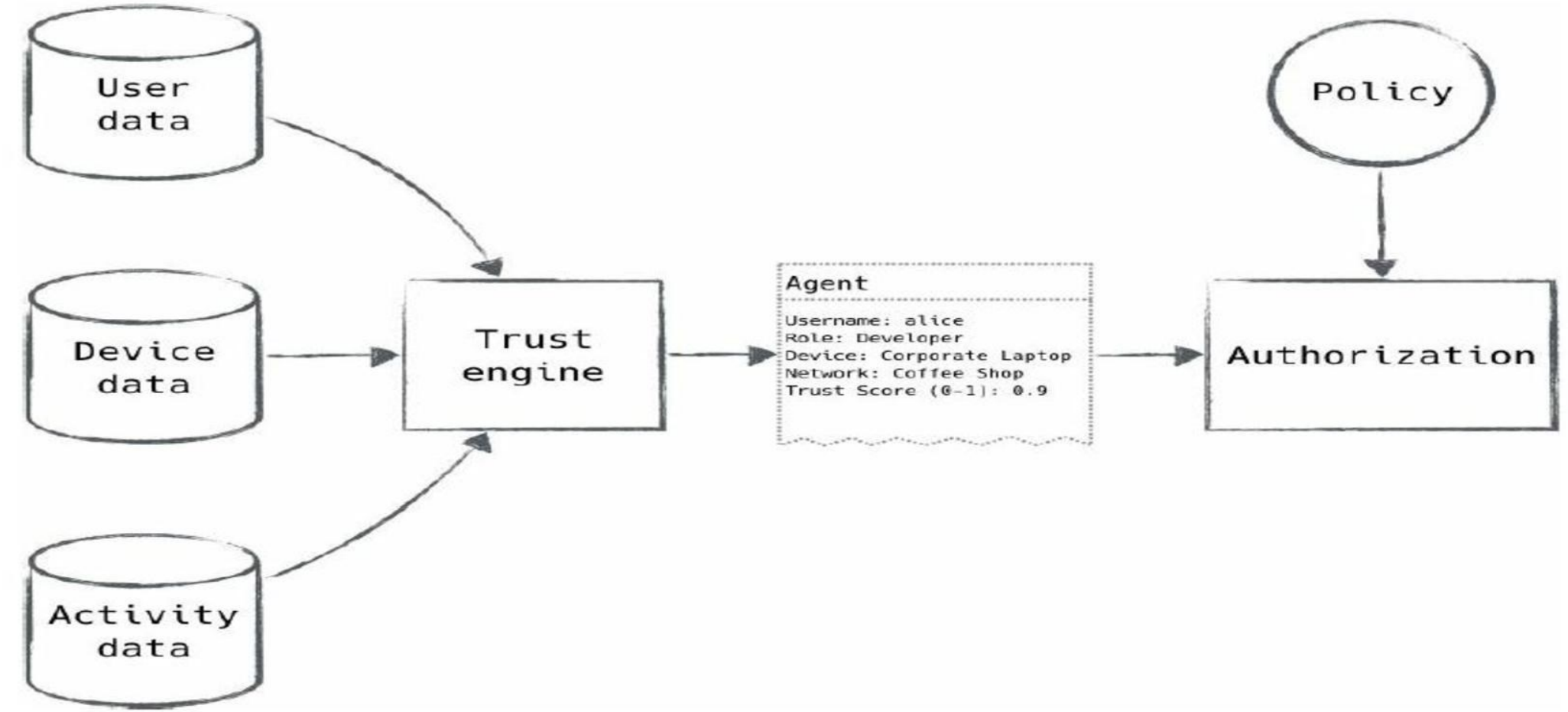
Assegnare privilegi sulla base di ruoli crea un pool di utenti super privilegiati che apre brecce

- Difficile manutenzione
- Difficile valutare l'impatto di un cambiamento
- Difficile o impossibile fare scelte «dinamiche»

Eppure assegnare fiducia in base a policy dinamiche è una cosa antica che facciamo tutti i giorni

Per esempio, un altro settore in cui regna lo Zero Trust: le compagnie di assicurazioni

Assegnazione del trust score



Lo score

Utente
- Chi è e come si è autenticato (password, token, certificato etc)
- Il ruolo o la posizione nella struttura
- Da dove si connette – posizione geografica
- Orario/giorno della settimana

Device
- Device personale o dell'organizzazione
- Versione del sistema operativo
- Versione del client

Comportamento aspettato/normale/storico
- Nuovo device
- Nuova location
- Nuovo orario/giorno

Processo dinamico di scoring

Più le informazioni sono dettagliate più possiamo essere granulari sul punteggio

Alla fine avremo un numero da confrontare con la policy
Dipende di requirement risorsa a cui si vuole accedere
La policy per esempio dice che: per accedere agli stipendi è necessario uno score di almeno 63

Se la condizione è verificata l'utente è autorizzato ad accedere alla risorsa che ha chiesto
Se la condizione non è verificata si può negare e droppare (e loggare)
Si può mandare una mail di avviso, o una notifica push: sei tu? Altrimenti cambiati la password
Oppure si può chiedere una forma ti autenticazione aggiuntiva più forte della precedente
L'utente si autentica con l'autenticazione richiesta, viene RI-calcolato lo score >63 -> Autorizzato

Sviluppi futuri

Al momento si cerca di recuperare più informazioni possibili sull'utente e sul client
Questo dipende da quello che abbiamo implementato nell'organizzazione
Sistemi di discovering, scanning della rete, monitoraggio, vulnerability assessment

Il modo di catalogare ed organizzare queste informazioni può essere qualsiasi
Al momento la best practise è uno o più db per ogni ambito: utenti, device, applicazioni, eventi

Per quanto riguarda il formato con cui trasmettere le informazioni non esiste uno standard
La condizione ottimale sarebbe sviluppare uno standard simile a SMNP con gli OID
Le soluzioni attualmente usate utilizzano perlopiù JSON

Processi decisionali di autorizzazione

«Agent»: combinazione di utente e suoi attributi + device e suoi attributi

Questa combinazione è la stessa che viene usata dagli «agent» software nell'architettura ZT NIST di tipo agent/gateway, ma la definizione va bene anche per le architetture «a portale»

Ogni richiesta di connessione è caratterizzata da un proprio agent

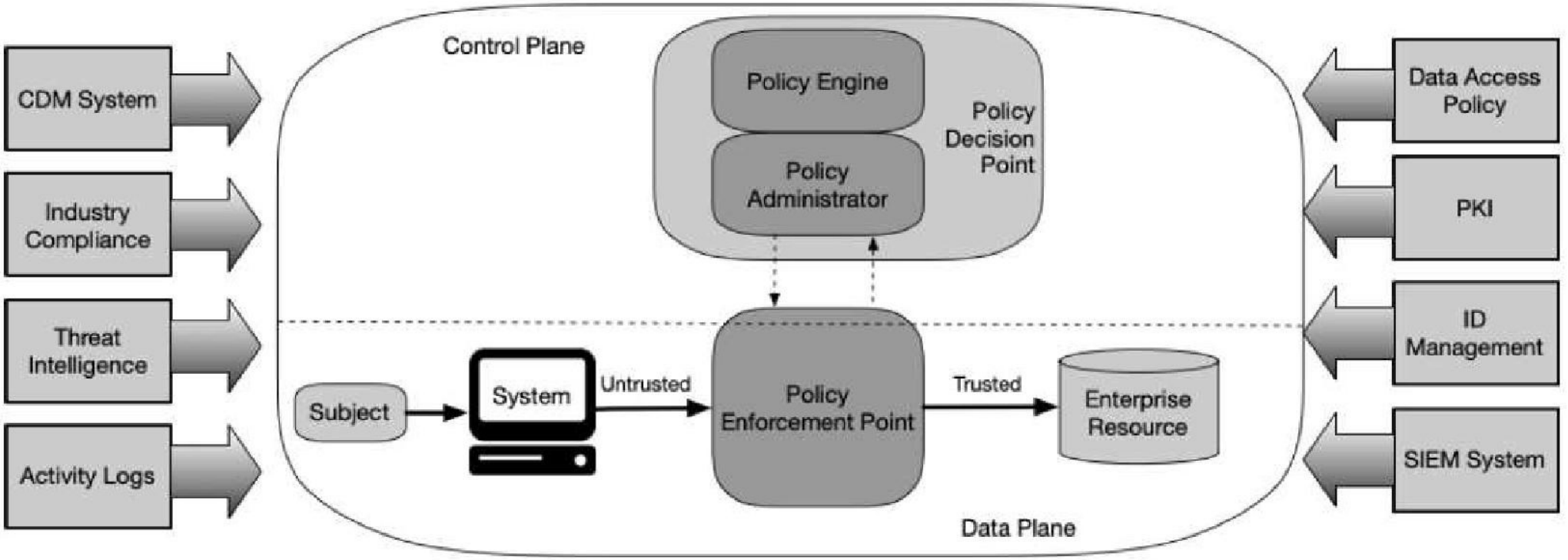
Ogni agent determina un proprio score sulla base delle policy

L'agent è definito e lo score determinato a seguito dell'autenticazione

Agent e score servono solo per l'autorizzazione

Adesso vediamo nel dettaglio come si può implementare il processo per decidere l'autorizzazione

Reminder


CDM System
Industry Compliance
Threat Intelligence
Activity Logs
Control Plane
Policy Engine
Policy Administrator
Policy Decision Point
Subject
System
Untrusted
Policy Enforcement Point
Trusted
Enterprise Resource
Data Plane
Data Access Policy
PKI
ID Management
SIEM System

Il Policy Enforcement Point - PEP

Il PEP sta più vicino possibile alla risorsa e in Zero Trust può essere

- Load balancer
- Reverse proxy
- NG Firewall
- Un Identity Aware Proxy (BeyondCorp di google o Akamai Zero Trust)
- Firewall più scrauso (ma difficile da far diventare «zero trust» compliant)

E' quello esposto alle richieste esterne ed è quello che «parla» con il Control panel (attenzione!)

Si occupa di far autenticare l'utente – Quando l'utente è già autenticato

Raccoglie i dati dell'agent e li trasmette al Policy Administrator per ottenere autorizzazione

Se autorizzato il PA consegna i token di autenticazione o certificati di connessione

PEP 2

A questo punto il PEP deve creare un path fra agent e risorsa e può farlo in due modi:

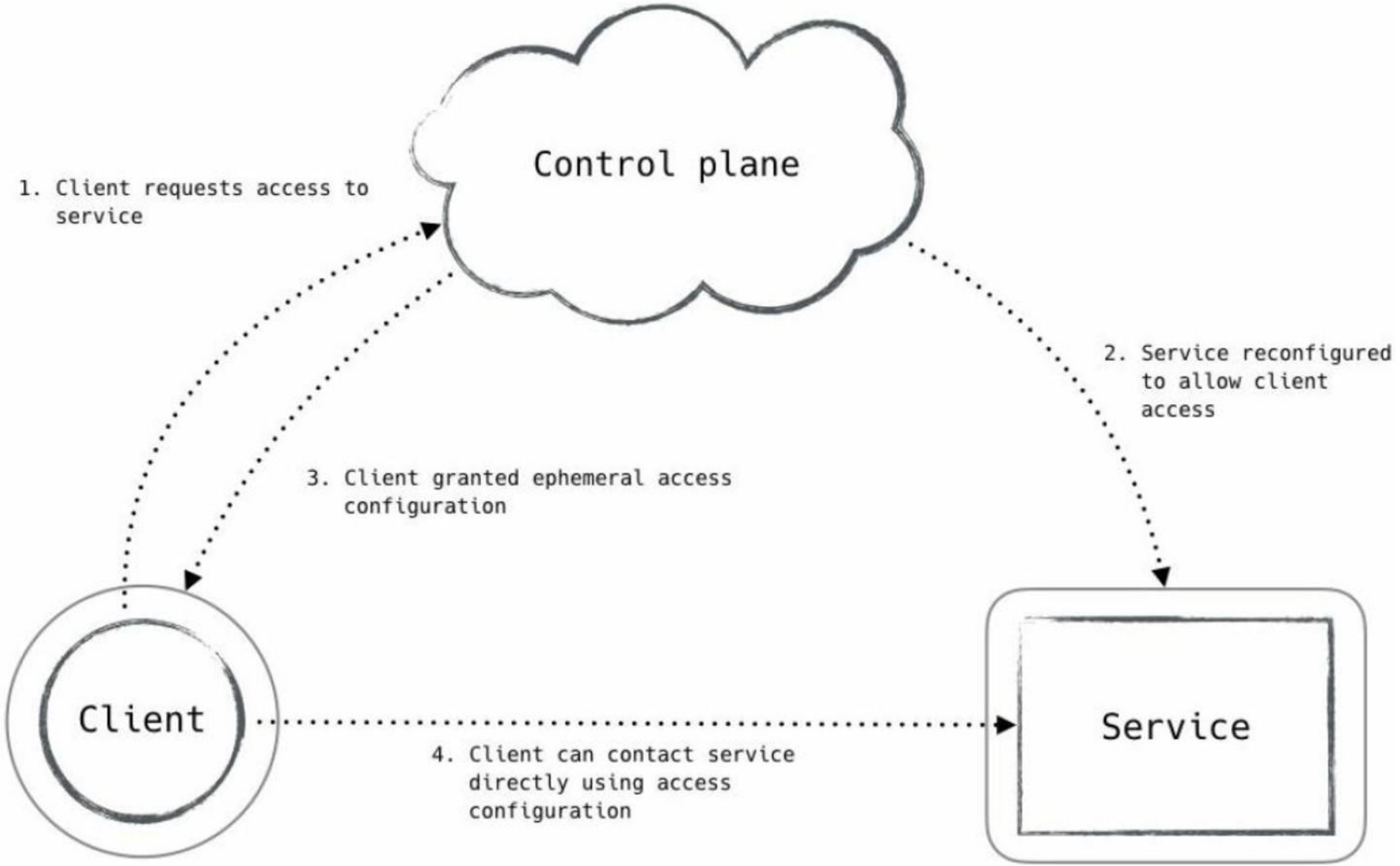
Se è un proxy identity aware (IAP di BeyondCorp – google o Akamai) si occuperà semplicemente di stabilire una connessione Layer7 proxy HTTPS con la risorsa

Nelle situazioni più tradizionali (o se non abbiamo un IAP) sarà necessario che il PEP riscriva le regole per aprire il proxy o il firewall e stabilisca un tunnel fra agent e risorsa

La cosa bella del PEP è che può essere leggero e virtualizzabile e replicabile

E' facile implementarli di fronte a tutte le risorse in Zero Trust

Control Panel e Data Panel



Policy Administrator - PA

Si occupa di prendere i dati dell'agent forniti dal PEP

Sulla base della risorsa chiede al PE di fare il conto solo su alcuni dati

Confronta lo score con le policy per quella risorsa

Autorizza o nega o chiede autenticazione aggiuntiva

Se nega o autenticazione aggiuntiva -> informa il PEP di chiudere la connessione o chiedere ulteriore autenticazione

Se autorizza -> genera un token per la connessione o tramite API interroga la PKI per procurarsi certificati a vita breve e consegna tutto al PEP

(Non ricorda un po' il RADIUS? :])

Per assegnare lo score chiede al PE (Policy Engine) di valutare l'analisi dei rischi della richiesta

PA 2

Il PA deve tenere lo storico di

- la lista di tutte le policy di assegnazione dello score
- La lista di tutte le policy di accesso alle risorse per ogni risorsa

La best practice raccomanda che siano immagazzinate secondo version control system

- Le variazioni possono essere tracciate
- Viene inserita la motivazione della modifica. E' come avere un piccolo DEVOPS.
- Può essere inserito un meccanismo di approvazione della policy manuale o automatico
- La definizione delle policy può essere delegata ai gestori della singola risorsa e validata col revision control system dal team di sicurezza

Le policy non devono far riferimento a IP ma a servizi, risorse, agent -> possono cambiare

Non esiste standard ne' per il tipo di policy ne' per la forma, il JSON è il più utilizzato

Policy JSON «alla Kubernetes»

```
metadata:
 name: test-network-policy
 namespace: default
spec:
 podSelector:
  matchLabels:
    role: db
 ingress:
  - from:
     - namespaceSelector:
        matchLabels:
         project: myproject
     - podSelector:
        matchLabels:
         role: frontend
```

Policy Engine - PE

E’ il cuore di Zero Trust. E’ lo strumento che fa l’analisi del rischio di ogni singola richiesta e assegna lo score

Es. Può accedere ai dati di inventario del dispositivo: ultime patch, funzionalità speciali

Decidere il fattore di rischio è molto difficile

Si possono stabilire regole del tipo:

- Se il dispositivo non ha le ultime patch abbasso il punteggio
- Se i tentativi di login falliti sono troppi in un tempo limitato diminuisco il punteggio

Regole statiche non sono sufficienti a difendersi da attacchi imprevisti

In reti e organizzazioni mature si usa AI o ML per far imparare

Nei deployment Zero Trust che conosciamo, perché hanno fornito documentazione di implementazione, si usa una combinazione di autoapprendimento e punteggio statico

Policy Engine – PE

Altri fattori problematici nella valutazione del rischio:

- Punteggio unico per l'agent (utente+dispositivo).
  Es. tentativi ripetuti di connessione alla risorsa X: screditare l'agent che fa i tentativi (probabilmente malevolo) e lasciar operare l'utente vero, che avrà fornito credenziali giuste da un altro dispositivo
- In altri casi è più opportuno valutare separatamente utente e dispositivo.
  Es. Computer con traffico sospetto o con bassa reputazione o (al limite) BYOD
  Un utente X che salta fisicamente da un chiosco all'altro nel tentativo di comprometterli

Quello che viene consigliato è valutare anche l'agent in se', non come semplice somma del punteggio utente+punteggio dispositivo

NB: esporre all'utente punteggio ottenuto in verdetto è esso stesso fattore di rischio

Feed di verità

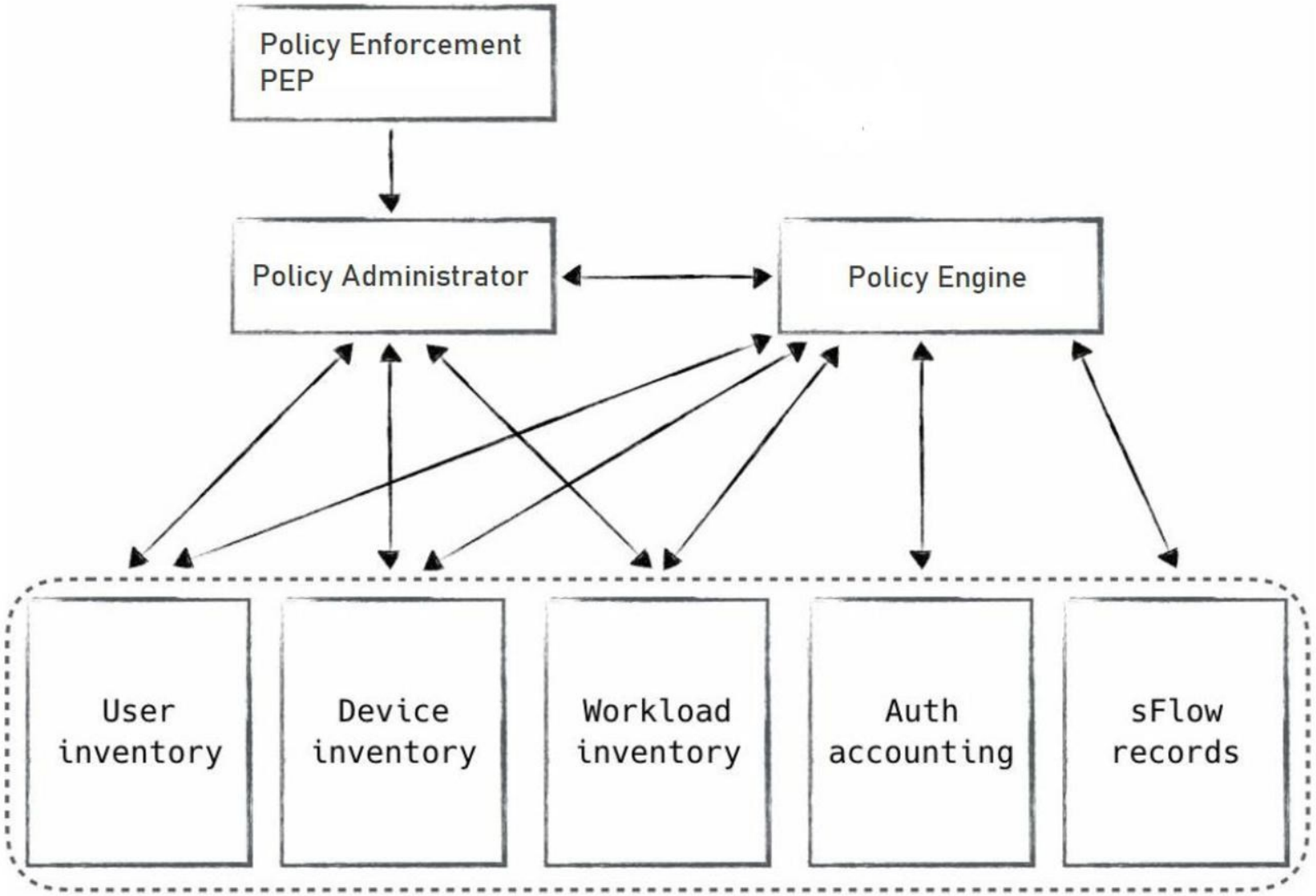
Per stabilire sia il verdetto che le policy in maniera automatica il sistema accede ad archivi che sono le sue fonti di verità attraverso API o feed. Solitamente stanno sul Monitoring plane

- Dati di inventario: inventario di utenti + attributi, inventario dispositivi, inventario applicazioni
- Log delle connessioni registrato dal PA
- Log dei vari sistemi di monitoraggio sulla rete, SIEM, IR, ThreatIntelligence, SOC etc

Possono cambiare il verdetto di autorizzazione praticamente in tempo reale

Garantiscono un elevato aumento della sicurezza

Feed per autorizzazione



Trust dei device

- Identificazione del device: certificato
  - Provisioning e firma
    - Gestione umana (casi umani)
    - Gestione automatizzata con umani
    - Auto-scaling automatico (macchine virtuali)
    - Dispositivi legacy
  - Chiave privata

- Inventario
- Inventario «esteso»
- Introduzione sicura in rete (la prima volta)
- Rotazione device e possibili alternative alla rotazione
- Trust score per device

Certificato per device

In Zero Trust il certificato per un device è fondamentale

- Identità - Per identificare il device in maniera univoca
- Per inserire informazioni aggiuntive utili in fase di autorizzazione (Campi O, OU, ST etc)
- Per collegare il dispositivo all'inventario
- Per stabilire connessioni criptate fra agente e risorsa
- Per fare «introduzione sicura» - la sua prima volta in rete

Approvazione, provisioning e firma

L'approvazione di un certificato è un'operazione molto delicata

In Zero Trust è fondamentale

Il certificato per essere approvato deve essere

- Conforme agli standard dettati dal team tecnico ed ereditati dalla CA
- Deve contenere tutti i campi che ci aspettiamo
- I campi devono essere corretti (es. se sta nell'edificio X poi ci deve stare)
- L'ultima parola che serve è una dichiarazione che effettivamente quel certificato è stato creato da/sul/per quel device

Casi d'uso

Se il sistema è piccolo per ogni device può pensarci il team tecnico (o della CA)

Se il sistema è più grande si può comunque continuare ad utilizzare l'intervento umano formando delle RA (Registration Authority) umana e delegando a loro l'approvazione dei propri certificati (es. dipartimenti)

Oppure fornendo direttamente al gestore della macchina una TOTP da utilizzare per chiedere la firma

(l'umano gestore della macchina deve essere «trusted» con un suo certificato)

Automazione del provisioning e firma

Se il sistema è molto grande questo processo può essere ulteriormente automatizzato

Questo è molto utile nei casi di macchine virtuali, container

La fiducia nel processo di approvazione di un certificato in Zero Trust può venire da tre fattori

- Umano
- Resource manager
- Immagine software del dispositivo

Il «caso umano» l'abbiamo già visto

Il caso del Resource manager può essere duplice:

- il resource manager è guidato da un umano nel creare macchine (caso prec.)
- Il resource manager è una macchina (host di virtualizzazione, container)

Provisioning e firma per macchine virtuali

Il resource manager ha un punto di vista privilegiato: crea e sorveglia tutte le macchine

Che succede però se il Resource manager è compromesso? O la macchina che chiede il certificato? (Zero trust!)

Ci basiamo anche sull'immagine della macchina di cui si chiede il certificato, che è registrata nei nostri sistemi per altre vie (Es. inventario) con una chiave unica

- Chiave immagine
- Indirizzo IP corretto
- Proprietà del certificato (campi O, OU etc)
- TOTP valido

Provisioning e firma per macchine virtuali 2

La coppia resource manager + chiave dell'immagine creano sufficiente trust per l'approvazione

- In questo caso se il RM viene bucato, non può comunque accedere alla macchina per fare richiesta di certificato
- Se viene bucata la macchina può generare il certificato e farselo approvare, ma non ha la chiave dell'immagine
- Se viene rubata la chiave dell'immagine non si può ne' generare il certificato ne' RM può approvarlo

Dispositivi legacy – Zero Trust Supplicant

Non tutti supportano X.509, ne' hanno modo di installare dentro di se' un certificato:

SCADA, videocamere di sorveglianza, alcune stampanti, dispositivi IoT in genere

Si ricorre a proxy di autenticazione che si occupano anche della parte di richiesta, gestione e presentazione del certificato alla rete e ai servizi

In questo caso è necessario che l'authentication proxy sia il più vicino possibile al SINGOLO dispositivo

Nel futuro potrebbe esserci spazio per l'implementazione di supplicant: device TPM con due interfacce, una verso il dispositivo legacy e una verso la rete

Ma al momento è fantascienza

Il problema della chiave privata

La chiave privata di un dispositivo va protetta con sistemi opportuni

Se il dispositivo è un dispositivo «utente» (PC) è sufficiente criptarla con una password

Nel caso di server non è possibile inserire la password ad ogni riavvio o restart del servizio, andrebbe memorizzata sul dispositivo stesso

Questo pone problemi di furto della chiave

In situazioni ad elevato rischio può essere opportuno memorizzare la chiave privata su dispositivi crittografici esterni: TPM

NB: nel caso di macchine virtuali non esistono TPM, in xen esiste un vTPM ma ancora è giovane

Inventario

Le informazioni sui device devono essere inserite in un inventario

Solitamente in una struttura possono esserci più inventari: quello istituzionale, quello creato dai vari configuration manager su un server centralizzato (ce l'avete un configuration manager?)

In inventario devono andare tutti i device di proprietà dell'organizzazione

- Device per utenti
- Server
- Macchine virtuali
- Apparati di rete
- Dispositivi di proprietà: IoT, sonde, telefonini di servizio

Per ognuno di questo è necessario inserire più informazioni possibili

Inventario «esteso»

In realtà molto grandi, o come prima migrazione a Zero Trust è necessario automatizzare

Si possono sfruttare «configuration Manager» per esempio Chef, Puppet, Ansible e CFEngine

- Non sono applicazioni per inventario, ma raccolgono tanti dati del device (Chef circa 1500)
- Compiono controlli sulle versioni del software, del firmware
- Possono essere usati per stabilire criteri di punteggio

NB: I configuration manager si basano su agent che girano sui device. Se il device fosse compromesso (zero trust!) potrebbero iniettare dati falsi

L'inventario istituzionale e il provisioning (umano o resource manager) sono le fonti autorevoli dei dati, non devono poter sovrascrivere modificando i dati fondamentali del device (Ruolo, IP, chiave pubblica). Sono dati aggiuntivi che servono come suggerimento per alimentare i sistemi decisionali.

Solitamente hanno un version control system che aiuta a ricostruire lo storico

Introduzione sicura del device

La prima volta che accendiamo un dispositivo nuovo ha ZERO TRUST, deve guadagnarsi fiducia

Il processo di guadagnarsi fiducia la prima volta si chiama «introduzione sicura»

- Golden image: caricare sui dispositivi una immagine nota e buona
- Registrare immagine e data ogni volta
- Boot Sicuro – password, Secure Boot
- Registrazione in inventario con le caratteristiche, posizione, responsabile
- Creazione del certificato

Per le macchine virtuali può esser fatto automaticamente con configuration manager che raccoglie i dati sui nuovi device verso un server centralizzato

Rotazione dei device

Per mantenere l'introduzione sicura è necessario stabilire cadenze in cui le condizioni di inserimento sicuro debbano essere rinnovate

Per i server è facile: di solito hanno una vita limitata e ogni tot anni si cambia tecnologia

Per le macchine virtuali è immediata: basta distruggerla e rifarla

Per i device utente è molto complicato

- Personalizzazioni
- Programmi aggiuntivi
- Backup corposi e lunghi

Spesso sul PC nuovo vogliono la copia specchio di quello vecchio...

Cercare una cadenza che possa andar bene senza creare troppo disagio agli utenti

Se non si può ruotare frequentemente

Ci sono vari sistemi per controllare lo stato di un device e capire se necessita di una rotazione

Locali:

Ci sono sia hardware (basati su TPM) che software: antivirus, EDR

L'EDR è importante perché aiuta a registrare lo storico del comportamento normale

Aiuta nel caso di processi e connessioni anomale sconosciute o non rilevate da AV

Rischio: se la macchina è compromessa sono inutili

Remoti: vulnerability assessment periodici

(patch management, VDI, workstation remote, remote browser isolation)

Idee di utilizzo dei dati di inventario device

Un dispositivo di un ingegnere (Ruolo) che cerca di collegarsi alle risorse umane (Ruolo)

Un utente che non si connette da oltre un anno da quel device (last seen)

- Magari era vera, gli abbassiamo il trust in modo che possa accedere alle risorse meno sensibili

L'inventario oltre a fornire indicazioni sull'autorizzazione di un device crea anche tanto contesto per il trust di un utente

Indicatori di trust di un device (suggerimenti)

- Tempo dall'immagine disco primordiale (golden image)
- Ultima volta visto online
- Dispositivo vecchio verso nuove risorse («screditare» le richieste verso nuove risorse)
- Aumento insolito di richieste del dispositivo ad una risorsa
- Improvvisi cambi di posizione di un device che si autentica
- Modifica nel traffico di rete del device rispetto alla norma
- Segnalazione alert di sicurezza da parte dei sistemi di monitoraggio

Conclusioni

- Abbiamo visto nel dettaglio come si assegna la fiducia in una rete Zero Trust partendo da zero
- Grazie a sistemi in grado di prendere decisioni e stabilire policy dinamiche (il cuore di ZT)
- Con l'aiuto di un sistema di monitoraggio in grado di provvedere feed di verità
- Abbiamo infine stabilito criteri di assegnazione del trust per device

Riferimenti

NIST SP 800-207 "Zero Trust Architecture"
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf

RFC 3552 – Internet Threat Model – https://tools.ietf.org/html/rfc3552

Eval Gilman e Doug Barth, Zero Trust Networks, 2017, O'Reilly Media

Modulo n. 3

Esempi

Integrazione

Migrazione

Possibili soluzioni/scenari

Agenda modulo n. 3

- Criteri di trust degli utenti
- Criteri di trust del traffico di rete (creare flussi criptati)

Esempi implementazioni Zero Trust: Google BeyondCorp

Suggerimenti per migrare a Zero Trust

Carrellata (ad alta quota) su quello che offre il mercato

User Experience

Conclusioni

Trust degli utenti

- Trust sulla identità dell'utente (prima immissione)
- Dati e database per autenticazione
- Modalità di autenticazione
  - Qualcosa che sai – Password
  - Qualcosa che hai – Token, Certificati
  - Qualcosa che sei – dati biometrici
  - Out-of-band
  - SSO
  - Autenticazione locale
  - Autenticazione per gruppi (Regola delle tre firme)

- Trust score per utenti

Trust sull'identità

- Il dispositivo non basta per autenticare verso una risorsa: perdita, PC incustodito etc
- Le persone hanno più dispositivi per accedere ai servizi

- E' necessario che l'utente dimostri al sistema di essere lui

Costruire una identità digitale è un compito delicato e non così banale

Dall'affidabilità della prima immissione deriva tutta la catena di fiducia che possiamo dare all'utenza digitale – «immissione sicura» vale anche per gli utenti!

Ci sono identità stabilite in via informale e autorevoli

Zero trust richiede identità autorevoli

Trust sull'identità 2

La prima volta che un utente arriva, ha Zero Trust – Dobbiamo fare «immissione sicura»

Ci affidiamo ad una autorità (il governo) che mediante l'emissione di documento di riconoscimento certificato garantisce che il tizio davanti a noi sia il tizio del documento

Costruiamo quindi la catena di fiducia su una forma forte di verifica della sua identità

- Garantita dal governo
- Verificata con la persona davanti ai nostri occhi (uff. assunzioni, segreterie studenti etc etc)

Il riconoscimento/attestazione di identità «de visu» la prima volta è fondamentale

Magari seguita da controlli

E' la forma più forte per creare la sua identità e stabilire fiducia nella sua identità

Le forme di immissione sicura da remoto (documenti via mail etc) sono fortemente sconsigliate

Memorizzazione identità

Directory utente (LDAP): registrazione centralizzata di tutti gli utenti

In zero trust si devono memorizzare centralmente anche dati estesi (geolocation, certificato)

I dati devono essere sempre aggiornati

- Se ci sono più metodi di immissione decidere quale sia quello autorevole
- Mantenere allineati tutti i sistemi di inserimento

Segmentazione dei dati

I dati non sono segreti, ma sono sensibili per l'autorizzazione e a rischio privacy

La best practice dice di immagazzinare i dati in database diversi –
segmentazione dei dati

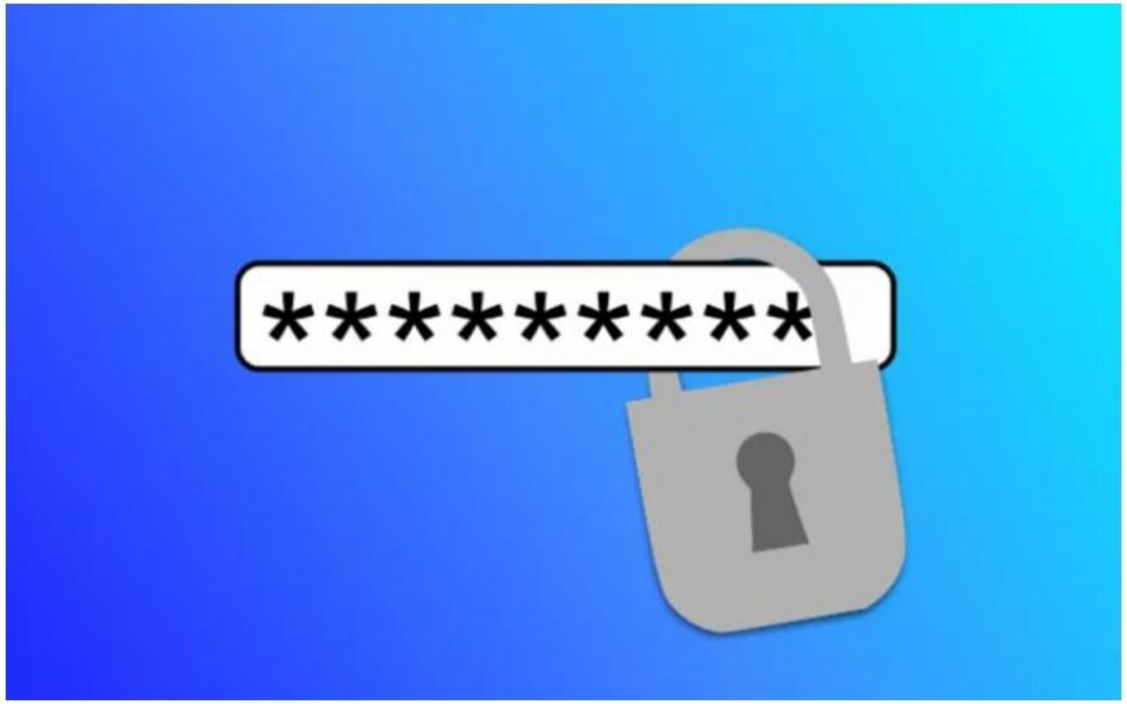
- Un DB primario a cui ci si collega dall'esterno per l'autenticazione
  Deve contenere/esporre la minima quantità di dati possibile e delegare sotto
  Es. username, Nome Cognome, mail, *dati richiesti per il recupero della password*

- I DB con le informazioni estese espongono API generiche senza divulgare dati dell'utente
  Es. L'utente si trova in Italia? (non: dammi le coordinate dell'utente e faccio il confronto io)
  Con che frequenza l'utente cambia posizione? (non: dammi le ultime 5 posizioni con orario)

Modalità di autenticazione

Qualcosa che sai

Password



Modalità di autenticazione

Qualcosa che hai

Token OTP (TOTP)

Modalità di autenticazione

Qualcosa che sei

Dati biometrici

Password

- Lunga minimo 8 caratteri (da standard)
- Massima a piacere (oltre i 20) – Dipende però dalle applicazioni!
- Frase semplice da mandare a memoria con caratteri speciali
- Chiamiamola ***passphrase***
- Consigliati per gli utenti i gestori di password

- Mai registrare le password in chiaro nei nostri DB
- Utilizzare hash sufficientemente robusti
- Gli standard per gli hash cambiano in funzione delle più potenti capacità di calcolo
- Il NIST tiene un elenco in continuo aggiornamento degli algoritmi di hash
  https://pages.nist.gov/800-63-3/sp800-63-3.html

Token

- Password monouso a vita breve (RFC 6328)
- Può essere hardware o software

- Sconsigliato mandare i token via SMS - passano in chiaro e possono essere letti da app

- Il token si basa su una chiave scambiata fra client e server
  - Mantenuta segreta
  - Scambiata su un canale di comunicazione sicuro
  - Sul server può essere criptata tramite un TPM
  - Sul client (spesso lo smartphone) non può essere criptata

ATTENZIONE! Il token può essere intercettato o il telefono compromesso!!

Certificati

- Il certificato utente può contenere dati aggiuntivi utili per l'autorizzazione (campi O, OU, etc)
  - Può essere usato il certificato da solo senza doversi attaccare al DB per prendersi dati aggiuntivi
- Attenzione alla conservazione della chiave privata – deve essere criptata
- Sui client possiamo utilizzare un token hardware (token USB, smartcard, Yubikeys)
  - Serve a depositare credenziali e tutti i certificati
  - Genera esso stesso le CSR dei certificati e conserva le chiavi private
  - Non è accessibile al sistema operativo se non tramite API non invasive
  - Le credenziali non possono essere trafugate ne' il client può sottomettere CSR senza token
  - Può essere rubato però
- Nuovo (futuro) protocollo standard: Universal second factor – U2F – alternativo a X.509

Dati biometrici

- Impronte digitali
- Impronte di mani
- Scansioni della retina
- Analisi vocale
- Riconoscimento facciale

Aspetti negativi

- A volte sono replicabili (deep fake sia audio che video)
- Lo scanner è ingannabile
- Le impronte vengono sparpagliate ovunque
- Non possono essere ruotate
- Possono esserci impedimenti all'utilizzo
- Problemi legali

Autenticazione Out-of-band

Usa un canale separato di comunicazione per autenticarsi

- Accedo ad un servizio per la prima volta – telefonata
- E-mail passiva: ti sei loggato da questo indirizzo, sei tu? Altrimenti cambia password
- E-mail attiva: hai tentato di loggarti da questo indirizzo, se sei tu clicca qua
- Notifiche push sul telefono: sei tu? Si/no
- Contattare (addirittura) terze parti

Autenticazione SingleSignON (SSO)

- SAML
- Kerberos
- CAS

Tecnologia matura e ampiamente usata – una sola password per numerosi servizi

All'autenticazione viene consegnato un token di sessione che dura «per sempre»

ATTENZIONE! Va contro i principi Zero trust!!

Zero Trust richiede una autenticazione decentralizzata attraverso il PEP sul control plane che inietta sul server le credenziali stabilendo un canale end-2-end client-server

Passare SSO in Zero Trust

Far assegnare i token SSO dal PEP invece che da SSO

Il PEP inietta token SSO a vita breve

Il PEP inietta token SSO di connessione e non di sessione

Permettere al PEP di modificare, annullare, revocare i token SSO e chiedere autorizzazioni aggiuntive se nel tempo il client perde punteggi di fiducia

Autenticazione locale

Sistema nuovo che si sta diffondendo e potrebbe standardizzarsi

- Gli utenti si autenticano su un dispositivo attendibile – il PC dell'organizzazione
- Il dispositivo è in grado di attestare la tua identità verso un server remoto

Per esempio lo standard UAF (Universal Autentication Framework) dell'Alleanza FIDO (Fast ID Online)

Utilizzano più fattori di autenticazione (password + biometrico) per spostare la fiducia dai servizi remoti gli 1-2 device dell'utente

E' come un gestore delle password ma gestisce le chiavi private – se lui presenta la chiave pubblica il server sa che il dispositivo è sicuramente in possesso anche della chiave privata, e apre

- Gli attacchi reply sono mitigati con generazione di token casuali
- I men-in-the-middle si evitano semplicemente imponendo in locale che la richiesta di canale criptato venga dallo stesso dominio che si sta visitando
- Il riutilizzo delle chiavi non esiste: vengono generate chiavi anche casuali, in modo banale

Autenticazione di gruppo

Regola delle tre firme: per entrare su un servizio bisogna essere almeno in tre (n)

Shamir's Secret Sharing

Viene diviso un secret in n parti e distribuito, in modo che per autenticare servono almeno k password

Su Linux/Unix è implementato tramite `ssss`

Red October di Cloudfare

Il server contiene le chiavi pubbliche/private degli n - ovviamente criptate con password di n(i)

Gli viene dato un algoritmo: Per decrittare serve Mario, poi Bruno, poi Elisa

Lui crea una chiave criptata con questo algoritmo in modo che vengano decrittate da Mario, Bruno, Elisa, in questo stesso ordine

DNS Root Zone Signing - La cerimonia di firma delle root zone in DNSSEC – BELLISSIMA!

Vengono generate le chiavi di firma delle root zone

Vengono scelti 7 attori con 7 ruoli diversi, HSM, dati biometrici, reti air gap

Alla fine viene prodotta la chiave pubblica/privata che vale tre mesi, fino alla prossima cerimonia!

Trust score per utenti

L'attività quotidiana degli utenti è un ottimo modo per scoprire problemi e screditarli

Qualora succeda una anomalia valutare se sia meno rischioso bloccare o richiedere una autenticazione aggiuntiva

Gli utenti tendono ad avere **modelli di accesso** prevedibili – è utile assegnare un punteggio

- Non tendono a cercare di autenticarsi più volte al secondo
- Non tendono a cercare di collegarsi 100 volte in un giorno

Modelli di utilizzo delle applicazioni

- Solitamente utilizzano un sottoinsieme ristretto di dati e applicazioni della struttura
- Si può procedere a screditare gli utenti che non hanno mai usato una applicazione
- Integrare l'attività utente con indirizzi presenti il blocklist permette di bloccare immediatamente le richieste che vengono da IP considerati malevoli

Trust score per utenti

Dispositivi utente

- Gli utenti potranno voler accedere in qualsiasi punto del mondo, ma avranno pochi dispositivi
- I dispositivi non usati da tanto tempo possono essere screditati (PC cambiato o perso)

Geolocalizzazione IP

- Controllare la posizione dell'utente con quelle passate o abituali per trovare anomalie
- Dispositivo utente passato improvvisamente dall'altra parte del mondo!
- Dispositivo dell'utente connesso da Italia e un altro dispositivo utente in USA
- *NB: la geolocalizzazione a volte sbaglia, l'utente dimentica i dispositivi a casa*
- *L'utente potrebbe aver usato una VPN americana sul dispositivo B - non va bene lo stesso, specialmente se B è un dispositivo dell'Organizzazione!*

Autenticazione del traffico – principi base di Zero Trust

- Zero Trust richiede che tutti i flussi siano autenticati e autorizzati
- Avvio del flusso di fiducia – tunnel end-to-end fra utente/dispositivo e risorsa
- Chi autorizza la richiesta di autorizzazione? (Problema del «primo pacchetto» o immissione sicura)
- Cosa ce ne facciamo del vecchio sistema di filtering basato su IP e reti?

Generazione di flussi autenticati

Come abbiamo visto una volta che l'utente si è autenticato

- PA fornisce al PEP gli strumenti necessari per stabilire una connessione end-to-end criptata fra soggetto e risorsa
- Il PEP apre i firewall necessari e comunica i certificati di connessione a soggetto e risorsa

In quale punto dello stack si inserisce Zero Trust?

A quale livello della pila ISO/OSI vengono stabilite queste connessioni criptate end-to-end?

Dobbiamo metterci nel caso più generale possibile e sviluppare un sistema che sia compatibile con tutti i dispositivi client, i dispositivi server, l'infrastruttura di rete e le politiche di firewalling di tutte le reti che vengono attraversate da soggetto a risorsa

TLS vs IPsec

I casi possibili sono due, ognuno con i suoi PRO e i suoi CONTRO

TLS e IPsec

- **TLS** vive nel layer di applicazione (livello 5 o 6 della pila ISO/OSI) – è gestito dall'applicazione

- **IPsec** vive in layer sottostanti (3 o 4 della pila ISO/OSI) – è implementato nel kernel del sistema operativo. IPsec è stato implementato come specifica IPv6

IPsec - PRO

Sicuramente per il fatto di essere indipendente dall'applicazione che poi dovrà stabilire traffico criptato, IPsec è preferibile per la sua posizione privilegiata

Essendo integrato in profondità nello stack di rete può essere configurato per consentire la trasmissione dei pacchetti solo dopo che si è stabilito il canale sicuro (crea un tunnel «vero»)

Il ricevente può essere configurato per elaborare soltanto pacchetti IPsec e drop gli altri

IPsec può facilmente prendersi la responsabilità del traffico dopo esserci autenticati/autorizzati in maniera canonica (non è necessario che ci si autentichi di nuovo)

Può essere utilizzato anche insieme a TLS per costruire tunnel più inespugnabili, criptati sia con IPsec che con TLS

IPsec - CONTRO

Problemi di supporto di rete

- IPsec introduce in rete due protocolli (di livello IP): ESP e AH che sono ben supportati
- Firewall configurati in modo errato o attraversamenti di NAT bloccano spesso i pacchetti
- Amazon AWS (e altri provider cloud) non consentono la trasmissione ESP e AH nelle loro reti
- Hotspot wi-fi pubblici (forse anche quelli nell'organizzazione) non sempre supportano questi pacchetti

In queste reti ostili a IPsec si può mitigare incapsulando i pacchetti IPsec in UDP

IPsec - CONTRO

Problemi di supporto del dispositivo

- IPsec è complesso e sia sul client che sul server deve essere configurato: ci sono infiniti protocolli e cipher su cui si devono mettere d'accordo
- La suite di cifratura cambia abbastanza spesso per compromissioni le patch per IPsec per supportare le nuove suite escono con lentezza perché è un modulo del kernel
- I dispositivi mobili supportano IPsec ma non sono implementati e non hanno moduli kernel come richiede Zero Trust (ESP, AH)

IPsec - CONTRO

Difficoltà di configurazione applicazioni (lato server)

- IPsec deve essere configurato attentamente sui server
- Devono essere abilitate nel kernel le suite desiderate
- Eseguire in user-space il demone IKE

TLS PRO

- E’ semplice da implementare
- E’ supportato largamente dai browser
- Esistono varianti UDP!!! (DTLS)

TLS CONTRO

- La configurazione di default non comprende il tunnel con certificati
- Funziona prevalentemente con applicazioni web

Per mitigare

- Si predispone un portale WEB perfettamente compatibile TLS da cui si accede ai servizi sotto (purché i browser rimangano aggiornati) – come fa Google

- Lato server si può implementare un demone TLS già configurato che sta sempre in ascolto e crea i tunnel su richiesta piuttosto che dovergli passare la configurazione di volta in volta

Alla ricerca di un compromesso – Utilizzare entrambi!

Per connessioni client/server (anche tramite un proxy/portale) via web	TLS
Per connessioni client/server non web	IPsec
Per connessioni server/server	IPsec
Per reti che non supportano IPsec	IPsec incapsulato in UDP

Considerazioni e Sviluppi futuri

- Microsoft Server Isolation e IPsec

Funziona solo per reti client/server Microsoft con gestione dell'autenticazione/autorizzazione tramite Active Directory

Attraverso Windows Firewall, Criteri di Rete e Criteri di Gruppo riescono ad automatizzare la configurazione di IPsec. Active Directory fornisce supporto per autenticazione granulare

- Per essere compatibile con Zero Trust IPsec va utilizzato nella modalità trasporto: incapsulamento solo del payload
  - Gli header IP rimangono invariati e questo permette al pacchetto di andare soltanto dal destinatario, ed essere spacchettato lì, end-to-end, senza essere «spacchettato» in un intermediario. In situazioni «migratorie» è accettabile la modalità tunnel

- IPsec è completamente integrato e compatibile con IPv6, potrebbe essere il vincitore sul lungo periodo

(e la vulnerabilità quantistica?)

First Packet Problem – Chi autorizza l'autorizzazione?

Supponiamo di avere una rete completamente Zero Trust

Quindi ogni flusso è autorizzato, autenticato, criptato ed end-to-end per definizione

Per alleggerire il traffico sulla rete posso droppare simpaticamente tutto il resto del traffico

Quel traffico NON SERVE A NIENTE!

E' traffico che non deve raggiungere nessun servizio in quanto non autenticato/autorizzato

La mattina dopo un collega arriva in ufficio o a casa e accende il PC, con l'idea di autenticarsi e ottenere l'autorizzazione per la risorsa su cui deve lavorare

Come fa a raggiungere il servizio di autorizzazione (PEP) se il traffico non autorizzato è bloccato?

Chi autorizza (e cripta) il traffico di richiesta autorizzazione?

«Immissione sicura» del primo pacchetto

Si usa un meccanismo di pre-autenticazione

SPA (Single Packet Authorization)

- La pre-autenticazione è l'autorizzazione di una richiesta di autorizzazione che imposta le aspettative della connessione che si andrà a creare
- Si implementa in UDP - non è necessaria la risposta, questo evita che il servizio SPA sia esposto alla rete: risponde solo se gli arriva il pacchetto giusto
- Il pacchetto UDP è crittografato con una chiave pre-impostata e condivisa col server
- A quel punto il server sa che di lì a poco si deve aspettare una richiesta di autorizzazione e apre il firewall solo a quella, per farla passare – limitata nel tempo

Implementazione e deployment di SPA

Si implementa attraverso una evoluzione del vecchio **port knocking!**

fwknop – FireWall KNock Operator - Open source (va messo sul PEP)

Quando il PEP riceve un pacchetto UDP criptato e valido lo decripta

Il payload contiene diversi campi fra cui la richiesta di apertura

La richiesta di apertura contiene l'IP del mittente, il protocollo, la porta di destinazione e (opzionale) la porta sorgente che si devono aprire per far partire l'autorizzazione

fwknop apre il firewall da quell'IP su quelle porte per 30 secondi (default: si può stringere)

La crittografia del pacchetto UDP avviene con chiavi GnuPG

Et voilà! Il collega si può autenticare!

NB: solitamente non si può installare su BYOD, ma solo su dispositivi di proprietà :(

https://www.cipherdyne.org/fwknop/index.html

Filtering dei pacchetti tradizionale

Zero Trust a livello puro consente di negare tutto il traffico che non sia autenticato e autorizzato

Ovviamente non sempre è facilmente raggiungibile, magari solo in settori di rete

Altrove ci può essere un avvicinamento sostanzioso a Zero Trust, ma non del tutto

Inoltre Zero Trust non è concepito per difendersi dai DDoS -> servono altri meccanismi

Il filtraggio e le regole di filtraggio «tradizionale» hanno ragione di rimanere in essere

Sicuramente restano NECESSARIE sul bordo:

- Antispoofing
- Filtraggio del grosso del traffico che altrimenti ricadrebbe nei punti interni della rete (meglio drop subito che sull'ultima foglia per alleggerire il traffico)
- Politiche generali di ACL basate su blocklist e servizi (non è necessario, ma tutto fa!)

Altri filtri necessari – filtro host

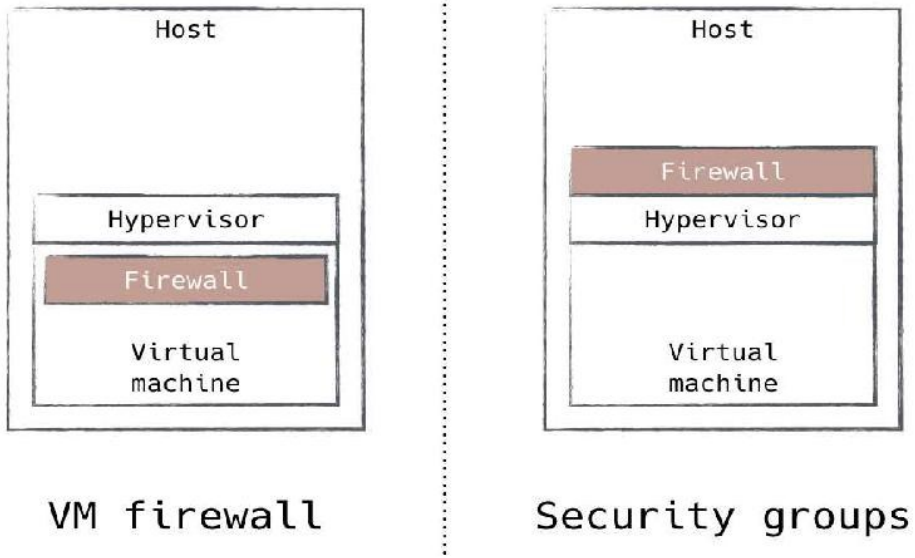
Filtro host - Regola i flussi direttamente sulla macchina

- Molto granulare – permette di aprire solo i servizi realmente esistenti sulla macchina
- Supportato nativamente da tutti i sistemi operativi, sia client che server (esclusi iOS e Android)
  - iptables su Linux
  - BPF (Berkley Packet Filtering) su *BSD
  - Mac OS da riga di comando o sull'applicazione
  - Windows Firewall
- In alcuni casi hanno già implementato SPA sia come server che client
  - Permette aperture/chiusure programmate tramite giorni/orari ma anche eventi
  - Permette di rimanere «nascosti» alla rete e presentarsi solo con pacchetti SPA speciali

Altri filtri necessari – filtro host

Svantaggi: Se l'host è compromesso possono essere disabilitati

- Richiedere il 'root' per modificare le impostazioni
- In ambienti virtualmente isolati si può delegare all' 'hypervisor' la gestione dei filtri sulla macchina (amazon lo fa con i gruppi di sicurezza EC2)
- Si può fare in ambienti SDN delegando al manager di creare la rete apposita

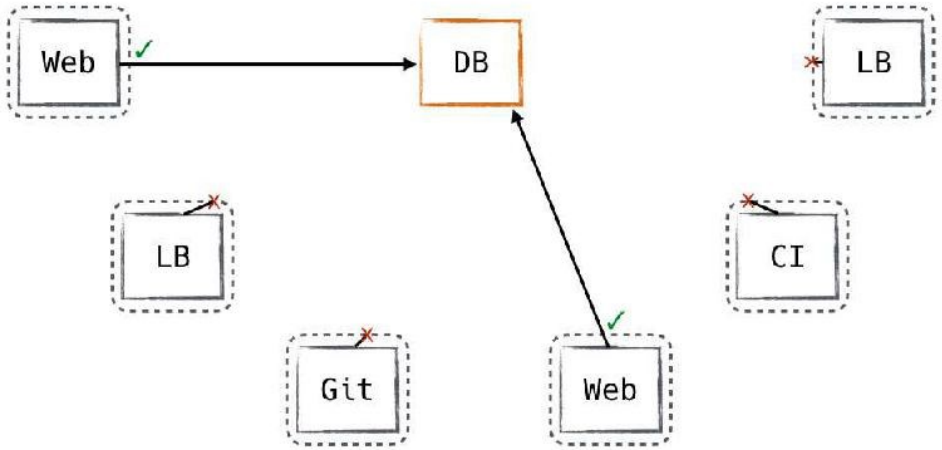


Altri filtri utili

Filtro bookended (booked-on-end)

E' un filtro host ma «in uscita» (Egress)

- Permette di stabilire granularmente il traffico in uscita (non tanto sui client ma SUI SERVER!)
- Aiuta a raggiungere **«l'immunità di gregge»** di un segmento di rete



Filtro intermedio (il meno utile di tutti)

In teoria in Zero Trust non dovrebbero servire

Ma se ci sono NAT già devono fare NAT, possono anche filtrare

Diciamo che se ci sono e non possono essere riconvertiti ad altro «male non fanno»

Ultime considerazioni

Zero Trust è basato fortemente sull'automazione…. Ma….

UpNP – NO!!!!! Non è autenticato e autorizzato! Fortemente sconsigliato per debolezze!

E invece….

SDN SI'! Anche per il routing!

I HAVE A DREAM:

SDN che, a seguito di una autenticazione/autorizzazione, installa SOLTANTO quel flusso di routing. Il client autorizzato viene configurato in una rete TUTTA sua! Col perimetro che tende a ZERO, come se fosse un cavo diretto o una /31 fra soggetto e risorsa!!!

Case Study – Google BeyondCorp

E' uno dei pochi completamente documentati – si possono prendere spunti (adesso la vendono)

https://cloud.google.com/beyondcorp

Nasce a fine 2010 e dura fino al 2014

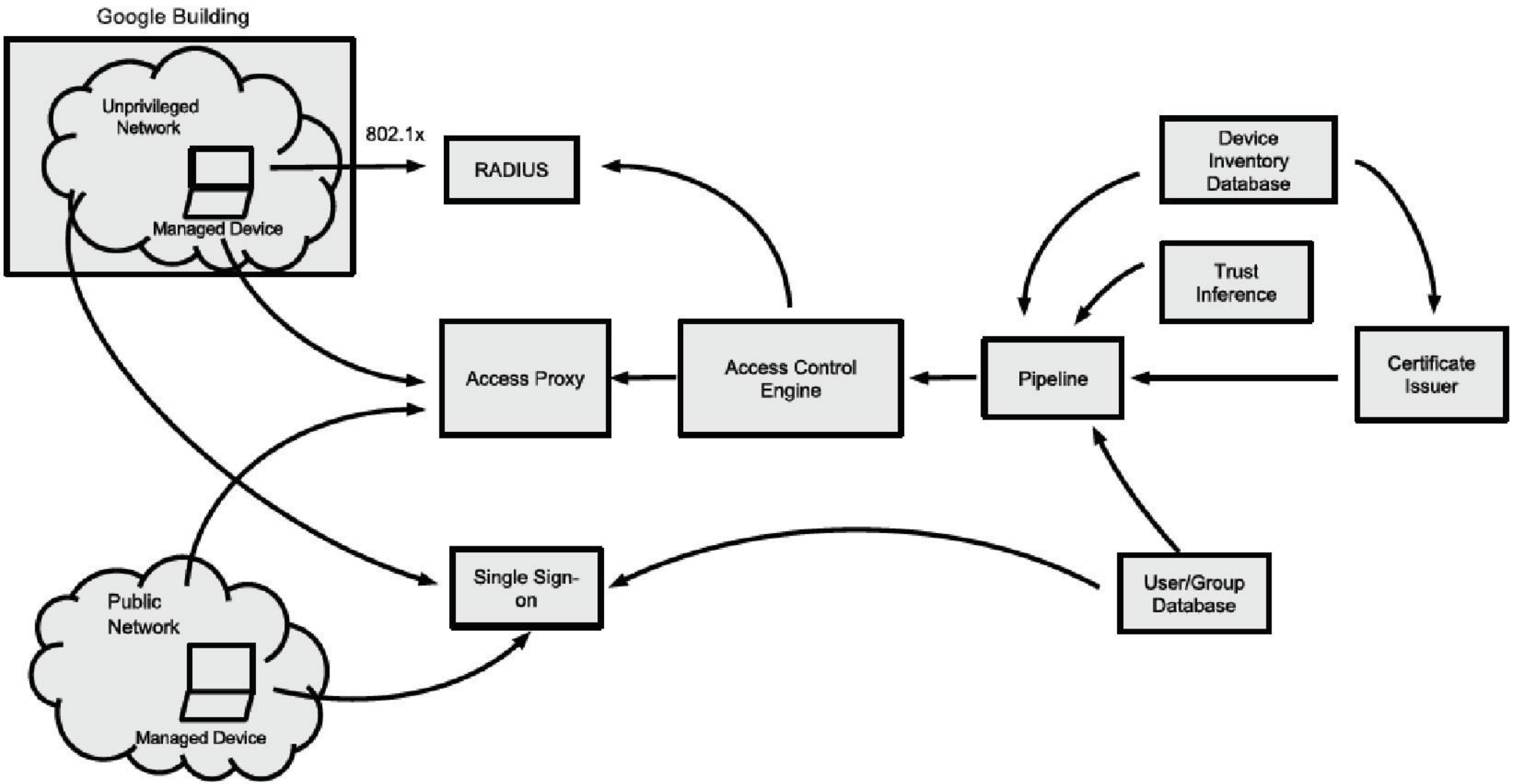
Lo scopo era **BUTTARE VIA LA VPN** (perché non scala) e far lavorare i dipendenti in maniera sicura in qualsiasi posto del mondo si trovassero

La loro idea era disfarsi del «castello medievale» della rete Google, in cui i dipendenti (perlopiù lavoratori remoti) dovevano fare salti mortali per valicare le mura inespugnabili

- Scarsa scalabilità di risorse e di capacità di gestione in un sistema sempre più complesso
- Dentro il «perimetro» di Google stava più extranet rispetto al personale in ufficio

Dalla città medievale bisognava passare ad una città moderna, mediando l'accesso al sistema in base a chi sei e non a quale rete utilizzi

Architettura Zero Trust di Google



1- Controllo totale di ogni dispositivo

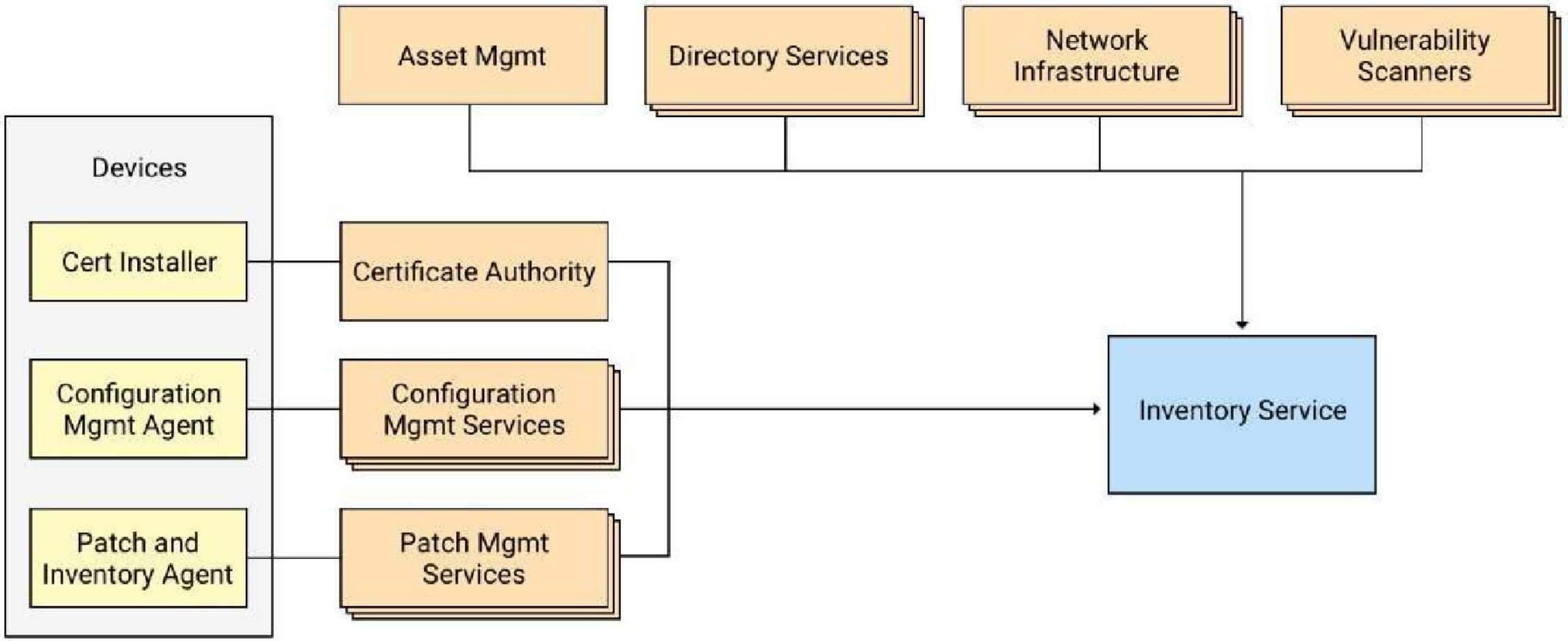
Controllo completo di tutti i dispositivi di proprietà – «dispositivo gestito»

- Acquisito dall'azienda e gestito attivamente attraverso l'inventario che è autorevole
- I dispositivi vengono costantemente monitorati
- I dati relativi ai dispositivi stanno in più database – creazione di un meta-inventario per uniformare

Controllo identità univoca del dispositivo

- Certificato per ogni dispositivo legato al suo record in inventario
- A seconda del device la chiave privata sta su TPM hardware o software
  - C'è un processo per stabilire se la chiave è archiviata in maniera sicura (cambia il punteggio)
  - Il processo si svolge consegnando un certificato intermedio a scadenza breve per ogni test superato
  - La scadenza di questi certificati «di qualità» indica che si deve ripetere il test di qualità
  - Il certificato è usato non solo per autorizzare ma come chiave per informazioni sul dispositivo

Dispositivi gestiti



2- Controllo degli utenti

Il database utenti è gestito completamente dalle risorse umane – ogni utente e gruppo è censito

- Database di utenti
- Database delle catalogazioni dei lavori (gruppi di lavoro, settori, uffici)
- Database delle appartenenze ai gruppi

Autenticazione tramite portale SSO centralizzato che fornisce un token di autorizzazione verso le applicazioni

3- Architettura Portale/Gateway (Access Proxy)

I soggetti che chiedono una risorsa arrivano sull'Access Proxy

AP è unico ma ha istanze completamente separate per servire ogni singola applicazione

(Retaggio di GFE – Google Front End)

- Fornisce raggiungibilità globale
- Bilanciamento di carico
- Protezione DDoS
- Proxy di autenticazione per tutti i backend
- Controllo accessi «a grana grossa»
- Avvenuta l'autorizzazione con i dati e protocolli di autenticazione chiesti dall'applicazione crea il tunnel criptato fra utente e risorsa che in realtà presenta il proprio PEP per una autenticazione granulare a piacere

4- Autenticazione su AP per tutti

L'AP verifica l'identità degli utenti sull'IdP Google

Verifica anche l'identità dei device sia per batch m2m che per i tunnel sull'inventario

- Desktop e Laptop con certificato inserito in inventario
- Mobile: Identifier Vendor per iOS, ID dispositivo per Android

Proxy di autenticazione per tutti i backend - non potevano cambiare i meccanismi di autenticazione su ogni singolo backend

- Ha dentro di se' tutti i driver di autenticazione dei backend sottostanti: OpenID Connect, Oauth, altri proprietari e altri fatti in casa, RADIUS, SSH Proxy

In questo modo, quando l'utente è autenticato e autorizzato «grosso» stabilisce il suo tunnel con il backend dove c'è un PEP (un reverse proxy) per autenticare in maniera granulare

Le comunicazioni fra AP e backend sono criptate e reciprocamente autenticate con i certificati

5- Autorizzazione a «maglie larghe» su AP

Controllo accessi a «grana grossa» o maglie larghe

- Database utenti e dispositivi con trust deciso dinamicamente sulla base di osservabili (monitor)
  - Screditare dispositivi che non hanno le ultime patch
  - Dare punteggi «fissi» a un particolare modello di telefono o tablet
  - Screditare utenti da posizioni (GeoIP) mai visti prima
  - Regole STATICHE ed EURISTICHE

L'autorizzazione tiene conto di

- Informazione su utenti e gruppi
- Certificato del dispositivo e artefatti del device (dall'inventario)
- Lo score
- Es. I bug del codice possono essere visti solo dai tecnici con dispositivo «ingegnere»
- Le applicazioni finanziarie: solo dal gruppo «finanziario» con dispositivo non tecnico
- Possono essere date anche limitazioni parziali ad una applicazione (sola lettura)

6- Policy per l'autorizzazione

Le policy per l'autorizzazione stanno nel Policy Administrator – in Google si chiama «ACL Engine»

L'ACL Engine viene interrogato tramite RPC

- Dal Proxy AP per le regole di autenticazione «grossa»
- Dai PEP dei singoli backend per l'autorizzazione granulare
- Altri servizi RADIUS, SSH Proxy

Le ACL sono scritte in un linguaggio specifico (non esistono standard)

7- Eliminazione di VPN e migrazione per singolo servizio

La migrazione per eliminare la VPN non è stata immediata, ne' istantanea, ne' per tutti lo stesso giorno

Sono migrate una o più servizi per volta, entrando piano piano ad essere integrati in AP

Nella situazione di transizione Google aveva creato una rete non privilegiata in sede da cui tutti gli utenti si connettevano come fossero su internet (per fare i test)

Per ogni applicazione le fasi del test sono state:

- Accesso da rete privilegiata e in VPN da non privilegiata
- Accesso da rete privilegiata e tramite AP da rete non privilegiata (split DNS)
- Accesso da AP sia per rete privilegiata che non privilegiata
- Eliminazione della VPN per quella applicazione

8- Estensione a tutti i servizi

- Limitazione degli utenti all'utilizzo della VPN solo per comprovate esigenze
- Monitoraggio del traffico VPN e disabilitazione utenti che non ne facevano uso da un periodo
- Per gli utenti attivi in VPN
  - Monitoraggio del traffico del singolo utente (incasinato, più servizi, AP, VPN, aiuto!)
  - Spostamento del traffico su AP solo se mano a mano tutti i loro singoli flussi di lavoro venivano spostati su AP

Questa procedura è stata automatizzata in modo che l'utente attivo sulla VPN dichiarasse da solo via via i flussi che erano passati su AP, e desse feedback

L'utente aveva a disposizione una simulazione perfettamente funzionante di un AP per provare

Alla fine, quando si è sentito pronto, l'utente da solo si è messo il flag per passare tutto in AP ed essere eliminato dalla VPN

9- Comunicazione verso gli utenti

Non troppo scarsa

- Utenti ignari, sorpresi, confusi
- Difficile intervenire e sistemare
- Carico eccessivo sullo staff tecnico

Non troppa

- Utenti resistenti al cambiamento tendono a sovrastimare l'impatto e cercare esenzioni o eccezioni
- Utenti che sottovalutano l'impatto o non prestano abbastanza attenzione e validano processi che in realtà non passerebbero i test
- Utenti che attribuiscono i loro problemi sempre «alla migrazione» invece che a cambiamenti strutturali in altri ambiti non correlati a BeyondCorp

Definire lo scopo per cui si fa:

- abolire il perimetro
- abolire la VPN
- tunnel verso le cloud esterne
- migliorare la sicurezza
- aumentare i controlli
- Tutto quanto sopra, etc

Pianificare tutto

Stabilire un piano per la raccolta dei dati

Fare diagrammi basati su ogni singolo servizio e sugli utenti che utilizzano quel servizio
Se c'è già il documento di analisi del rischio queste cose «si sanno già»

Fare un elenco dettagliato (per singola applicazione) dei flussi leciti che vanno e vengono da lì (utenze, batch file, backup remoti scansioni remote, configuration manager etc)

Fare un elenco dettagliato degli utenti (IdP, ID Manager, altri DB)

Fare un elenco dettagliato dei device (configuration and patch management, inventario e altro)

Pianificare una CA PKI automatizzata con API (fortemente raccomandata interna)

Di quel che c'è non si butta nulla!

Guardiamo quello che abbiamo con un occhio ai requirement e alle basi di Zero Trust

- Autenticazione/Autorizzazione IDM, IdP, SSO
- Inventario device/utenti/inventario servizi e applicazioni
- «inventario» di tutti i flussi leciti nell'organizzazione
- Segmentazione della rete ad oltranza e micro-segmentazione
- SDN, SDP, intent-based network
- Proxy e reverse proxy intelligenti, NSFW
- Monitoraggio

Varie possibilità per iniziare

- Google ha iniziato dai device
- Il NIST consiglia di iniziare o da un ID Managment oppure dalla parte di rete micro-segmentata
- American Council for Technology-Industry Advisory Council (ACT-IAC) consiglia di iniziare dagli utenti (e come secondo step l'ID Management)

  https://www.actiac.org/system/files/ACT-IAC%20Zero%20Trust%20Project%20Report%2004182019.pdf
- Altri consigliano di iniziare dal portare in Zero Trust i servizi che trattano dati più sensibili

- Stabilire priorità
- Riutilizzo di quel che già c'è

Consigli 1 – Identificare i dati

Quali dati sono sensibili per la mia organizzazione

Dove risiedono questi dati

Quali flussi procedurali sono associati a questi dati

- Come si accede al dato (HTTPS, applicazione specifica etc)
- Da quali device
- Da quali utenti e/o classi di utenti
- Quali sono i tipici casi d'uso

OUTPUT: Un «inventario» dal generico allo specifico

- Categorie generali dei dati
- Servizi/server – anche cloud
- Applicazioni
- Documenti specifici
- Processi e procedure per l'utilizzo dei dati

Consigli 2 – Capire i propri flussi – la rete – i device – le applicazioni

Flussi

- Come fluiscono i flussi all'interno dell'organizzazione
- Come fluiscono da/verso i servizi esterni
- Come è gestito il failover/ridondanza nei vari casi

Rete

Cosa considero parte della mia rete

- Cloud
- Sistemi ibridi
- Sistemi fisici

Device: Mobile, BYOD, workstation, dispositivi remoti

Applicazioni: quali applicazioni della mia rete sono visibili, sono servizi agli utenti

Consigli 3 – Implementare Monitoraggio e discovering

Sul monitoring plane separato dal control plane e dal data plane
- Utilizzare network detection tool
- Implementare politiche automatiche di response
- Ispezionare il traffico NSEW
- Per il traffico criptato ispezionare i metadati di traffico

Sul dataplane
- Inserire strumenti di discovering dei device
- EDR – Endpoint Detection and Response

Strumenti di prevenzione (generano molti bei log)
- NGFW, WAF, Proxy, SIEM

Enrichment dei dati
- Con strumenti di threat detection, blocklist, incident alert etc

Consigli 4 – Riepilogo

- Pianificare per scritto
- Considerare prioritario il discovering (device e applicazioni)
- Un buon sistema di monitoraggio (NDR, SOAR)
- ID Management, IdP, gestione delle identità
- Micro-segmentation

OpenSource?

Prima di tutto Zero Trust NON E' UN PRODOTTO!

Va fatto tutto quanto detto prima (monitoraggio, CA, IDM, inventario dei flussi, device, utenti)

Ci sono soluzioni opensource per i proxy di autenticazione/autorizzazione

Sono tutti basati sull'AP IAM di Google BeyondCorp

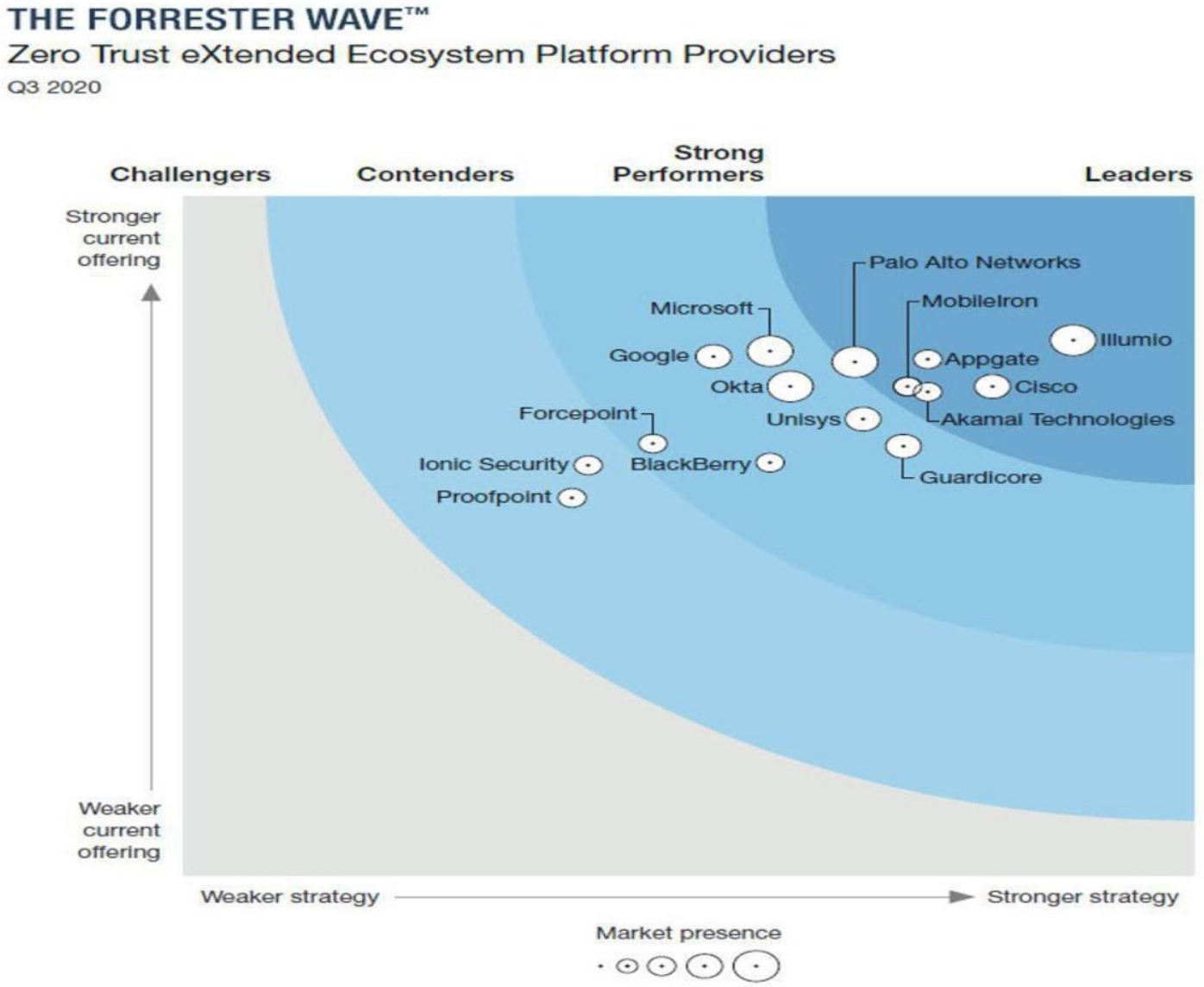
- Opensource per operazioni basiche
- A pagamento per integrare la gestione di alcune policy (Geolocation)
- In quasi tutti i casi forniscono solo la parte proxy, la parte policy va fatta a mano in casa
- In altri casi sono framework TLS

Per valutare bisogna aver chiaro lo scopo – alcuni sono fatti solo per cloud, o solo per HTTPS

Elenco di possibili (?) candidati OpenSource

Pritunl Zero	https://zero.pritunl.com/
Pomerium	https://www.pomerium.com/
Transcend	https://github.com/cogolabs/beyond
Trasa	https://github.com/seknox/trasa
Helios (early stage)	https://github.com/cyakimov/helios
Ory (solo HTTP ma ha un bel motore delle policy)	https://www.ory.sh/oathkeeper/docs/
Ziti	https://openziti.github.io/ziti/overview.html
SPIFFE/SPIRE (framework TLS)	https://spiffe.io/
step (API cert, Token, tunnel criptati)	https://smallstep.com/blog/zero-trust-swiss-army-knife/

Soluzioni «già confezionate»



I prossimi:

CrowdStrike
Fortinet
Zscaler
Symantec (SASE - Secure Access Service Edge - CloudSOC)

La parola agli utenti Zero Trust

Gli utenti sono più felici in un ambiente Zero Trust!

- Non sono bloccati
- «Scavalcano» muri di sicurezza e trust con una semplice forma di autenticazione in più
- Si sentono parte del ciclo della sicurezza e non la vedono come ostacolo alla produttività
- Lo staff IT/security dopo il lavoro iniziale ha molte meno richieste e molto più controllo
- Il trust device+utente rende gran parte dell'autenticazione trasparente all'utente

Zero Trust può essere considerato l'iperconvergenza della security

Conclusioni

Data la complessità delle nostre strutture cercare di fare sicurezza alla vecchia maniera non basta

Dobbiamo cambiare paradigma

- Coinvolgimento della Leadership e di tutta l'organizzazione
- Non più sicurezza puntuale ma basata **sull'analisi dei rischi**
- Non più sicurezza perimetrale ma **Zero Trust** (end-to-end)

Spero di aver lasciato qualcosa sulle **basi dello standard** che definisce questo processo

Spero di aver lasciato qualcosa sui **dettagli e implementazioni** di questo processo

- Le strutture logiche da cui è composto
- Il ciclo della fiducia
- La gestione delle policy dinamiche

Spero di aver lasciato qualcosa su **quello che comporta** a livello device, utente, flussi di traffico

Spero che l'esempio proposto sia valso a **prendere spunti** da coloro che sono più grandi di noi

Spero che i consigli siano utili **per iniziare a ragionare** (e forse a partire) in questi termini

- **Le soluzioni di sicurezza proposte «non Zero Trust» CONVERGONO comunque a questi principi**

Spero di aver lasciato l'idea che l'utente si sente più a suo agio e meno ostacolato in questa architettura

Riferimenti

NIST SP 800-207 "Zero Trust Architecture"
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf

fwknop
https://www.cipherdyne.org/fwknop/index.html

Implementazione Zero Trust in Google
https://cloud.google.com/beyondcorp

Implementazione Zero Trust (ancora in corso) in Microsoft
https://www.microsoft.com/en-us/itshowcase/implementing-a-zero-trust-security-model-at-microsoft

Microsoft Free assessment tool
https://info.microsoft.com/ww-landing-Zero-Trust-Assessment.html

Zero Trust from Zero to Zero Trust
https://gravitational.com/blog/zero-to-zero-trust/

Domande?

Vincenzo Calabrò