Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Definizione di un framework per la gestione della sicurezza dei dati nella comunicazione tra sistemi eterogenei in campo marittimo

Aniello Minutolo, Mario Sicuranza, Mario Ciampi

RT-ICAR-NA-2016-03 **Agosto 2016**

Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR) – Sede di Napoli, Via P. Castellino 111, I-80131 Napoli, Tel: +39-0816139508, Fax: +39-0816139531, e-mail: napoli@icar.cnr.it, URL: *www.na.icar.cnr.it*

Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Definizione di un framework per la gestione della sicurezza dei dati nella comunicazione tra sistemi eterogenei in campo marittimo

Aniello Minutolo, Mario Sicuranza, Mario Ciampi

Rapporto Tecnico N: **RT-ICAR-NA-2016-03** ***Data:*** **Agosto 2016**

Definizione di un framework per la gestione della sicurezza dei dati nella comunicazione tra sistemi eterogenei in campo marittimo

Aniello Minutolo, Mario Sicuranza, Mario Ciampi

Istituto di Calcolo e Reti ad Alte Prestazioni, Consiglio Nazionale delle Ricerche
Via Pietro Castellino, 111 – 80131 Napoli, Italia
E-mail: {aniello.minutolo, mario.sicuranza, mario.ciampi}@na.icar.cnr.it

Abstract

Questo lavoro descrive le problematiche fondamentali di sicurezza che si possono riscontrare in infrastrutture di comunicazione tra sistemi complessi ed eterogenei come quelli che si riscontrano in ambito marittimo in cui esistono diversi sistemi eterogenei che necessitano di comunicare tra loro in modo sicuro. In dettaglio, vista la crescente informatizzazione delle navi derivante dai complessi problemi decisionali continuamente coinvolti nelle attività navali, il presente rapporto tecnico descrive le principali problematiche di sicurezza che devono essere affrontate al fine di garantire un'opportuna gestione della sicurezza dei dati e delle informazioni, sia in termini di accesso e protezione dei dati, che in termini di integrità e disponibilità degli stessi. A tale scopo, a valle di uno studio preliminare delle problematiche fondamentali da affrontare, viene proposto un framework per la gestione sicura dei dati, che possa efficacemente garantire diversi requisiti di sicurezza per le informazioni di interesse per una piattaforma integrata di comunicazione tra diversi sistemi navali e portuali e dei vari sotto-sistemi che compongono l'ambiente marittimo.

Keywords: Sicurezza, Comunicazione Sicura, Intrusion Detection System, Access Control.

1 Introduzione

Il presente rapporto tecnico ha lo scopo di analizzare e proporre tecniche e soluzioni opportune per la messa in sicurezza della comunicazione tra sistemi complessi ed eterogenei, allo scopo di definire un framework che consenta la comunicazione sicura tra sistemi eterogenei in campo marittimo. In particolare, il rapporto tecnico fornisce innanzitutto una analisi delle problematiche di sicurezza legate a prodotti e infrastrutture software in ambito marittimo nel quale diversi sistemi eterogenei necessitano di comunicare tra loro in modo sicuro.

Per raggiungere tale scopo, il rapporto tecnico descrive, in prima istanza, uno studio delle principali problematiche relative alla messa in sicurezza dei dati e delle informazioni sia in termini di accesso e protezione dei dati, sia in termini di integrità e disponibilità degli stessi.

In secondo luogo, è presentato un framework per la gestione della sicurezza dei dati informatici che individua e soddisfa gli obiettivi di sicurezza fondamentali individuati, che una piattaforma sicura di scambio informazioni nel settore marittimo deve perseguire. Il framework proposto specifica, per ogni obiettivo di sicurezza identificato, le tecniche e le tecnologie più adatte alla realizzazione dello stesso. In particolare, al fine di gestire opportunamente la sicurezza dei dati, sono stati identificati i seguenti obiettivi fondamentali:

- **Identificazione di attori e/o eventi** che agiscono o coinvolgono i dati;
- **Protezione dei dati**;
- **Rilevamento delle intrusioni**.

Il rapporto tecnico è organizzato in 4 Paragrafi, i cui ruoli e contenuti sono specificati a seguire:

- Paragrafo 2: Contesto di riferimento
- Paragrafo 3: Problematiche di sicurezza
- Paragrafo 4: Framework per la sicurezza dei dati
- Paragrafo 5: Conclusioni

2 Contesto di riferimento

2.1 Architettura di base

L'ambiente nel quale la proposta si pone è relativo all'ambiente marittimo, nel quale sistemi installati a bordo di navi devono poter comunicare tra loro e con sistemi dislocati a terra. Si vuole proporre una piattaforma di comunicazione sicura per la comunicazione tra sistemi eterogenei, tra cui meccanismi intelligenti finalizzati al monitoraggio dei consumi delle navi. In particolare si prevede la realizzazione di diversi sotto-sistemi in grado di comunicare tra loro e scambiarsi dati e informazioni allo scopo di offrire funzionalità avanzate di supporto decisionale in ambito marittimo.

Figura 1 - Macro-blocchi fondamentali previsti per il supporto decisionale.

In dettaglio, in Figura 1 sono riportati i macro-blocchi fondamentali previsti:

1. **Sistema di Terra** – diretto alla gestione dell'intera flotta e al supporto di decisioni su scelte d'azione riguardanti la flotta e le singole unità che la costituiscono.
2. **Piattaforma di Comunicazione** – è il canale di collegamento efficiente ed economico tra il sistema di Terra e quello di Bordo.
3. **Sistema di bordo** – gestisce la raccolta dei dati operativi della nave e della situazione corrente, supporta la decisione delle scelte tattiche riguardanti il singolo viaggio della singola nave.

In particolare, in questo rapporto tecnico il focus è posto principalmente sulle problematiche da affrontare relative alla gestione della sicurezza dei dati e delle informazioni sia in termini di accesso e protezione dei dati, che in termini di integrità e la disponibilità degli stessi.

A tale scopo, nelle sezioni seguente vedremo come le problematiche di sicurezza siano oggi giorno sempre più sentite in ambito marittimo, vista, soprattutto, la crescente informatizzazione delle navi stesse a causa dei complessi problemi decisionali continuamente coinvolti nelle attività delle navi.

2.2 Sicurezza informatica e trasporto marittimo

Al giorno d'oggi, l'informatizzazione delle navi è sempre maggiore data la sempre crescente mole di informazioni da gestire e la complessità dei processi decisionali coinvolti nelle attività giornaliere svolte a bordo. Il personale della nave, quindi, sente sempre maggiore la necessità di utilizzare tecnologie e sistemi software di supporto alle proprie capacità e competenze. Di conseguenza, il numero delle persone di un equipaggio diminuisce continuamente dato che aumenta sempre più il numero di strumenti informatici di supporto alla navigazione, allo scarico merci, al trasporto e alla localizzazione delle merci.

D'altro canto, l'aumentare dei sistemi informatici a bordo nave, sta aumentando sempre più la vulnerabilità delle stesse alle tipiche problematiche di sicurezza informatica che normalmente affligge tali sistemi.

A tal proposito, bisogna sottolineare che, spesso, le vittime di un attacco informatico o di una generica vulnerabilità della propria sicurezza informatica, tipicamente tendono ad occultare tali eventi al fine di preservare la propria reputazione. Inoltre, spesso, i crimini informatici non sono rilevati affatto, e la stima dei danni subiti a causa delle problematiche di sicurezza si rivela in genere molto inferiore rispetto al fenomeno reale.

Basti considerare che, al giorno d'oggi, il trasporto marittimo costituisce una componente fondamentale dell'economia mondiale, e risulta essere il mezzo di trasporto merci maggiormente utilizzato al mondo, tanto che circa il 90% delle merci scambiate nel mondo passano per il trasporto marittimo. Quindi, risulta quanto mai evidente, ogni crimine informatico ai danni di una nave può generare perdite per diversi milioni di euro e portare alla bancarotta intere società di trasporto marittimo. Inoltre, anche l'economia di una nazione può essere messa a dura prova se il trasporto marittimo di quella nazione viene opportunamente attaccato da criminali informatici. Ad esempio, alterando i valori delle merci immagazzinate nei container, è possibile portare al crollo gli approvvigionamenti nazionali o regionali di un determinato paese. Infatti, ad esempio, il governo britannico ha rivelato che gli attacchi cibernetici sono costati alle industrie del gas e del petrolio inglesi circa 400 milioni di sterline all'anno (circa 563 milioni di euro).

Uno dei problemi tipici di sicurezza che possono essere rilevati a bordo nave riguarda il verificarsi di falle in dispositivi ed apparecchiature: GPS, sistemi di identificazione automatica navali AIS (Automatic Identification System) e ECDIS (Electronic Chart Display and Information System) usati per visualizzare le carte nautiche digitali. Un altro problema riguarda la mancanza di preparazione dell'equipaggio che, che in caso di crimine informatico, spesso non è in grado di gestirlo. Per esempio, una manomissione del GPS può mandare la nave fuori rotta facendo credere al personale di trovarsi in una zona diversa da quella reale, creando così potenziali situazione di pericolo e/o di ritardi nel viaggio in corso. In fatti, ad esempio, nel 2010 una trivellatrice è stata spostata dalla sua ubicazione in Corea del Sud verso il Sud America. I sistemi informatici e di controllo erano contagiati da molti virus e sono serviti 19 giorni per identificare l'hackeraggio e riparare il danno procurato. Ancora, per problemi relativi alla sicurezza informatica, una piattaforma petrolifera è stata chiusa per una settimana a causa della mancanza di personale bordo esperto in materia di sicurezza.

Nell'agosto del 2011, alcuni criminali informatici sono penetrati nei server di una compagnia navale iraniana, alterando le informazioni riguardanti il carico, le date di consegna e localizzazione della merce. Nessuno è riuscito a rintracciare la localizzazione di certi container ed una considerevole parte del carico è stata consegnata nel posto sbagliato e molta è andata persa. Nel 2013 alcuni criminali informatici hanno compromesso il sistema informatico di una nave dell'agenzia australiana responsabile della sicurezza delle frontiere, e sono riusciti a scoprire quali container erano sotto il controllo della polizia o delle autorità doganali. Con queste informazioni, avrebbero poi capito se fosse stato meglio abbandonare alcuni container che contenevano merce di contrabbando. Un altro episodio degno di nota, riguarda il porto di Anversa il cui funzionamento è stato compromesso da criminali informatici dal 2011 al 2013. Infatti, gli hacker hanno alterato il sistema informatico del porto riuscendo a consegnare diversi container di loro interesse a camionisti fidati senza che le autorità portuali se ne accorgessero, e poi hanno rimosso le informazioni sul contrabbando dei container da tutti i database. Quando la polizia ha scoperto il crimine, sono state ritrovate tonnellate di cocaina e armi in alcune valigette, e tutto questo era solo la punta dell'iceberg. Dagli esempi riportati risulta evidente che il trasporto marittimo risulta quanto mai vulnerabile alla sicurezza informatica dei dati e dei sistemi, e quindi risulta fondamentale analizzare le problematiche che possono verificarsi e progettare soluzioni di sicurezza mirate al contrasto e all'attenuazione dei possibili attacchi. Tuttavia, anche se il problema è molto ampio e di difficile soluzione, i sistemi e le infrastrutture proposte dovrebbero in qualche modo affrontare le problematiche di sicurezza informatica in modo da offrire opportune e efficaci. Alla luce di tali considerazioni, le prossime sezioni descrivono i principali aspetti da affrontare per garantire la sicurezza e l'affidabilità dei dati in sistemi e infrastrutture software complesse come quelle sviluppati nell'ambito del progetto.

3 Problematiche di sicurezza

Questa paragrafo descrive le problematiche di sicurezza da affrontare nella realizzazione della piattaforma informatica.

3.1 Analisi dei rischi per la privacy

Il problema della tutela dei dati personali ha molti aspetti in comune con la tematica della sicurezza delle informazioni, sia da un punto di vista tecnologico sia da quello metodologico.

Attualmente tali aspetti sono molto dibattuti e non esiste un'unica visione largamente accettata nella comunità scientifica. I principali rischi per la sicurezza sono i seguenti:

- privilegi d'accesso per gli utenti;
- conformità alle normative;
- ubicazione dei dati;
- separazione dei dati;
- recupero dei dati;

- indagine degli errori;
- sostenibilità a lungo termine.

In generale, in un settore come quello marittimo occorre esaminare non solo gli aspetti tradizionali di sicurezza, quali la gestione degli accessi e la protezione dei dati, ma anche aspetti specifici, come ad esempio la corretta definizione di trattamento dei dati e la separazione tra la titolarità e la responsabilità dei dati, che, se non considerati in maniera appropriata, potrebbero causare rischi per la confidenzialità, l'integrità e la disponibilità delle informazioni.

3.2 Titolarità delle informazioni

Si indica col termine trattamento dei dati qualunque operazione o insieme di operazioni per la raccolta, registrazione, gestione, diffusione e altre azioni sui dati personali.

In particolare, si individuano quattro attori principali:

- l'interessato è la persona, l'ente o l'associazione cui si riferiscono i dati personali;
- il titolare è la persona, la Pubblica Amministrazione o qualsiasi altro ente cui compete la modalità del trattamento dei dati personali;
- il responsabile è la persona, la Pubblica Amministrazione o qualsiasi altro ente responsabile della protezione e della gestione del dato personale;
- l'incaricato è la persona fisica incaricata dal titolare o dal responsabile a compiere azioni di trattamento dei dati.

Stabilire la titolarità dei dati, che compongono ad esempio un documento che viene memorizzato e/o scambiato, è di importanza fondamentale per permettere la loro protezione da possibili manipolazioni non autorizzate.

Con riferimento ai dati che costituiscono un documento, è possibile identificare differenti ruoli per la definizione della titolarità:

- creatore: l'utente o entità responsabile della creazione del documento;
- autore: l'utente o entità responsabile del contenuto del documento;
- responsabile: l'utente o entità responsabile della protezione e della gestione del documento

In generale, il titolare dei dati coincide con la persona che li ha creati oppure con l'ente a cui il creatore afferisce. Eventuali aggiornamenti e/o cancellazioni di informazioni devono poter avvenire solo se l'utente che intende realizzare tali operazioni sia in possesso della titolarità sui dati da alterare.

3.3 Autenticità e autenticazione

Nel settore marittimo, ma anche in altri settori, ciascun utente che effettua operazioni sui dati di un documento deve prendersi la responsabilità della sua azione.

Pertanto, un utente che intende accedere alle funzionalità o ai dati di un'applicazione deve esibire un insieme di affermazioni, dette *security claims*, che consentono di stabilire la veridicità (autenticità) delle sue azioni. Tali security claims devono essere opportunamente certificati, al fine di verificare che l'utente è effettivamente chi dichiara di essere (autenticazione). Un esempio di queste affermazioni possono essere il nome del soggetto, il suo ruolo, l'associazione con altri soggetti, le preferenze e/o le capacità. Esse sono tipicamente impacchettate in un artefatto chiamato security token, il quale può essere realizzato dal soggetto stesso (e certificato, per esempio, attraverso l'uso di una smartcard) oppure emesso da un'altra entità.

All'atto della richiesta dei servizi e dati, bisogna esibire i *security tokens*, i quali devono essere verificati al fine di provarne l'autenticità, l'integrità ed il non ripudio, come descritto di seguito. Lo strumento fondamentale in grado di soddisfare tali requisiti è rappresentato dalla firma digitale. Molto spesso, le applicazioni possono essere distribuite tra vari domini, ognuno caratterizzato da un proprio servizio di autenticazione con un proprio formalismo di definizione dei security tokens e dei relativi claims. È inoltre possibile che ogni dominio disponga di una propria piattaforma oppure che vari applicativi insistono su una stessa piattaforma. L'eterogeneità nel formato e la non centralità dell'autenticazione richiedono di tenere in considerazione due aspetti importanti:

- definire security tokens condivisi tra i domini di autenticazione in cui il sistema risulta partizionato, al fine di promuovere il Single Sign-On;
- nascondere l'eterogeneità nel formato dei tokens e dei claims con appositi servizi di interoperabilità.

3.4 Integrità

I dati che vengono memorizzati nella piattaforma devono essere preservati da modifiche non autorizzate. A tal proposito, la titolarità è uno degli aspetti da gestire per ostacolare tali manipolazioni, anche se, da sola, non è sufficiente. Infatti, è opportuno anche fornire meccanismi per garantire l'integrità di un documento, che permettono di verificare che il suo contenuto sia rimasto inalterato durante la sua trasmissione o memorizzazione.

3.5 Non ripudio

Lo scambio di documenti digitali tra diversi soggetti è una delle operazioni più diffuse, pertanto è cruciale poter fornire la prova incontestabile di un'avvenuta spedizione (non ripudio del mittente) o di un'avvenuta ricezione (non ripudio del destinatario) di un documento, al fine di evitare che un mittente o un destinatario possano disconoscere i dati trasmessi.

3.6 Confidenzialità

L'accesso in lettura ai dati deve avvenire consentendo solo agli utenti autorizzati di consultarli, mentre i dati devono invece rimanere nascosti agli altri soggetti.

Le principali tecniche per garantire la confidenzialità dei dati sono quelle di cifratura, da impiegare sia quando i dati devono essere memorizzati sia quando vengono scambiati all'interno della piattaforma o tra la piattaforma e utenti remoti. Tali tecniche possono essere raggruppate in cifratura a chiave simmetrica e cifratura a chiave asimmetrica. Nel primo caso, esiste un'unica chiave per eseguire operazioni di cifratura e decifratura, mentre nel secondo caso esistono due chiavi distinte. Generalmente gli algoritmi di crittografia simmetrica sono computazionalmente più veloci e semplici di quelli a chiave asimmetrica, anche se hanno lo svantaggio di dover gestire in maniera sicura lo scambio della chiave. Pertanto, di norma, viene preferito l'uso di questi ultimi integrandoli con l'uso degli altri per lo scambio delle chiavi da usare per la cifratura dei dati scambiati.

Nello specifico, la crittografia consiste nel processo di codifica dei dati da scambiare, allo scopo di trasformare tali dati in artefatti illeggibili, la cui comprensione può avvenire soltanto al termine del processo di decodifica, che si occupa di riportare gli artefatti stessi al loro stato originale. Il processo di codifica tipicamente avviene tramite algoritmi dedicati utilizzati tramite specifiche chiavi di codifica come input. Similarmente, la decodifica avviene tramite opportuni algoritmi che ricevono in input le chiavi di decodifica. Quindi, affinché i dati possano essere scambiati tra gli utenti in completa sicurezza, è sufficiente che le chiavi di codifica e decodifica vengano scambiate tra gli utenti tramite canali sicuri. Infine, in sistemi distribuiti, è stata individuata una problematica aggiuntiva che può ledere la confidenzialità dei dati: la cosiddetta *data segregation*. I dati sono spesso, infatti, memorizzati in un ambiente condiviso insieme a dati di altri utenti. Pertanto, una volta che i dati sensibili sono trasmessi, è essenziale che essi siano mantenuti saldamente separati da quelli non sensibili.

3.7 Autorizzazione

Una volta che un soggetto ha fornito il proprio insieme di security tokens, bisogna prendere delle decisioni di autorizzazione per valutare se è possibile abilitare l'accesso alla risorsa richiesta. Questo presuppone la definizione di apposite politiche di accesso per i vari utenti, o classi di utenti, che possono utilizzare le risorse, dati o funzioni, offerte dalle applicazioni.

L'Access Control disciplina quali risorse un dato utente/nodo è autorizzato ad accedere, così da autorizzare o meno l'accesso alle risorse stesse. Durante gli ultimi anni, diversi access control models sono stati proposti, e di seguite ne analizzeremo i metodi maggiormente utilizzati. Il metodo di access control più semplice è l'Access Control List (ACL), noto anche come Identity-Based Access Control (IBAC), che consiste in una lista di permessi associata alla risorsa di sistema da proteggere. Ad esempio, un ACL applicato all'interno di un servizio di publish/subscribe può essere utilizzato per definire i topic come risorse, specificare le operazioni che un client può effettuare sui topic come permessi (creazione e

cancellazione di certi topic, e pubblicazione o consumo delle notifiche relative determinati topic), e creare liste di operazioni ammesse per ogni topic da associare ai vari utenti del sistema. ACL fornisce una semplice soluzione per l'access control, ma presenta diversi svantaggi, specialmente quando un grande numero di utenti e permessi di accesso deve essere gestito. Un metodo di access control più avanzato, progettato proprio per superare le limitazioni di ACL, prevede la gestione dell'accesso alle risorse tramite la gestione dei ruoli che gli individui esercitano all'interno dell'organizzazione di controllo delle risorse. Tale soluzione, nota come Role-Based Access Control (RBAC), sostituire l'approccio resource-focused di ACL con un approccio role-based dove gruppi di individui condividono gli stessi permessi di accesso alle risorse, in quanto facenti parte della stessa categoria di individui che esercita un particolare ruolo di accesso alle risorse considerate. Tale approccio risulta maggiormente scalabile in caso di grandi quantità di individui e risorse, in quanto non bisogna indicare per ogni individuo tutti i suoi permessi di accesso alle varie risorse del sistema, ma soltanto i permessi relativi ai ruoli disponibili e poi assegnare gli utenti a uno o più ruoli di accesso. D'altro canto, anche RBAC presenta diverse limitazioni, tra cui il fatto che il controllo offerto è a grana grossa (coarse-grained access control) e non idoneo alla gestione degli accessi tra domini (cross-domain access).

Per garantire la gestione degli accessi ad una grana più fine rispetto a RBAC, è stato introdotto il metodo Attribute-Based Access Control (ABAC) in cui l'accesso alle risorse è deciso sulla base degli attributi associati ai richiedenti delle risorse, al contesto operativo e/o alle risorse stesse. Il beneficio di utilizzare ABAC è l'assenza del bisogno di conoscere il richiedente in anticipo (come avviene, per esempio, in ACL). Infatti, finché gli attributi del richiedente soddisfano gli attributi di accesso delle risorse, allora l'accesso viene garantito.

ABAC permette di risolvere le difficoltà che possono emergere in scenari e ambienti complessi, dove diversi attributi e meccanismi di accesso esistono.

Infine, per avere una gestione degli accessi maggiormente armonizzata tra organizzazioni distinte, è stato introdotto il metodo Policy-Based Access Control (PBAC), dove l'approccio attribute-based di ABAC viene standardizzato tramite la definizione di regole di accesso in termini di politiche formalizzate in linguaggi formali e condivisi, che permettono di comunicare e negoziare politiche di accesso tra distinte unità di un'organizzazione, o persino diverse organizzazioni federate.

3.8 Disponibilità

Le risorse offerte devono essere sempre disponibili per gli utilizzatori del sistema, e le elaborazioni devono poter restituire sempre il risultato atteso. Possibili dinamiche intenzionali o non intenzionali, come attacchi informatici, fenomeni di saturazione delle capacità della piattaforma o guasti, non devono poter compromettere la disponibilità dei servizi e delle risorse.

Soluzioni a tale aspetto spaziano dall'introduzione di meccanismi di replicazione per tollerare i guasti,

l'allocazione efficiente ed efficace di servizi e risorse nella piattaforma per evitare saturazione e l'utilizzo di Intrusion Detection Systems (IDS) per rilevare possibili intrusioni e applicare opportune contromisure. Il problema della giusta allocazione di servizi e risorse assume una notevole complessità vista la scala e l'eterogeneità dei sistemi sottesi ad una piattaforma. In più, tale problema ha spesso una natura dinamica e deve essere di volta in volta risolto quando una nuova richiesta è sottoposta dagli utenti. Esso, infatti, spesso può essere formalizzato come un problema di ricerca operativa di tipo NP-completo. Pertanto, si rendono spesso necessarie delle opportune euristiche al fine di rendere risolvibile il problema. Attualmente, esistono varie soluzioni IDS, ma di fronte ad alcuni scenari applicativi, gli approcci esistenti presentano diversi problemi. In primis, l'operatore degli IDS dovrebbe essere l'utente, non l'amministratore dell'infrastruttura. In secondo luogo, in molti IDS esistenti devono essere introdotti meccanismi di estensibilità, gestione efficiente e compatibilità con la virtualizzazione basata sul contesto.

3.9 Audit

Nell'ambito della sicurezza informatica, per audit si intende una valutazione tecnica misurabile, manuale o sistematica, di un sistema o componente software. A tal proposito, l'uso delle risorse, delle possibili interazioni con il mondo esterno e delle operazioni effettuate devono essere registrate in appositi sistemi di log, al fine di poter effettuare analisi nel caso si siano verificati malfunzionamenti o violazioni dei requisiti di sicurezza e risalire alle cause.

3.10 Accordi sui livelli di servizio

I servizi della piattaforma devono garantire un livello minimo di servizio misurabile in base a parametri oggettivi, raggiungendo con il cliente un accordo in merito. In questa ottica, è possibile definire un apposito Service Level Agreement (SLA), ovvero un contratto attraverso cui si definiscono le metriche di servizio (ad esempio, qualità del servizio), che devono essere rispettate nei confronti degli sviluppatori/gestori delle applicazioni ospitate. La qualità di servizio offerta dalla piattaforma deve essere continuamente monitorata ed è opportuno che sia verificato se essa rispetta puntualmente gli indicatori del SLA sottoscritto. Questo monitoraggio deve avvenire sia ai fini della monetizzazione del servizio offerto, sia per attestare il servizio di cui le applicazioni beneficiano. Limitatamente alla sicurezza, il SLA può contenere metriche di qualità come la disponibilità delle applicazioni, il livello di privacy e integrità e/o il numero di casi di violazione dei requisiti di sicurezza.

3.11 Intrusion Detection

I sistemi di Intrusion Detection (IDS) sono tipicamente dispositivi software e/o hardware utilizzati per identificare violazioni alle policies di sicurezza di un sistema (ad esempio accessi non autorizzati al sistema).

L'IDS è un costituito da un insieme di dispositivi che dislocati in punti strategici del sistema permettono di individuare e fornire dettagli sugli attacchi. Questi attacchi includono gli attacchi alle reti informatiche,

sfruttando una applicazione locale vulnerabile, attacchi attraverso l'invio di dati malformati, tentativi di accesso agli host tramite innalzamento illecito dei privilegi degli utenti, accessi non autorizzati a computer e file, e i classici programmi malevoli come virus e trojan. In genere un IDS è composto da quattro componenti:

- **uno o più sensori** utilizzati per catturare informazioni del sistema da monitorare;
- **una console** utilizzata per monitorare lo stato del sistema;
- **un motore** che analizza i dati prelevati dai sensori e provvede ad individuare eventuali falle nella sicurezza informatica;
- **un database** cui si appoggia il motore di analisi e dove sono memorizzate una serie di regole utilizzate per identificare violazioni della sicurezza.

Esistono diverse tipologie di IDS che si differenziano a seconda del loro compito specifico e delle metodologie utilizzate per individuare violazioni della sicurezza.

Un IDS consiste quindi in un insieme di tecniche e metodologie realizzate ad-hoc per rilevare le intrusioni.

Prima di proseguire con la classificazione delle varie soluzioni per realizzare sistemi di IDS, introduciamo una breve panoramica relativa alle principali minacce provenienti dalla rete.

3.11.1 Malware e minacce della rete

In questo paragrafo verrà presentata una panoramica sulle principali minacce provenienti dalla rete, con particolare attenzione per il malware.

Il termine malware deriva dalla contrazione di due termini inglesi, rispettivamente

MALicious e softWARE, e viene utilizzato per indicare tutte quelle applicazioni realizzate per danneggiare i computer da qui il nome di software malevolo. È possibile classificare queste minacce come segue:

- **Virus**, di sicuro la tipologia di malware più nota, questo programma una volta in esecuzione è in grado di replicarsi e quindi infettare altri file (che condividono il medesimo file system). Un modo per il quale è possibile mandarlo in esecuzione consiste nel copiare il codice del virus all'interno del file eseguibile di un'altra applicazione. La diffusione del virus avviene prevalentemente per mezzo di supporti fisici (floppy disk prima, chiavette USB oggi), email e programmi di file sharing, ma finché non è caricato in memoria non arreca alcun danno. Il virus per sopravvivere tende a replicarsi in zone nascoste del sistema. Esistono tipologie di virus la cui prolificazione è maggiore grazie alle proprie caratteristiche, i virus polimorfici e metamorfici. Queste nuove generazioni sono in grado di criptare il proprio codice binario all'atto della replicazione in modo da sfuggire ai sistemi antivirus basati su firme. I virus metamorfici sono in grado di mutare completamente il proprio codice arrivando anche a dividerlo in diverse parti da inserire in zone separate del file infettato.

- **Worm,** A differenza dei virus, i worm non hanno bisogno di infettare altri file per propagarsi, in quanto riescono ad essere "attivi" e avviarsi tramite la modifica di procedure a livello di sistema operativo. I worm spesso non arrecano danni direttamente, ma possono avere effetti collaterali quali consumo di risorse e introduzione di ulteriore malware. La diffusione avviene, cosi come per il virus tramite email, file sharing e la distribuzione di software con bug. I worm sono capaci di procedere autonomamente utilizzando la mail per diffondersi, o possono diffondersi mediante lo sfruttamento di bug di programmi diffusi. In questi casi l'utente non ha nessuna possibilità di controllo poiché il meccanismo è completamente automatico e l'unica soluzione possibile è l'aggiornamento costante sia del sistema operativo che dei software installati. I worm permettono la creazione di botnet, ovvero reti di computer infetti alle dipendenze di gruppi criminali che portano avanti una serie di alcune categorie di attacchi informatici.

- **Trojan**, un trojan o un trojan horse, ovvero cavallo di troia, deve il suo nome alla somiglianza del suo funzionamento con quello dell'antico dono che permise ai greci di espugnare Troia, infatti le sue funzionalità sono nascoste all'interno di un programma che è apparentemente utile, infatti si presenta come un programma, solitamente gratuito, che mette a disposizione funzionalità utili in modo da invogliare l'utente ad eseguire il codice, celando però le sue vere finalità.

  L'attrazione delle funzionalità offerte ha un ruolo molto importante, questi programmi infatti spesso non sono in grado di auto replicarsi, di conseguenza le uniche vie per una larga diffusione sono l'inserimento del codice all'interno di worm o in alternativa invogliare gli utenti a scaricare il programma dal web.

  Oggi col termine trojan ci si riferisce tipicamente ai trojan ad accesso remoto (detti anche RAT dall'inglese Remote Administration Tool), composti generalmente da 2 file: il file server, che viene installato nella macchina vittima, ed un file client, usato dall'attaccante per inviare istruzioni che il server esegue, che può consentire di controllare a distanza le macchine vittime. Le principali finalità sono la creazione di backdoor o l'installazione di keylogger, con il fine ultimo di carpire informazioni sensibili, come ad esempio i numeri e i codici di carte di credito.

- **Backdoor,** sono paragonabili a porte di servizio che consentono di aggirare parte o tutte le procedure di sicurezza del sistema dove sono installate. Le backdoor possono essere intenzionalmente create dai gestori dei sistemi informatici per agevolare gli interventi di manutenzione da remoto, molto spesso però sono create da cracker intenzionati a manomettere il sistema, spesso le porte sono installate da altri malware, come virus, worm o trojan, per garantire un accesso da remoto. I requisiti essenziali di ogni backdoor, al di là della sua "potenza", sono sicuramente l'invisibilità, ovvero la capacità di eseguire comandi senza che l'utilizzatore principale se ne accorga, e la versatilità, ovvero la capacità di adattarsi per superare i diversi sistemi di sicurezza che ogni pc può avere.

- **Rootkit,** è un programma software prodotto per avere il controllo sul sistema senza bisogno di autorizzazione da parte di un utente o di un amministratore, infatti Il nome deriva dall'utente amministratore di sistemi Unix, chiamato root. Il rootkit può indicare due diversi concetti, il primo indica un pacchetto di funzioni che permettano di avere il completo controllo sulla macchina su cui viene installato, il secondo viene utilizzato invece per indicare altre tipologie di malware in grado di usare avanzate tecniche di mascheramento. Entrambi i casi hanno in comune l'utilizzo di moduli del kernel, librerie o driver di sistema che rendono molto difficile sia l'individuazione che la rimozione. La tecnica del malware è semplice quanto efficace, ogni software installato su un sistema, compresi gli antivirus, basano il proprio funzionamento sulle chiamate a funzioni messe a disposizione dal sistema operativo ospitante. Se è vero che questa tecnologia è fondamentale per il buon funzionamento del sistema operativo, negli anni sono stati creati cavalli di Troia e altri programmi maligni volti ad ottenere il controllo di un computer da locale o da remoto in maniera nascosta, ossia non rilevabile dai comuni strumenti di amministrazione e controllo. In più i rootkit possono inviare dei virus (spam e cookie) per email. I rootkit vengono tipicamente usati per nascondere delle backdoor. Negli ultimi anni, tuttavia, si è molto diffusa la pratica, tra i creatori di malware, di utilizzare rootkit per rendere più difficile la rilevazione di particolari trojan e spyware, indipendentemente dalla presenza in essi di funzioni di backdoor, proprio grazie alla possibilità di occultarne i processi principali. Grazie all'alto livello di priorità con la quale sono in esecuzione, i rootkit sono molto difficili da rilevare e da rimuovere con i normali software antivirus.

- **Keylogger**, è uno strumento il cui scopo è quello di registrare tutto ciò che un utente digita sulla tastiera del proprio computer con la speranza di intercettare informazioni preziose come password e numeri di carte di credito. Questo strumento può essere sia hardware che software, ovviamente i malware sono di tipo software. I keylogger software sono semplici programmi o driver di periferica che rimangono in esecuzione captando ogni tasto che viene digitato, in alcuni casi trasmettono tali informazioni a un computer remoto.

  Spesso sono trasportati e installati nel computer da worm o trojan ricevuti tramite Internet e hanno in genere lo scopo di intercettare password e numeri di carte di credito e inviarle tramite posta elettronica al creatore degli stessi. Per svolgere il loro compito posso usare alcune tecniche tipiche dei rootkit, come la modifica dei driver delle tastiere o la modifica delle specifiche librerie del S.O. Nonostante siano di difficile rilevazione esistono due tecniche di difesa. La prima è quella di installare plugin in alcuni dei software più critici (come i browser) che vanno ad installare un secondo driver in grado di creare un collegamento sicuro tra tastiera e browser mediante l'utilizzo di comunicazioni crittografate.

- **Spyware**, uno spyware è un software che raccoglie informazioni riguardanti l'attività di un utente che svolge in rete senza il suo consenso. Lo spyware che deriva dall'unione dei termini SPY softWARE può essere classificato in due tipologie di malware. La prima tipologia di spyware

raccoglie informazioni riguardanti l'attività online di un utente (siti web visitati, iscrizioni a servizi, acquisti etc.) senza il suo consenso, trasmettendole tramite Internet ad un'organizzazione che le utilizzerà per trarne profitto, solitamente attraverso l'invio di pubblicità mirata. La seconda, classe risulta essere più pericolosa, infatti oltre alle abitudini dell'utente, tiene traccia delle informazioni quali password e numeri di carte di credito. Quest'ultima categoria può essere installata in coppia con un keylogger. Lo spyware non è capace di propagarsi in maniera autonoma, si ha il bisogno di un intervento dell'utente per potersi installare.

- **Adware** Il termine deriva dalla contrazione inglese di ADvertising-supported softWARE, che indica un software supportato dalla pubblicità, ovvero quei programmi che durante il loro utilizzo mostrano avvisi pubblicitari permettendone la distribuzione gratuita o comunque a prezzi ridotti. Successivamente l'utente potrà poi decidere di rimuovere tali avvisi mediante il pagamento di una licenza. Di per sé questi software non rientrerebbero nella categoria dei malware se non fosse che spesso implementano alcune caratteristiche tipiche degli spyware per poter presentare all'utente avvisi pubblicitari focalizzati sui suoi interessi.
- **Hijacker** Questo malware è capace di modificare la homepage dei browser con lo scopo di settarla con pagine contenenti altri tipi di malware in grado di diffondersi mediante o i bug dei browser stessi o dei plugin in essi installati.
- **Ransomware** Più che un malware rappresenta una classe di malware abbastanza recente, che può ulteriormente essere divisa in: i) cryptovirus; ii) cryptotrojan; iii) cryptoworm. Come si può intuire dai nomi si tratta di particolari versioni di virus, trojan e worm in grado di criptare il contenuto dell'hard disk della vittima. Lo scopo è quello di poter successivamente ricattare la vittima chiedendo un riscatto in cambio della password per decifrare i documenti.
- **Dialer** Questa categoria sta lentamente scomparendo grazie all'avvento delle linee ADSL e fibra ottica. In origine il loro unico scopo era quello di creare connessioni telefoniche per potersi connettere ad internet, quindi un atteggiamento lecito, ma in seguito cominciano a diffondersi versioni programmate per contattare numeri di particolari servizi telefonici caratterizzati da costi molto elevati e distinguibili dal loro prefisso, in origine 144, poi 166 e infine 899 e 892. Grazie all'utilizzo di modem ADSL, il computer non ha più la possibilità di effettuare tali tipi di connessioni, di conseguenza il bacino di potenziali vittime si sta riducendo rapidamente.
- **Rabbit** Anche questa classe di malware sta perdendo popolarità, ma a differenza dei dialer non per particolari innovazioni tecnologiche, ma a causa di un cambiamento radicale nelle motivazioni che portano alla diffusione di malware. Oggi infatti la stragrande maggioranza di programmi malevoli in circolazione ha lo scopo di far ottenere un guadagno alle organizzazioni criminali, mentre i rabbit hanno il solo scopo di riprodursi in continuazione fino a saturare le risorse del sistema e a differenza dei virus non infettano alcun file. Analizzato quindi quelle che sono le principali tipologie di

malware oggi conosciute, vediamo quali possono essere i lor principali impieghi nell'ambito di attacchi più complessi.

- **Man in the middle** In questo caso l'attaccante si intromette nella comunicazione tra due terminali intercettandone il contenuto. In base alle caratteristiche della comunicazione quest'attività può essere più o meno complessa, ma la procedura standard prevede che l'attaccante faccia da intermediario nella comunicazione tra i due terminali senza che quest'ultimi se ne accorgano e per fare questo deve riuscire ad ingannare entrambi spacciandosi per l'utente all'altro terminale grazie a credenziali false. In questo caso un malware potrebbe servire per reindirizzare le connessioni verso la macchina dell'attaccante senza che l'utente vittima se ne possa accorgere.
- **Denial of Service (DoS)** Si traduce in Negazione di Servizio e si può dividere in due categorie: sfruttamento di bug e esaurimento delle risorse. Nella prima categorie rientrano diverse tecniche che si basano su bug software, protocolli insicuri o cattiva implementazione. Ultimamente però si tende ad utilizzare il termine DoS per indicare la seconda categoria, cioè quella in cui, lavorando su uno dei parametri d'ingresso, si cerca di portare il funzionamento di un sistema informatico che fornisce un servizio, ad esempio un sito web, al limite delle prestazioni fino a renderlo non più in grado di erogare il servizio. Solitamente i due parametri su cui si lavora sono la saturazione della banda o il riempimento delle code di gestione delle richieste. L'aumento esponenziale del traffico sulla rete Internet, ha portato ad un proporzionale aumento della banda disponibile, soprattutto lato server, visto che la maggior parte dei collegamenti domestici sono di tipo asimmetrico (Asymmetric-DSL), cioè con una limitata banda in upload. Questa evoluzione ha reso irrealizzabili attacchi DoS lanciati da singoli terminali o da piccoli gruppi di utenti. Nascono così gli attacchi DDoS.
- **Distribuited Denial of Service (DDoS)** Come si intuisce dal nome, si tratta dello stesso tipo di attacco sopra definito, ma realizzato in modo distribuito. Esistono alcuni attacchi DDoS basati su vulnerabilità dei protocolli, ma solitamente per realizzare questo tipo di attacco è prima necessario procurarsi un vasto bacino di zombie, termine usato per indicare macchine infette da malware e sotto il controllo degli attaccanti. Come visto in precedenza esistono vari tipi di malware, in questi casi si fa spesso riferimento a worm che, grazie alla loro capacità di diffondersi molto rapidamente, sono incaricati di aprire backdoor e creare così una botnet, cioè una rete di zombie pronti ad eseguire i comandi che gli verranno impartiti dai server C&C (Command & Control). Spesso quest'ultimi sono dei semplici server IRC a cui i malware si andranno a connettere una volta installati e che possono essere gestiti direttamente dai diffusori del worm o indirettamente da bande criminali che ne facessero richiesta.
- **Distribuited Reflection Denial of Service (DRDoS)** Rappresenta l'evoluzione degli attacchi DDoS. Gli effetti finali sono gli stessi, ma per rendere più difficoltoso il rintracciamento delle macchine attaccanti (che potrebbero quindi essere bloccate), ci si appoggia a server che mettono a disposizione servizi pubblici. Lo scopo è di far riflettere i propri attacchi da questi server, da qui il nome

Reflection, utilizzando la tecnica dello spoofing. In poche parole l'attaccante, o gli zombie a sua disposizione, mandano pacchetti TCP ai server pubblici utilizzando come indirizzo sorgente spoofato (contraffatto) quello della vittima che si vuol colpire, in questo modo i server pubblici risponderanno a quest'ultima che si ritroverà improvvisamente sommersa di richieste. Inoltre per evitare di sovraccaricare i reflection server, l'attaccante utilizzerà quest'ultimi a turno in modo da non superare eventuali livelli di soglia che porterebbero a notificare l'evento ai rispettivi amministratori.

- **Antisocial network** Questo termine rappresenta semplicemente una provocazione ed è stato coniato da alcuni ricercatori che hanno messo in evidenza come la notevole diffusione dei social network possa essere un potenziale rischio per la sicurezza. Il titolo della ricerca è eloquente: "Antisocial Networks: Turning a Social Network into a Botnet", la ricerca infatti propone un punto di vista critico su queste nuove piattaforme già note per la loro vulnerabilità a codice maligno. Cosa accadrebbe infatti, se una delle applicazioni più diffuse, inserite dagli utenti nelle proprie pagine, fosse in realtà studiata per sovraccaricare di richieste i server della vittima prescelta? Una risposta la danno i ricercatori stessi che pubblicano i risultati di un test da loro effettuato. Dopo aver scelto facebook.com come piattaforma di partenza, hanno messo a disposizione un programma che prometteva di visualizzare la foto del giorno del National Geographic sulla pagina personale dell'utente che l'avrebbe installato. Nella realtà il software nascondeva altre 4 richieste che andavano a prelevare un'immagine da 600KB presso un host vittima senza che l'utente se ne accorgesse. Nel caso in questione, l'host vittima era una macchina controllata dai ricercatori che nell'arco di pochi giorni hanno visto crescere il traffico generato dalle mille installazioni fino a raggiungere il picco di 300 richieste/ora e di 6Mbit al secondo di banda. Se a prima vista questi numeri potrebbero non sembrare così impressionanti, si sottolinea come il programma sia stato rimosso dopo pochi giorni e come le richieste fossero limitate a 4, mentre un ipotetico malware potrebbe generare centinaia di richieste e rimanere online molto più tempo con conseguenze devastanti, alcuni degli applicativi più diffusi di Facebook oltrepassano infatti il milione di utenti giornalieri.
- **Phishing** Questo tipo di attacchi, non si basa direttamente sulla distribuzione di malware, ma sullo Spam che questo può generare. L'interesse crescente per questa categoria è legato alla sua diffusione in continua crescita. Lo scopo è quello di ottenere i dati relativi ai conti correnti, sia bancari che postali, delle vittime. Per raggiungerlo, viene inizialmente creata una pagina web che rispecchi fedelmente la pagina di login della banca scelta come esca e che sia in grado di registrare i dati inseriti dagli utenti, dopo di ch´e si pubblica ad un indirizzo che contenga alcuni nomi che richiamino l'URL della pagina originale, anche se ovviamente il dominio sarà diverso. A questo punto parte una campagna di Spam focalizzata il più possibile sull'area geografica interessata all'ente utilizzato come esca. Il termine Spam sta ad indicare email indesiderate, spesso contenenti pubblicità, ma che nel caso del phishing contengono messaggi appositamente studiati per allarmare

l'utente invitandolo poi a risolvere eventuali problemi dopo aver effettuato il login. Dopo questi avvisi viene riportato il link alla pagina web creata precedentemente, mascherato da sotto il titolo dell'URL originale. L'utente sprovveduto che dovesse cadere nella trappola si ritroverebbe con le credenziali rubate e ben presto col conto corrente svuotato. In questi casi è facile difendersi, innanzitutto è sempre bene diffidare di certi tipi di email, spesso la lingua utilizzata presenta diversi errori grammaticali (soprattutto in Italia), poi è buona abitudine diffidare dei link proposti, quindi aprire una nuova finestra dove poter digitare l'indirizzo che già si conosce.

- **Pharming** Molto simile al phishing con cui condivide le finalità e anche tutta la parte di preparazione. Fino alla realizzazione delle pagine web contraffatta infatti, le tue tecniche di attacco sono identiche, la differenza sta nel come si attirano le potenziali vittime. A differenza del phishing, nel pharming non si mandano email di spam, ma si attaccano i server DNS sparsi per la rete. Questi server sono di fondamentale importanza e servono per fornire ai computer la traduzione delle URL in indirizzi IP. L'attacco a questi server è molto più complesso rispetto all'inoltro di spam e ha come scopo sostituire l'indirizzo IP originale della banca con quello dei propri server. A questo punto l'utente che tentasse di connettersi per esigenze personali (e non più su richieste via email) verrebbe indirizzato sulla pagina contraffatta senza accorgersene. Questi attacchi sono di difficile realizzazione, ma se portati a termine mieteranno vittime anche tra gli utenti più esperti, questo perché non esiste modo di rendersi conto di quello che sta accadendo, salvo quello di far precedere l'autenticazione da un'approfondita navigazione, in questo caso infatti potrebbero uscire allo scoperto lacune nella contraffazione del sito originale.

3.11.2 Classificazione dei sistemi di Intrusion Detection

I sistemi di Intrusion Detection possono essere classificati in base al sistema che va a monitorare.

3.11.2.1 Host Intrusion Detection System (HIDS)

Questa tipologia di detection è caratterizzato da un agente che monitora e analizza i file di log del sistema host per la ricerca di intrusioni. Questa tipologia di IDS ha il vantaggio di essere installata sulla macchina end-point per l'aggressore, di conseguenza le tecniche di offuscamento degli attacchi non hanno alcun effetto. Lo svantaggio è relativo ad una visione limitata delle operazioni e attività sulla singola macchina. Un esempio di questa tipologia è Ossec (http://www.ossec.net/).

3.11.2.2 Network Intrusion Detection System (NIDS)

Questa classe di IDS ha l'obiettivo di analizzare il traffico di rete per identificare le intrusioni e per fare ciò monitora non solo un singolo host ma una rete completa. Si tratta di un sistema che ispeziona (in gergo sniffa) il traffico che passa sul segmento di rete a cui sono connessi i suoi sensori, cercando tracce di attacchi. A tale scopo il dislocamento di questi ultimi è di fondamentale importanza e dev'essere studiato

attentamente per garantire una quantità di traffico adeguata. Questa categoria, anche se soggetta a tecniche di offuscamento degli attacchi, consente una visione generale dello stato della rete. Un esempio di Network Intrusion Detection System è Snort, un prodotto open source e probabilmente il più diffuso.

3.11.2.3 Application Protocol Intrusion Detection System (APIDS)

Questa tipologia di IDS analizza il traffico concentrandosi su specifici protocolli applicativi, dei quali conosce le diverse implementazioni ed è in grado di notare eventuali anomalie.

3.11.2.4 Hybrid Intrusion Detection System

Come indica il nome si tratta di un modello ibrido, ovvero implementa più tecniche cercando di sfruttare i vantaggi di ognuna. I modelli più diffusi si basano su HIDS e NIDS e offrono sia una panoramica sullo stato della rete che un'analisi più specifica per ogni singolo host. Un esempio di questa categoria di IDS è rappresentato da Prelude.

Esiste infine un'ultima classificazione e viene fatta tra gli IDS passivi e quelli attivi:

- **passivi** sono la versione standard, si limitano alla segnalazione degli eventi rilevati e per loro vale tutto quello scritto in precedenza.
- **attivi** oltre alle funzioni di segnalazione, implementano metodi per intervenire sul sistema in modo da porre fine alle minacce rilevate. Gli interventi più diffusi riguardano le modifiche delle access list dei firewall, in modo da interrompere le connessioni con indirizzi esterni alla rete locale.

Ulteriori miglioramenti hanno reso disponibile il blocco anche del traffico interno alla rete, potendo così interrompere anche la diffusione di malware tra host interni. Tutte queste funzionalità hanno portato alla creazione degli IPS (Intrusion Prevention System).

3.11.3 Intrusion Prevention System

Rappresentano l'evoluzione degli IDS e non esiste in realtà una linea di demarcazione netta nei confronti delle versioni attive di quest'ultimi. In alcuni casi si tende ad identificare col termine IDS quelli esclusivamente passivi, mentre con IPS tutti gli strumenti che permettono di intervenire attivamente, in altri si aggira la questione utilizzando il termine IDPS, Intrusion Detection and Prevention System. Hanno molte similitudini col funzionamento dei firewall: sono entrambi in line, quindi fail close, e agiscono entrambi mediante l'utilizzo di liste di controllo degli accessi. Gli IPS, però sono più spostati a livello applicativo e nelle liste citate tendono ad inserire coppie utente-programma, piuttosto che indirizzo IP-porta. Esistono due categorie principali di questi strumenti.

Host IPS (HIPS)

come si deduce dal nome si tratta di implementazioni a livello di singolo host. Il controllo può essere molto

dettagliato e per ogni programma che dovrà essere eseguito sul sistema è possibile definire specifiche policy. Se non impostato in modo preciso, può risultare particolarmente invadente per l'utente dell'host.

Network IPS (NIPS)

hanno una visione a livello di rete, quindi più ampia, e si dividono in tre principali sottocategorie:

- **Protocol analysis IPS (PIPS):** ispezionano il traffico a livello di protocolli applicativi, come HTTP e FTP, cercando anomalie o violazioni degli standard.
- **Content Based IPS (CBIPS):** analizzano il payload dei pacchetti utilizzando solitamente tecniche basate su signature analysis, con tutti i vantaggi e svantaggi tipici di tale approccio.
- **Rate Based IPS (RBIPS):** si basano principalmente su tecniche di anomaly detection. Esiste una fase iniziale di studio delle statistiche sul traffico relativo alla rete monitorata, in particolare si osservano le percentuali (rate) dei pacchetti divisi per protocolli (TCP, UDP, ARP, ICMP). Finito lo studio iniziale l'IPS entra in funzione affiancato da un sistema di autoapprendimento, in modo da garantire un miglioramento delle prestazioni. Questo approccio risulta particolarmente efficace nel rilevamento di attacchi DoS o DDoS.

La scelta tra un HIPS o un NIDS va effettuata in base alle proprie esigenze, ma occorre tenere in considerazione come la seconda opzione richieda maggiori risorse e risulti un punto critico del sistema (single point of failure) a causa del suo comportamento fail close citato in precedenza.

4 Framework per la sicurezza dei dati

Questo paragrafo descrive il framework di sicurezza proposto, diagrammato in Figura 2, comprendente le principali componenti di sicurezza da tenere in considerazione, illustrando le tecniche e tecnologie più adatte alla risoluzione delle problematiche evidenziate nel capitolo precedente.

Il framework di sicurezza proposto identifica i seguenti obiettivi fondamentali per la corretta implementazione della sicurezza:

- Identificazione di attori e/o eventi che agiscono o coinvolgono i dati
- Protezione dei dati
- Rilevamento delle intrusioni

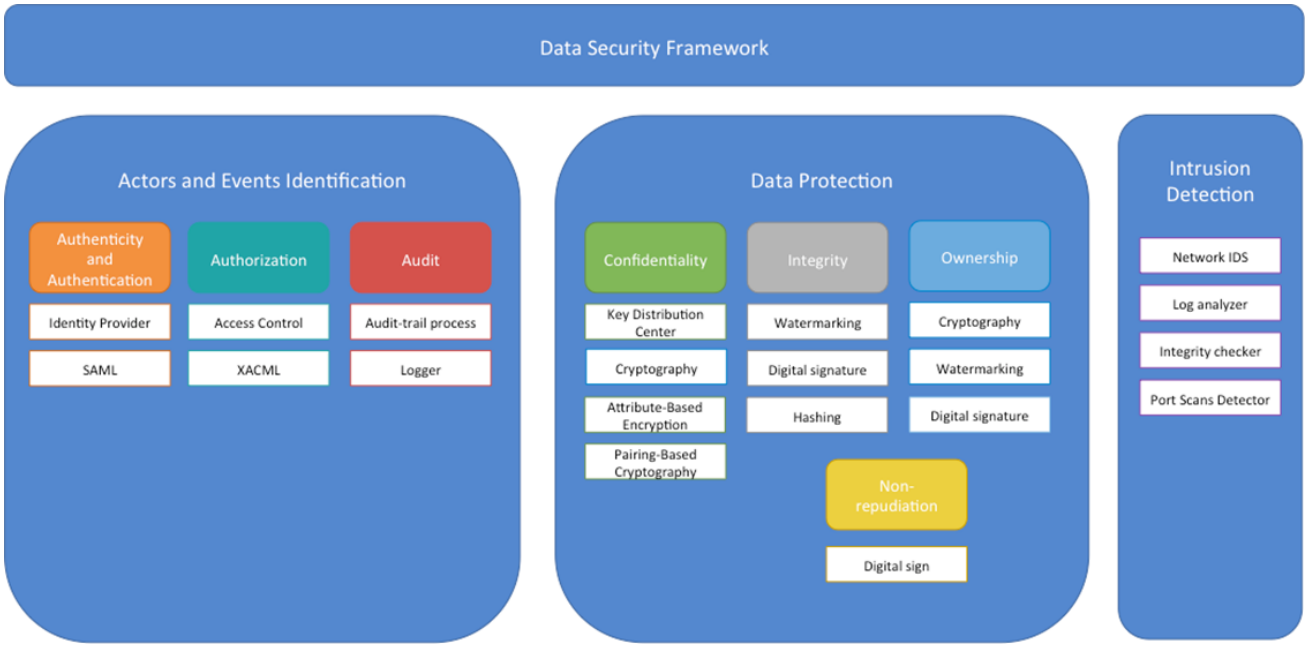

Figure 2 - Il framework di sicurezza proposto.

4.1 Identificazione di attori e/o eventi che agiscono o coinvolgono i dati

4.1.1 Autenticità e autenticazione

Un sistema di controllo degli accessi deve essere considerato come un componente indipendente dal sistema che lo contiene. Infatti, la tipologia di componenti presenti nel sistema dipende dalla natura dell'architettura realizzata, ma i componenti dedicati all'autenticità e all'autenticazione saranno sicuramente presenti, per fornire gli strumenti necessari alla costruzione delle altre funzionalità del sistema.

Per garantire autenticità e autenticazione dei dati e delle azioni in scenari altamente complessi composti da diversi sistemi e entità eterogenee bisogna definire soluzioni in grado di garantire e controllare l'accesso ai dati sia in termini di identificazione degli attori che richiedono l'accesso che in termini di identificazione delle funzionalità utilizzabili sulle risorse e/o dati richiesti. A tale scopo, si propone la definizione e utilizzo di servizi dedicati di tipo Identity Provider (IdP) e Service Provider (SP) con cui sia necessario comunicare per poter opportunamente accedere e scambiare dati.

Per quanto riguarda la fase di autenticazione, lo standard tipicamente utilizzato è Security Assertion Markup Language (SAML), che basa sull'utilizzo di asserzioni, cioè messaggi scambiati tra le entità dell'infrastruttura contenenti informazioni sulla sicurezza.

In uno scenario tipico, l'utente richiede un servizio al SP che deve ricevere dall' IdP una identity assertion e sulla base di questa asserzione, decidere se consentire o meno l'accesso alla risorsa. Prima di inviare l'asserzione al SP, l'IdP potrebbe richiedere dei dati all'utente (username e password) al fine di autenticarlo. Un IdP può inviare asserzioni a diversi SP, così come è possibile che un SP accetti asserzioni da più IdP.

Quindi, l'IdP ha il compito di inviare opportune asserzioni al SP contenenti "statement", sulla base dei quali il SP prenderà la decisione se consentire o meno l'accesso dell'utente ad una data risorsa. Esistono tre tipi di statement:

- **Authentication statements.** Contengono l'authentication context con le informazioni in merito all'autenticazione, sono utilizzati per garantire al SP che l'utente si sia effettivamente autenticato tramite l'IdP.
- **Attribute statements.** Comunicano l'associazione di certi attributi all'utente (sono una coppia nome-valore) in modo che sia possibile permettere o meno l'accesso dell'utente a determinate risorse.
- **Authorization decision statements.** Comunicano l'avvenuta autorizzazione data all'utente U relativamente al proseguimento dell'azione A in atto sulla risorsa R.

Il protocollo SAML viene utilizzato per definire la struttura dei messaggi scambiati, e quindi di come i messaggi di richieste e di risposta devono essere composti e interpretati. In particolare, un messaggio di richiesta (risposta) scambiato tra due entità tramite il protocollo SAML prende il nome di query.

Di seguito, si riportano le tipologie di query utilizzabili:

- Authentication query
- Attribute query
- Authorization decision query

Come detto, il protocollo SAML definisce la struttura dei messaggi, mentre per la comunicazione degli stessi bisogna utilizzare un opportuno protocollo di comunicazione. Il mapping tra un messaggio definito nel protocollo SAML e il protocollo di comunicazione utilizzato prende il nome di processo di SAML binding.

Ad esempio, il binding più comunemente utilizzato viene realizzato incapsulando messaggi SAML in messaggi SOAP, che a loro volta sono poi racchiusi in messaggi HTTP.

Quindi, l'effettivo scambio di query SAML tra due entità dell'infrastruttura comporta un insieme di azioni che devono essere effettuata rispettando i diversi protocolli coinvolti. L'insieme delle specifiche che bisogna seguire per scambiare correttamente messaggi SAML in un dato contesto, viene definito tramite un SAML profile, il più importante dei quali è il web-browser SSO.

4.1.2 Autorizzazione

Per realizzare servizi di autorizzazione, lo standard tecnologico tipicamente utilizzato è XACML, che definisce sia un linguaggio per la specifica di politiche di sicurezza, sia un processo di valutazione delle

richieste autorizzative secondo il modello PBAC. Nello standard XACML la richiesta di accesso a una risorsa viene specificata, tramite una sintassi XML-based, dall'insieme di attributi che descrivono il soggetto, l'azione e la risorsa coinvolti.

L'autorizzazione o meno alla richiesta pervenuta da un soggetto S di proseguire un'azione A su di una determinata risorsa R viene decisa tramite combinazione delle regole politiche di sicurezza applicabili alla richiesta stessa.

Nell versioni precedenti di XACML il processo di autorizzazione delle richieste dipendeva soltanto dalle regole applicate sugli attributi. Nelle versioni più recenti, invece, il processo di autorizzazione dipende anche dal risultato di azioni accessorie eseguite in fase di applicazione della decisione. Nel caso queste azioni non fossero eseguite con successo, si avrebbe la negazione dell'accesso alle risorse, creando quindi una maggior dinamicità delle politiche, in quanto l'autorizzazione non dipende soltanto dalle regole definite, ma anche da azioni eseguite a tempo di valutazione.

Lo standard XACML non si occupa soltanto della valutazione delle richieste autorizzative, ma prevede quattro domini applicati:

- valutazione delle richieste autorizzative;
- applicazione della decisione presa;
- gestione delle politiche di sicurezza;
- dialogo con i componenti esterni.

A tale scopo, XACML descrive un insieme di ruoli e le interazioni tra di essi, il cui flusso delle informazioni tra i ruoli è gestito da un context handler, allo scopo di delineare la struttura del sistema di controllo, senza però forzare specifiche soluzioni implementative.

Per quanto riguarda la valutazione delle richieste autorizzative, XACML utilizza il dialogo tra il sistema di controllo degli accessi e l'esterno, XACML prevede il ruolo del Policy Decision Point (PDP), che è quello definito con maggior dettaglio nello standard. L'applicazione delle decisioni prese e il dialogo con l'esterno, invece, è demandato al ruolo del Policy Enforcement Point (PEP). La gestione delle politiche, infine, è associata al Policy Information Point (PIP).

La definizione di ruoli e non di entità ben precise, permette di definire le funzionalità del sistema, lasciando la possibilità di integrarle, a seconda delle implementazioni, nei componenti opportuni. Ad esempio, si può avere un sistema definito per richieste non sviluppate in formato XACML, che devono quindi essere trasformate nella sintassi richiesta dal PDP. Questo compito può essere svolto dal PEP oppure dal context handler prima della chiamata del PDP.

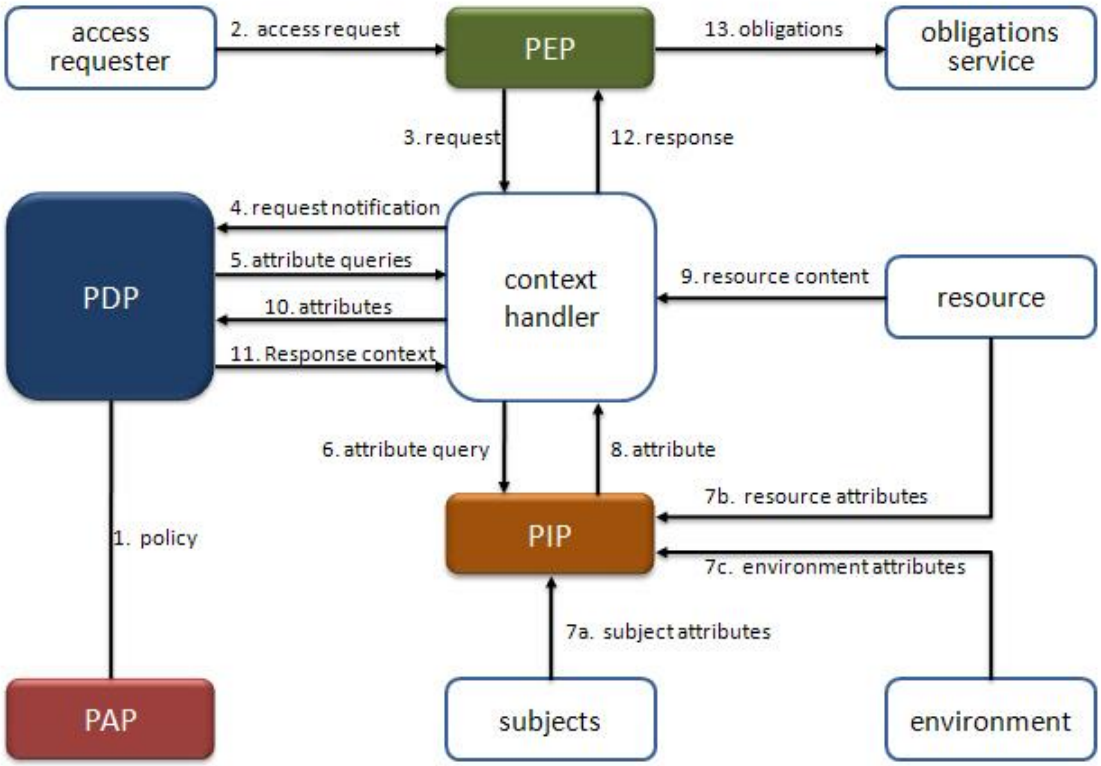

Figure 3 - Architettura XACML.

Il processo di autorizzazione si basa sui valori degli attributi presenti nelle richieste e può dipendere anche dal contesto di valutazione esterno alle richieste. Con contesto si intende l'insieme delle informazioni collegate all'ambiente di valutazione delle richieste; un tipico esempio è l'attributo tempo che indica la data e l'ora corrente. L'attributo è l'unità che contiene le informazioni nel contesto e nelle richieste ed è formato da una coppia nome-valore. Lo standard definisce una sintassi specifica per gli attributi, ma non è necessario utilizzarla nelle implementazioni.

Il flusso delle informazioni tra i ruoli prima presentati descrive il processo di valutazione alla XACML raffigurato in Figura 3. Si noti che XACML definisce soltanto il comportamento e i requisiti dei ruoli strettamente coinvolti nel processo di decisione, mentre gli altri sono definiti soltanto per requisiti generali. Ad esempio, il PDP è dettagliato in ogni comportamento e requisito funzionale, in quanto esso calcola la decisione associata a ogni richiesta., è dettagliato in ogni comportamento e requisito funzionale, a differenza degli altri ruoli definiti soltanto per requisiti generali. La definizione del ruolo del PDP, essa è effettuata tramite una sintassi specifica degli elementi request (richiesta), response (risposta) e attribute (attributo) presenti in figura 3.

Per quanto riguarda il ruolo del Policy Administration Point (PAP), invece, sono specificati soltanto a sintassi delle politiche senza stabilire requisiti funzionali per la sua implementazione. Infatti, poiché l'effettiva gestione delle politiche nel PAP, quali ad esempio

salvataggio, aggiornamento, modifica e recupero, dipendono dai vincoli applicativi e dalle tecnologie

utilizzate.

Il PDP può comunicare con gli altri ruoli tramite il context handler allo scopo di ricevere informazioni aggiuntive sulle risorse di cui deve valutare l'autorizzazione. Anche in questo caso, XACML non fornisce modalità implementativi sull'effettiva realizzazione di tale comunicazione. In proposito, il ruolo del PIP e le entità collegate al context handler potrebbero essere oggetto di una definizione più completa in futuri standard OASIS.

Per quanto riguarda il ruolo del PEP, XACML definisce soltanto le linee guida generali da seguire nella sua realizzazione, senza dare molti dettagli. Il PEP, in generale, è un livello di astrazione che implementa le azioni non eseguibili direttamente dal PDP e gestisce la comunicazione con l'esterno. Ad esempio, con riferimento alla figura 3, Il PEP per applicare la decisione ricevuta con l'elemento response ha il compito di eseguire e valutare tutte le azioni accessorie presenti e, in base all'esito delle azioni, permettere l'accesso alla risorsa o negarlo.

Ad esempio, in un servizio web che implementa il modello di valutazione XACML, il PEP può essere svolto dallo stub che riceve le richieste, spacchetta i vari attributi organizzandoli in richieste per il PDP e, una volta ricevuta la decisione, la comunica al client.

XACML non specifica le azioni eseguibili dal PEP e le politiche di sicurezza indicano soltanto gli identificatori e gli argomenti delle azioni. Quindi, affinché venga garantita la corretta esecuzione delle azioni, bisogna trovare un accordo tra chi realizza il ruolo del PEP e chi realizza le politiche di sicurezza allo scopo di associare lo stesso significato agli identificatori.

Infine, si noti che il ruolo del context handler è in generale garantire la comunicazione tra PDP, PEP, PIP e gli altri ruoli presenti in Figura 3, senza avere ulteriori compiti specifici. Tipicamente, il ruolo del context handler viene specificato all'interno del PDP o del PEP a seconda delle scelte applicative.

Riassumendo, durante il processo di valutazione di una richiesta, essa viene inviata al PEP che non fa altro che inoltrarla al context handler. Il context handler, a sua volta, prima formatta la richiesta secondo la sintassi XACML e poi provvede a inviarla al PDP. Il PDP si occupa del processo di valutazione della richiesta e, nel caso necessiti di attributi aggiuntivi, li richiede al context handler. Se li context handler è già in possesso degli attributi aggiuntivi, avendoli ricevuti in precedenza dal PEP, li fornisce al PDP altrimenti, se non li possiede, interroga il PIP. Una volte che il PDP ha ricevuto tutte le informazioni che gli necessitano per la sua valutazione, provvede alla stessa e ne invia la risposta al context handler, indicando sia la decisione presa sia eventuali azioni accessorie che devono essere eseguite prima di concedere l'autorizzazione finale. A tal proposito, la risposta del PDP viene inoltrata al PEP che, da un lato, esegue le eventuali azioni accessorie contenute nella risposta e, dall'altro, concede l'accesso alla risorsa in funzione del risultato delle eventuali azioni accessorie collegate.

4.1.3 Audit

Per poter offrire un adeguato servizio di audit nei sistemi software del progetto, si propone di creare un servizio dedicato alla gestione e aggiornamento di un audit-trail. Un audit-trail (detto anche audit-log) è un registro di informazioni rilevanti memorizzate cronologicamente, finalizzato alle tracciature della sequenza di attività che interessano, nel tempo, specifiche operazioni, procedure, o eventi. In particolare, il processo incaricato di gestire l'audit-trail del sistema deve poter essere eseguito in modalità privilegiata, così da poter sempre accedere e supervisionare tutte le azioni effettuate da tutti gli utenti. Inoltre, è fondamentale che i sistemi non permettano agli utenti normali di accedere al processo di gestione dell'audit-trail, e che non sia in grado di fermarlo o modificarlo. Allo stesso modo, il database contenente i records degli audit-trail non deve essere accessibile dagli utenti non autorizzati.

A tale scopo, si consiglia di implementare un role-based security model che governi e disciplini gli accessi al processo di audit, ed al database degli audit-trail gestiti.

4.2 Protezione dei dati

4.2.1 Audit: titolarità delle informazioni

Esistono differenti tecniche per proteggere la titolarità dei dati. Ad esempio, è possibile ricorrere ad una combinazione di tecniche di cifratura o di watermarking (letteralmente "filigranatura"), che possono essere applicate ai dati contenuti in un documento. Nel primo caso, il documento contiene il risultato dell'attuazione di un apposito algoritmo di cifratura sul suo contenuto in chiaro. Nel secondo caso, si effettua l'inclusione di informazioni all'interno del documento da proteggere. Tali informazioni aggiuntive devono rendere manifesto a tutti gli utenti chi sia il proprietario del documento. Uno dei problemi della cifratura è la gestione delle chiavi, ovvero dove memorizzare le chiavi e come proteggerle quando le si usano, al fine di evitare che siano violate o rubate. La soluzione allo scambio e alla protezione delle chiavi può essere a carico delle applicazioni. Anche le tecniche di watermarking possono essere soggette a sofisticati tipi di attacchi volti a violarne le garanzie di sicurezza.

4.2.2 Integrità

Anche in questo caso è possibile usufruire di tecniche come il watermarking, che possono essere usate per dimostrare l'originalità di un documento non contraffatto (in particolare viene applicata una "filigrana digitale" al documento).

Un'altra tecnica di largo uso è quella della firma digitale. Dato un documento ed una chiave di cifratura privata, viene generata una firma a partire dall'impronta del documento. Quando si vuole verificare l'integrità del documento, la sua firma viene processata con una chiave di cifratura pubblica e se il risultato corrisponde all'impronta del documento allora quest'ultimo non ha subito alterazioni. Come la cifratura,

anche il processo di firma digitale può essere gestito direttamente dall'applicazione, oppure affidata ad un servizio offerto dalla piattaforma.

4.2.3 Non ripudio

Tecniche di cifrature a chiave asimmetrica e di firma digitale offrono una soluzione al requisito di non ripudio del mittente. Tecniche basate sulla spedizione di ricevute di ritorno offrono invece soluzioni al requisito di non ripudio del destinatario.

4.2.4 Confidenzialità

Al fine di garantire la confidenzialità dei dati, tipicamente si utilizzano due schemi di crittografia:

- Simmetrico, un'unica chiave di cifratura viene utilizzata sia per criptazione che per la decriptazione;
- Asimmetrico, due chiavi distinte sono utilizzate, una per criptare i dati, e un'altra per la loro decriptazione.

Per quanto riguarda la crittografia simmetrica, essendo presente un'unica chiave di cifratura, il fornitore della stessa deve preoccuparsi di come diffondere la chiave a tutti gli attori che ne abbiano bisogno, oppure ricorrere ad un Key Distribution Center (KDC).

La crittografia asimmetrica è nota anche come crittografia a chiave pubblica, poiché la chiava di criptazione può essere resa pubblica e accessibile, in quanto la chiave di decriptazione associata non può essere derivata dalla chiave pubblica. La presenza di una doppia chiave di criptazione richiede una gestione molto più semplice delle chiavi stesse, e della confidenzialità necessaria alla loro diffusione. Infatti, soltanto le chiavi pubbliche necessitano di essere diffuse e possono esserlo senza pericolo, in quanto le chiavi private che restano invece segrete, non sono derivabili da esse. Lo svantaggio è che la cifratura asimmetrica richiede una complessità molto maggiore data la complessità maggiore delle chiavi utilizzate. Tradizionalmente, gli schemi di crittografia sono sempre definiti in modo da risultare deterministici, cioè dato un plaintext (testo in chiaro) esso sarà sempre trasformato in un unico ciphertext (testo cifrato). Ciò può facilitare un intercettatore malintenzionato nella possibilità di eseguire una analisi statistica dei messaggi trasmessi, e la correlazione di ciphertexts intercettati. Per affrontare tali problematiche, è possibile utilizzare schemi di crittografia probabilistica, come la cifratura omomorfica di Pailler, dove un plaintext può essere trasformato in più ciphertexts, e i ciphertexts effettivamente trasmessi sono scelti casualmente. Nonostante sia molto più robusta rispetto alla cripto-analisi, la crittografia probabilistica soffre di degrado delle performance in quanto i messaggi da scambiare hanno una dimensione molto maggiore rispetto alle alte tecniche di crittografia. Relativamente alla chiave di crittografia, essa spesso è costituita da una stringa non correlata all'identità del firmatario, quindi spesso risulta utile implementare un'Autority dei certificati. D'altro canto, l'uso dei certificati può causare inefficienza e overhead, che può essere risolta utilizzando un sistema di crittografia Identity-based, come ad esempio nella Pairing-Based Cryptography (PBC). In dettaglio, la chiave pubblica

di un utente è più facilmente processabile per correlarla all'identità dell'utente tramite tecniche di bilinear pairings, e senza utilizzare una certificate authority, ma un key manager è necessario per assegnare le chiavi private alle chiavi pubbliche relative. Si noti che, da un lato, visto il coinvolgimento di un manager, i sistemi di crittografia identity-based sono affetti dal problema della key escrow problem, che rende tali sistemi meno sicuri. D'altro lato, PBC permette la riduzione del numero delle chiavi da gestire. Se l'identità dell'utente è espressa tramitie un set di attributi, la cifratura può essere possibile soltanto se almeno k attributi della chiave dell'utente combaciano con gli attributi del ciphertext. Tale soluzione, nota come Attribute-Based Encryption (ABE), è tipicamente utilizzata per implementare l'access control su dati criptatati.

4.2.5 Intrusion Detection

Alla luce delle problematiche di intrusion detection analizzate nel capitolo precedente, la soluzione proposta prevede di realizzare un sistema NIDS in grado di:

riconoscere qualsiasi attività o evento sospetto che potrebbe potenzialmente essere un attacco;

- essere in grado di adattarsi ai cambiamenti dei metodi di un attacco;
- saper gestire attacchi multipli;
- essere scalabile e manipolabile;
- auto proteggersi dagli attacchi.

Un buon NIDS è un ottimo strumento per avere un'idea della variegata fauna di pacchetti che arrivano ad una rete pubblica e, se ben configurato, può indubbiamente troncare sul nascere molti tentativi di intrusione.

Alcune raccomandazioni di massima valgono per ogni NIDS:

- selezionare con cura le regole di packet matching, cercando di evitare falsi positivi troppo verbosi o relativi a rischi molto relativi (il logging di ogni PING ai nostri sistemi è assolutamente inutile).
- tenere ragionevolmente aggiornate le regole di matching, appoggiandosi ai database online.
- tenere in un luogo particolarmente protetto la macchina centrale, se esistono diverse sonde nella rete, o comunque quella in cui i dati vengono raccolti ed elabotati.
- controllare con costanza le segnalazioni di warning e minacce, avendo la pazienza di rifinire le proprie configurazioni per evidenziare solo gli eventi più significativi.
- non esagerare a implementare, o quantomeno verificare con regolarità, meccanismi proattivi che bloccano ogni comunicazione con IP da cui arrivano pacchetti molesti. Questi IP possono venire spoofati e un soggetto ostile può creare una sorta di auto DOS attack, facendo bloccare al nostro IDS

comunicazioni con normali e validi indirizzi IP.

A tale scopo, anche se i meccanismi di individuazione di attività sospette possono essere di diversa natura, si propone di concentrarsi maggiormente:

- verifica dei log di sistema o di specifici programmi per individuare attività anomale;
- controllo dell'integrità dei file locali, modifiche sospette possono essere sintono di una avvenuta irruzione;
- monitoring dei pacchetti destinati all'host, sia per reagire a pattern di attacco noti che per accorgersi di un port scan da remoto, generalmente prologo di un tentativo di intrusione.

Il mondo OpenSource offre una moltitudine di strumenti utili per questi scopi, si va da piccoli programmi per specifiche attività a sistemi più complessi di qualità evolute. Per essere veramente efficace, l'implementazione di sistemi di Intrusion Detection oltre a richiedere un sostanzioso intervento sistemistico per la configurazione e la customizzazione del software usato, deve essere tale da permettere una rapida analisi: troppi log e mail di notifica sono alla lunga controproducenti e inutili se a questo non segue un controllo effettivo.

Segue un breve elenco di alcuni dei programmi più noti per le attività di Intrusion Detection.

4.2.6 Log Analyzers

Sono programmi che monitorano le entry nei file di log di sistema e possono essere configurati per eseguire date operazioni in presenza di determinate righe di log. È bene che agiscano in tempo reale, dal momento che dopo una intrusione una delle prime occupazioni di un hacker è quella di cancellare le tracce eventualmente lasciate sui log vari. In questa categoria possono rientrare:

- Swatch - Può monitorare in tempo reale ogni tipo di file. È in Perl e richiede alcuni moduli generalmente non installati di default e scaricabili da CPAN, nei file di configurazione si settano le regole di pattern-matching dei log e i comportanti da adottare.
- Logsurfer - Ha caratteristiche simili a Swatch ma presenta alcuni miglioramenti, tra cui la possibilità di correlare gli output di diversi log e, al contempo, propone un file di configurazione più complesso (è consigliabile ispirarsi agli esempi di conf esistenti).
- LogWatch - Installato di default su alcune distribuzioni Linux come RedHat monitora diversi tipi di log, secondo impostazioni configurabili e genera i relativi alert e report.

A prescindere dal meccanismo di controllo dei log utilizzato, alcuni accorgimenti rendono l'operazione più

efficace:

- loggare, se possibile, su una macchina remota, in modo tale da impedire la manipolazione dei log dopo un'intrusione;
- definire le regole di log checking secondo una policy di logging di tutto tranne dei messaggi noti: definire delle regole di esclusione di righe di log lecite; definire regole per l'inclusione di speciali e noti eventi sospetti; includere tutto il resto (righe di log ignote o non previste).
- non eseguire i programmi di log check come utente root, eventuali stringhe maligne nei log potrebbero generare risultati incontrollabili;
- non visualizzare i log da un terminale potenzialmente sensibile ad un "TERMINAL EMULATOR ATTACK" tramite caratteri escape che vengono mal utilizzati o interpretati da certi terminali.

4.2.7 File Integrity Checkers

Una volta fatta con successo un'intrusione, oltre a guardarsi intorno e cercare di capire dove si trova, un hacker che vuole mantenere l'accesso e la possibilità di rientrare nel sistema provvede ad cancellare le proprie tracce dai log, ad installare trojan e programmi che aprano nuovi accessi remoti, controllino le attività degli amministratori (packet sniffers, key loggers...) servano per attacchi successivi, lanciati dall'host già violato. Queste attività sono facilitate e in parte automatizzate da rootkit dedicati ma, in ogni caso, lasciano tracce sul sistema: nuovi file copiati, log e binari modificati, cancellazioni ecc.

Gli strumenti di Integrity Check aiutano ad individuare simili manipolazioni e generalmente registrano cambiamenti nella data di creazione o modifica di un file, alterazioni dei permessi, degli attributi o dello contenuto di file di configurazione, binari di comandi più o meno comuni, testi di log ecc. Alcuni strumenti che si possono usare per l'Integrity Check sono:

- Tripwire – È uno dei primi, più evoluti e utilizzati sistemi di Integrity Check. Oltre alla versione OpenSource ne esistono complementi proprietari che facilitano l'integrazione e la centralizzazione dei dati da diversi sistemi remoti, rendendo più agevole il monitoraggio di molti host.
- Aide - Si presenta come l'alternativa completamente Free a Tripwire, ha una logica simile e prevede controlli della stessa natura tramite una varietà di algoritmi di checksum.
- Integrità - Altra promettente e performante alternativa a Tripwire che si presta ad essere integrata in un sistema di monitoring che utilizza diversi strumenti.
- Chkrootkit – È una serie di script dedicati alla individuazione di rootkit noti. Oltre a controllare lo stato di alcuni binari esegue controlli di altra natura (verifica sul proc filesystem ecc.), ma non va considerato come una soluzione generica.

Una caratteristica comune di questi e altri Integrity Checkers è quella di creare una snapshop preliminare dello stato dei file di un host "pulito", adattare la configurazione per il proprio specifico sistema, eliminare controlli che producono eccessivi false-positive (troppi warning tendono ad essere ignorati) e schedulare periodicamente un check dello stato attuale del sistema rispetto allo snapshot iniziale.

In tutti i casi ci sono alcune precedure di base che è giusto seguire per migliorare la sicurezza e l'affidabilità di simili strumenti:

- copiare i database di snapshot su un sistema remoto o comunque un mezzo non scrivibile dall'host a cui si riferiscono;
- considerare che un checksum non è infallibile ed esistono metodi per mantenere lo stesso checksum in un file modificato (quantomeno per md5, l'algoritmo più utilizzato in questi casi);
- controllare effettivamente i log generati e rifinire gradualmente la configurazione per evitare segnalazioni errate, falsi positivi, per file che vengono modificati a causa di normali attività sul sistema.

4.2.8 Port Scans Detectors

Preludio ad ogni tentativo di intrusione è quasi sempre un network scanning remoto, con cui l'attaccante cerca di individuare le porte aperte sui sistemi bersaglio. Nonostante le tecniche di port scanning siano piuttosto evolute (nmap, per esempio, permette almeno 6 diversi tipi di scanning, più o meno "stealth") esistono sistemi per individuarle e, quindi, sapere prima ancora di subire l'attacco, quali IP remoti stanno raccogliendo informazioni sui propri sistemi. Sebbene ogni cracker accorto cercherà di non eseguire alcuna operazione di probing o hacking, direttamente dal proprio IP, sapere da quali indirizzi può provenire una minaccia è sempre utile.

I programmi più noti per individuare un port scanning sono:

- ScanLogD - viene eseguito come un demone, costantemente in monitoraggio di collegamenti a porte TCP locali. Utilizza dei metodi per evitare disservizi o problemi nel gestire tentativi di DOS e discriminare dei veri e propri scan da attività più innocue.
- PortSentry - anch'esso progetto della Psionic di cui non esiste più un Home ufficiale, prevede meccanismi di detection anche di stealth scan, con la possibilità di bloccare tutti gli accessi dagli indirizzi che eseguono scan o azioni ostili.

In genere un semplice port scan detector ha funzioni limitate e può essere sostituito con migliore efficacia da un NIDS in grado di individuare, oltre a normali port scan una varietà di attività di rete sospette.

Per la realizzazione di un NIDS, il mondo OpenSource offre una discreta varietà di soluzione che, però, generalmente difettano nelle interfacce di reporting e gestione, oltre che nella facilità di installazione, per le

quali molte alternative commerciali offrono soluzioni più evolute:

- Snort – È il progetto NIDS più noto nella comunità OpenSource. Seppur di non banale gestione e con un sistema di reporting piuttosto grezzo, viene utilizzato come base da molti altri prodotti che ne estendono le funzionalità migliorando la gestione e i meccanismi di reporting. Le policy di packet matching sono costantemente aggiornate sulla base di nuove vulnerabilità.
- Tamandua – È un progetto relativamente poco conosciuto ma interessante, è possibile convertire le regole scritte per Snort sul database di Tamandua e sono previste tutte le features tipiche di un NIDS.
- Prelude - È un interessante ibrido fra un NIDS e un HIDS, che presenta sia sensori per il traffico di rete che sensori per il singolo host. Vanta prestazioni superiori a SNORT e una buona modularità.

5 Conclusioni

Nel presente rapporto tecnico sono state analizzate le principali problematiche di sicurezza legate a prodotti e infrastrutture software sviluppate in ambiente marittimo ed è stata proposta una piattaforma di comunicazione sicura tra sistemi eterogenei atta a gestire le criticità individaute.

In particolare, vista la crescente informatizzazione delle navi, e data la natura critica dei dati acquisiti e gestiti dai sistemi informatici che, in caso di hackeraggio possono altamente influenzare le attività in corso alterandone la ricaduta economica, ambientale e/o di sicurezza del personale di bordo, sono state studiate, in prima istanza, le principali problematiche da affrontare relative alla gestione in sicurezza dei dati e delle informazioni sia in termini di accesso e protezione dei dati, che in termini di integrità e disponibilità degli stessi. A valle dello studio effettuato, è stato proposto un framework per la gestione della sicurezza dei dati informatici che, da un lato, identifica gli obiettivi di sicurezza fondamentali che una piattaforma di comunicazione tra sistemi eterogenei dovrebbe perseguire, e di conseguenza i vari sotto-sistemi che la compongono. D'altro lato, il framework proposto specifica, per ogni obiettivo di sicurezza identificato, le tecniche e le tecnologie più adatte alla realizzazione dello stesso.

In particolare, al fine di gestire opportunamente la sicurezza dei dati, sono stati identificati i seguenti obiettivi fondamentali:

- identificazione di attori e/o eventi che agiscono o coinvolgono i dati;
- protezione dei dati;
- rilevamento delle intrusioni.

Il primo obiettivo fa riferimento alla necessità di identificare e autorizzare tutti gli attori e/o gli eventi che richiedono di accedere o modificare i dati del sistema, in modo da poter discriminare tra azioni permesse e non lecite. A tale scopo, nel dettaglio, il framework definito propone soluzioni per la gestione di:

- autenticità e autenticazione;
- autorizzazione;
- audit.

Il secondo obiettivo fa riferimento alla necessità di proteggere i dati in modo da renderli disponibili quando vengono richiesti, mantenendoli integri, autentici e non ripudiabili. A tale scopo, nel dettaglio, si propone di perseguire tale obiettivo tramite:

- confidenzialità;
- integrità;
- proprietà di non ripudio.

Infine, il terzo obiettivo fa riferimento alla necessità di prevedere meccanismi in grado di rilevare eventuali intrusioni e attacchi al sistema, che possano in qualche modo danneggiare o alterare la sicurezza dei dati. A tale scopo, nel dettaglio, sono state descritte le soluzioni maggiormente utilizzate nel campo dell'intrusion detection.

Riferimenti bibliografici

[1] AIS Ranjeet Vidwans, Michael Wessler, OCP & CISSP - IDaaS for Dummies - John Wiley & Sons, Inc., 2013.

[2] www.csoonline.com/article/2120384/identity-management/the-abcsof-identity-management.html

[3] www.oasisopen.org/committees/download.php/2713/Brief_Introduction_to_XACML.html

[4] Marcos A. P. Leandro, Tiago J. Nascimento, Daniel R. dos Santos, Carla M. Westphall, Carlos B. Westphall - Multi-Tenancy Authorization System with Federated Identity for Cloud-Based Environments Using Shibboleth - ICN 2012: The Eleventh International Conference on Networks

[5] https://shibboleth.net/about/

[6] https://www.oasis-open.org/committees/download.php/21265/drafthodges-HowToLearnSAML-01.html

[7] OASIS, 4 Agosto 2009, "SAML V2.0 Metadata Extension for Entity Attributes Version 1.0": http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-attr.pdf

[8] OASIS, 15 Marzo 2005, "Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0": http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf

[9] http://www.informationweek.com/software/informationmanagement/saml-the-secret-to-centralized-identity-management/d/did/1028656?

[10] http://www.slideshare.net/NRAJRAO/saml-in-cloud, 06/09/2014

[11] https://developer.salesforce.com/page/Single_SignOn_with_SAML_on_Force.com

[12] Brancik, Kenneth C. (2007). “Chapter 2: Related Research in Insider Computer Fraud and Information Security Controls”. Insider computer fraud: an in-depth framework for detecting and defending against insider IT attacks. CRC Press. pp. 18–19. ISBN 1-4200-4659-4