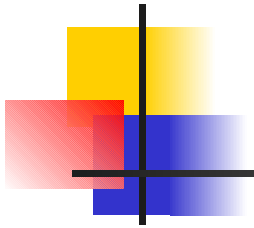
Un modello per il ranking delle vulnerabilità

F.Baiardi, S.Suin, C.Telmon
Dip. di Informatica,
Università di Pisa
M. Pioli, Enel Distribuzione



Problema

determinare l'ordine ottimale di eliminazione di vulnerabilità, dato un budget predefinito

Valutazione del rischio

- Associazione vulnerabilità - impatto
  - spesso molto informale
- Rischio = impatto * probabilità
  - ma quasi sempre mancano i dati per una valutazione oggettiva della probabilità
  - la probabilità diventa un “numero” messo lì dall’analista
  - Analisi poco ripetibile …

Ranking delle vulnerabilità

- Problemi comuni:
  - relativi a vulnerabilità comuni
  - non tengono conto della collocazione della vulnerabilità nel sistema
- Adatte a sistemi “generici” per una gestione ordinaria delle patch

La proposta

- Un modello per la descrizione del sistema funzionale all'associazione vulnerabilità/impatti
- Deduzione automatica di queste associazioni
- Utilizzo dei dati realmente disponibili su sistema e minacce
- Ranking che tenga conto della disponibilità di budget

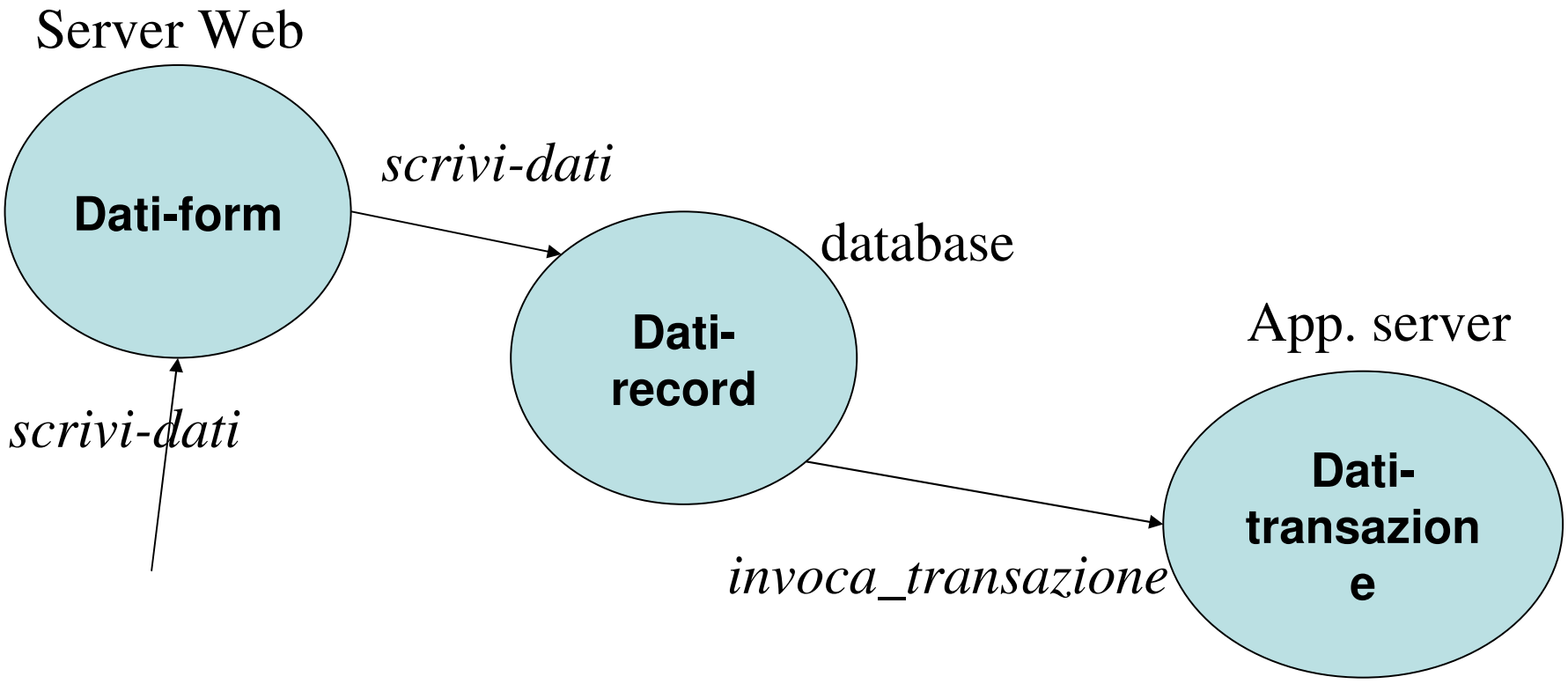
Informazioni utilizzate

- Minacce
  - chi ha interesse ad attaccare il sistema
  - che risorse ha a disposizione
- Vulnerabilità
  - per ognuna si suppone nota almeno una contromisura e il relativo costo

Oggetti e metodi

- Il sistema viene rappresentato come un insieme di oggetti, con diversi attributi, e di metodi per operare sugli attributi
  - si tratta di una nozione di metodo semplificata rispetto alla programmazione a oggetti
- Ogni utente ha dei diritti sui metodi, e attraverso gli attributi acquisisce diritti su altri metodi

Esempio


Server Web
Dati-form
scrivi-dati
database
Dati-record
App. server
Dati-transazione
scrivi-dati
invoca_transazione

Propagazione dei diritti

- La rappresentazione del sistema è un'astrazione che ha lo scopo di evidenziare la propagazione dei diritti
- Non è una rappresentazione dell'architettura reale
  - Ad esempio, il database non "invoca" nessun metodo dell'AS, ma è l'AS che accede al database per ottenere i dati necessari per la transazione
  - È però vero che se controllo i dati del database controllo il risultato della transazione

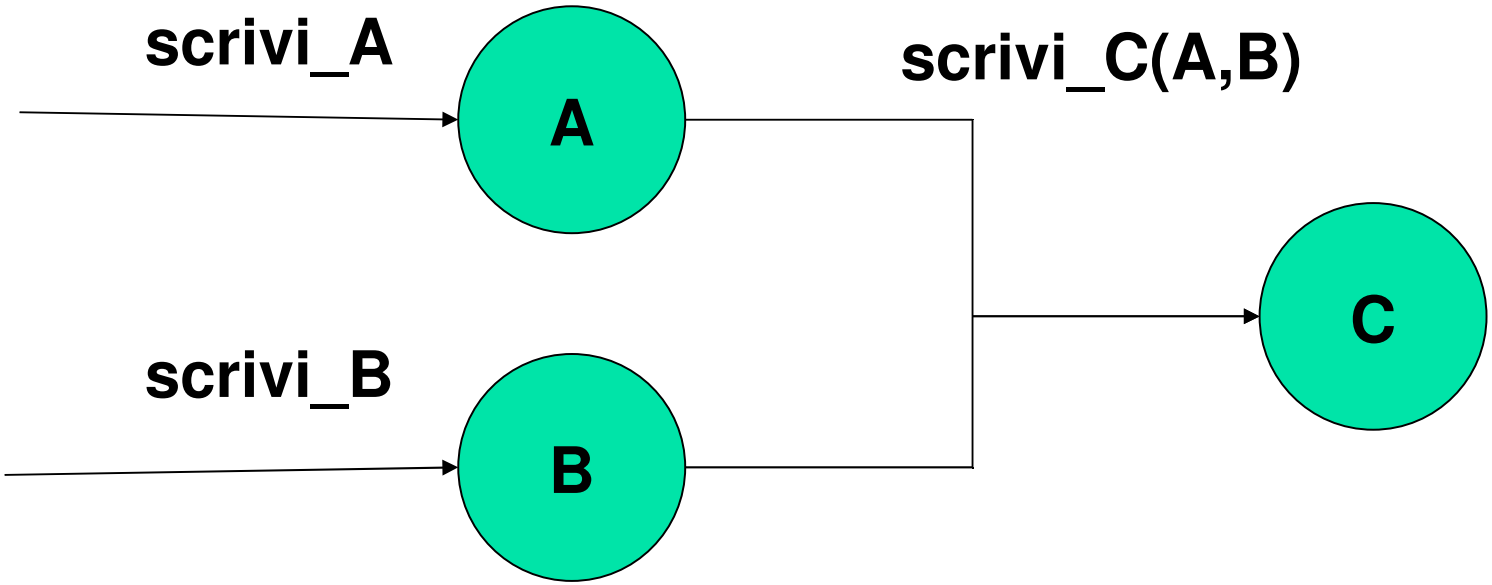
Livelli di astrazione

- I diversi componenti possono essere rappresentati a diversi livelli di astrazione
- Lo stesso componente può essere rappresentato più volte a diversi livelli di astrazione
  - attenzione alle interazioni!
- La competenza dell'analista e la sua conoscenza del sistema si concretizzano nella capacità di dare una rappresentazione efficace del sistema

Dipendenze

- La rappresentazione del sistema serve per definire un grafo delle dipendenze fra metodi
- Un metodo M dipende da un insieme I di altri metodi se chi ha diritto di invocare i metodi di I è in grado di controllare il risultato di M

Dipendenze: esempi



Il metodo scrivi_C prende come parametri A e B; il metodo C dipende quindi dai metodi scrivi_A e scrivi_B: chi ha il diritto di invocare scrivi_A e scrivi_B “ha diritto di invocare” scrivi_C

Chiusura transitiva

- Per la dipendenza fra metodi vale la proprietà transitiva:
  - se A dipende da B e B dipende da C, allora A dipende da C
- Dato un insieme iniziale di diritti, mediante il grafo delle dipendenze se ne può calcolare la chiusura transitiva:
  - tutti i metodi che dipendono da quell'insieme iniziale

Utenti, diritti e chiusura transitiva

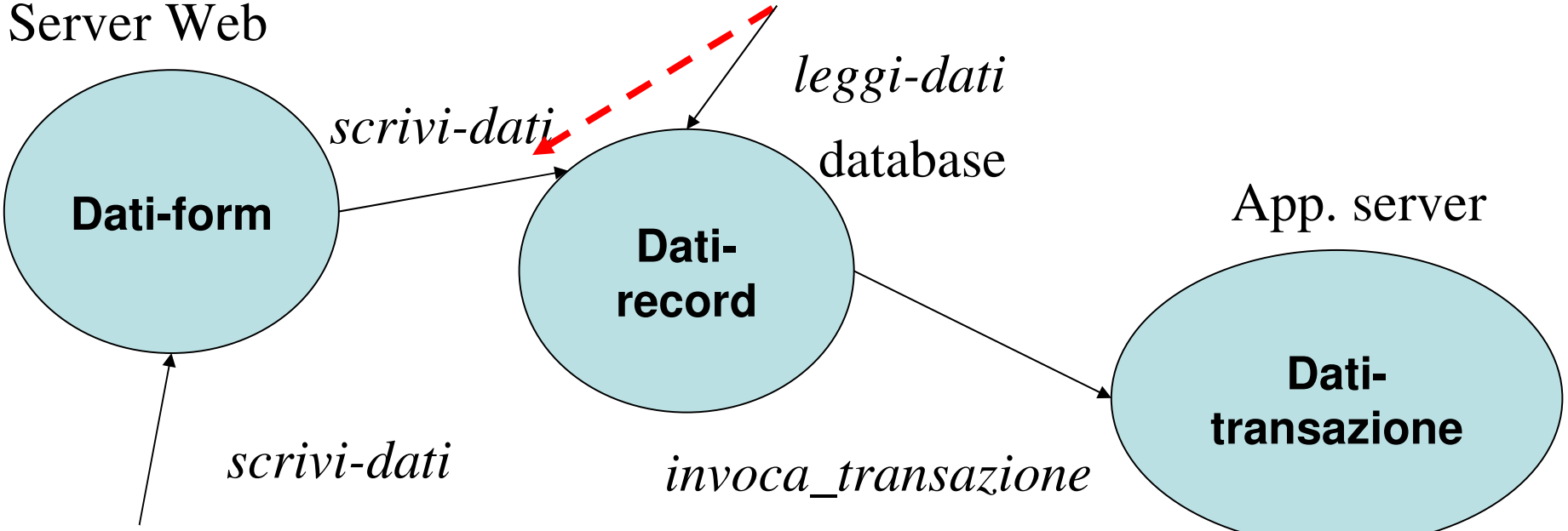
- Ogni utente ha un insieme iniziale di diritti
  - sono i diritti sui metodi che lo “interfacciano” al sistema
- La chiusura transitiva di questo insieme sono tutti i metodi che l’utente controlla nel sistema (il controllo non è esclusivo dell’utente!)
- Rappresentano le attività del sistema che l’utente è in grado di influenzare/controllare

Vulnerabilità e attacchi

- In questo contesto una vulnerabilità non è altro che un difetto del sistema che permette di praticare uno o più attacchi
- Un attacco, dati:
  - una vulnerabilità
  - un insieme di risorse (di calcolo, competenze, ecc.)
  - un insieme di diritti su metodi
- permette di acquisire diritti su uno o più metodi

Attacchi: esempio

Una vulnerabilità del DBMS permette un attacco: dato il diritto di invocare *leggi_dati* e la competenza per scrivere codice, permette di acquisire il diritto di invocare *scrivi_dati*



Le minacce

- Le minacce sono degli utenti che hanno:
  - un insieme iniziale di diritti (vuoto?)
  - delle risorse per praticare attacchi
  - degli obiettivi: diritti che vogliono acquisire su metodi; agli obiettivi sono associati degli impatti
- In questo modello, le minacce non agiscono a caso

Evoluzioni

- Una minaccia ha un insieme di diritti
  - Iniziali più quelli calcolati mediante chiusura transitiva
- La minaccia esegue un attacco
  - permesso dai diritti che ha e dalle risorse
  - acquisisce nuovi diritti; si calcola una nuova chiusura transitiva
- I nuovi diritti permettono un altro attacco
  - nuova chiusura transitiva…

Evoluzioni (2)

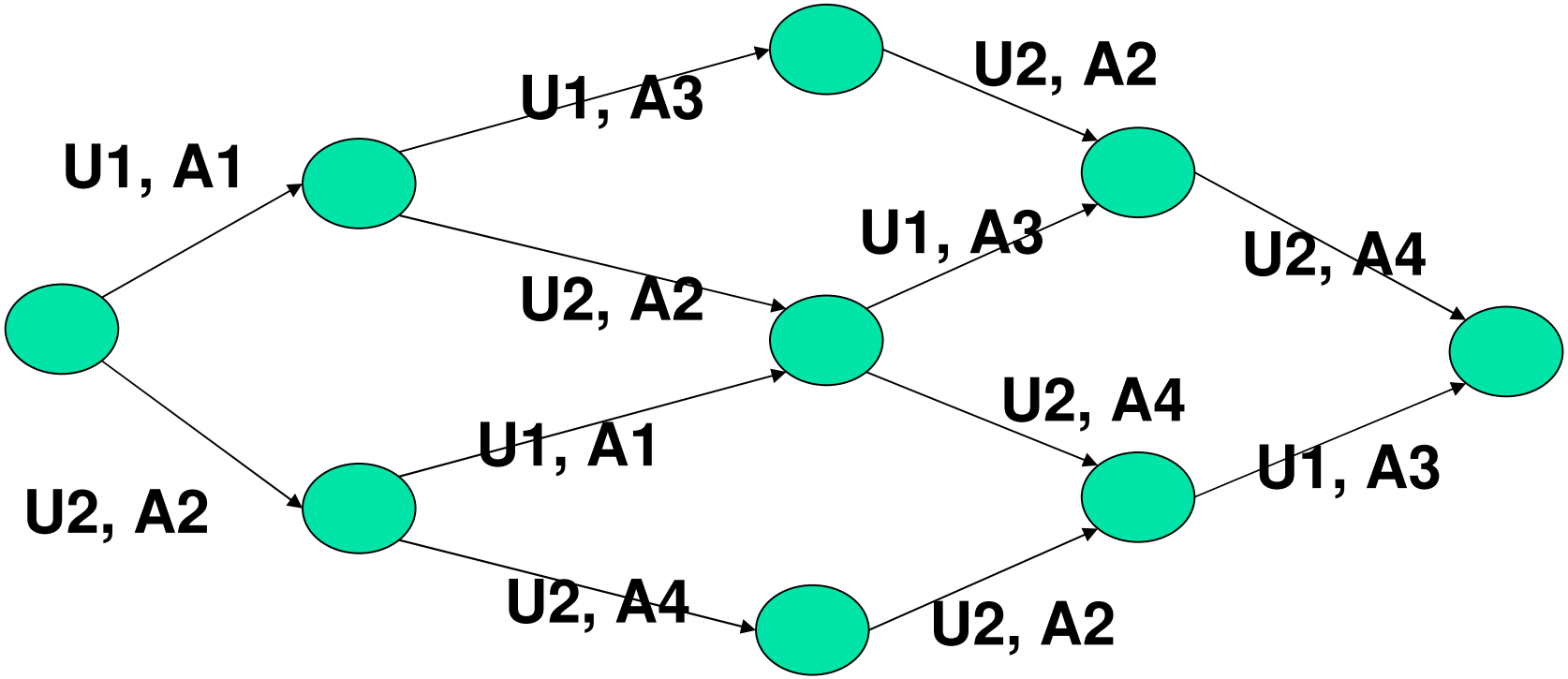
- Un'evoluzione rappresenta una sequenza di stati in cui le minacce, praticando attacchi, acquisiscono nuovi diritti
  - uno stato è definito dall'insieme di diritti di diversi utenti e minacce
- Il grafo delle evoluzioni permette di rappresentare tutti i diritti che le minacce riescono ad acquisire mediante attacchi
  - compresi gli obiettivi

Il grafo delle evoluzioni

- L'insieme delle possibili evoluzioni di un sistema può essere rappresentato mediante un grafo in cui:
  - un nodo è uno stato del sistema;
  - un arco (orientato) rappresenta un attacco praticato da una minaccia
    - con i diritti disponibili nel nodo di partenza
    - acquisendo (anche per chiusura transitiva) i diritti disponibili nel nodo di arrivo

Esempio di grafo delle evoluzioni

Questo esempio banale mostra la crescita esponenziale degli stati

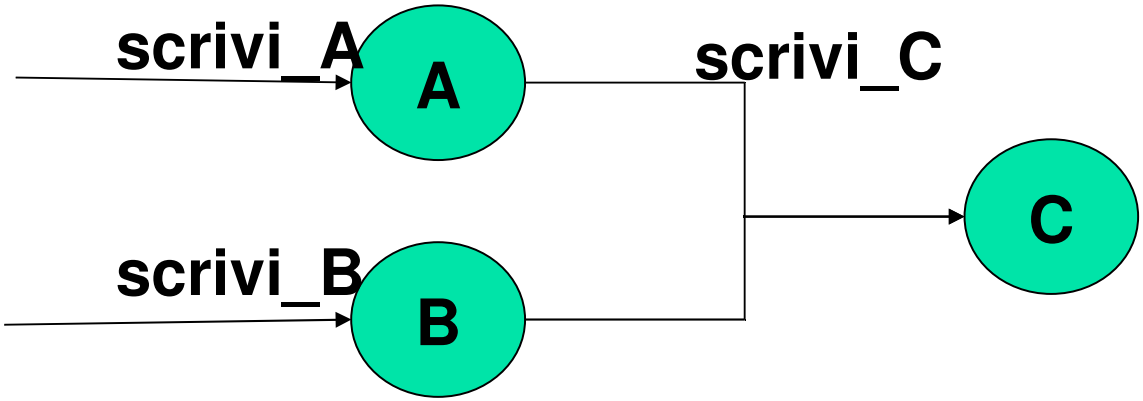


Il grafo delle evoluzioni (2)

- Alcuni nodi rappresentano obiettivi
- Nell'ipotesi che i diritti non vengano persi, il grafo è aciclico
- Il nodo iniziale rappresenta lo stato iniziale, prima dell'esecuzione di attacchi
- Ogni cammino del grafo rappresenta una possibile evoluzione

Dipendenze come deduzioni

- Le dipendenze fra metodi possono essere rappresentate come clausole di un programma logico:
  - scrivi_C(U) :- scrivi_A(U), scrivi_B(U)



Attacchi come deduzioni

- Anche gli attacchi possono essere rappresentati come clausole di un programma logico:
  - se una chiave k è debole, il dato in chiaro è noto e il messaggio può essere letto, allora è possibile conoscere la chiave
  - l(u,k):-weak(k), l(u,d), l(u,m), c(m,d,k)

Vulnerabilità e diritti iniziali come assiomi

- Dati alcuni fatti:
  - vulnerabilità: weak(K1)
  - diritti iniziali: l(U1,D1) , l(U1,M1)
- Altri fatti descrivono l'attacco specifico nel sistema:
  - c(K1,D1,M1)
- Allora la clausola l(u,k):-weak(k), l(u,d), l(u,m), c(m,d,k) permette di dedurre nuovi diritti:
  - l(U1,K1)

Vulnerabilità e deduzioni

- Dalle clausole che descrivono le dipendenze e dai diritti iniziali è possibile calcolare la chiusura transitiva dei diritti
- Aggiungendo le clausole che descrivono gli attacchi, è possibile calcolare quali (ulteriori) diritti possono essere acquisiti dalle minacce

- Partendo da una descrizione iniziale del sistema corretta, questo garantisce di calcolare tutti gli obiettivi che una minaccia può raggiungere
  - il rapporto fra vulnerabilità e impatti non è arbitrario

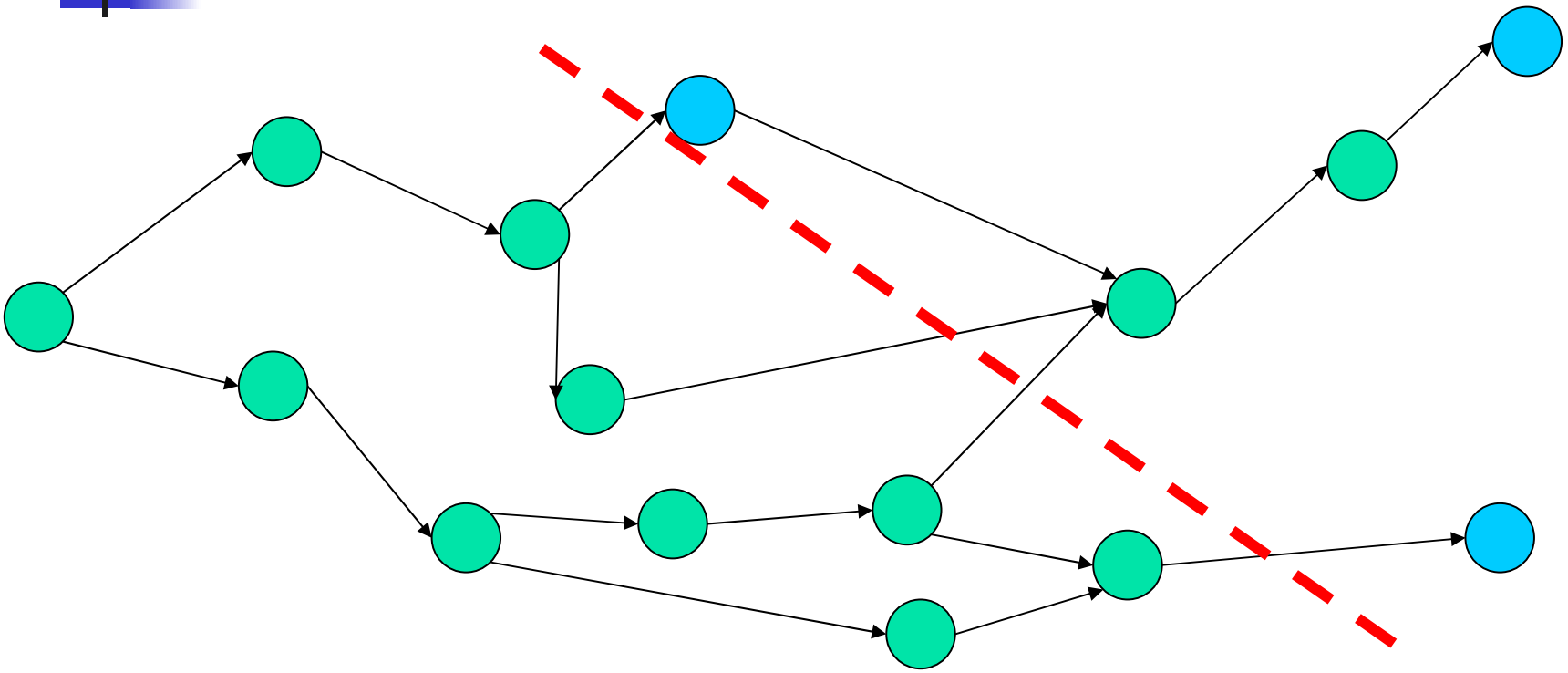
Rappresentazione del sistema

- La facilità nel trarre deduzioni è data dall'uniformità nella rappresentazione del sistema, indipendentemente dal livello di astrazione:
  - ci si concentra sui diritti (o sul controllo)
  - non ci si preoccupa del "come", né per i metodi né per le vulnerabilità
- Questo richiede uno sforzo per un'effettiva astrazione nella rappresentazione del sistema
- La gestione della crescita esponenziale degli stati del grafo delle dipendenze è gestita dal motore inferenziale

Vulnerabilità e tagli

- Ora che abbiamo un modo per associare vulnerabilità e obiettivi, dobbiamo capire quali vulnerabilità eliminare per impedire alle minacce di raggiungere i loro obiettivi

Tagli del grafo delle evoluzioni



Ci interessano i tagli che impediscono alle minacce di raggiungere i loro obiettivi (nodi azzurri)

Tagli e insiemi minimi di vulnerabilità

- Ogni taglio corrisponde a un insieme di attacchi, che corrisponde a un insieme di vulnerabilità
- Un insieme minimo di vulnerabilità:
  - corrisponde a un insieme di attacchi che costituiscono un taglio del grafo
  - è tale nessun suo sottoinsieme abbia la stessa caratteristica

Tagli e insiemi minimi di vulnerabilità (2)

- Eliminare le vulnerabilità di un insieme minimo vuole dire impedire alle minacce di raggiungere il loro obiettivi
- È conveniente eliminare altre vulnerabilità solo per ridondanza, o in previsione di nuove vulnerabilità future
  - tutte le vulnerabilità, indipendentemente dagli impatti?

Altre informazioni “utili”

- Dai grafi delle dipendenze e delle evoluzioni è possibile trarre altre informazioni sul sistema
- Esempio: grafo delle dipendenze
  - un metodo che dipende da molti metodi è più “robusto”, perché la minaccia avrà bisogno di più diritti (più attacchi) per controllarlo; più sono questi metodi, più il sistema è robusto nel complesso

Ranking delle vulnerabilità

- Ogni vulnerabilità ha una contromisura
- Ogni contromisura ha un costo
- L'eliminazione di un insieme minimo di vulnerabilità ha un costo
- Consideriamo solo gli insiemi minimi di costo minimo (in un intervallo)
- Vorremo implementare le contromisure di uno di questi
  - Quale? In che ordine?

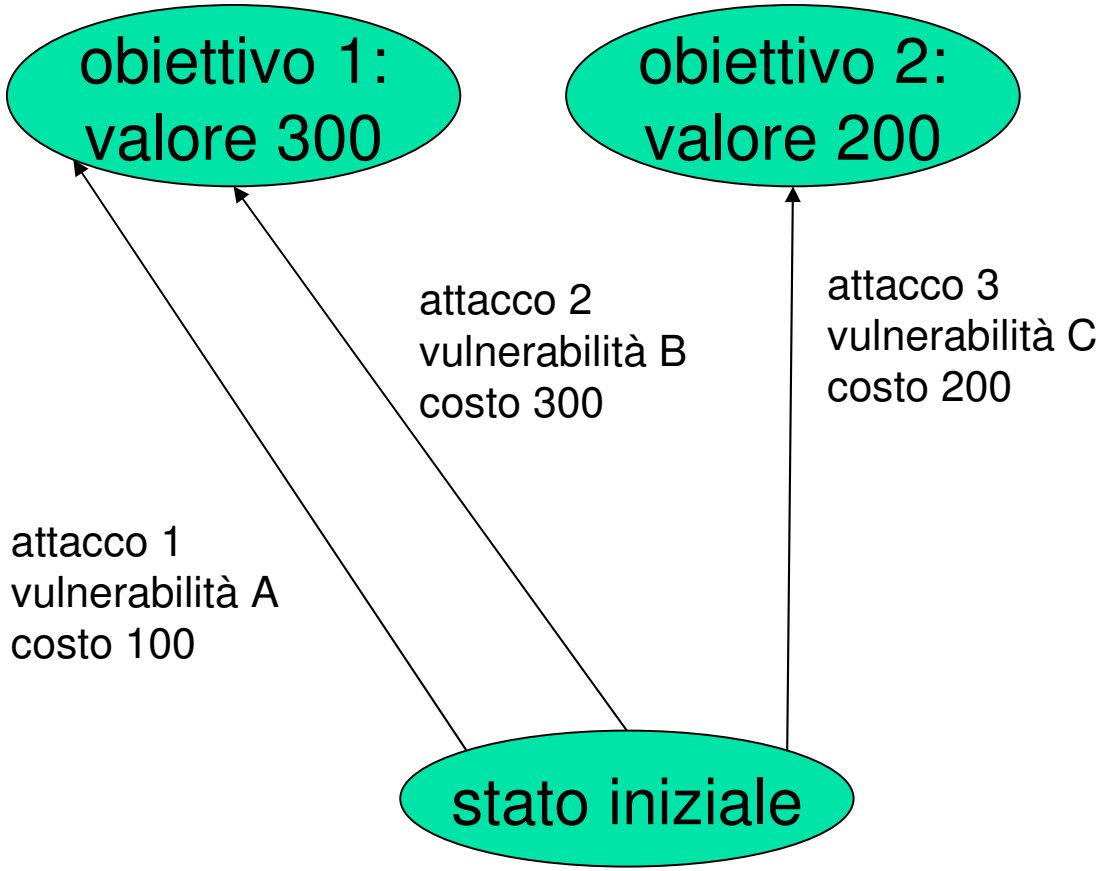
Ranking delle vulnerabilità (2)

- Ipotesi realistica: non abbiamo il budget per eliminare (subito) tutte le vulnerabilità di un insieme minimo
  - abbiamo un certo budget ogni anno
- Vogliamo eliminare prima le vulnerabilità associate agli impatti maggiori

Ordine di rimozione delle vulnerabilità

- Per ogni insieme minimo:
  - calcoliamo il suo insieme potenza
    - l'insieme di tutti i suoi sottoinsiemi
  - ne ordiniamo per inclusione gli elementi
  - eliminiamo gli elementi tali che un suo sottoinsieme impedisca di raggiungere gli stessi obiettivi
- Otteniamo tutte le sequenze di eliminazione delle vulnerabilità in quell'insieme minimo

Ordine di rimozione delle vulnerabilità - 1



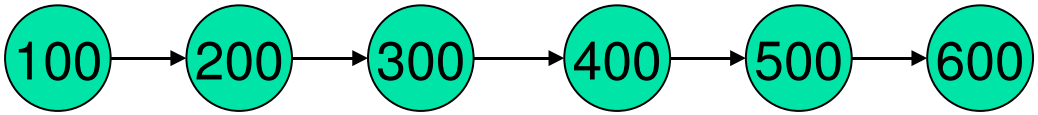
Insieme minimo:
{A,B,C}
Insieme potenza:
Ø,
{A},{B},{C}
{AB},{AC},{BC}
{A,B,C}
Eliminando gli insiemi non significativi:
Ø,{C},{AB},{ABC}

Ordine di rimozione delle vulnerabilità - 2

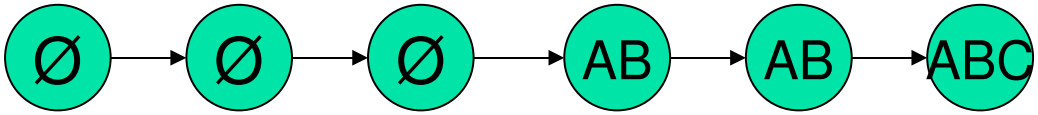
- Ordinamenti parziali
  - Ø,{AB},{ABC}
  - Ø,{C},{ABC}
- Ad esempio, non interessa l'insieme {AC}, perché non elimina più vulnerabilità di {C} (ma costa di più): eliminerò A solo insieme a B

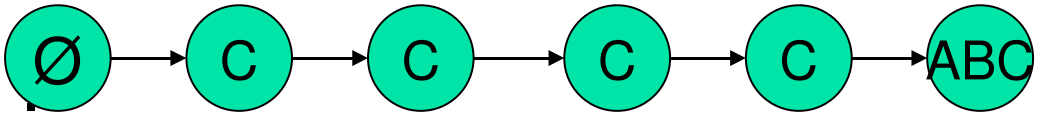
Ordine di rimozione delle vulnerabilità - 3

- Suppongo di avere a disposizione risorse per 100 ogni mese per sei mesi



- Le vulnerabilità dei due ordinamenti possono essere eliminate come segue:



- 

Ordine di rimozione delle vulnerabilità - 4

- Nel primo caso, abbiamo un impatto di 500 per 3 mesi e un impatto di 200 per i successivi 2 mesi, totale: 1900
- Nel secondo caso, abbiamo un impatto di 500 per un mese e un impatto di 300 per 4 mesi, totale: 1700
- Conviene il secondo ordine di rimozione
- Lo stesso meccanismo si applica a tutti gli insiemi minimi, fino a ottenere la sequenza possibile più conveniente
- Anche questi passaggi possono essere automatizzati

E il rischio?

- Nell'ipotesi che la minaccia abbia:
  - obiettivo
  - vulnerabilità
  - risorse
  - diritti iniziali
- cosa le impedisce di praticare con certezza l'attacco?
- La risposta è data dai dati storici

Integrazione dello storico

- In assenza di storico, parlare di probabilità è spesso arbitrario
- In presenza di dati significativi (ad es. sulla difficoltà per la minaccia di scoprire la vulnerabilità) si può ridurre la probabilità, in pratica “pesando” gli attacchi o l’impatto

Vantaggi della metodologia

- Ci si concentra su una rappresentazione efficace del sistema, il resto è deducibile in modo automatico
  - ripetibile e verificabile
- Riusa il lavoro già fatto in caso di nuove vulnerabilità
- È possibile “mescolare” livelli di astrazione
- Molti sviluppi futuri: ad esempio, agire sulla chiusura transitiva modificando il sistema

Cosa manca?

- Descrizione del sistema: alcune possibili caratteristiche del sistema non sono ancora trattate
  - es. contromisure dinamiche
- Programmazione logica:
  - lo strumento in fase iniziale di sviluppo
- E naturalmente, stiamo testando…

Bibliografia

- Per una trattazione più formale:

  Assessing the Risk of an Information Infrastructure through Security Dependencies, Critis 2006

- Grafi degli attacchi
  - Introduzione: Attack Modeling for Information Security and Survivability
  - Survey: An Annotated Review of Past Papers on Attack Graphs