

LinSTAT: Un Sistema di Rilevamento degli Attacchi Informatici

Tesi di laurea
Ingegneria Informatica
A.A. 2002-2003

Marco Cova
marco.cova@studio.unibo.it

Indice

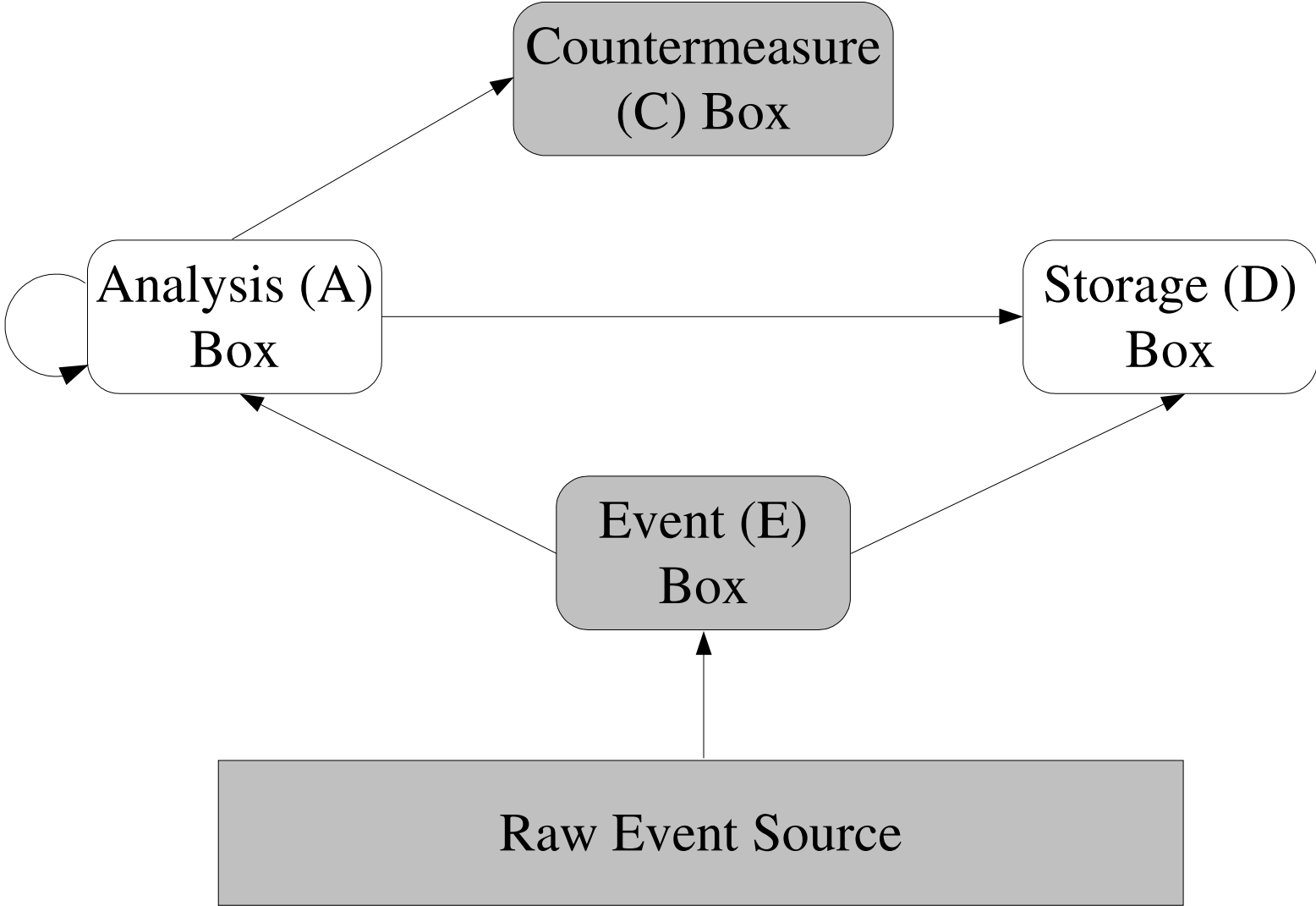
- 1) Introduzione a sistemi di rilevamento delle intrusioni
- 2) LinSTAT

Intrusion Detection System (IDS)

Un sistema che permette di:

- Monitorare le attività di un sistema informatico
- Rilevare attacchi diretti contro di esso
- Mettere in atto azioni correttive

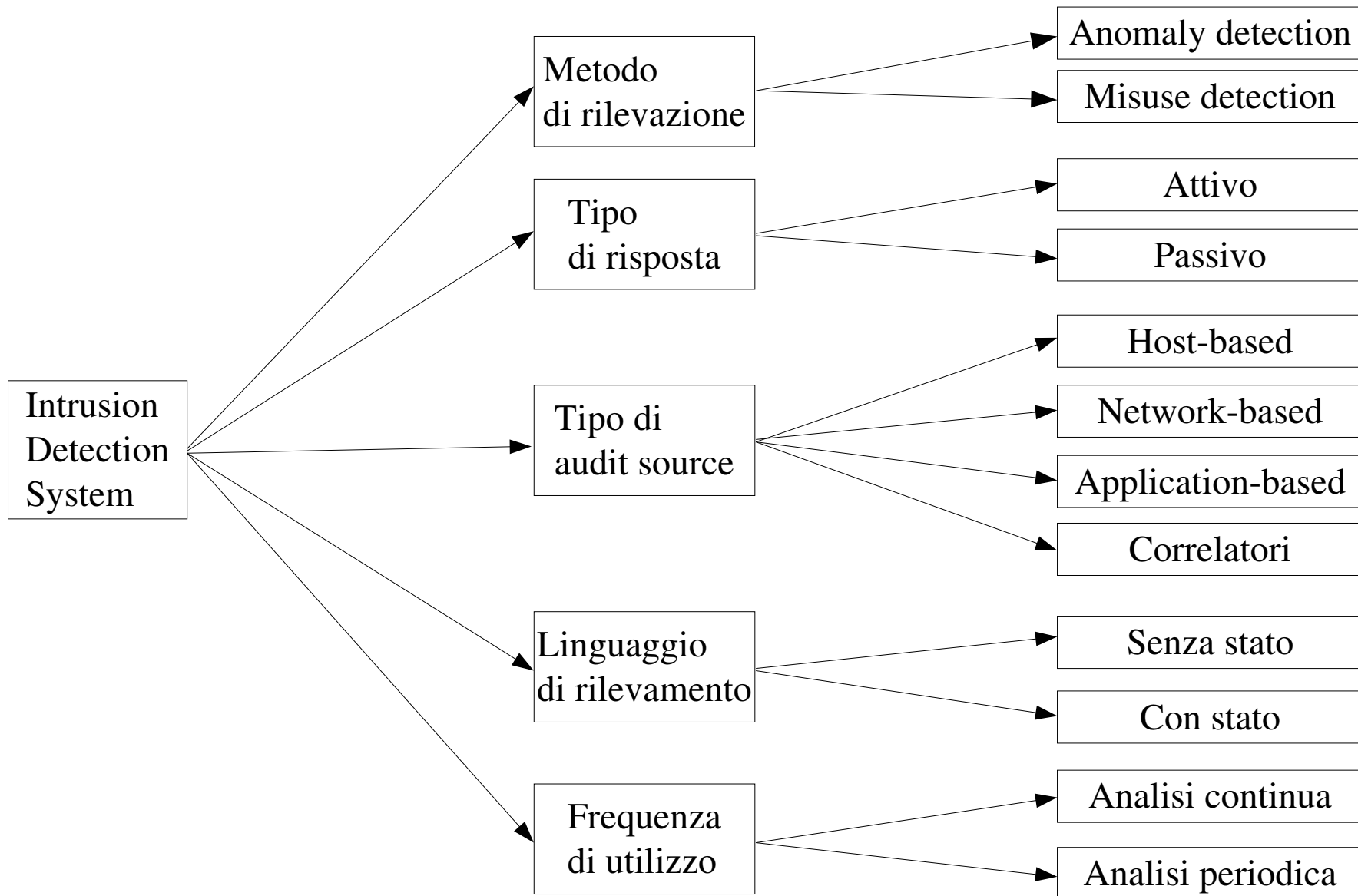
Architettura



Metriche di efficacia

- **Completezza:** rilevamento di *tutti* gli attacchi (numero di falsi negativi)
- **Accuratezza:** rilevamento corretto (numero di falsi positivi)
- **Performance di processing:** numero di eventi al secondo
- **Responsività**
- **Tolleranza a guasti e attacchi**

Tassonomia



Metodo di rilevazione

Anomaly detection

- Definisce cosa è “normale” (ipotesi: attività anomale parte di attacco)
- Pro: rileva attacchi nuovi, poche regole
- Contro: training, spostamenti da “norma”, ambiente dinamico?

Misuse detection

- Definisce cosa è “male” attraverso modelli di attacchi (*signature*)
- Pro: accurato
- Contro: completezza?

Tipo di risposta

Attivo

- Abbattere connessioni di rete
- Riconfigurare il firewall
- Modificare configurazione dei servizi

Problemi?

Passivo

- Allarme (log file, mail all'amministratore, SMS, ecc.)

Tipo di sorgente di audit

Host-based

- Informazioni di sistema
(*vmstat, top, netstat*)
- Syslog
- Dati a livello di kernel

Correlatori

- Alert generati da altri IDS

Network-based

- Traffico di rete

Application-based

- Modifica dei sorgenti
- Librerie di interposizione
- Hook di estensione

Linguaggio di rilevamento

Senza stato

- Pattern matching
- Macchina combinatoria

Con stato

- Modellazione del sistema in stati (sicuri e insicuri) e transizioni tra stati in risposta ad eventi
- Macchina con stato

Frequenza di utilizzo

Analisi continua

- Pro: protezione real-time
- Contro: impatto sulle performance del sistema monitorato

Analisi periodica

- Pro: scarso impatto sulle performance del sistema
- Contro: solo analisi *post-mortem*

Problemi aperti

- **Efficacia:** diminuire i falsi positivi e rilevare attacchi sconosciuti
- **Audit throughput:** incremento costante della quantità di dati da analizzare (reti ad elevatissima velocità)
- **Cooperazione tra IDS:** usare IDS diversi per complementarne punti di forza

STAT

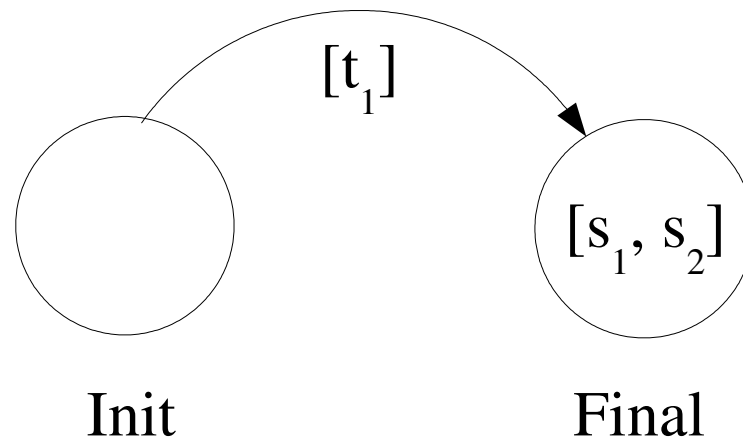
- Sviluppato presso l'Università della California a Santa Barbara (UCSB) a partire dagli anni '90
- *State Transition Analysis Technique*: modella attacchi come azioni che causano transizioni nello spazio degli stati di sicurezza di un sistema
- IDS basati su STAT estendono *STAT Framework*: una serie di componenti progettati in modo da facilitare lo sviluppo di IDS in diversi domini e piattaforme

STAT Framework

Fornisce i seguenti concetti e componenti:

- **Tecnica STAT:** metodo di modellazione degli attacchi
- **STATL:** linguaggio di descrizione degli attacchi
- **STAT Core:** runtime di STATL
- **MetaSTAT:** infrastruttura per la comunicazione tra sensori basati su STAT e la gestione remote

Tecnica STAT



Caratteristiche salienti:

- Rappresentazione di alto livello di attacchi
- Modellazione di attacchi polimorfici
- Modellazione che permette l'interruzione di attacchi

STATL

- Linguaggio di descrizione degli attacchi usato per scrivere scenari di attacco
- Fornisce una serie di concetti indipendenti dal particolare dominio applicativo: stati, transizioni, timer
- Estensibile per esprimere elementi propri dello specifico dominio applicativo: eventi, tipi, predicati

STAT Core

- Runtime del linguaggio STATL di cui implementa le caratteristiche indipendenti dal dominio applicativo
- Svolge il ruolo di *A-box* facendo il matching tra lo stream di eventi (tipicamente fornito da *E-box*) e un insieme di scenari caricati
- Deve essere esteso per supportare le estensioni a STATL

MetaSTAT

Infrastruttura di comunicazione e gestione che supporta:

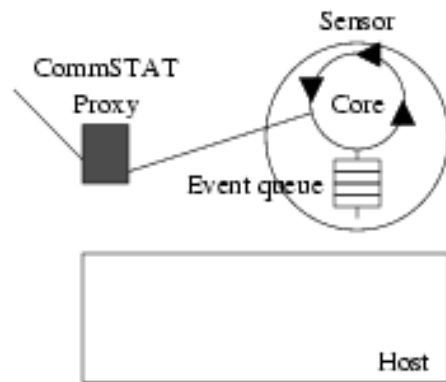
- Rilevamento distribuito di attacchi
- Configurazione dinamica dei vari componenti del sistema
- Comunicazione sicura (SSL) tra i componenti

Sensore STAT

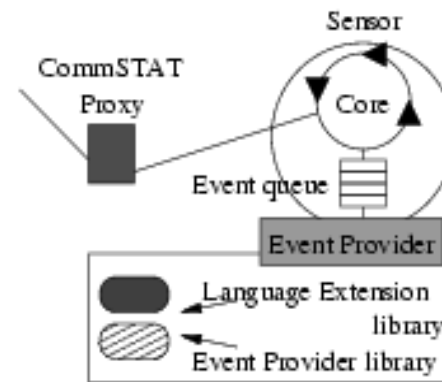
IDS basato su STAT Framework che estende con:

- **Event provider:** *E-box*
- **Scenari:** descrizioni di attacchi modellati secondo la tecnica STAT
- **Risposte:** contromisure da attivare in caso di rilevamento di attacchi

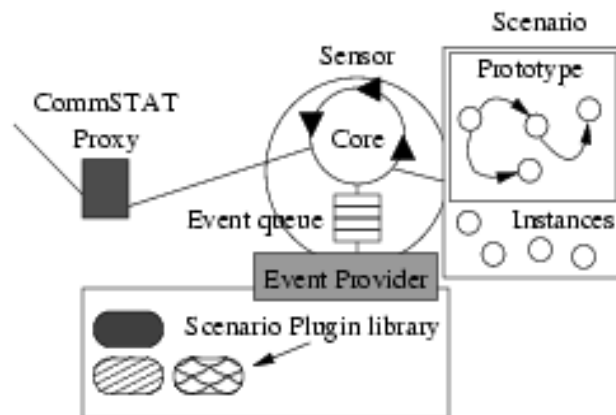
Configurazione di Sensore STAT



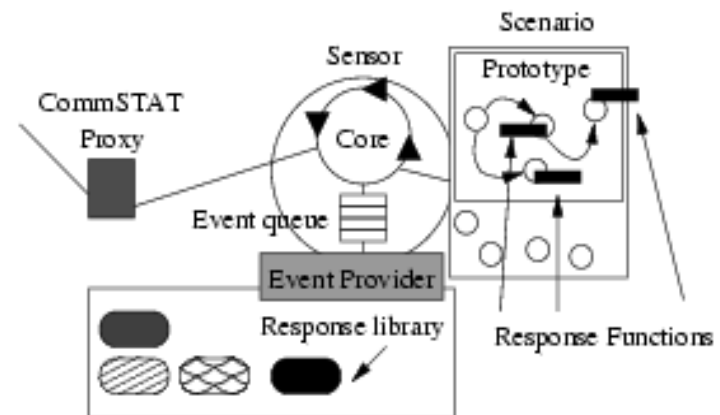
(a) Bare Sensor



(b) Sensor with Event Provider



(c) Sensor with Scenario Plugin

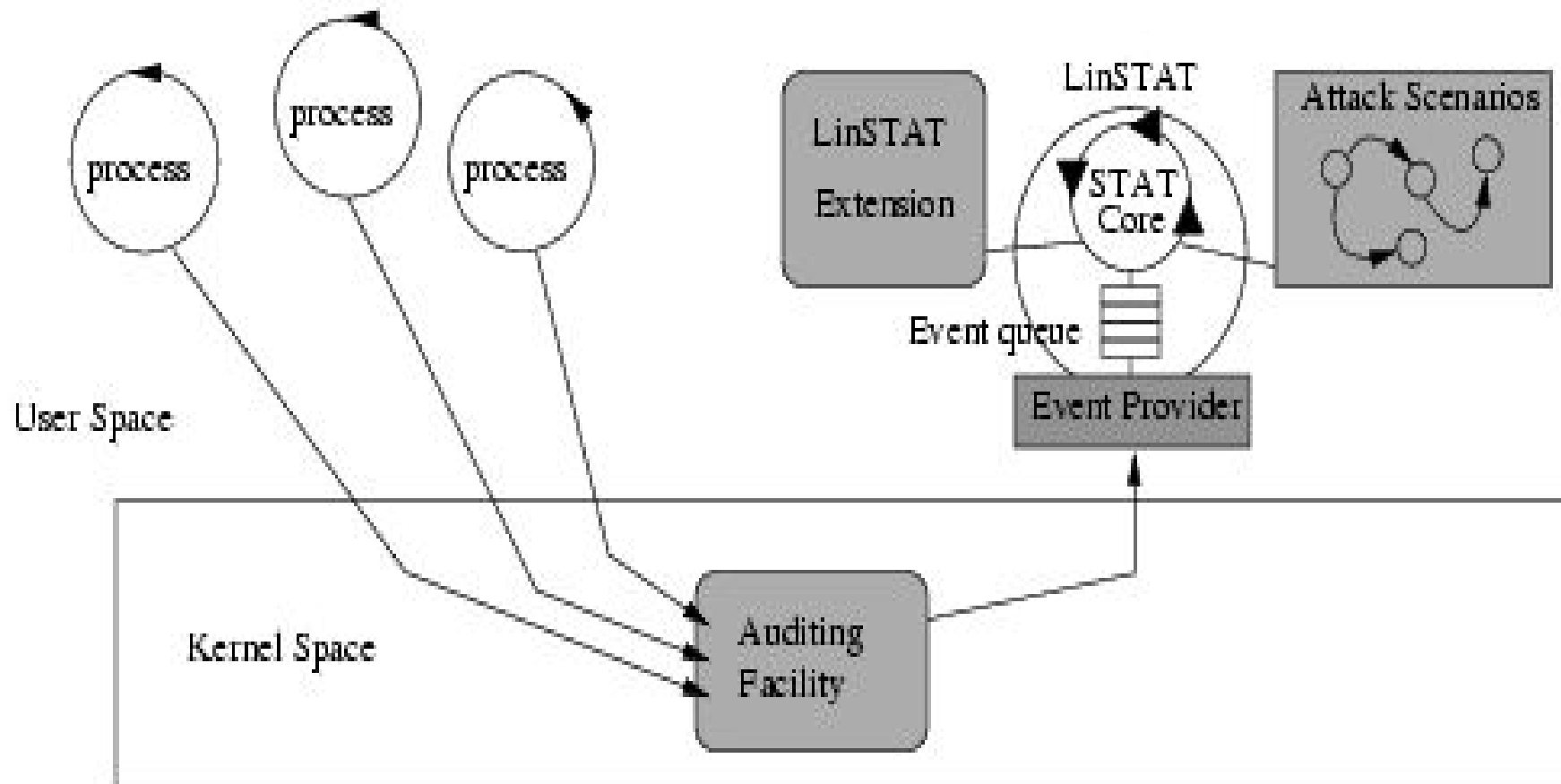


(d) Scenario Plugin with Responses

LinSTAT

- Sensore STAT (IDS basato sul framework STAT) per Linux
- Implementa le parti dipendenti dalla piattaforma del framework: eventi, tipi, predicati, scenari
- Fornisce anche il componente denominato “raw input source” nell'architettura generale, perché Linux non è dotato di una sua sorgente di dati di audit

LinSTAT – Architettura



Audit Source

- Modulo per il kernel di Linux che fa da wrapper alle system call e raccoglie informazioni di audit che rende disponibili in user space
- System call monitorate: file (open, creat, mkdir, unlink, mknod, rmdir, chown, chmod, symlink, link, rename, truncate), processi (execve, exit), privilegi (set*uid), rete (connect, accept), amministrazione (reboot, chroot, create_module)
- Informazioni ottenibili: processo (pid, invocazione da riga di comando), utente (uid, gid, euid, egid), timestamp, altro (dipendente dalla specifica system call)

Evento di esempio

```
act: EXECUTE, time: Thu Aug 14 18:10:37 2003, retcode: 0,  
exec_args: cat /proc/devices, pathname: /bin/bash, uid: 0,  
gid: 0, euid:0, egid: 0, pid: 13921, ppid: 13920,  
pwd: /home/mcova, objname: /bin/cat, owner: 0, gowner: 0,  
inode: 3817535, dev: 774, perm: rwxr-xr-x
```

L'utente con uid 0 in data 14 Agosto 2003 alle ore 18:10:27 ha lanciato da riga di comando il comando *cat /proc/devices*. La directory corrente al momento dell'esecuzione era /home/mcova e il programma effettivamente eseguito era contenuto nel file /bin/cat, di cui si conoscono proprietario (l'utente con uid 0), inode, device su cui è contenuto e maschera dei permessi

Esempio di scenario

```
initial state init { }
```

```
transition read (init->s2) consuming
```

```
{
```

```
  [READ r]: // evento READ...
```

```
    (r.retcode >= 0) && // ...che ha avuto successo
```

```
    member(r.objname, "RESTRICTED_READ_OBJECTS", stat) && // ... di un file ad accesso ristretto
```

```
    !in_prog_set(r.pathname, "RESTRICTED_READ_OBJECTS", stat) //... con un programma non autorizz.
```

```
{
```

```
  // info per l>alert
```

```
  SOURCE_USERNAME = userid2name(r.uid, stat);
```

```
  SOURCE_USERID = toString(r.uid);
```

```
  SOURCE_PROC_PATH = r.pathname;
```

```
  SOURCE_PROC_NAME = r.exec_args;
```

```
  TARGET_PROC_PATH = r.objname;
```

```
  IMPACT = "bad-unknown";
```

```
  auid = r.uid;
```

```
  pid = r.pid;
```

```
}
```

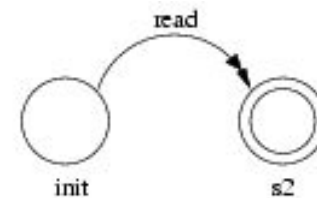
```
}
```

```
state s2
```

```
{ log("%d: %s read by user %d, program %s",
```

```
    auid, TARGET_PROC_PATH, auid, SOURCE_PROC_PATH);
```

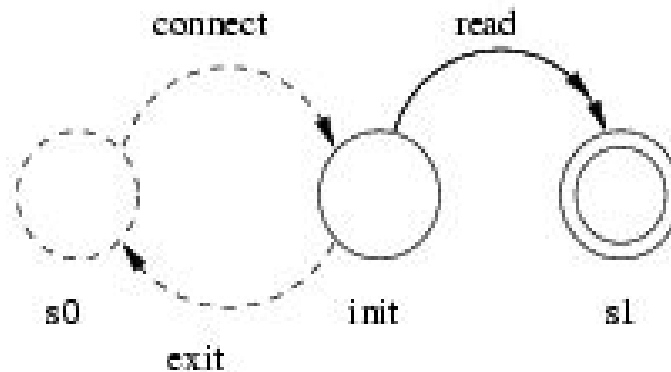
```
}
```



Esempio di alert

```
<IDMEF-Message version="0.3">  
  <Source spoofed="unknown">  
    <User>  
      <UserId type="current-user">  
        <name>mcova</name><number>502</number>  
      </UserId>  
      <UserId type="current-group"> <name>502</name> </UserId>  
    </User>  
    <Process>  
      <name>cat /etc/passwd</name> <path>/bin/cat</path> <pid>1579</pid>  
    </Process>  
  </Source>  
  <Target>  
    <Process><name>/etc/passwd</name> <path>/etc/passwd</path>  
  </Target>  
</IDMEF-Message>
```

Tainting



- Meccanismo che permette di “marcare” processi generati da utenti non locali
- Consente di scrivere scenari sofisticati: generare alert in corrispondenza di azioni lecite per utenti locali, ma illecite se dovute a processi di origine esterna

Testing

Impatto sulle prestazioni del sistema monitorato:

- Crescente al crescere del numero di scenari caricati (15 scenari < 6%)
- Maggiore per processi I/O bound

Scenari sviluppati:

- Policy-based
- Attacchi che modificano il flusso di controllo di programma
- Exploit specifici