

State Transition Analysis: A Rule-Based Intrusion Detection Approach

Koral Ilgun, Richard Kemmerer and
Philip A. Porras

Current approach to intrusion detection

- Threshold detection
- Anomaly detection
- Rule-based penetration Identification
- Model-based intrusion detection
- Intrusion Prevention

Threshold detection

- Goals:
 - Recording of each occurrence of a specific event
 - Detection of number of occurrence of events that surpass a prefixed amount
- Problems:
 - Identify the threshold number and window size for a specific event
- Weakness/strength:
 - Poor detector of intrusions
 - Always implemented as a sub component of larger IDS

Anomaly detection

- **Goals:**

- Establish usage patterns within user audit trails over a duration of time
- Use of the usage patterns as profile of normal system activity

- **Problems:**

- Fine tuning of the anomaly detection tools

- **Strength/Weakness:**

- Means of detecting intrusions provided without a priori knowledge of the security flaws in the target system

Rule-based penetration Identification

- Goals:
 - An expert system whose rules fire when audit records appear to indicate suspicious or illegal user activity
 - Single auditable events that represent a threat for the system or sequence of events that represent an entire penetration scenario are recognized
- Problems:
 - Only intrusions that can be anticipated prior to their occurrence can be detected
- Strength/Weakness:
 - Rule-based penetration identifiers are supplemental component of intrusion detection systems

Model Based Intrusion Detection

- Goals:
 - Specifications of scenarios models that represent the characteristic behavior of intrusions
- Problems:
 - Administrators should be able also to generate their representation of the penetration directly
- Strength/Weakness
 - Instructions are modeled at a higher level of abstraction than audit records

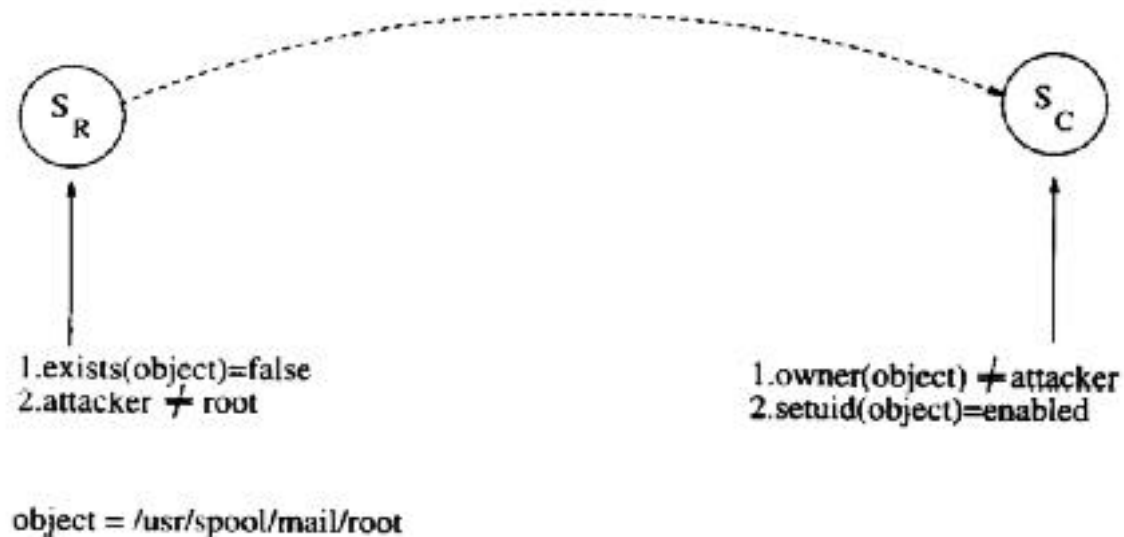
State Transition Analysis

- Premises:
 - Penetrations require the attacker to possess some minimum prerequisite access to the target system
 - All penetrations lead to the acquisition of some previously unheld ability
- How it works?
 - An initial state and compromised state are identified
 - Intermediate state transitions are also defined
 - Signature actions are identified
 - The information previously produced is represented graphically as a state transition diagram

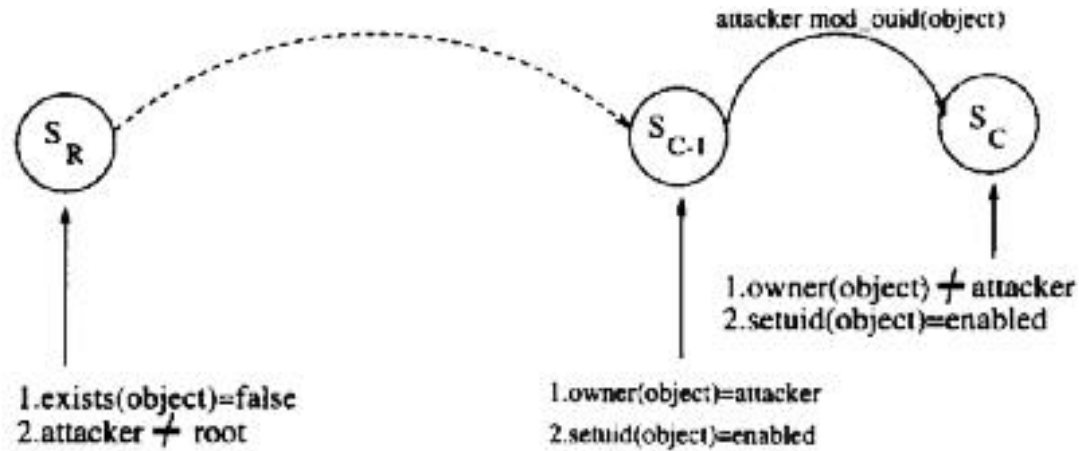
State transition Analysis

PENETRATION SCENARIO 1

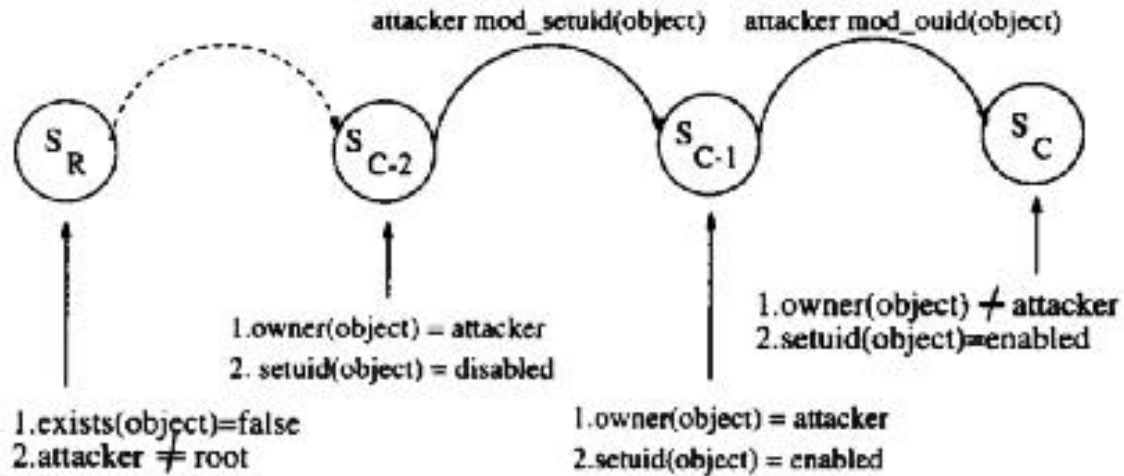
Step	Command	Comment
1.	%cp /bin/csh /usr/spool/mail/root	- assumes no root mail file
2.	%chmod 4755 /usr/spool/mail/root	- make setuid file
3.	%touch x	- create empty file
4.	%mail root < x	- mail root empty file
5.	%/usr/spool/mail/root	- execute setuid-to-root shell
6.	root%	



State transition Analysis

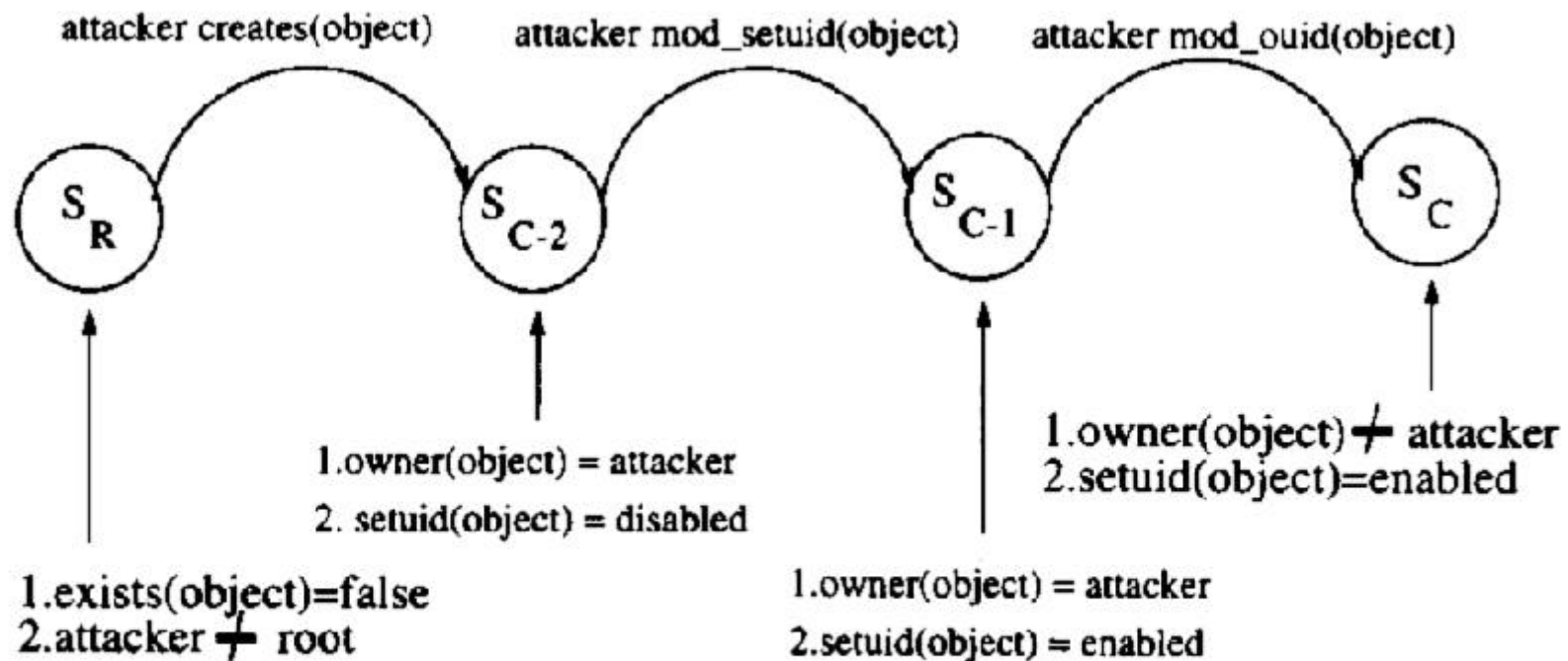


object = /usr/spool/mail/root



object = /usr/spool/mail/root

State transition Analysis

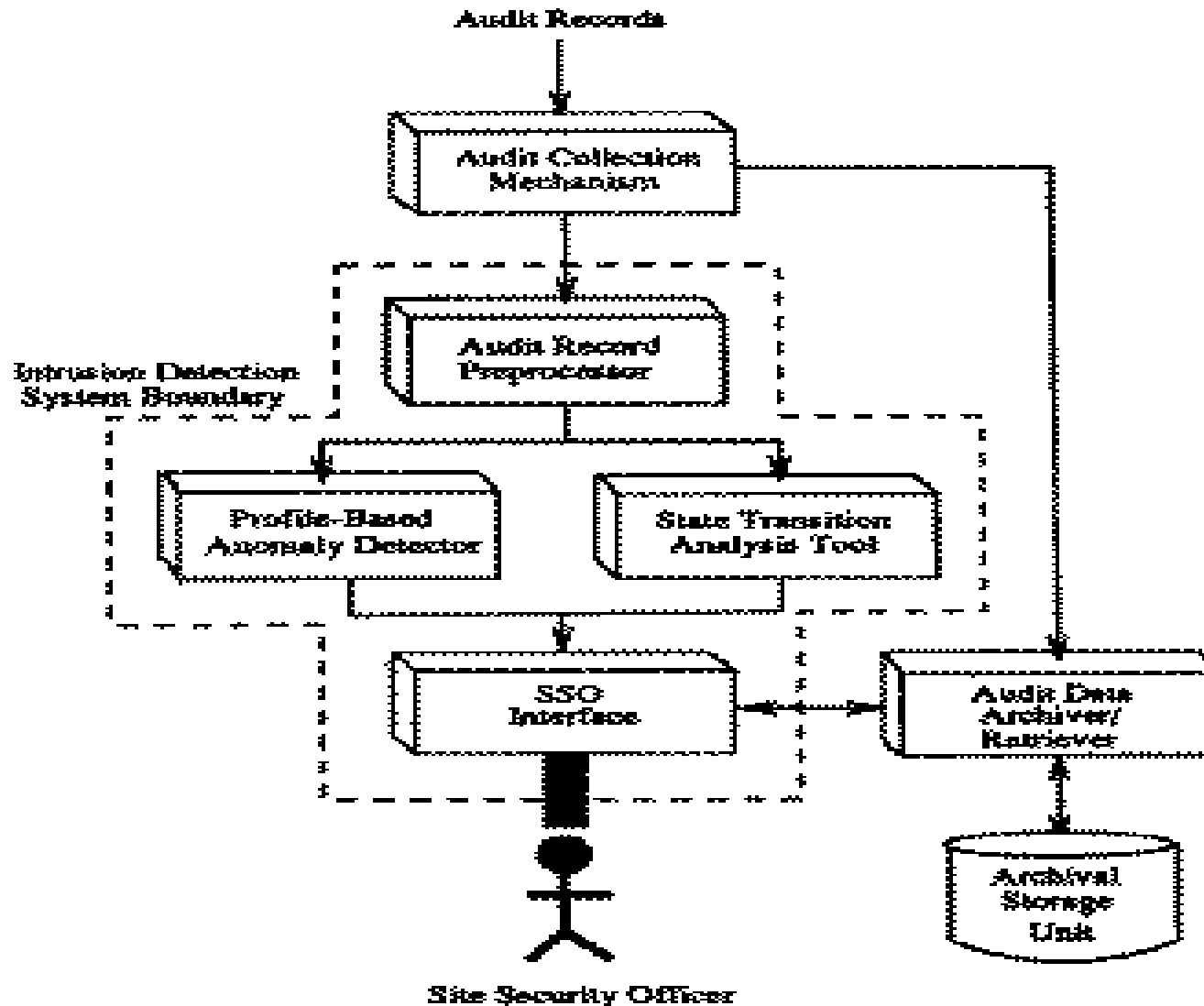


object = /usr/spool/mail/root

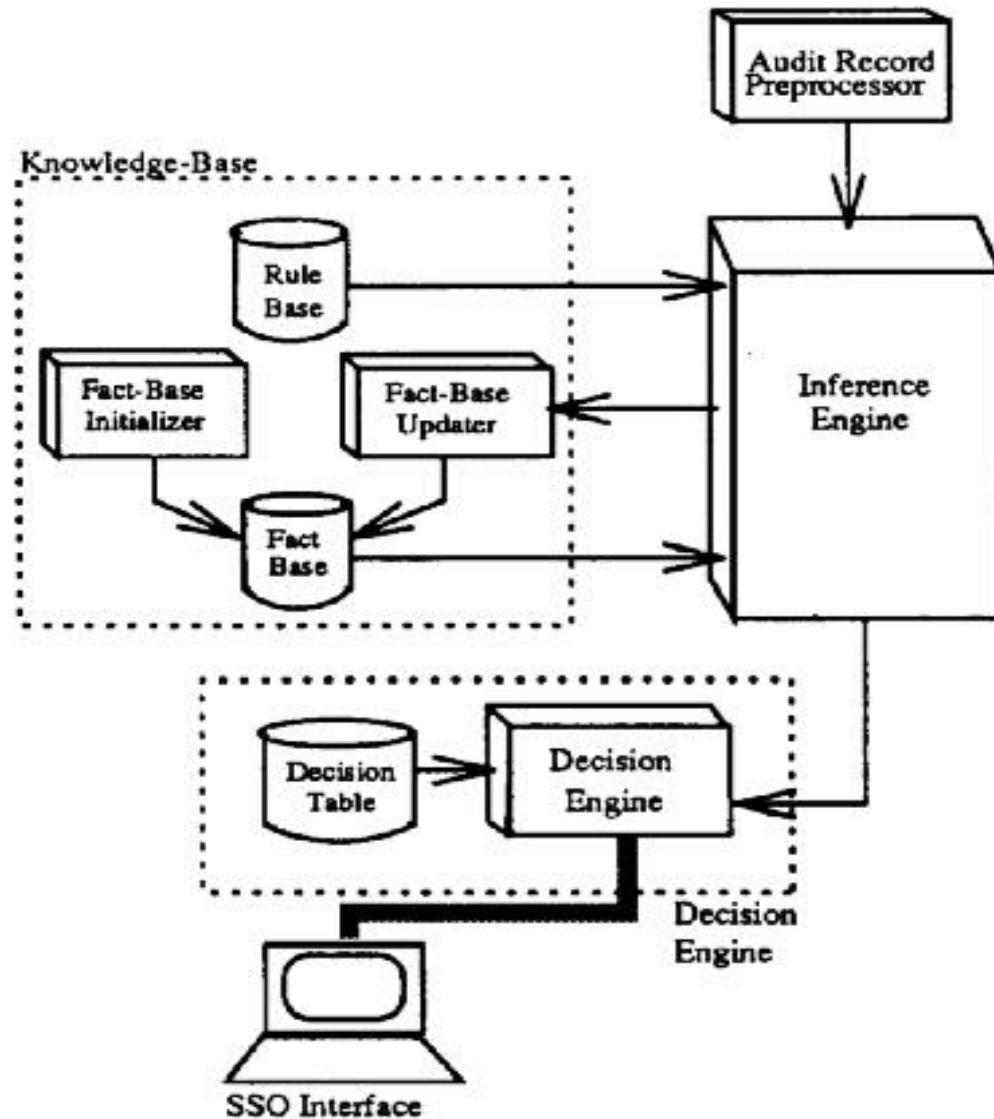
Applying state transition analysis to actual Computer states

- What can be
 - All known penetration scenarios that lead to an “identifiable” compromised state on the system are targeted
- When is a penetration representable with a STA?
 - The compromise must produce visible change to the system state
 - The compromised state must be recognizable without external knowledge
- What is needed to apply a STA to actual computer systems?
 - State changes in system attributes are required
 - Audit facilities, that record the state changes made by users on monitored system attributes, are required

Applying the STA Approach to Intrusion Detection



State Transition Analysis Tool



State Transition Analysis Tool

- Audit record processor:
 - Raw audit record reformatted and passed to the inference engine
- Inference engine:
 - Monitoring of the state transition extracted from the reformatted audit records
 - Comparing of the state transitions to the state transition scenarios represented in the knowledge-base
- Knowledge-base:
 - Collection and integration of all facts regarding STAT's execution environment and rules for detecting penetrations
- Decision Engine
 - result of the inference Engine monitored and action(s) taken based on the inference engine's findings

On a Pattern-Oriented Model for Intrusion Detection

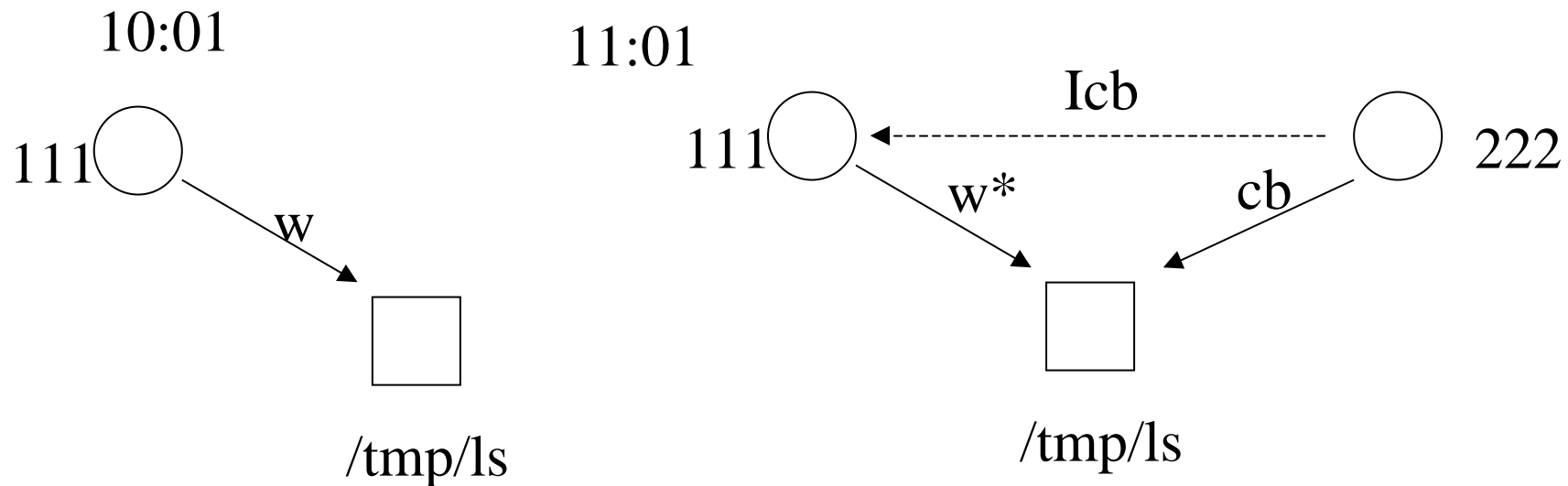
Shiuh-Pyng Shieh and Virgil d.
Glidor

Introduction

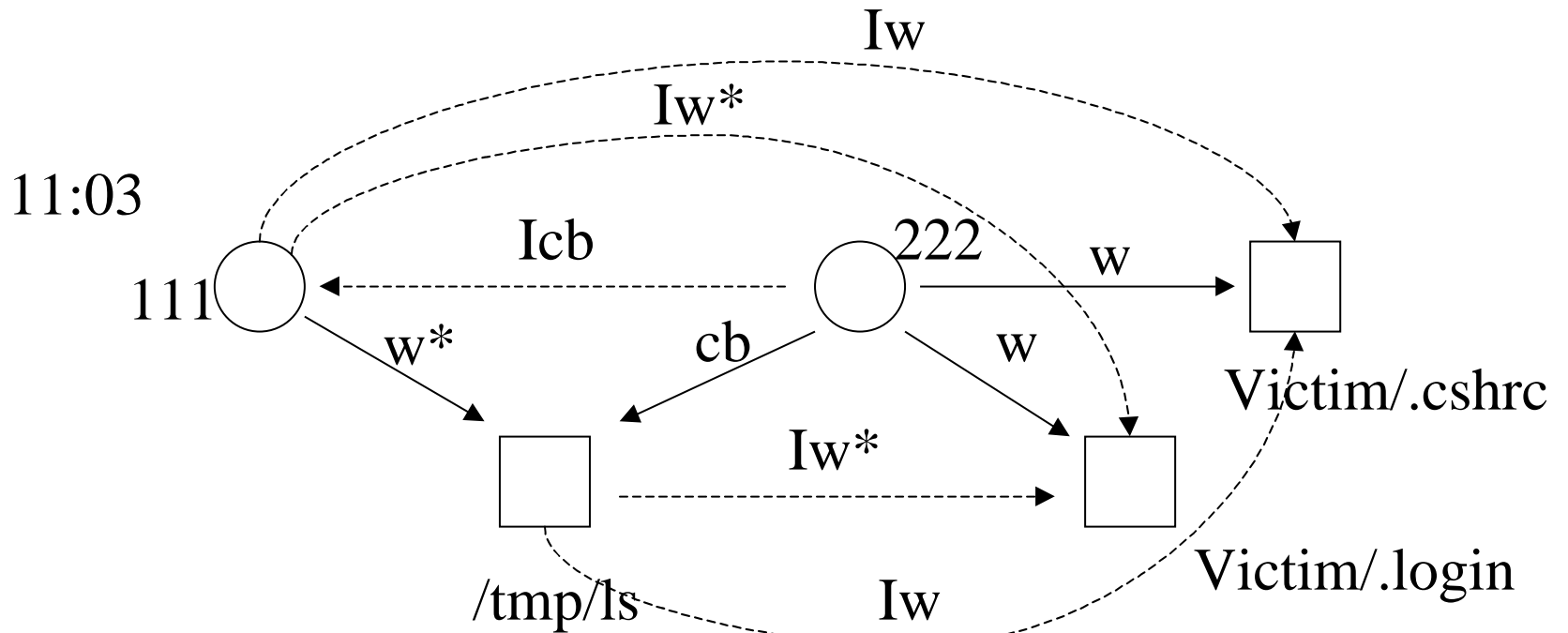
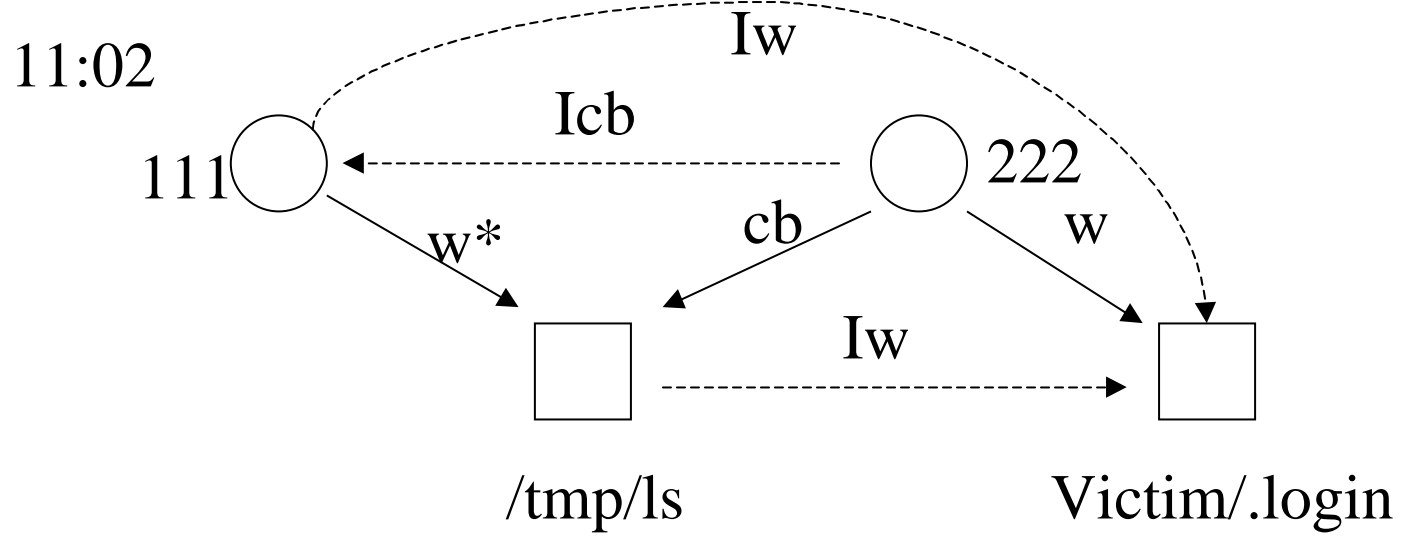
- Potential data and privilege flows between subjects and objects tracked
- Definition and discovery of specific intrusion patterns in audit trails enabled
- In particular patterns of data and privilege defined, that characterize operational security problems
- Only intrusions detected that can be anticipated prior to their occurrence

Example

Process	User	Primitive	Object	Time
111	Intruder	write	/tmp/ls	10:01
222	Victim	exec	/tmp/ls	11:01
222	Victim	write	Victim/.login	11:02
222	Victim	Write	Victim/.cshrc	11:03



Example



Comparisons

	STAT	IDES	W&S	Nadir	Pattern-oriented Model
Represent site specific policies	Yes	Yes	Yes	Yes	Yes
Detect cooperating attackers	Yes	No	No	No	Yes
Structured rule development (not ad-hoc)	Yes	No	No	No	Yes
Supports permutable event sequences	Yes	No	No	No	Yes
Support s longer rule chains/detects impending compromise	Yes	No	No	No	Yes
Capture the dynamic, potential flows of privileges and data	No	No	No	No	Yes

Conclusions

- All rule-based Intrusion Detection Systems:
 - Only intrusions that can be anticipated prior to their occurrence can be detected
 - Rule-based IDS must be integrated with statistical approaches for intrusion detection
- STAT:
 - Intrusion defined as a sequence of state changes that can be adopted to different changes of subject behavior
 - Still not able to track data and privilege flows between object and subjects
 - Rules should be easily extendable also by SSOs, but is that true?
- Pattern Oriented Model:
 - Greater level of abstraction from the audit data
 - Still a model, with no realization and never tested
 - How to extend the rules and protection graphs?

Bibliography

- T.F. Lunt, R Jagannathan, R. Lee, and A. Whitehurst, "Knowledge-based intrusion detection" in Proc. 1989 AI Syst. Government Conf., Mar. 1989, pp102-107 (IDES)
- S.H Teng, K. Chen and S.C. Lu, "Adaptive Real-time Anomaly Detection Using Inductively Generated Sequential patterns" Proc. IEEE Symp. Research in Security and Privacy, Okland, Calif., May 1990
- H.S. Vaccaro and G.E.Liepins "Detection of anomalous computer session activity", in Proc. IEEE Symp. Res. Security, Privacy, Okland, CA, May 1989, pp. 280-289 (W&S)
- T.D. Garvey and T.F. Lunt, "Model Based intrusion detection" in Proc. 14th Nat. Comput. Security Conf., MBaltimore,MD, Oct 1991 (Gister)