QUANTITATIVE SECURITY RESEARCH AT ILLINOIS: FROM DATA & MODELS TO METRICS

ZBIGNIEW KALBARCZYK

EMAIL: KALBARCZ@ILLINOIS.EDU

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

IFIP

JANUARY 2014

Outline

- Elements of quantitative security assessment
- Tools for security assessment
- Case for data-drive security metrics and monitoring
  - Early detection and mitigation of attacks
- Conclusions and lessons learned

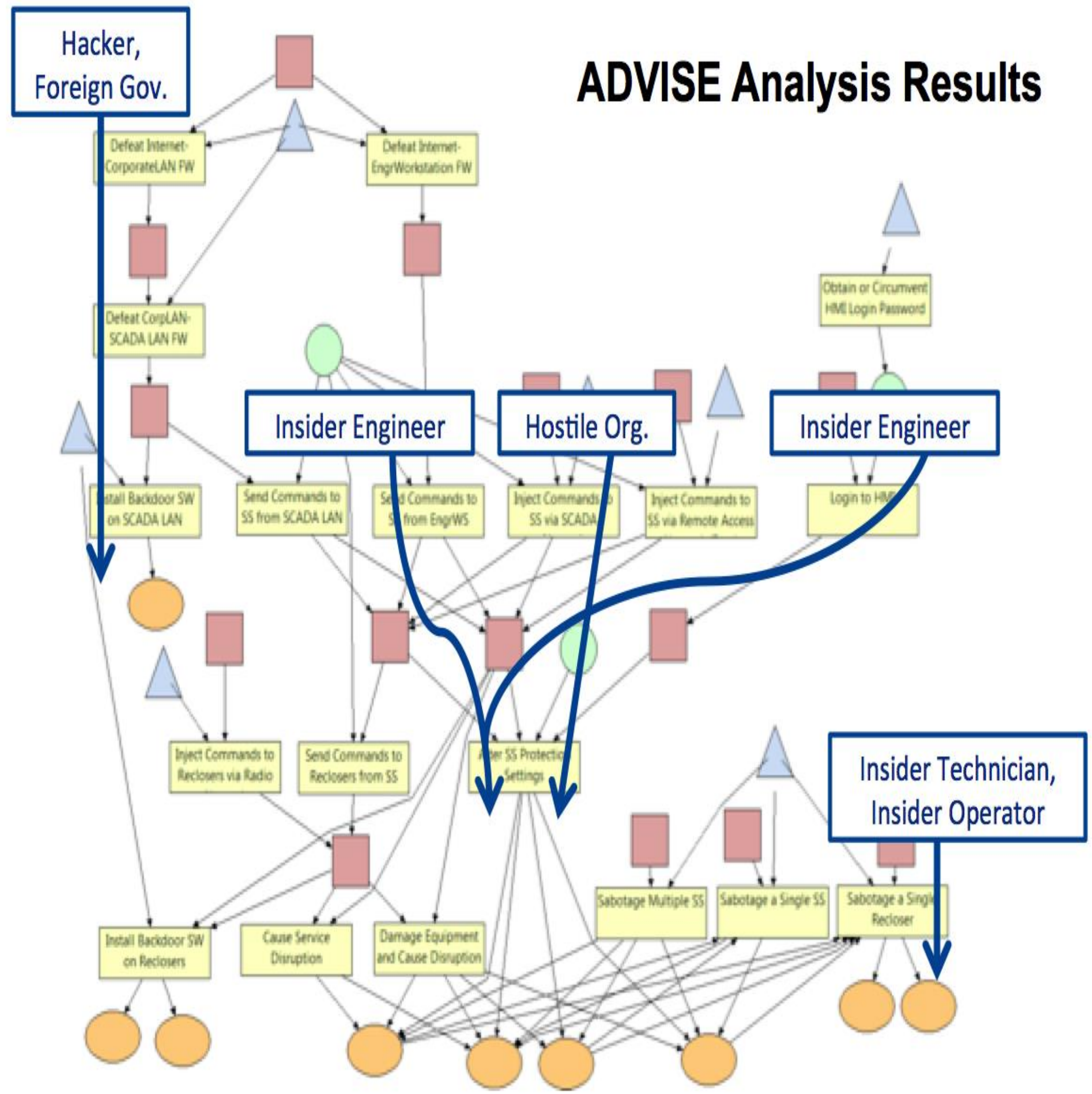
Elements of Quantitative Assessment of Security

- **Metrics**
  - *should either predict or confirm that a cyber system preserves a given set of security properties in a given context*
  - *data-driven*
  - *metrics on multiple levels (e.g., operational-level and technical metrics) must be integrated*

- **Models and Tools (examples)**
  - **ADVISE:** Design-time quantitative security assessment
  - **CyberSAGE:** Workflow-oriented security assessment
  - **MÖBIUS:** Model-based evaluation of systems

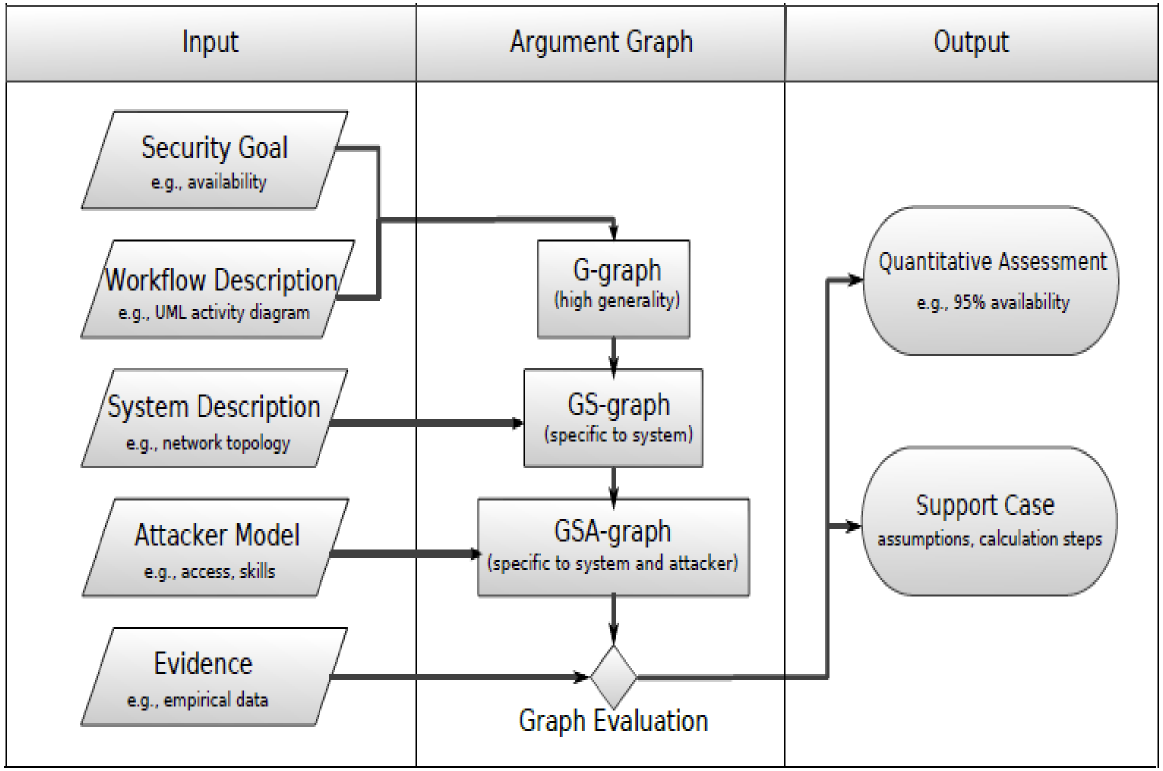
ADVISE: DESIGN-TIME QUANTITATIVE SECURITY ASSESSMENT

- ADVISE creates an executable state-based security model of a system and an adversary
- An attack decision function uses information about adversary attack preferences and possible attacks to mimic how the adversary selects the most attractive next attack step
- System architects can use ADVISE to compare the
  - security strength of system architecture variants
  - analyze threats posed by different adversaries.



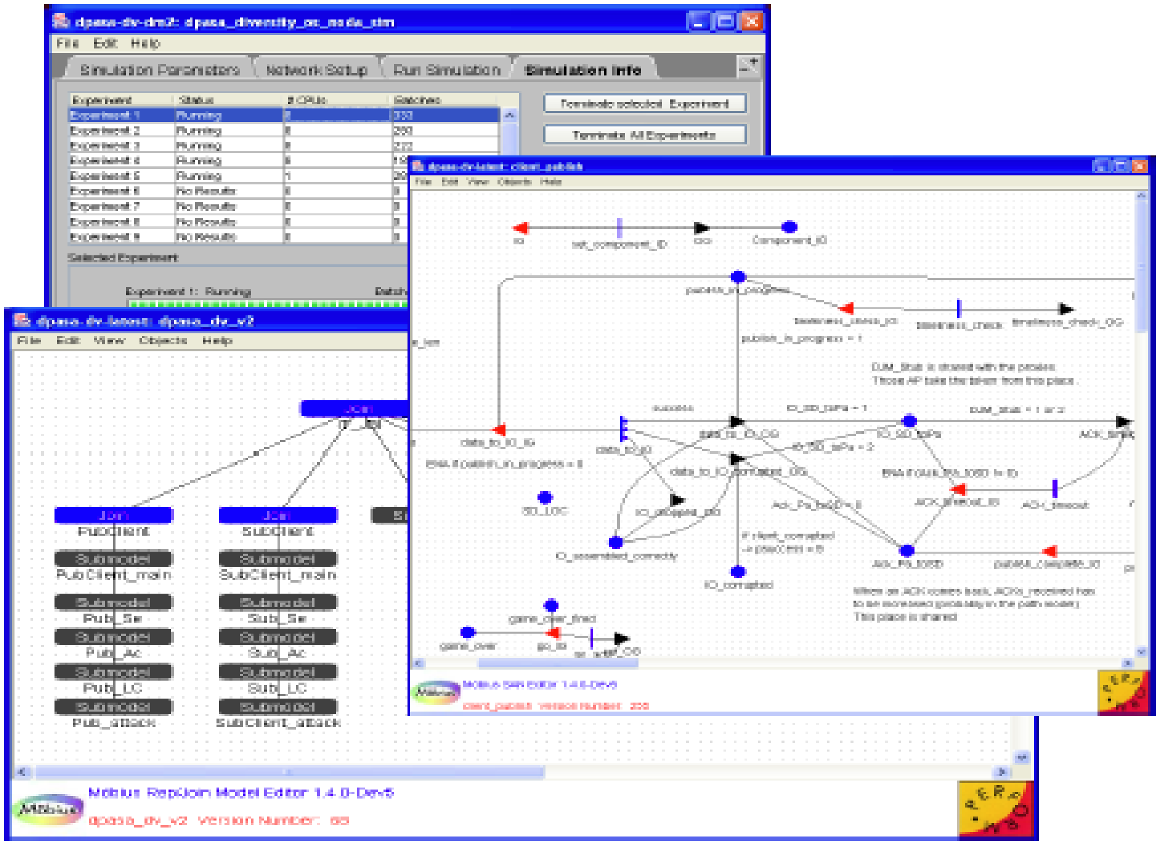
CyberSAGE: WORKFLOW-ORIENTED SECURITY ASSESSMENT

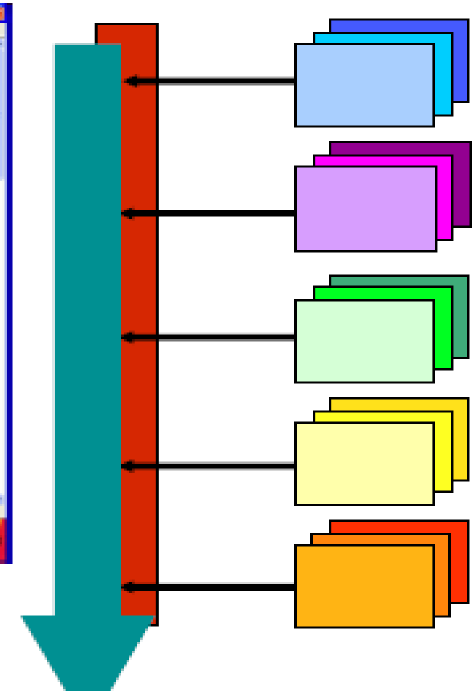
- Use the concept of **workflow** as a pillar of cybersecurity analysis
- Introduce a holistic workflow-oriented assessment framework
- Provides unify information about:
  - system components,
  - components properties,
  - possible attacks



- **to argue** about a security goal
- The argument is expressed in a graph structure, based on input from distinct classes that are integrated in a systematic manner to provide quantitative assessment in an automated fashion

MÖBIUS: MODEL-BASED EVALUATION OF SYSTEM

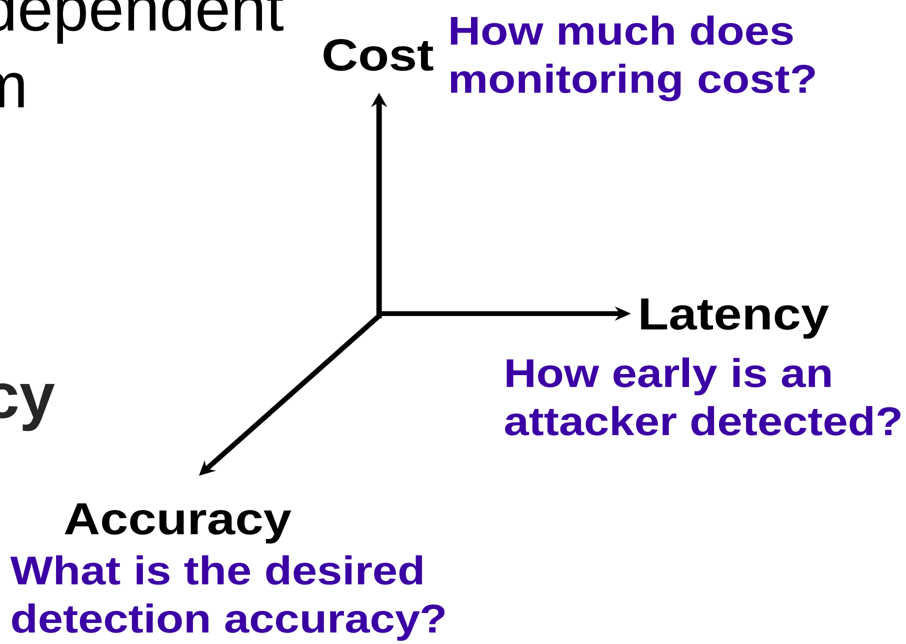




- Site licenses at hundreds of academic sites for teaching and research.
- Corporate licenses to a range of industries: Defense/Military, satellites, telecommunications, biology/genetics
- Development of new plugins for Möbius: Univ. of Dortmund, Univ. of Edinburgh, Univ. of Twente, Carleton University, and many others

Data-drive Security Metrics and Monitoring

- Use data on security incidents (NCSA security data) to:
  - drive development of security metrics
  - drive design of mechanisms for continuous monitoring
  - enable preemptive (i.e., before the system misuse) detection of attacks, e.g., execution under probation
- Search for solutions that are independent of a specific method/mechanism used to penetrate the system
- Fundamental tradeoffs:
  - **Cost** vs **latency** vs **accuracy**

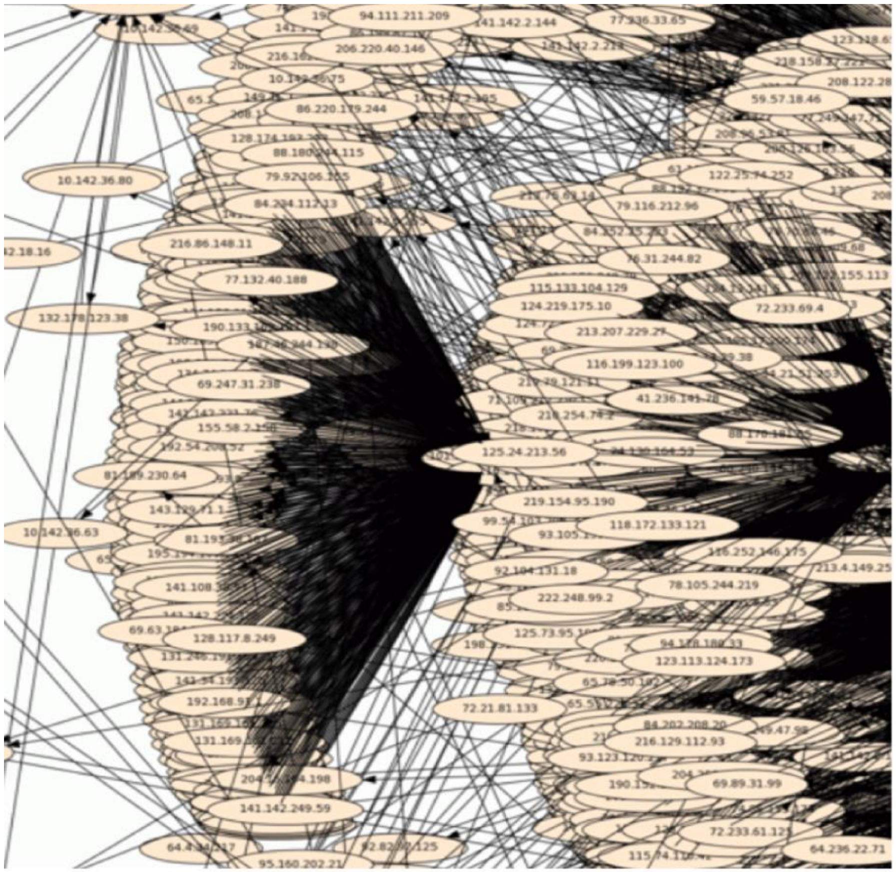


EARLY DETECTION AND MITIGATION OF ATTACKS: DATA-DRIVEN APPROACH

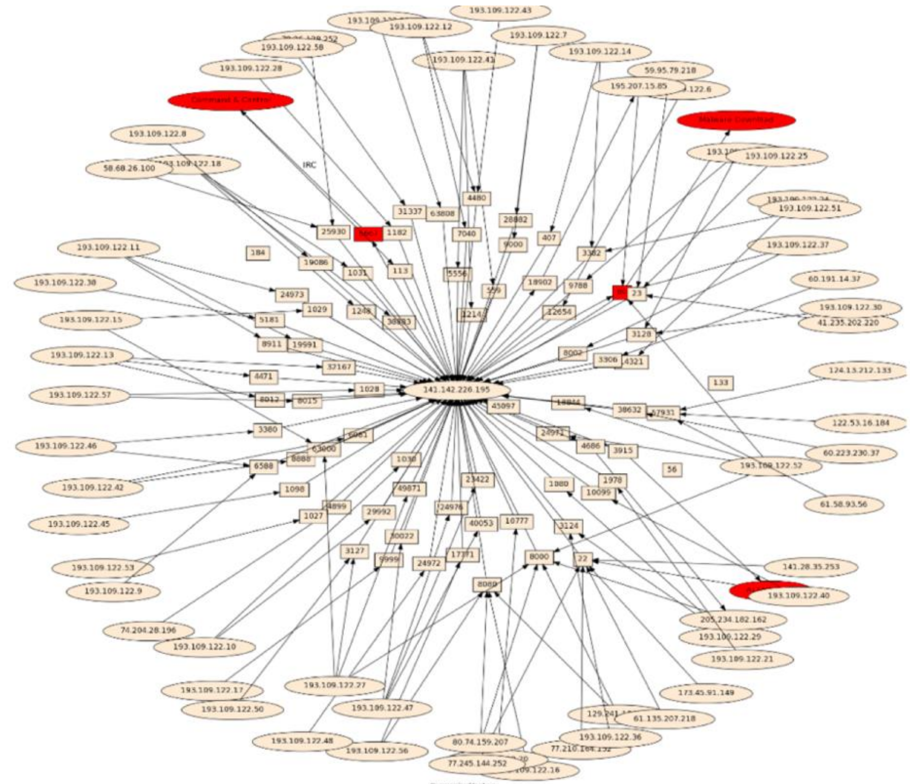
Goals

- Develop data-driven methods for uncovering attack patterns in large computing networked infrastructure
- Develop metrics to enable adaptive approaches to mitigate and contain the spread of attacks
- Achieve that in the presence of changes in the under-lying infrastructure and growing sophistication of attackers
- Build monitoring system and pre-emptive IDS for an early detection of security threats
  - detection before the system is misused

Magnitude of the Problem: Five-Minute of In-and-Out Traffic within NCSA



(a)

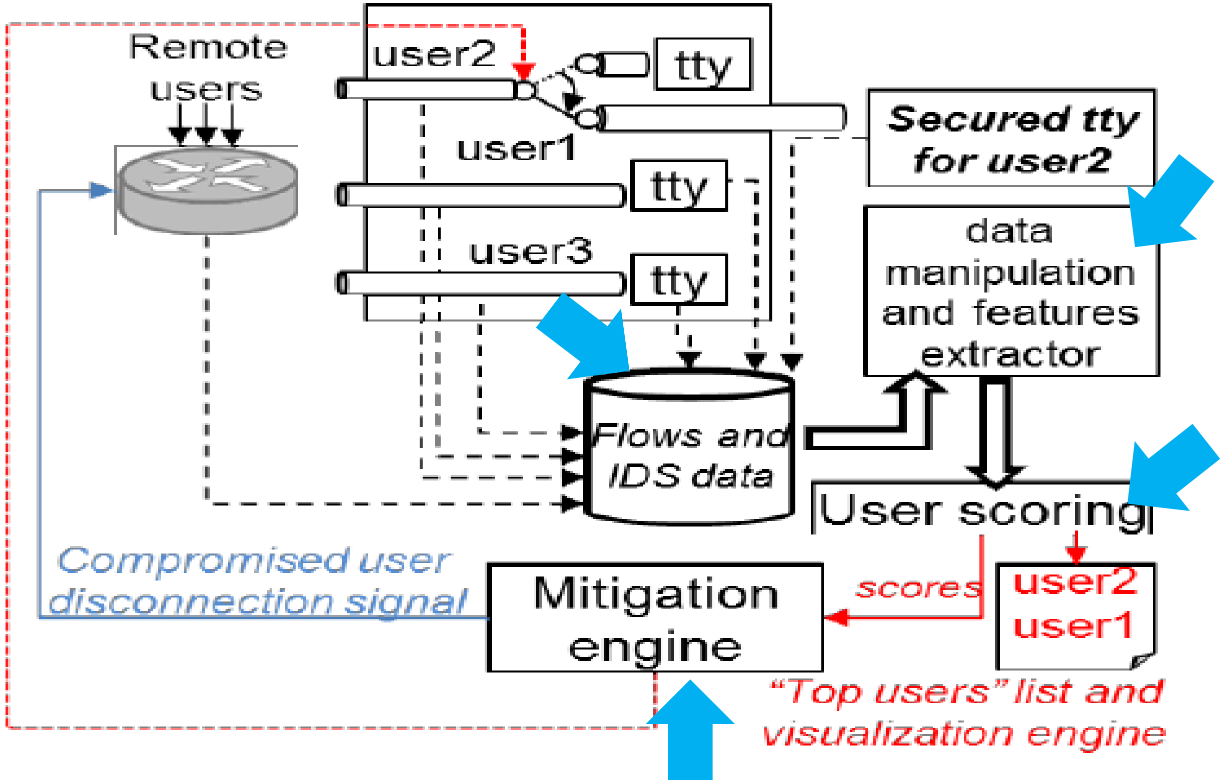


(b)

Approach

- Develop data-driven framework (SPOT) that integrates
  - runtime analysis of data collected by the monitoring tools
  - online detection of compromised users
  - attack containment techniques
- Provide low-latency high accuracy detection of compromised users
- Force suspect users to progress under close scrutiny in a secure terminal, i.e., a terminal with limited functionalities (e.g., limited set of commands) until the real intentions are clear

SPOT System Architecture



- Inputs: data from system level monitors: IDS logs, syslog, network flows, file system logs
- Scoring function: combines Bayesian network, rate of generated alerts, and entropy or alert diversity

Alerts Sample

Alert	Description
A1	**unknown address**: login comes from a previously unknown IP address, i.e., the user never logged from that IP according to his/her profile
A2	**multiple login**: the same external IP address is used by multiple users to log into the system
A3	**command anomaly**: a suspicious command is executed by the user
A4	**HotClusterConn**: a node of the computing infrastructure performs a download, although it is never expected to execute this action
A5	**HTTP sensitive URI**: downloading of well-known exploits, rootkits, and malwares (via HTTP get);
A6	**subsequent anomalous activities**: the remote IP address used to perform a login is involved in subsequent anomalous activities, e.g., A13, A14
A7	**watchlist**: the user logs from a blacklisted IP address; the list of suspicious addresses is hold and distributed among security professionals
A8	**suspicious multiple login activities**: generated if a user responsible for a multiple login is potentially related to other alerts in the security logs
A9	**FTP Sensitive URI**: downloading of well-known exploits, rootkits, and malwares (via FTP get);
A10	**unknown authentication**: according to the profile data, the user has never logged into the system by using that authentication mechanism
A11	**anomalous host**: the login is reported by a node within the infrastructure that has never been used by the user

- Total: 32 (A1- A32) unique alerts are available
- Analyzed alerts pertain to credential stealing incidents
  - 12 unique incidents
  - 1021 users involved
  - 324,424 total alerts

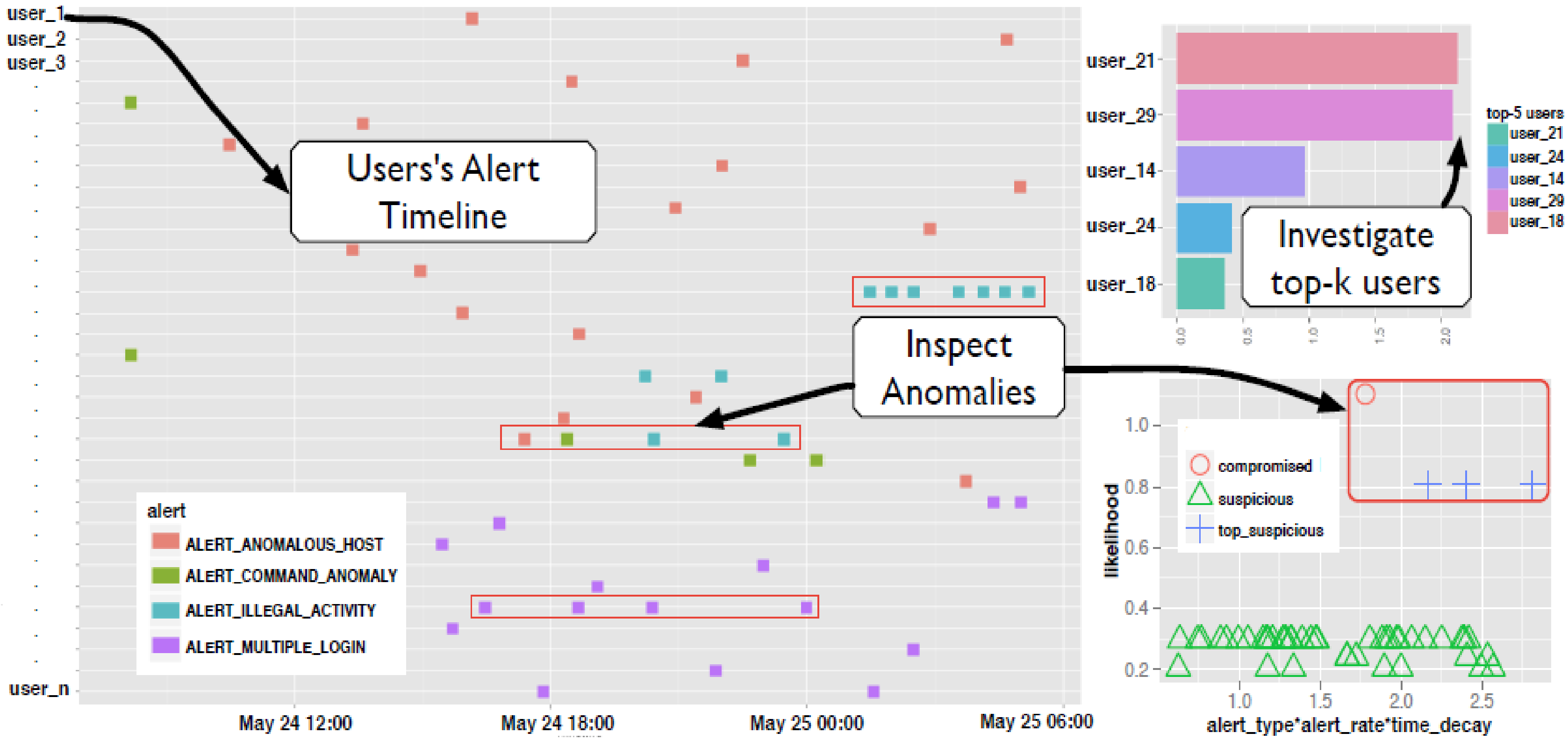
Scoring Mechanisms

- Score *(User Suspiciousness Metric*) of each user is proportional to:
  - likelihood of being an compromised user
  - type of alerts (alert variability) – the entropy of an alert set raised by a user over time.
  - rate of alerts – e.g., our prior work revealed one to five security alerts per hour
  - a time decay function, which decrease the suspicious score exponentially over time

$$Score = Likelihood \times Alert\_Types \times Alert\_Rates \times Decay$$

- User is declared as compromised if:
  - user appears in the top-k list at time of query t_{now}
  - the user *Suspiciousness Metric* is δ times standard deviation $\sigma_{t_{now}}$ from the mean $\mu_{t_{now}}$

System Dashboard: Alerts Timeline & Score



(i) timeline of alerts generated by each user (left part of the graph),

(ii) top-k most suspicious users (right upper corner)

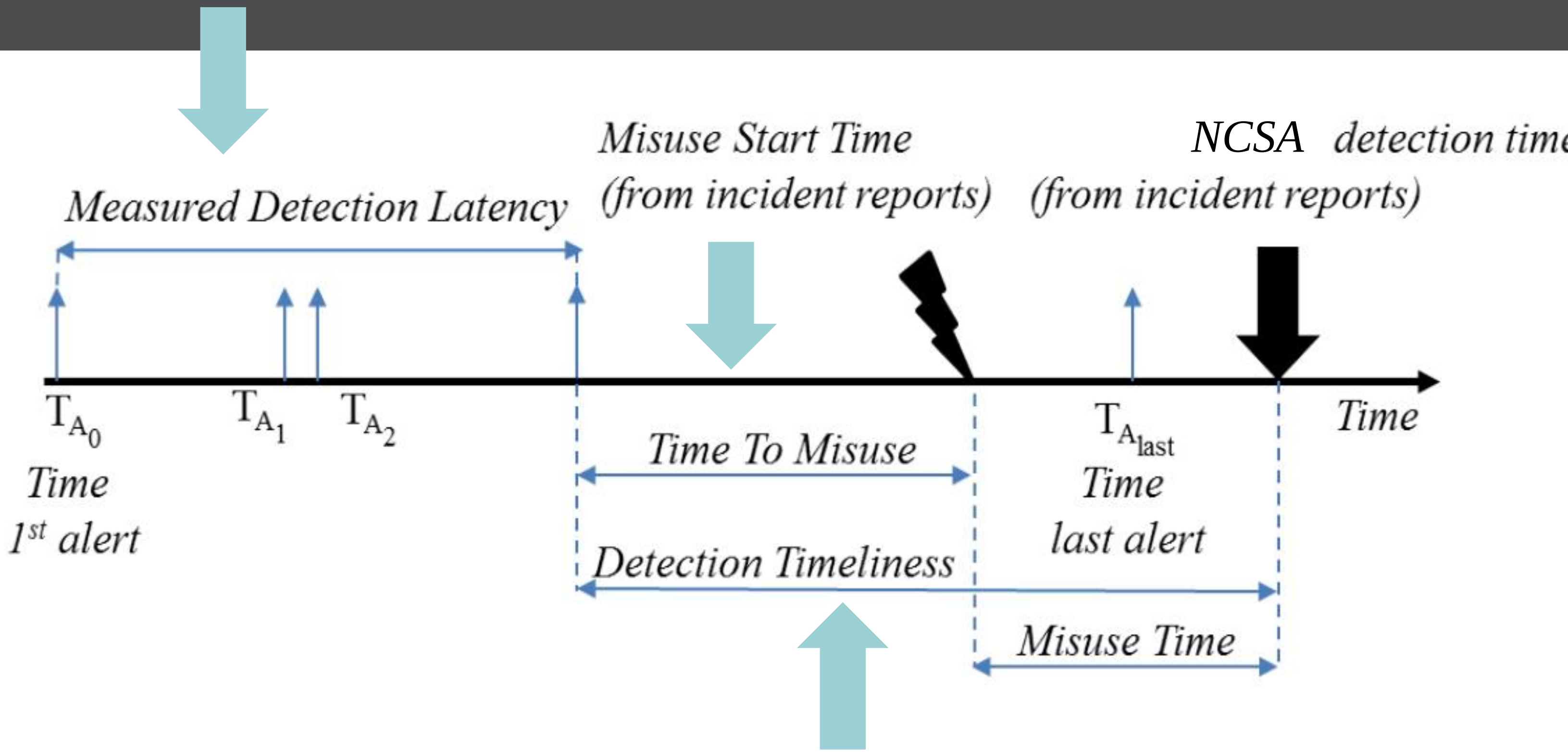
(iii) visualization of the score function for the users (right bottom corner)

x axis represents alert types, rate and time decay of alerts generated by the user

y axis represents likelihood the user is a compromised user.

cluster near the x axis captures the suspicious users and cluster (at the top) consists of the top suspicious users

Evaluation: Time Metrics



Detection Latency: time needed to detect a compromised user

Detection Timeliness: how much ahead of NCSA detection time we detect the compromised user

Time to Misuse: how much ahead of the misuse we detect the compromised user

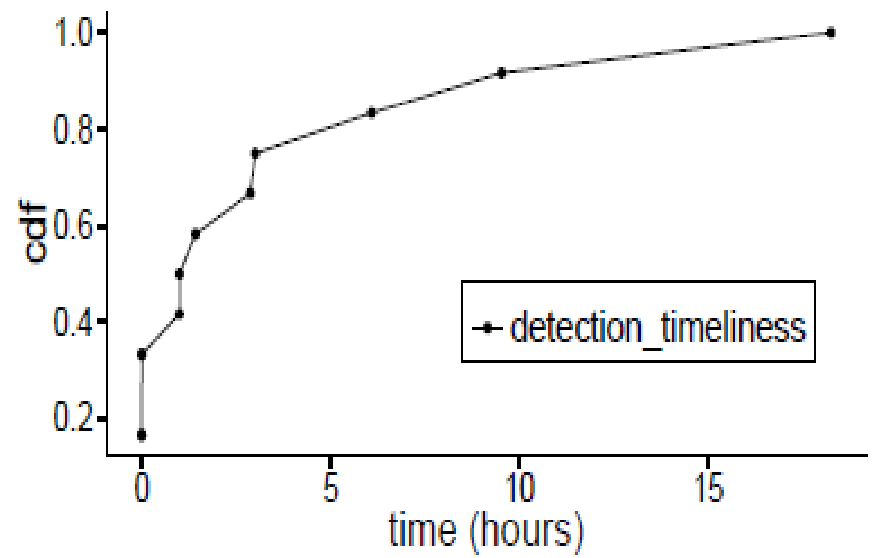
Evaluation: *Pre-emptive* Attack Detection

- Early detection of an attack before system misuse
- In average, SPOT detects attackers 1.2h ahead of system misuse
  - NCSA data analysis shows that 97% of incidents are detected after a real compromise

✓ 80% of attacks are detected 5 hours before the real misuse

✓ best case early detection time is 18h before the misuse

✓ worst case, SPOT misses only one attack and detect two attacks after the misuse



Evaluation: Scoring Function Effectiveness

- Attack detection rate $$\frac{detected\ compromised\ users}{total\ number\ of\ compromisedusers}$$

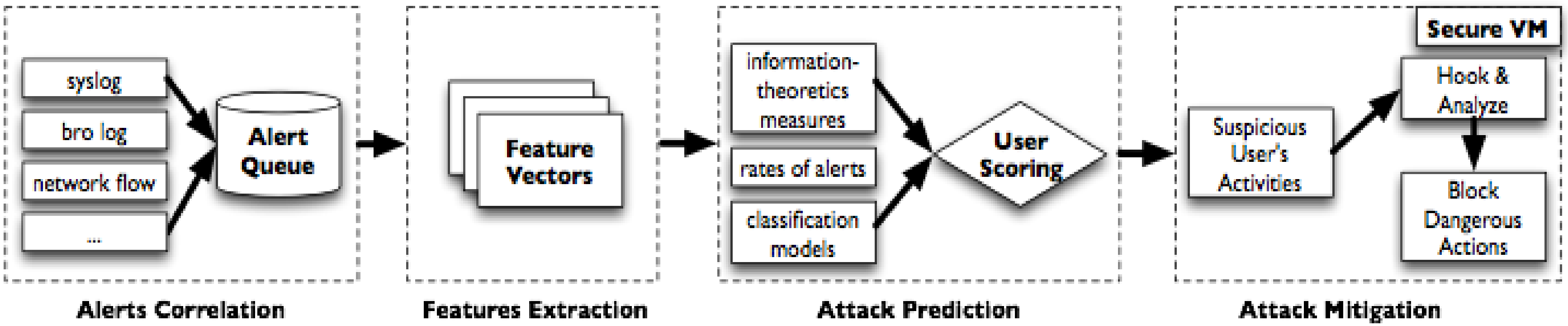
- False detection rate $$\frac{detected\ suspicious\ users\ as\ compromised\ user}{total\ number\ of\ suspicious\ users}$$

- Detection accuracy $$\frac{detected\ compromised\ users + detected\ suspicious\ user}{total\ users}$$

- Sample classification results:

- Attack detection rate: 93%
- False detection rate: 21% → reduced to 4% by execution under probation (secure terminal in our study)
- Detection accuracy: 78%

Toward *Pre-emptive IDS (or IPS)*



Alert Correlation:
Correlates alerts of system and network events to users.

Features Extraction
Extracts meaningful features from raw log data to classify malicious users.

Attack Prediction
Assigns score and ranks suspicious users.
Puts the top-k suspicious users to probabtion (jail).

Attack Mitigation:
Prevents attackers from executing malicious commands.

Conclusions

- Develop sound methods for uncovering attack patterns in large computing networked infrastructure
  - extract the underlying models,
  - develop methods and tools
- Build monitoring system and *pre-emptive* IDS for an early detection of security threats
  - Explore a new scoring mechanism for ranking (and detecting) suspicious users based on alerts collected from IDS
- Proposed approach (tested using credential stealing incidents) can provide early detection of intruders
- Need to evaluate the approach for other types of incidents