Efficient Discovery of Abnormal Event Sequences in Enterprise Security Systems

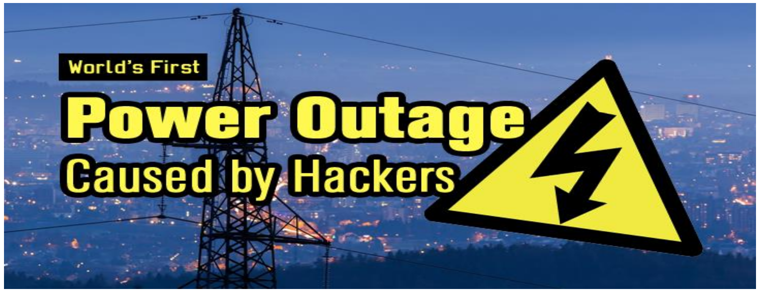
Zhengzhang (Zach) Chen
NEC Laboratories America, Inc.

Other Authors: Boxiang Dong, Hui Wang, Lu-An Tang, Kai Zhang, Ying Lin, Zhichun Li, and Haifeng Chen

Motivation: Cyber-Attacks Seriously Impact Our Society

- Data breach and business secret loss
  - Example: **Target Breach** (Nov-Dec 2013)
    - **TARGET** (the 2nd largest retailer in US, Global500:#97)
    - **40 million** credit cards leaked, **140+ lawsuits**
    - Net earnings down for **$1.02 Billon [¥106 Billion] (30%)**
    - **CEO, CIO, CISO** all got replaced
  - Many similar cases for the companies who have **large amount of consumer data**: Equifax, Yahoo, CHASE Bank, SONY, Ebay, JAL, KDDI etc
    - Data is business essentials but also a liability
- Use cyber-attacks to affect physical infrastructure
  - Example: Ukrainian power grid outages affected 225K customers in 12/2015

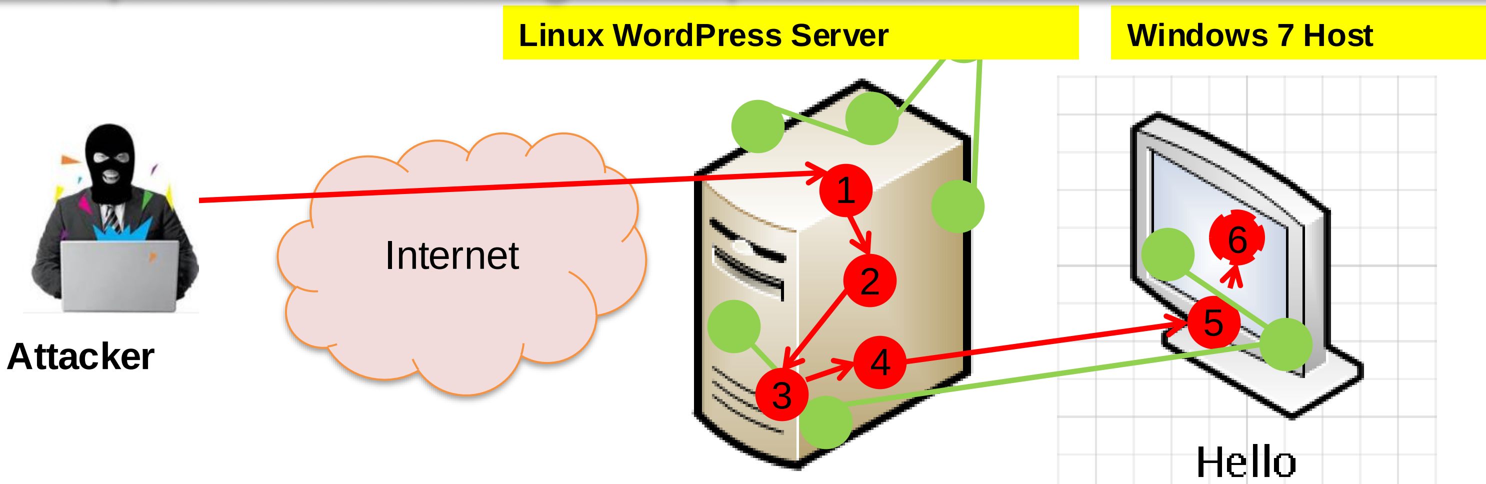
http://blog.mslgroup.com/the-target-breach-has-changed-everything-even-sxsw/



http://thehackernews.com/2016/01/Ukraine-power-system-hacked.html

Motivation: Connecting Suspicious Dots

Attacks (like APT) involve multiple steps; Causal path connects individual attack points based on logical dependencies.

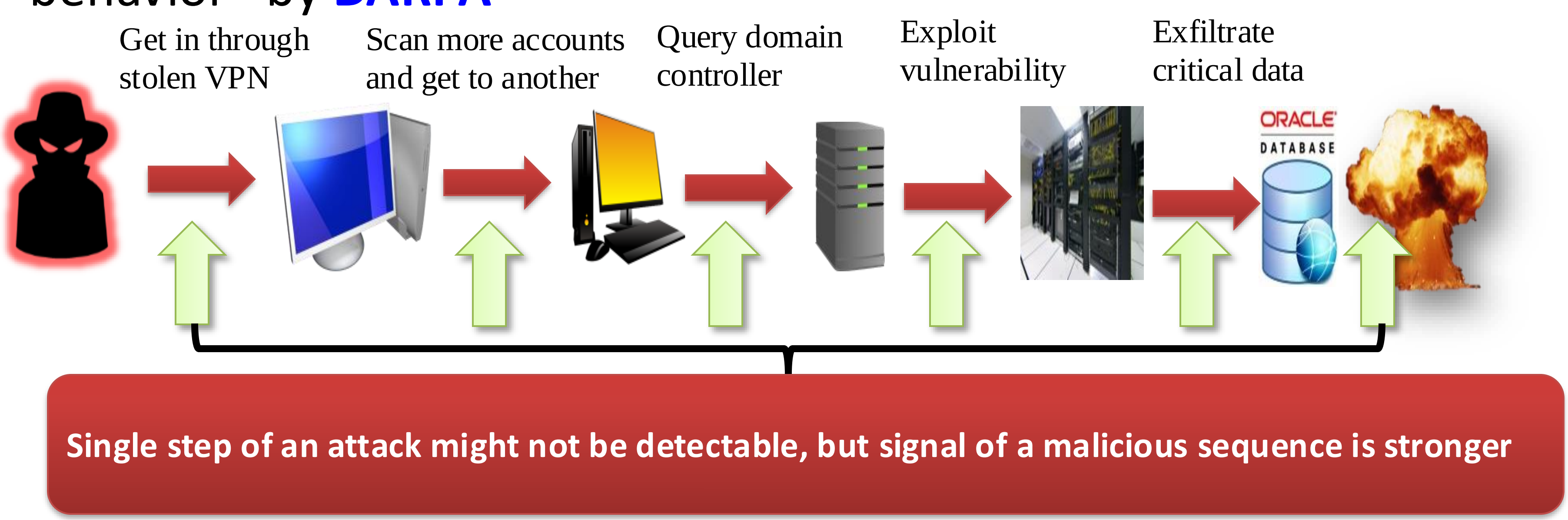


Example: Attacker penetrates Linux server and jumps to Windows machine to steal user credentials

❶ Exploit WordPress vulnerability (CVE-2015-1172)	❹ Host Trojan malware in Apache server on the same Linux server
❷ Drop malicious PHP file through WordPress	❺ Windows machine downloads Trojan from Apache server via IE
❸ WordPress loads malicious PHP downloading Trojan malware	❻ Execute Trojan malware to steal user passwords stored on Windows machine

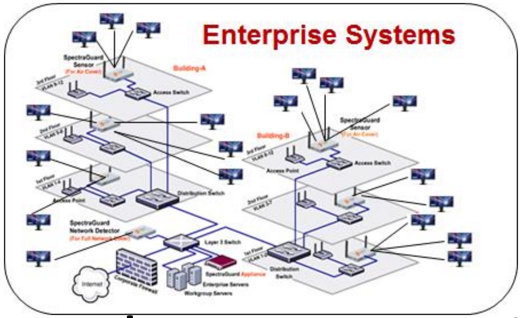
Motivation: Discover the Whole Chain of Attackers' Activity and Their Intention

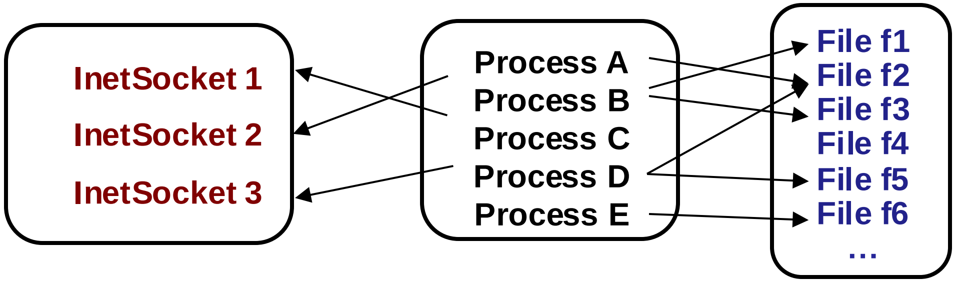
- Single entity/event based or signature-based model will not work
- Hacking behaviors usually consist of a sequence of events
- Not every step is suspicious and can be detected initially
- Isolated entities/events can't serve as strong evidence for users
- "**Connecting the dots**" **across multiple events** that "are may individually legitimate but collectively indicate malicious or abnormal behavior" by **DARPA**



Problem Statement: Malicious Event Path Discovery

- System monitoring data that contains a set of heterogeneous events





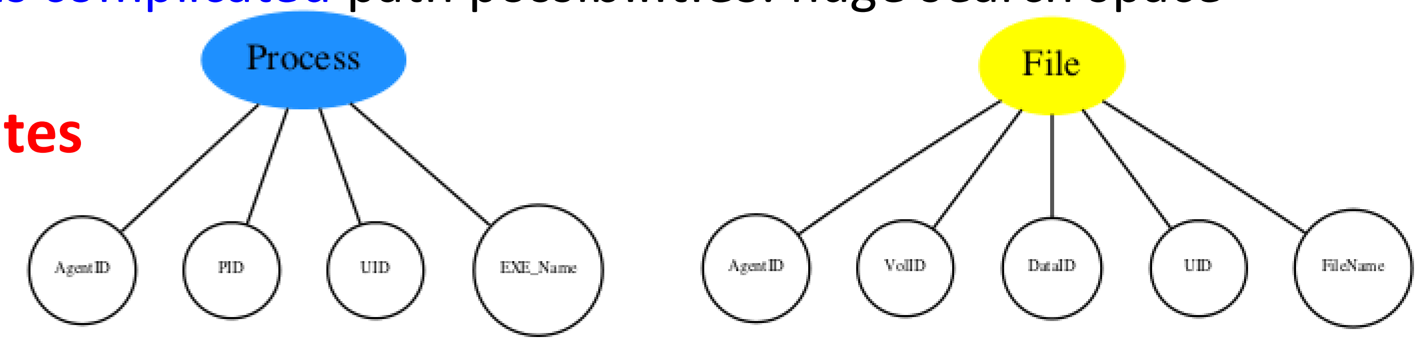
- Given the user-specified positive integers L and K, and time window size Δt, find the top K abnormal event sequences that include at most L system events occurring within the time period of Δt

Key Issues:
(1) how to define and compute the anomaly score of event sequence containing heterogeneous entities; and,
(2) how to rank the event sequences of different lengths

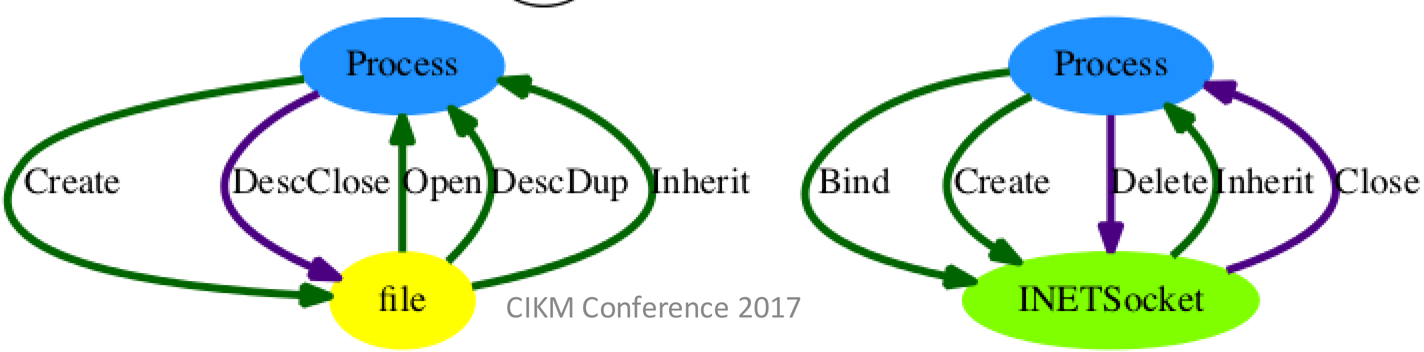
Challenges: Massive Heterogeneous Categorical Data

- Data complexity
  - Data volume: 100s hosts, 1 day with 10 million nodes, and 100 million edges(events)
  - Heterogeneous categorical data: different types of entities and events; each entity is associated with a number of categorical attributes
  - Highly dynamic: evolve with time
  - Numerous complicated path possibilities: huge search space

entity-attributes



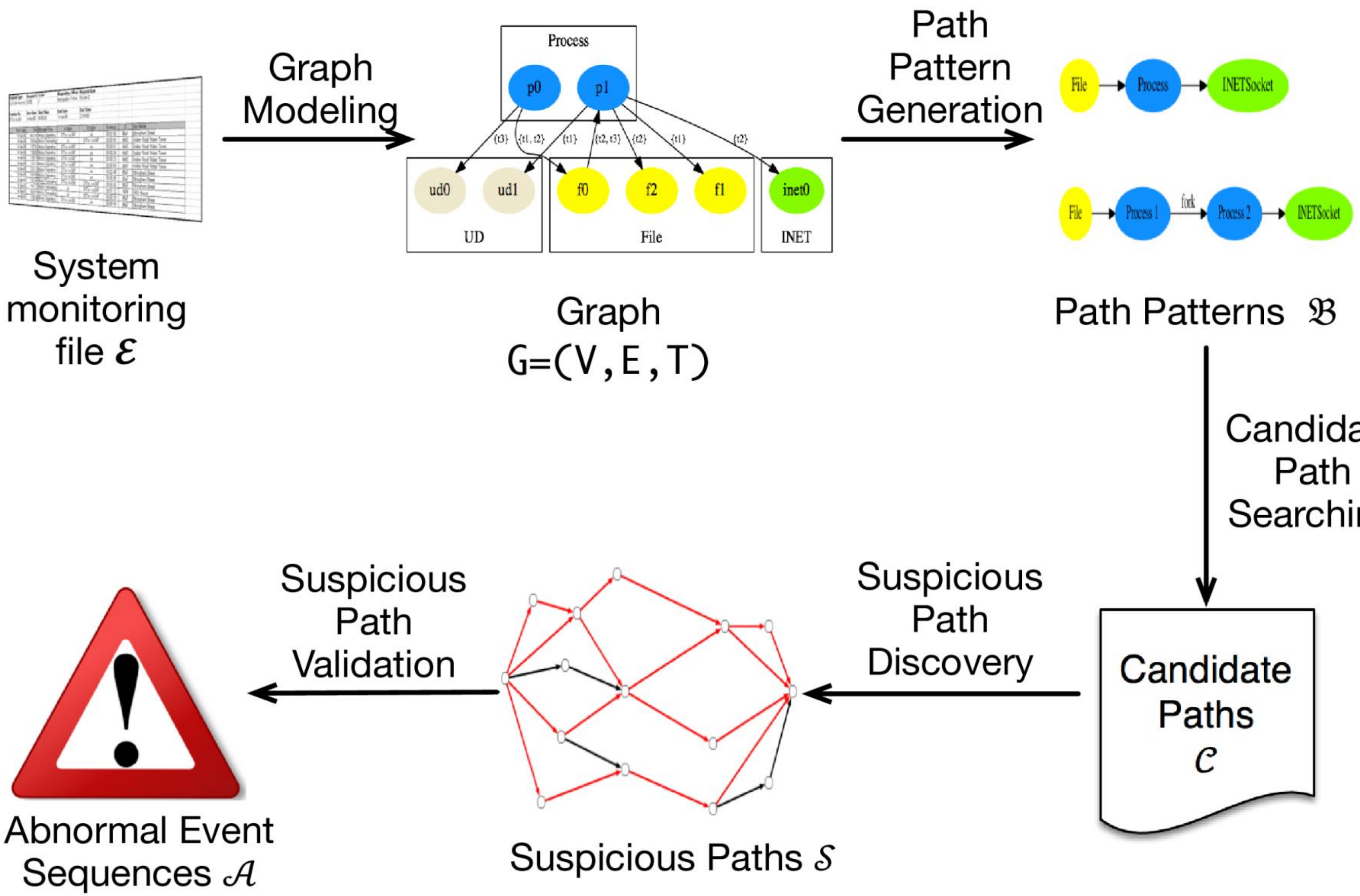
entity-entity relations



Challenges: Data Driven and Real-time

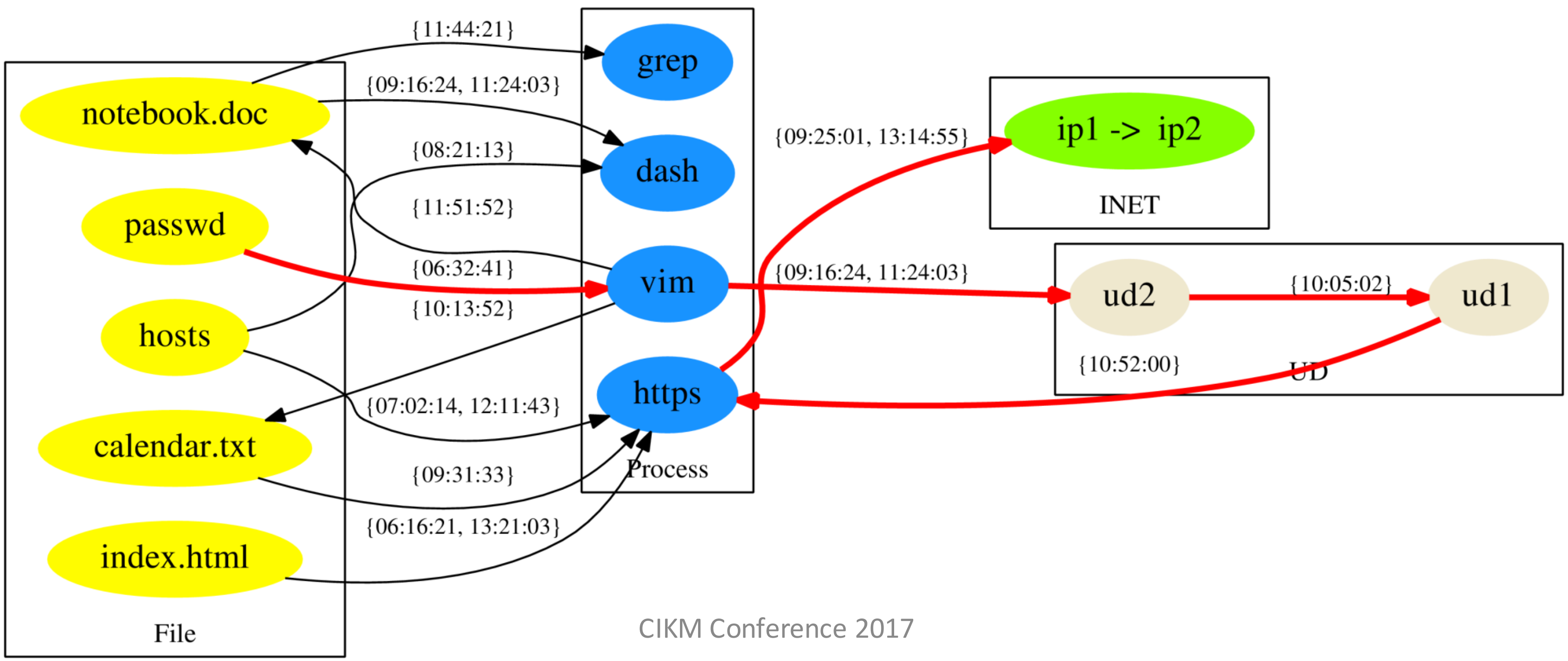
- Data complexity
  - Data volume: 10s host, 1 day with 10 million nodes, and 100 million edges(events)
  - Heterogeneous categorical data: different types of entities and events; each entity is associated with a number of categorical attributes
  - Highly dynamic: evolve with time
  - Numerous complicated path possibilities: Huge search space
- Data driven/unsupervised: no normal behavior profile
  - Existing anomaly detection algorithms (k-grams based, trajectory based) often require to profile normal behavior at first
- Real-time detection
  - An efficient algorithm is required

Framework: Graph-based Intrusion Detection



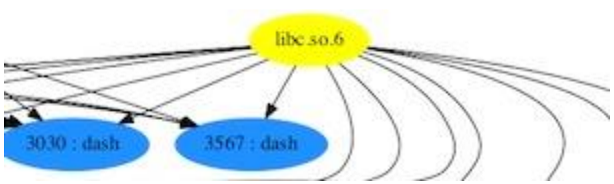
Method: Graph Modeling

- System event data are often redundant
  - Events involve the same entities
  - Attributes of entities are repeatedly stored
- How to represent the massive highly dynamic events as a graph?
- Build a blue-print graph G per host per time window

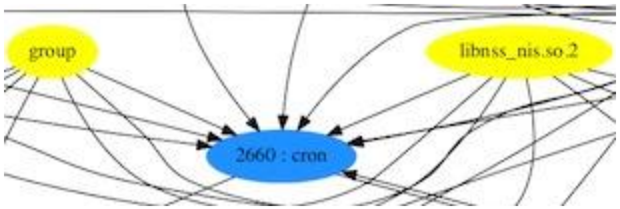


Method: Path Anomaly Score Calculation

- Basic idea: define anomaly based on both nodes and edges



- Each node has two roles:
  - Information sender
  - Information receiver



- Send score: a good sender tends to send information to many good receivers
- Receive score: a good receiver tends to receive information from many good senders
- Intuitively, if there are so many bad senders and receivers in the path, it is suspicious

Method: Probabilistic Model for Path Anomaly Score Calculation

- Information flow matrix A^{N*N} of G
  - $A[i][j] = prob(v_i \rightarrow v_j)$

transition probability

- Assign initial send and receive scores
  - Send score V_0: $V_0[i] = v_i's\ initial\ send\ score$
  - Receive score U_0: $U_0[i] = v_i's\ initial\ receive\ score$

- Iteratively update score

A node's send score is the sum of the receive score of the nodes it points to.

  - $$\begin{cases} V_{k+1} = A * U_k \\ U_{k+1} = A^t * V_k \end{cases} \xrightarrow{\text{yields}} \begin{cases} V_{k+1} = (A * A^t) * V_{k-1} \\ U_{k+1} = (A^t * A) * U_{k-1} \end{cases}$$

A node's receive score is the sum of the send score of the nodes that point to it.

Path Anomaly Score Calculation: Theoretical Convergence Proof

- Convergence problem
  - Bipartite or cyclic multipartite graph: has been proved to be a convex problem
  - In most cases, our graph is not a strongly connected graph
  - Acyclic multipartite graph: unknown
  - A^{N*N} is not irreducible, not to mention $A * A^t$ or $A^t * A$.
  - V and U do not converge
- Convergence with restart

  restart probability

  - $\tilde{A} = (1 - c) * A + c * R, c \in (0, 1), R[i][j] = \frac{1}{N}$

  restart matrix

  - $\tilde{A}$ is both irreducible and aperiodic
  - $\begin{cases} V_{k+1} = \tilde{A} * U_k \\ U_{k+1} = \tilde{A}^t * V_k \end{cases}.$
  - $\tilde{A}$ makes the score calculation converge
  - The convergence rate depends on $(1 - c)$

Method: Suspicious Path Detection

- Path score calculation
  - $AS(path\colon v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_n) = 1 - NS(path) = 1 - \prod_{i=1}^{n-1} V(v_i) * U(v_{i+1}) = 1 - \prod_{i=1}^{n-1} sed(v_i) * rec(v_{i+1})$.

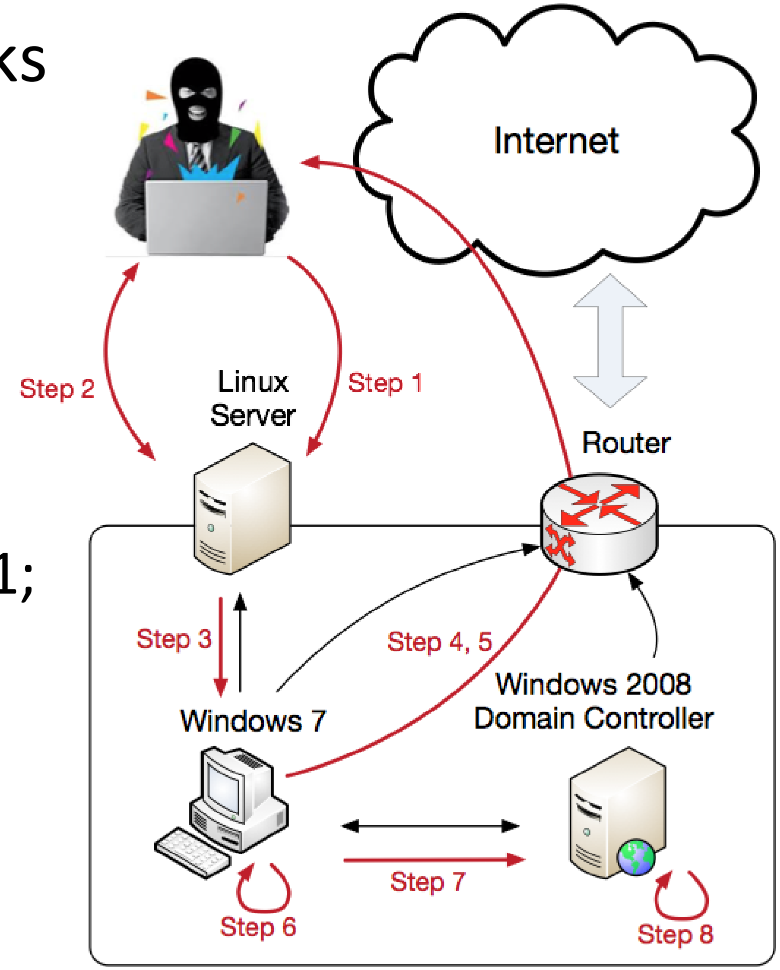
edge $v_i \rightarrow v_{i+1}$'s normality score

- Suspicious path
  - $path\colon v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_n$ is suspicious if $AS(path) > \alpha$.

- To eliminate the score bias from the path lengths, normalize the scores using Box-Cox power transformation function

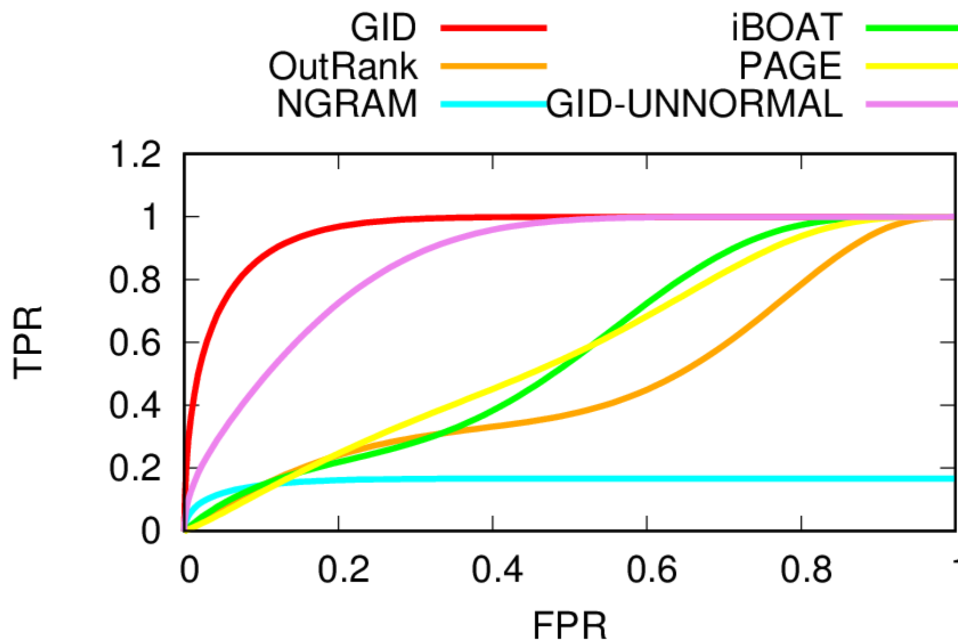
Experiment Setting

- Real-world system monitoring datasets from enterprise networks
  - Design our agent to collect the monitoring data from 33 UNIX computers of **NEC Lab at Princeton**
  - Total: 3 days' 157 GB data with 440 million system events
  - Processes: 410,166; Files: 1,797,501; Sockets: 203,467
  - **10 different types** of attacks ran by **Russian hackers** with steps/lengths varies from 3 to 7
  - Both **offline and online**

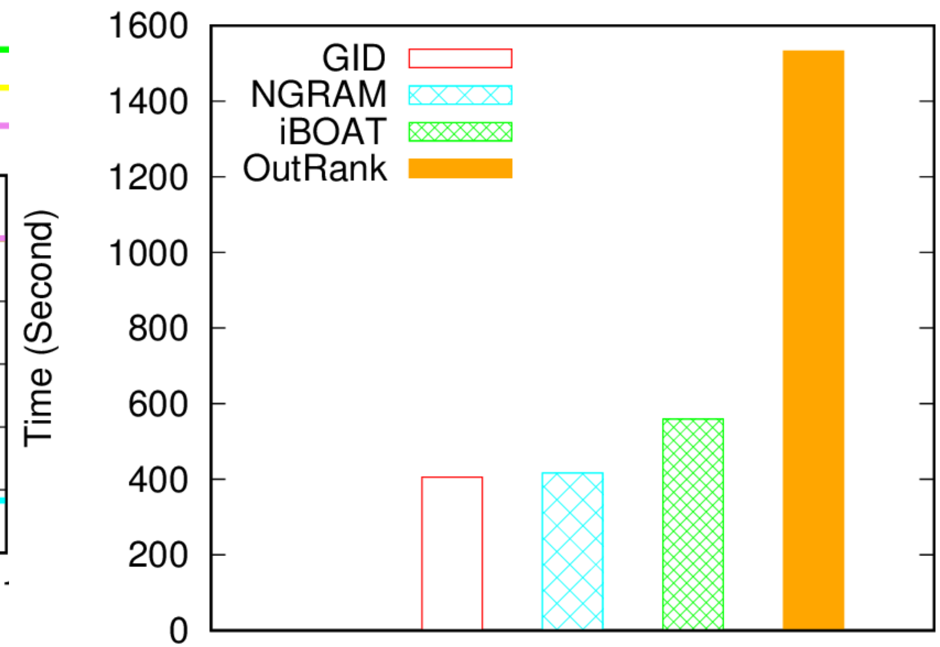


Experiment: Static Evaluation

Setting: all data (8 million events) is fed to detection; stored in memory



Accuracy comparison (ROC curves)

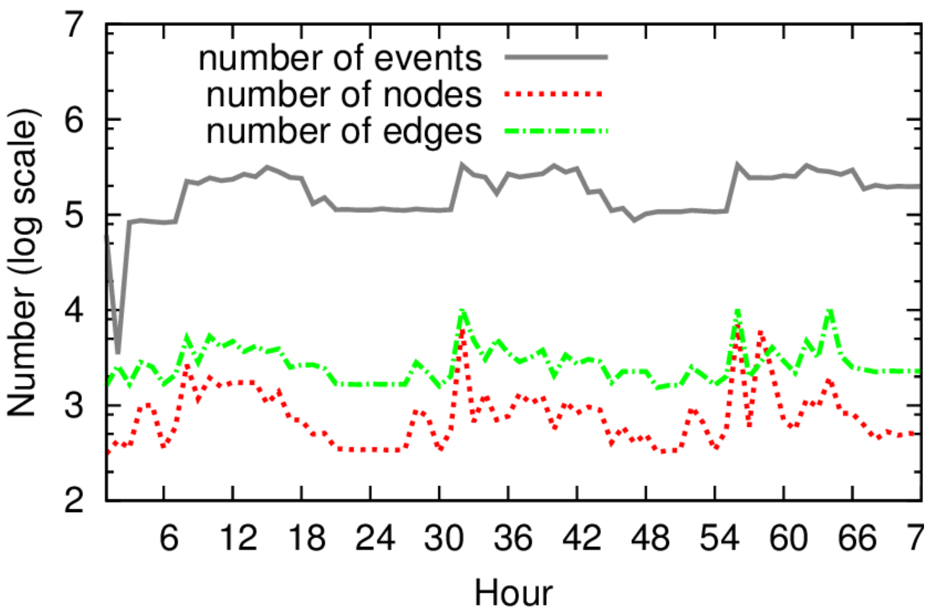


Running time comparison

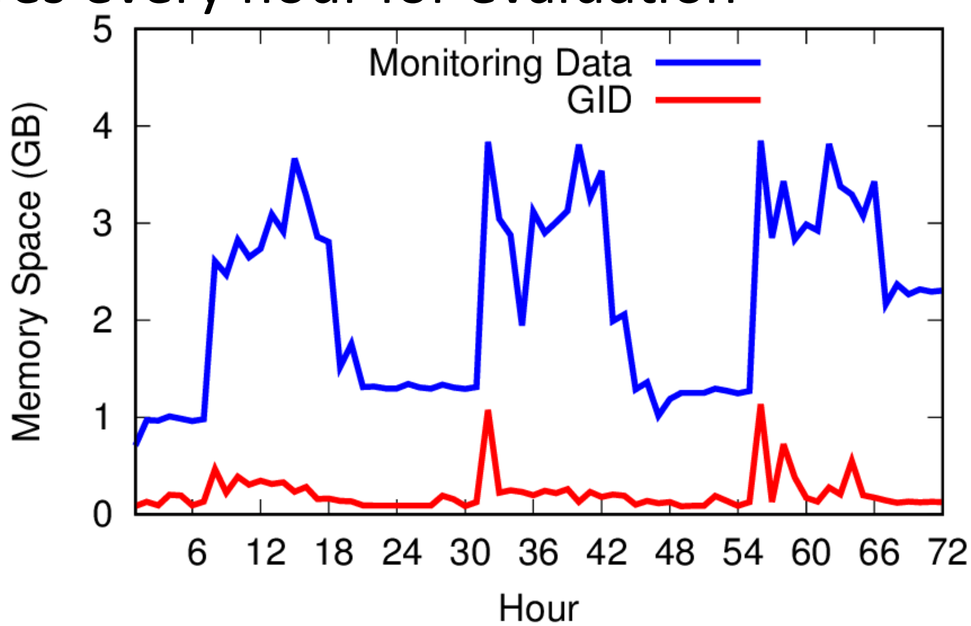
GID outperforms all baselines in terms of accuracy and running time

Experiment: Streaming Evaluation

- System events are processed one at a time, update the graph and sender/receiver scores using incoming events
- Retain a snapshot of entity scores every hour for evaluation



Graph size vs. number of events



Memory usage comparison

GID maintains the scalability to be deployed for real-time detection

Conclusion

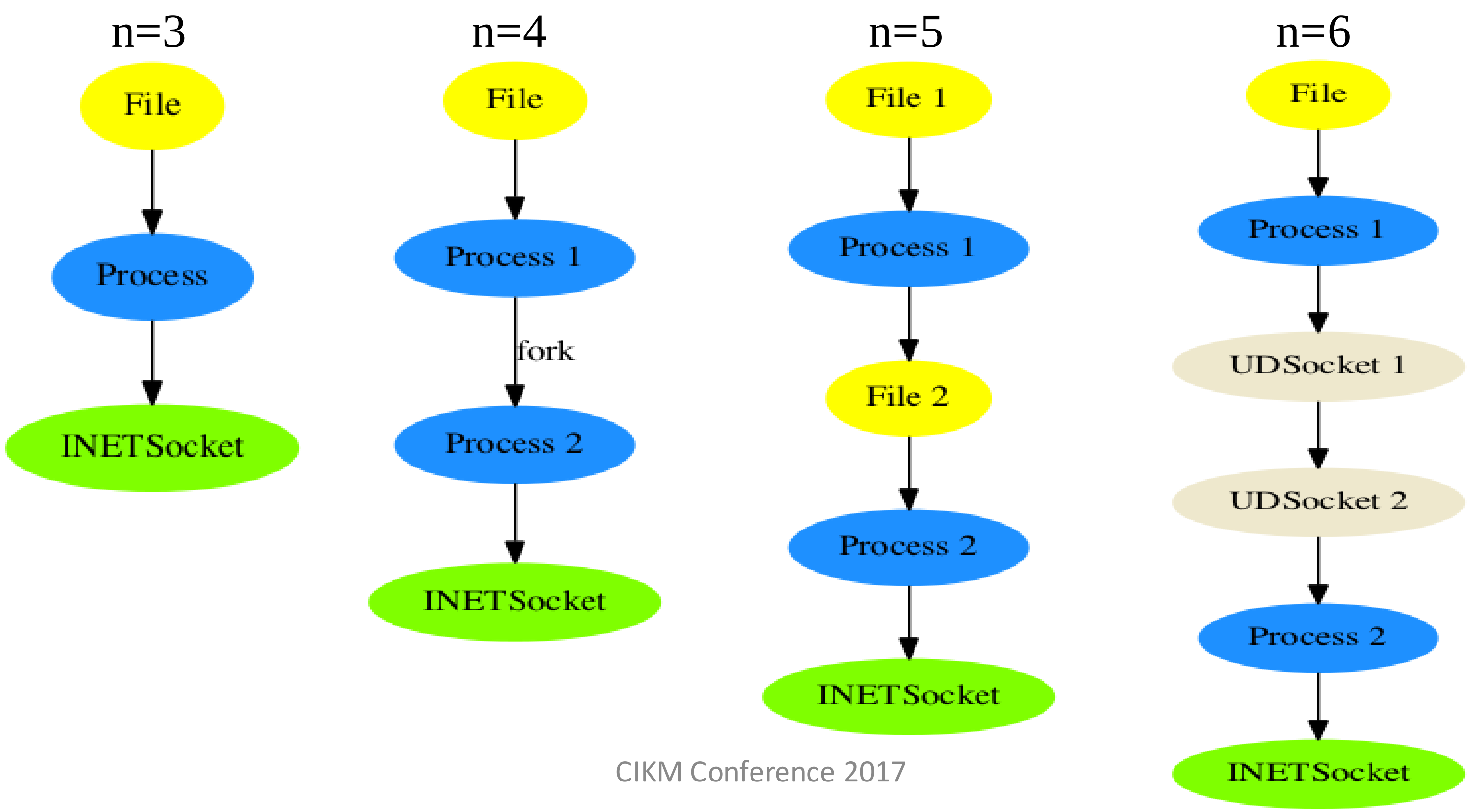
- Identify an important and challenging problem of suspicious event sequence detection
- Develop an efficient suspicious path discovery algorithm and prove its convergence on directed acyclic graphs
- Experimental results show the effectiveness and efficiency
- Fully develop the detection engine and deploy it into a real enterprise security system
- Can be generalized to detect suspicious event sequences in other domains/applications

QUESTIONS?

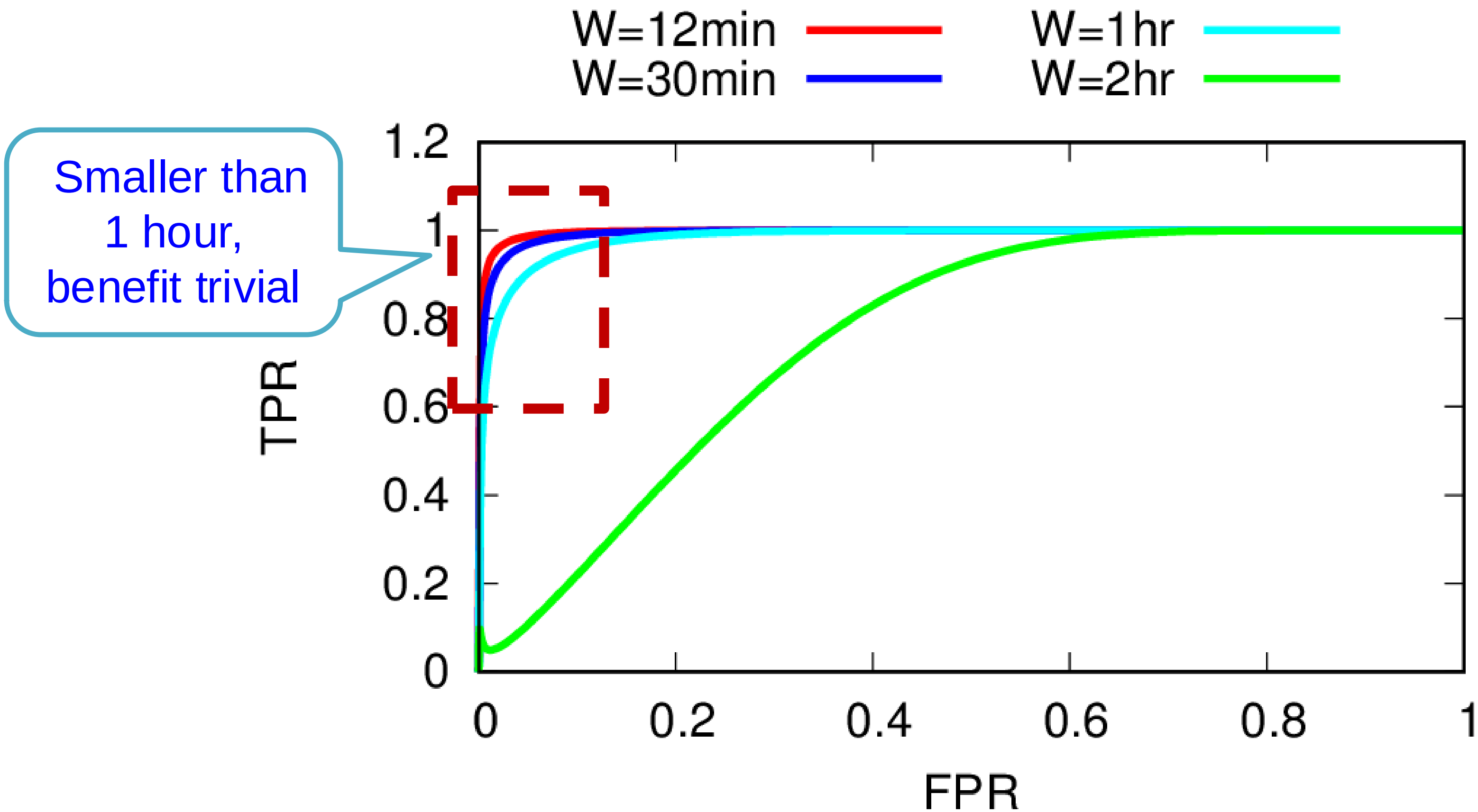
THANK YOU!

Method: Candidate Path Searching

- Domain knowledge
  - File -> … -> INET
- Candidate path searching: search for paths consistent with patterns, follow the time-order and length constraints



Experiment: Streaming Evaluation



ROC curve w.r.t snapshot update period