Towards a Timely Causality Analysis for Enterprise Security

Yushan Liu∗, Mu Zhang†, Ding Li‡, Kangkook Jee‡,
Zhichun Li‡, Zhenyu Wu‡, Junghwan Rhee‡, Prateek Mittal∗

∗Princeton University, †Cornell University, ‡NEC Labs America

Outline

- Introduction
- System Overview
  - Reference Model
  - Priority Score
- Evaluations
- Conclusion

Outline

- [x] **Introduction**
- [ ] System Overview
  - Reference Model
  - Priority Score
- [ ] Evaluations
- [ ] Conclusion

Advanced Persistent Threat (APT)

HOME SEARCH The New York Times

A Home Depot in New Orleans. Andrew Burton/Getty Images

Home Depot

September 2014

Home Depot said about 56 million payment cards were probably compromised in an attack that ran from April through September and affected stores in the United States and Canada.

BBC Sign in News Sport Weather Shop Earth Travel

NEWS

Home Video World US & Canada UK Business Tech Science Magazine

Sony cyber-attack: North Korea faces new US sanctions

The US has imposed new sanctions on North Korea in response to a cyber-attack against Sony Pictures Entertainment.

President Barack Obama signed an executive order on Friday allowing sanctions on three North Korean organisations and 10 individuals.

The White House said the move was a response to North Korea's "provocative, destabilising, and repressive actions".

Richard Forrest reports.

03 Jan 2015 | US & Canada

REUTERS World Business Markets Politics TV

#BUSINESS NEWS DECEMBER 18, 2013 / 7:05 PM / 4 YEARS AGO

Target cyber breach hits 40 million payment cards at holiday peak

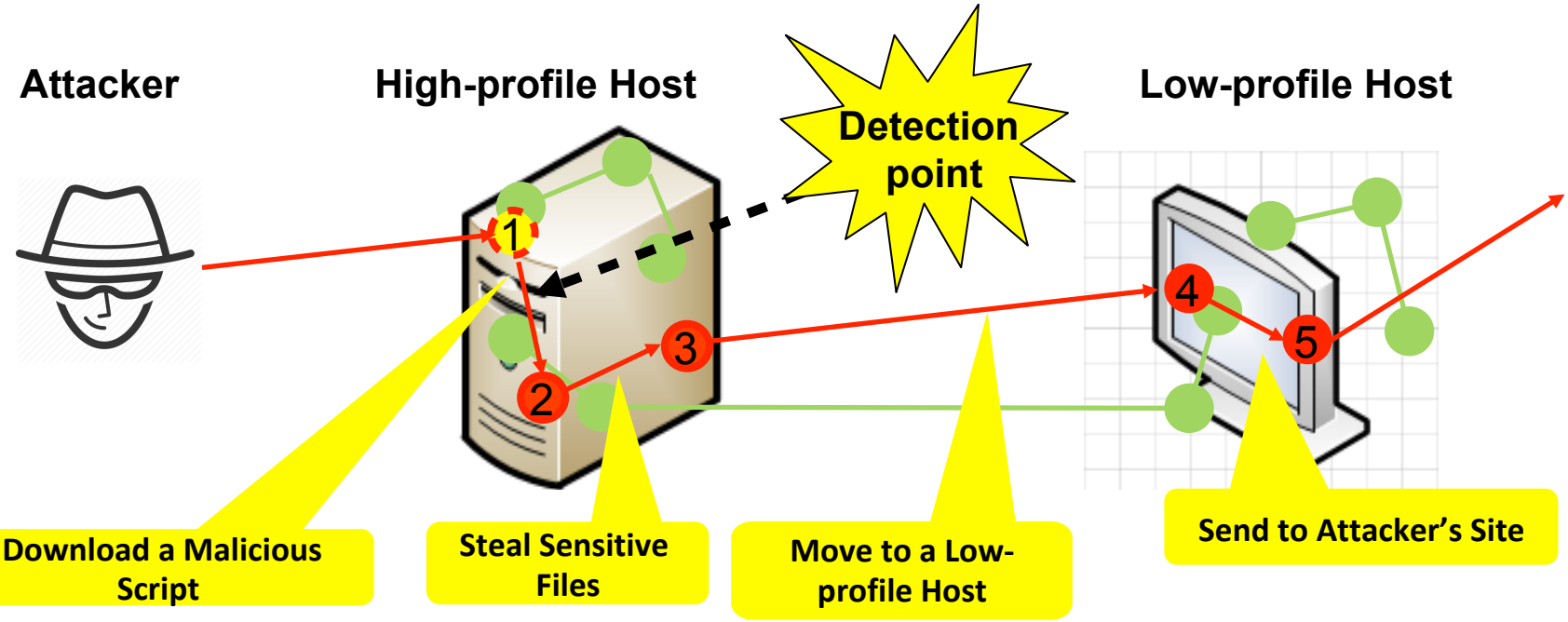
Jim Finkle, Dhanya Skariachan 8 MIN READ

BOSTON (Reuters) - Target Corp said hackers have stolen data from up to 40 million credit and debit cards of shoppers who visited its stores during the first three weeks of the holiday season in the second-largest such breach reported by a U.S. retailer.

Merchandise baskets are lined up outside a Target department store in Palm Coast, Florida, ... 9, 2013. REUTERS/Larry Downing

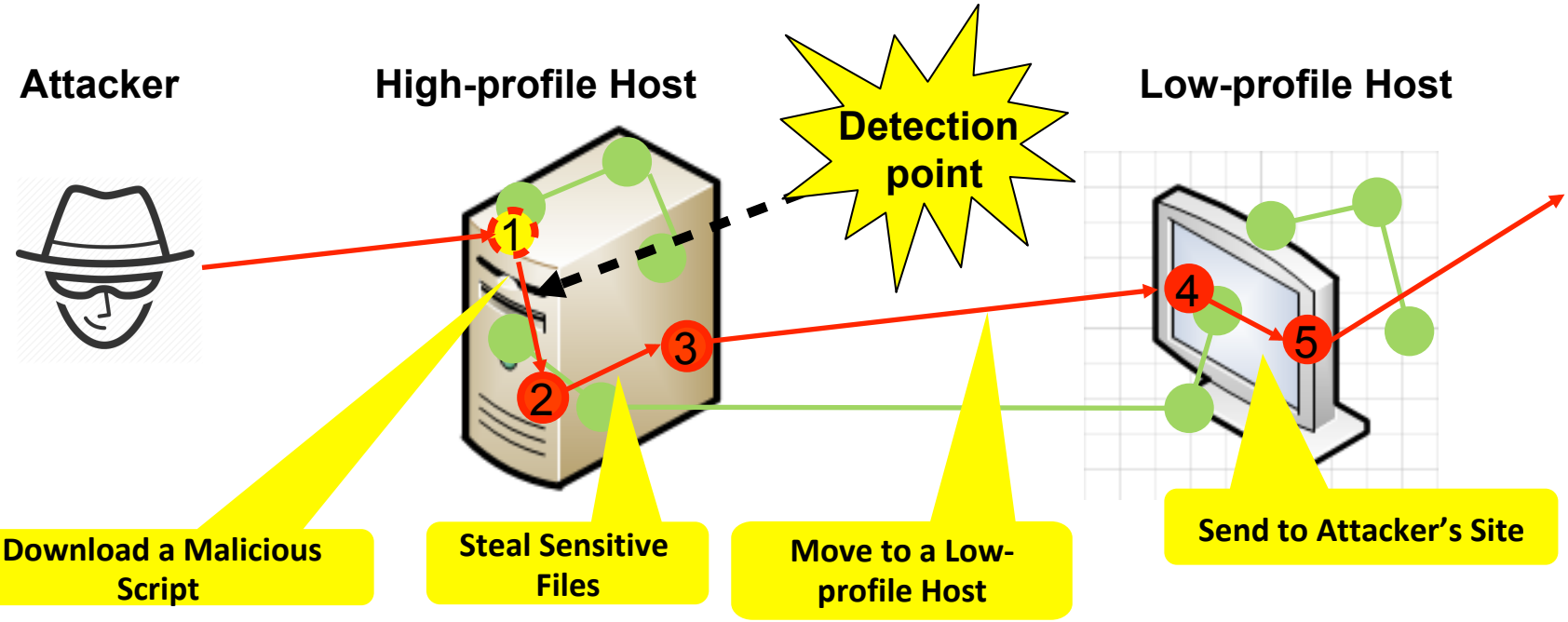
- **6,000 severe incidents reported in the past decade**
- **conducted in multiple stages, stealthy**

Example of an ATP attack: Insider Data Theft



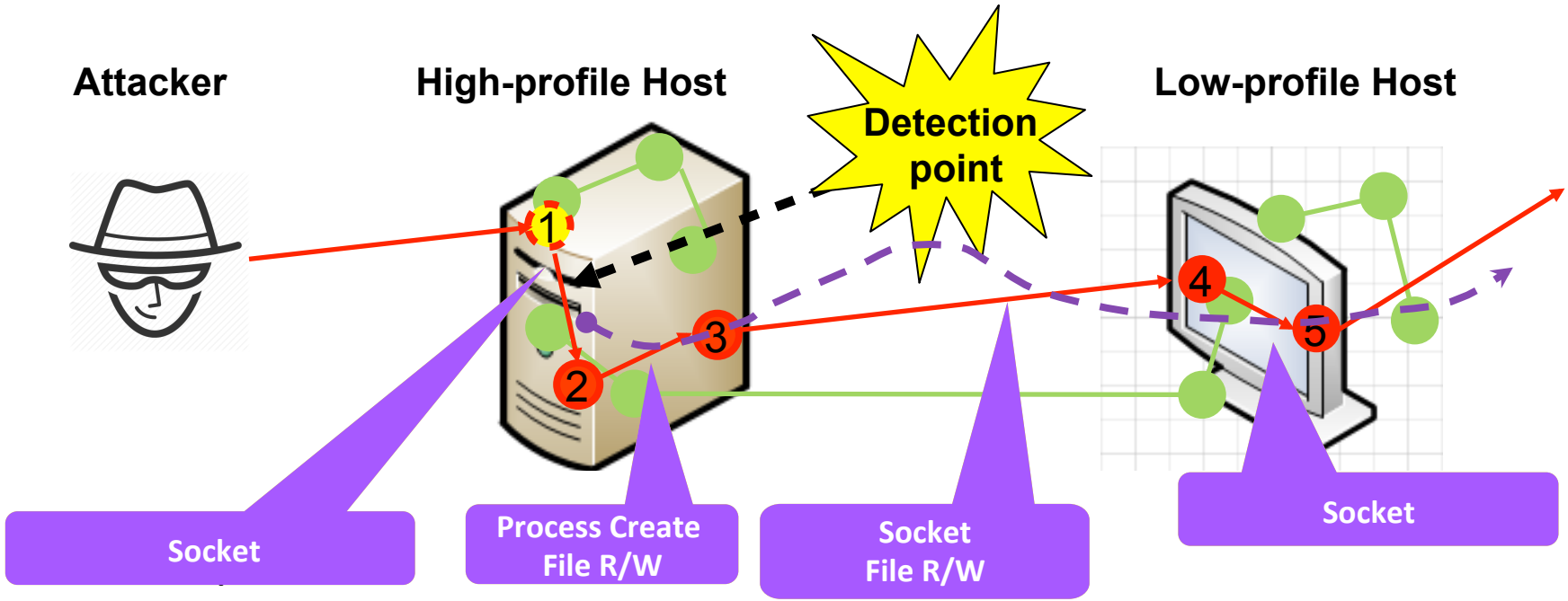
- An intrusion may be detected at one of the stages
  - detection only reveals a small portion of attack traces
  - many individual footprints are seemingly insignificant and keep undetected

To Connect the Dots: Attack Causality Analysis



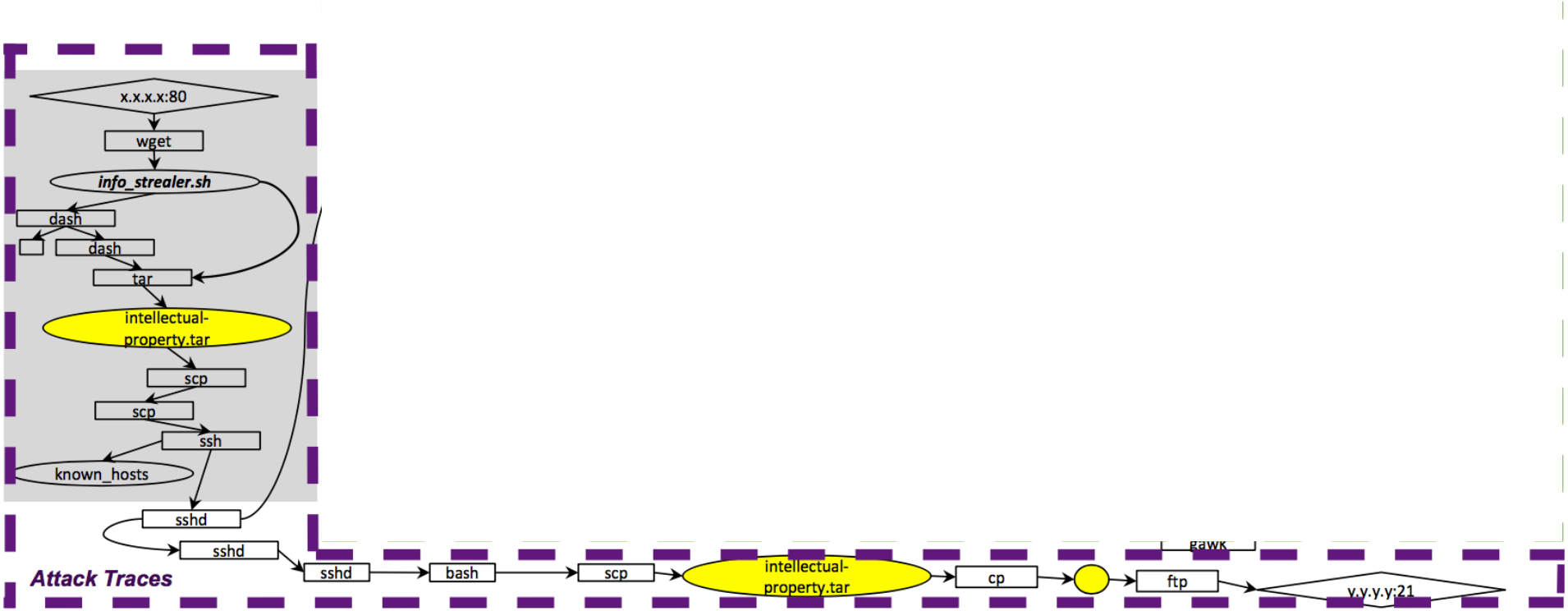
- To achieve a thorough understanding of the detected attack
- Perform attack causality analysis
  - discover all attack traces (provenances & consequences)

Attack Causality Analysis: Use Audit Logs



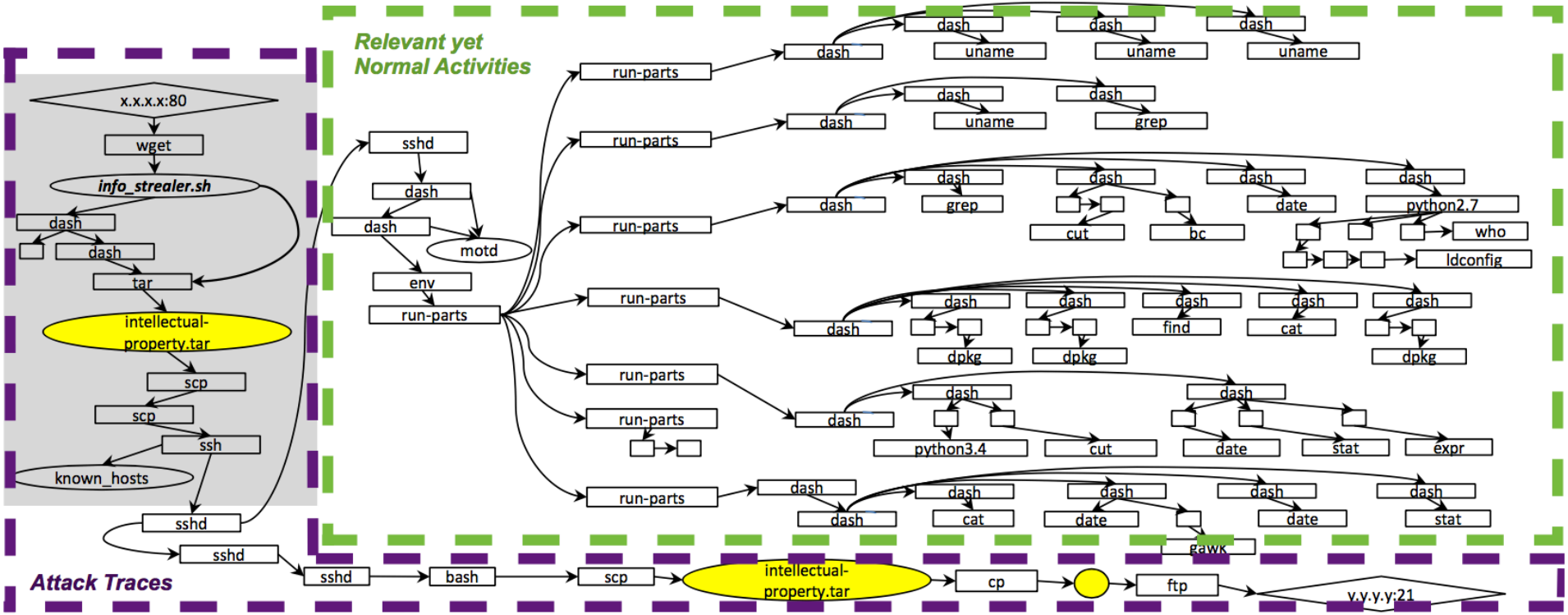
- Reconstruct multi-hop causal dependencies
  - between system objects: processes, files and sockets
  - cross-host tracking: reverse IP mapping

Forward Tracking Graph



- Ideally, only attack events are included

Forward Tracking Graph w. Noise



- Ideally, only attack events are included
- Numerous noises introduced due to system routines

Attack Causality Analysis is Time-sensitive

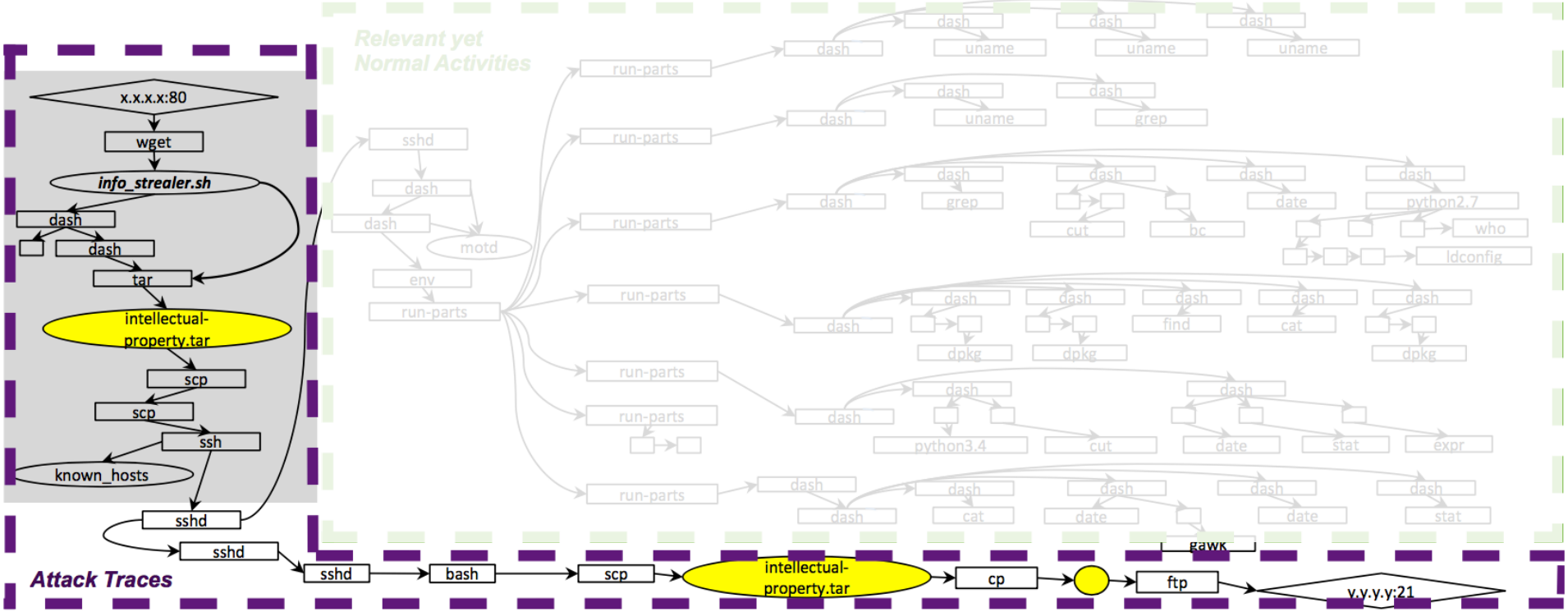
- Increase system downtime
  - systems require complete cleanup before returning to normal operation
  - grow to millions of dollars
- Lose chance to prevent future attacks
  - intrusion may further develop
  - need to take prompt responses

Timely Causality Analysis for Enterprise Security!

Prior Work

- Focus on the dependency explosion problem via *data reduction*
  - eliminate *irrelevant* system dependencies
  - still invest excessive time in analyzing numerous *relevant, yet benign and complex* OS events
- Lack the ability to differentiate unusual activities from common system
  - treat all (abnormal and normal) equally
  - keep track of every relevant relation

Our Solution: Prioritize Abnormal Events



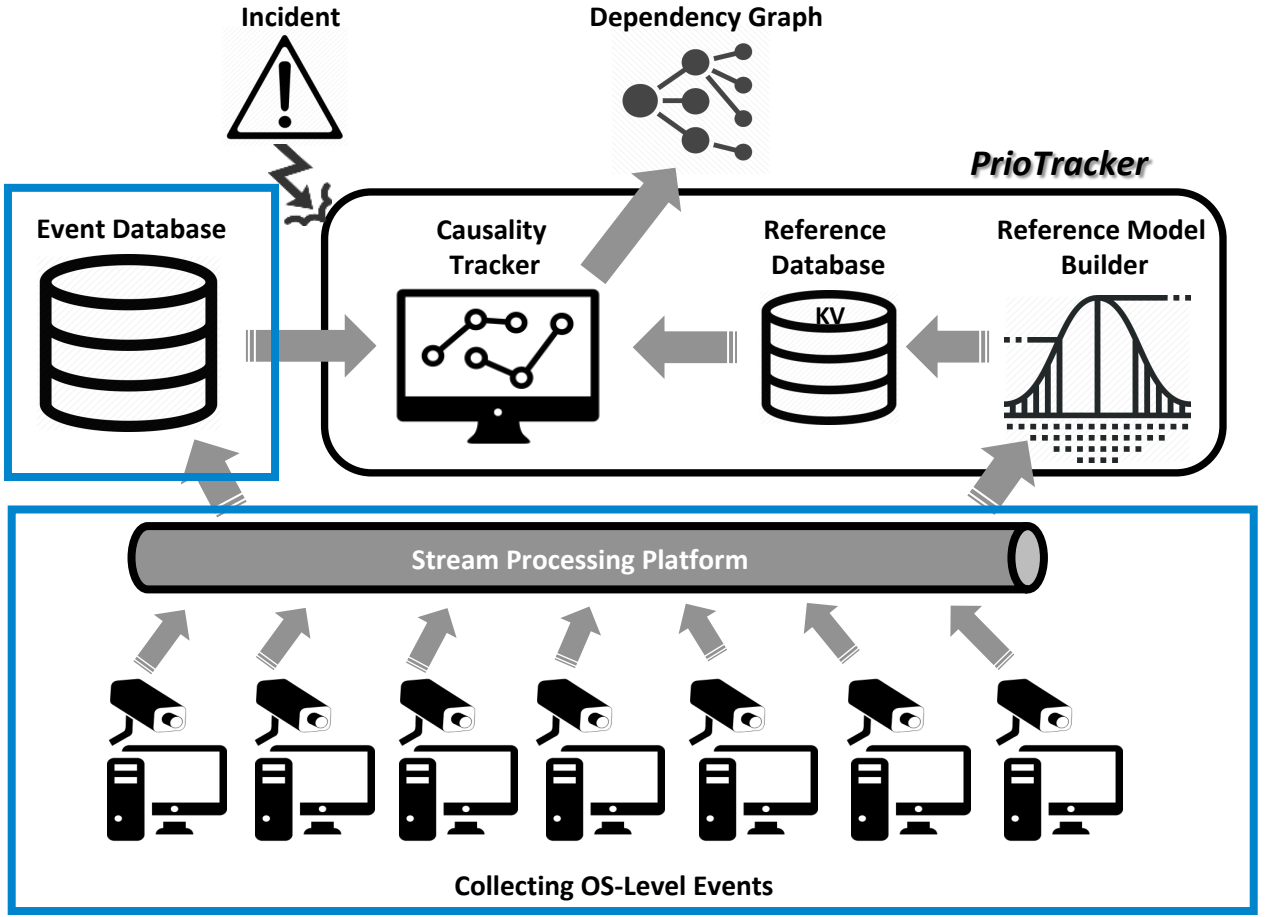
- We designed a causality tracker *PrioTracker*
  - prioritizes the search for abnormal causal dependencies
  - first to introduce priority to graph construction

Outline

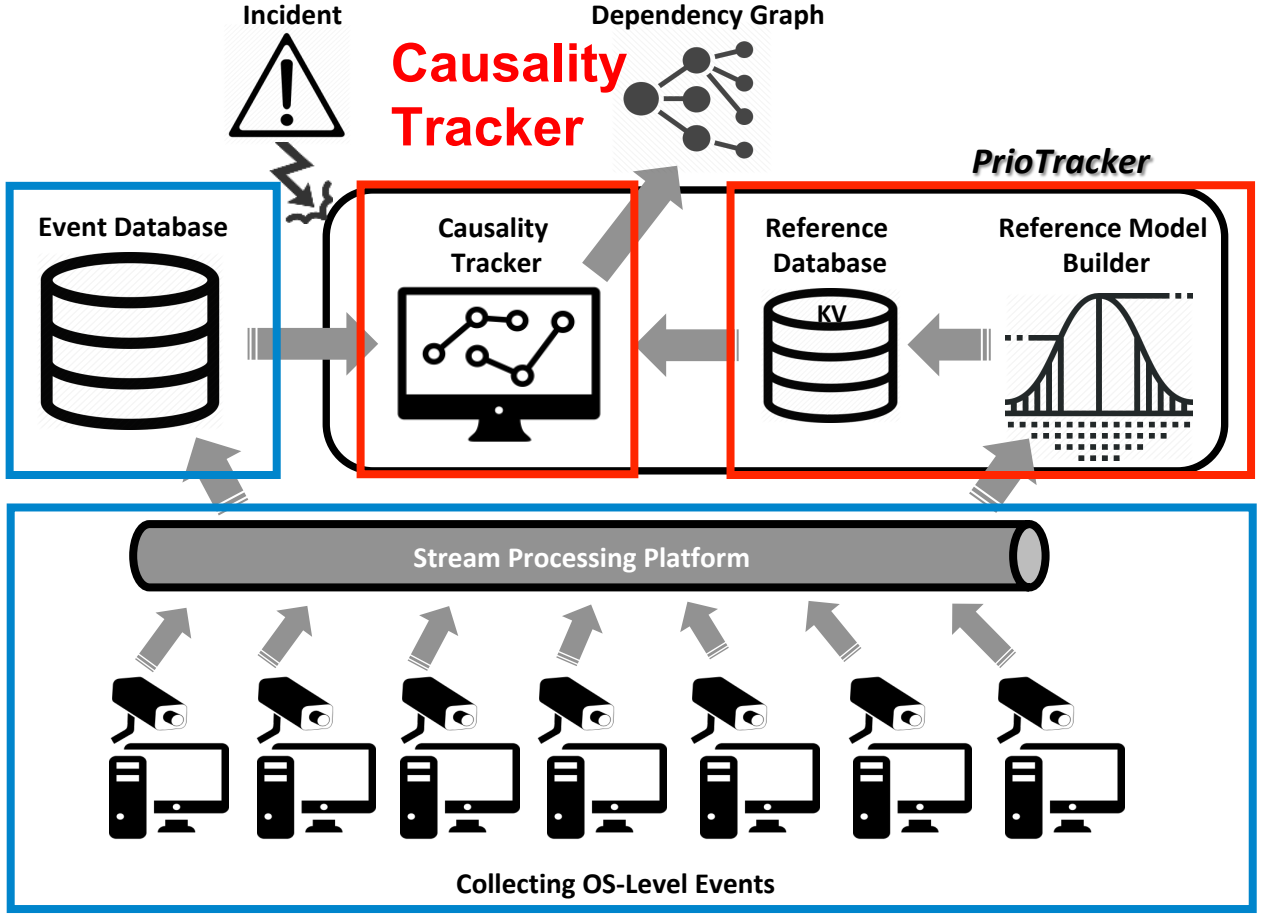
- ☑ Introduction
- ☑ **System Overview**
  - **Reference Model**
  - **Priority Score**
- ☐ Evaluations
- ☐ Conclusion

System Overview: Collection & Storage

- **Event Storage**
  - 1TB of 2.5 billion events collected in one week
  - largest dataset
  - a centralized global view
  - support cross-host tracking
- **Data Collection**
  - 150 hosts in an enterprise



System Overview: PrioTracker



Reference Model

Java:

PrioTracker

20K LOC

Ref Model

10K LOC

Reference Model

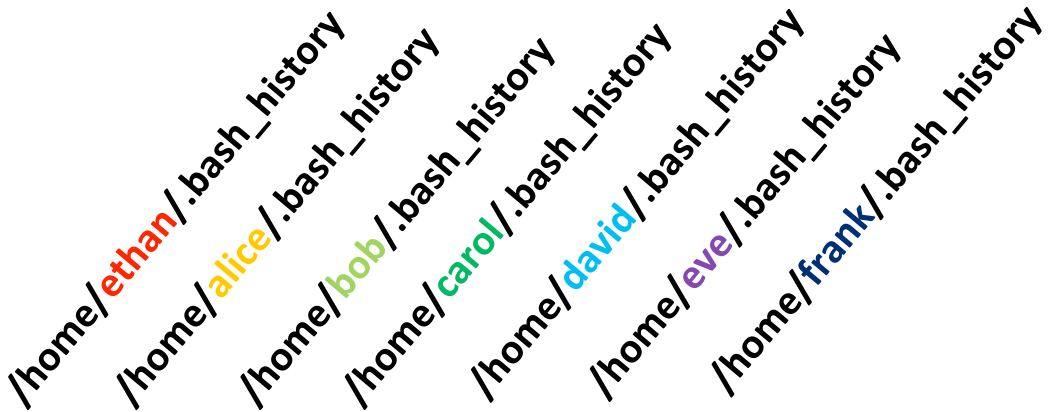
- Record common routine activities across a group of homogeneous hosts in corporate computer systems
- Output a rareness score
  - count the occurrences of an event
  - the less commonly seen, the higher rareness

env -> run-parts ✓ ✗ ✓ ✓ ✓ ✓ ✗ Normal

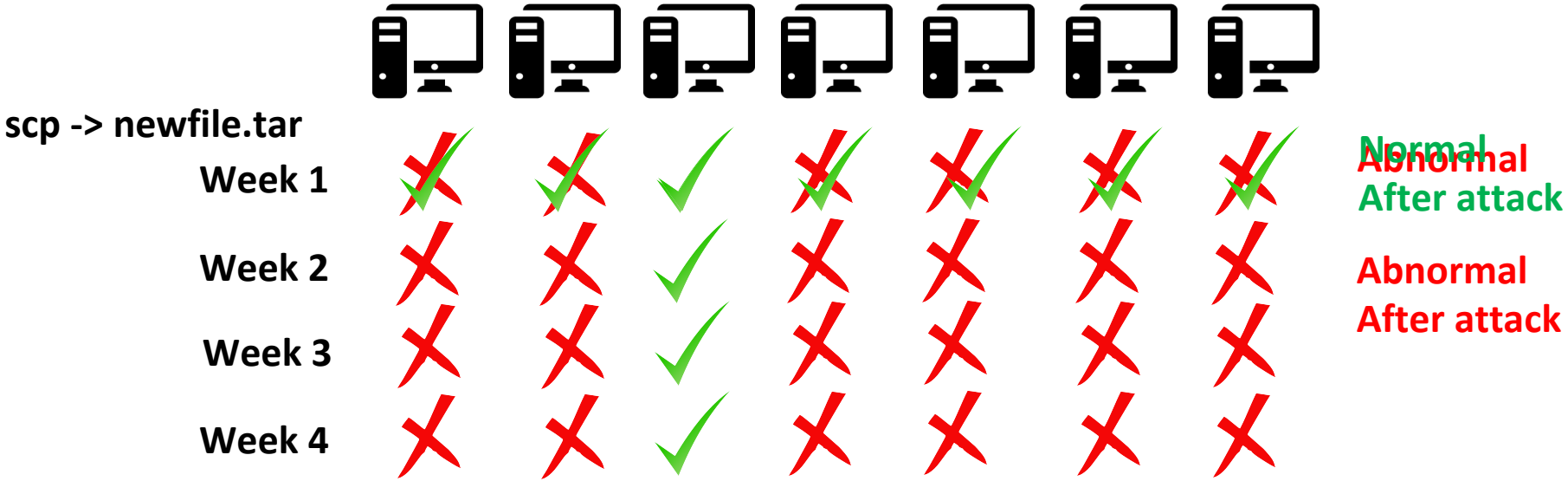
scp -> newfile.tar ✗ ✗ ✓ ✗ ✗ ✗ ✗ Abnormal

Reference Model: Generality

- **Generality**: how to match OS events that can be diverse across hosts
  - event abstraction
    - process: =executable path, file: =path name
    - socket: =remote IP address + remote port number
  - path normalization

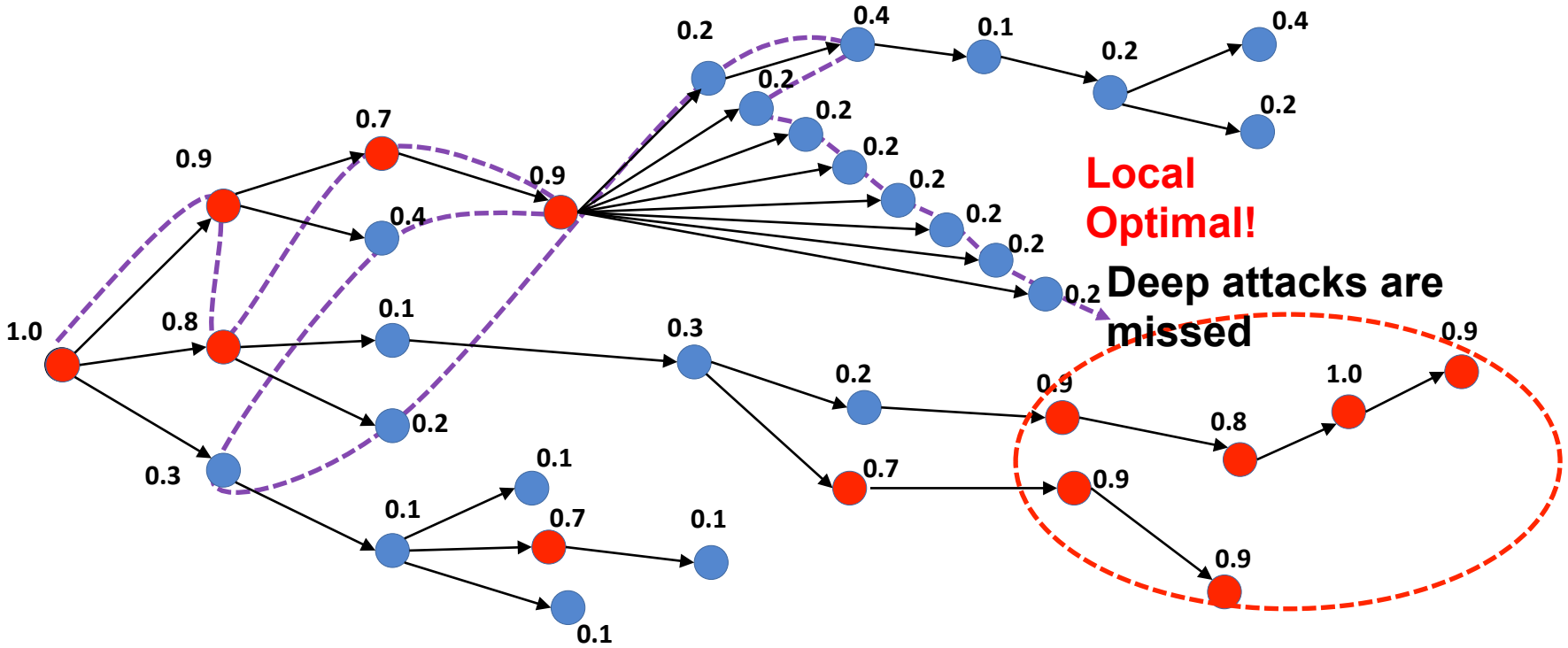


Reference Model: Robustness



- **Robustness**: poisoning attacks
  - attacks compromise machines
  - repeated observations of malicious behavior on the same host will only be considered once

Priority-based Causality Tracking



- select the most uncommon event at each step?

Global Optimization Problem

- Formalize ‘timely’
  - Goal: track the maximum unusual events within a time limit

- Several Factors

Rareness Score
- less common, higher score
- reflect the local rareness

Fanout Score
- lower fanout, higher score
- expand the search area by fast exploring low-fanout events

…

- Priority Score = α × rareness score + β × fanout score

Hill Climbing Algorithm!

- Trade-off between analysis coverage and time effectiveness

Outline

- [x] Introduction
- [x] System Overview
  - Reference Model
  - Priority Score
- [x] **Evaluations**
- [ ] Conclusion

Experiment Setup

- 54 Linux and 96 Windows in an enterprise
- Training for weights in priority function
  - 1,113 random starting points that lead to big graphs (up to 73,221 edges)
  - time limit = 60 min
- Test
  - 1TB of 2.5 billion events collected from 150 hosts in one week
  - time limit = 1 day
- Eight representative attack cases
  - consider noise interleaved with attack traces due to program logic, shared files and

Results: FNR (accuracy)

- **We can capture the attack traces missed by existing trackers**

Attack Case	Critical Events	Rare CE	Baseline			Prio		
			CE	All	FNR	CE	All	FNR
Data Theft	13	12	13	297	0%	13	297	0%
Phishing Email	148	148	148	3282	0%	148	3282	0%
Shellshock	25	23	25	11252	0%	25	11262	0%
Netcat Backdoor	14	14	14	1355	0%	14	1361	0%
Cheating Student	37	33	14	7526	62%	37	7201	0%
Illegal Storage	12	10	12	8048	0%	12	8201	0%
wget-gcc	25	23	25	6415	0%	25	6742	0%
passwd-gzip-scp	15	11	9	2718	40%	15	2364	0%

Baseline:
- **62% FNR** for Chearting Student
- **40% FNR** for wget-gcc

PrioTracker:
- **0% FNR** for both

Results: Time Effectiveness

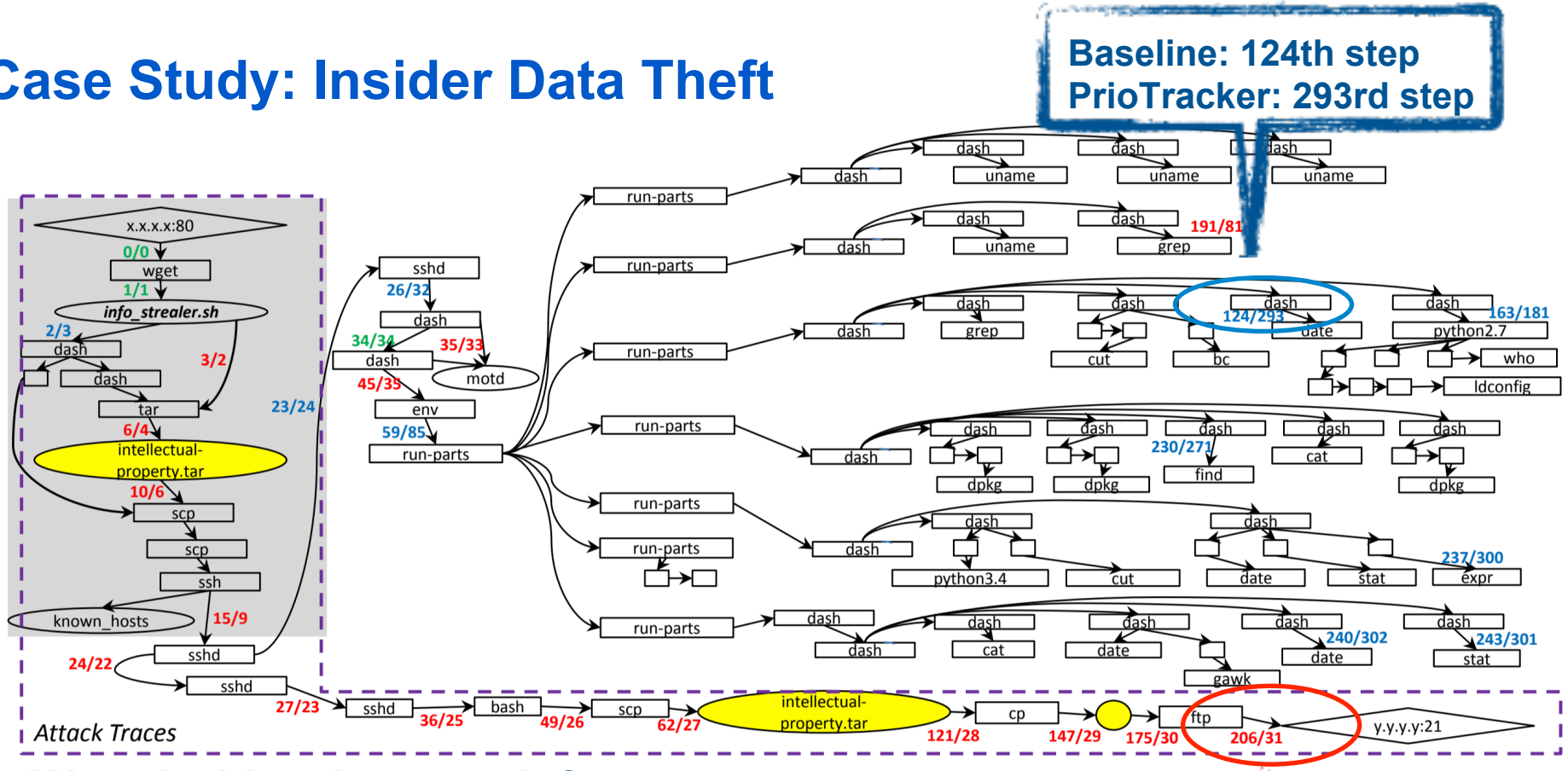
- We can reduce the analysis time by up to two orders of magnitude

Attack Case	Runtime (100% CEs)	
	Baseline	Prio
Data Theft	16.76s	2.62s
Phishing Email	1m2s	1m4s
Shellshock	2m3s	12.08s
Netcat Backdoor	8.83s	1.28s
Cheating Student	> 1d	40m21s
Illegal Storage	27m51s	14m10s
wget-gcc	42m9s	6m23s
passwd-gzip-scp	> 1d	1m24s

Baseline:
- **> one day** for both

PrioTracker:
- **40m** for Cheating Student
- **6m** for wget-gcc

Case Study: Insider Data Theft



- We prioritize the search for abnormal events over routine Linux maintenance

Outline

- ☑ Introduction
- ☑ System Overview
  - Reference Model
  - Priority Score
- ☑ Evaluations
- ☑ **Conclusion**

Conclusion

- First to formalize timely attack causality analysis and to introduce priority to *attack graph construction*.
- Implemented PRIOTRACKER that computes priority based on its rareness and topological characteristics in the causality graph
- Built a reference model to differentiate unusual behaviors from normal ones
- Evaluated on 1TB of 2.5 billion events from 150 hosts
- Captured attack traces that are missed by existing trackers and reduced the analysis time by up to two orders of magnitude

Thank you!

Q & A

Discussion: Evasion Using High-fanout Events

- Attacks cannot be launched solely using dependencies with big fanout
  - e.g, apache —> bash is an essential low-fanout step
  - Other attack edges can be discovered from thousands of benign edges in a faster fashion
- Fast-tracking benign events with low fanout only incurs a small delay
  - processing benign dependencies with huge fanout (up to tens of thousands) can be time consuming

Discussion: Distributed Causality Tracker

- Construction of causality graphs can be potentially parallelized with distributed computing
  - any individual branch to be explored can be processed separately
  - branches may bear different priorities and therefore are assigned with corresponding computing resources
  - dependencies on each host can also be pre-computed in parallel
  - cross-host tracking thus becomes the concatenation of multiple generated graph
- Massive and pervasive dependencies among system events bring significant challenges

Attack Cases

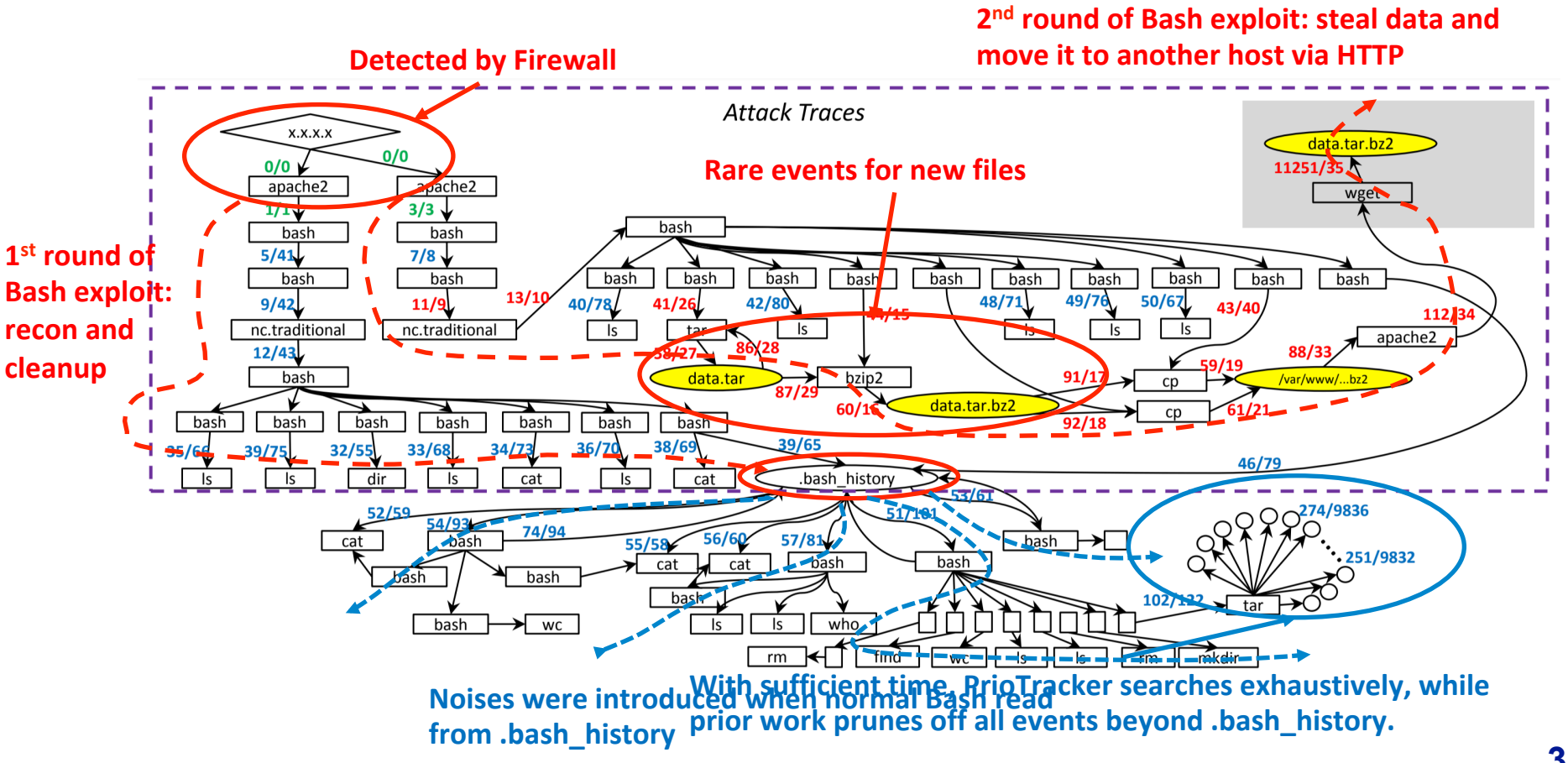
Attack Case	Description of Scenario
Data Theft	An insider stole sensitive intellectual property, secure-copied the data to a low-profile machine and then leaked it via Internet.
Phishing Email	A malicious Trojan was downloaded as an Outlook attachment and the enclosed macro was triggered by Excel to create a fake "java.exe", and the malicious "java.exe" further SQL exploited a vulnerable server to start cmd.exe in order to create an info-stealer.
Shellshock	An attacker utilized an Apache server to trigger the Shellshock vulnerability in Bash multiple times. In each round, she started several Linux commands and cleared Bash history in the end.
Netcat Backdoor	An attack downloaded the netcat utility and used it to open a Backdoor, from which a port scanner was then downloaded and executed.
Cheating Student	A student downloaded midterm scores from Apache, and uploaded a modified version.
Illegal Storage	A server administrator created a directory under another user's home directory, and downloaded the illegal files to the directory.
wget-gcc	Malicious source files were downloaded and then compiled.
passwd-gzip-scp	An attack stole user account information from passwd file, compressed it using gzip and transferred the data to a remote machine.

Examples of Noises (Normal Activities)

- **Normal activities connect the malicious activities to benign ones and cause graph explosion**

Noise Source	Activity	Reason to Interleave	Description of Scenario
sshd-run-parts	Cascade Forking	Program Behavior	Upon receiving file transferring request, SSH daemon starts a large number of routine processes to update messages (e.g., motd) used for user interaction.
sshd-ypserv	Cascade Forking	Conditional Program Behavior	When a global account requests a SSH connection, SSHD checks user credential from directory service (i.e., NIS) via ypserv, which forks tons of child processes.
.bash_history	Multiple Reads	Shared Log File	Once an attacker has cleared the .bash_history to erase her attack footprints, the same history file will further get read by all future benign Bash terminals.
Explorer	Multiple Writes	Shared GUI Program	Once an attacker has dropped some malware to a local directory, the file Properties are viewed in Explorer by a normal user, who later uncompresses a ZIP file and therefore creates many files using 7ZIP from the same Explorer.

Case Study: Shellshock

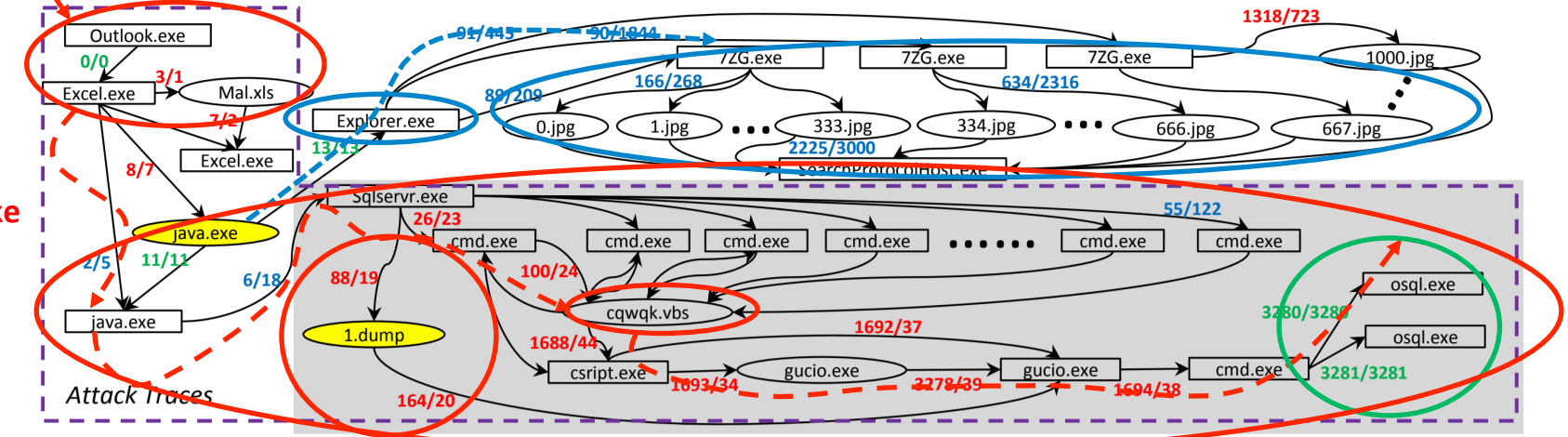


Case Study: Phishing Email

Malicious Attachment Detected by Antivirus

Noise was introduced by Explorer.exe, which further ran 7ZG.exe that wrote to massive amount of files

ProTracker explores the creation of .vbs malware and database dump first due to their rareness.



Start fake java.exe

Database SQL server already been discovered at 18th iteration. ProTracker was not able to prioritize osql.exe

20th iteration to uncover the connection of attacks gucio.exe, which ran two SQL queries via command line to dump data.

created malicious .vbs

Time Effectiveness: 75 Random POI

- **PrioTracker can always find more rare events than baseline tracker**