Enabling Efficient Cyber Threat Hunting With Cyber Threat Intelligence

Peng Gao[1], Fei Shao[2], Xiaoyuan Liu[1], Xusheng Xiao[2], Zheng Qin[3], Fengyuan Xu[3], Prateek Mittal[4], Sanjeev R. Kulkarni[4], Dawn Song[1]

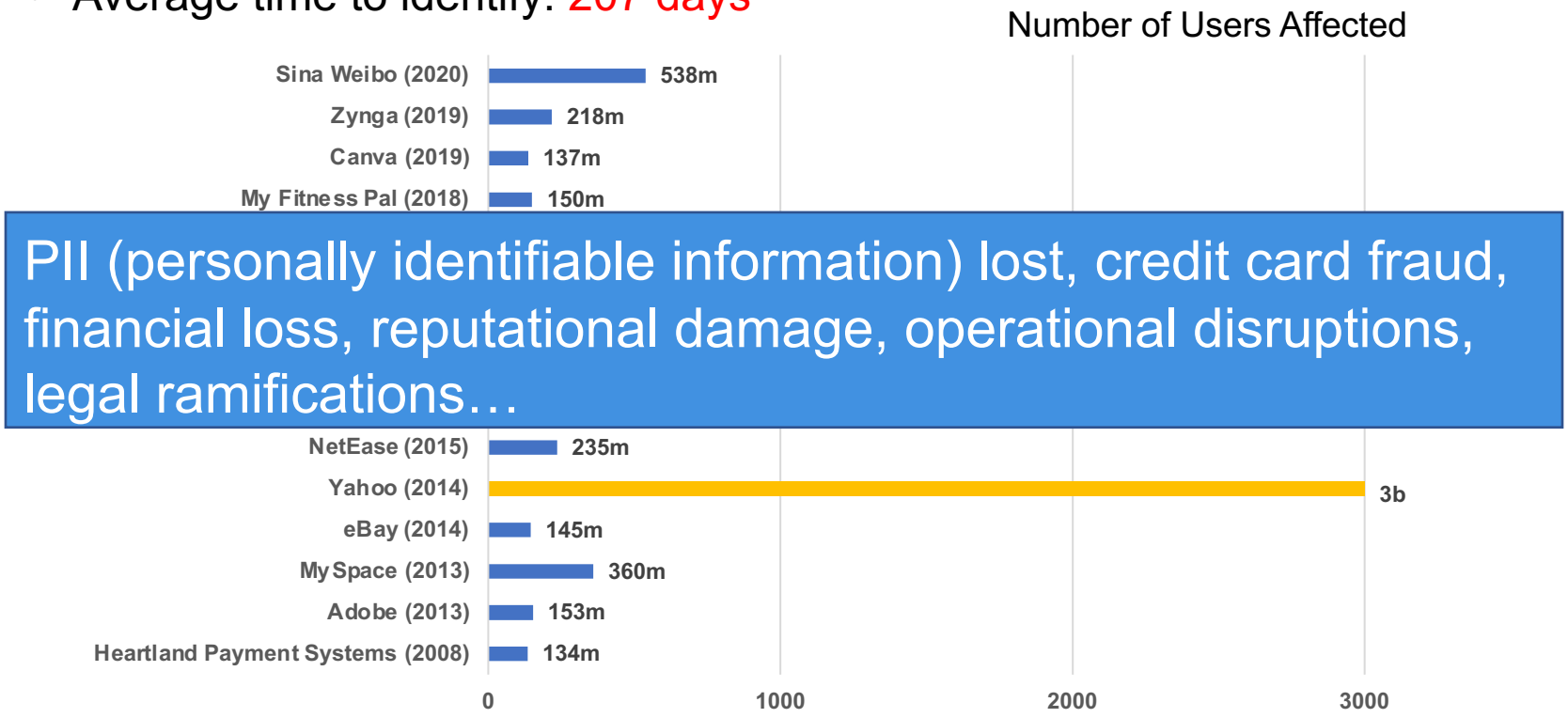
[1]University of California, Berkeley
[2]Case Western Reserve University
[3]Nanjing University
[4]Princeton University

Massive Data Breaches

Biggest Data Breaches of the 21st Century

Statistics (2020):

- Average total cost: 3.86 million
- Average time to identify: 207 days



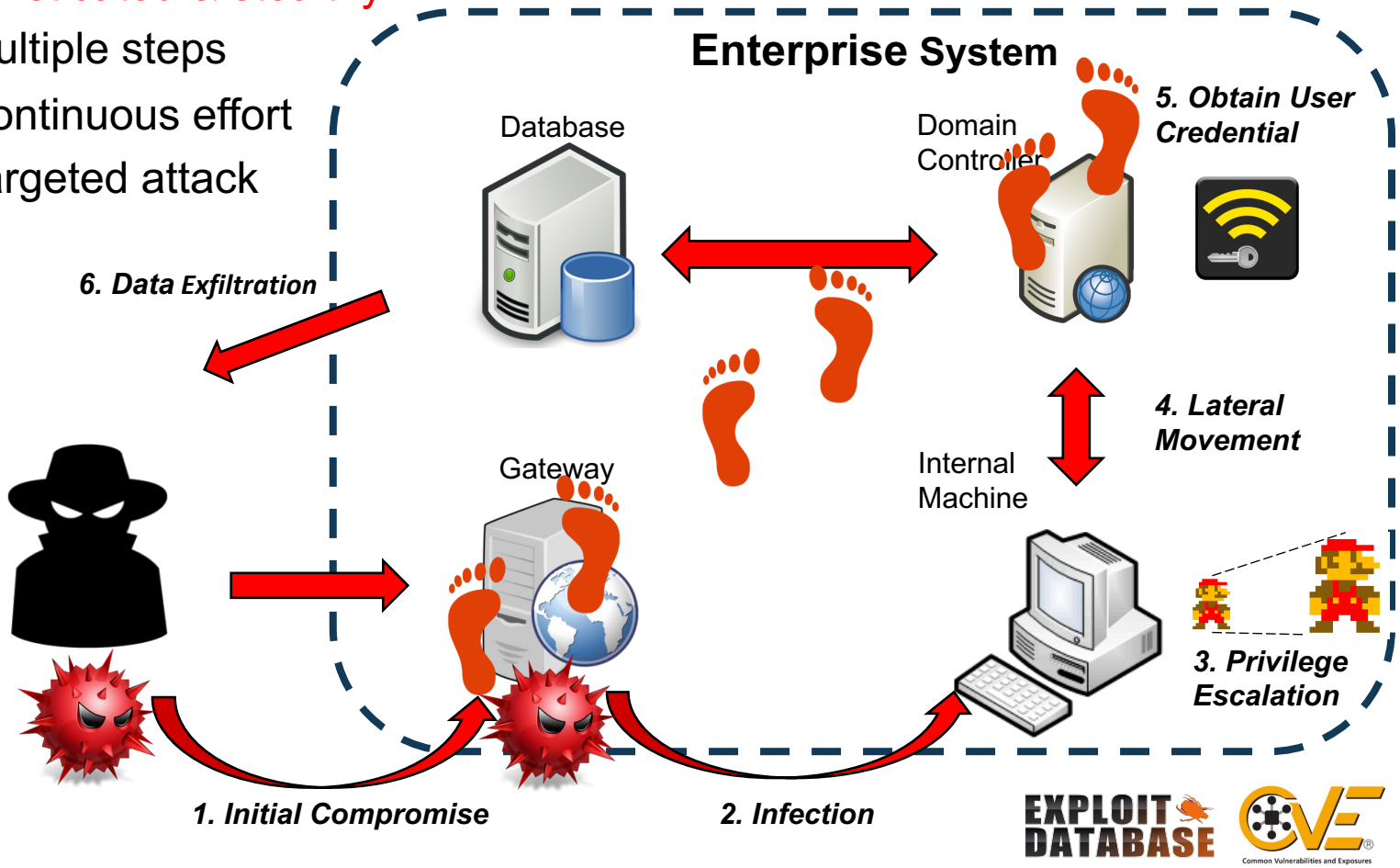
PII (personally identifiable information) lost, credit card fraud, financial loss, reputational damage, operational disruptions, legal ramifications…

Source: csooline.com, IBM Cost of a Data Breach Report 2020

Attack Behind the Data Breaches: Advanced Persistent Threat (APT)

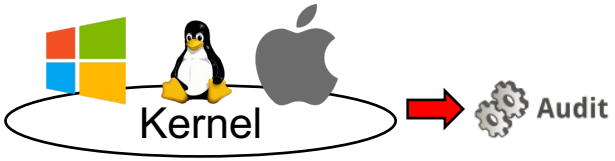
Sophisticated & stealthy

- Multiple steps
- Continuous effort
- Targeted attack



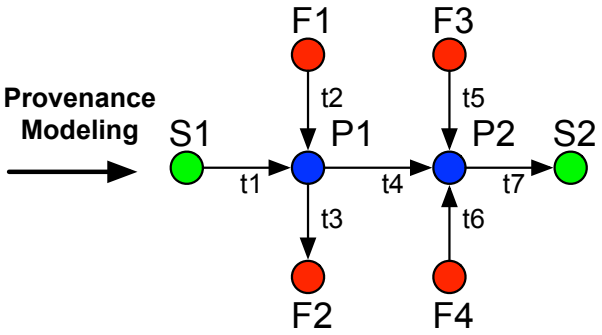
Transparent Computing Through Ubiquitous System Auditing

- Ubiquitous system auditing
  - Monitor every host
  - Monitor system activities through selective system calls
    - File access, process creation, network access



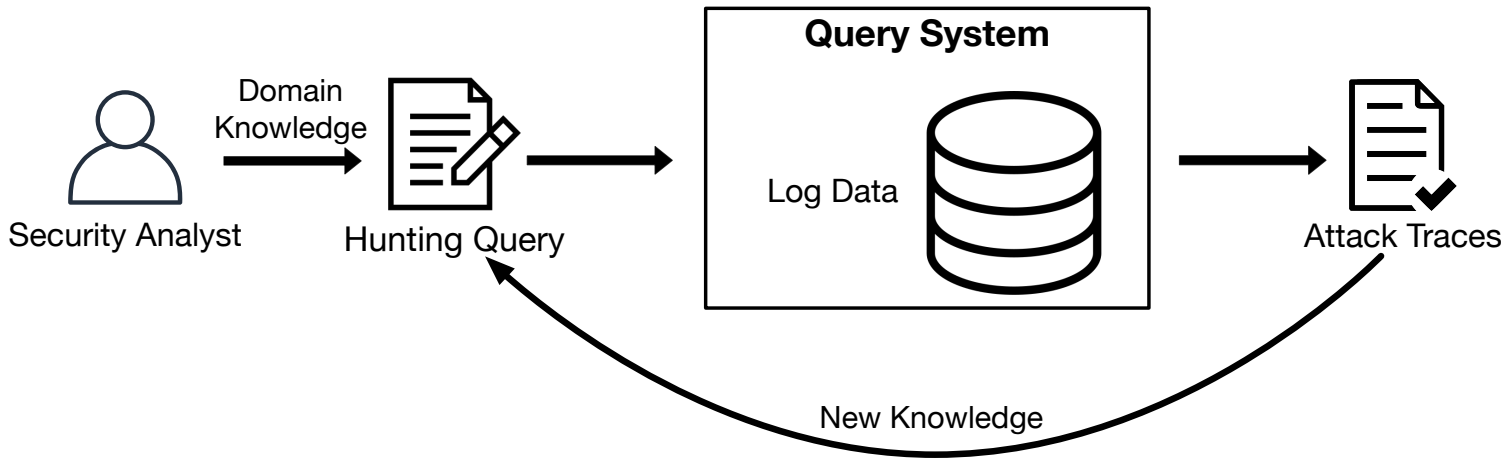
- System audit logs => system provenance graph
  - System entities: files, processes, network sockets
  - System event, <subject_entity, action, object_entity>
  - Global view

t1: Process P1 read Socket S1
t2: Process P1 read File F1
t3: Process P1 write File F2
t4: Process P1 exec Process P2
t5: Process P2 read File F3
t6: Process P2 read File F4
t7: Process P2 write Socket S2



Cyber Threat Hunting via Querying System Audit Logs

- Cyber threat hunting



- Limitations:
  - Manual query construction => labor-intensive
  - What to search for? => cold start

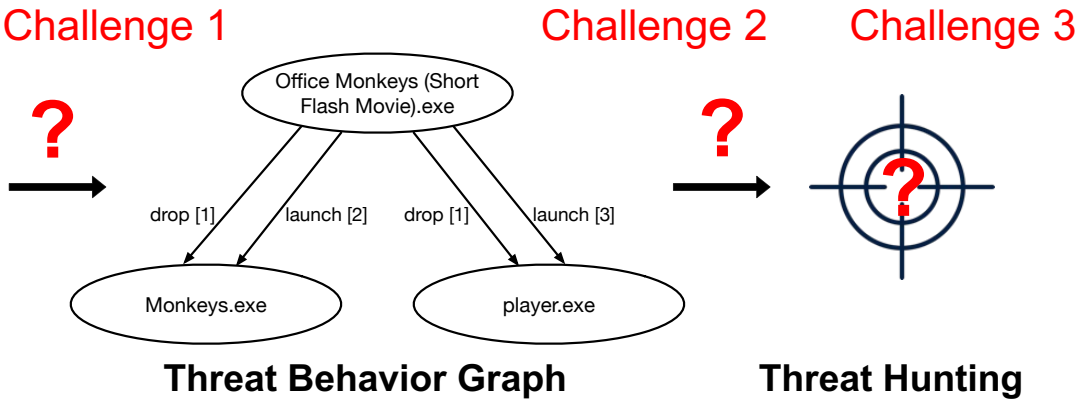
Threat Intelligence

- Threat intelligence: rich knowledge about threats
  - IOCs (Indicators of Compromise) and IOC relations
    - Malware signatures, malicious file/process names, IPs

The CozyDuke actor spearphishes a targeted victim with e-mails containing a link to a hacked website hosting a ZIP file. The victim clicks on the link and downloads the ZIP file. After being downloaded, the ZIP file self-extracts to Office Monkeys (Short Flash Movie).exe. This file in turn drops two executables: Monkeys.exe and player.exe. It first launches Monkeys.exe, a decoy playing a self-contained, very funny video of white-collar tie wearing chimpanzees working in a high rise office with a human colleague. It then launches player.exe, a CozyDuke dropper maintaining anti-detection techniques.

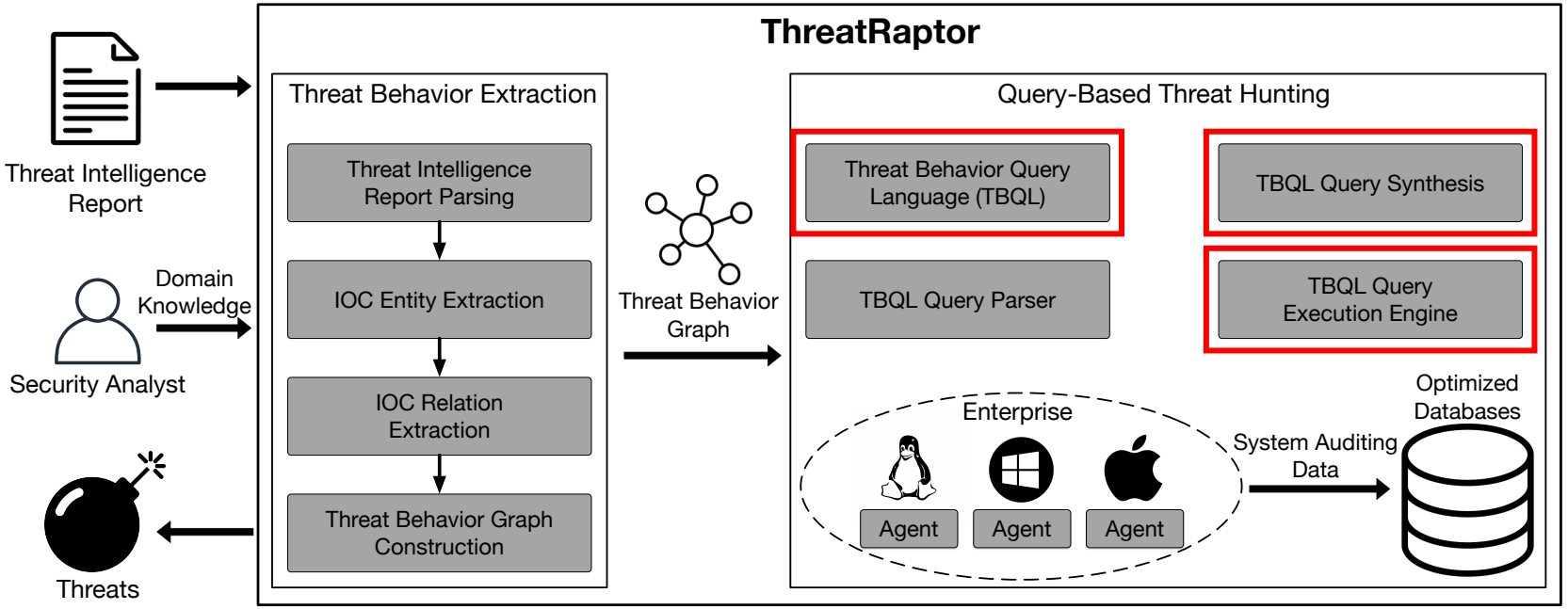
Threat Intelligence Report Snippet

Source: securelist.com



ThreatRaptor: Automated Threat Hunting Using Threat Intelligence

- A system for automated threat hunting using threat intelligence



https://youtu.be/SrcTDQwRF_M

Threat Behavior Extraction: Main Ideas

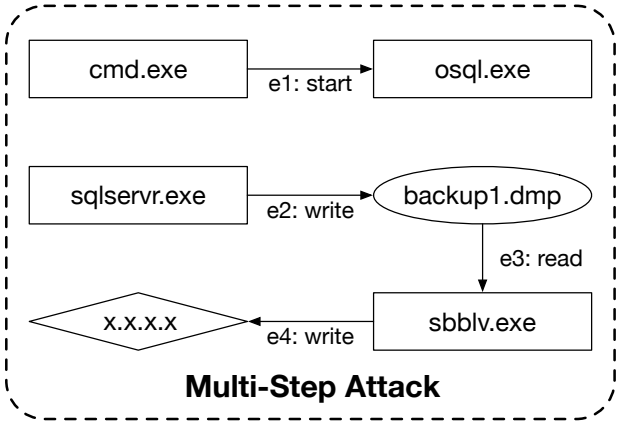
- Extracting IOCs and IOC relations
- Challenges:
  - Accuracy: massive nuances in security text
  - Efficiency: timely threat hunting
- Unsupervised, light-weight, accurate NLP pipeline
  - Regex rules for IOC extraction
  - IOC protection: replacing IOCs with a dummy word
  - Dependency-parsing-based IOC relation extraction
- Outperforming general information extraction approaches (Stanford Open IE, Open IE 5)

Query-Based Threat Hunting

- Leveraging our prior work
  - Gao et al. “AIQL: Enabling Efficient Attack Investigation from System Monitoring Data.” USENIX ATC, 2018.
- Data collection & storage
  - PostgreSQL, Neo4j
- Threat Behavior Query Language (TBQL)
  - Declarative, expressive, easy to write
- Efficient query execution engine
  - Leveraging domain-specific optimizations

TBQL Event Pattern Syntax

- Global constraint
- Event: <subject, action, object>
  - <subject>: process
  - <object>: process, file, network socket
  - Attribute filter
  - Boolean operator: proc p[“%chrome%” && pid=100 read || write file f[“%profile%” && amount=10KB”]
- Event relationship
  - Temporal relationship
  - Attribute relationship
    - Graph dependency: shared entity
- Syntax sugar




```
1 at “mm/dd/yyyy” exe_name = “%cmd.exe”
2 agentid = host(“sql_database_server”)
3 proc p1[“%cmd.exe”] start proc p2[“%osql.exe”] as evt1
4 proc p3[“%sqlservr.exe”] write file f1[“%backup1.dmp”] as evt2
5 proc p4[“%sbblv.exe”] read file f1 as evt3 name = “%backup1.dmp”
6 proc p4 write ip i1[“x.x.x.x”] as evt4 dst_ip = “x.x.x.x”
7 with evt1 before evt2, evt2 before evt3, evt3 before evt4
8 return distinct p1, p2, f1, p4, i1
```

p1.exe_name, p2.exe_name, p3.exe_name, f1.name, p4.exe_name, i1.dst_ip

Query-Based Threat Hunting: New Challenges & Solutions

- Query synthesis mechanism
- Challenge 1: an edge in threat behavior graph <=> a path of system events
  - New query language syntax: variable-length paths

```
1 proc p1 ~>[read] ip i1 as evt1
2 proc p2 ~>(2~5)[write] file f1 as evt2
```

- Challenge 2: imprecision in threat intelligence text
  - New query execution mode: fuzzy search
    - Based on inexact graph pattern matching

Conclusion

- ThreatRaptor: automated threat hunting using threat intelligence
  - Threat behavior extraction
  - Query-based threat hunting
    - Threat Behavior Query Language (TBQL)
  - Query synthesis

Q & A

Thanks!