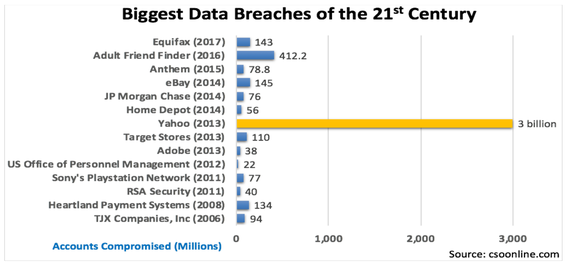
SAQL: A Stream-based Query System for Real-Time Abnormal System Behavior Detection

Peng Gao[1], Xusheng Xiao[2], Ding Li[3], Zhichun Li[3], Kangkook Jee[3], Zhenyu Wu[3], Chung Hwan Kim[3], Sanjeev R. Kulkarni[1], Prateek Mittal[1]

[1] PRINCETON UNIVERSITY [2] CASE WESTERN RESERVE UNIVERSITY [3] NEC Laboratories America
Relentless passion for innovation

Impact of Advanced Persistent Threat (APT) Attacks

APT attacks have plagued many well-protected businesses with significant financial losses.

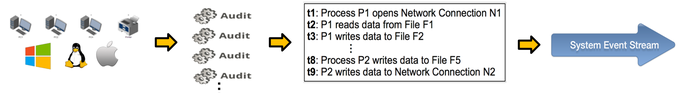


- **Advanced**: sophisticated techniques exploiting multiple vulnerabilities
- **Persistent**: adversaries are continuously observing and stealing data from the target
- **Threat**: strong economical or political motives

Effectively Detecting APT Attacks in Real-Time

Ubiquitous system monitoring

- Recording system behaviors from kernel as system events (<subject, operation, object>, e.g., **proc p read file f**)
- Unified structure of audit logs (not bound to applications), presenting a global view of system behaviors



Approach: timely anomaly detection via **querying** the real-time stream of system monitoring data

Challenges: finding various damaging needles in very large haystacks

- How to incorporate expert knowledge effectively? => **expressive** and **concise** **domain-specific language**
- How to analyze "big data" (~50GB for 100 hosts per day) efficiently? => **efficient** **query execution engine** with optimizations tailored to the domain characteristics of the data and the semantics of the query

SAQL System Architecture

SAQL: a novel stream-based query system (**50K LOC**) for real-time abnormal system behavior detection (paper accepted in USENIX Security'18)

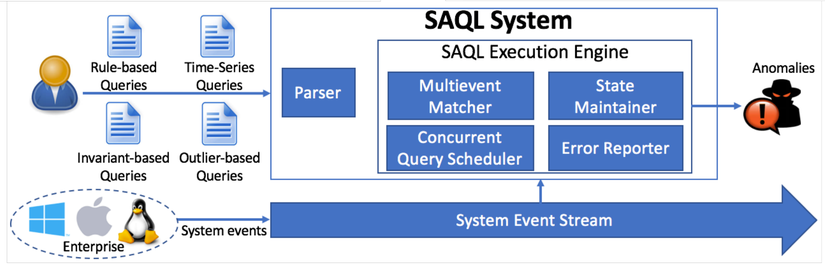
- **Agents** (built on top of auditd, ETW, DTrace) deployed across enterprise hosts to collect critical attributes

Table 1: Representative attributes of system entities

Entity	Attributes
File	Name, Owner/Group, VolID, DataID, etc.
Process	PID, Name, User, Cmd, Binary Signature, etc.
Network Connection	IP, Port, Protocol

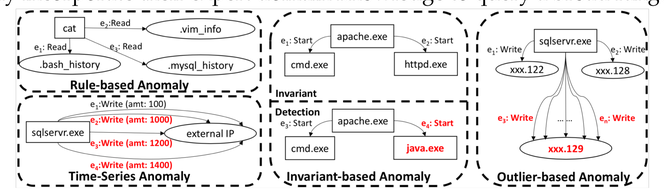
Table 2: Representative attributes of system events

Operation	Read/Write, Execute, Start/End, Rename/Delete
Time/Sequence	Start Time/End Time, Event Sequence
Misc.	Subject ID, Object ID, Failure Code



Stream-based Anomaly Query Language (SAQL) for Expert Knowledge Incorporation

SAQL provides an **expressive** and **concise** domain-specific language that **uniquely** integrates a series of critical primitives, which enables security experts to easily incorporate their expert domain knowledge to query a broad range of attack behaviors.



- Event patterns
- Event attribute constraints
- Event relationship constraints
- Sliding windows
- **Stateful computation**
- Syntax shortcuts

```
agentid = XXX // db server
proc p1["%cmd.exe"] start proc p2["%osql.exe"] as evt1
proc p3["%sqlservr.exe"] write file f1["%backup1.dmp"] as evt2
proc p4["%sbblv.exe"] read file f1 as evt3
proc p4 read || write ip i1[dstip="XXX"] as evt4
with evt1 -> evt2 -> evt3 -> evt4
return p1, p2, p3, f1, p4, i1, evt1.starttime, evt2.starttime, evt3.starttime, evt4.starttime, evt4.amount
```

Rule-based SAQL query

```
agentid = XXX // db server
proc p write ip i as evt #time(10 min)
state[3] ss {
  avg_amount := avg(evt.amount)
} group by p
alert (ss[0].avg_amount > (ss[0].avg_amount + ss[1].avg_amount + ss[2].avg_amount) / 3) && (ss[0].avg_amount > 10000)
return p, ss[0].avg_amount, ss[1].avg_amount, ss[2].avg_amount
```

Time-Series SAQL query

```
proc p1["%apache.exe"] start proc p2 as evt #time(10 s)
state ss {
  set_proc := set(p2.exe_name)
} group by p1
invariant[10][offline] {
  a := empty_set
  a = a union ss.set_proc
}
alert |ss.set_proc diff a| > 0
return p1, ss.set_proc
```

Invariant-based SAQL query

```
agentid = xxx // db server
proc p["%sqlservr.exe"] read || write ip i as evt #time(10 min)
state ss {
  amt := sum(evt.amount)
} group by i.dstip
cluster(points=all(ss.amt), distance="ed", method="DBSCAN(100000, 5)")
alert cluster.outlier && ss.amt > 1000000
return i.dstip, ss.amt
```

Outlier-based SAQL query

SAQL Execution Engine for Timely "Big Data" Security Analytics

SAQL provides an **efficient** query execution engine (built on top of Siddhi CEP) with **novel** optimizations that are tailored to the domain characteristics of the data and the semantics of the query.

Challenge: executing multiple concurrent queries incurs significant performance overhead

Solution: **Master-Dependent-Query scheme** employed in the **concurrent query scheduler** (**Key Insight**: share intermediate execution results among queries)

- Partition concurrent queries into master-dependent groups
- Only master query has direct access to the stream

```
proc p read || write file f["/etc/passwd" || "%.ssh/id_rsa" || "%.bash_history" || "/var/log/wtmp"] as evt #time(1 min)
state ss {
  e1 := count(evt.id)
  e2 := sum(evt.amount)
} group by p
return p, ss.e1, ss.e2
```

Master query

```
proc p read || write file f["/etc/passwd" || "%.ssh/id_rsa" || "%.bash_history" || "/var/log/wtmp"] as evt #time(1 min)
state ss {
  e1 := count(evt.id)
} group by p
return p, ss.e1
```

Dependent query 1

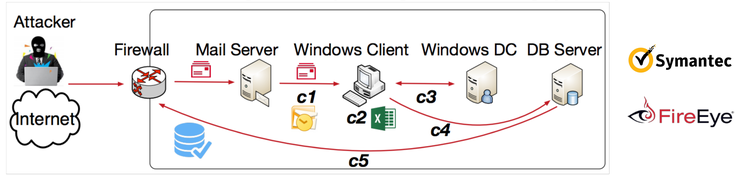
```
proc p read || write file f["/etc/passwd" || "%.ssh/id_rsa" || "%.bash_history" || "/var/log/wtmp"] as evt #time(1 min)
state ss {
  e2 := sum(evt.amount)
} group by p
return p, ss.e2
```

Dependent query 2

Real-World Deployment and Evaluation

We deployed SAQL on 150 hosts of NEC Labs (generating 3750 events/s). We performed **realistic attacks** in the deployed environment and collected 1.1TB of real system monitoring data for evaluation (containing 3.3 billion system events).

- **Evaluation 1**: Case study on major attacks (17 SAQL queries): **low detection latency (< 2s)**
- **Evaluation 2**: Pressure test on the deployed server (12 cores, 128GB of RAM): **high system throughput (110,000 events/s; supporting ~4000 hosts)**
- **Evaluation 3**: Performance evaluation of the concurrent query scheduler (64 micro-benchmark queries): **30% average memory savings than existing systems**



Symantec

FireEye

Real-World Impact

SAQL has been selected as part of commercialization process and integrated in the NEC's security intelligence solution, which won the first place in the Town Life and Society Innovation Category at CEATEC Award 2016.

A complementary work by us:

- **AIQL**: enabling efficient attack investigation via querying the historical system monitoring data [USENIX ATC'18]
- Together, **SAQL and AIQL** work seamlessly for enabling effective and efficient APT defenses.