Towards Secure Provenance-Based Access Control in Cloud Environments

Adam Bates  Ben Mood  Masoud Valafar

Kevin Butler

CODASPY’13, San Antonio, TX, USA
19 February 2013
Data Provenance

Def: prov·en·ance \prä-və-nän(t)s\ n:

- A metadata history detailing an object’s derivation.
- Provides context needed to answer questions like:
  - What applications operated on this data?
  - What datasets helped produce this data?
  - In what environment was this data processed?
Cloud Provenance Uses

• Regulatory compliance.
• Debug experimental results.*
• Detect and avoid faulty data propagation.*
• Improve text search results.*
• Digital attack forensics.†

Data processing in the cloud benefits from the added context of data provenance.*

* Muniswamy-Reddy et al, FAST’10
† Galante et al. “Sony Network Breach...”
Obstacles

Challenges between us and our provenance-aware cloud:

- Host-level collection
- Storage
- Distributed security
- Distributed management
- Killer apps
Obstacles

Challenges between us and our provenance-aware cloud:

• Host-level collection
• Storage
• Distributed security
• Distributed management
• Killer apps
This work introduces:

- System and protocols to secure and manage provenance sent to cloud.
- Proof-of-concept provenance-aware cloud access control mechanisms.
- Performance evaluation that demonstrates minimal imposed overhead (~14%).
Cloud Provenance Authority

Host System

Data

Cloud Storage
Cloud Provenance Authority

**Enforcement Points**: Interact with clients and mediate access to cloud storage.

**Decision Points**: Issues access decisions and stores provenance and security policies.
Cloud Provenance Authority

**Enforcement Points**: Interact with clients and mediate access to cloud storage.

**Decision Points**: Issues access decisions and stores provenance and security policies.
Enforcement Points: Interact with clients and mediate access to cloud storage.

Decision Points: Issues access decisions and stores provenance and security policies.
```
Commitment Protocol

Client (C)

nonce_C, ID_C, ID_Obj

nonce_{CPA}, Prov(Obj,t-1), Sign[K^-_{CPA}, Prov(Obj,t-1)||nonce_C]

Obj_t, Prov(Obj,t), Sign[K^-_C, Prov(Obj,t)||nonce_{CPA}]

Sign[K^-_{CPA}, Prov(Obj,t)]

Enforcement Point
```
Commitment Protocol

Enforcement Point

Decision Point

Is ok if $ID_C$ writes to $ID_{Obj}$?

... Yup.
Commitment Protocol

Client (C)

nonce_C, ID_C, ID_{Obj}

nonce_{CPA}, Prov(Obj, t-1), Sign[K_{CPA}, Prov(Obj, t-1) || nonce_C]

Obj_t, Prov(Obj, t), Sign[K_C, Prov(Obj, t) || nonce_{CPA}]

Sign[K_{CPA}, Prov(Obj, t)]

Enforcement Point
Provenance Chains

Client (C)

nonce_C, ID_C, ID_{Obj}

nonce_{CPA}, Prov(Obj,t−1), Sign[K_{CPA},Prov(Obj,t−1)||nonce_C]

Obj_t, Prov(Obj,t), Sign[K_C,Prov(Obj,t)||nonce_{CPA}]

Sign[K_{CPA},Prov(Obj,t)]

Cloud Provenance Authority (CPA)
Management & Retrieval

Client

Org. 1

Org. 2
Management & Retrieval

(1) Read, ID$_C$

Client → Org. 1 → Org. 2
Management & Retrieval

(1) Read, $ID_C$

(2) $Prov_{k,t}$, $Sign[K_{O_1}^{-}\text{,Delegate:}O_2]$
Management & Retrieval

(1) Read, ID_C

(2) Prov_k,t, Sign[K_{O_1},Delegate:O_2]

(3) Sign[K_{O_1},Delegation,O_2]
Management & Retrieval

(1) Read, $ID_C$

(2) $Prov_{k,t}$, $Sign[K_{O_1}^−, Delegation: O_2]$

(3) $Sign[K_{O_1}^−, Delegation, O_2]$

(4) $Prov_{0,k−1}$
Client attempts to write 2.1 to Amazon AWS EU (Ireland)...
Client attempts to write 2.1 to Amazon AWS EU (Ireland)...

Write Request is *denied* at the Policy Decision Point because Object 2.1 is derived from ITAR data (Object 2.0)!
Implementation

**Cloud**: University of Oregon ACISS OpenStack KVM.

**Components**: VMs with 2vCPUs, 4 GB memory.

**Communication**: Amazon S3 REST API.
• Major overhead imposed by redundant data transmission (Client to PEP, PEP to Storage).
• Major overhead imposed by redundant data transmission (Client to PEP, PEP to Storage).
  
• By distributing the PEP workload, we reduced overhead to just 14%.
Performance

- Major overhead imposed by redundant data transmission (Client to PEP, PEP to Storage).
- By distributing the PEP workload, we reduced overhead to just 14%.
- Under realistic server workloads, our access control mechanism handled 1000 requests per second.
Since provenance is append-only, we cached previous access decisions in order to achieve amortized constant time.
Conclusion

• *Cloud Provenance Authorities* bring us one step closer to secure, provenance-aware distributed applications.

• We don’t need to wait on cloud providers to offer provenance services -- organizations can deploy *Cloud Provenance Authorities* using their own instances.

• Provenance applications such as access control can scale in the cloud when policy updates are infrequent.
Questions?

Adam Bates

amb@cs.uoregon.edu