Defining the Cloud Battlefield

Supporting Security Assessments by Cloud Customers

Sören Bleikertz [1] Toni Mastelić [2] Sebastian Pape [3]
Wolter Pieters [4] Trajce Dimkov [5]

[1]IBM Research - Zurich

[2]Vienna University of Technology

[3]TU Dortmund

[4]TU Delft / University of Twente

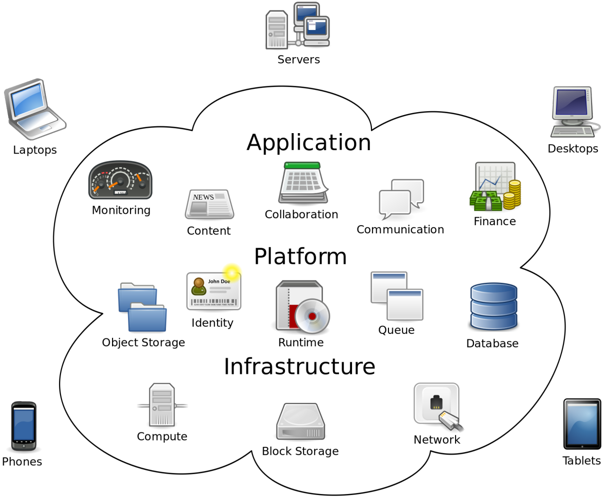
[5]Deloitte LLP

IEEE International Conference on Cloud Engineering 2013
(IC2E)

Outline

1. Introduction
    - Background
    - Research goal
2. System Model
3. Security Model
    - Security Objectives
    - Attacker Model
    - Threat Model
4. Model Applications
    - Applying the Model to Practical Attacks
    - Constructing What-if Attack Scenarios
5. Conclusions and Future Work

Introduction
Background: Cloud Computing



Cloud Computing

[wikipedia]

Introduction

Background: Security Concerns in Cloud Computing

- Security is a major concern [Mell and Grance, 2009]
- Analysis of risks and threats
  [Cloud Security Alliance, 2010], [ENISA, 2009]

  ⇒ insider attacks and malicious insiders are a major technical risk
- Risk amplified due disappearance of physical boundaries
  [Hay et al., 2011], [Pieters, 2011]
- Variety of parties involved in a cloud service

  ⇒ cloud customers face difficulties in assessing risks and threats

Introduction

Background: Sample Threats in Cloud Computing

- Malicious cloud administrator attacks virtual machine [Rocha and Correia, 2011]
- Malicious cloud customer attacks other customers who share physical resources [Ristenpart et al., 2009]
- Honest fault of a cloud administrator
  ⇒ outage of Amazon EC2 in 2011 [Amazon Web Services, 2011]
- Honest fault of cloud customers [Bugiel et al., 2011]:
  - SSH public key for administrator account in image
  - private SSH keys, Amazon credentials in image

Introduction

Background: Sample Threats in Cloud Computing

- Malicious cloud administrator attacks virtual machine [Rocha and Correia, 2011]
- Malicious cloud customer attacks other customers who share physical resources [Ristenpart et al., 2009]
- Honest fault of a cloud administrator
  ⇒ outage of Amazon EC2 in 2011 [Amazon Web Services, 2011]
- Honest fault of cloud customers [Bugiel et al., 2011]:
  - SSH public key for administrator account in image
  - private SSH keys, Amazon credentials in image

Samples cover only:

- Two entities: Cloud administrator and customer
- Two characteristics of attacker: honest faults and malicious

Introduction

Research goal: Supporting Security Assessment of Infrastructure Clouds

Aim:

- More fine-grained trust and attacker models
- Systematic specification of parties / capabilities / motivations

→ obtain a complete picture

→ support cloud customer's risk and threat assessments

- Model for cloud customers

→ understandability and usability are important

→ informal model is more accessible to this audience.

Challenge:

- Appropriate level of abstraction
- Combination of expressiveness and understandability

Introduction
Framework Overview

In summary, our framework combines

- System model of infrastructure clouds
  - entities
  - system components
- Security model
  - security objectives of cloud customers
  - attacker characteristics and motivation
  - threats

Introduction

Methodology: Designing an IaaS Threat Model

- Focus on infrastructure clouds (IaaS)
  - partly covers higher layers
  - needed for analysis of higher layers

- Design system model
- Design security model
- Identify and analyse attack scenarios
- Evaluation by mapping existing attacks to model
- Several iterations

- System. analysis by HAZOP approach [Winther et al., 2001]
  1. Identifying known attacks and map them to the model
  2. Analyze remaining combinations of entities, attacker, threats

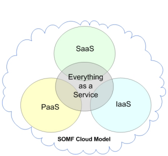
     → reveal possible unknown attacks

System Model

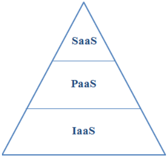
Background Cloud Computing

- Different abstraction layers: *IaaS*, *PaaS*, *SaaS*
- Focus on IaaS
  - ⇐ generic threat model too hard for all layers
  - increasing diversion → SaaS
    - c.f. Google GMail vs. Salesforce CRM
    - ⇒ application-specific attack models
- Existing models not suitable

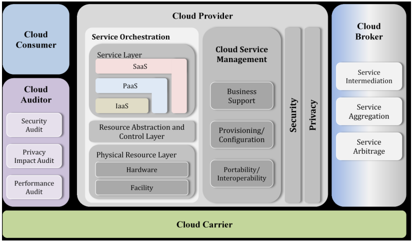
⇒ New cloud system model on IaaS layer consisting of entities and components.



SOMF Model



Cloud Pyramid



NIST Cloud Model

System Model

Entities

Chosen entities for the system model:

Provider manages and operates a cloud infrastructure

Manufacturer produces hardware resources used by the *provider*

Developer produce software used by the *provider*

Customer user of the cloud service provided by the *provider*

Third-party not directly involved in IaaS service,
represents user on higher layers of the cloud service
(e.g., SaaS)

System Model
Components

Each entity has access to one or more components:

Administration service, **logical access** to the cloud infrastructure

Technical Support service, **physical access** to the cloud infrastr.

Hardware e.g. hard-disk, processor, produced by a *manufacturer*, part of a cloud data center.

Software e.g. hypervisor, cloud management software produced by a *developer*, part of a cloud infrastructure.

Data information stored on hardware or being transmitted.

Appliance executable piece of software deployed by a *customer*,
includes higher layers of a cloud service,
black box completely controlled by a *customer*.
non running appliances considered as *data*

Usage represents usage by *third-party*,
logical access of an appliance

System Model

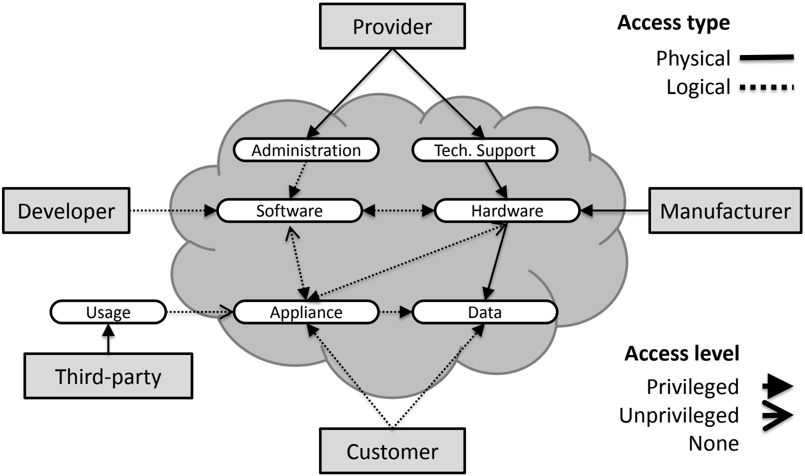


Figure: System model with relations between entities and components.

System Model

Access Type / Periods

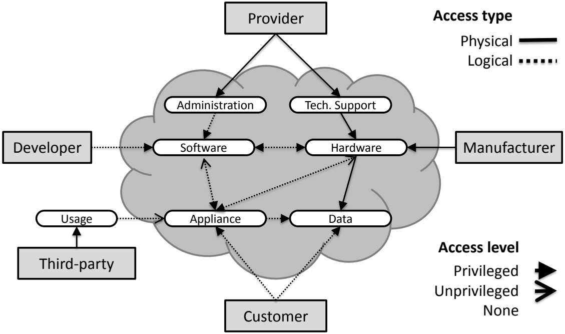


Figure: System model with relations between entities and components.

Access attributes

- direction
- transitivity

Access Type

- physically
- logically

Access Periods

- One-time
- Periodic
- Permanent

System Model

Access Level

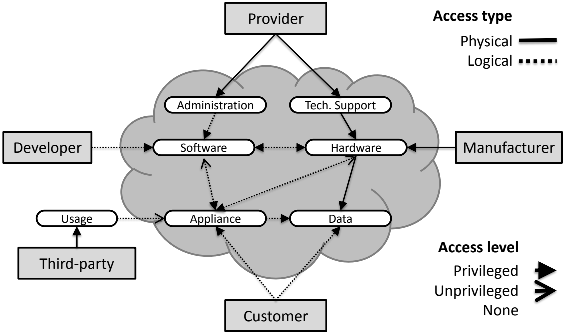


Figure: System model with relations between entities and components.

Access Level

levels:

- privileged
- unprivileged
- none

between:

- entity/comp. (priv.)
- comp./comp.

1. Introduction
   - Background
   - Research goal
2. System Model
3. Security Model
   - Security Objectives
   - Attacker Model
   - Threat Model
4. Model Applications
   - Applying the Model to Practical Attacks
   - Constructing What-if Attack Scenarios
5. Conclusions and Future Work

Security Model

Security Objectives of Cloud Customers

- Security objectives from a cloud customer's point of view
- Primary concern: exposure of sensitive data
- Focus on *(CIA)*
  - confidentiality
  - integrity
  - availability
- with regard to
  - computing
  - storage
  - network resources

Security Model

Security Objectives of Cloud Customers

Confidentiality of:

- *S1* executed appliances
- *S2* stored data
- *S3* transmitted data and appliances

Integrity of:

- *S4* executed appliances (comp. resources)
- *S5* stored data
- *S6* transmitted data and appliances
- *S7* software: hypervisor & management software

Integrity of: (cont.)

- *S8* hardware

Availability of:

- *S9* appliances: for customers & 3rd parties
- *S10* data: for customers and appliances
- *S11* software: mgmt. infrastructure & hypervisor
- *S12* hardware (analog to software)

Security Model

Attacker Model: Goals and Skills

- Goals
  - what a party wants to achieve
  - may use utility functions, with input
    - damage caused
    - expected gain
    - costs
    - risks associated
- Skills
  - the ability to realize these goals
  - determine outcome when parties have conflicting goals
  - may include a notion of available resources

Security Model

Attacker Model: Archetypes

Archetypes combine goals and skills

malicious (intentionally contribute to an attack): increases risk and associated damage to others for its own gain

ostrich (knowingly contribute to an attack): does not intend to increase risk for others, but fails to take action upon being informed about this (lazy)

charlatan (failing to acquire essential knowledge about contributing to an attack): increases risk for others, could/should have known (sloppy)

stepping stone (unknowingly contribute to an attack): increases risk for others, but could not have known (sloppy)

Security Model

Attacker Model: Archetypes

Archetypes combine goals and skills

malicious (intentionally contribute to an attack): increases risk and associated damage to others for its own gain

ostrich (knowingly contribute to an attack): does not intend to increase risk for others, but fails to take action upon being informed about this (lazy)

charlatan (failing to acquire essential knowledge about contributing to an attack): increases risk for others, could/should have known (sloppy)

stepping stone (unknowingly contribute to an attack): increases risk for others, but could not have known (sloppy)

- malicious and ostrich archetypes are driven by goals
  - ⇒ skill level determines the success of reaching such goals
- charlatan and stepping stone archetypes have low skills
  - ⇒ goal of providing a secure cloud service unsuccessful

Security Model

Attacker Model: Archetypes

defender (actively tries to prevent an attack): entity reduces risk for others

Motivation for a defender:

reputationalist (tries to improve utility of others to maintain reputation and thereby its own utility)

altruist (tries to improve the utility of others without necessarily benefiting itself)

- Archetypes applied on entities
- Components inherit the archetypes from their entities
- Archetype inherited represents a best possible archetype
  - e.g., *provider* can be a *charlatan*, but *administration* can be worse, i.e. *malicious*.

Security Model
Threat Model

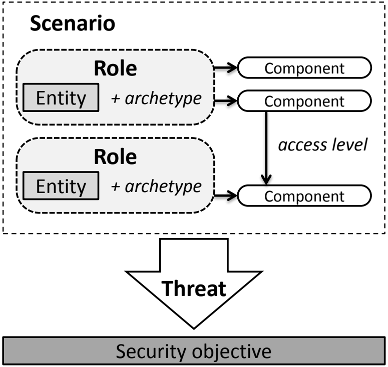


Figure: Deriving a threat from a role based scenario and security objective.

- Define a scenario by using a system model and archetypes
- Combine with security objective

→ Analyze a *threat*

⇒ A threat signals a particular scenario may violate a particular security objective through an attack

- Likelihood of a threat is influenced by attacker's
  - access levels
  - characteristics (including skills and goals)

Model Applications

Evaluation and Purpose

Evaluation:

- Assembled security threats from
  - Cloud Security Alliance [Brunette, 2010]
  - ENISA [Catteddu and Hogben, 2009]
  - Deloitte Cloud Risk Map [Deloitte, 2012]
- developed attack scenarios using subsets from our model

Practical purpose of model:

- Explain success of existing attacks and possible mitigations
- Produce a systematic set of threats
  - → input in developing a security assessment for a cloud solution
- Analyze behavior and motivation of entities
  - → insights into causes of threats
  - → cost-benefit assessment
- Define possible attack scenarios by presenting what-if scenarios in a consistent language

Applying the Model to Practical Attacks

Malicious Administrator Attacks - Scenario Description

- Several known attacks
- Oberheide et. al. [Oberheide et al., 2008]
  - attack on VMWare or Xen
  - administrator targets live migration of virtual machines
  - man-in-the-middle attacks during the migration
  - change of memory data or injection of an SSH key
- Rocha and Correia [Rocha and Correia, 2011]
  - administrator has access on the hypervisor
  - administrator has no access on the virtual machine itself
  - administrator uses memory dumps to derive clear text passwords or cryptographic keys

Applying the Model to Practical Attacks

Malicious Administrator Attacks - Model Application

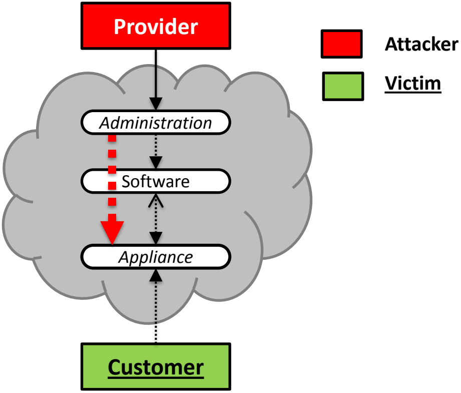


Figure: Malicious administration manipulating an appliance.

- *malicious administrator*
- provider itself may be *malicious* or: *ostrich* to *stepping stone*
- confidentiality and integrity of running *appliance* is violated
- corrupt the *appliance's* template when it is stored or transmitted over the network
- security objectives regarding availability concerned
- *administration* has permanent/periodic access

Applying the Model to Practical Attacks

Malicious Administrator Attacks - Mitigation and Assessment

- differences between possible archetypes of the provider
- no functional
  - *charlatan provider* hires a *malicious administrator*
  - *charlatan provider* fails to implement proper handling of security vulnerability reporting
  - *ostrich* does not perform necessary patch management
- technical mitigation
  - Trusted hypervisors [Garfinkel et al., 2003, Zhang et al., 2011]
  - Access control approaches [Bleikertz et al., 2012]
  - Fully homomorphic encryption [Gentry, 2009]
    still practically infeasible [Van Dijk and Juels, 2010]
  - A two-person administration [Potter et al., 2009]

Applying the Model to Practical Attacks

App Store Scenario - Model Application

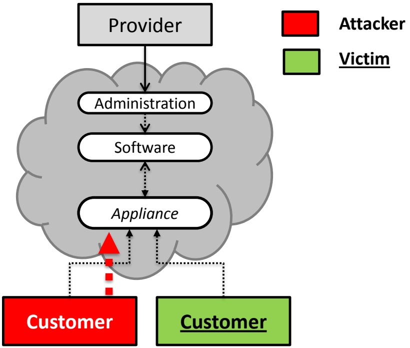


Figure: Attacking other customers through appliances.

- Relevant entities: *provider*, two instances of *customers*
- Two *customers* attack each other at appliance level
- Two scenarios
- leak of confidential information
  - ⇒ availability
  - ⇒ integrity of computations and stored data
  - ⇒ conf. of computations
- *provider* = app store owner
- *provider*: *ostrich*, *charlatan*, *stepping stone* or *defender*

Applying the Model to Practical Attacks

App Store Scenario - Mitigation and Assessment

- Amazon changed from *stepping stone* to *defender* (reputationalist)
- Requires scanning and cleaning of infected/malicious images [Balduzzi et al., 2012]
- Alternatively: pre-emptive image management system that provides a secured access to images [Wei et al., 2009]
- *defender provider* could patch VM images [Zhou et al., 2010]

Applying the Model to Practical Attacks

Side-channel Attacks - Model Application

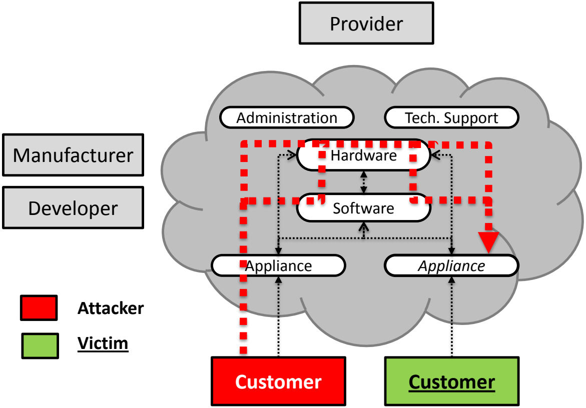


Figure: Attacking other customers through side-channels in hardware and/or software.

Applying the Model to Practical Attacks

Virtual Machine Escapes - Model Application

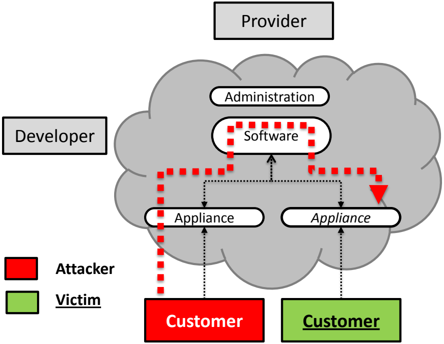


Figure: Attacking customer escapes appliance's environment to attack other customers.

- involved entities
  - attacking and victim *customer*
  - *ostrich* to *stepping stone* or defender cloud *provider*
  - *ostrich* to *stepping stone* or defender software *developer*.
- confidentiality and integrity of the running *appliance* is affected
- integrity of stored or transmitted *appliance*

Applying the Model to Practical Attacks

Constructing What-if Attack Scenarios

- Model also useful for constructing “what-if” scenarios
  - combine multiple entities of our model with attacker roles
  - change an attacker’s characteristic
  - structured assessment of infrastructure cloud security
  - may lead to new attacks

Applying the Model to Practical Attacks

What-if Scenarios: Large Scale VM Escape Attacks

- VM escape attack
- Malicious *customer* + *ostrich*/*charlatan developer*
- Insecure cloud management *software*
- Cloud *provider* and *customers* at large can be attacked
- Injection of management commands into the insecure management *software*
  - ⇒ attacker can terminate appliances
  - ⇒ attacker can consume resources from the *provider* for free
- Additionally: *manufacturer* is *ostrich* or *charlatan*

⇒ hardware could be damaged

Applying the Model to Practical Attacks

What-if Scenarios: Insecure Cloud Management Software / Collusion Attacks in Cloud-of-Clouds

- Insecure Cloud Management Software may lead to the same consequences as VM Escape Attacks
- Cloud-of-Clouds systems aggregate multiple clouds
  - → tolerate byzantine faults of single clouds
  - operated by different organizations
  - ⇒ *administration* and *technical support* of the *providers* do not collude
  - may use the same *software* or *hardware* provided by *malicious/ostrich/charlatan developers* or *manufacturers*
  - ⇒ diminish the security advantages of cloud-of-clouds systems

Applying the Model to Practical Attacks

What-if Scenarios: Hardware Trojans

- [Skorobogatov and Woods, 2012] claim to have discovered hardware trojan
- Not seen in cloud computing, yet
- *Manufacturer* also becomes a *customer* in public clouds that use its *hardware*

→ Malicious *manufacturer* has one-time access to the hardware

→ *Customer* has permanent access to his *appliance*

- May change the way hardware works
- Threats: availability and integrity for
  - other *appliances*
  - the hypervisor and management software

Conclusions and Future Work

- We proposed a cloud security threat model that combines
  - Comprehensive system model of infrastructure clouds
  - Security model focusing on cloud customer security objectives
  - Threat model with characteristics and motivations of attackers
- We used our model to
  - systematic categorization
  - analysis of existing attacks
  - construction of "what-if" attack scenarios
- Customers can apply the approach to competing cloud providers
  - Requires sufficient data about the architecture or Trusted Third Party [Probst et al., 2012].

Conclusions and Future Work

- Model forced a structured approach in describing existing attacks
- Model is well-suited for attacks involving technical infrastructure and behavior of entities
- Threats involving governance and compliance, or threats to security monitoring, cannot be easily expressed
- By considering entities not directly involved in an attack, amplification or reduction of threats by these entities can be made visible

Future Work

- Formalization of our model
  - process calculi for the system model
  - utility functions for the attacker goals
- Extend scope of our model
  - upper abstraction layers in cloud computing, e.g. PaaS
  - consider non-technical security threats such as legal or compliance ones
- Systematic categorization and analysis of protection mechanisms

Amazon Web Services (2011).
Summary of the Amazon EC2 and Amazon RDS Service Disruption in the US East Region.
http://aws.amazon.com/message/65648/.

Balduzzi, M., Zaddach, J., Balzarotti, D., Kirda, E., and Loureiro, S. (2012).
A Security Analysis of Amazon's Elastic Compute Cloud Service.
In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, pages 1427–1434, New York, NY, USA. ACM.

Bleikertz, S., Kurmus, A., Nagy, Z. A., and Schunter, M. (2012).
Secure cloud maintenance - protecting workloads against insider attacks.
In *7th ACM Symposium on Information, Computer and Communications Security (ASIACCS'12)*. ACM.

Brunette, G. Mogull, R. (2010).
Security guidance for critical areas of focus in cloud computing.

Cloud Security Alliance.

Bugiel, S., Nürnberger, S., Pöppelmann, T., Sadeghi, A.-R., and Schneider, T. (2011).
Amazonia: when elasticity snaps back.
In *Proceedings of the 18th ACM conference on Computer and communications security*, CCS '11, pages 389–400, New York, NY, USA. ACM.

Catteddu, D. and Hogben, G. (2009).
Cloud computing risk assessment.
European Network and Information Security Agency (ENISA).

Cloud Security Alliance (2010).
Top threats to cloud computing v1.0.
`https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf`.

Deloitte (2012).
Cloud security risk map.
http://tinyurl.com/935ktap.

ENISA (2009).
Cloud Computing Risk Assessment.
Technical report, ENISA.

Garfinkel, T., Pfaff, B., Chow, J., Rosenblum, M., and Boneh, D. (2003).
Terra: A Virtual Machine-Based Platform for Trusted Computing.
SIGOPS Oper. Syst. Rev., 37(5):193–206.

Gentry, C. (2009).
Fully homomorphic encryption using ideal lattices.
In *41st annual ACM symposium on Theory of Computing*. ACM.

Hay, B., Nance, K., and Bishop, M. (2011).

Storm Clouds Rising: Security Challenges for IaaS Cloud Computing.
In *Proceedings of the 2011 44th Hawaii International Conference on System Sciences*, HICSS '11, pages 1–7, Washington, DC, USA. IEEE Computer Society.

Mell, P. and Grance, T. (2009).
Effectively and Securely Using the Cloud Computing Paradigm.

Oberheide, J., Cooke, E., and Jahanian, F. (2008).
Exploiting Live Virtual Machine Migration.
In *BlackHat DC Briefings*, Washington DC.

Pieters, W. (2011).
Security and privacy in the clouds: a bird's eye view.
In Gutwirth, S., Poullet, Y., De Hert, P., and Leenes, R., editors, *Computers, Privacy and Data Protection: an Element of Choice*, pages 445–457. Springer, Dordrecht.

Potter, S., Bellovin, S. M., and Nieh, J. (2009).

Two-Person Control Administration: Preventing Administration Faults Through Duplication.
In *Proceedings of the 23rd conference on Large installation system administration*, LISA'09, pages 15–27, Berkeley, CA, USA. USENIX Association.

Probst, C. W., Sasse, M. A., Pieters, W., Dimkov, T., Luysterborg, E., and Arnaud, M. (2012).
Privacy penetration testing: How to establish trust in your cloud provider.
In Gutwirth, S., Leenes, R., De Hert, P., and Poullet, Y., editors, *European Data Protection: In Good Health?*, pages 251–265. Springer Netherlands.

Ristenpart, T., Tromer, E., Shacham, H., and Savage, S. (2009).
Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds.

In *CCS '09: Proceedings of the 16th ACM conference on Computer and Communications Security*, pages 199–212, New York, NY, USA. ACM.

Rocha, F. and Correia, M. (2011).
Lucy in the sky without diamonds: Stealing confidential data in the cloud.
In *Proceedings of the 1st International Workshop on Dependability of Clouds, Data Centers and Virtual Computing Environments (DCDV, with DSN'11)*.

Skorobogatov, S. and Woods, C. (2012).
Breakthrough silicon scanning discovers backdoor in military chip.
In *CHES*, pages 23–40.

Van Dijk, M. and Juels, A. (2010).
On the impossibility of cryptography alone for privacy-preserving cloud computing.

In *5th USENIX conference on Hot topics in security (HotSec'10)*. USENIX.

Wei, J., Zhang, X., Ammons, G., Bala, V., and Ning, P. (2009).
Managing security of virtual machine images in a cloud environment.
In *Proceedings of the 2009 ACM workshop on Cloud computing security*, CCSW '09, pages 91–96, New York, NY, USA. ACM.

Winther, R., Johnsen, O.-A., and Gran, B. (2001).
Security assessments of safety critical systems using HAZOPs.
In Voges, U., editor, *Computer Safety, Reliability and Security*, volume 2187 of *Lecture Notes in Computer Science*, pages 14–24. Springer Berlin / Heidelberg.

Zhang, F., Chen, J., Chen, H., and Zang, B. (2011).
Cloudvisor: retrofitting protection of virtual machines in multi-tenant cloud with nested virtualization.

In *23rd ACM Symposium on Operating Systems Principles (SOSP'11)*. ACM.

Zhou, W., Ning, P., Zhang, X., Ammons, G., Wang, R., and Bala, V. (2010).
Always up-to-date: scalable offline patching of vm images in a compute cloud.
In *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, pages 377–386, New York, NY, USA. ACM.