Understanding Cloud Computing Vulnerabilities

Discussions about cloud computing security often fail to distinguish general issues from cloud-specific issues. To clarify the discussions regarding vulnerabilities, the authors define indicators based on sound definitions of risk factors and cloud computing.

Bernd Grobauer, Tobias Walloschek, and Elmar Stöcker
Siemens

Each day, a fresh news item, blog entry, or other publication warns us about cloud computing's security risks and threats; in most cases, security is cited as the most substantial roadblock for cloud computing uptake. But this discourse about cloud computing security issues makes it difficult to formulate a well-founded assessment of the actual security impact for two key reasons. First, in many of these discussions about risk, basic vocabulary terms—including *risk*, *threat*, and *vulnerability*—are often used interchangeably, without regard to their respective definitions. Second, not every issue raised is specific to cloud computing.

To achieve a well-founded understanding of the "delta" that cloud computing adds with respect to security issues, we must analyze how cloud computing influences established security issues. A key factor here is security *vulnerabilities*: cloud computing makes certain well-understood vulnerabilities more significant as well as adds new ones to the mix. Before we take a closer look at cloud-specific vulnerabilities, however, we must first establish what a "vulnerability" really is.

Vulnerability: An Overview

Vulnerability is a prominent factor of risk. ISO 27005 defines risk as "the potential that a given threat will exploit vulnerabilities of an asset or group of assets and thereby cause harm to the organization," measuring it in terms of both the likelihood of an event and its consequence.[1] The Open Group's risk taxonomy (www.opengroup.org/onlinepubs/9699919899/toc.pdf) offers a useful overview of risk factors (see Figure 1).

The Open Group's taxonomy uses the same two top-level risk factors as ISO 27005: the likelihood of a harmful event (here, *loss event frequency*) and its consequence (here, *probable loss magnitude*).[1] The probable loss magnitude's subfactors (on the right in Figure 1) influence a harmful event's ultimate cost. The loss event frequency subfactors (on the left) are a bit more complicated. A loss event occurs when a threat agent (such as a hacker) successfully exploits a vulnerability. The frequency with which this happens depends on two factors:

- The frequency with which threat agents try to exploit a vulnerability. This frequency is determined by both the agents' motivation (What can they gain with an attack? How much effort does it take? What is the risk for the attackers?) and how much access ("contact") the agents have to the attack targets.
- The difference between the threat agents' attack capabilities and the system's strength to resist the attack.

This second factor brings us toward a useful definition of vulnerability.

Defining Vulnerability

According to the Open Group's risk taxonomy,

> Vulnerability is the probability that an asset will be unable to resist the actions of a threat agent. Vulnerability exists when there is a difference between the

1540-7993/11/$26.00 © 2011 IEEE

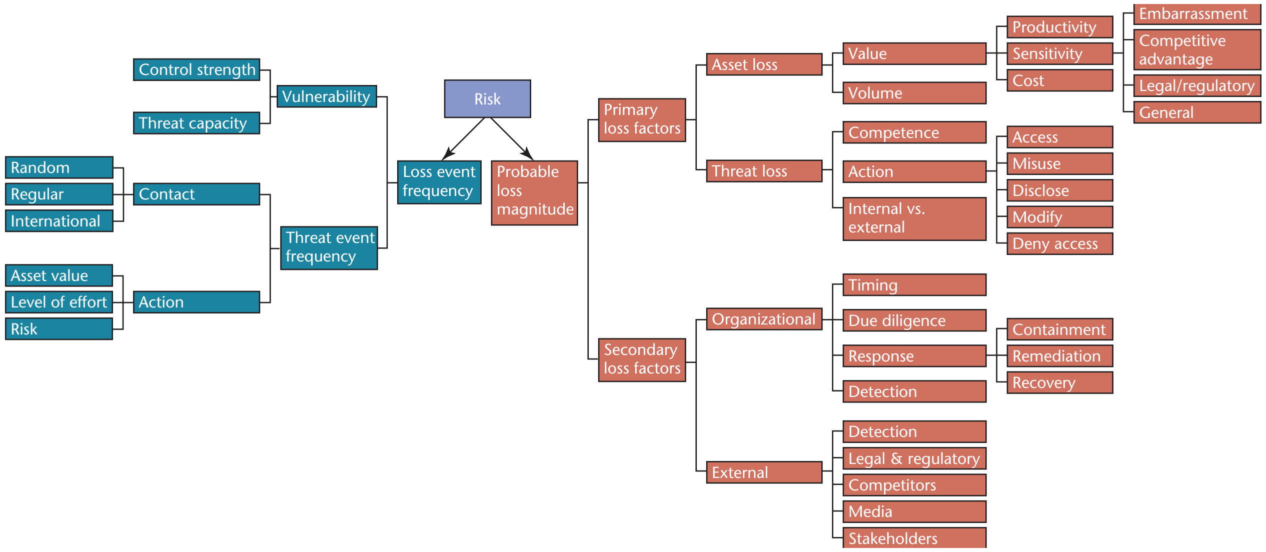

Figure 1. Factors contributing to risk according to the Open Group's risk taxonomy. Risk corresponds to the product of loss event frequency (left) and probable loss magnitude (right). Vulnerabilities influence the loss event frequency.

> force being applied by the threat agent, and an object's ability to resist that force.

So, vulnerability must always be described in terms of resistance to a certain type of attack. To provide a real-world example, a car's inability to protect its driver against injury when hit frontally by a truck driving 60 mph is a vulnerability; the resistance of the car's crumple zone is simply too weak compared to the truck's force. Against the "attack" of a biker, or even a small car driving at a more moderate speed, the car's resistance strength is perfectly adequate.

We can also describe computer vulnerability—that is, security-related bugs that you close with vendor-provided patches—as a weakening or removal of a certain resistance strength. A buffer-overflow vulnerability, for example, weakens the system's resistance to arbitrary code execution. Whether attackers can exploit this vulnerability depends on their capabilities.

Vulnerabilities and Cloud Risk

We'll now examine how cloud computing influences the risk factors in Figure 1, starting with the right-hand side of the risk factor tree.

From a cloud customer perspective, the right-hand side dealing with probable magnitude of future loss isn't changed at all by cloud computing: the consequences and ultimate cost of, say, a confidentiality breach, is exactly the same regardless of whether the data breach occurred within a cloud or a conventional IT infrastructure. For a cloud service provider, things look somewhat different: because cloud computing systems were previously separated on the same infrastructure, a loss event could entail a considerably larger impact. But this fact is easily grasped and incorporated into a risk assessment: no conceptual work for adapting impact analysis to cloud computing seems necessary.

So, we must search for changes on Figure 1's left-hand side—the loss event frequency. Cloud computing could change the probability of a harmful event's occurrence. As we show later, cloud computing causes significant changes in the vulnerability factor. Of course, moving to a cloud infrastructure might change the attackers' access level and motivation, as well as the effort and risk—a fact that must be considered as future work. But, for supporting a cloud-specific risk assessment, it seems most profitable to start by examining the exact nature of cloud-specific vulnerabilities.

Cloud Computing

Is there such a thing as a "cloud-specific" vulnerability? If so, certain factors in cloud computing's nature must make a vulnerability cloud-specific.

Essentially, cloud computing combines known technologies (such as virtualization) in ingenious ways to provide IT services "from the conveyor belt" using economies of scale. We'll now look closer at what the core technologies are and which characteristics of their use in cloud computing are essential.

Core Cloud Computing Technologies

Cloud computing builds heavily on capabilities available through several core technologies:

- *Web applications and services.* Software as a service

(SaaS) and platform as a service (PaaS) are unthinkable without Web application and Web services technologies: SaaS offerings are typically implemented as Web applications, while PaaS offerings provide development and runtime environments for Web applications and services. For infrastructure as a service (IaaS) offerings, administrators typically implement associated services and APIs, such as the management access for customers, using Web application/service technologies.
- *Virtualization IaaS offerings.* These technologies have virtualization techniques at their very heart; because PaaS and SaaS services are usually built on top of a supporting IaaS infrastructure, the importance of virtualization also extends to these service models. In the future, we expect virtualization to develop from virtualized servers toward computational resources that can be used more readily for executing SaaS services.
- *Cryptography.* Many cloud computing security requirements are solvable only by using cryptographic techniques.

As cloud computing develops, the list of core technologies is likely to expand.

Essential Characteristics

In its description of essential cloud characteristics,[2] the US National Institute of Standards and Technology (NIST) captures well what it means to provide IT services from the conveyor belt using economies of scale:

- *On-demand self-service.* Users can order and manage services without human interaction with the service provider, using, for example, a Web portal and management interface. Provisioning and de-provisioning of services and associated resources occur automatically at the provider.
- *Ubiquitous network access.* Cloud services are accessed via the network (usually the Internet), using standard mechanisms and protocols.
- *Resource pooling.* Computing resources used to provide the cloud service are realized using a homogeneous infrastructure that's shared between all service users.
- *Rapid elasticity.* Resources can be scaled up and down rapidly and elastically.
- *Measured service.* Resource/service usage is constantly metered, supporting optimization of resource usage, usage reporting to the customer, and pay-as-you-go business models.

NIST's definition framework for cloud computing with its list of essential characteristics has by now evolved into the de facto standard for defining cloud computing.

Cloud-Specific Vulnerabilities

Based on the abstract view of cloud computing we presented earlier, we can now move toward a definition of what constitutes a cloud-specific vulnerability. A vulnerability is cloud specific if it

- is intrinsic to or prevalent in a core cloud computing technology,
- has its root cause in one of NIST's essential cloud characteristics,
- is caused when cloud innovations make tried-and-tested security controls difficult or impossible to implement, or
- is prevalent in established state-of-the-art cloud offerings.

We now examine each of these four indicators.

Core-Technology Vulnerabilities

Cloud computing's core technologies—Web applications and services, virtualization, and cryptography—have vulnerabilities that are either intrinsic to the technology or prevalent in the technology's state-of-the-art implementations. Three examples of such vulnerabilities are virtual machine escape, session riding and hijacking, and insecure or obsolete cryptography.

First, the possibility that an attacker might successfully escape from a virtualized environment lies in virtualization's very nature. Hence, we must consider this vulnerability as intrinsic to virtualization and highly relevant to cloud computing.

Second, Web application technologies must overcome the problem that, by design, the HTTP protocol is a stateless protocol, whereas Web applications require some notion of session state. Many techniques implement session handling and—as any security professional knowledgeable in Web application security will testify—many session handling implementations are vulnerable to session riding and session hijacking. Whether session riding/hijacking vulnerabilities are intrinsic to Web application technologies or are "only" prevalent in many current implementations is arguable; in any case, such vulnerabilities are certainly relevant for cloud computing.

Finally, cryptoanalysis advances can render any cryptographic mechanism or algorithm insecure as novel methods of breaking them are discovered. It's even more common to find crucial flaws in cryptographic algorithm implementations, which can turn strong encryption into weak encryption (or sometimes no encryption at all). Because broad uptake of cloud computing is unthinkable without the use of cryptography to protect data confidentiality and integrity in the cloud, insecure or obsolete cryptography vulnerabilities are highly relevant for cloud computing.

Essential Cloud Characteristic Vulnerabilities

As we noted earlier, NIST describes five essential cloud characteristics: on-demand self-service, ubiquitous network access, resource pooling, rapid elasticity, and measured service.

Following are examples of vulnerabilities with root causes in one or more of these characteristics:

- *Unauthorized access to management interface.* The cloud characteristic on-demand self-service requires a management interface that's accessible to cloud service users. Unauthorized access to the management interface is therefore an especially relevant vulnerability for cloud systems: the probability that unauthorized access could occur is much higher than for traditional systems where the management functionality is accessible only to a few administrators.
- *Internet protocol vulnerabilities*. The cloud characteristic ubiquitous network access means that cloud services are accessed via network using standard protocols. In most cases, this network is the Internet, which must be considered untrusted. Internet protocol vulnerabilities—such as vulnerabilities that allow man-in-the-middle attacks—are therefore relevant for cloud computing.
- *Data recovery vulnerability.* The cloud characteristics of pooling and elasticity entail that resources allocated to one user will be reallocated to a different user at a later time. For memory or storage resources, it might therefore be possible to recover data written by a previous user.
- *Metering and billing evasion*. The cloud characteristic of measured service means that any cloud service has a metering capability at an abstraction level appropriate to the service type (such as storage, processing, and active user accounts). Metering data is used to optimize service delivery as well as billing. Relevant vulnerabilities include metering and billing data manipulation and billing evasion.

Thus, we can leverage NIST's well-founded definition of cloud computing in reasoning about cloud computing issues.

Defects in Known Security Controls

Vulnerabilities in standard security controls must be considered cloud specific if cloud innovations directly cause the difficulties in implementing the controls. Such vulnerabilities are also known as *control challenges*.

Here, we treat three examples of such control challenges. First, virtualized networks offer insufficient network-based controls. Given the nature of cloud services, the administrative access to IaaS network infrastructure and the ability to tailor network infrastructure are typically limited; hence, standard controls such as IP-based network zoning can't be applied. Also, standard techniques such as network-based vulnerability scanning are usually forbidden by IaaS providers because, for example, friendly scans can't be distinguished from attacker activity. Finally, technologies such as virtualization mean that network traffic occurs on both real and virtual networks, such as when two virtual machine environments (VMEs) hosted on the same server communicate. Such issues constitute a control challenge because tried and tested network-level security controls might not work in a given cloud environment.

The second challenge is in poor key management procedures. As noted in a recent European Network and Information Security Agency study,[3] cloud computing infrastructures require management and storage of many different kinds of keys. Because virtual machines don't have a fixed hardware infrastructure and cloud-based content is often geographically distributed, it's more difficult to apply standard controls—such as hardware security module (HSM) storage—to keys on cloud infrastructures.

Finally, security metrics aren't adapted to cloud infrastructures. Currently, there are no standardized cloud-specific security metrics that cloud customers can use to monitor the security status of their cloud resources. Until such standard security metrics are developed and implemented, controls for security assessment, audit, and accountability are more difficult and costly, and might even be impossible to employ.

Prevalent Vulnerabilities in State-of-the-Art Cloud Offerings

Although cloud computing is relatively young, there are already myriad cloud offerings on the market. Hence, we can complement the three cloud-specific vulnerability indicators presented earlier with a forth, empirical indicator: if a vulnerability is prevalent in state-of-the-art cloud offerings, it must be regarded as cloud-specific. Examples of such vulnerabilities include injection vulnerabilities and weak authentication schemes.

Injection vulnerabilities are exploited by manipulating service or application inputs to interpret and execute parts of them against the programmer's intentions. Examples of injection vulnerabilities include

- SQL injection, in which the input contains SQL code that's erroneously executed in the database back end;
- command injection, in which the input contains commands that are erroneously executed via the OS; and
- cross-site scripting, in which the input contains JavaScript code that's erroneously executed by a victim's browser.

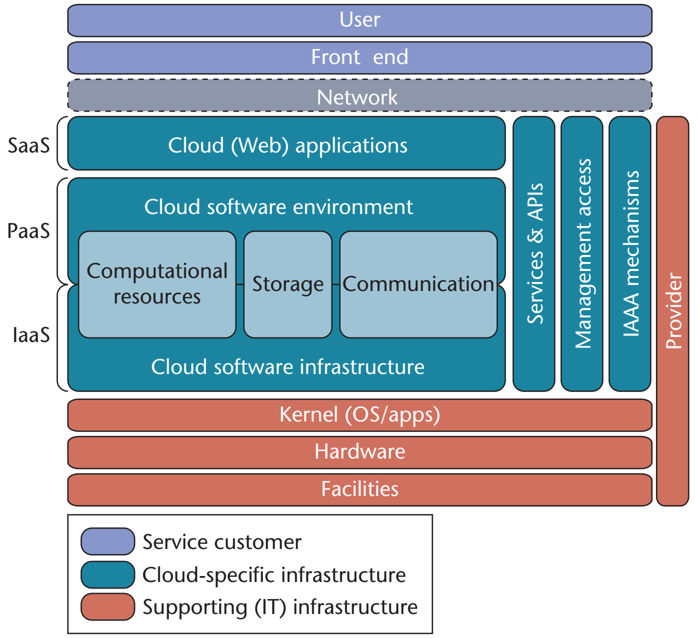

Figure 2. The cloud reference architecture. We map cloud-specific vulnerabilities to components of this reference architecture, which gives us an overview of which vulnerabilities might be relevant for a given cloud service.

In addition, many widely used authentication mechanisms are weak. For example, usernames and passwords for authentication are weak due to

- insecure user behavior (choosing weak passwords, reusing passwords, and so on), and
- inherent limitations of one-factor authentication mechanisms.

Also, the authentication mechanisms' implementation might have weaknesses and allow, for example, credential interception and replay. The majority of Web applications in current state-of-the-art cloud services employ usernames and passwords as authentication mechanism.

Architectural Components and Vulnerabilities

Cloud service models are commonly divided into SaaS, PaaS, and IaaS, and each model influences the vulnerabilities exhibited by a given cloud infrastructure. It's helpful to add more structure to the service model stacks: Figure 2 shows a cloud reference architecture that makes the most important security-relevant cloud components explicit and provides an abstract overview of cloud computing for security issue analysis.

The reference architecture is based on work carried out at the University of California, Los Angeles, and IBM.[4] It inherits the layered approach in that layers can encompass one or more service components. Here, we use "service" in the broad sense of providing something that might be both material (such as shelter, power, and hardware) and immaterial (such as a runtime environment). For two layers, the cloud software environment and the cloud software infrastructure, the model makes the layers' three main service components—computation, storage, and communication—explicit. Top layer services also can be implemented on layers further down the stack, in effect skipping intermediate layers. For example, a cloud Web application can be implemented and operated in the traditional way—that is, running on top of a standard OS without using dedicated cloud software infrastructure and environment components. Layering and compositionality imply that the transition from providing some service or function in-house to sourcing the service or function can take place between any of the model's layers.

In addition to the original model, we've identified supporting functions relevant to services in several layers and added them to the model as vertical spans over several horizontal layers.

Our cloud reference architecture has three main parts:

- *Supporting (IT) infrastructure.* These are facilities and services common to any IT service, cloud or otherwise. We include them in the architecture because we want to provide the complete picture; a full treatment of IT security must account for a cloud service's non-cloud-specific components.
- *Cloud-specific infrastructure.* These components constitute the heart of a cloud service; cloud-specific vulnerabilities and corresponding controls are typically mapped to these components.
- *Cloud service consumer.* Again, we include the cloud service customer in the reference architecture because it's relevant to an all-encompassing security treatment.

Also, we make explicit the network that separates the cloud service consumer from the cloud infrastructure; the fact that access to cloud resources is carried out via a (usually untrusted) network is one of cloud computing's main characteristics.

Using the cloud reference architecture's structure, we can now run through the architecture's components and give examples of each component's cloud-specific vulnerabilities.

Cloud Software Infrastructure and Environment

The *cloud software infrastructure* layer provides an abstraction level for basic IT resources that are offered as services to higher layers: computational resources (usually VMEs), storage, and (network) communication. These services can be used individually, as is typically the case with storage services, but they're often bundled such that servers are delivered with certain network connectivity and (often) access to storage. This bundle, with or without storage, is usually referred to as IaaS.

The *cloud software environment* layer provides services at the application platform level:

- a development and runtime environment for services and applications written in one or more supported languages;
- storage services (a database interface rather than file share); and
- communication infrastructure, such as Microsoft's Azure service bus.

Vulnerabilities in both the infrastructure and environment layers are usually specific to one of the three resource types provided by these two layers. However, cross-tenant access vulnerabilities are relevant for all three resource types. The virtual machine escape vulnerability we described earlier is a prime example. We used it to demonstrate a vulnerability that's intrinsic to the core virtualization technology, but it can also be seen as having its root cause in the essential characteristic of resource pooling: whenever resources are pooled, unauthorized access across resources becomes an issue. Hence, for PaaS, where the technology to separate different tenants (and tenant services) isn't necessarily based on virtualization (although that will be increasingly true), cross-tenant access vulnerabilities play an important role as well. Similarly, cloud storage is prone to cross-tenant storage access, and cloud communication—in the form of virtual networking—is prone to cross-tenant network access.

Computational Resources

A highly relevant set of computational resource vulnerabilities concerns how virtual machine images are handled: the only feasible way of providing nearly identical server images—thus providing on-demand service for virtual servers—is by cloning template images.

Vulnerable virtual machine template images cause OS or application vulnerabilities to spread over many systems. An attacker might be able to analyze configuration, patch level, and code in detail using administrative rights by renting a virtual server as a service customer and thereby gaining knowledge helpful in attacking other customers' images. A related problem is that an image can be taken from an untrustworthy source, a new phenomenon brought on especially by the emerging marketplace of virtual images for IaaS services. In this case, an image might, for example, have been manipulated so as to provide back-door access for an attacker.

Data leakage by virtual machine replication is a vulnerability that's also rooted in the use of cloning for providing on-demand service. Cloning leads to data leakage problems regarding machine secrets: certain elements of an OS—such as host keys and cryptographic salt values—are meant to be private to a single host. Cloning can violate this privacy assumption. Again, the emerging marketplace for virtual machine images, as in Amazon EC2, leads to a related problem: users can provide template images for other users by turning a running image into a template. Depending on how the image was used before creating a template from it, it could contain data that the user doesn't wish to make public.

There are also control challenges here, including those related to cryptography use. Cryptographic vulnerabilities due to weak random number generation might exist if the abstraction layer between the hardware and OS kernel introduced by virtualization is problematic for generating random numbers within a VME. Such generation requires an entropy source on the hardware level. Virtualization might have flawed mechanisms for tapping that entropy source, or having several VMEs on the same host might exhaust the available entropy, leading to weak random number generation. As we noted earlier, this abstraction layer also complicates the use of advanced security controls, such as hardware security modules, possibly leading to poor key management procedures.

Storage

In addition to data recovery vulnerability due to resource pooling and elasticity, there's a related control challenge in media sanitization, which is often hard or impossible to implement in a cloud context. For example, data destruction policies applicable at the end of a life cycle that require physical disk destruction can't be carried out if a disk is still being used by another tenant.

Because cryptography is frequently used to overcome storage-related vulnerabilities, this core technology's vulnerabilities—insecure or obsolete cryptography and poor key management—play a special role for cloud storage.

Communication

The most prominent example of a cloud communications service is the networking provided for VMEs in an IaaS environment. Because of resource pooling, several customers are likely to share certain network infrastructure components: vulnerabilities of shared network infrastructure components, such as vulnerabilities in a DNS server, Dynamic Host Configuration Protocol, and IP protocol vulnerabilities, might enable network-based cross-tenant attacks in an IaaS infrastructure.

Virtualized networking also presents a control challenge: again, in cloud services, the administrative access to IaaS network infrastructure and the possibility for tailoring network infrastructure are usually limited. Also, using technologies such as virtualization leads to a situation where network traffic occurs not only on "real" networks but also within virtualized networks (such as for communication between two VMEs hosted on the same server); most implementations of virtual networking offer limited possibilities for integrating network-based security. All in all, this constitutes a control challenge of insufficient network-based controls because tried-and-tested network-level security controls might not work in a given cloud environment.

Cloud Web Applications

A Web application uses browser technology as the front end for user interaction. With the increased uptake of browser-based computing technologies such as JavaScript, Java, Flash, and Silverlight, a Web cloud application falls into two parts:

- an application component operated somewhere in the cloud, and
- a browser component running within the user's browser.

In the future, developers will increasingly use technologies such as Google Gears to permit offline usage of a Web application's browser component for use cases that don't require constant access to remote data. We've already described two typical vulnerabilities for Web application technologies: session riding and hijacking vulnerabilities and injection vulnerabilities.

Other Web-application-specific vulnerabilities concern the browser's front-end component. Among them are client-side data manipulation vulnerabilities, in which users attack Web applications by manipulating data sent from their application component to the server's application component. In other words, the input received by the server component isn't the "expected" input sent by the client-side component, but altered or completely user-generated input. Furthermore, Web applications also rely on browser mechanisms for isolating third-party content embedded in the application (such as advertisements, mashup components, and so on). Browser isolation vulnerabilities might thus allow third-party content to manipulate the Web application.

Services and APIs

It might seem obvious that all layers of the cloud infrastructure offer services, but for examining cloud infrastructure security, it's worthwhile to explicitly think about all of the infrastructure's service and application programming interfaces. Most services are likely Web services, which share many vulnerabilities with Web applications. Indeed, the Web application layer might be realized completely by one or more Web services such that the application URL would only give the user a browser component. Thus the supporting services and API functions share many vulnerabilities with the Web applications layer.

Management Access

NIST's definition of cloud computing states that one of cloud services' central characteristics is that they can be rapidly provisioned and released with minimal management effort or service provider interaction. Consequently, a common element of each cloud service is a management interface—which leads directly to the vulnerability concerning unauthorized access to the management interface. Furthermore, because management access is often realized using a Web application or service, it often shares the vulnerabilities of the Web application layer and services/API component.

Identity, Authentication, Authorization, and Auditing Mechanisms

All cloud services (and each cloud service's management interface) require mechanisms for identity management, authentication, authorization, and auditing (IAAA). To a certain extent, parts of these mechanisms might be factored out as a stand-alone IAAA service to be used by other services. Two IAAA elements that must be part of each service implementation are execution of adequate authorization checks (which, of course, use authentication and/or authorization information received from an IAA service) and cloud infrastructure auditing.

Most vulnerabilities associated with the IAAA component must be regarded as cloud-specific because they're prevalent in state-of-the-art cloud offerings. Earlier, we gave the example of weak user authentication mechanisms; other examples include

- *Denial of service by account lockout.* One often-used security control—especially for authentication with username and password—is to lock out accounts that have received several unsuccessful authentication attempts in quick succession. Attackers can use such attempts to launch DoS attacks against a user.
- *Weak credential-reset mechanisms.* When cloud computing providers manage user credentials themselves rather than using federated authentication, they must provide a mechanism for resetting credentials in the case of forgotten or lost credentials. In the past, password-recovery mechanisms have proven particularly weak.
- *Insufficient or faulty authorization checks.* State-of-the-

art Web application and service cloud offerings are often vulnerable to insufficient or faulty authorization checks that can make unauthorized information or actions available to users. Missing authorization checks, for example, are the root cause of URL-guessing attacks. In such attacks, users modify URLs to display information of other user accounts.
- *Coarse authorization control.* Cloud services' management interfaces are particularly prone to offering authorization control models that are too coarse. Thus, standard security measures, such as duty separation, can't be implemented because it's impossible to provide users with only those privileges they strictly require to carry out their work.
- *Insufficient logging and monitoring possibilities.* Currently, no standards or mechanisms exist to give cloud customers logging and monitoring facilities within cloud resources. This gives rise to an acute problem: log files record all tenant events and can't easily be pruned for a single tenant. Also, the provider's security monitoring is often hampered by insufficient monitoring capabilities. Until we develop and implement usable logging and monitoring standards and facilities, it's difficult—if not impossible—to implement security controls that require logging and monitoring.

Of all these IAAA vulnerabilities, in the experience of cloud service providers, currently, authentication issues are the primary vulnerability that puts user data in cloud services at risk. [5]

Provider

Vulnerabilities that are relevant for all cloud computing components typically concern the provider—or rather users' inability to control cloud infrastructure as they do their own infrastructure. Among the control challenges are insufficient security audit possibilities, and the fact that certification schemes and security metrics aren't adopted to cloud computing. Further, standard security controls regarding audit, certification, and continuous security monitoring can't be implemented effectively.

Cloud computing is in constant development; as the field matures, additional cloud-specific vulnerabilities certainly will emerge, while others will become less of an issue. Using a precise definition of what constitutes a vulnerability from the Open Group's risk taxonomy and the four indicators of cloud-specific vulnerabilities we identify here offers a precision and clarity level often lacking in current discourse about cloud computing security.

Control challenges typically highlight situations in which otherwise successful security controls are ineffective in a cloud setting. Thus, these challenges are of special interest for further cloud computing security research. Indeed, many current efforts—such as the development of security metrics and certification schemes, and the move toward full-featured virtualized network components—directly address control challenges by enabling the use of such tried-and-tested controls for cloud computing. □

References

1. *ISO/IEC 27005:2007 Information Technology—Security Techniques—Information Security Risk Management*, Int'l Org. Standardization, 2007.
2. P. Mell and T. Grance, "Effectively and Securely Using the Cloud Computing Paradigm (v0.25)," presentation, US Nat'l Inst. Standards and Technology, 2009; http://csrc.nist.gov/groups/SNS/cloud-computing.
3. European Network and Information Security Agency (ENISA), *Cloud Computing: Benefits, Risks and Recommendations for Information Security*, Nov. 2009; www.enisa.europa.eu/act/rm/files/deliverables/cloud-computing-risk-assessment/at_download/fullReport.
4. L. Youseff, M. Butrico, and D. Da Silva, "Towards a Unified Ontology of Cloud Computing," *Proc. Grid Computing Environments Workshop* (GCE), IEEE Press, 2008; doi: 10.1109/GCE.2008.4738443.
5. E. Grosse, "Security at Scale," invited talk, ACM Cloud Security Workshop (CCSW), 2010; http://wn.com/2010_Google_Faculty_Summit_Security_at_Scale.

***Bernd Grobauer** is a senior consultant in information security and leads the Siemens Computer Emergency Response Team's (CERT's) research activities in incident detection and handling, malware defense, and cloud computing security. Grobauer has a PhD in computer science from Aarhus University, Denmark. He's on the membership advisory committee of the International Information Integrity Institute. Contact him at bernd.grobauer@siemens.com.*

***Tobias Walloschek** is a senior management consultant at Siemens IT Solutions and Services GmbH. His research interests are cloud computing security and business adoption strategies. Walloschek has a bachelor's degree in business administration from the University of Applied Sciences in Ingolstadt, Germany. He is a Certified Information Systems Security Professional. Contact him at tobias.walloschek@siemens.com.*

***Elmar Stöcker** is a manager at Siemens IT Solutions and Services GmbH, where he's responsible for the portfolio strategy and governance of the professional services portfolio; he also leads the cloud computing security and PaaS activities. Stöcker has a master's degree in computer science from RWTH Aachen, Germany. Contact him at elmar.stoecker@siemens.com.*