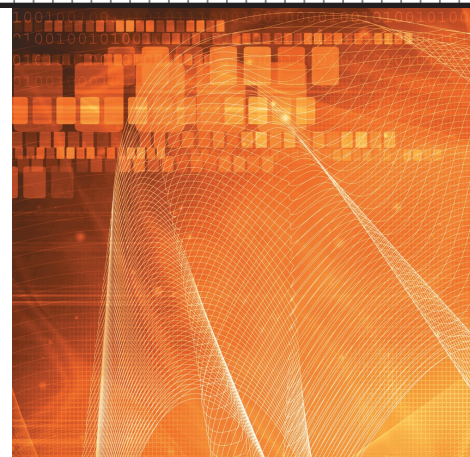


Cloud Computing and the Common Man

➔ John Viega, McAfee



The cloud offers several advantages, but until some of its risks are better understood many major players might hold back.

Cloud computing is one of the biggest technology buzzwords these days. It operates from the idea that work done on the client side can be moved to some unseen cluster of resources on the Internet.

In many respects, cloud computing is an old idea. In the 1990s, for example, Sun tried to promote thin clients that performed all computing work on the Internet. More recently, we've heard about application service providers (ASPs) and software-as-a-service (SaaS), a term popularized with the rise of Salesforce.com.

CLOUD SYSTEMS TODAY

Currently, developers work with three major cloud systems categories: *software-as-a-service*, *platform-as-a-service*, and *infrastructure-as-a-service*.

SaaS

In the SaaS model, the user buys a subscription to some software product, but some or all of the data and code resides remotely. For example, Google Docs offers an alternative to Microsoft Office that stores documents on Google's server. It doesn't keep any code on the client machine, even though some code might execute on the client temporarily. For example, Google Docs relies on JavaScript,

which runs in the Web browser. In this model, applications could run entirely on the network, with the user interface living on a thin client.

PaaS

From the consumer's viewpoint, PaaS software probably resembles SaaS, but instead of software developers building the program to run on their own Web infrastructure, they build it to run on someone else's. For example, Google offers Google App Engine, a service that lets development organizations write programs to run specifically on Google's infrastructure.

IaaS

Similar to PaaS, IaaS lets the development organization define its own software environment. This basically delivers virtual machine images to the IaaS provider, instead of programs, and the machines can contain whatever the developers want. The provider can automatically grow or shrink the number of virtual machines running at any given time so that programs can more easily scale to high workloads, saving money when resources aren't needed.

CLOUD SYSTEMS USERS

From the end user's viewpoint, there's usually little difference

between these three models. The system's security depends on things mostly out of the user's control, such as ensuring the IT environment's security from outside attackers who, for example, attempt to break into backend systems and steal data. Likewise, security from within the environment must be ensured if, for example, someone uses a flaw in an application to scan other users' data or tamper with other applications hosted in the same environment. The authentication and encryption methods used, as well as the plaintext protocols with passwords, pose a risk factor for all involved.

Danger: Open environment

In this case, the IT environment isn't something the software developer has much control over. The cloud provider must be upfront about its policies, and application developers must do what they can. For example, when using IaaS, it's best practice to create a virtual machine image that runs with no unneeded functionality and to have examples use encryption to talk to each other, in case another customer might be able to intercept such transmissions at the network level.

If the cloud system is well designed, it offers the significant advantage that all the interesting code an intruder

might want to exploit will reside on the server side instead of being downloaded to the browser. Although the intruder can still attack the servers, it can't get hold of the code unless it can access obvious flaws from the user interface or leverage brute-force testing techniques to find the problems. It's also easy for the IT staff to monitor these techniques, which makes the application more defensible because they can find flaws before the intruders do.

Compare this situation to a typical architecture in which someone can buy a copy of the server-side application to host on his own network environment. Anyone can buy the application, including intruders. While the vendor typically won't ship source code, intruders will at least have access to the binary code, which they can read—although not as easily as if they had the source code. Thus, in an SaaS model, access to the binary code for server-side components should be extremely limited.

It's still possible for developers to put important things on the client side that should be on the server side. The developer must assume that intruders will have complete access to everything on the client, no matter what countermeasures have been put in place. For example, if the client is responsible for constructing and validating database queries and the server executes them blindly, the intruder will always be able to modify the client-side code to do whatever he has permission to do with the back-end database. Usually, that means the intruder can read, change, or delete data at will.

Lean and mean approach

Yet because the attacker has so much less information, it seems quite justifiable to rely on an incredibly modest application security program. The exact scope depends on the software developer's specific requirements, but for many organizations the cost-effective reality

is, "Hire someone to do fairly cheap security testing," invest some modest resources in the authentication and encryption coding everyone obviously should be doing, and quite possibly leave it at that. Developers can most likely skip training and code review altogether if nobody is championing those activities.

Whenever I suggest caution about spending on application security, people often seem shocked because I've coauthored so many books on this topic, including the first, *Building Secure Software* (Addison-Wesley Professional, 2001). I've been getting that reaction again when talking about application security in the cloud.

The developer must assume that intruders will have complete access to everything on the client, no matter what countermeasures have been put in place.

And yes, this approach will demonstrably not lead you to the most secure solution possible, but business is about maximizing profit, not security. This is the right place to set the dial on the risk knob for most companies doing business in the cloud.

However, in several use cases this advice might not apply. For example, if a company deploys its solution in the cloud but sells the same code base onsite, an intruder again has access to that code.

CLOUDED CONCERNS

Beyond generic concerns regarding the cloud approach, each of the three models has its own security concerns.

SaaS concerns

With SaaS, users must rely heavily on their cloud providers for security. The provider must do the work to keep multiple companies or users

from seeing each other's data without permission. In addition, the provider must protect the underlying infrastructure from break-ins and generally has responsibility for all authentication and encryption. It's difficult for the customer to get the details that help ensure that the right things are being done. Similarly, it's difficult to get assurance that the application will be available when needed.

PaaS concerns

With PaaS, the provider might give some control to the people building applications atop its platform. For example, developers might be able to craft their own authentication systems and data encryption, but any security below the application level—such as host or network intrusion prevention—will still be completely in the provider's hands. Usually, the provider will offer little or no visibility into its practices. Plus, the platform provider must be able to offer strong assurances that the data remains inaccessible between applications. Large banks don't want to run a program delivered in the cloud that risks compromising their data through interaction with some other program.

IaaS concerns

With IaaS, the developer has much better control over the security environment, primarily because applications run on virtual machines separated from other virtual machines running on the same physical machine, as long as there is no gaping security hole in the virtual machine manager. This control makes it easier to ensure that developers properly address security and compliance concerns—with the downside that building the application can be more expensive and time-consuming.

Backing up data poses another concern. Even though some providers do their own backups for the customer, much can still go wrong. Maybe they increase their prices and make it difficult to get data off their network.

Sometimes companies go into Chapter 7 bankruptcy suddenly, leaving customers with no data access.

DATA AND SECURITY

If the organization uses a cloud-based solution, it should maintain its own data backups in addition to those saved by the cloud provider. This is generally far easier with IaaS than with the other two models.

Authentication credential management poses another cloud security concern. For example, Amazon provides IaaS services through its elastic cloud computing (EC2). These account holders receive public-key credentials for connecting to the servers and managing their applications, particularly virtual machine examples that run as the client's user.

Unfortunately, Amazon currently lets a user have only one set of credentials per account. This makes it difficult to run applications in multiple pieces, with each piece administered separately either by business function or geography. This architecture violates the security principle of least privilege: If one administrator's credentials are compromised, the customer's entire site is at risk.

While there are many cloud security concerns, most have straightforward solutions. The cloud obviously has many advantages, such as cost sharing across companies, which can make it more cost-effective than purchasing a complete infrastructure. It also helps handle

situations in which an application becomes popular and must scale quickly.

The cloud also offers security advantages. For example, intruders have no access to the source code, and providers often work hard to provide clean, unbreakable barriers between customers. Security can differ greatly from application to application, from platform to platform, and from provider to provider, however. Yet on the whole, the cloud holds much promise for better security.

RISKY BUSINESS?

To many, however, the cloud still seems risky. With people now having more data and code residing on the same few sites, such sites become more tempting targets.


Those who own the applications running on these sites must worry about all the problems that other development organizations worry about—except that the consequences could be much worse for the big sites because so many more people and companies can be placing their data at risk.

Developers writing SaaS applications must avoid the application flaws that might appear when one customer's data becomes available to another's or an oversight or intruder uncovers data from multiple customers. IaaS providers must make sure their customers are protected from each other: If a site suffers a security breach, the

site's other customers shouldn't be endangered by it.

Many companies are concerned that their applications and data might reside in a shared space alongside other applications and data. For example, several large financial organizations and some government agencies have indicated that they will not consider PaaS or IaaS anytime soon because they have no good way to quantify their risks. They are concerned about applications running on a platform that wasn't designed with security in mind. When those applications are attacked, can the underlying platform prevent access to other applications' data? In theory, virtual machines might be able to address these kinds of problems in an IaaS model, but in practice, plenty of security problems have arisen in virtual machine technologies themselves.

While the cloud offers several advantages, until some of the risks are better understood, many of the major players will be tempted to hold back. Requiring some standardization in the security environment—and third-party certifications to ensure that standards are met—will likely be necessary as well.

Ultimately, though, people will agree that the cloud is cool. It can help companies save costs on applications, hardware, and people. But while cloud providers currently enjoy a profound opportunity in the marketplace, they must ensure that they get the security aspects right, for they are the ones who will shoulder the responsibility if things go wrong. 

John Viega is the CTO of McAfee's SaaS Business Unit. Contact him at viega@list.org.

Join the
IEEE Computer Society

www.computer.org

Editor: Jeffrey Voas, Science
Applications International Corporation;
j.voas@ieee.org