Linux security in the cloud

Cloud Concerns

Although you give up control of the underlying infrastructure when you use cloud computing, you can still maintain some control over security. *By Kurt Seifried*

Love it or hate it, the cloud is here to stay (a safe bet given that COBOL and Fortran applications are still around 50 years later). The good news is that Linux is uniquely suited for cloud computing for both providers and users for a number of reasons.

One of the biggest challenges of cloud computing – and one reason it's so attractive to some users – is the lack of control over the underlying hardware, network infrastructure, storage, and so on. If you're in the cloud, you are probably sharing that infrastructure with others (e.g., on EC2, I have no idea with whom I am sharing servers and storage). This shared tenancy is great for providers, because they can extract every penny from their hardware and even oversell capacity in some cases, but it can still be a problem for users.

The range of uses for cloud computing is also growing; computationally intensive tasks are suddenly cheap and affordable for everyone (e.g., WiFi cracking using ranked NVidia Tesla GPUs). Service providers like Google Apps allow you to deploy applications that can handle sudden spikes in traffic with no effort on your part, and you now have access to the same services as the big boys.

To deal with users' different needs, providers must be sure that customers are properly segregated. For IaaS (Infrastructure as a Service), things aren't too bad; most solutions like VMware, Xen, and KVM have a solid history at this point and keep things separated by default.

The PaaS (Platform as a Service) guys like Google Apps have things a little harder. Very few programming languages can be deployed in a "safe" manner if you have users who are potentially hostile. Witness Java, which has had a sandbox from the very beginning. Although it was designed to be secure, flaws are found consistently. Finally, the SaaS (Service as a Software) guys are just basically deploying applications and shoving as many users as possible onto them. I hope they're doing so without including any SQL injection vulnerabilities.

Data Encryption

Data that lives in the cloud is different in several ways, but the biggest difference is that it's living on systems you do not control. What's worse, it's usually living on systems that are shared, so if one customer does something naughty and attracts the attention of law enforcement, for example, you could suffer the consequences of an overly broad search warrant, or the provider might not be able to service the request completely without handing everything over [1].

Other concerns can arise if a cloud provider sends out a defective hard drive (with your data on it) to be repaired or sells old equipment without first wiping it – assuming it can be wiped [2]. Or, your provider might be using co-location space in multiple locations in a way that seems to be secure but actually results in your data being sent over insecure networks.

Even if the provider has a security policy that covers this problem, the policy likely will not survive a bankruptcy. For example, toysmart.com promised "never" to sell private information, but when they went bankrupt, they did. Then, the Federal Trade Commission got

KURT SEIFRIED

Kurt Seifried is an Information Security Consultant specializing in Linux and networks since 1996. He often wonders how it is that technology works on a large scale but often fails on a small scale.

involved, and the company came to an agreement that "forbids the sale of this customer information except under very limited circumstances" [3].

The first step to protect yourself is to encrypt your data at rest. Wherever your data is stored, it should be encrypted, and this means much more than just disk storage and databases. If you use message queues to handle job requests, the data might be written to disk (e.g., beanstalkd's binary log). If you use NoSQL back ends to store data, data might be written to disk either intentionally or through swap space.

Fortunately, Linux has mature disk encryption support including whole-disk encryption [4] that is available by default. The downside, however, is that not all providers give true console access, so you might not be able to encrypt the entire system (because it will require a password to boot up). So, before committing to using whole-disk encryption as part of your security solution, you need to make sure it's actually possible.

Of course, data transmitted between systems must also be encrypted. Through the use of VLANs and tunneling, you might think two systems reside next to each other, when in reality they are in different parts of the data center or even in different buildings. I wonder how long it will be until we hear reports of some cloud provider finding out that their routing and switching infrastructure has been compromised and was used to listen to all the customers. Depending on your application and design, you have several options that can help, including IPsec, for encrypting everything between systems, or technologies like SSL and SSH, for encrypting specific sessions and data.

Unfortunately, you are still exposed when your information is stored in system memory. In the past, this wasn't much of an issue because you owned the hardware or were the only person using it. But, with the advent of virtualized systems, the hypervisor (the underlying operating system) is always able to view and dump the contents of memory – otherwise it couldn't suspend a machine, for example.

Therefore, extremely sensitive data like encryption keys, private SSL certificates, and credit card data will be accessible to some degree, so if you don't trust your provider, don't do anything sensitive in the cloud.

Live Patching

Many cloud computing systems are built to be as stateless as possible, so failures won't bring things down, and, in an extreme case, users even embrace failure [5]. But, certain components must be reliable, such as data storage (which can be replicated), message queues, file storage, and so on. Tools like Ksplice [6] give Linux an advantage over other operating systems because, at least in theory, you can have virtually unlimited uptime and keep things patched and up to date.

For providers, the hypervisor systems must be as reliable as possible. Having to transfer virtual machines from one machine to another and patch and reboot an entire infrastructure is probably something they'd like to avoid. Ksplice is currently available for the Linux kernel, and with any luck, it could be extended to popular applications in the future.

Security Policy Framework

The military tends to be pretty paranoid about communications secrecy and computer security, probably because they know how effective it is to read an attacker's communications (e.g., the German Enigma in World War II). Fortunately for us, the NSA sponsored the development of SELinux [7], which allows Mandatory Access Controls (MAC) to be implemented. Like most people, I love the idea of SELinux; you can lock down applications and services so that even if an attacker gains access to that application or service, they won't be able to get any farther than is allowed.

In practice, however, like many administrators, I've had to disable SELinux or set it to permissive mode because it breaks things like Samba (if you share data from non-standard locations) and various web applications. However, for service providers, especially PaaS providers, tools like SELinux are invaluable.

The Bad News

None of these security measures matter much for customers if you have a hostile provider. Most forms of encryption rely on protecting a secret key. A hostile provider can simply dump the contents of memory to find your encryption keys. Or, they can use the span ports on their switches to mirror all traffic to their own systems for analysis. Once they have your encryption keys, they can quickly decrypt all the information, such as credit card numbers. Alternatively, an attacker could break in and use these techniques to attack many clients.

Providers must guard against such attacks as well. A hostile user might attempt to gain access either by breaking into accounts or by using stolen credit cards to buy access. Most traditional security methods rely on strong perimeters, and I think the best analogy still is "hard and crunchy outside, soft and chewy inside."

The Good News

Despite the challenges I've mentioned, the good news is that Linux provides the foundation necessary to build on securely. Also, SELinux has been available for more than a decade now, and most of the bugs have been shaken out. Now, it's just a matter of using the available tools properly to maintain some measure of security in the cloud. ■■■

INFO

[1] Cloud computing, law enforcement, and business continuity: *http://berkeleyclouds.blogspot.com/2009/04/cloud-computing-law-enforcement-and.html*

[2] The Non-Volatile Systems Laboratory – Sanitizing SSDs: *http://nvsl.ucsd.edu/sanitize/*

[3] FTC Announces Settlement with Bankrupt Website, Toysmart.com: *http://www.ftc.gov/opa/2000/07/toysmart2.shtm*

[4] "Atick Security" by Kristian Kißling, *Linux Magazine*, July 2009: *http://www.linux-magazine.com/Issues/2009/104/Encrypting-USB-Sticks*

[5] 5 Lessons We've Learned Using AWS: *http://techblog.netflix.com/2010/12/5-lessons-weve-learned-using-aws.html*

[6] "Upgrade 2.0" by Kurt Seifried, *Linux Magazine*, October 2009: *https://www.linux-magazine.com/w3/issue/107/066-067_kurt.pdf*

[7] "Mandatory Access Control (MAC) with SELinux" by Thorsten Scherf, *Linux Magazine*, June 2008: *http://www.linux-magazine.com/w3/issue/91/036-043_selinux.pdf*