Develop secure cloud-aware applications

Tackle the security challenges of cloud-aware apps with these best practices

Robi Sen

May 20, 2015

Cloud-aware application architectures and designs are becoming increasingly prevalent as developers and organizations recognize their increasing value. Cloud-aware applications tend to be very elastic and easy to scale, faster to develop, and more affordable. Yet while cloud-aware applications have numerous benefits, their architectures can create numerous security challenges for the unaware. This article describes some of the security challenges created by cloud-aware applications and how you can address some of them by following various best practices.

Characteristics of cloud-aware applications

Cloud-aware applications are very simple applications that are aware of the cloud services and infrastructure they are leveraging, allowing for seamless and automated scaling, draw down, failover, and more. Unlike simple web applications deployed in the cloud that are essentially a multi-tiered software stack deployed on a virtual hardware stack, cloud-aware applications tend to have a more distributed and stateless model that makes use of services and actor models of computation. For a high-level comparison of the characteristics of traditional n-tiered applications and cloud-aware applications, see Table 1 and Table 2.

### Table 1. N-tiered application characteristics

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronous</td>
<td>Request and reply interactions are synchronous.</td>
</tr>
<tr>
<td>Centralized state</td>
<td>Generally state is preserved in a single database.</td>
</tr>
<tr>
<td>Clustered</td>
<td>Uses numerous clustered services working together to allow for scaling.</td>
</tr>
<tr>
<td>Siloed architecture</td>
<td>N-Tiered applications tend to have vertically siloed architectures where services and data stay within the environment.</td>
</tr>
</tbody>
</table>

### Table 2. Cloud-aware application characteristics

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resilient to failure</td>
<td>Resiliency is designed into the application. Failures in cloud infrastructure are handled fluidly without interruption of service.</td>
</tr>
</tbody>
</table>
As you can see from Tables 1 and 2, cloud-aware applications are designed to leverage numerous services and endpoints unlike more traditional siloed applications that run on the cloud. (See the Resources section at the end of this article for a link to an Open Data Center Alliance paper on "Architecting Cloud-Aware Applications.") As a result, this can create security issues since your data or applications' function might rely on third-party or partner services over which you have no control. While there is an inherent risk of cloud services, it's especially prevalent in cloud-aware applications where your data may be exposed at rest, in transit, and potentially in use. Furthermore, in cloud-aware applications, your data may end up being managed or used by third parties. Partners may have poor security and may unintentionally release data or be exploited by attackers, which is unfortunately a common occurrence even with some of the most respected cloud applications and services. Another issue is that even if you use SSL, your data can be compromised since it is sent over the wire. This is especially true if your certificates or partners are poorly configured. Data can also be exploited, since your data is in use by cloud environments that are potentially compromised, leaking your encryption keys as well as important data. Finally, due to the automated and federated nature of cloud-aware applications, they are often vulnerable to having their API endpoints exploited unless their access is controlled.

### The need for encryption in cloud-aware applications

With rare exceptions, one of the most important assets for any company is its data. Your data may take the form of financial information, proprietary sales information, marketing information, healthcare information, intellectual property (IP), and more. Losing your data could negatively affect operations and potentially shut down your organization. Having that information stolen or released to the general public could damage your reputation, lead to legal issues, and negatively impact your brand, as seen in the recent attacks on Sony Pictures. So, protecting your data can and should be something of major business importance, regardless of whether you're a two-person...
start up or a multinational corporation. To protect your data in the cloud, you need to do many things—from controlling access to hardening your operating systems. Numerous data breaches have shown how important it is to take crucial steps like creating policies on how data is moved and accessed, and hardening your network. No matter how many steps you take to secure your system, chances are someone will find a way to compromise your data. New threats are emerging all the time, ranging from more conventional insider threats to nation-state attacks to fundamental flaws in software that surprise even security experts and totally compromise systems. Indeed, as I have already pointed out, cloud-aware applications should have security built in with encryption both at rest and in transit.

The three states of encryption and the cloud's shared risk model

Most security experts and encryption experts classify encryption into three states. The first is *data at rest*, which describes the state of your data when it’s on your file system not being used, such as data in files and databases. The second is *data in motion*, such as when you move data from one server to another. All of the bits going over the network are considered data in motion. And finally we have *data in use*, which is data that’s being used in your CPU or RAM. It is important to understand the three states of encryption so as to better understand your vulnerabilities in the cloud.

Cloud-aware applications create unique security challenges in that both Infrastructure as a Service (IaaS) providers and Platform as a Service (PaaS) providers make use of a shared-risk model. This means that your cloud partners will usually provide a protected data center, some network security, high availability storage and hardware, and other services—but you are expected to meet certain security considerations yourself. These include identity access management and control, intrusion detection, monitoring, hardening of your operating system, patching of your systems, and of course encryption. Further, because of the shared risk model, since you do not have physical control over your machines and who might touch them, there is a potentially greater risk from potential exploitation by vendor employees or third parties who might physically access your systems. There is also the risk (although currently it has not appeared) of other tenants within a cloud system attacking your systems or infecting them. As such, the Cloud Security Alliance specifically calls for encrypting all sensitive data on your cloud systems as well as securing all your data in transit.

Policy before technology: compliance and control

Before getting into what approaches to use to secure your cloud-aware applications, you should consider your organization's security drivers so that you can come up with clear and well-defined polices around securing your applications. Those same policies need to also drive your Secure Software Development Lifecycle (S-SDLC) to not only create a more secure application, but to manage it throughout its lifecycle. Some places to start, when developing your policies are:

1. Make sure all major stakeholders who touch your systems are involved, from HR to operations to finance to your senior IT management. Often organizations forget to include different departments, leading to issues like we saw with Sony Pictures where contracts, which contain highly sensitive data, were available in plain text on servers.
2. Design a policy to determine who has access to your system and implement a robust identity management solution—not just for your internal users, but also for users of exposed services and end points. Your solution must support the ability to generate access logs and role-based access, as well as encryption and key management. Finally, think early on about multi-tenancy, since often even single-tenant applications need to eventually support multi-tenancy at some point.

3. Decide who will have access to your encryption keys at a high level and where they will reside. While specific implementations will dictate how you manage the keys, you should first have a good sense of who should be able to access them.

4. Carefully consider what regulations you must comply with, such as PCI, FISMA, HIPPA, and GLBA. Most regulations require specific levels of encryption or dictate how encryption keys are managed. Other regulations require things like automatic encryption of employee information or hashing of credit card numbers.

5. Work carefully with your stakeholders to figure out what data is important to your company. This is much harder than it sounds. While companies with sensitive intellectual property or proprietary information should definitely encrypt their data, many companies have information that may not seem sensitive at first assessment. For example, many companies believe that customer emails or personal information are not important to anyone. Yet criminals highly value such information for reselling to spammers, phishers, or identity thieves. Losing such data can seriously harm an organization’s brand and the trust of its customers.

6. Decide on a data breach policy early on. Odds are, at some point, you will have a data breach or leakage. If you have followed the above steps, then you will already have a good sense of which data is critical and which is not. You will also know what regulatory compliance you must conform to and let users understand if data, such as personal information, has been leaked and be able to respond according to whichever regulations (such as HIPPA or PCI) govern your application.

With these high-level guidelines and an understanding of the cloud’s shared-risk model, you should now have a good idea of what you need to secure. Next, I’ll show you the methods of encryption at various levels and securing application end points.

**Data at rest**

There are two basic approaches to securing data at rest: **Full Disc Encryption (FDK)** involves encrypting everything on a drive, while **File Level Encryption (FLE)** involves encrypting only sensitive files or data. These approaches are pretty self explanatory. With FDK, everything is encrypted on the disc. It has many advantages, including greater simplicity in management since everything is automatically encrypted. Some downsides are that it can be computationally expensive, affect system performance, and sometimes take up too much storage space. That being said, most companies that do use FDK say it is worth the expense. In general, though, most cloud service providers make the setup and deployment of FDK easy with various services or APIs. If you have a lot of sensitive data in your application, or at a specific tier such as your database, FDK makes a lot of sense. If only small pieces of information are sensitive, then FLE makes more sense. Most cloud services and software stacks for cloud-aware applications, such as Cloudant, can easily store encrypted binary data using FLE.
Another thing to consider with data at rest is anti-replication technologies or Digital Rights Management (DRM), with various techniques such as tagging or integrating tools like Adobe’s DRM. Depending on your needs for such tools, you can greatly reduce your chances of data leakage and exposure due to insider threats. This is important with cloud-aware applications where data may be constantly moving into and out of different end nodes. In many cases, you may want to use anti-replication technologies along with both FDK and FLE. For example, you might have an application with data that's encrypted in a database. Your application also generates reports, using a report generation service, that are sensitive and need to be encrypted; these are stored in a data storage service that supports FDK (such as Glacier or Nearline). In this case, encrypted files and binary data are stored on a totally encrypted PaaS archival solution. This common setup leads me to my next point: Where should you do your encryption and where do you store your keys with a cloud-aware application that shares and uses data from numerous end points?

Most cloud-aware applications leverage numerous external services. For example, your cloud-aware application might leverage a messaging queue service, an image cache service, a data storage platform, an archival service, and other services, all coordinated via a service orchestration platform. Your system would send and consume numerous messages containing important data over services you have little control over. For this reason, your data must be encrypted all of the time. Unfortunately, too many organizations move data from service to service without encrypting their data or files, trusting that each service will protect the integrity of their data. Such practices have resulted in large companies leaking user data through third-party partners who do not store data securely or implement poor data in transit mechanisms. As such, you should encrypt all sensitive data and leave it encrypted until it needs to be decrypted. This can be complex, requiring you to manage encryption keys yourself. Fortunately, numerous companies such as KeyNexus or CloudCipher also provide products and services that allow you to use your own keys or PKI infrastructure to always maintain control over your data. Many IaaS providers also provide APIs that enable you to manage encryption within their services using your own encryption keys. When selecting a provider, you should understand your data at rest needs, whether you need to control your own keys, and whether your cloud application stack will support it. This is often a critical decision.

Data in motion

In general, any time your application needs to send or receive sensitive data, you want to not only encrypt the data that's being sent, such as a file, but also encrypt the communication/data transfer. For example, if your site accepts customer information or credit cards, you should make sure these communications are encrypted, since the data is transferred from the user's web browser to your cloud application. In this situation, you need to use HTTP Secure protocol (HTTPS) instead of HTTP. HTTPS makes use of SSL/TLS, a protocol for encrypting data in motion. Other Internet protocols can make use of SSL/TLS to encrypt data in motion, and in cloud-aware applications there are best practices for creating Virtual Private Networks between each service, as well as between your systems and your application whenever possible. This reduces the risks of an attacker compromising part of your system and being able to collect data from the rest of it. You should protect data in motion with a reputable SSL certificate. If you are new to using SSL, review the excellent cheat sheet on SSL from the Open Web Application Security Project (see Resources).
Lastly, many organizations make the mistake of trusting solely in SSL and not using encryption at rest. This is a huge mistake. There are numerous, although potentially complicated, ways to exploit HTTPS sessions. This is because of poor configuration, setup, or underlying flaws in software and their SSL implementations like we saw with the Heart Bleed bug. For this reason, it is important to, whenever possible, encrypt sensitive data before it is transmitted over the network, as well as when you transmit it.

Data in use

In general, the encryption for data in use is very rare. This is primarily because it is not considered necessary for organizations that host or control their own data centers where they feel comfortable. This is so that no one can easily access a server while it is running and inspect its RAM for unencrypted data or actual encryption keys. Yet with cloud-based applications, you have no way of confirming the physical integrity of the systems your application is on. Encryption for data in use suddenly becomes very relevant. Indeed, as of 2012 the Cloud Security Alliance started recommending encryption for data in use as a best practice.

Currently, no IaaS or PaaS vendors offer encryption for data in use, but companies like PrivateCore and Vaultive offer systems that enable data in use within your cloud environment of choice. For most cloud-aware applications, data in use is most likely overkill, but for organizations that have a high risk of attack by criminal organizations and state actors, it should be a serious consideration. In part, this is because some attackers specifically use techniques that are designed to extract encryption keys or data direction from RAM.

Securing end points

Another major characteristic of cloud-aware applications is the use of APIs, both internally and externally, to design flexible and agile applications. Although many developers neglect to do so, it is a best practice to secure end points when using APIs or when wrapping any end point into a good abstracted API. The reason for this is malicious actors often attack end points without security as a form of denial of service. More savvy ones will exploit weak authorization on end points to steal data or manipulate applications in unintended ways. As such, all end points should be secured with some form of authorization mechanism such as OAuth or SAML. You should also be able to restrict access by region or IP subnet, as well as log access to your APIs by users or systems.

Conclusion

Developing secure, cloud-aware applications is a complex subject that is greatly affected by the specifics of your business needs and regulatory environment. That being said, we have looked at some of the major items that should allow you to start generating policies for securing your application, deploying encryption, using access controls, and securing your end points. I have also shown you some of the best practices for encrypting your data, and provided resources to help you create a plan for implementing cloud-aware security best practices. The Resources section provides additional information on best practices for cloud-aware applications.