An essential guide to app security in the public cloud: 3 solid approaches

Edited 2019 by Dr. Bernard Bowitz

IT security has been focused on infrastructure for the past 30 years. All you needed to do was simply place layers of security around your applications and data and all would be well. But even if that ever worked, it certainly doesn't anymore. With hackers testing the limits of security systems on a daily basis, the pressure is on development teams to provide better security at the application layer as well. And moving your apps to the public cloud is yet another game changer.

The cost of not integrating application security is steep. Home Depot, for instance, was involved in a headline-making cyberattack that targeted its payment terminals. The security breach left approximately 56 million credit and debit card numbers exposed. Multiply that by Ponemon Institute's estimated cost of $194 per compromised record in the average data breach and you can see the enormity of the risk. Those costs include investigation, remediation, notification to individuals, identity theft repair and credit monitoring, regulatory fines, disruptions in normal business operations, lost business, and related lawsuits. Bottom line: The dollars you spend to protect against breaches that could take down your entire business could be the best security investment you’ll make this year.

On the positive side, moving your apps to the public cloud does not necessarily mean that you're giving an inch on security. Indeed, the approaches and mechanisms available to developers and administrators in the public cloud are often *better* than the tools and methods you use within the enterprise.

However, in the context of the cloud, you need to look at security as a systemic concept. Long gone are the days when you could just build fences around applications and data and call it a day. Just ask Home Depot, Sony, Target, and other victims of major breaches if traditional approaches to application and data security worked for them. **Here's what does work.**

Cover your basics first

Developers who build applications to run in public clouds, or migrate and refactor applications for the cloud, should focus on a few basic security concepts, including authorization, auditing, confidentiality, and integrity.

- **Authorization** addresses the question, what are you authorized to do? This process governs the resources and operations that the authenticated user has permission to access. Resources include files, databases, tables, rows, and so on. Users can either access the entire resource, a part of the resource, or none of it.
- **Auditing and logging** guarantees that a user cannot deny an operation or initiate a transaction without the activity being recorded. In cloud applications, this means you log the use of the application or the data store for compliance or other legal reasons. Auditing and logging also lets you spot patterns of use that may indicate a breach and take defensive action.
- **Confidentiality**, or *privacy*, is the process of making sure that your data remains confidential. This means ensuring that the data cannot be viewed by unauthorized users or eavesdroppers monitoring the flow of traffic across a network. Use encryption to enforce

confidentiality, and consider using it whenever data is at rest or moving within a system.

- **Integrity** is the guarantee that data is protected from accidental or malicious modification. For example, ensuring that a hacker cannot take money from your bank account without you receiving a notification.

Three security approaches you need for public cloud

Now that we've reviewed a few basics, let’s focus on dealing with security at the application level on a public cloud. Here are three ways to make your public cloud applications more secure.

1. Focus on the data

Application developers should have a laser focus on data security, because that's where most attacks occur, but don't let your applications give hackers a path to that data. Think of data security in the cloud as a series of levels:

- **The platform level.** This is the operating system of the machine instance, including items such as data files. Inadequate protection of the platform is a fundamental flaw that most application developers fail to consider. They may protect access to the data but not the database itself, which is exposed in the platform. In order to deal with this vulnerability, make sure you encrypt the data. That way, if someone copies the data files, they'll be useless. While this is the best approach, it sometimes can cause performance problems, so many developers prefer not to use it.

- **The database level.** Most databases have their own security systems, and when leveraging databases within pubic clouds, it’s a good idea to use them. These include data encryption, as well as the ability to allow only certain users to access certain parts of the database based upon authorization level. Make sure you select a cloud-based database that offers these security features.
- **The application level.** Since applications are authorized to read and write to a database, you need to focus on security there as well. This means setting up identity-based access to the application itself and monitoring activity to ensure that the user does not display hacker patterns, such as coming in from an unknown IP address, missed log-ins, and so on.

2. It’s all about identity

Use identity and access management (IAM) technology to initiate, capture, record, and manage user identities and related access permissions. IAM ensures that access privileges are granted according to policy set by both the developers and security administrators. Moreover, IAM verifies that all individuals and services are properly authenticated, authorized, and audited.

Cloud application developers must understand IAM. Don’t just attach it to resources such as data and services, but build it right into your applications. IAM systems include APIs that you can use for such things as rechecking that the user is authorized to access the application, the platform, the services, and the data. Any of these can be de-authorized at any time, and so it’s never an all-or-nothing approach.

IAM systems should automate the initiation, capturing, recording, and management of user identities that use a centralized directory service. This central directory prevents credentials from ending up recorded haphazardly in files and sticky notes, which is the way humans respond to security systems that are too intrusive and complex. It’s your job as the developer to ensure that your cloud application is easy to use as well as secure.

3. Move from DevOps to DevSecOps

The rise of DevOps, and the use of cloud-based platforms as the target platform for applications, provides a lot of additional exposure for security breaches, but it also presents opportunities to improve security. You need to focus more on DevSecOps, or development security operations, where you deal with testing security within the DevOps processes. DevSecOps means that when you do continuous testing, you include continuous security testing as well. You must constantly check applications for the proper use of IAM services, encryption, and other security processes that should be built into the application and make sure they're all functioning correctly.

Also, after you stage and deploy an application in the cloud, continue with your security operations focus during the continuous operations phase. Review operations of IAM and encryption within the applications, data storage, and the platforms to ensure that you're as protected as you should be, and that all protections are active and functioning correctly.

Your approach to DevSecOps will vary greatly depending upon your applications, your industry, and the brand of public cloud on which you deploy. The best practices here are to continually improve your approach to security, and be proactive in monitoring applications in operations to look for activities that could be leading up to attacks or represent attacks that are underway.

Public cloud best practice: Oversecure your apps

IT executives often believe that they must give up security to get value out of public clouds. That’s not the case. Security is related to the approaches and technology you leverage, as well as your commitment to bake in security at many levels. Most cloud-based applications are more secure than traditional applications for this reason.

That said, it’s your responsibility as the developer to ensure that security is systemic to your cloud-based applications at the application, data, and platform levels. This approach to oversecure cloud applications, as well as to leverage better operational practices, will serve application owners well.

The integration of these security best practices with your DevOps processes is where the rubber meets the road. The automation of security building, testing, and operating means that, as a developer, you don’t have to be constantly paranoid about security. Security is simply built into the development process as well as the automated tools that facilitate it.

These days security is an easy thing to implement in public cloud-based applications, given the availability of modern tools and approaches. The bad news is that security must continuously change and evolve to respond to changing risks, and it's no longer just a problem for IT security and the infrastructure team. Developers are now in the fight.

What are you doing to protect your apps?