7 Common Security Pitfalls to Avoid When Migrating to The Cloud

No one migrates to the cloud to become less secure than before the migration. Read on to learn how to prevent such a security regression when migrating to the cloud.

So You Want to Move to the Cloud. It's Okay. You're Not Alone.

96 percent of decision makers in one survey have cloud initiatives underway. Enterprise IT teams will soon reach the tipping point, **spending over 50 percent** on cloud apps and services instead of on-premises deployments for the first time. **Even the NSA is joining the fun**. It's no wonder, since there are many benefits to moving workloads to the cloud.

Cost is one major benefit of cloud deployments. Xero, a New Zealand based cloud accounting firm, improved its gross margin by **81 percent** using AWS services. Another driver is access to strong machine learning technologies, which helped the NSA to decide to migrate. However, there are challenges to migrating to the cloud, and **security is a big one**.

Attackers are finding more scope and attack surfaces to go after. **At the same time,** security teams can't hire enough people to keep up with the extra demand. Security teams need to face facts—keeping up isn't going to happen by hiring more workers. Testing for security problems needs to happen in a scalable and effective manner.

Knowing what to expect when it comes to security will allow you to avoid pitfalls and slow migrations. Let's take a look at 7 common security mistakes organizations make when migrating to the cloud. We'll also see how to tackle each one and what strategy companies need to keep up with the rate of change.

96%

of decision makers have cloud initiatives underway

+50%

Enterprise IT teams will soon spend over 50 percent on cloud apps and services instead of on-premises deployments for the first time.

1 Exposing Sensitive Information

There's a reason the **OWASP Top 10** lists sensitive information disclosure as the number three security risk.

Sensitive data breaches have been a problem for some time. However, a new risk appears when migrating to the cloud.

Amazon Web Services (AWS) offers a service called **S3**. S3 provides a simple storage service (hence the name) to users. You can place data into S3 "buckets" for use in other services or for backup purposes. For instance, you can place your database backups into S3 to cheaply store them. Files needed for batch processing could also be placed in S3.

Unfortunately, there are many S3 buckets that are publicly exposed, meaning anyone with an Internet connection can access the data inside them. These exposed buckets have caused **major data breaches** when bad guys found them.

Booz Allen Hamilton, U.S. defense contractor, **leaked** battlefield imagery and administrator credentials to sensitive systems through an insecurely configured S3 bucket. Accenture **lost the "keys to the kingdom"**, exposing technical details of its cloud platform, 40,000 passwords in plain text, decryption keys, and admin login credentials stored in at least four S3 buckets set to be publicly accessible.

1 Exposing Sensitive Information

Time Warner Cable **lost the personally identifiable information (PII) of 4 million customers** when two publicly exposed S3 buckets were hit. Verizon **coughed up the PII of 6 million customers** and later **leaked proprietary technical information** about its systems from misconfigured S3 buckets.

You don't have to be the next S3 data breach. There are concrete steps you can take to make sure your S3 buckets are safe.

- Create a build pipeline. Developers shouldn't be touching S3 bucket configuration directly. **S3 has sensible defaults** which make your bucket private. Any changes should be made through a build pipeline and not directly by a user.

- Consider using Defense-in-Depth strategies. A recommended strategy from Amazon is to use Amazon CloudFront and bucket policies to front your S3 buckets, thus preventing them from being exposed using the S3 URL. **Check out the docs** for details.

- Use good IAM policies. AWS has a rich Identity and Access Management (IAM) service with **best practices** to help you. Use application roles and bucket policies to control bucket access from various other services, such as EC2 or Lambda. Your S3 buckets should be treated like a "backend" system in legacy data centers. Only your applications should access them, so set up your IAM policies and roles accordingly.

You can find other great tips **here**. In the end, enforcing sensible policies leads to secure buckets.

2 Leaking Credentials

Highly privileged credentials are necessary for any application to run properly.

That's the reality we have to face. Problems occur when companies don't take good care of these credentials. Credentials can be user names, passwords, or secret access keys for services such as AWS.

In fact, **Hackers are sending bots to scour GitHub**, a popular source code repository, for credentials. Infrastructure and application code, when credentials are hard-coded or stored in source control, can expose secret keys used for authentication to AWS services.

What can you do to prevent sensitive credentials from being uploaded to GitHub?

- Use Git to detect sensitive data. Git is a powerful source code administration tool and is what GitHub is based on. The technical details of Git are out of scope for this discussion. However, **this nifty library** will allow developers and security team members to scan Git repositories and code commits for secrets which shouldn't be exposed.

- Use your cloud provider's tools to reduce the need to store credentials in source code or files. Amazon has released two tools, **Secrets Manager** and **Macie**, which can help prevent and detect credential leaks. Microsoft's Azure now has a **key vault** which protects your secrets. Use these tools to allow applications the access they require without needing to store secrets in plain text.

Learn how to use these tools properly and credential leakage won't slow down your migration.

3 Lack of Clear Policies (and Enforcing Them)

One of the great things about cloud environments is their flexibility and scalability.

But these properties also introduce interesting challenges for security teams. Unfortunately, some companies don't have policies at all and don't enforce them if they do. This leads to a "wild west" of cloud infrastructure where every developer is an admin or all admins have access to every service.

So what can companies do to define and enforce policies?

- Training is important. **Having too many admins** can be just a bad as not having enough. But even the admins you do have need to understand what they're allowed to do and what they aren't.

- Create automated ways to enforce these policies. Tools like **evident.io** or **Dome9** can automate and alert on rogue cloud services. Open source tools like **Netflix's Conformity Monkey, Security Monkey, and Janitor Monkey** can help keep your environment clean.

- Use imaginative ways to use compute power. Imagine you have a policy that requires all documents to be scanned for viruses. It's pretty easy to scan documents coming into your network through email. But what about documents uploaded to S3 buckets? **You can use AWS Lambda** to scan uploads to S3 for viruses automatically. You can use the power of the cloud in creative ways to help enforce policies that are difficult to enforce otherwise.

Christoffer Fjellström, a developer at Swedish security firm Detectify says, "The main problem is that companies really don't have policies for it or they don't follow up and make sure those policies are followed." Don't be one of those companies.

4 Not Vetting Your Vendors

Using vendors in the IT world is inevitable.

You will always need some specialized skill that you don't have in house. However, how you choose and deal with vendors can be detrimental to your security if not done right.

Verizon's data breach which exposed 6 million customer records was caused when a Verizon partner, Nice Systems, placed log information into a publicly accessible S3 bucket. **Time Warner Cable's breach** also occurred due to the poor practices of a third-party vendor.

You'll notice a common thread in these breaches. The news didn't read, "Nice Systems had a data breach." It read, "Verizon had a data breach." Your data is your responsibility. It's also your responsibility to make sure vendors are kept to the same standards of security as your employees.

Here's how to handle your vendors:

- Before choosing a vendor, thoroughly vet their security practices. Make sure your security policies can be enforced. Ensure your vendors view security as important. Don't sign a contract or use their services before you understand how they'll keep your data safe.

- Make security part of your contract. Google asks all their suppliers if they have a way for external parties to submit vulnerabilities to them. Dropbox **recently held a live hacking event** where some of their vendor's assets were in scope. Make sure any security requirements you have are thoroughly spelled out in your contract. Hold vendors accountable for the security of your data.

5 Accounts Run Amok

If employees can create AWS accounts on behalf of the company, sensitive data can be exposed.

The **Verizon breach mentioned earlier** was caused by a rogue S3 account made by an employee. Clear policies and monitoring of accounts is essential to preventing data breaches like that one. Using the least privilege security principle to prevent employees from having unlimited access is also essential.

Thankfully, AWS has created **a set of best practices** for account management:

- Clearly define an AWS account-creation process. This can be centralized but doesn't have to be as long as you keep an inventory of accounts.
- Define a company-wide AWS usage policy. This eliminates confusion and helps align business needs with security.
- Create a security account structure for managing multiple accounts. You can more easily assess your AWS account security if you create a security relationship between accounts.
- Leverage AWS APIs and scripts. Using scripts and AWS APIs will allow you to set baseline security configuration across all accounts.

If you're feeling like this is overwhelming and you don't know where to start, don't worry. AWS has an **implementation guide** to help you get started with this framework. Whether you do it Amazon's way or find another, you need a framework for account management in AWS.

6 Network Misconfigurations Leave Your Network Exposed

A technical pitfall you might run into if you're new to cloud is **improperly configured networks**. Keep an eye on **Virtual Private Cloud (VPC)** and **network Access Control List (ACL)** settings.

VPCs allow you to build a custom subnet in the cloud. If the Internet is the world, your VPC is your house. It's a place where you can go and have privacy. No one should have access to your VPC unless you allow them in. If your VPC allows access to any IP address on the Internet, it's like leaving your front door wide open.

Network ACLs act like a firewall in the cloud and control how networks talk to each other. Think of an ACL like border patrol. It inspects the packets coming into a VPC and decides whether or not they are allowed in. Similar to VPCs, ACLs can be made to accept traffic from any and all IP addresses.

There is one best practice for VPCs and ACLs to remember. Don't allow all IP addresses access to your AWS network. Only allow IP addresses from your trusted endpoints, such as the network proxies your employees use. This way, your front door won't be open to the world.

7 Not Using Proper Encryption

Proper encryption is absolutely necessary when using cloud services.

Accenture's data breach was really bad because of 40,000 plain text passwords stored in S3. You don't want data to get out at all. However, if it does, encryption will give you extra protection (as long as your keys aren't found either).

Many AWS services have encryption options available. S3 has **default encryption options**, along with **many encryption options** for different requirements. Using **Key Management Services** allows you to securely encrypt your data in Amazon services. Amazon's **Elastic Block Store (EBS)** and **Relational Database Service (RDS)** all have options to encrypt your data at rest.

Encryption in transit can be done by using a VPN connection to AWS. All Amazon endpoints are served over HTTPS by default. **AWS Certificate Manager** is a way to manage TLS certificates in your cloud infrastructure.

Don't forget all of the encryption tools available to you with cloud services. Not using them can lead to disaster, so make sure you understand what is available and how to use it. Check out **this slideshare** for AWS encryption best practices.

Unleashing the Power of Hackers (The Good Kind)

Looking out for these pitfalls will help you to prepare for your cloud migration. However, there may be problems with your cloud infrastructure that are harder to find.

The speed of cloud migration is outpacing the speed of security team expansion.

It's a matter of scope and resources. The scope is large and resources are scarce. Consider crowdsourcing your security. Crowdsourcing is the way to shore up your security by incentivizing many people to look via a bug bounty program or having a way for researchers to contact you via a formal vulnerability disclosure program (VDP).

HackerOne is the most comprehensive hacker-powered security platform. Having more people testing for you than you could on your own helps you find security problems before the bad guys do. You can have cloud security "as-a-service" through the eyes of many ethical hackers.

AlienVault, an incident response and threat detection company, recently **saw the benefits of creating a VDP** with HackerOne Response. HackerOne's hacktivity feed displays various reports of cloud security problems, such as **private key disclosure** and **SSRF vulnerabilities**, being found by our worldwide network of hackers.

You don't have to be the next cloud casualty. **Get in touch** with us to see how crowdsourcing your cloud security can help you avoid the pitfalls that slow down your cloud migration.

Migrating to the cloud is a large and scary proposition. Don't go in blind. Understand the pitfalls you'll run into and plan for them. Scale up your security with crowdsourcing to help with the pitfalls you can't see. Then you'll reach your destination: a strong, secure cloud infrastructure taking your business to new heights.

hackerone

ABOUT US

HackerOne is the #1 **hacker-powered security platform**, helping organizations find and fix critical vulnerabilities before they can be exploited. More Fortune 500 and Forbes Global 1000 companies trust HackerOne than any other hacker-powered security alternative. The U.S. Department of Defense, General Motors, Google, Twitter, GitHub, Nintendo, Lufthansa, Panasonic Avionics, Qualcomm, Starbucks, Dropbox, Intel, the CERT Coordination Center and over 1,000 other organizations have partnered with HackerOne to resolve over 80,000 vulnerabilities and award over $35M in **bug bounties**. HackerOne is headquartered in San Francisco with offices in London, New York, and the Netherlands.

Contact us to get started.