Data Security in the SaaS Age

Version 1.2
Released: August 14, 2020

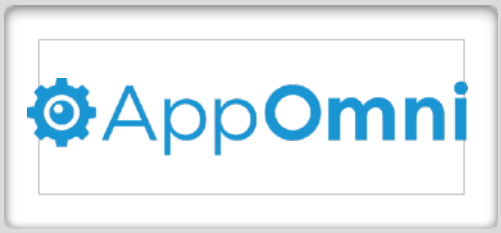
Securosis, L.L.C. 515 E. Carefree Highway Suite #766 Phoenix, AZ 85085 **T** 602-412-3051 info@securosis.com www.securosis.com

Author's Note

The content in this report was developed independently of any sponsors. It is based on material originally posted on the Securosis blog but has been enhanced, reviewed, and professionally edited.

Special thanks to Chris Pepper for editing and content support.

This report is licensed by AppOmni.



appomni.com

AppOmni is the leading provider of SaaS Security Posture Management (SSPM). AppOmni provides unprecedented data access visibility, management and security of SaaS solutions, enabling organizations to secure mission-critical and sensitive data. AppOmni's patent-pending technology deeply scans APIs, security controls, and configuration settings to evaluate the current state of SaaS deployments and compare against best practices and business intent. With AppOmni, organizations can establish rules for data access, data sharing and third-party applications that will be continuously and automatically validated.

Copyright

This report is licensed under Creative Commons Attribution-Noncommercial-No Derivative Works 3.0.

http://creativecommons.org/licenses/by-nc-nd/3.0/us/

Data Security in the SaaS Age

Table of Contents

Rethinking Data Security 4

Focus on What You Control 7

Thinking Small 11

Quick Wins 15

About the Analyst 18

About Securosis 19

Rethinking Data Security

Securosis has a long history of following and publishing on data security. Yet over the past few years, we've gotten distracted by this cloud thing, and we haven't gone back to refresh our research in light of major shifts in how data is used and stored, with SaaS driving the front office and IaaS/PaaS upending data centers. We described a lot of our thinking on the early stages of this transition in Tidal Forces 1 and Tidal Forces 3, and it seems (miraculously!) that a lot of what we expected three years ago has since come to pass.

But data security remains elusive. You can think of it as something of a holy grail. We've been espousing the idea of **data-centric security** for years, focusing on protecting the data, so you can worry less about securing devices, networks, and associated infrastructure. As with most big ideas, it seemed like a good idea at the time.

In practice, data-centric security has been underwhelming — it gradually became clear that having security policy and protection travel along with the data, as it spreads to every SaaS service you know about (and a bunch you don't), was just too much to count on.

In practice, data-centric security has been underwhelming — it gradually became clear that having security policy and protection travel along with the data, as it spreads to every SaaS service you know about (and a bunch you don't), was just too much to count on. How did Digital Rights Management work at scale? Right.

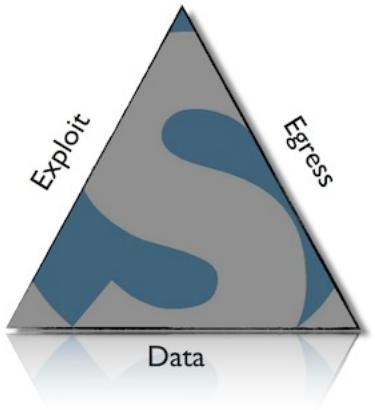
The industry scaled back expectations and started to rely on techniques like tactical encryption, mostly using built-in capabilities (FDE for structured data and embedded encryption for file systems). Providing a path of least resistance to both achieve compliance requirements, as well as 'feel' like the data was protected. But to be clear, this was mostly security theater — compromising the application still provided unfettered access to the data.

Other techniques, such as masking and tokenization, also provided mechanisms to shield sensitive data from interlopers. New tactics like test data generation tools offer more options to ensure that developers don't inadvertently expose production data. But even with all these techniques, most organizations still struggle to protect data. And it's not getting easier.

The Data Breach Triangle

Back in 2009, we introduced the Data Breach Triangle, which gave us a simple construct to enumerate a few different ways to stop data breaches. You just need to break one leg of the triangle.

- **Data:** The equivalent of fuel — information to steal or misuse.
- **Exploit:** A vulnerability or exploit path which allows an attacker unapproved access to the data.
- **Egress:** A path for the data to leave the organization. It could be digital, such as a network egress, or physical, such as portable storage or a stolen hard drive.



Most of the modern security industry focuses on stopping the exploit, either by impeding the ability to deliver the exploit, such as firewall/IPS, or preventing device compromise with endpoint protection. There have also been attempts to stop egress of sensitive data — via outbound filters, firewalls, web proxies, and DLP.

As described above, attempts to either protect or shield data at scale have proven very difficult. So what do we get? Consistent breaches. Normalized breaches. An organization losing tens of millions of identities no longer even registers as news.

SaaS Exacerbates the Issue

Protecting data continues to get more complicated. SaaS has won. As we described in Tidal Forces, SaaS is the new back office. If anything, remote working, driven by the inability to congregate in offices safely, will accelerate this trend.

Protecting data was hard enough when we knew where it was. I used to joke about how unsettling it was back in 1990 when my company outsourced the mainframe to Dallas, while we stayed in our building in Arlington, VA. At least all our data was in one place, even if the big iron was not in our control. Now, most organizations have dozens (or hundreds) of third parties controlling some critical corporate data. The problem isn't getting easier.

Rethinking Data Security

What we've been doing hasn't worked. Not at scale anyway. We need to take a step back and stop trying to solve yesterday's problem. Protecting data by encrypting it, masking it, tokenizing it, or wrapping a heavy usage policy around it wasn't the answer, for various reasons. The technology industry has rethought applications and the creation, usage, and storage of data. We security people need to rethink data security for this new SaaS reality. We must rethink both our expectations of what data security means and the potential solutions.

We security people need to rethink data security for this new SaaS reality. We must rethink both our expectations of what data security means and the potential solutions.

Focus on What You Control

To rethink data security, we need to revisit where we have been. That brings us back to our Data Security Lifecycle, last updated in 2011 (parts one, two, and three).

Lifecycle Challenges

At the highest level, the Data Security Lifecycle lifecycle lays out six phases from creation to destruction. We depict it as a linear progression, but data can bounce between phases without restriction and need not pass through all stages (in particular, not all data eventually gets destroyed).

1. **Create:** We should call this Create/Update because it applies to creating or changing a data/content element, not just a document or database. Creation is generating new digital content or altering/updating existing content.
2. **Store:** Storing is the act committing digital data to some sort of storage repository, and typically occurs nearly simultaneously with creation.
3. **Use:** Data is viewed, processed, or otherwise used in some sort of activity.
4. **Share:** Exchange of data between users, customers, or partners.

5. **Archive:** Data leaves active use and enters long-term storage.
6. **Destroy:** Permanent destruction of data using physical or digital means such as crypto-shredding.

With this lifecycle, you can evaluate data and make decisions about appropriate locations and access. You need to figure out where data can reside, which controls apply to each possible location, and how to protect data as it moves. Then work through a similar exercise to specify access rules, determining who can access the data and how. Your data security strategy depends on protecting all critical data, so you need to run through this exercise for each important data type.

To be clear, you are still accountable for protecting critical data — that hasn't changed. But you can share responsibility. Managing this shared responsibility becomes the most significant change in how we view data security.

Then dig another level down to figure out which data processing functions (such as Access, Write, or Store) apply during each phase of the lifecycle. Finally, you can determine which controls enable data usage for which functions. Sound complicated? It is — enough that it's impractical to use this model at scale. That's why we need to rethink data security.

Self-flagellation aside, we can take advantage of the many innovations we've seen since 2011 in the areas of application consumption and data provenance. We are building fewer applications and embracing SaaS. For the applications you still build, you can leverage cloud storage and other platform services. So data security is not just your problem any more.

To be clear, you are still accountable for protecting critical data — that hasn't changed. But you can share responsibility. You set policies within the framework your provider supports. Managing this shared responsibility becomes the most significant change in how we view data security. And we need this firmly in mind when we think about security controls.

Adapting to What You Control

Returning to the Data Breach Triangle, you can stop a breach by 'eliminating' the data to steal, stopping the exploit, or preventing egress/exfiltration. In SaaS you cannot control the exploit, so forget that. You also probably don't see the traffic going directly to a SaaS provider unless you inefficiently force all traffic through an inspection point, so focusing on egress/exfiltration is probably a non-starter.

That leaves you to control the data itself. *Specifically to prevent access to sensitive data, and restrict usage to authorized parties.* If you prevent unauthorized parties from accessing data, it's tough for them to steal it. If we can ensure that only authorized parties can perform certain functions with data, it's hard for them to misuse it. And yes — we understand this is much easier said than done.

Restated, data security in a SaaS world requires much more focus on access and application entitlements. You handle it by managing entitlements at scale. An entitlement ensures the right identity (user, process, or service) can perform the required function at an approved time. Screw this up and you don't have many chances left to protect your data because you can't see the network or control application code.

If we dig back into the traditional Data Security Lifecycle, the SaaS provider handles a lot of these functions — including creation, storage, archiving, and destruction. You can indeed extract data from a SaaS provider for backup or migration, but we're not going there now — we'll focus on the Use and Share phases.

This isn't much of a lifecycle anymore, is it? Alas, we should probably relegate the full lifecycle to the dustbin of "it seemed like a good idea at the time." The modern critical requirements for data security involve setting up data access policies, determining the right level of authorization for each SaaS application, and continuously monitoring and enforcing policies.

The Role of Identity in Data Protection

You may have heard the adage, "Identity is the new perimeter." Platitudes aside, it's basically true, and SaaS data security offers a good demonstration. Every data access policy associates with an identity. Authorization policies within SaaS apps depend on identity as well.

It's not just employees who can be considered 'users' by your identity infrastructure. Third-party apps and business partners may have access to the information in your SaaS apps, so an expanded view of identity is important for understanding the true extent of your attack surface.

SaaS data security strategy hinges on identity management. This dependency puts a premium on federation, because managing hundreds of user lists and handling the provisioning/deprovisioning process individually for each application doesn't scale.

SaaS data security strategy hinges on identity management, like most other things in the cloud. This dependency puts a premium on federation, because managing hundreds of user lists and handling the provisioning/deprovisioning process individually for each application doesn't scale. A much more workable plan is to implement an identity broker to interface with your authoritative source and federate identities to each SaaS application. This becomes part of your critical path to provide data security. But that's a bit afield from this research, so we need to leave it at that.

Data Guardrails and Behavioral Analytics

If managing data security for SaaS applications boils down to being able to set and enforce policies for each SaaS app, your efforts require a clear understanding of each specific application, so you can set appropriate access and authorization policies. Yes, SaaS vendors should make that easy, but in reality… not so much.

But setting policies is only the first step. Environments change regularly, as new modules become operational and users onboard and off-board. Policies frequently change, which creates an opportunity for mistakes and attacks.

That all brings us to Data Guardrails and Behavioral Analytics. We first introduced this concept back in late 2018. For the backstory, you can take a deep dive into the concept in two parts (Part 1 and Part 2). But here's a high-level recap of the concepts:

- **Data Guardrails:** We see Guardrails as a means of enforcing best practices without slowing down or impacting typical operations. Typically used within the context of cloud security (like, er, DisruptOps), a data guardrail enables data to be used in approved ways while blocking unauthorized use. To bust out an old network security term, you can think of guardrails like *default-deny* for data. Define acceptable practices, and don't allow anything else.
- **Data Behavioral Analytics:** Many of you have heard of UBA (User Behavioral Analytics), which profiles all user activity and then monitors for anomalous activities, which could indicate an insider risk. But what if you turned UBA inside out and focused on data? Using similar analytics you could profile all data usage in your environment and then look for abnormal patterns that warrant investigation.

It's not an either/or choice. You start by defining data guardrails, which provide the ability to establish allowed activities in each SaaS application for each user, group, and role. Then you monitor data usage for potentially malicious activities.

The key difference is that data guardrails leverage this knowledge with deterministic models and processes to define who can do what and stop everything else. Data behavioral analytics extend the analysis to include current and historical activity, using machine learning algorithms to identify unusual patterns that bypass other data security controls.

It's not an either/or choice, as we figure out how to enforce data security policies in all these SaaS environments. You start by defining data guardrails, which provide the ability to establish allowed activities in each SaaS application for each user, group, and role. Then you monitor data usage for potentially malicious activities.

We favor this approach to dealing with SaaS (and most cloud-based data usage) because it enables you to focus on what you control. You still think about data through its lifecycle — but your responsibilities within the lifecycle have changed to accommodate shared responsibilities.

Thinking Small

If you agree that the *use* and *sharing* phases of the (partially defunct) Data Security Lifecycle remain relevant, how do these phases relate to the SaaS world? The old approach hinges on a detailed understanding of each application to define appropriate policies. You need to understand what is allowed and by whom. But these are not — and cannot be — generic rules. Each SaaS application is different, so you (or a vendor or service provider) needs to dig into the application to understand what it does, who can use it, and how.

Now the fun part. The typical enterprise has hundreds or thousands of SaaS services. And each SaaS application can have hundreds of configuration options, so you are dealing with a huge number of settings to assess and configure securely. So what's the best approach to secure those applications? Any answer requires gratuitous use of many platitudes, including both "How do you eat an elephant? One bite at a time." and that other old favorite, "You can't boil the ocean."

The typical enterprise has hundreds or thousands of SaaS services. And each SaaS application can have hundreds of configuration options. So what's the best approach to secure those applications? You need to think small, by setting policies to protect one application or service at a time.

Whichever pithy analogy you use to describe providing data security for SaaS, you need to *think small,* by setting policies to protect one application or service at a time. We're looking for baby steps, not big bangs. The big bang killed initiatives like DLP. (You remember DLP, right?) Not that folks don't do DLP successfully today — they do — but if you try to classify all the data and build rules for every possible data loss up front... you'll get overwhelmed, and then it's hard to complete the project.

We've been preaching this small and measured approach for massive, challenging projects like SIEM for years. You don't set up all the SIEM rules and use cases at once — at least not if you want the project to succeed. The noise will bury you, and you'll stop using the tool. People with successful SIEM implementations under their belts started small with a few use cases, then added more once they figured out how to make a few sets work.

The Pareto principle applies here, big-time. You can eliminate the bulk of your risk by protecting 20% of your SaaS apps. But if you use 1,000 SaaS apps, that means you still need to analyze and set policies for 200 apps, a legitimately daunting task. *We're talking about a journey here — one that takes a while.* So prioritizing your SaaS applications is essential.

We'll also discuss opportunities to accelerate the process later on — you can jump the proverbial line with smart technology use.

The Process

The first SaaS app you run through the process should be an essential app with pretty sensitive data. We can bet it will be either your office suite (Office 365 or G Suite), your CRM tool (likely Salesforce), your file storage service (typically Dropbox or Box), or your ERP or HR package (SAP, Workday, or Oracle).

These applications represent your most sensitive data, so next you'll want to maximize risk mitigation. Start with the app with the largest user base. We'll illustrate the process with CRM. We get going by answering a few standard questions:

1. **What's the data?** Your CRM has all your marketing and sales data, including a lot of sensitive customer/prospect data. It may also have your customer support cases, which are themselves sensitive.

2. **Who needs to see the data?** Define who needs to see the data and use the groups or roles within your identity store — no need to reinvent the wheel. We discussed federation earlier in this paper, and this is why. Don't forget to consider external constituencies — auditors, contractors, and even customers.

3. **What do they need to do with the data?** For each role or group, figure out whether they need to read, write, or otherwise manage data. You can get more specific and define different rights for different data types as required. For example, finance people may have read access to the sales pipeline, while sales operations folks have full access.

Do you see what we did there? That's right; we just built a simple entitlement matrix. That wasn't so scary, was it?

Once you have the entitlement matrix documented, write policies. Finally, you load your policies into the application. Then wash, rinse, and repeat for the other SaaS apps you need to protect.

Each SaaS app will have a different process to implement these policies, so there isn't a whole lot of leverage to be gained in this effort. But you probably aren't starting from scratch either. A lot of this work happens when initially deploying applications. Hopefully, it's a matter of revisiting original entitlements for effectiveness and consistency. But not always. To the accelerate PoC, vendors use default entitlements, and the operations team doesn't always revisit them when the application goes from testing into production deployment.

Continuous Monitoring

Once the entitlements are defined or revisited, and you've implemented acceptable policies in the application, you reach the operational stage. Many organizations fail here. They get excited to lock things down during initial deployment but seem to forget that moves, adds, and changes happen every day. New capabilities get rolled out weekly. So when they periodically check policies every quarter or year, they are surprised by how much changed, and the resulting security issues.

> Security doesn't need to have operational responsibility for SaaS applications, but they need to assess the risk of access when building the entitlement matrix, and to monitor the application to ensure changes don't violate policy or add attack surface.

Continuous monitoring is critical for maintaining the integrity of data in SaaS apps. You need to watch for changes, with a mechanism to ensure they are authorized and legitimate. It sounds like a change control process, right? What happens if the security team (or even the IT team!) doesn't operate these apps?

We've seen this movie before. It's like dealing with an application built in the cloud by a business unit. The BU may have operational responsibility for the application, but the security team should assume responsibility for enforcing governance policies. Security needs access to the SaaS app to monitor changes and ensure policy adherence.

And that's the point. Security doesn't need to have operational responsibility for SaaS applications. Yet they need to assess the risk of access when building the entitlement matrix and to monitor the application to ensure changes don't violate policy or add attack surface.

Build vs. Buy

A described above, thinking small, building access rules, and monitoring each SaaS app is resource-intensive. So is there a scalable way to secure SaaS apps? SaaS providers may provide deployment tools to define access rules quickly and efficiently. It's in their interest to get your organization up and running as quickly as possible.

You could also send alerts about entitlement changes to your SIEM or security analytics tool to monitor for potentially malicious activity. Once you've identified a problem, it can be sent to your work management system (ServiceNow, Jira, etc.) to apply the requested changes. Or you can DIY (Do It Yourself).

Alternatively, you can accelerate your security process by looking at SaaS management tools that have already done the work to understand each SaaS app. These tools provide a quick assessment of current entitlements, and then both manage and monitor access rules going forward. These tools can also provision and deprovision application users, and stay on top of frequent changes within the SaaS app, so you don't have to. The build versus buy decision hinges not only on the adoption of more SaaS apps in the future but also on how frequently you need to reassess the security of your SaaS apps in use.

For example, when Salesforce rolls out a new service (or more likely acquires it and bundles it into your package), how should that impact your entitlements? DIY requires that you evaluate the new service yourself, while a vendor should have those changes enumerated in their tool sooner than you can do it. We can't underestimate the challenge of keeping the security posture of many SaaS applications current, given the amount of change in both your environment and application capabilities. Something likely has to give, and that's when an issue (read 'breach') can happen. The process was fine, but there weren't enough resources to execute the process consistently over time.

Are these tools a panacea? Of course not — no tool offers everything you need for every application you need to protect. It comes down to resource allocation and risk management. If you have the resources to do it yourself, have at it. And if you aren't worried about losing data from a particular SaaS app, don't worry about it. It's not like you don't have other worries.

But if your security team remains resource-constrained and a SaaS app stores sensitive data, a SaaS management tool warrants investigation. Here are a few things to consider:

- **Application Support:** Does the tool support your top 10-15 applications? Over time the library of supported applications should grow, but initially you need to protect your highest-profile applications. Understand how well the vendor SDK can support new and unsupported apps and how well the vendor will support the SDK.
- **Policy Granularity:** Does the tool offer the ability to set sufficiently detailed policies *for the application*? Can you restrict access to certain parts of the application? Are geographic limitations and device restrictions supported? Many policy knobs are available — make sure you have what you'll need to protect your applications.
- **Supported Use Cases:** Does the tool offer the capabilities you need? Think about the key use cases of access control, data security, and provisioning/deprovisioning; and understand how the application's API capabilities will constrain your use cases.
- **Integration with Existing Operations Tools:** Make sure the tool will integrate with your SIEM/monitoring platform, work management tool (to track service tickets), and automation tool.

The build versus buy decision is challenging in a young market. The tool you like may only support a handful of the SaaS applications you use, so its immediate value may be limited. But don't just consider today's capabilities — also think about which new applications will be supported over a reasonable timeframe (perhaps 18 months), and how confident you are in the vendor hitting those milestones. Or build it yourself.

Quick Wins

As we wrap up this paper on Data Security in the SaaS Age, let's work through a scenario to show how the concepts apply in a specific scenario. We'll revisit the "small but rapidly growing pharmaceutical company" we used as an example in our Data Guardrails and Behavioral Analytics paper. The CISO has seen the adoption of SaaS accelerate over the past two years. Given the increasing demand to *work from anywhere* at all organizations, the CTO and CEO decided to minimize on-premise technology assets.

A few years ago, they shifted their approach to use data guardrails and behavioral analytics to protect the sensitive research and clinical trial data the business generates. But they still need a structured program and appropriate tools to protect their SaaS applications. With hundreds of SaaS applications in use and many more coming, it can be a bit overwhelming to the team, who needs to both understand their extended attack surface and figure out how to protect it at scale. Sound familiar? With guidance from their friends at Securosis, they start by looking at a combination of risk (primarily to high-profile data) and broad usage within the business, as they figure out which SaaS application to focus on protecting first.

With hundreds of SaaS applications in use and many more coming, it can be a bit overwhelming to the team, who needs to both understand their extended attack surface and figure out how to protect it at scale. Sound familiar?

The senior team decides to start with CRM. Why? After file storage/office automation, CRM tends to be the most widespread application, representing the most sensitive information stored in a SaaS application: customer data. They also have many business partners and vendors accessing the data and the application, because they have multiple (larger) organizations bringing their drugs to market; they want to make sure all those constituencies have the proper entitlements within their CRM. Oh yeah, and their auditors were in a few months back and *suggested* that assessing their SaaS applications needs to be a priority, given the sensitive data they hold.

As we described earlier, the process of setting access policies involves determining **who** should use the data and **how**. For simplicity's sake, we'll generalize and answer these questions at a high level, but you should dig much deeper to drive policy.

1. **What's the data?** The CRM has detailed data on all the doctors visited by the sales force. It also contains an extract of prescribing data to provide results to field reps. The CRM has data from across the globe, even though business partners distribute the products in non-US geographies, to provide an overview of sales and activity trends for each product.

2. **Who needs to see the data?** Everyone in the company's US field organization needs access to the data, as well as the marketing and branding teams focused on targeting more effective advertising. It gets a little squishy with the business partners who also need access to the data. But multiple business partners are serving different geographies, so tagging is critical to ensure each customer is associated with the proper distribution partner. Federated identity allows business partner personnel to access the CRM system with limited privileges.

3. **What do they need to do with the data?** The field team needs to be able to create and modify customer records. The marketing team just needs read-only access. Business partners update the information in the CRM but cannot create new accounts. That happens through a provider registration process to ensure multiple partners don't call on the same doctor or medical office. Finally, doctors want to see their prescribing history, so they need access as well.

If the team was starting from scratch, they would enumerate and build out the policies from whole cloth, and then deploy the CRM with the right rules the first time. But that train has already left the station. Thousands of people (internal personnel, business partners, and customers) already access the CRM system, so the first order of business is a quick assessment of the SaaS application's current configuration.

Quick Assessment

They didn't have the internal bandwidth to perform the assessment manually during the timeframe required by the auditors, so they engaged a consulting firm which leveraged a SaaS management tool for the assessment. What they found was problematic. The initial entitlements allowed medical practices to access their prescribing history. But with overly broad privileges, any authorized user within a medical practice could see all their practice's full customer records — which included not just the history of all interactions, but also notes from the sales rep. And let's just say some of the reps were brutally honest about what they thought of some doctors.

Given the potential to upset important customers, it's time to hit the fire alarm and kick in the damage control process. The internal IT team managing the CRM took a quick look and realized the access rule change happened within the last 48 hours. And only a handful of customers accessed their records since then. They reverted to the more restrictive policy, removed access to the affected records, and asked some (fairly irate) VPs to call customers to smooth over some ruffled feathers. The cardiologist who probably should have taken their own advice about health and fitness appreciated this gesture (and mentioned enjoying the humble pie).

There were a few other over-privileged settings, but they mostly affected internal resources. For example, the clinical team had access to detailed feedback on a recent trial, even though company

policy is only to share anonymized information with clinicians. Though not a compliance issue, this did violate internal policy. They also found some problems with business partner access rules — business partners in Asia could see all the Asian doctors. They couldn't make changes (such as reassigning doctors to other partners), but partners should only see the data for doctors they registered.

The other policies still reflect current business practices. After addressing these issues the team felt good about their security posture.

Continuous Monitoring

The security team cannot afford to get too comfortable given the constant flow of new customers, new partners, and new attacks. The last aspect of the SaaS data security program is monitoring. It's essential to monitor the SaaS application and gain an understanding of how each change impacts security posture. It's also critical to have clear notification and automated remediation processes for specific issues. Before claiming victory over the security of the CRM application and moving on to the next SaaS application, the team also needs to make sure there is a schedule of periodic assessment to revisit entitlements.

For never-acceptable issues, the security team is empowered to make necessary changes to protect data. Period. In some situations you take action immediately, and then deal with any fallout, because critical data is at risk.

In our pharmaceutical company, the security and ops teams got together to define and document ongoing assessment and remediation. A set of issues were identified as *never acceptable*, including things such as privilege escalation to make customer changes without authorization, downloading clinical data, and providing global access to customer data, among others. For never-acceptable issues, the security team is empowered to make necessary changes to protect data. Period. In some situations, you take action immediately and *then* deal with any fallout, because critical data is at risk.

For issues with less risk, the ops team set an expectation with the security team for what information they need to assess the urgency and make appropriate fixes. They also set SLAs to respond to issues received based on data criticality.

After we first published Tidal Forces back in early 2017, it became clear that SaaS would supplant pretty much all back-office applications. That has already largely come to fruition and is accelerating now that on-premise resources are mostly out of reach. Many of our tried and true approaches to data security must evolve. We hope we have laid out a path to help with your journey.

If you have any questions on this topic, or want to discuss your situation specifically, feel free to send us a note at info@securosis.com.

About the Analyst

Mike Rothman, Analyst and President

Mike's bold perspectives and irreverent style are invaluable as companies determine effective strategies to grapple with the dynamic security threatscape. Mike specializes in the sexy aspects of security — such as protecting networks and endpoints, security management, and compliance. After 20 years in and around security, he's one of the guys who "knows where the bodies are buried" in the space.

Starting his career as a programmer and networking consultant, Mike was an analyst at META Group prior to founding SHYM Technology, and then held executive roles at CipherTrust and TruSecure. Mike then started Security Incite in 2006 to provide a voice of reason in an over-hyped yet underwhelming security industry. After taking a short detour as Senior VP, Strategy at eIQnetworks, Mike joined Securosis with a rejuvenated cynicism about the state of security.

Mike published The Pragmatic CSO <http://www.pragmaticcso.com/> in 2007 to introduce technically oriented security professionals to the nuances of what is required to be a senior security professional. He also possesses a very expensive engineering degree in Operations Research and Industrial Engineering from Cornell University. His folks are overjoyed that he uses literally zero percent of his education on a daily basis.

About Securosis

Securosis, LLC is an independent research and analysis firm dedicated to thought leadership, objectivity, and transparency. Our analysts have all held executive level positions and are dedicated to providing high-value, pragmatic advisory services. Our services include:

- **Primary research publishing**: We publish the vast majority of our research for free through our blog, and package the research as papers that can be licensed for distribution on an annual basis. All published materials and presentations meet our strict objectivity requirements, and follow our Totally Transparent Research policy.
- **Cloud Security Project Accelerators**: Securosis Project Accelerators (SPA) are packaged consulting offerings to bring our applied research and battle-tested field experiences to your cloud deployments. These in-depth programs combine assessment, tailored workshops, and ongoing support to ensure you can secure your cloud projects better and faster. They are designed to cut months or years off your projects while integrating leading-edge cloud security practices into your existing operations.
- **Cloud Security Training:** We are the team that built the Cloud Security Alliance CCSK training class and our own Advanced Cloud Security and Applied SecDevOps program. Attend one of our public classes or bring us in for a private, customized experience.
- **Advisory services for vendors**: We offer a number of advisory services to help our vendor clients bring the right product/service to market in the right way to hit on critical market requirements. Securosis is known for telling our clients what they NEED to hear, not what they want to hear. Clients typically start with a strategy day engagement, and then can engage with us on a retainer basis for ongoing support. Services available as part of our advisory services include market and product analysis and strategy, technology roadmap guidance, competitive strategies, etc. Though keep in mind, we maintain our strict objectivity and confidentiality requirements on all engagements.
- **Custom Research, Speaking and Advisory**: Need a custom research report on a new technology or security issue? A highly-rated speaker for an internal or public security event? An outside expert for a merger or acquisition due diligence? An expert to evaluate your security strategy, identify gaps, and build a roadmap forward? These defined projects bridge the gap when you need more than a strategy day but less than a long-term consulting engagement.

Our clients range from stealth startups to some of the best known technology vendors and end users. Clients include large financial institutions, institutional investors, mid-sized enterprises, and major security vendors. For more information about Securosis, visit our website: <http://securosis.com/>.