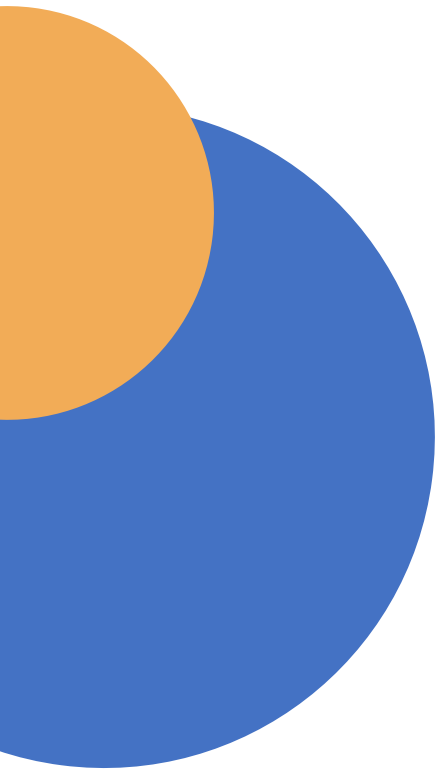




TECHNICAL WHITEPAPER

# Traditional Cybersecurity does not work in the cloud. Enter Shift-Left-Security



# TABLE OF CONTENTS

---

<b>Introduction: Exact Payments in the Dot Com Days</b>	<b>3</b>
<b>Managing the Security of the Applications</b>	<b>4</b>
<b>Creating Our First Proof of Concept</b>	<b>6</b>
<b>The Move to Shift-Left</b>	<b>6</b>
<b>Summarization</b>	<b>9</b>





## EXECUTIVE SUMMARY

---

Exact Payments has been operating as a payment gateway since the late 1990s, but much has changed in the business landscape since then. Recent decades have seen significant changes in terms of compliance and cybersecurity, and to help our customers not only keep pace but actually excel in the midst of these new developments, we modernized our applications with an emphasis on shift-left-security.


This whitepaper will discuss the following key topics around this transition, including:

- How PCI-DSS changed credit card transactions and datacenters, as well as the roles that were born from these changes
- Factors that drove Exact Payments to adopt “the cloud”
- How Exact Payments built on its first proof of concept and took code from development to production

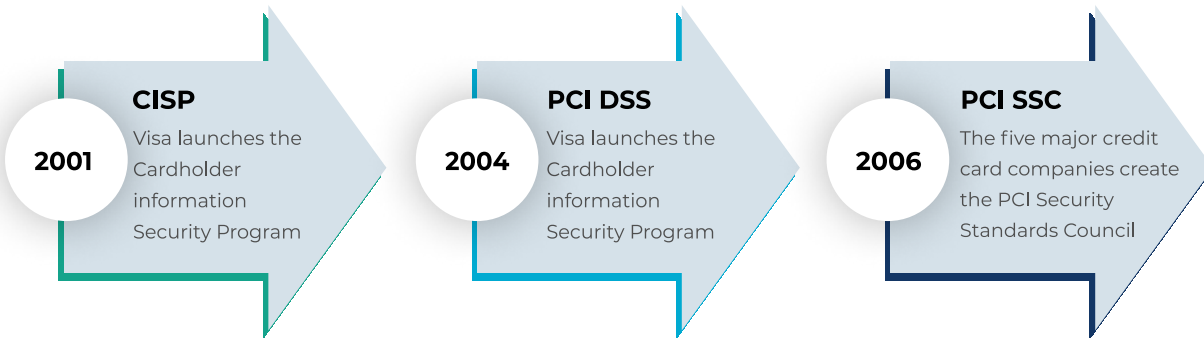
# INTRODUCTION: EXACT PAYMENTS IN THE DOT COM DAYS

“Moving from a physical datacentre to a cloud provider such as AWS, Azure or GCP cannot be done with a traditional mindset. “

Exact Payments has been around since the dot com days. We started out as a payment gateway company that provides an interface for merchants to process credit cards for their e-commerce sites. In our history, we have built and torn down many datacentres all over North America. When the company started out in 1999, we did not even think about security, as we focused on the business primarily. As business ramped up and cyber threats became a reality the landscape for payments changed for the better good of humanity.

 As business ramped up and cyber threats became a reality, the landscape for payments changed for the better good of humanity.

## *The Mid-2000s: New Compliance Standards & The Need for Change*



On September 7, 2006, PCI-DSS was born: the first compliance standard that focused on the secure handling of credit card data. The standard is broken down into 12 requirements that focus on network, operating systems, software, and development security.

When PCI-DSS was first introduced, the tooling to check for vulnerabilities and security flaws was nothing like the gamut of tools we have today. As we progressed through each iteration of the standard, we started to introduce more and more tools into our arsenal. PCI also brought in a new breed of System Administrators called Security Officers. These newly appointed guardians of everything security were tightly integrated into both the operations and development teams.



The security officer was born as a necessity to keep both the datacentres and the software the company built safe from “the bad guys.” These specialists must wear all hats in the company, as they must attend countless meetings and perform vulnerability, penetration, static code and dynamic code tests.

In addition to security officers, data centers have their own magical tool set. The introduction of hypervisors allowed us to scale down the number of physical resources required to operate our business. Being able to have high availability and long-lived virtual machines that could be backed up and cloned was a boon for our industry vulnerability scanners such as Tenable Nessus Professional became the a standard tools of choice for most companies. Having a history of a virtual machines patch lifecycle was critical.

## MANAGING THE SECURITY OF THE APPLICATIONS

Software developers are critical in making code secure. Without proper training, software developers can inject vulnerabilities into the code without knowing it. In the early days, doing code review with the developers was necessary to identify any potential flaws in the code. This process only covered the code that was added or modified. During the code reviews most of the code was not being looked at all. Tools for software development have come a long way over the last decade, providing excellent tooling for finding OWASP top 10 and SANS 25 vulnerabilities.

### ***Countless Audits and Certifications***

Beyond addressing coding issues, the security team is also responsible for keeping everything in compliance with the latest standards to ensure smooth business operations. This on its own is a full-time job for a team. Imagine every aspect of your business being looked at from a microscopic level year-round with a detail-oriented auditor waiting to catch a flaw, mistake, or misconfiguration. Creating a schedule or recurring tasks is important to avoid missing any key points that the auditors drill down on. The security team in our company has 328 recurring tasks that are completed monthly. If even one of these tasks is missing, it will be flagged and reported — which is precisely the point of the audits. Recurring tasks can be broken down into the following categories:

- Access controls
- Accountability
- Auditing (logs)
- Backup & Recovery
- Business Continuity
- Disaster Response
- Incident Response
- Monitoring
- Onboarding & Offboarding Employees
- Patching (Operating systems, Custom software)
- Policy Management
- Risk Assessments
- System Hardening
- Threat Management
- Training & Certifications
- Vendor Management
- Vulnerability Management

The scope of our audits expands every year. With new requirements being added continuously, the security team has to re-train and execute the new requirements to the entire company.

### ***The Key to Making Everything Easier***

When it was time to adopt the cloud in 2021, cloud vendors such as AWS, Azure, GCP and third party vendors had the ability to accelerate new developments or migrations away from a traditional datacentre. We did our homework to see what this would look like from both a security perspective and an administrative perspective. Our previous attempts at running anything in “the cloud” before consisted of a ton of click ops and a lot of misconfigurations. This seemed like a non-starter as it would introduce considerable risk.

We did some research and decided to go with AWS as a starting point, as we have had experience with it in the past. AWS was promoting cloud formation as a concept we never heard of before, “Infrastructure as code.”. This was a cool idea, and after digging in we realized that we did not want to be vendor locked so we looked at other forms of IaC. We continued to search and shop around, and we found a company that was selling pre-made templates that would get us up and running very fast. We found that in Terragrunt.

# CREATING OUR FIRST PROOF OF CONCEPT

Using Terragrunt to get our first proof of concept up and running, we were able to bring up all the necessary resources using IaC. Now that we had our infrastructure up, we were in a scramble to find a way to deploy our application. In the past, we used various configuration utilities to deploy our code to virtual machines; however, the products that we were using were getting old and we needed more features. We started by picking Ansible to launch our legacy apps into our newly spun up EC2 instances. This worked well, but we realized that spinning up an exact replica of our on-premises datacenters in AWS was not really moving forward and would cost us a fortune to operate.

## *Modernizing Our Applications*

Using our long history of payment processing, we decided to re-write everything we had from the ground up using microservices. There was a lot of logic behind our decision. First, we could cut costs by reducing our EC2 instance dependency from 40 to just four. Secondly, by rewriting our code, we were able to eliminate two decades of old unused code.



The most noteworthy reason for this decision is that it allowed us to introduce shift-left from the very beginning.

# THE MOVE TO SHIFT-LEFT

“Shift-left” simply means to move to the beginning. In the context of security, it means moving all security processes and procedures to the beginning of design, planning, development, execution and run-time.

Starting from scratch on all aspects (including infrastructure) was a huge deal. We were able to run security checks on all code that our developers produced as well as all infrastructure code our DevOps produced. What this gave us is clean, vulnerability-free starting code that we could build from and easily maintain. Before we began, we started off with some in-depth developer training on development security. This gave all our developers and DevOps the right knowledge to make the right decisions from day one.

We made our developer training very comprehensive with topics such as:

- AAA - Authentication, Authorization and Accounting
- Encryption
- Logging Standards
- OWASP
- PCI-DSS compliance standards (3.2.1)
- Privacy compliance standards (CCPA, GDPR, PIPEDA)
- SANS 25
- Secure coding best practices
- Software Development Lifecycle (Best Practices)

From there, we bought various security tools that we could integrate into the developer's IDE as well as bring it in line with the CI/CD pipelines we wanted to build in the very near future.

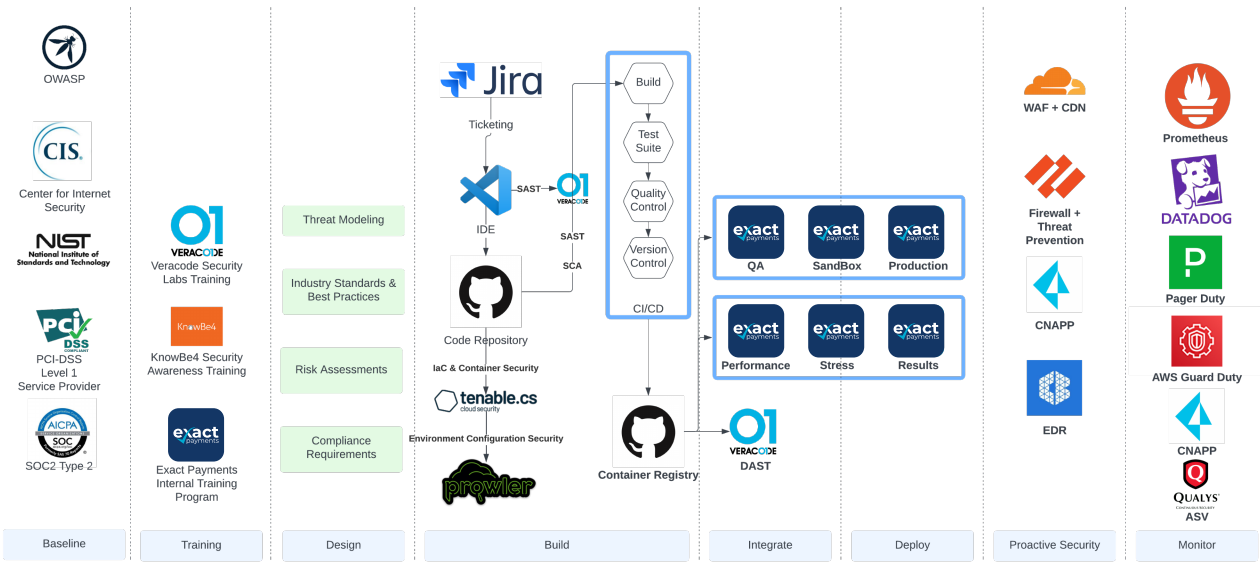
The beauty of all of the training upfront is now when our developers commit code to Git Hub and issue a "pull request" the developer that reviews the code has all of the secure coding and best practices training to identify any flaw or any code that does not follow our standards. This has greatly lifted the pressure on the security team and in turn has sped up the development process.

### ***Automating Deployments From Development to Production***

With a lot of CI products on the market such as CircleCI, Jenkins, GitHub Actions and more, we decided to pick a product that was inline and easy to use. We started with the basics, packaging our code to make images that we could deploy. Then we progressed to deploying the images to the EKS cluster. Once this was working in the test environment, we then had the foundations to build a very robust pipeline.



## Overview of Shift Left Security



### Glossary for Acronyms

- ASV - Approved Scanning Vendor
- CDN - Content Delivery Network
- CNAPP - Cloud-Native Application Protection Platform
- EDR - Endpoint Detect & Response
- IaC - Infrastructure As Code
- IDE - Integrated Development Environment
- DAST - Dynamic Application Security Testing
- SAST - Static Application Security Testing
- SCA - Software Composition Analysis
- WAF - Web Application Firewall

We started off with the security first. Before we packaged the code, we would run a SAST and SCA scan on our code. With basic business logic, we stated that if we found any vulnerability with a CVE > 4, we would fail the pipeline. Our developers jumped to the task to clear all issues >4. But within two weeks the pipeline would not halt, and we figured we could do better.

We reduced the scope to anything >0 and we scrubbed our code extremely clean.

### What about the containers?

One of the biggest misconceptions is that we can pull a docker image from a trusted source and put our code in it and everything will be nice and clean. This is not the case. Most of the images you can find have a lot of extra components in them, which have their own requirements for patching. If you scour the internet, you can find minimalist images like Alpine that have just enough to get the code working. However, this does not go far enough. Distrosless was created to remove the packager from the whole process to make it even smaller and with fewer vulnerabilities to manage.

### Securing the Infrastructure, Shift-Left Style

Using a few different types of tools, we were able to see any insecure configuration at every level of the infrastructure.

We created a DevOps pipeline that scanned both our IaC and the environment. With the ability to track any drift between the two, we had a much tighter control. The days of click ops have now been eliminated. After going through a few iterations of the IaC we quickly realized that we do not require users in the environments. Being able to strip out active directory was a huge win, now with the entire environment on auto pilot and auto remediation, we no longer required the biggest risk to any infrastructure: the human users. The final step, of course, was to manage the AWS console users, which was handled by using a federated directory. We sourced out a managed federated user directory provider to connect all of our SaaS accounts which greatly simplified the user experience and increased our security by using their advanced authentication technology. With this approach, we were able to use AWS SSO in combination with our IdP to allow console and CLI access to the AWS environment.

## SUMMARIZATION

After all of that hard work, we now have successfully built a model where we can start with security first. Every action from a day-to-day perspective will start with security, but won't block features and capabilities that customers need to have. If we are to break these steps down into pillars, you will see them as:

- Training
- Automated Software Vulnerability Scanning
- Automated Infrastructure Vulnerability Scanning
- Ongoing Remediation
- Structured Development LifeCycles
- Least Privilege Permissions
- Multi Factor Authentication Everywhere

Ultimately, it's critical for your team to think about security and all risks that they could introduce. No security plan is perfect and will always need to be revised and tested. If you take one thing away from this article, it should be to start with security first, as it is very hard to prioritize it later on.



## Give us a call. We're available:

Monday - Friday

7:30 AM PDT - 5:30 PM PDT

1-877-303-9228

[support@exactpay.com](mailto:support@exactpay.com)

Learn More:

[exactpay.com](https://exactpay.com)

Get Started:

[exactpay.com/get-started/](https://exactpay.com/get-started/)

---

## More Resources

Enjoy this guide?

Visit <https://exactpay.com/resources/> for more best practice guides, blogs, and ebooks on payment solutions

---



Exact Payments is a leading provider of high performance payment solutions for bank partners, software platforms and omnichannel merchants. The Company's cloud technology is built for scale and offers modern REST APIs, PCI-compliant hosted payment pages and an intuitive virtual terminal/management portal. Exact is integrated with leading processors in the U.S. and Canada including Fiserv/First Data, Elavon, Global Payments/TSYS, Chase Canada and Moneris. Learn more at [www.exactpay.com](https://www.exactpay.com).

**Phone** +1 877-303-9228 | [success@exactpay.com](mailto:success@exactpay.com)



@Exact\_Payments



[/company/exactpayments/](https://www.linkedin.com/company/exactpayments/)

Ver. 2023.01