ThreatModeler® | aws | Azure

DevSecOps Blueprint for CYBERSECURITY

Achieve Security by Design in 30 days

Contents

Introduction

Increasingly, cybercriminals are finding new ways to compromise the privacy and confidentiality of organizations. Not only are cybercrimes increasing in volume, they are becoming more severe. According to the 2019 Cost of a Data Breach Report (Ponemon Institute, sponsored by IBM Security), there was a 21% increase in data breaches worldwide caused with malicious or criminal intent. Taking steps to implement a robust cybersecurity program will help an organization to protect the data it processes. By strengthening an organization's cybersecurity posture, CISOs can prevent data breaches, ensure business continuity and save on overall costs.

With infrastructure evolving — in the cloud, and with containers and microservices, for example — DevOps teams are aligning security with their software development life cycle (SDLC). Stakeholders are vested in increasing efficiency, reliability and availability, while maintaining data integrity. Organizations are also finding ways to detect security threats in workload environments and exert security controls from end-to-end.

DevSecOps takes the traditional approach of security testing, verification and control and "bakes it in" as early as the planning stages, as opposed to "latching it on" towards the end. This White Paper will explain how organizations are developing application workloads in their cloud deployment and benefiting. We will also provide a road map to transition to DevSecOps within the cloud. This White Paper will also reveal how to:

- **Apply DevSecOps to the entire CDLC**
- **Integrate a comprehensive, inclusive DevSecOps program**
- **Introduce tools that automate your DevSecOps program**

What is DevOps?

The goal of DevOps is to integrate the software development (Dev) process with IT operations (Ops). Enterprises that are interested in shortening software development life cycle (SDLC) time — including the release of features, patches and updates — will implement DevOps. The application of DevOps removes boundaries between dev and ops silos.

DevOps teams have security demands that need to be met due to compliance, or promises made to customers for particular levels of security. In addition to enabling continuous integration and continuous development (CI/CD), the benefits of DevOps include:

- Speed and rapid deployment, including time-to-market
- More stable, more reliable operating environments
- Reduced new release failure rate
- Higher quality output
- More time devoted to achieve business objectives, including innovation
- Less time needed for downtime, recovery, maintenance and bug fixes

The concept of DevOps began in 2008, and is lean by approach

Within DevOps engineering, developers apply the agile methodology in small, but frequent development iterations. DevOps enables programmers to work in a testable environment that mirrors the operations environment. The DevOps deployment framework can exist via on-premise, private cloud, public cloud or hybrid.

More Organizations are Migrating to the Cloud

DevSecOps is Replacing DevOps as the Preferred Approach

The software development climate is constantly changing as more organizations migrate to the cloud. Cloud-based application development, also known as the cloud development life cycle (CDLC), is paving the way for organizations to easily scale, e.g., meet peak data processing demands. CDLC offers a flexible place to store content. In most cloud provider arrangements, you pay for what you need, making scalability more accessible as your organization grows.

CDLC also enables developers and cloud security architects (CSAs) to more easily manage updates, bug fixes and improvements with as little downtime as possible. Based on your needs, you can opt for:

1. **Private cloud** – an entire cloud solution that is allocated to a single organization's data, with the data either stored on-premise or offsite through a third-party Cloud Service Provider (CSP). Private cloud offers more control, for example,over cloud protocols, configurability and metrics.
2. **Public cloud** – each public cloud customer has their own service level agreements(SLAs) with the CSP. In this instance, CSPs are responsible for the security "of" the cloud and the customer is responsible for the security "in" their cloud. Public cloud clients benefit from the elasticity needed to adjust to workflow demands.
3. **Hybrid cloud** – a mix of private and public clouds, hybrid cloud stores data on-premise and off-site. Hybrid enables enterprises to exert more control, such as with private SaaS customers, who require that data be kept in private clouds. While extra control is an added benefit, configuration and integration issues may come into play depending on the complexity of IT infrastructure.

Security Becomes a Key Concern in Technology Development Life Cycles

Security was previously considered separate from DevOps. Leaders in the DevOps space, such as Gene Kim, considered DevOps, with security embedded, a necessary step to evolve. Gradually, a change in development approaches occurred, as development teams started speeding up the delivery of infrastructure. People also started talking about tooling the infrastructure. The emergence of CI/CD made pre-decisioning and security necessary, while removing the human element.

Several factors that needed addressing became clear. With static analysis tools, it takes time to process all the data. Security teams look for application bugs that stop them from guaranteeing a secure application. What is considered a showstopper? What is presenting risk?

In the event of a data breach, an enterprise faces steep fines, sometimes for millions of dollars. Mandated in May of 2018, the EU General Data Protection Regulation (GDPR) imposes fines of €20 million (approximately $24 million) or 4% of a company's global annual revenue — whichever is greater — in the event of a data breach.

Additionally, an enterprise faces damage to its reputation and an overall loss of consumer trust. Increasingly, organizations are implementing measures to make their DevOps programs more cybersecure.

> **Due to its elasticity, CDLC enables developers and cloud security architects (CSAs) to more easily manage updates, bug fixes and improvements with as little downtime as possible.**

Security and Compliance Become Primary Enforcement Objectives

That's where DevSecOps comes in. Within DevSecOps, security and privacy measures are administered throughout the entire CDLC process. DevSecOps helps to foster engagement and collaboration across the enterprise.

Plan

Define business objectives and requirements. Determine business metrics. Establish the security policy and requirements.

Code

Design the software and create the application code. Automate certain developer code activities such as: compliance, infrastructure, security and testing.

Build

Oversee software builds, including version control. Compile, package and store code for future production release. Developers also package infrastructure components in code or package repositories that will be used in product release.

Test

Ensure that the software is built to meet the highest quality standards. Types of testing include acceptance testing,regression testing, configuration testing,and security threat analysis.

Deploy

Manage the software release, including coordinating and scheduling the release into production. Automation helps to schedule a release timeline into the targeted environment.

Operate

After release, developers manage software during production. Provisioning and configuring occur as needed to infrastructure, databases, networks and applications.

Monitor

Users across the organization provide feedback about software issues. Developers review issues related to specific releases for their impact on end-users. Documentation is an important step in monitoring, as infrastructure performance, end user experience and other metrics are recorded. This information is often used to influence Plan activities for updates and new release cycles.

Secure by Design: Security Should be Built into the Code

Secure by design is a concept asserting that security should be built-in from the foundation, i.e., as early as the planning stages. Rather than reacting to an outage or data breach, developers will spend their time implementing preventive measures. For example, designers will look for potential threats and attack vectors that contribute to an organization's risk.

Examples of threats include security design flaws, code defects and code imperfections. DevSecOps introduces shift-left practices to better secure an organization's IT infrastructure from its foundation. Shift left entails implementing security earlier in the software development process rather than towards the end. Within the shift-left approach, consider the sensitive and proprietary enterprise information plus the personally identifiable information (PII) of your customers.

An organization that has implemented DevSecOps will code security into their systems and applications, whether they are on-premise or in the cloud. The DevSecOps approach benefits an organization by:

Introducing design constraints, which leave less room for error

Addressing design flaws and code weaknesses that contribute to risk

Removing silos

Reducing the need for patches and bug fixes

Minimizing the impact of security risk

Maximizing system and application uptime

Increasing focus on strategy and innovation

Security by design is more prominent than ever as organizations implement strategies to become as resilient and resistant in their cloud to cyberattacks, old, new and emerging.

Cloud DevSecOps

When Everyone at Every Level Is Involved in Security

DevSecOps cal ls for greater collaboration, starting at the top and working its way to the bottom within an organization. It is important to maintain collaboration between developers, business owners, operations and project managers. Stakeholder engagement is crucial.

DevSecOps requires the writing of code that manages, automates and enforces policy. As part of the planning stages, collaborators can partake in processes to maximize their understanding of security threats and risks:

Risk Assessment

Identify the information assets within an organization (laptops, login credentials, customer data and servers), and the various threats that put them at risk of a data breach or cyberattack.

Threat Modeling

By using process flow diagrams, security teams can threat model their infrastructure to better understand information assets, how they are managed and secured, plus the threat vectors that may impact them. Threat modeling is an important part of security by design.

Maximize Throughput

Key stakeholders can take risk assessment and threat modeling results and, armed with a full view of the security and compliance posture, make data-driven results more quickly. Risk management decision making includes steps to mitigate, remediate or explain accepted risk.

DevSecOps Integrates Security as Much as Possible, While Ensuring Security Architects and Stakeholders Stay Informed

After identifying risk during planning, DevSecOps collaborators work together to enforce secure code writing, implement security controls and manage security checks throughout the CI/CD pipeline. Collaboration ensures that checks and balances are in place to make sure an organization is compliant with internal policy and external regulations. This includes restricting access (such as through the least privilege model), continuous monitoring (with reporting), and keeping applications updated with the latest versions containing patches and bug fixes. Cloud security architects and developers work hand-in-hand to ensure security controls are implemented throughout the CDLC process. Validation occurs through reporting and documentation designed for CISO sign-off, checking against compliance rules, mitigated threats, backlog.

Increased deployment frequency with low-risk releases

Shortened lead times

Increased threat detection

Trust, improved relationships within the organization

Empowered monitoring

Global Average Data Breach Cost to Organizations:

$3.92 Million

Achieve Flexibility and Scalability with DevSecOps in the Cloud

Leading cloud providers include: Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), Oracle Cloud and IBM Cloud. There are a number of key benefits to cloud migration, including:

- **Scalability that can be automated to handle peaks. Cost flexibility based on your needs.**
- **Continuity in the event of outages.**
- **Simplification with the ability to isolate servers for testing, maintenance and use cases.**
- **Additional support from cloud vendors.**

Global Average Cost Savings with a DevSecOps Approach for Organizations:

$280,000

DevSecOps Tools and Processes Ensure Secure CI/CD

Within DevSecOps, security and operations converge in automated and integrated workflows. Tools that enable collaboration, and quick and easy communication, should be utilized as part of a unified DevSecOps tool set. Examples include tools for compiling threat intelligence, sharing alerts and information, penetration testing and visualizing data (with dashboards).

Automation tools can harden your infrastructure, while standardizing processes and making them more efficient – enabling teams to focus on innovation rather than maintenance. Tools for processes that were formerly manual are now being automated to keep your CDLC defended. Security, from start-to-finish, cannot slow the CDLC down. Automation is crucial for the success of a CI/CD pipeline that has adopted DevSecOps. Automation brings several benefits, including increased efficiencies, shorter feedback loops, quicker threat and risk detection, and faster bug fixes.

Automation also Lends Itself to Security by Design

Validating the proper implementation of security controls is one of the most important steps.

Users can validate security controls by logging into their AWS cloud system and make surreal controls are implemented. ThreatModeler Cloud Edition includes a new feature that continuously monitors threat models, allowing users to certify the controls are implemented and they can be aware of any changes and updates. Users can review those changes and incorporate them to a new version of their current threat model.

1 ---------- 2 ---------- 3 ---------- 4

Version control (GitHub, AWS CodeCommit, Microsoft Team Foundation Server)

Continuous integration, testing and monitoring (Jenkins for CI, IBM Rational Functional Tester, Jira for defect tracking)

Configuration and provisioning (Ansible, CFEngine, Chef)

Orchestration (Microsoft Azure Cloud Management, BMC Control M, CA Workload Automation)

Advances in Threat Modeling Drives Security by Design in DevSecOps

Threat modeling became a prominent part of understanding information security threats beginning in 2011. Typically a manual, ad hoc process, advancements in threat modeling have made it an integral, necessary process in DevSecOps. Threat modeling helps to ensure that security is applied earlier rather than later, empowering organizations to better understand their attack surface and mitigate security risk.

CISOs and stakeholders will benefit from threat modeling, which enables them to fully understand the security threats that impact their infrastructure and applications. ThreatModeler™ is the industry's leading threat modeling tool, with automation features that make it well-suited to integrate within your DevSecOps driven CDLC.

ThreatModeler Uses the VAST Methodology Approach

Visual, Agile, Simple Threat modeling (VAST) represents the most comprehensive, clean and concise format. VAST utilizes process flow diagramming to deconstruct system or application architecture, which can help you to understand the different security threats from the perspective of a hacker. Since threat modeling enables you to better understand and document security threats, the activity helps DevSecOps teams to prioritize risk.

Threat modeling is a key, essential activity in shifting security left in the cloud. With the right application, you can improve security, contribute to ease of collaboration and scale across the enterprise. A person with little-to-no technical skills can create an accurate ThreatModeler process flow diagram. Users can review existing threats, keep abreast of new and emerging threats, and implement security requirements and controls to mitigate threats.

ThreatModeler seamlessly integrates with third-party pipeline tools to stay up-to-date with threats. ThreatModeler features a bidirectional API, for synchronization with CI/CD pipeline solutions such as Jira and Jenkins. ThreatModeler integrates seamlessly with third party business systems to build and test software projects.

Technology Partners: ThreatModeler and AWS

ThreatModeler's automation increases security architect output by 10 times compared to manual, ad hoc threat modeling. The automated software application also integrates with leading cloud service providers Azure DevOps and Amazon Web Services (AWS). Through its Technology Partnership with AWS, ThreatModeler stays up-to-date with the latest AWS cloud security threats and requirements.

The platform's fully-integrated AWS Assist feature enables you to create complete, accurate process flow diagrams that capture and map out AWS application deployments. It's AWS Accelerator accurately simulates and mirrors what is in the AWS environment. ThreatModeler also features continuous monitoring for material changes in your Amazon Virtual Private Cloud (VPC). It completely checks for newly placed or modified AWS process flow diagram components. In addition, AWS Assist automatically creates new tasks in the Task Pane to notify users of AWS deployment requirements.

ThreatModeler, integrated with AWS IAM, AWS SSM and AWS Security Hub, enables CSAs to enforce policy governance that is based on the least privilege model of access. For example, CSAs can review groups and individuals assigned to groups, then specify who is allowed to access what data resources and under what conditions.

At enterprise scale, a huge number of human users accessing multiple tools need permissions. CDLC and microservice-based applications add to the complexity. ThreatModeler provides CSAs with a simulated environment to visualize policy changes, see how the changes will impact the IT environment and make secure access management decisions.

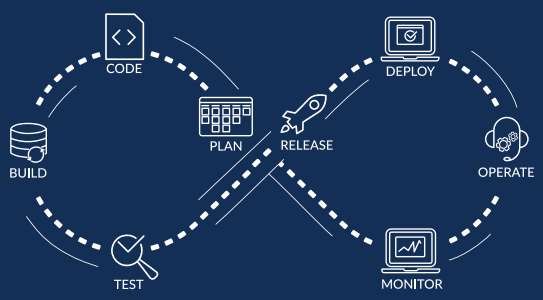
Additionally, the ThreatModeler integration with AWS SSM provides CSAs with data that is linked to the software used. With the information, CSAs can push threat mitigation as individual tasks through IT project management tools such as Jira. ThreatModeler's dashboard provides a quick overview of an organization's top threats. Users can dive deeper to better understand the threats and security requirements that will address them.

ThreatModeler's reporting module helps DevSecOps team members to view and share important information about their attack surface, including:

- **Executive Summary**
- **Requirements**
- **Identified Threats**
- **Test Cases**
- **Listing of Linked Projects**
- **Data Exposure** (of application or system being threat modeled)
- **Components**

Out-of-the-box, ThreatModeler offers a preventative approach to securing cloud environments. Due to its fast learning curve, even a novice user can complete a threat model in less than an hour. Not only can users create threat models from scratch, they can import diagrams from process flow creation applications including: Visio, LucidChart, draw.io and Gliffy.

ThreatModeler also features a Wizard, which asks the user questions to intuitively guide them through the creation of his or her threat model. Each time a user builds a threat model, the process flow diagram is saved in the ThreatModeler library. Users may then import existing threat models and build upon them, making for a completely scalable tool.



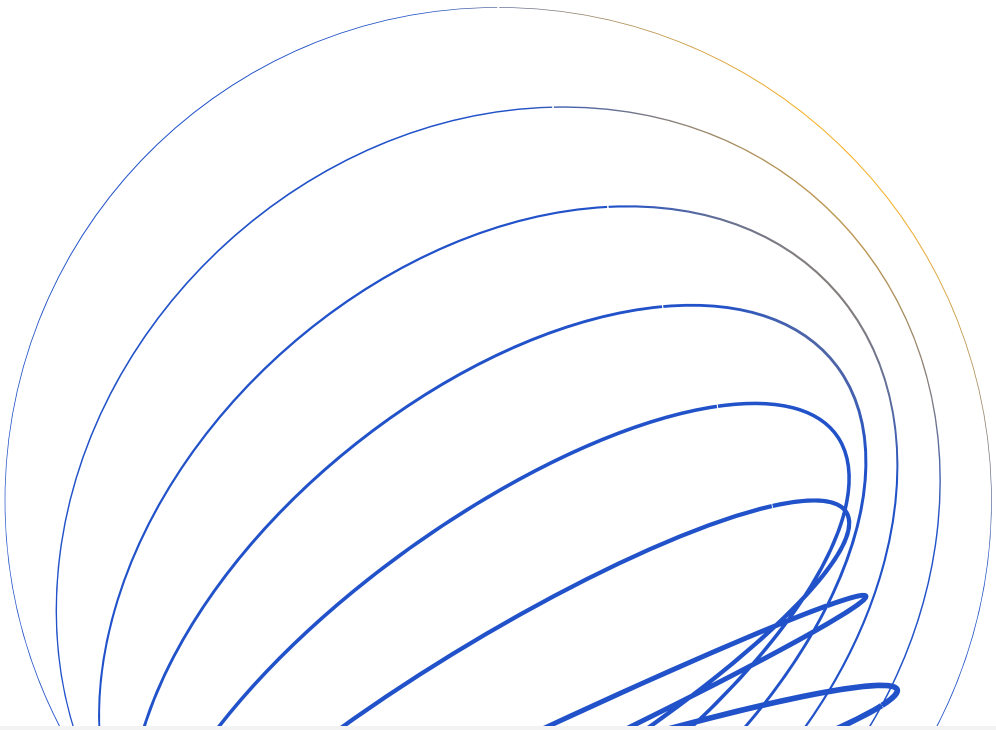
Conclusion

DevSecOps seeks to implement security privacy, policy and controls throughout their CDLCs within an organization; ultimately benefiting from the acknowledgment that their system and application architecture is protected. Over time, everyone involved in DevSecOps will know their roles and responsibilities, facilitating a culture that is collaborative and intrinsically security-minded. Organizations will also benefit from writing code that enforces security policy and compliance from the get-go.

Automation whenever possible will enable continuity and repeatability with crucial processes, providing a holistic visualization of the attack surface and accurate audit trails. Continuous collaboration on shared processes throughout the CDLC will foster a business process driven by ongoing security decisions. As you integrate automation across the organization, scale down on manual processes, freeing up resources to focus on accelerating productivity. The less humans are involved in doing the work, the more they can focus on other tasks, such as innovating and delivering high-quality applications.

Scalability is an important feature for organizations to leverage if they are making the transition to the cloud. With scalability, programmers can adjust output according to storage and/or capabilities. Developers can engineer containers — say, in the AWS cloud — with automated security activities, such as closing off compromised servers, redirecting traffic and notifying CISOs of a breach.

Threat modeling is an essential activity that helps DevSecOps teams to pinpoint security threats and the security controls that mitigate security risk. The industry leading tool easily integrates with your cloud and CI/CD toolchain. ThreatModeler can help entire organizations to shift left and achieve end-to-end security throughout their technology development life cycles.



For more information, support, or inquiries, please contact us at:

support@threatmodeler.com +1 201 266-0510 threatmodeler.com

© 2025. All Rights Reserved. Privacy Policy • Terms of Service

Transition to DevSecOps in 30 Days

Build Security Into Each Step of the DevSecOps Process

	ACTIVITY	ADDITIONAL DETAILS
☐	**Conduct a company-wide data and asset inventory of: operating systems, software, owners and administrators, locations (e.g. of hardware) and logical addresses. Categorize assets according to sensitivity level.**	The information captured will inform your security decisions. Tools and resources that contain this information include: penetration testing tools, software operating licenses, and infrastructure devices.
☐	**Assemble your DevSecOps team.**	Include CISOs, business owners, stakeholders, security specialists and software developers. Make sure to align everyone on core business objectives, including the integration of end-to-end security. Ensure each team member is security minded, and knows their roles and responsibilities as they pertain to DevSecOps.
☐	**Establish policies and procedures according to your cybersecurity needs.**	If your organization already has infosec policies in place, set a regular evaluation schedule for continual improvement. Ensure policies are followed and maintained.
☐	**Implement or improve upon an organization-wide cybersecurity training program.**	Enforce security protocol throughout the entire organization in all areas.
☐	**Planning: Gather your requirements with security and compliance in mind.**	Requirements will be functional and technical. Security requirements should address risk management issues — not only internally, e.g., confidential documents with proprietary information, but also to protect the personally identifiable information (PII) of consumers.
☐	**Set metrics and benchmark them to measure your information security maturity level.**	The US General Services Administration agency has outlined metrics that should be implemented as part of your SDLC. The more security built into your technology development life cycle, the more you can focus on other processes, such as quality improvement and innovation.
☐	**Automate core security tasks earlier in the CDLC and wherever possible.**	Automation of recurring tasks will reduce human error, harden IT systems and applications throughout the CI/CD pipeline and enable you to allocate resources to other activities.
☐	**Enable automated continuous security testing within the CI/CD pipeline.**	Continuous security testing occurs when DevSecOps teams test for bugs and defects whenever a change is made. After adopting continuous security testing, you will have more secure code, hardened infrastructure and more efficient, higher quality development life cycle output.
☐	**Code security in whenever possible.**	Security controls should be coded in that adhere to the cybersecurity triad of confidentiality, integrity and availability (CIA).
☐	**Implement continuous monitoring and logging.**	Make sure security measures are enforced through proactive monitoring, detection and analysis. Continuously monitor feedback for improvement opportunities. Cybersecurity feedback examples include user access restriction levels, number of known attack surface vectors, and meantime- to-detect (and respond).