Hype Cycle for Application Security, 2021

Published 12 July 2021 - ID G00747408 - 115 min read

By Analyst(s): Joerg Fritsch

Initiatives: Security of Applications and Data

> Security and risk management leaders need to adopt a system view of application security. They should focus on orchestrating multiple application security innovations to serve as a coherent defense, rather than relying on a set of stand-alone products.

Additional Perspectives

- Invest Implications: Hype Cycle for Application Security, 2021
  (20 July 2021)
- Summary Translation: Hype Cycle for Application Security, 2021
  (06 August 2021)

Analysis

What You Need to Know

The continued adoption of cloud-native design patterns and the mainstreaming of microservices architectures, containers and functions have accelerated the adoption of application security controls. Organizations have increased their abilities to release better software and protect it thoroughly. Gartner's Enabling Cloud-Native DevSecOps Survey for 2021 showed that more than two-thirds of the participating organizations are using static application security testing (SAST) in development to secure cloud-native applications. Web application firewalls (WAFs) and web application and API protection (WAAP) are used in production by 75% of respondents. [1]

Security and risk management (SRM) leaders must adopt a global view on application security and incorporate security from the outset in all environments. In too many instances, security is limited to the build and release management pipeline. It is now the time to focus on orchestrating multiple application security innovations to serve as a coherent defense, rather than relying on a set of stand-alone products.

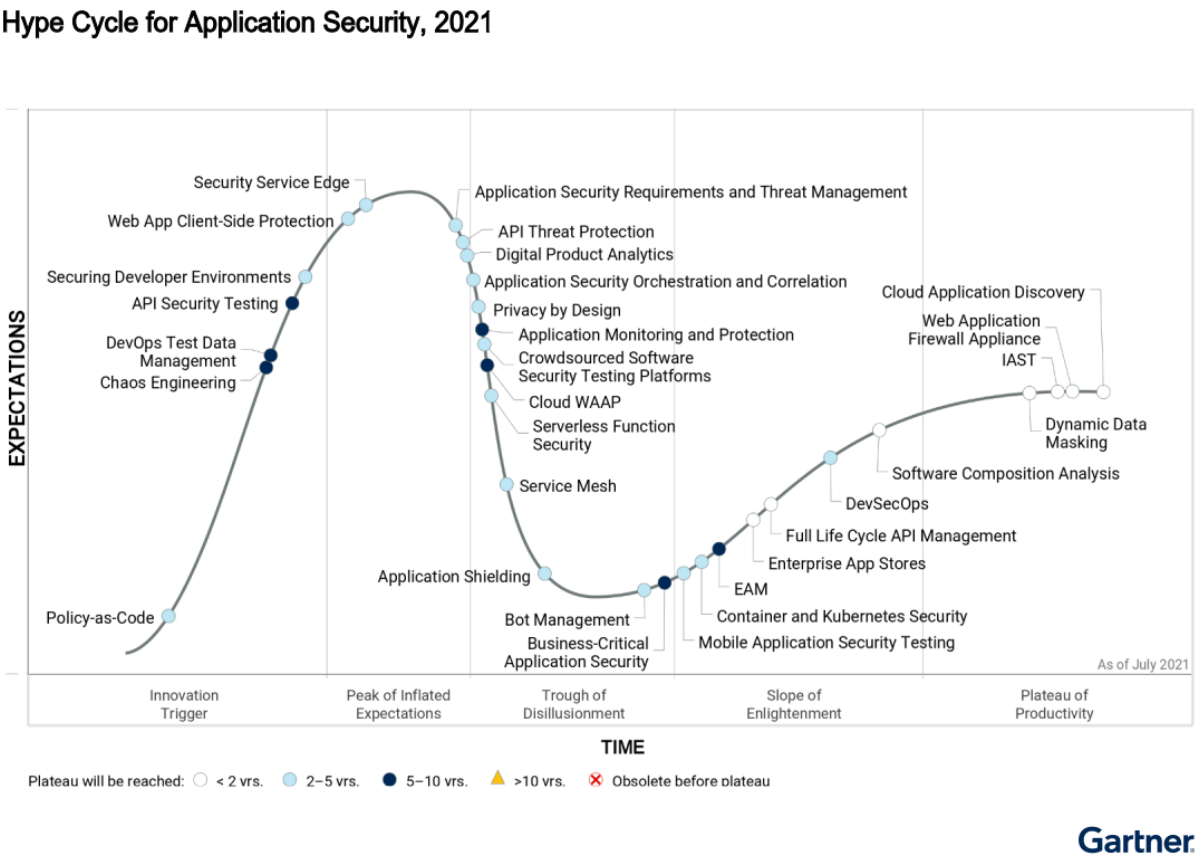
The Hype Cycle

In the 2021 Hype Cycle for application security, three new categories demonstrate the widening breadth and the increased adoption of a system view on application security. These categories are policy as code (PaC), security service edge and externalized authorization management (EAM). Gartner expects that more tooling supporting application security from a system (and systems-thinking) perspective will evolve, and also that these tools will find better hooks. This will allow these tools to integrate themselves as meaningful parts into continuous integration/continuous delivery (CI/CD) pipelines. For example, to date, application security requirements and threat management (ASRTM) tools are not very well-integrated into DevSecOps workflows.

Technologies rarely enter the Hype Cycle at the point at which EAM has entered it. Over the past year, this technology has shown extraordinary momentum, specifically around PaC, indicating that the technology is becoming relevant for application development and security. The key here is that authentication and authorization are externalized. They are running outside of the application, supporting the observation that application security is moving toward a systems view.

A lightly populated Innovation Trigger illustrates that SRM leaders, developers and vendors are still digesting and refining the deployment and maturity of a host of important technologies. These technologies have been maturing quickly, fueled by the urgency to adopt cloud-native design patterns, microservices and containers. Vendors are gaining more experience with these platforms. Consequently, CASB, for example, has morphed from a stand-alone tool to an integrated capability or feature within larger security service edge platforms. This has made the CASB innovation profile obsolete.Most innovation profiles in this Hype Cycle have passed the Peak of Inflated Expectations. The majority will reach a plateau within two to five years, or five to 10 years. This indicates that it is a good time for SRM leaders to evaluate the potential use cases for, and the maturity of, their required technologies. However, caution is needed. Risks will remain as innovations pass through the Trough of Disillusionment before reaching the plateau, so users should review their viability consistently. For example, service mesh, serverless function security, and container and Kubernetes security are currently sliding into the Trough of Disillusionment. Hype has turned to reassessment, and, in some cases, a certain amount of doubt has crept in, as these technologies continue to mature quickly and traditional development teams struggle to operationalize them. Not to be overlooked are the market mature profiles that have become foundational to enabling mature application security strategies. Their value and use are well-understood. SRM leaders that have not implemented them yet must put them onto their very near-term application security roadmap, since there is little excuse for not having them. Good examples of such technologies are web application firewall appliances and static application security testing.

Figure 1: Hype Cycle for Application Security, 2021



Source: Gartner (July 2021)

Downloadable graphic: Hype Cycle for Application Security, 2021

The Priority Matrix

DevSecOps is the transformational technology that enables security teams to keep pace with development and operations teams in modern development, and to deliver deep integration and automation of security tooling. Lessons learned in implementing DevSecOps processes can also be applied to more traditional development models — boosting the security of those applications as well. The range of tools and services profiled in this research should be used to deliver and operate secure software for the organization and its clients.

Most modern applications are part of a larger (often unrecognized) ecosystem — whether it's an enterprise, a set of managed services or a mobile device. As a result, no single application security innovation can deliver the major shift in industry dynamics that a transformational technology requires.

Table 1: Priority Matrix for Application Security, 2021

(Enlarged table in Appendix)

Benefit	Years to Mainstream Adoption			
↓	Less Than 2 Years ↓	2 - 5 Years ↓	5 - 10 Years ↓	More Than 10 Years ↓
Transformational		DevSecOps		
High	Full Life Cycle API Management IAST Software Composition Analysis Web Application Firewall Appliance	API Threat Protection Application Security Orchestration and Correlation Application Security Requirements and Threat Management Application Shielding Bot Management Crowdsourced Software Security Testing Platforms Policy-as-Code Security Service Edge Web App Client-Side Protection	API Security Testing Chaos Engineering Cloud WAAP	
Moderate	Cloud Application Discovery Dynamic Data Masking Enterprise App Stores	Container and Kubernetes Security Digital Product Analytics Mobile Application Security Testing Privacy by Design Securing Developer Environments Serverless Function Security Service Mesh	Application Monitoring and Protection Business-Critical Application Security DevOps Test Data Management EAM	
Low				

Source: Gartner (July 2021)

Off the Hype Cycle

- **Cloud security access broker (CASB)** — For application security, the innovation profile for secure service edge is subsuming CASB. This is because many secure service edge platforms include CASB functionality.

- **Mobile threat defense (MTD)** — MTD has been replaced by two innovations that are more application-focused. Web application client-side security and application shielding have replaced MTD.

- **Runtime application self-protection (RASP)** — RASP has been removed from this year's Hype Cycle.

On the Rise

Policy-as-Code

Analysis By: Paul Delory

Benefit Rating: High

Market Penetration: 1% to 5% of target audience

Maturity: Emerging

Definition:

Policy-as-code languages express governance and compliance rules as code, so they can be enforced programmatically by automation tools. PaC languages are often domain-specific and declarative. With PaC, policies are treated as software, making them subject to version control, code review and functional testing. The most mature PaC tools can render any business logic in code. You can use them to enforce architectural standards, corporate policies, regulatory requirements, budgets and more.

Why This Is Important

In the most mature automation pipelines, I&O engineers mostly spend time on optimization, governance and compliance. They no longer build infrastructure; that work has been automated and turned over to end users. Now, I&O builds the guardrails around the infrastructure services that their end-users consume. I&O must align with security and compliance teams. PaC brings policy enforcement into their automation pipelines, while preserving a separation of duties that mirrors a typical IT org chart.

Business Impact

- **Security, compliance and automation:** PaC combined with infrastructure automation provides direct enforcement of policies with implicit compliance guarantees.
- **Alignment of security and Ops teams:** PaC allows security and compliance teams to interface directly with automation pipelines to ensure conformance.
- **Visibility and auditability:** PaC provides both unambiguous documentation of policies and evidence they are being enforced.
- **Less toil:** PaC reduces the overhead of creating and enforcing policies.

Drivers

- **New PaC tools:** Several dedicated PaC tools are now on the market, many of them open source. Most prominent is the Open Policy Agent, a Cloud Native Computing Foundation project. But other options abound, including: InSpec (part of the Chef suite), Sentinel (part of the HashiCorp suite) and Bridgecrew (recently acquired by Palo Alto). All of the major hyperscale public clouds also have their own native policy engines.
- **Increasing regulation:** The advent of new regulations such as GDPR has increased both the difficulty of compliance and the pressure on compliance teams. PaC allows compliance teams and auditors to document their policies in detail, and to verify they are being enforced.
- **Security breaches:** Similarly, a spate of newsworthy security breaches at public companies have put every IT organization's security and compliance practices under increased scrutiny. No I&O team wants their security failures to be the reason their company ended up in the headlines.
- **Growth of DevOps and DevSecOps:** More and more companies are embracing DevOps and DevSecOps — which means more and more companies are encountering the hard governance problems of automation, which PaC can help solve.
- **Cloud optimization and cost control:** Beside their benefits for security and compliance, PaC tools can also be used to enforce the I&O department's build standards for infrastructure — including enforcing budgets. In the public cloud, where oversized or unnecessary infrastructure incurs direct out-of-pocket costs, programmatically enforced policies can help to control spending.

Obstacles

- **Immaturity of tools:** PaC tools are not yet fully ripe. They will not gain real traction until they have an extensive library of community-generated content. Ideally, users would simply download the policies they need from a free, public repository — rather than having to write their own policies. Over time, as the user base expands and commercial offerings see increased adoption, PaC tools will reach a critical mass of downloadable content that enables real-world use cases.

- **Skill set:** Many I&O professionals lack the skills to operate automation and PaC tools effectively. Gartner clients routinely report that their automation and policy management are hindered primarily by people, not tools. PaC will magnify these existing skills challenges.
- **Organizational inertia:** PaC promises improved collaboration between I&O and security/compliance teams. But in some organizations, this change would be unwanted. Internal resistance of this kind will slow the rate, scope and scale of PaC initiatives.

User Recommendations

- **Start small:** Choose a pilot use case where PaC is likely to provide real business benefits. Once PaC has proven its value, expand to other use cases over time.
- **Prioritize existing templates:** Focus your PaC efforts on use cases that have ready-made implementation templates — ideally, publicly-available downloadable content. For example, almost every PaC tool on the market has a canned implementation of the CIS benchmarks.
- **Break down team silos:** Use PaC to build a common workflow for automation and policy enforcement that spans I&O, security and compliance teams.
- **Integrate PaC into automation pipelines:** To automate a process, you must know *what* to do before you can determine *how* to do it. PaC creates specific and enforceable guidelines for automation tools to follow, solving this problem.
- **Measure before and after:** Use observability tools and value stream mapping to define your starting state, then compare it to the end state. Collect real data to quantify the value of PaC.

Gartner Recommended Reading

Using Cloud-Native 'Policy as Code' to Secure Deployments at Scale

Innovation Insight for Continuous Compliance Automation

Market Guide for DevOps Value Stream Delivery Platforms

2021 Planning Guide for Security and Risk Management

2021 Planning Guide for IT Operations and Cloud Management

Chaos Engineering

Analysis By: Jim Scheibmeir, Dennis Smith

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

Chaos engineering is the use of experimental and potentially destructive failure testing or fault injection testing to uncover vulnerabilities and weaknesses within a complex system. It is systematically planned, documented, executed and analyzed as an attack plan to test components and whole systems both before and after implementation. Chaos engineering is often utilized by site reliability engineering teams to proactively prove and improve resilience during fault conditions.

Why This Is Important

Many organization's stake success on test plans that overemphasize software functionality and underemphasize the importance of validating the system's reliability. Chaos engineering (CE) moves the focus of testing a system from the "happy path" — instead, testing how the system can gracefully degrade or even continue to be useful while under various levels of impact. CE can also help identify where product documentation is less than sufficient or understanding of a system is lacking or siloed.

Business Impact

With chaos engineering, we minimize time to recovery and change failure rate, while maximizing uptime and availability. Addressing these elements helps improve customer experience, customer satisfaction, customer retention and new customer acquisition.

Drivers

- Gartner's 2020 Achieving Business Agility with Automation, Continuous Quality and DevOps Survey found that 18% of respondents were currently using or planning to use chaos engineering to improve software quality.
- Complexity in systems and increasing customer expectations are the two largest drivers of this practice and the associated tools. As systems become more rich in features, they also become more complex in their composition, and more critical to business success.
- Overall, chaos engineering helps organizations create more reliable systems and prove that systems are reliable.

Obstacles

- The first obstacle to chaos engineering is perception. Within many organizations, the predominant view is that the practice is random, done first in production, and due to these leads to more risk than less.
- Another obstacle to chaos engineering is organizational culture and attitude toward quality and testing. When quality and testing are only viewed as overhead, then there will be a focus on feature development over application reliability.
- Another common obstacle is simply gaining the time and budget to invest in learning the practice and associated technologies. Organizations must reach minimum levels of expertise where value is then returned.

User Recommendations

- Utilize a test-first approach by practicing chaos engineering in preproduction environments. Then move findings into production environments.
- Incorporate chaos engineering into your system development or testing process.
- Build-out incident response protocols and procedures, as well as monitoring, alerting, and observability capabilities in tandem with advancement of the chaos engineering practice.
- Utilize scenario-based tests — known as “game days” — to evaluate and learn about how individual IT systems would respond to certain types of outages or events.
- Investigate opportunities to use chaos engineering in production to facilitate learning and improvement at scale as the practice matures. However, be warned that Gartner believes that there are still very few organizations purposely using chaos engineering in their production environments.
- Formalize the practice by adopting a platform or tool to track the activities and create metrics to build feedback for continuous improvements.

Sample Vendors

Alibaba Cloud; Amazon Web Services; ChaosIQ; Gremlin; steadybit; Verica

Gartner Recommended Reading

Improve Software Quality by Building Digital Immunity

Innovation Insight for Chaos Engineering

How to Safely Begin Chaos Engineering to Improve Reliability

DevOps Teams Must Use Site Reliability Engineering to Maximize Customer Value

DevOps Test Data Management

Analysis By: Dale Gardner

Benefit Rating: Moderate

Market Penetration: 20% to 50% of target audience

Maturity: Early mainstream

Definition:

DevOps test data management is the process of providing DevOps teams with data to evaluate the performance, functionality and security of applications. It typically includes copying production data, anonymization or masking and virtualization. In some use cases, specialized techniques, such as synthetic data generation, are appropriate. Given potential compliance and privacy issues, the process more frequently involves members of application and data security teams.

Why This Is Important

Test data management is inconsistently adopted across organizations, with many teams copying production data for use in test environments. As organizations shift to DevOps and the pace of development increases, this traditional approach is increasingly at odds with requirements for efficiency, privacy and security, and even the increased complexity of modern applications. This opens organizations to a variety of legal, security and operational risks.

Business Impact

Quick provisioning of test data helps ensure the pace of development isn't slowed. It's also increasingly important to remain compliant with the growing number of privacy mandates to which organizations are subject. This helps avoid fines and remediation and mitigation costs, along with the inevitable delays associated with audits and investigations. Finally, by providing application teams with anonymized or synthetic data, the risk of data breaches is reduced.

Drivers

- Test data management is generally viewed as a mature, relatively uncomplicated, technology. However, the reality is the combination of the increased pace of development from DevOps and a growing number of privacy mandates and constraints have strained traditional approaches — yielding new approaches and technology.
- More traditional test data management has been inconsistently adopted, with many organizations either simply using copies of production data or generating "dummy data" (distinct from emerging synthetic data generation techniques). The data privacy requirements and complexity issues noted have prompted organizations to revisit and update their processes with an eye toward scalability and automation. Updated technologies may also be a requirement. For example, requirements for speed and agility have created a need for data virtualization tools.
- Data protection is cited by a growing number of clients in inquiries regarding test data management. Privacy and data protection requirements mean it's no longer safe to simply provide development teams with a copy of production data. The practice leaves organizations open to increased risk of regulatory violations, data breaches and other security issues.
- With modern applications relying on an increasing number of interconnected data stores (many of which themselves are technologically vastly more complex), applications, and APIs to function, testing has become more complex. Such complexity demands tools support the ability to coordinate and synchronize changes across different data stores to ensure relational consistency while still addressing security and speed mandates.

Obstacles

- In the absence of a strong culture of security, processes and technologies to protect sensitive information during development and testing will encounter friction. Conflicting needs for rapid development and privacy require attention to a mix of organizational and cultural issues to strike a balance across groups.
- Responsibility for test data management in organizations has been shared by application development and database administration. New technologies and processes may shift those responsibilities to include security, complicating organizational dynamics and potentially creating the need for additional staffing.
- Implementation can be a burden, especially where little or no data sensitivity classification has been done. This must be accomplished before teams can proceed with required data transformation and masking. These efforts are typically combined with an analysis of data relationships, so that relational integrity can be assured.

User Recommendations

- Involve stakeholders — application development and test teams (to understand consumption patterns and needs), data management, privacy and security teams, compliance teams, other security teams, and legal counsel — as appropriate.
- Document existing test data management practices so tools and processes can be evaluated against data protection mandates.
- Coordinate to avoid duplication of effort and tooling since data masking tools may also be used by analytics teams (e.g., to provide data for machine learning or other purposes).
- Evaluate data masking tooling by considering support for databases and other stores, data discovery capabilities, types of masking supported and the ability to coordinate change to ensure consistency (e.g., key fields) across multiple sources.
- For DevOps use cases where frequent updates to test data are required, data virtualization may also be useful. Virtualization can speed the process of providing copies of safe data.

Sample Vendors

Actifio; BMC Compuware; Broadcom (CA Technologies); Delphix; Hazy; Informatica; MENTIS; Micro Focus; Solix Technologies

Gartner Recommended Reading

Market Guide for Data Masking

Elevating Test Data Management for DevOps

API Security Testing

Analysis By: Dale Gardner

Benefit Rating: High

Market Penetration: 1% to 5% of target audience

Maturity: Emerging

Definition:

API security testing is a specialized type of application security testing (AST) aimed at identifying vulnerabilities in application programming interfaces (APIs). Checks should include both traditional application vulnerabilities (e.g., injection attacks) and API-specific issues (e.g., broken-object-level authorization). Availability of an API definition (e.g., OpenAPI) is often, but not always, a prerequisite for effective testing; discovery helps ensure that unknown APIs are identified.

Why This Is Important

Attacks on applications are shifting to APIs, and their pace is increasing. By 2022, API abuses will be the most-frequent attack vector, and will result in data breaches. DevSecOps teams are focusing attention on the need for improved API testing in development, looking to a mix of traditional tools — e.g., static AST (SAST) and dynamic AST (DAST) — and emerging solutions focused on the requirements of API testing to identify the optimal approach to API testing.

Business Impact

APIs are a foundational element of many organizations' digital transformation efforts. Hence, securing APIs from attack and misuse is a growing concern for many security and risk management (SRM) professionals. API-specific testing, pre- and post-deployment, builds a solid foundation for an overall API security strategy. API discovery is needed, because many organizations struggle to maintain an inventory of APIs and need help locating them and ensuring they are tested and managed.

Drivers

- APIs enable information flow and support transactions between processes, programs and systems, so APIs have become common in application architectures, especially in support of digital transformation efforts. However, that growth has brought increased attention from attackers, and APIs have become the primary attack surface for many systems. Gartner has long noted that attacks and abuse against APIs are a common attack vector.
- These attacks have resulted in an endless stream of data breaches and other security incidents, yielding significant damage to organizations and individuals. As a consequence, DevSecOps teams — along with the business leaders whose applications are supported by APIs — are increasingly interested in API testing and security.
- Because the creation, development and deployment of APIs may be loosely managed, security teams often have to contend with unknown APIs, which exist outside normal processes and controls. Such APIs may be especially deficient in secure design and testing, posing an even greater risk. Hence, the discovery of APIs deployed to production is another important need.
- Given their increasing importance in application architecture and their sensitivity, identifying and remediating exploitable vulnerabilities in APIs has become an obvious goal. API security testing helps avert the tangible and intangible costs associated with breaches and other types of security incidents.
- Traditional AST tools — SAST, DAST and interactive AST(IAST) — were not originally designed to test for vulnerabilities associated with typical attacks against APIs, or for newer types of APIs (e.g., GraphQL). Although vendors have added support, these gaps have created an opportunity for specialist API security vendors, focused on testing, discovery of running APIs and protection from threats — or some combination of the three.

Obstacles

- Existing tools must be extended to address API-related vulnerabilities, and incorporate new testing approaches. APIs are susceptible to most, if not all, traditional attacks against applications; however, newer attack types have also emerged. For example, improper authorization checks around object identifiers (such as a user identifier) have been cited in a number of breaches.
- APIs in a number of forms are using various protocols. Simple Object Access Protocol (SOAP) remains in widespread use, although has been supplanted by Representational State Transfer (REST) APIs. GraphQL (a combination of query language and runtime) is now challenging REST. This requires additional support from tool vendors for effective tests.
- There is confusion in the marketplace, with various types of companies (traditional AST vendors, along with API testing and protection vendors, etc.) offering products. This complicates evaluation and selection efforts.

User Recommendations

- Do not invest in tools without a solid inventory of APIs — unless the tool is specifically intended to help address shortcomings in this area.
- Assess the overall role APIs play in your application portfolio —their criticality to the organization, their security and business risk, and the technical requirements they pose — to begin evaluation and selection efforts.
- Examine the testing capabilities provided by existing tools in your application security portfolio. Many have added support for APIs, although actual tests may still focus primarily on traditional application vulnerabilities.
- Where gaps are found, evaluate newer alternatives. They may also offer additional options, including audit of design-stage API specifications, as well as vulnerability scans and threat protection.
- API security tool capabilities — including discovery, testing and threat protection — overlap, so evaluate the ability of a given tool to address multiple requirements.

Sample Vendors

42Crunch; APIsec; Checkmarx; Contrast Security; Data Theorem; Imvision; NTT Application Security; Salt; Synopsys; Veracode

Gartner Recommended Reading

API Security: What You Need to Do to Protect Your APIs

Solution Path for Forming an API Security Strategy

Magic Quadrant for Application Security Testing

Critical Capabilities for Application Security Testing

Securing Developer Environments

Analysis By: Mark Horvath, Michael Isbitski

Benefit Rating: Moderate

Market Penetration: 1% to 5% of target audience

Maturity: Emerging

Definition:

Securing developer environments, which often contain artifacts like proprietary IP or trade secrets, is often overlooked by both developers and IT security. Securing these artifacts as well as source code repositories, scripts, infrastructure as code is critical in a world of remote workers and supply chain attacks.

Why This Is Important

While developers have tools for writing, creating and managing code and data, IT security is often unaware of the security posture and risks of these tools and hard-pressed to secure them. This is even though code repositories often contain proprietary IP and trade secrets. Solutions exist for securing other types of data, like confidential documents, business plans and CRM data; however, developer artifacts are often ignored, and few tools have migrated from traditional forms of IP protection (like DLP) into the DevOps environment.

Business Impact

The overall attack surface is expanding because of:

- The high number of complex tools in DevOps which may hide malicious code to exfiltrate data.

- Unsecured code repositories.
- Remote development environments which are more difficult to secure.
- Sharing of code and other information with the OSS community that may contain proprietary information.
- Increased incidence of software supply chain attacks.

Drivers

- 2020 and the COVID-19 pandemic forced a lot of software developers to work from home, keeping proprietary IP on laptops connected through their home network.
- Intellectual property is increasingly being realized in soft, easily copied assets like scripts, code and configuration files.
- Cloud native development often has a significantly expanded attack surface, in part because assumptions about the traditional, on-premises defense in depth are no longer valid.
- DLP solutions, while good for documents, are often not an ideal fit software code repositories and are largely incompatible with some modern development styles like containers and microservices.
- Increased decentralization of the developer workforce due to work from home and large-scale collaboration/contracting development.
- Supply chain attacks have increased awareness of the vulnerability of code to malicious tampering.

Obstacles

- Source code repositories and version control systems are usually widely available across an enterprise with very few controls around fine-grained entitlements.
- CI/CD pipelines, containers and registries are all difficult to lockdown, especially since they are optimized for speed and security is often viewed as friction.
- Developers often have root access on their local machines, making it easy to thwart controls.
- Attack vectors from new tools like CI/CD pipelines and code repositories are often difficult to recognize by developers who don't see them as unsafe and by security who may not have a deep familiarity with the tools.

User Recommendations

- Leverage the added security capabilities of commercial distributions, such as commit integrity checks and access control integrated with external IAM and entitlements management.
- Use secret management tooling to keep secrets and sensitive data outside of source code and infrastructure artifacts.
- Adopt an immutable infrastructure approach, leverage hardened baselines within infrastructure automation practices and maintain vetted builds in VCSs for organizational teams to provide consistent, "known good" baselines and to avoid configuration errors and unauthorized alterations.
- Apply strict controls at build, delivery and runtime to account for new dependencies or environment drift, treating all endpoints as potentially hostile.
- Secure supporting CI/CD systems by enforcing access control, enabling logging and auditing, and ensuring the underlying software is current. Pipeline definitions should be governed and monitored to ensure unauthorized parties don't make changes.

Gartner Recommended Reading

Best Practices for Securing Continuous Delivery Systems and Artifacts

At the Peak

Web App Client-Side Protection

Analysis By: Dionisio Zumerle

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Emerging

Definition:

Web app client-side protection is protection against application-level attacks that initiate on the client side of the application, rather than on the server side. A typical example of an attack would be a malicious script injection against client-side JavaScript, such as Magecart.

Why This Is Important

Client-side attacks have proliferated in recent periods, aided by the design of modern application architectures. In particular, single-page applications migrate the control and software logic on the client side where it is exposed to attacks. For example, by injecting malicious scripts into JavaScript applications, attackers have lured thousands of visitors to banking and online commerce websites into handing over their credit card information.

Client-side security innovations protect from these attacks by monitoring the activity and identifying malicious actions and components.

Business Impact

- The adopters of client-side security are mainly enterprises that have public-facing applications with client-side logic that have enough control and software logic to become an attack target.
- Companies in financial services, ticketing and online retailers have turned to client-side security recently to protect their applications from fraudulent scripts that steal their customers' payment information.

Drivers

- The past two years have seen a rise in client-side attacks, such as the ones carried out by the Magecart groups. Attacking primarily online retail, airline and ticketing websites, these attackers inject malicious scripts that lure visitors to the websites into handing over their credit card details.
- The effects of the global pandemic have urged many retailers to speed up their digital transformation, shifting more business to online web applications, thereby making them more profitable targets.
- Despite there not being an established best practice to provide this protection, most approaches in this space gravitate to web JavaScript monitoring. After a period of observation, the protection can start identifying abnormal behavior. Depending on the settings, it can report or block it.
- Alternatives include the use of allowed component lists, including using Content Security Policy (CSP) and Subresource Integrity (SRI) as mechanisms to do so.

Obstacles

- The topic of client-side security is technologically complex. Many organizations will need time to reach the maturity and understanding needed to arrive at the conclusion that protection for client-side applications is needed.
- Client-side protection is new and maturing. The efficacy of the protection is still to be proven, along with any possible business disruption effects.
- There is a variety of approaches that all have merits, but there is no established standard that enterprises can follow yet.
- Three different entities in enterprises are involved in the buying process, from development to security to the line of business. The lack of a clear owner for this technology slows adoption down.
- Concerns about performance impacts and false positives that block business traffic often steer enterprises away from these protection products.

User Recommendations

- Identify candidate applications that might benefit from client-side security, such as JavaScript web applications that are public facing and contain software logic on the client side. Single-page web applications more broadly, as well as mobile web apps, will also be good candidates to receive this sort of protection.

- Assess your incumbent application security vendors (e.g., WAAP), and the self-defending capabilities they currently offer or plan to introduce in their roadmaps, before looking at stand-alone products.
- Implement client-side security protection for critical web applications that are used to carry out bookings or transactions. Do so favoring approaches that monitor JavaScript and identify malicious or abnormal behavior.

Sample Vendors

Akamai; DEVCON Detect; Ensighten; Imperva; Jscrambler; PerimeterX; Source Defense

Gartner Recommended Reading

Teach Your Applications the Art of Self-Defense

Defining Cloud Web Application and API Protection Services

Critical Capabilities for Cloud Web Application and API Protection

Security Service Edge

Analysis By: Neil MacDonald, John Watts

Benefit Rating: High

Market Penetration: 1% to 5% of target audience

Maturity: Emerging

Definition:

Security service edge (SSE) secures access to the web, cloud services and private applications. Capabilities include access control, threat protection, data security, security monitoring and acceptable use control enforced by network-based and API-based integration. SSE is primarily delivered as a cloud-based service and may include on-premises or agent-based components.

Why This Is Important

The overall SASE market convergence is driven by the increasing usage of cloud services, combined with a shift to remote work. SSE offerings reduce complexity and improve user experience by consolidating multiple disparate security capabilities (e.g., secure web gateways [SWGs]; cloud access security brokers [CASBs]; zero trust network access [ZTNA]; remote browser isolation [RBI]; and firewall as a service [FWaaS]) into a single-vendor, cloud-centric converged capability.

Business Impact

The shift to remote work and the adoption of public cloud services was underway and further accelerated by COVID-19. SSE allows the organization to support the anywhere, anytime workers using a cloud-centric approach for the enforcement of security policy. SSE offers immediate opportunities to reduce complexity, costs and the number of vendors.

Drivers

- Users, applications, and enterprise data are everywhere. Old data center-centric security architectures/products need adjustment. SASE offerings are the necessary adjustment.
- Organizations choose SSE when looking to add flexible cloud-based network security for users and devices without tying it to their choice of network infrastructure (e.g., if the organization has already deployed SD-WAN).
- SSE tends to have more mature security features, appealing to buyers looking for deeper security capabilities as compared to some SD-WAN vendors that only recently added a minimal set of security features to their offering.
- Zero-trust, least-privileged access based on identity and context is a core capability of leading SSE offerings.
- By consolidating vendors, organizations can reduce complexity, costs and the number of consoles used to define security policy. This also helps to eliminate risk created by gaps in coverage or inconsistencies with the use of multiple disparate offerings.
- Sensitive data inspection and malware inspection can be made consistent and in parallel across all channels of access — SaaS, internet and private applications — with better performance than doing this separately.
- Organizations improve user experience by providing exactly the same secured experience remotely, in a branch or in the main office.

Obstacles

- Some organizations want to strategically combine and unify their secure access strategy using SD-WAN and SSE from a single vendor rather than relying on two separate vendors.
- Most leading SD-WAN vendors now have a set of SSE services natively or through partnerships, placing competitive pressure on SSE vendors to add basic SD-WAN capabilities.
- Because the market is being formed by convergence of capabilities, most vendors are better in a single category and have gaps in other categories. Further, some vendors don't yet have a complete suite of SSE services (e.g., they are missing FWaaS or other security services).
- Some vendors are weak in sensitive data identification and protection, and this capability is critical for risk- and context-based access decisions.
- Being cloud-centric, SSE typically doesn't address the need for on-premises (e.g., file servers) and endpoint DLP.
- Not all vendors will commit to performance SLAs on all services.

User Recommendations

- Consolidate vendors, and cut complexity and costs as contracts renew for SWGs, CASBs and VPNs (replacing with a ZTNA approach). Leverage a converged market that emerges by combining these services.
- Inventory equipment and contracts to implement a multiyear phase out of on-premises perimeter and branch security hardware in favor of cloud-based delivery of SSE. Target consolidation of on-premises equipment ideally to a single appliance.
- Actively engage with initiatives for branch office transformation, SD-WAN and Multiprotocol Label Switching (MPLS) offload in order to integrate cloud-based SSE into the scope of project planning.

Sample Vendors

Bitglass; Broadcom (Symantec); Forcepoint; iboss; McAfee; Menlo Security; Netskope; Palo Alto Networks; Versa; VMware; Zscaler

Gartner Recommended Reading

2021 Strategic Roadmap for SASE Convergence

Magic Quadrant for Secure Web Gateways

Magic Quadrant for Cloud Access Security Brokers

Critical Capabilities for Cloud Access Security Brokers

Market Guide for Zero Trust Network Access

Application Security Requirements and Threat Management

Analysis By: Dale Gardner

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

Application security requirements and threat management (ASRTM) tools automate the creation of security requirements and threat models. ASRTM can integrate with SDLC tools to manage requirements and perform validation. Tool updates are derived from changes to regulations, platforms, frameworks, and evolving threats. ASRTM dynamically highlights potential security ramifications of functional requirements and recommends appropriate secure coding practices or architectural counter measures.

Why This Is Important

Threat modeling and security requirements creation are essential in the effort to create secure applications. ASRTM tools facilitate these processes by shifting security even further left, beyond the start of the SDLC. While ASRTM does not secure code, it eases the task of creating secure code and application architectures. It also helps ensure appropriate security requirements, in contrast to broad standards that inadequately secure high-risk apps, while over-burdening low risk projects.

Business Impact

Automated ASRTM tools significantly decrease the effort to create and maintain threat models, security requirements, and risk assessments. This ensures security specifications that are specific to individual projects can be defined early, while costs and risks are low — rather than after. This offers significant benefits to multiple parts of the organization, including architects, developers, security teams and even business stakeholders.

Drivers

- Threat modeling is a best practice in application development. Despite this, adoption has been restrained by the generally manual nature of the task, and the need to involve multiple individuals. The time and complexity required deter the effort for all but the most critical applications.
- The ready availability and increasing sophistication of ASRTM tools enable individuals or a small group to carry out threat modeling and requirements generation with a fraction of the time and effort typically required. They can also make the task much more approachable for architectural and development staff that may lack training in application security concepts and requirements.
- Organizations of all kinds continue to struggle to create secure applications. Issues include both the creation — and, hopefully, detection and elimination — of inadvertent vulnerabilities, as well as essential design flaws leaving an application susceptible to attack. ASRTM can assist with both problems. With relatively limited effort — especially as compared to manual approaches — the tools can generate relevant security-related requirements aligned to the threat model and attack surface of a given application. The data generated in the process can also guide triage and assessment of vulnerabilities discovered, based on their potential impact.
- ASRTM tools can also be integrated into a variety of SDLC tools, as well as with integrated risk management systems. In the first case, this can ensure the simplified transfer of functional requirements and user stories to developers for inclusion on development activities. Links to testing frameworks can also help ensure that security requirements have been fully and properly implemented. In the latter case, links to risk management aid in reporting on the specific risks associated with an application, and elevate visibility to management.

Obstacles

- Capabilities vary. Free and open source tools enable easy adoption, but fall short when representing complex systems. That prompts consideration of commercial tools, although limitations constrain suitability for advanced users.
- Another challenge is awareness. In the face of demands for improved application security, most organizations continue to focus on application security testing tools. While an essential element of a program, such tools fail at identifying design flaws, and produce more output than can be effectively managed in rapidly moving DevOps environments.
- Currency — the ability to accurately represent a rapidly changing application — remains an Achilles heel for tools. A threat model is only so good as its ability to provide an accurate representation of a system — and the bulk of tools today require effort to update models as applications change, leading to abandonment of the effort. Tool vendors have begun to link systems directly to cloud platforms, ensuring changes are reflected automatically in the model, producing updated guidance.

User Recommendations

- Treat threat modeling, security requirements generation and enforcement as best practices within a secure software development life cycle model. These tools can be leveraged to help automate these tasks while incorporating content knowledge from the tool vendors on emerging threats and security requirements.
- Use a risk-based approach, which the tools help enable, to align the level of effort in threat modeling with the risk posed by the application.
- Leverage ASRTM tools to automate what are otherwise manual or overlooked efforts, in order to ensure threat modeling and security requirements generation activities are incorporated into their software development life cycle process and development workflow.
- Train development, engineering, operations, and architectural staff and the use, and value, of the tools and encourage their use early, and continuously, in the development process and post deployment (to validate application threat protection efforts).

Sample Vendors

foreseeti; IriusRisk; Microsoft; OWASP Foundation; Security Compass; we45

Gartner Recommended Reading

12 Things to Get Right for Successful DevSecOps

Use Threat Modeling to Teach Secure Design (ADP)

5 Ways EA Can Help the Organization Focus on Security

API Threat Protection

Analysis By: Mark O'Neill, Jeremy D'Hoinne

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

API threat protection is the defense of web APIs from exploits, abuse, access violations and denial of service (DoS). It is required both for external and internal APIs. API gateways, web application and API protection (WAAP), and specialist API security tools provide API threat protection through a combination of content inspection of API parameters and payloads, traffic management, and traffic analysis for anomaly detection.

Why This Is Important

Every connected mobile, modern web or cloud-hosted application uses and exposes APIs. These APIs are used to access data and to call application functionality. APIs are easy to expose but difficult to defend. This creates a large and growing attack surface, leading to a growing number of publicized API attacks and breaches. Traditional network and web protection tools do not protect against all the security threats facing APIs, including many of those described in the OWASP API Security Top 10.

Business Impact

Because APIs are typically used for access to data or application functionality, often linked to systems of record, the impact of an API breach can be substantial. Privacy regulations typically require reporting if private data is breached through an insecure API. APIs are easily and intentionally programmable, so a vulnerability can leak large volumes of data. That it can be challenging to separate valid API use from nefarious access raises the risk of blocking valid use.

Drivers

- APIs account for increasing amounts of Internet traffic. Analysis by Akamai, a content delivery network provider, indicates that API requests make up 83% of all hits, with API hits growing by 30% year-over-year and expected to reach 42 trillion hits by 2024 (see The State of the Internet).
- The OWASP API Security Top Ten has driven awareness of the requirement for API threat protection.
- A growing number of API breaches are occurring. For a nonexhaustive list, see API Security: What You Need to Do to Protect Your APIs.
- Transactional APIs, such as those used in financial services or ad tracking, drive the need for API threat protection because they tie directly into business capabilities.
- Vendors focused on API threat protection are beginning to gain market traction.
- Some established vendors use machine learning to detect potentially harmful API usage patterns and thus protect APIs from threats. They include Akamai, Ping Identity (with the PingIntelligence for APIs security product), as well as bot mitigation vendors such as PerimeterX and Distil Networks (acquired by Imperva). Other vendors, such as CriticalBlue and Data Theorem, have linked mobile security with API threat protection.

Obstacles

- Many organizations lack visibility of their APIs, as many APIs are used as part of web or mobile applications and not published directly. This means that a key requirement of API threat protection is API discovery, since, as every security professional knows, you can't secure what you don't know.
- Organizational ownership of API threat protection is a challenge. Whereas the security team, under a CISO, typically manages a web application firewall (WAF), API gateways are managed by API platform teams. This can lead to API threat protection being neglected.
- To date, most API threat protection has focused on traffic throttling, payload risk analysis (e.g., looking for private data in requests and responses), and/or validation of API traffic against Swagger/OpenAPI Specification (OAS) API definitions. This basic approach is taken by many API mediation products, particularly API gateways, but also web application and API protection (WAAP) products.

User Recommendations

- Discover and categorize your APIs as the first step in API threat protection.
- Implement a layered API security model. At the outer (typically cloud-based) layer, use volumetric distributed denial of service (DDoS) protection and bot detection. Behind this layer, apply API-specific authentication, authorization and traffic management.
- Assess the API protection provided by your current WAF or API gateway using Critical Capabilities for Cloud Web Application and API Protection and Critical Capabilities for Full Life Cycle API Management. If they provide immature or insufficient API protection, investigate API threat protection specialists such as 42Crunch, Imvision, Cequence Security, Neosec and Salt.
- API protection rules should be adaptable, based on the nature of the API itself. Static rate limits or IP allow/block lists are rarely useful in production environments or at scale.

Sample Vendors

42Crunch; Akamai; APIsec; Cequence Security; CriticalBlue; Imperva (CloudVector); Imvision; Neosec; Noname Security; Salt

Gartner Recommended Reading

API Security: What You Need to Do to Protect Your APIs

How to Implement API Discovery and Detection to Improve API Management and Security

Solution Path for Forming an API Security Strategy

Gartner Peer Connect Perspectives: How to Address API Security?

Digital Product Analytics

Analysis By: Melissa Davis, Jason Wong, Aapo Markkanen

Benefit Rating: Moderate

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

Digital product analytics are tools to analyze digital product usage and performance to better understand and improve end-user outcomes. These tools typically apply web and mobile app analytics, along with other feedback mechanisms, to generate specific insights on usage and performance KPIs for product enhancements.

Why This Is Important

Digital product analytics have been around for years and used by digital-native product companies. Tools are used to analyze data from mobile and web apps, but are also increasingly applied to IoT devices and enterprise software. Consumer brands use product analytics to refine the user experience and optimize engagement. As enterprises shift from project delivery to product delivery, mainstream companies such as Capital One, and Adidas are adopting digital product analytics.

Business Impact

End users responsible for digital products can improve adoption, CX, retention and revenue by:

- Tracking app usage to find upsell or cross-sell opportunities and licensing compliance gaps.
- Providing personalized intervention to guide users through real-time in-app messaging, driven by the user's interactions.
- Improving the UX by finding trouble spots in workflows/customer journeys
- Prioritizing enhancements most relevant to users.
- Exposing detailed usage metrics.
- Refining app packaging and pricing.

Drivers

- A high priority has been placed on improving customer experience based on the digital experience.
- There is a trend toward data-driven decision making for product investment prioritization.
- Purpose-built tools for product managers and product teams are increasingly available.
- The COVID-19 crisis has elevated the need for optimized digital user experience as organizations respond to changing priorities from physical to digital experiences.

Obstacles

End users responsible for digital products should take actions to minimize obstacles such as:

- Usability and technical depth of digital product analytics tools require careful balancing as the addressable user base continues to expand from data scientists and analysts to less-technical users. Many of the available tools still require a relatively steep learning curve from business users.
- Data privacy considerations restrict real-life application of product analytics, despite anonymization and other privacy-enhancing features. In particular, products with small user bases, which are common in B2B applications, can be complicated to analyze in that regard.

User Recommendations

End users responsible for digital products should:

- Architect any new digital products for use of digital product analytics and study the options for optimizing existing products for the same.
- Think about employee-facing applications (particularly the custom-developed ones), not just digital product analytics, as digital products with a life cycle to be maintained and optimized.
- Build a data management and governance framework that helps control analytics orchestration and processes as use of the tools expands beyond analytics experts.

- Leverage the quantitative data from product analytics to support data-driven, data-informed and data-inspired decision making in product management, thus complementing the insights available from qualitative techniques such as interviews and surveys.
- Explore the potential to improve go-to-market elements of the product strategy — for example by refining pricing models and tiers through analysis of customer segments and conversion drivers

Sample Vendors

Amplitude; Countly; FullStory; Gainsight; Heap; Mixpanel; Pendo; Pyze; Quantum Metric

Gartner Recommended Reading

Market Guide for Web, Product and Digital Experience Analytics

Competitive Landscape: Product Management Tools

Worlds Collide as Augmented Analytics Draws Analytics, BI and Data Science Together

Product Manager Insight: Changes in Product Data and Insights Through 2025

Sliding into the Trough

Application Security Orchestration and Correlation

Analysis By: Dale Gardner

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

Application security orchestration and correlation (ASOC) tools ease software vulnerability testing and remediation by automating workflows and processing findings. They automate security testing within and across development life cycle while ingesting data from multiple sources. ASOC tools correlate and analyze findings to centralize efforts for easier interpretation, triage, and remediation. They act as a management and orchestration layer between application development and security testing.

Why This Is Important

ASOC products support broad integration and interoperability with commercial application security testing products, enabling greater control over and visibility into testing. Orchestration capabilities allow solutions to interact with continuous integration/continuous delivery (CI/CD) toolchains to specify testing, and control the release of a given build based on results. Some tools include prioritization functions, enabling focus on the most critical vulnerabilities.

Business Impact

- Coordinating security testing across multiple development projects is a complex task, requiring significant initial and ongoing effort. ASOC tools aid teams by interacting with development tools and coordinating tests.
- By managing test data, ASOC tools enable increased effectiveness in prioritizing resources in resolving the most critical vulnerabilities.
- ASOC can be an enabler in implementing DevSecOps, and they promise substantial benefits via more consistent testing and smoother operations.

Drivers

- Gartner clients struggle with prioritizing vulnerability remediation and mitigation efforts during and after development, given the growing volume of information provided by application security testing tools. ASOC tools address this challenge by ingesting information from multiple testing sources, correlating results, and increasingly aiding in the automation of prioritization and triage tasks. This helps to identify those vulnerabilities posing the greatest risk to an application, enabling development and security teams to streamline remediation efforts.
- Developers and operations teams also encounter difficulty in reporting the risk posture of applications, absent meaningful business metrics and threat intelligence. ASOC products can assist in translating raw vulnerability data into a form more relevant to executives and application owners (e.g., based on an application or system centric view).
- The prioritized flow of application security testing data can also help spur developer acceptance of application security testing programs and results. With the requisite prioritization capabilities ASOC tools can ensure remediation efforts are aimed at those areas which will deliver the greatest return on effort.
- The tools have the potential to benefit organizations employing all types of development methodologies. However, they are particularly beneficial to organizations seeking to incorporate more robust security practices into rapid application development and release practices as part of DevOps and a structured CI/CD build pipeline.

Obstacles

- A barrier to adoption of ASOC tools remains a lack of awareness of their existence, although interest is increasing rapidly. In many cases, organizations focus efforts on more traditional approaches, which scale poorly in contrast to ASOC.
- Vendors tend to emphasize integration with either development or operations scanning tools. Individually, this is useful — particularly for practitioners in a particular group. However, it presents a barrier to delivering a “full stack” view of an application’s security risks, spanning both developed code and infrastructure components. Progress to more integrated products is evident.

- Effective automation of security testing presumes an organization understands the overall risk posture of an application, the types of testing needed, and how best — or whether — to respond to findings. As most organizations focus initial efforts on testing itself, the requisite threat modeling may not have been performed. This complicates efforts to articulate the underlying policies on which prioritization and triage efforts rely.

User Recommendations

Security and risk management leaders should leverage ASOC tools based on their ability to perform two functions:

- Act as enablers in application security, particularly for organizations pursuing DevSecOps. The automation of security testing within a CI/CD pipeline requires integration of disparate native capabilities, via bidirectional APIs or command line scripting, across multiple toolsets. With high volumes of development and multiple toolsets, this quickly becomes a resource-intensive effort. ASOC tooling alleviates the integration and management burden will be a significant benefit.
- The analysis and correlation of testing results from multiple tools across different development projects is technically challenging and time-consuming. Correlation and analysis capabilities, combined with integration with defect tracking systems, speed the process while ensuring developers are presented with the most significant vulnerabilities for remediation.

Sample Vendors

Brinqa; Code Dx; Kenna Security; RiskSense; Sevren; Vulcan Cyber; we45; ZeroNorth

Gartner Recommended Reading

Magic Quadrant for Application Security Testing

Critical Capabilities for Application Security Testing

How to Deploy and Perform Application Security Testing

Structuring Application Security Tools and Practices for DevOps and DevSecOps

Privacy by Design

Analysis By: Bart Willemsen, Bernard Woo

Benefit Rating: Moderate

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

Privacy by Design (PbD) is a set of privacy principles that underpin many modern privacy regulatory requirements. PbD is about protecting privacy proactively by embedding it often and early in technology (e.g., application or customer interaction design), as well as into procedures and processes (e.g., through privacy impact assessments). No global finite principle list exists, yet PbD as best practice is globally applicable to the basis of any privacy program.

Why This Is Important

Privacy is one of the core tenants for organizations seeking to earn trust with their customers and drive increased revenue opportunities. In addition, the number of new or significantly revamped regulations continue to increase. Organizations can expect to gain efficiencies in operations by adopting PbD and embedding privacy considerations throughout their processing activities.

Business Impact

Privacy must be built in from the beginning, as they help enhance consumer trust, to prevent violations (such as costly data breaches) before they occur, and reduce their damage if they do (such as fines or brand damage). All technology design must account for protection of any personal data at the core to mitigate privacy risk, which is at unprecedented heights with current data volumes processed.

Drivers

- Systems should be designed so that the collection of privacy-sensitive data is transparent to the data subject. Some technology-focused ideas for implementing PbD are: reduction in amounts of personal data and retention (data minimization); working on the original data (rather than copies), and applying pseudonymization where possible, alongside adequate authorization and access controls.
- SRM leaders should continuously evaluate the risks of reidentification and traceability, and include data location in their considerations for clarity on regulatory impact. Moreover, implementing PbD can drive effect on other changes too, such as designating a privacy officer with reach, procurement activities for new IT services or frequently conducting privacy impact assessments.
- PbD and one of its subcomponents, privacy engineering, enable an approach to business process and technology architecture that combines various methodologies in design, deployment and governance. Properly implemented, it yields an end result with both easily accessible functionality to fulfill the Organisation for Economic Co-operation and Development's (OECD's) eight privacy principles, and mitigation against the impact of a breach of personal data by reimagining defense in depth from a privacy-centric vantage.
- The process involves ongoing recalculation and rebalancing of the risk to the individual data owner while preserving optimum utility for personal data processing use cases. As a result, organizations can rely on the right data being available at the right time with maximized information retention and trust in a compliant operation. They will also find benefit in data footprint and accompanying breach exposure risk reduction, consistent delivery to subjects upon a privacy promise, and the collateral enhanced customer trust and engagement levels.

Obstacles

- Adoption and widespread recognition of PbD has been hampered by a lack of industry recognized list of principles and regulatory support. The IPC of Ontario did describe seven key elements: proactivity, privacy by default, privacy embedded into design, full functionality, end-to-end security, visibility and transparency, and user centricity. In the U.S., a report by the Federal Trade Commission (FTC) of 2012 is the most visible early support for the PbD principle.
- The GDPR does require "data protection by design and by default," implying a PbD approach to all activities — a concept that slowly finds more references in other jurisdictions. Though precedent shaping rulings have long been expected, few were presented. Vendors have added statements like "product X was designed with PbD in mind," sometimes with little reference material to support the claim. Only as privacy becomes a more organic part of the development process, the need for PbD increases, as does the benefit rating.

User Recommendations

- Tackle privacy by design in manageable steps; a wholesale shift will be too much to handle. Privacy by design is a cultural change about the processing of personal data. This pertains both to existing operations and to innovations.
- Adjust the existing operations through business process reengineering. Especially in innovative developments and new processes, the change begins by asking questions such as: Can we achieve the purpose set out by using less personal data? Can we end the personal data life cycle earlier? Can we provide the same functionality or customer experience without using the identifiable data? Does the customer understand what we are processing about them and why? Can we adequately protect what we process?

Gartner Recommended Reading

Use a Privacy Impact Assessment to Ensure Baseline Privacy Criteria

Preserve Privacy When Initiating Your IoT Strategy

What You Need to Know About Privacy in LATAM

Use Privacy to Build Trust and Personalize Customer Experiences

Application Monitoring and Protection

Analysis By: Neil MacDonald

Benefit Rating: Moderate

Market Penetration: 1% to 5% of target audience

Maturity: Emerging

Definition:

Application monitoring and protection is the convergence of security and operational application monitoring to consolidate approaches and simplify instrumentation. It may alert application owners to anomalous application behaviors indicative of a potential service failure caused by hardware or software issues or malicious actors. It could also take protective actions such as moving the workload, spinning up a new image, throttling a request and blocking a potential attack.

Why This Is Important

Over the next decade, operations monitoring, security monitoring and protection use cases will converge for cloud-native applications, driven by DevSecOps-style collaboration and integration. In most mature development organizations, the DevOps product teams or site reliability engineering teams will be responsible for service continuity regardless of whether the issue is a software bug, hardware failure or an attacker. Converged security and application performance monitoring will enable this.

Business Impact

From the business unit and end user perspectives, IT service delivery failures due to hardware failure or security attack are not separate problems. By combining monitoring and protection efforts, IT organizations should improve application stability and performance as well as service quality. They should also avoid suboptimal results caused by a lack of coordination between potential performance and security processes, and reduce the total number of vendors and the licensing and support costs.

Drivers

- Increasingly, DevOps product owners are taking primary responsibility for their products' service levels, including availability and security.
- The adoption of managed container and serverless code services is increasing, and information security has to deal with reduced instrumentation surfaces.
- Using one vendor for security monitoring and another vendor for operational monitoring duplicates efforts, increases complexity, and potentially inhibits the performance and stability of the application they were intended to protect.
- Multiple monitoring vendors are pursuing this strategy of combining application performance and security-monitoring capabilities for cloud-native applications and are bringing competitive offerings to market.
- Container- and Kubernetes-based applications offer new approaches for instrumentation, including privileged containers, daemon sets and sidecars.
- Initiatives such as OpenTelemetry may consolidate monitoring approaches.

Obstacles

- The need for this type of monitoring is driven by two or three separate teams — security, application operations and infrastructure operations — which may have different vendor preferences and requirements as well as budgets.
- There is reluctance to adopt any security monitoring by teams that are incorporating APM solely for production visibility because of fears of performance and instability.
- Operational-monitoring vendors are immature in security capabilities, and security-monitoring vendors are immature in operational monitoring.
- Many organizations don't have the skills on a DevSecOps team to handle frontline security-monitoring responsibility.
- The proliferation of languages and frameworks has made it difficult to pursue a single monitoring strategy.
- Combining security operations and application monitoring potentially violates the principle of separation of duties.

User Recommendations

- Design a process for continuous IT service risk and operational monitoring by converging service security and operational monitoring responsibilities and giving this to the DevSecOps product teams.
- Initiate projects that assess the opportunity for application monitoring and protection and ASM tool consolidation by establishing a small working group comprising key IT operations and security personnel.
- Evaluate the pros and cons of alternative application security protection strategies — for example, by deploying application monitoring and protection in the network or in the cloud using APIs, or by instrumenting the application platform using agents or containers. Don't assume that application instrumentation is the best approach.
- Require application monitoring and protection vendors to provide ASM insight (ideally, also providing attack blocking and prevention capabilities that are part of runtime application self-protection) by working with them to assess opportunities in the organization.

Sample Vendors

Cisco; Datadog; Dynatrace; Elastic; Fastly; GitLab; Rapid7; Splunk; Sysdig

Gartner Recommended Reading

12 Things to Get Right for Successful DevSecOps

Align NetOps and SecOps Tool Objectives With Shared Use Cases

How to Make Cloud More Secure Than Your Own Data Center

Magic Quadrant for Application Performance Monitoring

Application Performance Monitoring and Application Security Monitoring Are Converging

Crowdsourced Software Security Testing Platforms

Analysis By: Dale Gardner

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

Crowdsourced software security testing platforms (CSSTPs) deliver a mix of application security testing services, including the more commonly known bug bounty as well as management of vulnerability disclosure programs. They leverage the talents of hundreds or thousands of private security researchers — the crowd — as the defining element of their services. Buyers can often select from different groups within the crowd, ranging from essentially anyone to a small, highly vetted team.

Why This Is Important

CSSTP vendors offer a variety of advantages over traditional penetration testing firms, including speed, flexibility and specialized skills. Application security testing skills are scarce. CSSTP firms address the gap by drawing on a large crowd of researchers. Vendors can supplement internal application security programs via short- and long-term engagements supporting vulnerability disclosure, bug bounties and related services.

Business Impact

CSSTP offerings can benefit a range of organizations, but are most applicable to firms seeking to supplement internal application security and vulnerability assessment programs on a continuous basis with the skills of a large and diverse group of security researchers. Firms receiving an inordinately large number of vulnerability reports (e.g., software and technology) may benefit from external management of responsible disclosure programs.

Drivers

- The large and diverse nature of the "crowd," individual security researchers participating in the offerings of one or more CSSTP vendors, are the defining characteristic and chief benefit of these firms and the services they provide. In the case of bug bounties, companies benefit from a diverse range of individuals, with unique approaches and skills, performing testing.
- Firms requiring assistance in managing the inbound flow of vulnerability reports — especially technology and software vendors — can benefit by offloading the triage of reports and the associated administrative and communication activities. A significant uptick in interest in such programs has been observed over the last year. This is due to mandates requiring governmental agencies to establish such programs. Also, firms are seeking to expand the scope of their application security programs in the face of an increasing number of breaches and security incidents, along with growing regulatory requirements.
- Because CSSTP providers take on engagements directly with independent third-party testers and oversee various administrative and testing activities (such as vetting reported vulnerabilities), they free up application security teams. This allows those teams to focus on digesting reported vulnerabilities and performing remediation or mitigation rather than dealing with logistical processes.
- CSSTP vendors have continued to expand the scope of their offerings, seeking to monetize the skills of their respective crowds, and establish long-term relationships with clients. Originally focused on conducting only crowdsourced bug bounty programs, companies have expanded the range of services offered to include coordinated vulnerability disclosure, compliance testing and "red team" tests.

Obstacles

- CSSTPs may have a disruptive impact on existing application security testing (AST) activities within an organization if prior attention to integration of results, or standard testing efforts, have not been undertaken. Buyers should not view these programs as a substitute for an internal AST program.
- By leveraging a diverse range of skills that might otherwise be difficult to replicate, CSSTPs can augment the application security expertise of an organization — although buyers should be clear in the type of findings they seek.
- Security researchers continue to voice concerns over issues such as safe harbor (protections for good faith security research activities), buyers choosing NDAs, and the level of remuneration provided for vulnerabilities.
- The perceived risks associated with these programs are limited to few organizations. Those organizations should consider whether their requirements can be better addressed through conventional means, or through an adaptation of standard CSSTP programs.

User Recommendations

- Examine CSSTPs as a viable alternative to traditional penetration testing firms.
- Ensure “rules of engagement” are defined and agreed to before engaging firms and researchers. Since researchers have leeway in how they pursue flaws, buyers must include any limitations of the scope of testing, protections against the disruption of critical systems or the disclosure of sensitive information, and provisions for confidentiality.
- Integrate internal AST programs with external efforts, and eliminate overlap or duplication with existing efforts. Services delivered by CSSTP vendors must be considered an augmentation of existing application security programs, and not as a replacement for them. As such, buyers must include programmatic and process integration between internal AST programs, defect tracking and assessment systems, and external efforts.

Sample Vendors

Bugcrowd; HackerOne; Synack; YesWeHack; Zerocopter

Gartner Recommended Reading

7 Tips to Set Up an Application Security Program Without Breaking the Bank

How to Deploy and Perform Application Security Testing

12 Things to Get Right for Successful DevSecOps

Cloud WAAP

Analysis By: Jeremy D'Hoinne, Adam Hils, John Watts, Rajpreet Kaur, Shilpi Handa

Benefit Rating: High

Market Penetration: 20% to 50% of target audience

Maturity: Early mainstream

Definition:

Cloud web application and API protection (WAAP) products are cloud-delivered multifunction web application security products, integrating at least four core features: web application firewall, DDoS protection, bot management and API protection. WAAP is an evolution of the role of the web application firewall, driven by enterprises' need to better defend against multiple threat vectors while significantly growing their number of publicly exposed web applications and APIs.

Why This Is Important

Organizations moving critical web applications to the public cloud frequently select cloud WAAP solutions from WAF, CDN or infrastructure-as-a-service (IaaS) providers to shield these applications. These solutions can be delivered and managed more flexibly than a traditional virtual appliance due to their ability to be easily deployed. Cloud WAAP roadmaps are dynamic and continue to improve their detection capabilities and the maturity of their management and monitoring consoles.

Business Impact

Public-facing web applications are at high risk of breach. As more and more critical business processes and sensitive data are hosted on these applications, protecting them becomes paramount. Cloud WAAP solutions can also be deployed more easily and managed more efficiently than their WAF appliance counterparts. The value of cloud WAAP goes beyond the bundling approach, with potential benefits from the global visibility provided by cloud deployment.

Drivers

Cloud WAAP simplifies the deployment of runtime application security controls in front of one or many applications. For smaller organizations, compliance requirements represent a primary driver for deploying WAAP in front of public-facing applications. Digital natives, B2C verticals (e.g., retail) and global organizations deploy cloud WAAP to protect assets that they consider critical. In addition to the web application firewall, they value the three other core features of cloud WAAP:

- **Protection against denial of service:** This is important to avoid denial of service attacks, which could impact the business and hurt the brand.
- **Bot mitigation:** This is especially important for the online retail, gaming and travel industries, but all verticals gain from better understanding the impact of automated traffic on their applications.
- **API protection:** Security teams might not have full visibility on it, but a growing share of web application traffic is API-driven. Dedicated controls are required to protect API, and some cloud WAAP providers are focusing on this issue.

Obstacles

The most frequent obstacles facing organizations selecting a cloud WAAP solution are:

- **Privacy-related:** Compliance requirements or fear of legal issues or complex project approval can be issues for organizations. Some organizations don't trust the cloud to decrypt and log the application traffic and host related secrets.
- **Due to existing applications:** Some organizations face issues because of an existing on-premises or hybrid web application deployment. They prioritize unified management and monitoring, or don't want to embark on a new learning curve when they are satisfied with their existing products.
- **Cost:** WAAP as a feature of an ADC might be less costly than a cloud WAAP solution, especially when including the cost of switching to a new architecture.
- **Regional support:** The availability of skilled and experienced support teams might vary for the more recent cloud WAAP products in regions not well supported by vendors, or when no PoPs are located near the organization's origin servers.

User Recommendations

Prospective buyers should:

- Build their application security strategy for the present and the future of their application architecture by applying a cloud-first strategy or “follow the app” principle when deciding between on-premises WAF appliance and cloud WAAP.
- Carefully evaluate the expected benefits and challenges of cloud WAAP. This includes simplicity, data privacy, DDoS protection, bot mitigation and API security, as well as deployment challenges such as certificate management for TLS decryption.
- Continue to improve their stance against bots and other automated attacks by measuring the efficacy of existing controls and adding new techniques when results decline.
- Implement products with automated API discovery and anomaly detection. Note that many WAAP solutions do not yet offer best-of-breed API security capabilities; compare them with offerings from dedicated API security vendors.
- Consider integrations with API gateways or vendors that provide gateways which help with API management.

Sample Vendors

Akamai; Barracuda Networks; Cloudflare; F5; Fastly; Fortinet; Imperva; Radware; ThreatX

Gartner Recommended Reading

Magic Quadrant for Web Application Firewalls

Critical Capabilities for Cloud Web Application and API Protection

Serverless Function Security

Analysis By: Neil MacDonald

Benefit Rating: Moderate

Market Penetration: 1% to 5% of target audience

Maturity: Emerging

Definition:

Serverless function (also referred to as “function PaaS”) security technologies are designed to address the unique security and compliance requirements of serverless function protection. Comprehensive solutions start with proactive vulnerability and configuration scanning in development typically combined with lightweight runtime protection.

Why This Is Important

Serverless functions are available in all hyperscale platforms and are commonly included in cloud-native applications as a simple and scalable way to add logic to a program. Like the rapid adoption of containers, serverless computing appeals to developers, enabling them to focus on writing code without having to worry about all the necessary layers below the code. These small workload units need to be secure, compliant and protected like any other workload.

Business Impact

By ensuring the security and compliance of the serverless functions they create, information security organizations can enable the adoption of these technologies (which is being driven by developers) with acceptable risk and without slowing down development. Longer term, serverless should dramatically improve overall enterprise security profiles as it moves the responsibility for patching of the OS and the application platform — commonly targeted by hackers — to the cloud providers.

Drivers

- Driven by developers, adoption of serverless functions is increasing across all IaaS providers.
- The greatest risk comes from the use of vulnerable code and misconfiguration of the serverless function environment, driving the need for security capabilities such as vulnerability scanning, API security, and correct and compliant serverless PaaS configuration.
- New types of attacks will emerge against serverless PaaS, requiring new protection approaches and techniques.

Obstacles

- In most cases, information security is blind to the use of serverless functions and unaware of the risks they pose.
- Developers will push back on security tooling if it slows them down unnecessarily.
- At runtime, since serverless functions live for a matter of seconds or minutes, the need for additional runtime security other than monitoring is minimal.
- Serverless security tools are still maturing and standards for secure deployment across multiple platforms are yet to be defined.
- At runtime, very few options are available short of injecting or wrapping serverless functions with runtime protection code.

User Recommendations

- Engage with cloud-native development teams now to understand the time frame for use of serverless. Run a discovery project to see if serverless code is in use that you aren't aware of.
- Scan for vulnerabilities and misconfiguration automatically during development as you would with containers.
- Require your cloud security posture management (CSPM) tool to provide risk visibility and configuration/permissions management of the entire IaaS configuration, including serverless.
- Require IaaS vendors to provide granular access controls on all serverless and PaaS capabilities.
- Adopt a least-privilege security posture, including serverless function permissions and network connectivity.
- Require an API gateway or event broker for invocation, providing a visibility and control point.
- Evaluate carefully the need to introduce third-party runtime serverless security controls.
- Require your CWPP vendor to offer serverless function security and compliance capabilities now or as a roadmap item.

Sample Vendors

Amazon Web Services; Aqua Security; Check Point Software Technologies; McAfee; Palo Alto Networks; Rapid7; Snyk; Trend Micro

Gartner Recommended Reading

Market Guide for Cloud Workload Protection Platforms

5 Things You Must Absolutely Get Right for Secure IaaS and PaaS

How to Make Cloud More Secure Than Your Own Data Center

Service Mesh

Analysis By: Anne Thomas

Benefit Rating: Moderate

Market Penetration: 1% to 5% of target audience

Maturity: Adolescent

Definition:

A service mesh is a distributed computing middleware that optimizes communications between application services within managed container systems. It provides lightweight mediation for service-to-service communications, and supports functions such as authentication, authorization, encryption, service discovery, request routing, load balancing, self-healing recovery and service instrumentation.

Why This Is Important

A service mesh is lightweight middleware for managing and monitoring service-to-service (east-west) communications, especially among microservices running in ephemeral managed container systems, such as Kubernetes. It provides visibility into service interactions, enabling proactive operations and faster diagnostics. It automates complex communication concerns, thereby improving developer productivity and ensuring that certain standards and policies are enforced consistently across applications.

Business Impact

- A service mesh helps ensure resilient and secure request-response communication between services deployed in Kubernetes and other managed container systems.
- Service mesh middleware is one of many management technologies that provide software infrastructure for distributed applications deployed in managed container systems.
- This type of middleware, along with other management and security middleware, helps provide a stable environment that supports “Day 2” operations of containerized workloads.

Drivers

- Service mesh adoption is closely aligned with microservices architectures and managed container systems like Kubernetes. Service mesh supports needed functionality in ephemeral environments, such as service discovery and mutual Transport Layer Security between services.
- As microservice deployments scale and grow more complex, DevOps teams need better ways to track operations, anticipate problems and trace errors. Service mesh automatically instruments the services and feeds logs to visualization dashboards.
- A service mesh implements the various communication stability patterns (including retries, circuit breakers and bulkheads) that enable applications to be more self-healing.
- Many managed container systems now include a service mesh, inspiring DevOps teams to use it. The hyperscale cloud vendors provide a service mesh that is also integrated with their other cloud-native services.
- Independent vendors, such as Buoyant, HashiCorp and Kong provide service meshes that support multiple environments.

Obstacles

- Service mesh technology is immature and complex, and most development teams don't need it. It can be useful when deploying microservices in Kubernetes, but it's never required.
- Users are confused by the overlap in functionality among service meshes, ingress controllers, API gateways and other API proxies. Management and interoperability among these technologies hasn't yet been addressed by the vendor community.
- Many people associate service mesh exclusively with Istio, even though it isn't the most mature product in the market and has a reputation for complexity.
- Independent service mesh solutions face challenges from the availability of platform-integrated service meshes from the major cloud and platform providers.

User Recommendations

- Delay adoption of service mesh until your teams start building applications that will get value from a mesh, such as applications deployed in managed container systems with a large number of service-to-service (east-west) interactions.
- Favor the service meshes that come integrated with your managed container system unless you have a requirement to support a federated model.
- Reduce cultural issues and turf wars by assigning service mesh ownership to a cross-functional PlatformOps team that solicits input and collaborates with networking, security and development teams.
- Accelerate knowledge transfer and consistent application of security policies by collaborating with I&O and security teams that manage existing API gateways and application delivery controllers.

Sample Vendors

Amazon Web Services; Buoyant; Decipher Technology Studios; Envoy; F5; Google; HashiCorp; Istio; Kong; Microsoft; Red Hat; Solo.io; Tetrate; VMware

Gartner Recommended Reading

How a Service Mesh Fits Into Your API Mediation Strategy

Assessing Service Mesh for Use in Microservices Architectures

Application Shielding

Analysis By: Dionisio Zumerle

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

Application shielding refers to protection capabilities that are implemented directly within the application, rather than on the network or the operating system, to prevent and detect attacks such as tampering and reverse engineering. Application shielding can be used for any type of application, but we currently see a particular focus on mobile apps.

Why This Is Important

Newer application architectures tend to have more software logic exposed on the front end, be it a mobile app on a smartphone, code on a connected Internet of Things (IoT) device or simply a single-page web app. These architectures create security gaps that, unless shielded, can lead to breaches and security incidents.

As more and more applications emerge or migrate to newer architectures, application shielding increases in importance and relevance for all types of organizations.

Business Impact

Application shielding protects enterprise software and applications from cloning, fraud, IP theft and other forms of abuse. In certain industries, such as financial services and online retail, it can also be used to improve user experience. By hardening the application, an online retailer can minimize the restrictions and additional forms of verification (such as a step-up authentication) requests made to its customers.

Drivers

Application shielding adoption is driven by the following trends:

- Application shielding currently provides two families of protections: hardening and anti-tampering. Hardening hinders the attacker from stealing information (such as intellectual property [IP] or credentials) or cloning the application, by making reverse-engineering harder. Hardening includes techniques such as code obfuscation and white-box cryptography. Anti-tampering performs reconnaissance of the environment the application runs in, to identify potential risks. Anti-tampering techniques include emulation and debugging detection, for example.
- Application shielding is a good protection option for applications that run on untrusted environments, and therefore adoption for consumer-facing mobile apps is growing in particular, especially in the financial services, online retail, healthcare, insurance and automotive industry verticals.
- Recently, buyers of application shielding have been inquiring for products that could provide all-round security functionality. In addition to hardening and anti-tampering, buyers increasingly look for anti-malware capabilities and even functionality that goes beyond shielding, such as multifactor authentication, to minimize the number of security products to integrate in their application.
- Adoption is growing as enterprises and independent software vendors (ISVs) are becoming more aware of the availability of these solutions.

Obstacles

- Hardening techniques are mature, deriving from protection of set-top boxes and digital media. However, they have to be adapted to newer devices and operating systems, such as the mobile and IoT devices, which decreases maturity.
- Protection techniques must keep pace with new and advanced attacks and, therefore, are in constant evolution themselves.
- Application shielding is based on complex and research intensive technology that many seasoned security and development practitioners are not particularly familiar with. Even if the risks may be clear, the techniques and efficacy to counter these risks may not be immediately apparent to end users. Perhaps it is for this reason that application shielding products are perceived as costly from these stakeholders.

User Recommendations

- Identify your apps that contain high value (payment data or IP) and run in untrusted environments (for example, customer devices). Start from mobile devices, but also consider IoT and web applications with software logic on the client side.
- Examine financial services, online retail, e-commerce, insurance and healthcare providers as examples of organizations that should adopt application shielding.
- Determine whether you can implement application shielding directly in the source code, where functionality is extensive, or opt for postcoding on the compiled binary when you do not have access to the source code or do not want to impact the development life cycle.

Sample Vendors

Appdome; Build38; Digital.ai; Guardsquare; Inside Secure; KOBIL Systems; OneSpan; Promon; Verimatrix

Gartner Recommended Reading

Teach Your Applications the Art of Self-Defense

Market Guide for In-App Protection

Protecting Web Applications and APIs From Exploits and Abuse

Building Security Into Mobile Apps Using Checklists, SDKs, App Wrapping and App Hardening

Survey Analysis: The Mobile App Development Trends That Will Impact Your Enterprise in 2017

Bot Management

Analysis By: Jeremy D'Hoinne, Ramon Krikken, Akif Khan

Benefit Rating: High

Market Penetration: 20% to 50% of target audience

Maturity: Early mainstream

Definition:

Bot management encompasses a range of techniques used to detect and respond to automated applications and scripts (bots) that interact with web applications and APIs. It aims at blocking unwanted automated activity, while ensuring that legitimate bots, such as search engine crawlers, work as intended by business. Bot management can be delivered as a dedicated solution, or as a capability integrated into other solutions such as cloud web application and API protection (WAAP).

Why This Is Important

Bot traffic continues to rise, along with increasing sophistication of bots. Attackers leverage bots to automate their attacks onto enterprises' online assets, including scraping, scalping and credential stuffing. The rise of "hu-bot," combination of specialized bots with human-operated fraud farm services, requires ever-improving sophistication in detection and response.

Business Impact

Important primarily for large B2C and B2B web applications, bot management is becoming more and more relevant to any public-facing web application where malicious automated traffic can directly impact business outcomes. Since bots are heavily used when testing stolen credentials and other forms of user account abuse, organizations also need to prevent attackers from gaining access to reusable credentials.

Drivers

- Need to prevent large-scale automated attacks from the more basic bots.
- Security and IT leaders' increasing recognition of the fact that bot management crosses multiple use cases and business units, thus making these capabilities more sought-after.
- Targeted organizations' worry about the direct financial impact and potential brand erosion due to the existence of very targeted bots, sometimes augmented by humans to solve the challenges that enterprises use as a way to discern bots from humans.
- Bot management solutions' ability to not only identify bad bots, but also preserve user experience of legitimate application users and authorized bots. Enterprise-approved bots can include search engine crawlers, automated testing and web application monitoring software, robotic process automation (RPA) or other machine-to-machine communication. These include, for example, bots that run in environments like Microsoft Teams, Slack or Stride, and some from bot marketplaces.

Obstacles

- Absence of realization of the scale of the bot problem. Bots represent more than half of the traffic for many web applications, but the fear of blocking a single legitimate customer is often higher than the perception of the bot problem.
- Bundled offerings from WAAP providers also create difficulties when they can justify deployment of bot management only for some applications or to protect only certain key features of an application (such as login pages).
- Cost is another major constraint and a direct consequence of the other two obstacles as many organizations struggle to get funding for a dedicated solution.

User Recommendations

- Begin by inventorying and categorizing web-, mobile- and public-facing API applications at risk for malicious bot activity.
- Identify the business stakeholders next, who need to be involved before selecting or deploying bot mitigation.
- Assess the level of threat to business assets and the impact caused by bots, starting with the most business-critical and most exposed web applications and APIs.
- Evaluate the capabilities of your existing solutions and determine whether their capabilities are sufficient or if you should get ready to purchase a specialized solution. When current capabilities are not adequate, look toward dedicated bot mitigation solutions.
- Select solutions based on their ability to detect malicious bots via a range of techniques, rather than relying primarily on reputation controls to detect “known bad” sources (e.g., IP reputation) or attack techniques (signatures).

Sample Vendors

Arkose Labs; Cequence Security; DataDome; F5; HUMAN; Imperva; Netacea; PerimeterX; Radware

Gartner Recommended Reading

Critical Capabilities for Cloud Web Application and API Protection

Market Guide for Online Fraud Detection

Business-Critical Application Security

Analysis By: Mark Horvath, Neil MacDonald

Benefit Rating: Moderate

Market Penetration: 1% to 5% of target audience

Maturity: Emerging

Definition:

Business-critical application security is the set of processes and technologies that focuses on the security, risk and compliance of business-critical applications, including enterprise resource planning, human resources management, marketing and other critical business applications.

Why This Is Important

As business-critical applications are opened up to partners, exposed on the public internet and targeted by attackers, their risk profile is changing. Most business-critical frameworks allow developers to add extensions and workflows, expanding the attack surface by changing the application's security context. In many cases, IT departments have limited visibility and control for these systems, and security issues such as patching and configuration management are handled by the business unit.

Business Impact

Line-of-business owners for business-critical applications are increasingly worried about attackers looking to disrupt the business for profit (ransomware) or to steal secrets, data or intellectual property. Securing core ERP systems by protecting the application stack and other critical enterprise systems preserves not only data and IP but also the reputation of the organization as a reliable business partner.

Drivers

- Remote work and outsourced development are opening business- and mission-critical apps to attack, resulting in downtime, data loss and data theft.
- Misconfiguration and administration mistakes also expose business-critical applications to downtime.
- Governments worldwide are becoming aware of the nation-state-level interest in attacking the sensitive IP held by companies in their critical business systems, and government entities worldwide are encouraging commercial enterprises to secure their software supply chains.
- Business-critical systems needed to keep the commercial enterprise going are often deeply intertwined with mission-critical systems that are designed to accomplish the purpose of the business, resulting in an increased attack surface for both.

Obstacles

- Information security still has no visibility or control of certain business-critical systems, and security tools applied to products don't often cover these systems due to technical and organizational constraints.
- Patching critical business systems is haphazard and often outside the control of IT and information security.
- Downtime for patching is avoided in favor of high application availability.
- The lack of security expertise in specific business application frameworks.
- The high cost of some solutions may slow adoption.
- Some offerings only focus on ERP systems.

User Recommendations

- Enforce strong network segmentation and restricted access (ideally zero trust network access) for users and developers.
- Review separation of duties to meet requirements of the Sarbanes-Oxley Act and similar regulations.
- Secure custom code by scanning extensions that your enterprise has added to the business-critical application.
- Conduct periodic vulnerability assessments of missing patches and misconfiguration of the application software and platform.
- Monitor and ensure that patching is taking place as expected, especially for any systems with public internet exposure.
- Intrusion monitoring of the business-critical application OS and transaction logs for indications of exfiltration attack, compromise or collusion.
- Intrusion prevention enabling proactive blocking of attack vectors, for example, preventing configuration changes that would result in an exposure.
- Protection from theft of proprietary data and sensitive information.

Sample Vendors

akquinet; ERPScan; Idera (Kiuwan); Indra (Minsait); Onapsis; SAP; SecureLink

Gartner Recommended Reading

Magic Quadrant for Application Security Testing

Critical Capabilities for Application Security Testing

Climbing the Slope

Mobile Application Security Testing

Analysis By: Dionisio Zumerle

Benefit Rating: Moderate

Market Penetration: 20% to 50% of target audience

Maturity: Early mainstream

Definition:

Mobile application security testing (AST) identifies and helps remediate vulnerabilities within mobile apps for iOS and Android devices. Mobile AST analyzes source, byte or binary code, and observes the behavior of mobile apps to identify coding, design, packaging, deployment and runtime conditions that introduce security vulnerabilities.

Why This Is Important

Mobile applications are central to a company's digital transformation. Ensuring these apps cannot be exploited is essential to enable this transformative process. There is evidence, for example, that more than half of payment fraud attempted is done via mobile devices.

Even though AST techniques remain the same, technologically AST needs to adapt to the mobile device environment and the typically more agile mobile development processes.

Business Impact

- Depending on the organizational structure, mobile AST is meant to be used by security and application development teams, and sometimes directly by the line of business departments.
- Even though every organization should perform AST, regulated and other high-security industries such as financial services, healthcare and online retail are adopting mobile AST.

Drivers

- Mobile AST uses techniques similar to traditional AST, which are adapted for mobile environments and a particular focus is added on vulnerabilities that affect mobile apps, such as unsecured storage, hardcoded credentials and others.

- Companies that use AST for their traditional applications express the need for something that identifies vulnerabilities quickly, perhaps not with the same level of detail as a traditional AST tool and ideally low in cost than a comprehensive AST suite.
- While based on traditional AST, mobile AST adds attention to identifying specific mobile app exposures, such as hard-coded credentials and exposure to man-in-the-middle attacks, as well as excessive device permissions and malicious code. Operations support system (OSS) components and software development kits (SDKs) are frequently used with mobile applications, which makes the ability to test these third-party code essential for mobile AST.

Obstacles

- The bases of the technologies used in mobile AST are static AST (SAST), dynamic AST (DAST), software composition analysis (SCA) and interactive AST (IAST), which have been used for years to test web-based applications and are therefore mature. When applied to mobile apps, testing has to adapt to the peculiarities of mobile and enable effective testing of all client- and server-side codes. Mobile platforms are still evolving, albeit slower than in the past, and therefore mobile AST has not yet reached full maturity.
- Many organizations have less advanced application security programs and are not yet placing attention to testing mobile app code. In many cases, the main source of risk is in the back end, making mobile app code less of a critical priority to include in the application security program.

User Recommendations

- Perform mobile AST on your mobile apps, whether they are workforce or consumer-facing, starting from the most critical ones — applications that run on untrusted environments, with software logic on the clients' side and applications with transactional or IP value.
- Leverage traditional AST vendors that provide mobile AST as a part of a larger suite, in enterprises that require thorough mobile AST and have broader deals in place with AST vendors.
- Look into dedicated mobile AST vendor offerings if you do not have an incumbent AST solution and need specialized features and dedicated support.

- Ensure that organizations that work with third-party developers alternatively require their independent software vendor (ISV) to perform mobile AST on the apps, and provide reports with findings and implemented corrective actions.

Sample Vendors

Appknox; App-Ray; Data Theorem; HCL Technologies; iJiami; ImmuniWeb; Kryptowire; Lookout; NowSecure; Zimperium

Gartner Recommended Reading

Critical Capabilities for Application Security Testing

Container and Kubernetes Security

Analysis By: Neil MacDonald, Tom Croll

Benefit Rating: Moderate

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

Container security is the application of security processes, testing and controls to container-based environments, ideally with support for Kubernetes. Full life cycle container security starts in development by assessing the risk/trust of the container's contents, secrets management and Kubernetes configuration and should extend into production with runtime container segmentation, threat protection and access control.

Why This Is Important

Adoption of container-based applications is mainstream, driven by developers in the name of speed and agility to streamline development in the CI/CD pipeline. Rapid adoption of orchestration platforms, such as Kubernetes, has left traditional vendors and some security teams without appropriate tools to ensure secure application deployment. Container security requires a life cycle approach, starting with scanning of containers in development and protection of containers at runtime.

Business Impact

Containers are not inherently unsecure but are being deployed unsecurely with known vulnerabilities and configuration issues. Without proper controls, developers can introduce vulnerabilities into development and, subsequently, production environments, exposing organizations to avoidable risk. Further, security has been slow to embrace secure container development practices and tools, leaving organizations unaware of potential risks and unprepared to respond to attacks.

Drivers

- Container and Kubernetes adoption is driven by developers and by a need for agility in service development and deployment.
- Security and risk management leaders must address container security issues around vulnerabilities, visibility, compromise and compliance.
- Multiple point solution vendors that have evolved (with some acquired) over the past several years can integrate transparently into the CI/CD pipeline and DevOps processes and are able to proactively scan containers for security and compliance issues.
- Traditional workload protection vendors such as McAfee, CrowdStrike, SentinelOne and Trend Micro have now added support for containers and Kubernetes integration.
- DevOps-style development combined with containers and Kubernetes-based microservices applications deployed onto programmatic cloud infrastructure are a natural fit.

Obstacles

- Container security must start in development, yet many security vendors and enterprises treat container security as a runtime only problem.
- Application security blurs with infrastructure security, creating overlap in vendors, offerings and responsibilities.
- Simply identifying vulnerabilities doesn't provide enough context to know if the code is actually used in production or is reachable from the outside.
- Some container orchestration platforms other than Kubernetes are not supported by all vendors, leaving alternative environments, such as Apache Mesos, exposed.
- Developers will push back on security tooling that slows them down, doesn't integrate with their CI/CD pipeline and wastes their time with false positives or low risk issues.

User Recommendations

- Use a hardened, patched OS for the container host OS.
- Scan containers in development for configuration and vulnerability issues of all code — custom, OS libraries and third-party software — before production.
- Continuously assess the container orchestration environment's (typically Kubernetes) security posture for patch levels and correct and compliant configuration in development and in production using automated tools.
- Pressure existing workload protection vendors to provide complete solutions for container security that address end-to-end container security pipelines.
- Examine the processes expected to run in containers along with their behaviors, and use this information to replace signature-based deny-listing with allow-listing-based lockdown.
- Require container security solutions to explicitly support Kubernetes, including IaaS-based managed Kubernetes services.
- Design single purpose containers and design clear tagging mechanisms to track data sensitivity.

Sample Vendors

Aqua Security; IBM; NeuVector; McAfee; Palo Alto Networks; Rapid7; Red Hat; SafeDog (China only); Snyk; Styra; Sysdig; Trend Micro

Gartner Recommended Reading

Market Guide for Cloud Workload Protection Platforms

How to Make Cloud More Secure Than Your Own Data Center

Guidance Framework for Securing Kubernetes

Containers: 11 Threats and How to Control Them

EAM

Analysis By: Felix Gaehtgens

Benefit Rating: Moderate

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

Externalized authorization management (EAM) provides runtime controls — including policy management, policy enforcement, and decision modeling — for fine-grained authorization to infrastructure, applications, services, transactions and data. EAM is also sometimes referred to as “dynamic authorization.” EAM solutions usually offer a centralized policy server and can implement multiple authorization methodologies, including attribute-based/role-based access control (ABAC/RBAC).

Why This Is Important

EAM delivers security and lower maintenance due to a tighter, integrated and standardized common policy mechanism (what is allowed, and under which circumstances, for example). EAM defines and orchestrates authorization policies across different applications, systems and services (instead of having to define policies separately for every one of them). It can add fine-grained runtime authorization to DevOps tools, cloud resources, microservices and applications.

Business Impact

EAM is a long-term investment in an architecture for granular control of application and data access:

- Its use in applications and data systems can significantly change the consistency, flexibility and granularity of authorization.
- It results in better access control, audit and compliance results over a broader set of infrastructure, applications, services, transactions and data.
- It helps developers save time by focusing on delivering functionality while outsourcing access decisions to the EAM tool.

Drivers

EAM was a niche technology for the last decade, where it was used predominantly in large and complex environments that feature significant custom development. Its use case focused on providing a runtime authorization layer across multiple components or applications. Attribute-based access control (ABAC) was another driver for its adoption, with its capability to use dynamic runtime attributes and additional context rather than the more limited and static role-based access control (RBAC).

Typical examples of strategic deployments are in healthcare, airline ticketing systems, banking systems, or supporting trading or supply chain systems that consist of a large number of custom-developed components into which EAM can be integrated.

EAM was also used by organizations to provide more fine-grained access control consistently over distinct applications such as Microsoft Dynamics 365, SAP, or SharePoint that supported integration with EAM tools through proprietary mechanisms.

In recent years, EAM has become more popular, and importantly, more accessible because of several drivers:

- Compliance continues to be a key driver for EAM — an example being privacy regulation such as CCPR, GDPR, and LGPD, which require organizations to support more-granular data access authorization. Much of this is in combination with customer identity and access management (CIAM) initiatives.

- While the predominant standard for EAM has been the Extensible Access Control Markup Language (XACML), nowadays much traction happens around the Open Policy Agent. It supports out-of-the-box integrations for policy enforcement with many common components in cloud-native and DevOps architectures.

Additionally, recent acquisitions in the space have happened:

- Ping Identity acquired Symphonic Software in late 2020.
- Twilio acquired Ionic Security in 2021.

Obstacles

- Externalized authorization requires thoughtful planning from the design phase. Retrofitting dynamic authorization into existing applications is difficult. This is because often, applications spread authorization functions to many places within the code. Many commercial off-the-shelf applications do not offer APIs for authorization to be externalized in a plug-in manner. Instead, integration external authorization can only be loosely coupled: translation of policies into a set of entitlements, and creating tokens with those entitlement combinations for users accessing the applications.
- For developers to fully embrace EAM and be productive, they will require an easy-to-use self-service interface for policy design and testing. EAM has a learning curve and associated complexity.
- EAM requires strong governance to be in place for data and entitlements that are used as input to access decisions. Consequently, data quality has a direct impact on the viability of data-driven ABAC rules.

User Recommendations

- Use EAM to replace custom authorization code for decision and enforcement from homegrown applications with an easier-to-maintain, consistent framework for centralized policy management.
- Socialize your plans and set solid foundations for EAM by offering self-service functions and ensuring buy-in from developers.
- When evaluating EAM solutions, conduct evaluations against two sets of requirements: policy management services and policy runtime services (decision and enforcement).
- When looking to retrofit existing applications with EAM, choose tools that cover immediate needs by integrating with security tokens that deliver entitlement, but also provide standardized authorization methods (typically XACML and/or OPA support) for future developments.
- Mandate robust management of attributes and data that is used in policy evaluation, because EAM requires it.
- Explore integration with API gateways and AM tools to extend the scope of authorization and enhance the overall security posture.

Sample Vendors

Axiomatics; ForgeRock; Jericho Systems; NextLabs; ObjectSecurity; Ping Identity; PlainID; Styra; WSO2

Gartner Recommended Reading

Architecting Modern Policy-Based Runtime Authorization

Guidance for Modernizing Authorization Architecture

Enterprise App Stores

Analysis By: Chris Silva

Benefit Rating: Moderate

Market Penetration: More than 50% of target audience

Maturity: Mature mainstream

Definition:

Enterprise App Stores can be delivered as stand-alone offerings, and as integrated functionality of an endpoint management platform. Unlike commercial app stores, where publishers and ISV list apps for consumers, enterprise app stores are curated, private marketplaces for employee-facing apps.

Why This Is Important

Enterprise app stores distribute native applications developed internally or from a public catalog of apps; web-based apps (as links) and virtual apps and desktops in a single catalog can serve mobile and traditional endpoint devices. Stores present a curated set of apps to help users discover and install to relevant apps without requiring IT intervention.

Business Impact

The capability of enterprise app stores has expanded beyond the mobile apps that often led to the creation of a private storefront. Now able to push apps for PC devices and mobile devices along with virtual, SaaS and web applications, these tools are relevant for all endpoints in a digital workplace — not just mobile devices. These tools are a critical piece of automating end user device deployments, approvals and license management.

Drivers

Application catalogs have grown larger and are increasingly leaned on as a means to streamline user access to applications — whether for their native platform or as a means to direct users to virtual and web apps that extend the utility of mobile devices.

Stand-alone versions of these tools are ideal to service users outside the organization while existing UEM tools, which should already be managing both user- and organization-owned devices, can fill this purpose for employees.

The need to keep employees productive, even when traditional office and IT trappings are absent, has driven interest in using these tools as a central clearinghouse to provide users access to key tools — regardless of the device or platform being used.

Enterprise app stores provide three key benefits:

- **License management:** The use of enterprise app stores for mobile apps provide IT a means to offer paid apps to users without the hassle of expense reports and the cost of losing access to licenses held by individual users that cannot be reassigned. An enterprise app store is required to take advantage of programs, such as Apple's Volume Purchase Program (VPP), which allows centralized buying and redistribution of licenses for paid apps.
- **Consistent delivery and support:** Using a single touchpoint for delivering all apps across end user devices will generate operational complexity without tooling to centralize the various, curated app collections for users and make distributing apps and tracking any app licenses very cumbersome.
- **Improved performance and usage data:** The use of either a stand-alone enterprise app store — or the app store capabilities of a UEM, when used in conjunction with the app store vendor's SDK or app wrapper — can unlock detailed usage metrics and performance telemetry to better assess the utility and impact of applications, and identify apps or app features that are not providing utility to users.

Obstacles

Deploying apps directly to users' devices requires enrolling the device in a management tool with a management profile installed on the device. In many cases, this management relationship is not possible due to:

- Using UEM or MDM tools to push apps to devices that cannot be enrolled due to an existing MDM or UEM enrollment, requiring use of a stand-alone MAM tool with manual user installation or self-guided installation.
- Users' personal devices are increasingly being sanctioned for use at work, though concerns on privacy have changed strategies from defaulting to managing the device, preferring to offer a less intrusive management construct for these devices. This removes the ability to push apps directly to devices.
- Enterprise app stores' impact on the business increases with the size and complexity of an app catalog, but the licenses required and operational — and overhead of establishing a store for few applications can negate any savings that such a tool can bring through automation.

User Recommendations

- Use existing tools like UEM to deliver an app store experience for an audience consisting of internal employees and focused on devices that are managed by the UEM. It should be used for creating and delivering the Enterprise App storefront, curating app collections and managing application licenses.
- Use a stand-alone MAM tool to provide app store functionality independent of device management in cases where another entity (e.g., contractor's primary employer) already manages the device — or when internal policy, privacy legislation and/or user demands require partial management of end-user devices, and lack the ability to automatically push apps to those devices.
- Deliver modern and co-management transitions. Effective ones succeed based on delivering apps at first boot "over the wire" or "over the air," by creating a customized app catalog with rule-sets to autonomously present and configure users' apps — without the need for IT involvement or manual request workflows.

Gartner Recommended Reading

Security Best Practices for Work-From-Home Scenarios

Avoid Mobile Application Security Pitfalls

How to Protect Sensitive Data in Office 365 Using Microsoft Native Controls

Full Life Cycle API Management

Analysis By: Shameen Pillai

Benefit Rating: High

Market Penetration: More than 50% of target audience

Maturity: Early mainstream

Definition:

Full life cycle API management involves the planning, design, implementation, testing, publication, operation, consumption, versioning and retirement of APIs. API management tools enable API ecosystems and publishing APIs that securely operate and collect analytics for monitoring and business value reporting. These capabilities are typically packaged as a combination of developer portal, API gateway, API design, development and testing tools as well as policy management and analytics.

Why This Is Important

APIs are widely used and accepted as the primary choice to connect systems, applications and things to build modern composable software architectures. The use of APIs as digital products monetized directly or indirectly is also on the rise. Advancing digital transformation initiatives across the world have emphasized the need for creation, management, operations and security of APIs and made full life cycle API management an essential foundational capability every organization must have.

Business Impact

Full life cycle API management provides the framework and tools necessary to manage and govern APIs that are foundational elements of multiexperience applications, composable architectures and key enablers of digital transformations. It enables the creation of API products, which may be directly or indirectly monetized, while its security features serve to protect organizations from the business impact of API breaches.

Drivers

- Organizations are facing an explosion of APIs, stemming from the need to connect systems, devices and other businesses. Use of APIs in internal, external, B2B, private and public sharing of data is driving up the need to manage and govern APIs using full life cycle API management.
- APIs that package data, services and insights are increasingly being treated as products that are monetized (directly or indirectly) and enable platform business models. Full life cycle API management provides the tooling to treat APIs as products.
- Digital transformation drives increased use of APIs, which in turn increases the demand for API management.
- APIs provide the foundational elements required for growth acceleration and business resilience.
- Developer mind share for APIs is growing. Newer approaches to event-based APIs, design innovations and modeling approaches such as GraphQL, are driving interest, experimentation and growth in full life cycle API management.
- Cloud adoption and cloud-native architectural approaches to computing (including serverless computing) are increasing the use of APIs in software engineering architectures, especially in the context of microservices, service mesh and serverless.
- Regulated, industry-specific initiatives such as open banking and connected healthcare, along with nonregulated, opportunistic approaches in other industries are increasing the demand for full life cycle API management.

Obstacles

- Lack of commitment to adequate organizational governance processes hinders adoption of full life cycle API management. This can be due to lack of skills or know-how, or due to too much focus on bureaucratic approaches rather than federated and automated governance approaches.
- Lack of strategic focus on business value (quantifiable business growth or operational efficiencies) and too much focus on technical use cases can disengage business users and sponsors. This is particularly apparent in cases where API programs fail to deliver promised return on investment.
- Traditional, single-gateway approaches to API management do not fit well to a modern, distributed application environment.
- Partial or full set of API management capabilities provided by vendors in other markets such as application development, integration platforms, security solutions, B2B offerings, etc., can create confusion and potentially shrink the market opportunities.

User Recommendations

- Use full life cycle API management to power your API strategy that addresses both technical and business requirements for APIs. Select offerings that have the ability to address needs well beyond the first year.
- Treat APIs as products managed by API product managers in a federated API platform team.
- Choose a functionally broad API management solution that supports modern API trends, including microservices, multigateway and multicloud architectures. Ensure that the chosen solution covers the entire API life cycle, not just the runtime or operational aspects.
- Use full life cycle API management to enable governance of all APIs (not just APIs you produce), including third-party (private or public) APIs you consume.
- Question full life cycle API management vendors on their support for automation of API validation and other capabilities, as well as their support for a modern, low-footprint API gateway.

Sample Vendors

Axway; Google; IBM; Kong; Microsoft; MuleSoft; Software AG

Gartner Recommended Reading

Magic Quadrant for Full Life Cycle API Management

The Evolving Role of the API Product Manager in Digital Product Management

How to Use KPIs to Measure the Business Value of APIs

API Security: What You Need to Do to Protect Your APIs

Top 10 Things Software Engineering Leaders Need to Know About APIs

DevSecOps

Analysis By: Neil MacDonald, Mark Horvath

Benefit Rating: Transformational

Market Penetration: 20% to 50% of target audience

Maturity: Early mainstream

Definition:

DevSecOps is the integration and automation of security and compliance testing into agile IT and DevOps development pipelines as seamlessly and transparently as possible, without reducing the agility or speed of developers or requiring them to leave their development toolchain. Ideally, offerings provide security protection at runtime as well.

Why This Is Important

As IT development processes become more agile (including shifts to DevOps operating models), security should be proactively and seamlessly integrated into them — DevSecOps. DevSecOps offers a means of effectively integrating security into the development process, eliminating or reducing friction between security and development, and pragmatically achieving a secure, workable software development life cycle (SDLC).

Business Impact

The goal is to enable development to move faster without compromising on security and compliance. Furthermore, the externalization of security policy enables business units and security organizations, not developers, to define appropriate policies. Policy-driven automation of security infrastructure improves compliance, the quality of security enforcement and developer efficiency, as well as overall IT effectiveness.

Drivers

- Adoption of DevOps and other rapid development practices requires security and compliance testing that can keep up with the pace of development.
- DevSecOps offerings are applied as early as possible in the development process whereas traditional application security testing (AST) tools associated with older development models are applied late in the development cycle, frustrating developers and business stakeholders.
- Testing results need to be integrated into the development process in ways that complement developer's existing workflows, not require them to learn skills seemingly unrelated to their goals.
- The use of open source has greatly increased the risk to the inadvertent use of known vulnerable components and frameworks by developers.

Obstacles

- Developers view security testing tools as slowing them down in getting new releases out.
- Implemented incorrectly, siloed and cumbersome security testing is the antithesis of DevOps.
- Developers don't want to leave their development (continuous integration/continuous delivery [CI/CD]) pipeline to view results of security and compliance testing tools.
- Historically, static application security testing (SAST) and dynamic application security testing (DAST) tools have been plagued with false positives or vague information, frustrating developers.
- The diversity of developer tools used in a modern CI/CD pipeline will complicate seamless integration of DevSecOps offerings.

User Recommendations

- Prepare security and risk management teams for automated integration with DevOps initiatives, and identify the primary skills and technology gaps.
- “Shift left” and make security testing tools and processes available earlier in the development process, ideally as the developers are writing code.
- As zero vulnerability applications aren’t possible, favor automated tools with fast turnaround times with a focus on reducing false positives and allowing developers to concentrate on the most critical vulnerabilities first.
- Prioritize the identification of open-source software (OSS) components and vulnerabilities in development (referred to as software composition analysis).
- Require your security vendors to support out-of-the-box integration with common development toolchain vendors and also support full API enablement of their offerings for automation.
- Require security controls to understand and apply security policies in container- and Kubernetes-based environments.
- Favor offerings that can link scanning in development (including containers) to correct configuration and protection at runtime.

Sample Vendors

Apiiro; Aqua Security; Contrast Security; IBM (Red Hat), NowSecure; Palo Alto Networks; ShiftLeft; Snyk; Sonatype; Veracode

Gartner Recommended Reading

12 Things to Get Right for Successful DevSecOps

Integrating Security Into the DevSecOps Toolchain

Market Guide for Cloud Workload Protection Platforms

Magic Quadrant for Application Security Testing

Critical Capabilities for Application Security Testing

Market Guide for Software Composition Analysis

How to Make Cloud More Secure Than Your Own Data Center

DevSecOps Security Metrics: Use Your Code Repository to Start a Virtuous Cycle

Software Composition Analysis

Analysis By: Dale Gardner

Benefit Rating: High

Market Penetration: 20% to 50% of target audience

Maturity: Early mainstream

Definition:

Software composition analysis (SCA) products are specialized application security testing tools that detect open-source software (OSS) and third-party components known to have security and/or functionality vulnerabilities, and to identify potentially adverse OSS licensing terms. It is an essential element in strategies to ensure an organization's software supply chain includes secure and trusted components and, therefore, aids in secure application development and assembly.

Why This Is Important

SCA's ability to help ensure the software supply chain is current, free of known vulnerabilities, and properly licensed supports the use of OSS in application development. It is an essential element of application security testing, given the ubiquity of OSS in applications and the potential for significant risk.

Business Impact

- SCA is broadly applicable and beneficial to all types of organizations with application development efforts. It should be considered a foundational element of application security testing.
- Its primary users are application development and security teams, tasked with ensuring the integrity of open-source components — and less frequently commercial software — in development.
- License assessments, once the primary use case for SCA, remain an important function for legal and sourcing groups.

Drivers

- SCA has moved from a mature to an early mainstream position in this Hype Cycle. This unusual reversal is a consequence of multiple changes, including a number of significant technological innovations (e.g., greater integration with other types of testing, more sophisticated analysis of the potential impacts of a package upgrade) and its vastly broader applicability to all types of organizations.
- The underlying driver for the growing importance of SCA is the increased use of OSS in application development. The prevalence of OSS, both as individual packages and within container images, has led to SCA becoming a key control on CI/CD pipelines and containerized environments.
- Repeated instances of high-impact vulnerabilities in OSS, rendered pervasive because of the underlying components' broad use across organizations, along with ever-present supply chain attacks, demonstrate the need to better understand the risks posed by OSS and commercial software packages.
- SCA analyzes code in use and reports issues gathered from information collected from sources such as OSS communities, private research and publicly available security vulnerability databases. Information typically includes intellectual property (IP) ownership, known security and functionality vulnerabilities, and references to the most recent versions of components. Some tools supplement this information with guidance on a preferred update version, balancing stability, remediation of flaws and potential adverse impacts on the functionality of existing code, all of which boosts developer and security team productivity and efficiency. In a few cases, tools have begun to incorporate assessments of operational risk, in an effort to help prevent supply chain attacks.
- SCA technology can help ensure developers are meeting compliance requirements and ethical and legal standards for code use, reducing the likelihood of unwanted or unapproved code that creates risks to an organization's intellectual property.

Obstacles

- The quantity of OSS is inestimable. The number of packages, languages and interdependencies creates an intricacy impossible for a single tool to manage. SCA tools specialize in a subset of languages, limiting utility. Development teams with diverse portfolios will turn to multiple tools, increasing costs and complexity.
- SCA is — incorrectly — seen as a solution to software supply chain attacks. While the tools report known vulnerabilities, with few exceptions they don't test code to identify new issues. Vendors will expand the scope of their research to identify signs of package compromise, and expand testing to discover issues proactively.
- Commercial software libraries are often outside the scope of SCA, but pose many of the same risks. This leaves a protection gap.
- Tools are increasingly optimized for application development use cases. But, legal and sourcing teams still rely on them to identify licensing issues. Tools will develop alternative interfaces to support different users.

User Recommendations

- Examine SCA technologies as a key ingredient of every software security program, used in conjunction with other AST tools. AST vendors should provide this capability built-in or via a partnership with a dedicated SCA vendor, and through dedicated commercial and open source tools.
- Analyze SCA warnings on licensing issues to address the legal issues stemming from components' IP ownership or license terms. This should be done by legal experts.
- Use SCA tools on a regular basis to audit repositories that contain software assets to ensure the software developed and/or used by the enterprise meets organizational standards.
- Utilize SCA tools to inspect the components application developers plan to use. SCA tools fit well within DevSecOps workflows, where scanning can be automated as part of the rapid development processes.
- Use SCA tools in conjunction with a formal corporate IP strategy that has established clear responsibility across the company.

Sample Vendors

Checkmarx; GrammaTech; Hdiv Security; JFrog; Snyk; Sonatype; Synopsys; Veracode; WhiteSource

Gartner Recommended Reading

Market Guide for Software Composition Analysis

Critical Capabilities for Application Security Testing

Magic Quadrant for Application Security Testing

How to Manage Open-Source Software Risks Using Software Composition Analysis

Entering the Plateau

Dynamic Data Masking

Analysis By: Dale Gardner

Benefit Rating: Moderate

Market Penetration: 20% to 50% of target audience

Maturity: Mature mainstream

Definition:

Dynamic data masking (DDM) transforms data in real time, on request, to prevent the disclosure of sensitive information contained within underlying data stores. It is a mature technology that has long been used to provide differentiated access to data within business applications. DDM does not protect the original data at the storage level.

Why This Is Important

Adopting DDM helps organizations address evolving compliance and threat landscapes by minimizing the exposure of sensitive data, without extensively modifying existing applications or systems. Tools interactively request and present only the portions of data appropriate for the user, typically by establishing a proxy between requesting applications and databases.

Business Impact

DDM has long been a useful tool, given its ability to interactively redact or transform data presented to individuals within an application. This limits the ability of individuals to access information inappropriately, helps prevent data exfiltration and breaches, and aids in ensuring compliance with ever-growing privacy regulations.

Drivers

- The continuing impact of privacy-focused regulatory requirements, initially led by the General Data Protection Regulation (GDPR) in 2018, then followed by a host of others in the interim, remains the most significant driver for DDM and supporting technologies.

- Insider threats grow in importance, as the number and type of employees or contractors increase to include additional users of production databases (for example, data scientists, programmers, testers and database administrators) or users of analytical and training databases. Organizations are also learning, in light of greater risks, that many users may have access to insufficiently protected production environments.
- DDM is not a useful technique for protection of data at rest. Nevertheless, external threats (such as data exfiltration by various techniques that exploit overexposed sensitive data) or anxiety over movement of data to the cloud spur interest in masking, in conjunction with other privacy-enhancing computation techniques and controls.

Obstacles

- Increasing complexity on multiple fronts slows adoption as both buyers and vendors adapt to new data protection technologies, types and formats of data to be protected, and architectural approaches.
- Traditionally, DDM solutions have intercepted queries and responses at the database layer, with some solutions operating at the application layer. The approach is fit for only structured data and will impact performance to varying degrees.
- Data analytics and machine learning are highly sensitive to input data and require any transformed data to retain its underlying statistical relationships. Typical masking techniques used in DDM tools generally lack the ability to retain those relationships, prompting adoption of alternative privacy-preserving computation techniques.
- The incorporation of native masking capabilities in database tools, the expansion in scope among products from traditional vendors and the emergence of a number of new vendors complicate evaluation and selection.

User Recommendations

- Use DDM to address data masking requirements in production environments, or where organizations cannot make changes to data in the underlying database, and data-at-rest protection is covered by other means.

- Evaluate use of DDM, in combination with attribute-based access control (ABAC), differential privacy, format-preserving encryption (FPE) and tokenization, for the protection of sensitive data in applications and in storage, especially for more sophisticated use cases.
- Assess the suitability of solutions against the specific, targeted portfolio of applications and target use cases as part of the DDM tool selection process. Assess support for databases and other storage technologies currently in use and planned. Consider proofs of concept (POCs) that include performance impact.
- Evaluate DDM in the context of your overall data protection needs. There are synergies among DDM, data activity monitoring (DAM), tokenization and encryption, and some vendors are offering combinations of these capabilities in a single tool.

Sample Vendors

DataSunrise; IBM; Informatica; Mentis Technology; Microsoft; Oracle; PKWARE (Dataguise); Privitar; SAP; SecuPi

Gartner Recommended Reading

Market Guide for Data Masking

Guide to Data Security Concepts

Top Strategic Technology Trends for 2021: Privacy-Enhancing Computation

Preserving Privacy While Using Personal Data for AI Training

IAST

Analysis By: Mark Horvath

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Early mainstream

Definition:

Interactive application security testing (IAST) uses instrumentation to combine the benefits of dynamic application security testing (DAST) and static analysis security testing (SAST). Instrumentation allows DAST-like confirmation of exploit success and SAST-like coverage of the application code. IAST can be either delivered passively, leveraging general application testing, or actively induced via DAST.

Why This Is Important

IAST overcomes problems that occur when SAST or DAST is run alone:

- SAST scanners suffer from a high rate of false positives.
- SAST scanners can't tell developers if a given set of code is reachable by attackers.
- DAST scanners can tell developers that a vulnerability exists, is reachable and can be exploited, but doesn't tell where the code containing the vulnerability is located.

Instrumentation allows DAST-like confirmation of exploit success and SAST-like coverage of the application code.

Business Impact

By combining DAST and SAST techniques in an interactive fashion, the security vulnerabilities identified by IAST contain fewer false positives (or fewer false negatives) and higher speed than SAST and DAST. However, "fewer" doesn't mean "none," so be skeptical of vendor claims. Because of this, IAST works well in agile, continuous delivery and DevSecOps environments, where rapid turnaround is paramount.

Drivers

- The increasing speed of development and SAST's high rate of false positives is pushing organizations to get more efficient with their application security testing. IAST allows organizations to focus on vulnerabilities that are true positives.
- IAST can be run in a "passive mode," observing code execution while other tools run standard QA tests. This is seen as a big win for efficiency, especially in organizations that rely on test-driven development (TDD).
- IAST does a good job visualizing complex data flows and dependencies between applications, services and other components in modern applications.
- AST vendors have seen IAST as either a way to differentiate themselves from other vendors who may only be offering standard SAST/DAST/SCA products.
- Because IAST supports many of the functions of SAST and DAST, some clients are replacing traditional tools (usually DAST) with IAST.

Obstacles

- Organizations that are just beginning to secure SDLC tend to favor tools like SAST and SCA, which not only reduce their immediate risk to OWASP Top 20, but help train their developers around common security issues.
- Building in instrumentation into the applications for IAST to use is not a trivial amount of work, and many organizations have trouble weighing this cost against the reduction in false positives, especially early in their security maturity.
- Traditional AST organizations are using techniques like correlation or human direction to simulate IAST-like functionality without investing in instrumentation.
- IAST can have an adverse effect on application performance and is limited in coverage by what "code paths" are exercised. In the absence of a full QA or TDD suite, coverage gaps could be significant.

User Recommendations

- Use IAST for high-assurance testing by IT and DevOps organizations that develop their own applications. IAST has reached the Plateau of Productivity, with multiple vendors offering it as a stand-alone tool or as part of a broader AST tool suite.
- Reevaluate your DAST and SAST approaches using accuracy gains achieved through IAST.
- Apply lightweight SAST during coding, combined with IAST and DAST during the test and prerelease phases. Enterprises should feel confident about using IAST as a mature AST technology.
- Allow the application to self-test, without requiring security testing experts to run the tests, speeding the development and testing cycle while simultaneously adding some security benefits.

Sample Vendors

Acunetix; Checkmarx; Contrast Security; HCL Software; Hdiv Security; Micro Focus; MoreSec; Synopsys; Veracode

Gartner Recommended Reading

Critical Capabilities for Application Security Testing

Structuring Application Security Tools and Practices for DevOps and DevSecOps

How to Deploy and Perform Application Security Testing

Web Application Firewall Appliance

Analysis By: Shilpi Handa, Jeremy D'Hoinne

Benefit Rating: High

Market Penetration: More than 50% of target audience

Maturity: Early mainstream

Definition:

Web application firewall (WAF) is a technology deployed in-line to protect web applications and APIs. WAF appliances focus primarily on exploits, such as cross-site scripting (XSS) and SQL injection, in commercial applications or custom-developed code. They may include protection from other attacks, such as session manipulation and logic abuse.

Why This Is Important

Organizations deploy WAF appliances to protect their traditional corporate websites, B2C and B2B web applications, as well as a growing number of API-driven applications. Large B2C applications face a higher risk of automated attacks coming from bots. Some organizations opt to host websites powered by third-party content management systems, which frequently include various vulnerabilities that might expose them to data leakage. WAF appliances are also important from a compliance perspective.

Business Impact

WAFs are used to:

- Protect data center servers and hosted applications.
- Prevent attacks that could give access to important data that often lives behind web applications.
- Provide documentation and features to support compliance requirements, such as PCI and GDPR.

Drivers

WAF appliances are quickly reaching the Plateau of Productivity. Cloud-delivered web application and API protection (WAAP) is leading the race for most of the “cloud first” strategy organizations, and COVID-19 has added velocity to its swift adoption. WAF appliances are largely adopted for specialized control and customized policy and rules development. WAF appliances remain the deployment of choice for organizations with existing deployments in the EMEA and Asia/Pacific region and for organizations that are heavily regulated and need data to reside in-house.

Obstacles

WAF appliances will see further competition this year as WAAP service providers have started to strengthen their offerings of API protection and bot mitigation and have increased the number of POPs.

The major obstacles are:

- Deploying and maintaining WAF appliances in-house is resource intensive and some organizations find this challenging
- Most WAF vendors are making R&D investment into their cloud offerings leaving many appliances lagging behind when it comes to new and advanced features

User Recommendations

Enterprises deciding on deployment options should keep in mind that:

- Appliances provide more control, and they are not dependent on the service provider's management and maintenance of technology.
- Appliances give an option to have customized policies and in-depth rules.
- Appliances benefit customers with data privacy concerns with more control over WAF deployment location, log storage, access and retention.
- An appliance's bot mitigation, DDoS and API protection features might not have parity with cloud options. Understanding the difference is important for specific use cases.
- WAFs are increasingly API-driven and DevOps environments favor WAFs that expose APIs to facilitate automated deployment.
- Seeking information on WAF integration with other security controls is important for increased visibility and efficiency. For example integration with security monitoring tools, web access management (WAM), API gateways, bot management, content delivery network, DDoS and online fraud detection.

Sample Vendors

Akamai; Barracuda Networks; Citrix; F5; Fortinet; Imperva; Radware

Gartner Recommended Reading

Magic Quadrant for Web Application Firewalls

Critical Capabilities for Cloud Web Application Firewalls Services

Defining Cloud Web Application and API Protection Services

Cloud Application Discovery

Analysis By: Jay Heiser

Benefit Rating: Moderate

Market Penetration: 20% to 50% of target audience

Maturity: Mature mainstream

Definition:

Cloud application discovery (CAD) is a mechanism that provides information on which public cloud services are being accessed. Discovery reports include application name and type, and usage by individual and department. Information on cloud services in use is vital for a variety of corporate functions, including security, licensing, compliance and I&O.

Why This Is Important

Most organizations have hundreds, even thousands, of SaaS applications in regular use, many of which the IT department is unaware of, and most of which are not IT's direct responsibility. Cloud application discovery mechanisms identify and report on this "unsanctioned IT," a critical first step for any organization which wants all public cloud use to be secure, compliant with regulation and usage agreements.

Business Impact

Organizations that have chosen to explicitly manage use of SaaS and cloud services with CAD-type mechanisms are often able to noticeably reduce SaaS expenditures. Organizations that govern the use of SaaS experience significantly less frustration with service renewal and are less likely to experience downtime or security incidents.

Drivers

- IT professionals increasingly recognize that identification and analysis of unsanctioned services are critical elements of data governance, cybersecurity and IT spend management. This recognition drives uptake of CAD. Regulations, especially the EU GDPR and CCPA, provide additional incentive to determine which SaaS platforms are in use to determine whether that use is consistent with regulatory requirements.
- CAD is not a stand-alone market, but rather a feature of several categories of multifunction management tools. It is a primary capability of cloud access security broker tools and SaaS management platforms, and a secondary capability of software asset management, DNS solutions, firewalls and secure web gateway products.
- Although the level of actual CAD utilization remains surprisingly low, the widespread inclusion of this feature in a variety of product categories means that most organizations can apply it if they wish.

Obstacles

- Surprisingly, CAD is available to most organizations through existing tools, although no offerings include the full set of capabilities needed by every corporate function with a stake in understanding and managing the external applications being used. Dozens of multifunction products offer different forms of CAD functionality, typically as an add-on feature to some other more strategic security, license management or SaaS management function.
- The CAD reporting of these tools is usually aimed at one of those corporate constituencies, and may not be in a form that supports the cloud control needs of other organizational departments. Most organizations have a relatively low concern about shadow IT.
- Without a top-down mandate to control the use of SaaS, security and procurement leaders rightly recognize that the rest of their organization would prefer that they not report on the significant use of SaaS. Consequently, CAD features are often underutilized, or not used at all.

User Recommendations

- Start by reviewing expenditures in corporate accounting records to see which cloud services are being purchased off the IT budget.
- Extend visibility across free services and services paid for by individuals by using CAD tools that draw from the activity logs of firewalls. Secure web gateways provide a relatively convenient source of data on all external sites that are accessed from the corporate network.
- Route all remote traffic to the enterprise through a VPN (which is relatively uncommon) if your organization is highly motivated to track the cloud access activities of travelling and work from home systems. Ensure that all remote systems are routing all traffic through a Secure Web Gateway, or install a CAD mechanism that uses a local agent.
- Use CAD mechanisms with risk priority reporting to help IT security, privacy, compliance and other IT governance functions focus their attention where it can be most impactful.

Sample Vendors

Bitglass; Censornet; Eracent; Flexera; McAfee; Microsoft; Netskope; Scalable Software; Snow Software; Broadcom (Symantec)

Gartner Recommended Reading

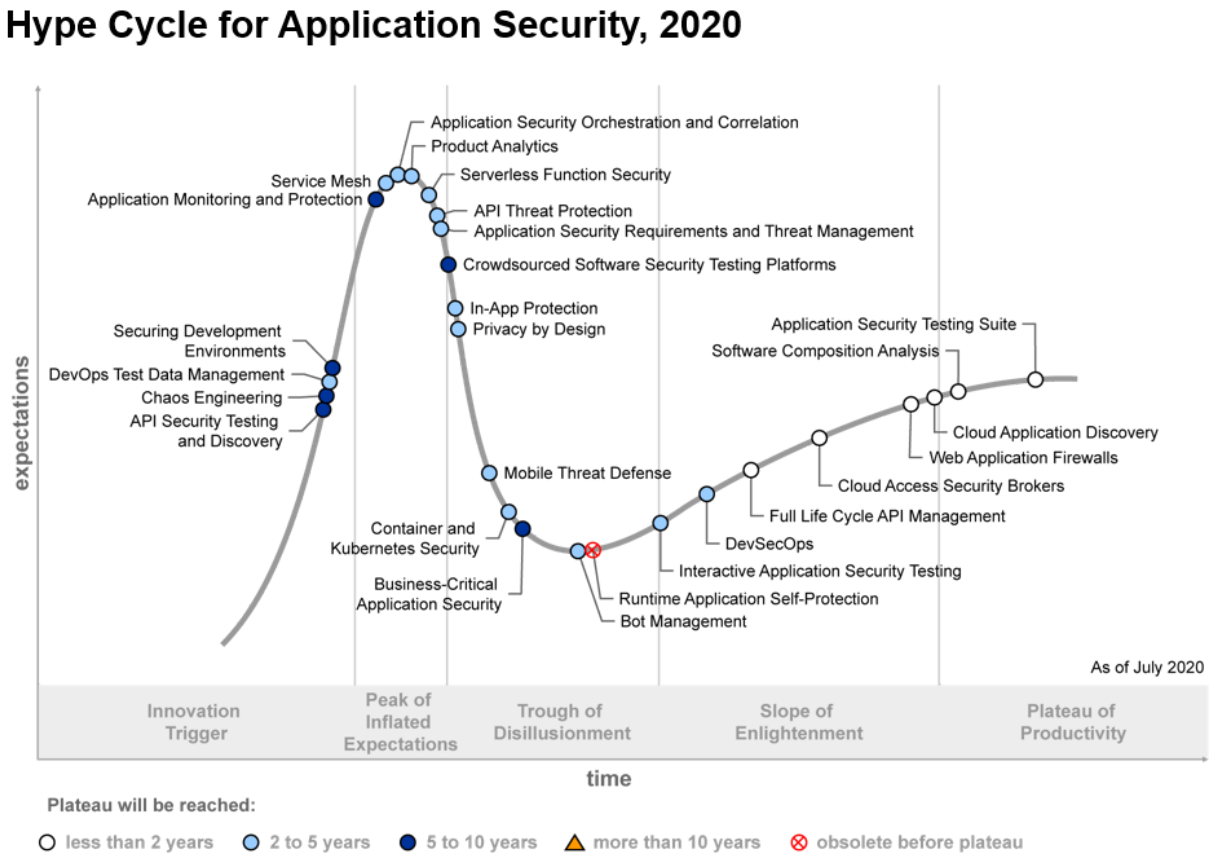
Market Guide for SaaS Management Platforms

Magic Quadrant for Cloud Access Security Brokers

How to Develop a SaaS Governance Framework

Appendixes

Figure 2. Hype Cycle for Application Security, 2020



Source: Gartner
ID: 448216

Gartner

Hype Cycle Phases, Benefit Ratings and Maturity Levels

Table 2: Hype Cycle Phases

(Enlarged table in Appendix)

Phase ↓	Definition ↓
Innovation Trigger	A breakthrough, public demonstration, product launch or other event generates significant media and industry interest.
Peak of Inflated Expectations	During this phase of overenthusiasm and unrealistic projections, a flurry of well-publicized activity by technology leaders results in some successes, but more failures, as the innovation is pushed to its limits. The only enterprises making money are conference organizers and content publishers.
Trough of Disillusionment	Because the innovation does not live up to its overinflated expectations, it rapidly becomes unfashionable. Media interest wanes, except for a few cautionary tales.
Slope of Enlightenment	Focused experimentation and solid hard work by an increasingly diverse range of organizations lead to a true understanding of the innovation's applicability, risks and benefits. Commercial off-the-shelf methodologies and tools ease the development process.
Plateau of Productivity	The real-world benefits of the innovation are demonstrated and accepted. Tools and methodologies are increasingly stable as they enter their second and third generations. Growing numbers of organizations feel comfortable with the reduced level of risk; the rapid growth phase of adoption begins. Approximately 20% of the technology's target audience has adopted or is adopting the technology as it enters this phase.
Years to Mainstream Adoption	The time required for the innovation to reach the Plateau of Productivity.

Source: Gartner (July 2021)

Table 3: Benefit Ratings

Benefit Rating ↓	*Definition* ↓
Transformational	Enables new ways of doing business across industries that will result in major shifts in industry dynamics
High	Enables new ways of performing horizontal or vertical processes that will result in significantly increased revenue or cost savings for an enterprise
Moderate	Provides incremental improvements to established processes that will result in increased revenue or cost savings for an enterprise
Low	Slightly improves processes (for example, improved user experience) that will be difficult to translate into increased revenue or cost savings

Source: Gartner (July 2021)

Table 4: Maturity Levels

(Enlarged table in Appendix)

Maturity Levels	Status	Products/Vendors
Embryonic	In labs	None
Emerging	Commercialization by vendors Pilots and deployments by industry leaders	First generation High price Much customization
Adolescent	Maturing technology capabilities and process understanding Uptake beyond early adopters	Second generation Less customization
Early mainstream	Proven technology Vendors, technology and adoption rapidly evolving	Third generation More out-of-box methodologies
Mature mainstream	Robust technology Not much evolution in vendors or technology	Several dominant vendors
Legacy	Not appropriate for new developments Cost of migration constrains replacement	Maintenance revenue focus
Obsolete	Rarely used	Used/resale market only

Source: Gartner (July 2021)

Acronym Key and Glossary Terms

ASRTM	application security requirements and threat management
CASB	cloud access security broker
CI/CD	continuous integration/continuous delivery
EAM	externalized authorization management
SAST	static application security testing
SRM	security and risk management
WAAP	web application and API protection
WAF	web application firewall

Evidence

[1] Gartner's Enabling Cloud-Native DevSecOps Survey 2021 was conducted from 12 May through 21 May 2021, among 85 respondents.

Document Revision History

Hype Cycle for Application Security, 2020 - 27 July 2020

Hype Cycle for Application Security, 2019 - 30 July 2019

Hype Cycle for Application Security, 2018 - 27 July 2018

Hype Cycle for Application Security, 2017 - 28 July 2017

Hype Cycle for Application Security, 2016 - 13 July 2016

Hype Cycle for Application Security, 2015 - 9 July 2015

Hype Cycle for Application Security, 2014 - 28 July 2014

Hype Cycle for Application Security, 2013 - 25 July 2013

Hype Cycle for Application Security, 2012 - 20 July 2012

Recommended by the Author

Some documents may not be available as part of your current Gartner subscription.

Understanding Gartner's Hype Cycles

Create Your Own Hype Cycle With Gartner's Hype Cycle Builder

Magic Quadrant for Application Security Testing

Magic Quadrant for Full Life Cycle API Management

Market Guide for Zero Trust Network Access

Integrating Security Into the DevSecOps Toolchain

Cool Vendors in DevSecOps

Market Guide for Software Composition Analysis

Table 1: Priority Matrix for Application Security, 2021

Benefit ↓	*Years to Mainstream Adoption*			
	Less Than 2 Years ↓	***2 - 5 Years*** ↓	***5 - 10 Years*** ↓	***More Than 10 Years*** ↓
Transformational		DevSecOps		
High	Full Life Cycle API Management IAST Software Composition Analysis Web Application Firewall Appliance	API Threat Protection Application Security Orchestration and Correlation Application Security Requirements and Threat Management Application Shielding Bot Management Crowdsourced Software Security Testing Platforms Policy-as-Code Security Service Edge Web App Client-Side Protection	API Security Testing Chaos Engineering Cloud WAAP	

Benefit	*Years to Mainstream Adoption*			
↓	***Less Than 2 Years*** ↓	***2 - 5 Years*** ↓	***5 - 10 Years*** ↓	***More Than 10 Years*** ↓
Moderate	Cloud Application Discovery Dynamic Data Masking Enterprise App Stores	Container and Kubernetes Security Digital Product Analytics Mobile Application Security Testing Privacy by Design Securing Developer Environments Serverless Function Security Service Mesh	Application Monitoring and Protection Business-Critical Application Security DevOps Test Data Management EAM	
Low				

Source: Gartner (July 2021)

Table 2: Hype Cycle Phases

Phase	*Definition*
Innovation Trigger	A breakthrough, public demonstration, product launch or other event generates significant media and industry interest.
Peak of Inflated Expectations	During this phase of overenthusiasm and unrealistic projections, a flurry of well-publicized activity by technology leaders results in some successes, but more failures, as the innovation is pushed to its limits. The only enterprises making money are conference organizers and content publishers.
Trough of Disillusionment	Because the innovation does not live up to its overinflated expectations, it rapidly becomes unfashionable. Media interest wanes, except for a few cautionary tales.
Slope of Enlightenment	Focused experimentation and solid hard work by an increasingly diverse range of organizations lead to a true understanding of the innovation's applicability, risks and benefits. Commercial off-the-shelf methodologies and tools ease the development process.
Plateau of Productivity	The real-world benefits of the innovation are demonstrated and accepted. Tools and methodologies are increasingly stable as they enter their second and third generations. Growing numbers of organizations feel comfortable with the reduced level of risk; the rapid growth phase of adoption begins. Approximately 20% of the technology's target audience has adopted or is adopting the technology as it enters this phase.
Years to Mainstream Adoption	The time required for the innovation to reach the Plateau of Productivity.

Phase ↓	Definition ↓

Source: Gartner (July 2021)

Table 3: Benefit Ratings

Benefit Rating ↓	Definition ↓
Transformational	Enables new ways of doing business across industries that will result in major shifts in industry dynamics
High	Enables new ways of performing horizontal or vertical processes that will result in significantly increased revenue or cost savings for an enterprise
Moderate	Provides incremental improvements to established processes that will result in increased revenue or cost savings for an enterprise
Low	Slightly improves processes (for example, improved user experience) that will be difficult to translate into increased revenue or cost savings

Source: Gartner (July 2021)

Table 4: Maturity Levels

Maturity Levels ↓	*Status* ↓	*Products/Vendors* ↓
Embryonic	In labs	None
Emerging	Commercialization by vendors Pilots and deployments by industry leaders	First generation High price Much customization
Adolescent	Maturing technology capabilities and process understanding Uptake beyond early adopters	Second generation Less customization
Early mainstream	Proven technology Vendors, technology and adoption rapidly evolving	Third generation More out-of-box methodologies
Mature mainstream	Robust technology Not much evolution in vendors or technology	Several dominant vendors
Legacy	Not appropriate for new developments Cost of migration constrains replacement	Maintenance revenue focus
Obsolete	Rarely used	Used/resale market only

Source: Gartner (July 2021)