PRAETORIAN

WHITE PAPER

The Evolution of Application Security Testing for Rapid Software Development

Security programs built on expertise, engineered to scale, and unified through software

INSIDE THIS REPORT

INTRODUCTION

The Need for Evolution

With mass adoption in cloud and container technologies, Internet-based companies are shipping code at unprecedented speed. The new pace in which code is being pushed to production is causing security teams to reexamine how they integrate security verification into the software development life cycle (SDLC).

Monolithic, one-time security gates are not satisfying the needs of these agile development teams. Daily code updates, through continuous integration and continuous delivery methods, often render the results of the annual security assessment obsolete.

To meet the needs of emerging developer methodologies, application technologies, attack vectors, and business objectives, high-performing software development teams and security professionals must continuously evolve their approach.

> **The Challenge:** Organizations must determine the appropriate combination of internal and external security efforts, such as leveraging external security expertise and bug bounty programs for vulnerability identification.

Currently, the security industry offers many different approaches for identifying security vulnerabilities in applications:

- **Automated scanning** - Subscription-based, fire-and-forget solutions
- **Bug bounty programs** - Crowdsourced vulnerability discovery
- **Professional services** - Expert-led comprehensive assessments
- **Continuous security testing** - CI/CD-integrated ongoing verification

APPROACH #1

Automated Scanners

Automated scanners spider your application and use scripts to check for the presence of security bugs. The process is usually "fire and forget," requiring you only to enter a target and then review the report that's generated at the end.

Strengths

Since it doesn't require a dedicated person on the provider's side, an automated scanner is usually offered as a subscription at a lower cost. These scanners are also often launched on demand, allowing you to easily verify remediation of previously identified vulnerabilities.

Limitations

Because they are entirely automated, these solutions can identify only test cases that are programmed. That often limits scanners to finding vulnerabilities that are simple to identify and/or exploit. They're completely unable to find issues that rely on human analysis, such as many authorization bypasses and logic flaws.

The resulting scan reports can have high false positive rates, as vendors rely on users for verification. While automated scanners provide value for quick checks and regression testing, they cannot replace human expertise for comprehensive security assessments.

APPROACH #2

Bug Bounty Programs

A bug bounty harnesses the crowd to identify vulnerabilities in your application by encouraging security researchers to find flaws. Those researchers are then paid for any security bugs they find, with higher severity vulnerabilities receiving higher awards.

Strengths

It can be a great way to quickly and economically identify security gaps in your platform, while staying within a defined budget. The crowd is always there—you don't need to wait for availability of an engineer with the right skillset.

APPROACH #2

Bug Bounty Program Challenges

As researchers are paid only for found issues, they are motivated to find low-hanging fruit that will yield a quick payday. This consequently motivates them to move on to new bounty targets, which presumably have been less scrutinized and hence are more likely to have easily found bugs.

> **Key Insight:** An application can receive a great deal of attention when its bug bounty program begins, but that typically wanes as researchers move on to new targets.

Hidden Total Costs

Most bug bounty programs require significant administrative overhead to:

- Triage incoming bug reports
- Weed out duplicates and false positives
- Manage the internal patch process
- Administer payments to researchers

> **The administrative overhead for managing a bug bounty program often exceeds the amounts paid as bounties, frequently by a factor of up to three times.**

Bug bounties cannot be relied on to provide thorough reviews, and it's far less likely that the researchers will identify exotic bugs that would be found by more persistent attackers.

APPROACH #3

Professional Services

A professional services assessment involves one or more engineers using a variety of tools (including automated scanners) and manual testing to identify application vulnerabilities. These assessments are typically performed within a defined window of time.

Strengths

Unlike bug bounties, the costs are fixed. Since you are paying for an expert's dedicated time, you should receive a more thorough assessment. You should also receive tailored remediation suggestions for each finding, strategic analysis, and follow-on advice.

There's also a benefit in having a dedicated engineer perform the assessment—they can be mindful of your specific concerns and risks to your clients. Many professional firms conduct these assessments according to an industry standard; Praetorian uses OWASP's Application Security Verification Standard.

Professional Security Evaluation Techniques

- **Penetration testing** - Simulated attacks to identify vulnerabilities
- **Run-time analysis** - Dynamic testing of running applications
- **Binary analysis** - Examination of compiled code
- **Code analysis** - Static review of source code
- **Design analysis** - Review of application architecture

Limitation: A professional assessment can provide an accurate picture of an application's risk at the time of assessment, but that assessment will grow stale as changes are made to the application and new vulnerabilities are discovered.

APPROACH #4

Continuous Security Testing

Emerging software development practices rely on rapid-iteration supported by continuous integration and delivery (CI/CD) technologies. Continuous security testing uses multiple analysis methods to identify new vulnerabilities introduced by incremental code movement.

Continuous security testing provides ongoing, comprehensive, and efficient security testing coverage at the speed of DevOps.

How It Works

The process starts with a complete professional assessment, during which the engineer annotates the code base for security-relevant portions and security bugs. The security assessment process continues when new code is pushed in the CI/CD pipeline, at which point a security review is triggered.

The new code is compared to the annotated code and, using a combination of machine learning and manual analysis, potential security issues can be identified quickly in security-relevant functions and new features.

The Shift Left Movement

High-performing development teams are "shifting left" more and more of their software delivery practices—including security testing. Activities that were traditionally done at the tail end of the SDLC or after a software release process are now moving earlier in the software delivery pipeline.

Goal of "Shifting Left": Identify security issues earlier, ultimately reducing their impact and cost in terms of risk and remediation efforts.

ANALYSIS

Strength of Various Approaches

While each approach delivers a unique set of strengths and weaknesses, successful application security programs use multiple analysis methods to identify new vulnerabilities introduced by incremental code movement.

Strength	Automated Scanners	Bug Bounty	Professional Services	Continuous Testing
Low Cost per Bug	x	x		x
Continuous and Scalable	x			x
Low False Positive Rate			x	x
Relatively On-Demand	x	x		x
Thorough Coverage			x	x
High Assurance			x	x
Industry Standards			x	x
Root Cause Analysis			x	x

> **Natural Synergy:** There's a natural synergy in combining the different techniques, as they individually excel at different things. Lowered costs and efficiencies come from using each assessment methodology in the use case where it delivers highest value.

The goal is to help our clients deliver the most secure application possible, at the lowest cost point, while balancing risk with time-to-market pressures.

COST ANALYSIS

Total Cost of Ownership: Bug Bounty Programs

Accounting for total cost of ownership across each approach is an important practice when measuring the value of a security program. One of the most underrepresented sources of unforeseen costs is often found in bug bounty programs.

> **Slack Case Study:** Slack's engineering team reported receiving up to 1,660 total bug report submissions during the first year of its public bug bounty program. Approximately 1,000 of those reports were received during the first four-month surge.

Year 1 Bug Bounty Costs (Slack Example)

Cost Component	Amount
Bounty Payments	$67,000
Processing Fees (20%)	$13,400
Management and Triage (2 FTE)	$181,250
Total Year 1 Cost	**$261,650**

> **Key Finding: Only 16.9% of bug reports are accepted and resolved on average. The majority are either not applicable, informative in nature, or duplicate reports.**

The total cost of ownership can be up to four times that of a program's bounty payout pool during the first year. These costs can be managed more effectively in subsequent years by outsourcing the bug triage.

CONCLUSION

Praetorian's Integrated Approach

Each approach has its strengths and weaknesses. There are tradeoffs among speed, coverage, assurance, timeliness, and cost. Combining bug bounty, professional services, and continuous security testing can create efficiencies by allowing each technique to do what it does best.

Recommended Workflow:

- Start with a professional assessment to bring a product up to a "known secure" baseline
- Follow with bug bounty and/or continuous security testing for DevOps
- Monitor the CI/CD pipeline to carry security assurance forward over time

Praetorian is unique in that we are able to perform all of the techniques and offer clients a unified dashboard in Diana to manage each chosen assessment technique. Diana provides a single platform to monitor progress, see results, track risk over time, and seek remediation verification.

Ready to Evolve Your AppSec Program?

Partner with Praetorian to implement the right combination of security testing approaches for your organization.

Contact Us

PRAETORIAN

www.praetorian.com | info@praetorian.com