WHITE PAPER

The Evolution of Application Security Testing for Rapid Software Development

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

praetorian

Integrated Approaches for Vulnerability Identification

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. In addition to embracing new methods for security verification and vulnerability identification, product teams must determine the appropriate combination of internal and external security efforts, such as leveraging external security expertise and bug bounty programs.

Currently, the security industry offers many different approaches for identifying security vulnerabilities in applications: automated scanning, bug bounty programs, professional services, and continuous security testing solutions. Any of the approaches will provide most organizations with some value. That said, we often have conversations in which we try to help our clients identify which approach will provide them the most value, while meeting their other goals and staying within budget.

We thought it might be worthwhile to try to distill those conversations, compare and contrast the approaches, and explain how Praetorian leverages different approaches in concert for increase efficiencies.

INSIDE THIS REPORT

- Learn how high-performing software teams are keeping pace with rapid release cycles by adopting new approaches for vulnerability identification while managing risk and time to market pressures.
- Explore new strategies for maximizing outcomes across bug bounty programs, professional services, and continuous security testing.
- Understand true cost of ownership when evaluating and measuring success across various methods for vulnerability identification.

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.

> ! Before delving into differences between vulnerability finding techniques, it's worth taking a moment to explain what each is. It's also worth noting that the first technique, automated scanning, is typically used by security researchers or engineers performing the other three methods.

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. Since it doesn't require a dedicated person on the provider's side, an automated scanner is usually offered as a subscription, and at a lower cost, compared to professional services.

These scanners are also often launched on demand, allowing you to easily verify remediation of previously identified vulnerabilities. Unfortunately, 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.

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. It can be a great way to quickly and economically identify security gaps in your platform, while staying within a defined budget.

There are some downsides to this approach, though. 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. These incentives mean that 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.

!

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.

One final note on bug bounties is that they can have higher, 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, and administer payments to researchers. Praetorian offers a concierge service to assist with these complications.

Professional Services

A professional services assessment involves one or more engineers using a variety of tools (to include automated scanners) and manual testing to identify application vulnerabilities. These assessments are typically performed within a defined window of time, with the costs determined on the basis of the hours allocated to the assessment. During the sales process, account managers will seek to determine the appropriate length of time (and subsequent costs), based on the application's size, complexity, and depth of assessment.

Unlike bug bounties, the costs are fixed, and often higher, but since you are paying for a expert's dedicated time, you should receive a more thorough assessment in return. You should also receive tailored remediation suggestions for each finding, strategic analysis, and follow-

→ Professional security evaluations employ a variety of techniques for uncovering unknown vulnerabilities, including:

- Penetration testing
- Run-time analysis
- Binary analysis
- Code analysis
- Design analysis

on advice. There's also a benefit in having a dedicated engineer perform the assessment, in that they can be mindful of your specific concerns and risks to your clients.

Automated scanners and bug bounties typically don't (or can't) approach an assessment with that mindset. Many professional firms conduct these assessments according to an industry standard or established methodology; Praetorian uses OWASP's Application Security Verification Standard.

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.

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

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 continuous integration and continuous delivery (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.

Whereas a traditional assessment can provide insight into risk at the time of the assessment, that insight becomes outdated as new code is pushed and features are added. The traditional approach to assessments has struggled to provide ongoing assurance with agile development, and that challenge has increased as organizations embrace multiple code pushes on a daily/weekly basis with DevOps.

A continuous security testing approach provides ongoing assurance for products developed via a rapid, iterative process. The approach also allows engineers to focus on incremental security-relevant code changes and new features, and thereby keeps time and costs down.

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. A summary of the strengths across various approaches is provided in the matrix below.

One emerging trend we're seeing is teams who are "shifting" more of their software delivery practices to the "left," including security testing.

→

High-performing development teams are "shifting left" more and more of their software delivery practices—including security testing.

Activities that were traditionally done after deployment and production, or things that are typically done later in the development or release process, are now moving earlier in the pipeline (or, to the left).

The goal of this left-shift is to identify security issues earlier, reducing their impact and cost.

ATTACK STRENGTH	Automated Scanners	Bug Bounty Programs	Professional Services	Continuous Security Settings
Low Cost per Bug	X	X		
Continuous and Scalable	X	X		X
Low False Positive Rate			X	X
Relatively On-Demand	X	X		X
Thorough Coverage of Security Controls			X	X
High Assurance			X	X
Industry Standards			X	
Root Cause Analysis			X	X

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 (or to the left). The goal of this "left-shift" is to identify security issues earlier, ultimately reducing their impact and cost in terms of risk and remediation efforts.

With that said, high-performing product teams should incorporate an appropriate mix of both pre- and post-release application security technologies and outside expertise support. It is possible to match the speed of DevOps while continuing to leverage trusted third-party expertise for annual assessments and penetration tests by allowing the various approaches to work in concert. A more detailed description of security programs built on expertise, engineered to scale, and unified through software is provided in following sections.

Consider Industry Standards for Professional Services

Between project deadlines and user demand for new features, security generally is not the highest priority for product development teams. Too often, identifying and remediating vulnerabilities is a task performed during the testing phase at the tail end of the software development lifecycle (SDLC). When it comes to secure coding, this reactive secure development approach is setting software teams up for failure.

To help product teams address emerging security challenges, Praetorian has created research-driven evaluation methodologies that incorporate guidance from the OWASP Application Security Verification Standard (ASVS), which normalizes the range in coverage and level of rigor applied to each application. With its 3 levels of testing rigor, 17 security control categories, and 211 defined test cases, this approach allows our team to meet your unique testing and budget goals by offering tiered pricing based on the comprehensiveness of the security review.

Praetorian follows the OWASP ASVS standards, which normalizes the range in coverage and level of rigor applied to each application. As part of a professional security evaluation, and depending on the level of rigor, Praetorian will employ a variety of techniques for uncovering unknown vulnerabilities.

COVERAGE KEY	
Excellent	🟦
Good	🟪
Fair	▨
Inadequate	⬜

→

This professional security evaluation methodology defines specific test cases that are in scope for each level of testing.

LEVEL 1 is meant for all software.

LEVEL 2 is for applications that contain sensitive data, which requires protection.

LEVEL 3 is for the most critical applications that perform high value transactions, contain sensitive medical data, or any application that requires the highest level of trust.

SECURITY CONTROL CATEGORY	LEVEL 1 OPPORTUNISTIC	LEVEL 2 STANDARD	LEVEL 3 ADVANCED
Architecture, Design and Threat Modeling	1/11	8/11	11/11
Authentication	17/26	24/26	26/26
Session Management	11/13	13/13	13/13
Access Control	7/12	11/12	12/12
Malicious Input Handling	10/21	20/21	21/21
Cryptography at Rest	2/10	7/10	10/10
Error Handling and Logging	3/13	9/13	13/13
Data Protection	4/11	8/11	11/11
Communications	7/13	9/13	13/13
HTTP Security Configuration	6/8	8/8	8/8
Malicious Controls	0/2	0/2	2/2
Business Logic	0/2	2/2	2/2
File and Resources	7/9	9/9	9/9
Mobile	7/11	10/11	11/11
Web Services	7/10	10/10	10/10
Configuration	1/10	5/10	10/10
Device Hardware NEW	10/29	20/29	29/29
TOTAL TEST CASES BY LEVEL	100	173	211

Praetorian's Approach to Integrated and Holistic Assessments

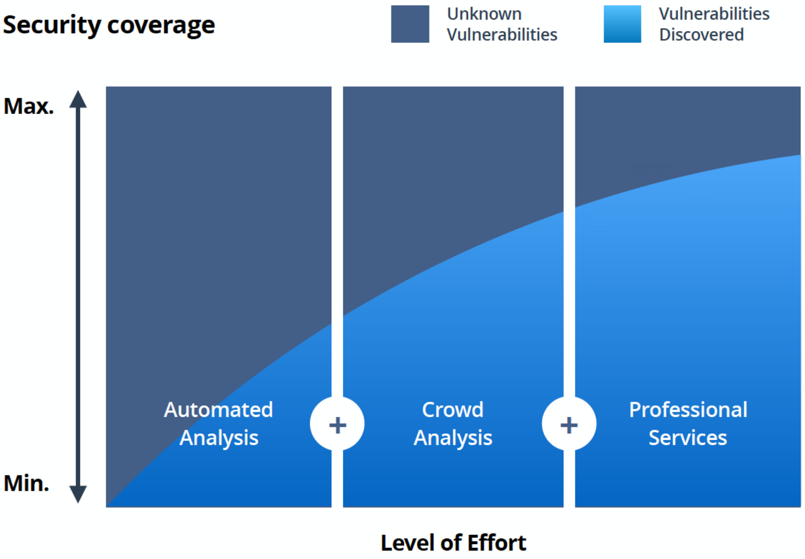
Each approach has its strengths and weaknesses. There are tradeoffs among speed, coverage, assurance, timeliness, and cost. Combining bug bounty, professional services, and continuous security testing can create efficiencies, though, by allowing each technique to do what it does best. Starting with a professional assessment, with a thorough approach and follow-on remediation advice, can help bring a product up to a "known secure" baseline. By following with bug bounty and/or continuous security testing for DevOps by monitoring the CI/CD pipeline, security assurance is then carried forward over time, across code pushes.

→

ASVS LEVEL 1 is meant for all software.

ASVS LEVEL 2 is for applications that contain sensitive data, which requires protection.

ASVS LEVEL 3 is for the most critical applications that perform high value transactions, contain sensitive medical data, or any application that requires the highest level of trust.



When organizations do perform more than one of these assessment techniques, they are typically done by different vendors, in isolation, and the client is responsible for aggregating and making sense of the results. 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.

→

High-performing development teams are "shifting left" more and more of their software delivery practices—including security testing.

Activities that were traditionally done after deployment and production, or things that are typically done later in the development or release process, are now moving earlier in the pipeline (or, to the left).

The goal of this left-shift is to identify security issues earlier, reducing their impact and cost.

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.

In essence, Diana is a security-as-a-service platform. By integrating bug bounty, professional services, and continuous security testing, Diana compensates for the shortcomings in each technique and allows for deep, continuous assessments. Security assurance can now be delivered continuously at today's speed of development, without the limitations of any one assessment method. Most importantly, through Diana, your security posture updates and evolves as the threat evolves, shrinking your windows of vulnerability instead of waiting until our next annual security assessment. The Diana portal provides a dashboard to track known risks, remediation statuses, and metrics over time.

This approach isn't going to be appropriate for all organizations. For many, it's overkill. If a robust and mature security program hasn't been in place, it can lead to an overwhelming number of findings.

For organizations starting security assessments for the first time, it's advisable to start with a professional services assessment in order to receive both a thorough review and remediation suggestions. Your security partner can then provide specific recommendations for follow-on activities, based on their knowledge of your product and its unique security concerns.

Summary of Strengths and Weaknesses

→

Continue to leverage the trusted, in-house expertise from your security services partners while scaling up with new approaches.

Extends security evaluations that represent a single, snapshot in time with professionally managed bug bounty programs and continuous security analysis supported by continuous integration and delivery (CI/CD) technologies.

APPROACH	ADDITIONAL DETAILS ON STRENGTHS AND WEAKNESSES
Automated Scanners	Least expensive. On demand. Typically run and administered in house. X High false positive and false negative rates. X Cannot identify several classes of vulnerabilities.
Bug Bounty Programs	Set a defined ceiling for bounty payouts based on your budget. Flexible timing. The crowd is always there. Don't need to wait for availability of an engineer with the right skillset. Can identify low-hanging fruit before a professional assessment, allowing the comprehensive assessment more time to identify harder-to-find bugs. Little assurance. No guarantee of depth, thoroughness. X Less likely to uncover certain classes of bugs. Testers motivated to find highpaying, low-hanging fruit and then move on. Unlikely to discover esoteric, hard-to-find bugs. X Typically insufficient for savvy clients, regulators, compliance. X Limited to black box testing in most cases. X High total cost of ownership — administrative overhead related to triage, verification, bounty payments, researcher communication, etc. X No remediation support.
Professional Services	Thorough, can be benchmarked to standards. Opportunity for more in-depth approaches than bug bounty, through sharing source code and knowledge. More likely to detect bugs unique to the tested application. Support before and after the assessment with preparations, remediation suggestions, prioritization, best practice, etc. "There's someone you can call." X More expensive than bug bounties. X Typically performed on annual or semi-annual cycle because of expense, not in coordination with a release cycle. X Timing restricted based on consultant availability.
Continuous Security Testing	Following a comprehensive assessment, allows for same degree of thorough testing against new code pushes. Time-efficient and cost-effective way to thoroughly test new features, updates. Continuously evolving to address new threats. Integrates with CI/CD pipeline to "trigger" assessments when code pushes X Requires a comprehensive assessment as a starting point.

Considering total cost of ownership across approaches

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

As previously described, this specific crowdsourcing model defines a payout scale for vulnerabilities identified, typically based on criticality, and invites a broad or select pool of security researchers to hunt for bugs until the bounty purse is exhausted. Many people new to bug bounty programs incorrectly assume that the majority of a program's investment stems from paying the bounties themselves for security bugs identified and professionally validated. However, that does not account for the lion's share of program costs, which should also include resources required for program management, bug report triage, and processing fees. Most organizations find that the administrative overhead for managing a bug bounty program exceeds the amounts paid as bounties, often by a factor of up to three times. Despite the allure of bug bounty scalability, this approach is still very much a human endeavor rooted in managing labor costs as a program evolves.

The following cost table and program analysis details a hypothetical total cost of ownership breakout for a successful bug bounty program. The analysis is based on publicly available information sourced from major bug bounty platform providers, bug bounty customer experiences, and Praetorian's direct involvement in major bug bounty initiatives.

→

The concept of a bug bounty program has been around for many years at larger software organizations, many companies are experimenting with them for the first time as they try to keep pace with today's accelerated development life cycle.

While not a true replacement for other vulnerability identification activities, modern-day bug bounty programs will continue to be a critical component of maturing and ongoing security programs.

BUG BOUNTY PROGRAM	Customer Managed Programs and Triage		Outsourced Triage
	Year 1	Year 2	Year 3
Bug Reports Received	1,660	775	775
Signal to Noise Ratio	16.9%	16.9%	16.9%
Bugs Expected Resolved	281	131	131
Management and Triage	$181,250	$145,000	$58,125
Triage Cost per Bug	$109	$187	$75
Bounty Payments	$67,000	$63,000	$63,000
Processing Fees (20%)	$13,400	$12,600	$12,600
TOTAL COST PER YEAR	$261,650	$220,600	$133,725
Average Bounty Payout	$238	$481	$481
Cost per Accepted Bug	$931	$1,684	$1,021

Slack (slack.com), the provider of cloud-based collaboration tools and services reportedly valued at $5 billion, recently shared its experience after three years of running a bug bounty program on the HackerOne platform. The analysis above leverages data points that Max Feldman, staff product security engineer at Slack, shared in his online post, Slack Bug Bounty: Three Years Later [1].

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. A surge is experienced during the first few months immediately following the launch of most new programs. This is primarily a result of bug hunters being attracted to the prospect of low hanging fruit and quick wins—a perceived gold rush.

Slack also reports being underprepared to handle the triage process internally during its surge, which led it to ultimately outsource to a third-party triage service the following year. The end-to-end bug bounty process at Slack consisted of these steps:

→

A bug bounty triage process involves validating vulnerabilities, removing false positives, removing duplicate reports, assigning severity, providing remediation guidance, and explaining to hackers why bug report submissions are rejected.

1. A researcher submits a vulnerability report

2. Slack's engineers would evaluate the report and determine if the bug is valid

3. If the bug is valid they file an internal ticket to track and fix the issue

4. Slack fixes the issue

5. Slack reaches out to the researcher for verification of the fix

6. Slack rewards the researcher for their finding

During that initial four-month period, Slack reports paying out approximately $30,000 in bounties and a total of $67,000 over the course of its first year. But, as you can see from the table above, there is more to account for when considering the total cost of ownership.

Slack also reports that 16.9% of bug reports are accepted and resolved on average. The majority of bug reports are either found to be not applicable, informative in nature, and/or duplicate reports. This indicates that Slack experienced a relatively high "signal to noise" ratio when compared to Facebook's (4%)[2] and GitHub's (1.4%)[3] security bug report validity rate. Applying this average ratio to its first year, Slack would have accepted and resolved approximately 280 of the 1,660 total bug reports received at an average bounty payout of $238.

Rapid response rates and triage velocity (i.e., steps 1–6 listed above) are critical to maintaining overall program health and sustained interest from "the crowd." Slack admits its engineering team was underprepared during the initial surge. Given the high volume and flow of bug report submissions Slack received during its surge, it's reasonable to assume that it would have taken two full-time resources during that four-month period to maintain pace with its triage needs while maintaining optimal program health metrics.

Following the surge, Slack could then scale back to a single dedicated resource for the remainder of year one and into year two if it were to continue to internally manage its triage process, which it did not.

→

Bug bounty program health is largely dependent on your organization's rapid response to bug report submissions, triage velocity, and sustained interest from "the crowd."

After the surge, it's an arms race for attention.

With its engineering team primarily based in San Francisco, Slack faces an average cost of $145,000 for a full-time security engineer, as reported by a Glassdoor[4]. Assuming the need to leverage up to two full-time resources during the surge, the total true cost of internal program management and triage would be approximately $181,250 for the first year. Add to it the annual cost for bounty payouts and the platform's required 20% processing fee of $13,400, which according to HackerOne[5] covers "access to many of the platform features, payment remittance to hackers, and all other associated services for payment processing such as tax form collection, year-end 1099 issuance, etc.," and you get up to $261,650—closer to the true cost of ownership for year one.

It's common for organizations new to bug bounty programs to focus on bounty payments only and assume they are the primary cost driver tied to a program as it scales. The reality is that the total cost of ownership can be up to four times that of a program's bounty payout pool during the first year. As Slack experienced, these costs can be managed more effectively going into subsequent years by outsourcing the bug triage, which has the effect of significantly reducing the cost per bug triaged and response time. All of these are important metrics to monitor when evaluating ongoing program health.

[1] https://slack.engineering/slack-bug-bounty-three-years-later-ad59e9188603

[2] https://www.facebook.com/notes/facebook-bug-bounty/2015 highlights-less-low-hanging-fruit/1225168744164016

[3] https://github.com/blog/2099-two-years-of-bounties

Kyle Randolph[6], security engineer at Optimizely, also concluded after running a bounty program on Cobalt that "the biggest cost of the program is not the rewards themselves, but the time spent triaging bugs." Similar to Slack, Optimizely invests in a triage service to help filter out duplicate and poor quality reports, but "still invest[s] significant time reviewing and reproducing reports." Running its program on Cobalt's platform while leveraging Cobalt's triage services, Optimizely reported that many of the original bug report submissions "still required an engineer to spend time reviewing" and that the "reports tend to be Low severity, resulting in more time spent" on Low severity bugs than Randolph's team would like. But this is to be expected. Ning Wang, chief financial officer at HackerOne, acknowledged in a published article that "responding to the incoming security reports is interrupt-driven work and can be time consuming."

→

Bug bounty program health is largely dependent on your organization's rapid response to bug report submissions, triage velocity, and sustained interest from "the crowd."

After the surge, it's an arms race for attention.

Furthermore, Optimizely believed that all low-hanging, High-severity fruit had been found by researchers after the first year of its bug bounty program. At the time of Randolph's documented experience, it had been five months since they received a High or Critical severity bug. To encourage advanced security researchers to maintain interest in digging deeper for High severity bugs, Optimizely recognized the need for stronger incentives. Going into the program's second year, Kyle's team increased the High severity bounty by five times, and going forward they are committed to paying a minimum of $5,000 for a single High severity bug.

[4] https://www.glassdoor.com/Salaries/san-francisco-security-engineer-salary-SRCH_IL.0,13_IM759_KO14,31.htm

[5] https://www.hackerone.com/blog/bug-bounty-budget

[6] https://medium.com/engineers-optimizely/raising-the-security-bug-bounty-b8abeb46409a

We are the Security Experts.

As a collective of highly technical engineers and developers offering deep security expertise, Praetorian solves the toughest challenges faced by today's leading organizations across an ever-evolving digital threat landscape. Our solutions enable clients to find, fix, stop, and ultimately solve cybersecurity problems across their entire enterprise and product portfolios.

VISION To solve the cybersecurity problem.

MISSION To make the world a safer and more secure place.

Praetorian is a collective of highly technical engineers and developers with decades of industry experience. Our singular focus on information security solutions delivers unbiased expertise.

The value we provide stems directly from our engineering culture—a continuous pursuit of efficiency and improvement in all operations. From proprietary methodologies and toolsets to project management and back office operations, we deliver quality results while decreasing your costs.

- Internet of Things
- SaaS Applications
- Mobile Applications
- Cloud Infrastructure
- Corporate Infrastructure
- Critical Infrastructure

Read to Get Started?

We provide deep security expertise to teams in today's leading organizations.

Are you ready to discuss your next security initiative?

Contact us at (866) 477-1028

www.praetorian.com
sales@praetorian.com

We are the security experts solving your cybersecurity problems. Gain confidence that your place in the next wave of innovation is secured. Learn more at **praetorian.com.**

praetorian