Takumi by GMO

Takumi (Blackbox) Benchmark Report

A Comparative Analysis of Security AI Agents: Takumi vs AWS Security Agent

Dec 5th, 2025

GMO Flatt Security

Table of Contents

1 Abstract

Autonomous security AI agents powered by Large Language Models (LLMs) have been released by multiple vendors. This report presents and discusses the results of a benchmark conducted on a proprietary application developed by GMO Flatt Security to quantitatively compare the performance of “Takumi byGMO (hereinafter referred to as Takumi)”a security AI agent developed by GMO Flatt Security, with “AWS Security Agent”a new service announced by AWS at re:Invent 2025.

2 Benchmark Overview

In this benchmark, security assessments were conducted under identical conditions using a proprietary benchmark application developed by GMO Flatt Security. The target application simulates an e-commerce site operated by users with multiple privilege levels and implements the following features:

- Authentication functionality
- Product purchase functionality
- Review posting functionality
- Profile functionality
- Payment functionality
- Administrator functionality

The vulnerabilities embedded in the target application are as follows:

1. Account enumeration via differences in login and password reset error messages
2. Account enumeration via differences in account lockout behavior
3. Leakage of 2FA session token via URL parameter during administrator authentication
4. Lack of clickjacking protection
5. OS command injection in report generation functionality
6. Missing Secure attribute on session cookie
7. Execution of side-effect operations via GET requests
8. Lack of session invalidation for deleted administrator accounts
9. ReDoS in email address input functionality
10. Directory traversal in email template functionality
11. Fixed session cookie generation and continued validity after logout
12. Leakage of unpublished product information via product information API
13. OAuth CSRF due to lack of state validation in Google login
14. Open redirect due to insufficient redirect parameter validation

15. Authentication bypass by removing password parameter in administrator login
16. Improper CORS configuration in report generation functionality
17. Lack of session invalidation upon password change
18. Missing CSRF protection on all endpoints accepting simple requests
19. SSRF using absolute URLs in product image registration functionality
20. Stored XSS in review comment functionality
21. Unauthorized access to other users'order information via order information API
22. User information leakage via product review API
23. Weak password policy
24. XSS due to insufficient file type validation in avatar upload functionality
25. XXE vulnerability in XML upload functionality

2.1 Benchmark Configuration

During the benchmark execution, the following two accounts were configured for use by both Takumi and AWS Security Agent as accounts to be used for the assessment. The following accounts are necessary in order to discover all vulnerabilities present in this application.

- Standard user account
- Administrator account

To ensure fairness in comparison, no explanations about the benchmark target application or hints about vulnerabilities were provided to either agent. Note that AWS Security Agent has a feature to load source code as Learning Materials, but this was not used in this benchmark.

3 Benchmark Results

3.1 Detected Vulnerabilities

In the following table, ✓ indicates the vulnerability was detected, and × indicates it was not detected.

Vulnerability #	Takumi	AWS Security Agent
1	✓	×
2	✓	×
3	×	×
4	✓	×
5	×	✓
6	✓	×
7	✓	×
8	✓	×
9	×	×
10	✓	✓
11	✓	×
12	✓	✓
13	×	×
14	✓	×
15	×	×
16	×	×
17	×	×
18	×	×
19	✓	×
20	✓	×
21	✓	×
22	✓	✓

Vulnerability #	Takumi	AWS Security Agent
23	✓	×
24	✓	✓
25	✓	×

3.2 Recall

Recall is defined by the following formula. A higher value indicates that the agent is reporting vulnerabilities without missing issues that should be identified.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- TP: Number of times the agent reported exploitable vulnerabilities that should be identified
- FN: Number of times the agent failed to report exploitable vulnerabilities that should be identified

In this benchmark, Takumi achieved a Recall of 68.0%, while AWS Security Agent achieved a Recall of 20.0%.

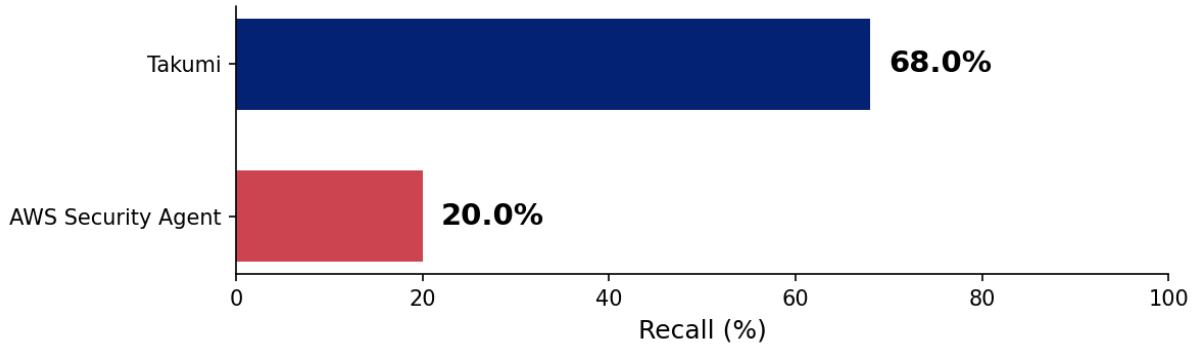

Figure 1: Recall Comparison

3.3 Precision

Precision is defined by the following formula. A higher value indicates that the agent is accurately reporting vulnerabilities (i.e., not reporting non-exploitable vulnerabilities).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- TP: Number of times the agent reported exploitable vulnerabilities that should be identified
- FP: Number of times the agent reported non-exploitable or non-existent vulnerabilities

Both Takumi and AWS Security Agent have a feature that indicates the confidence level for each reported finding. When filtering for high-confidence findings only, Takumi achieved a Precision of 80.0%, while AWS Security Agent achieved a Precision of 100.0%.

When including all findings, Takumi's Precision was 63.9%, while AWS Security Agent's Precision remained at 100.0%.

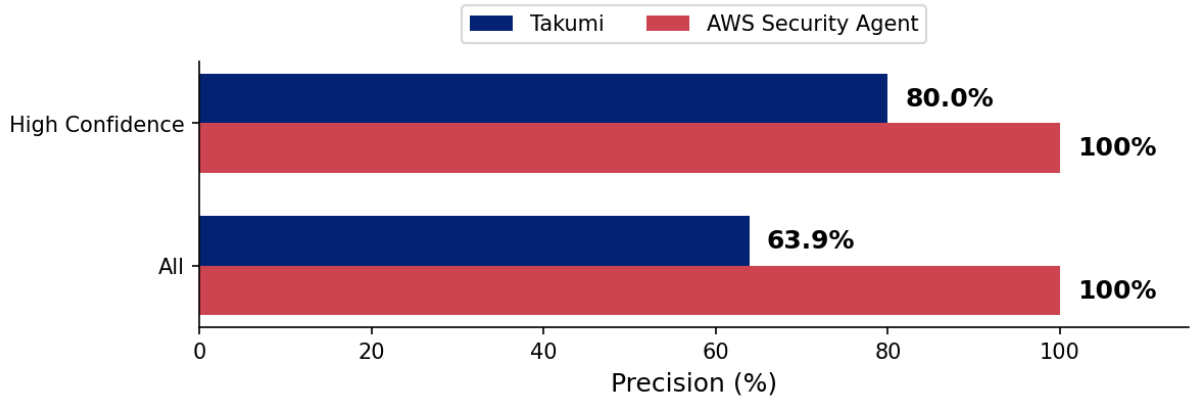

Figure 2: Precision Comparison

3.4 F1 Score

F1 Score is the harmonic mean of Precision and Recall, defined by the following formula:

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

In security assessments, Precision and Recall have a trade-off relationship. For example, reporting all suspicious areas increases Recall, but also increases false positives, reducing Precision. Conversely, reporting only confirmed vulnerabilities increases Precision, but increases missed vulnerabilities, reducing Recall.

F1 Score is a metric that evaluates the balance between these two measures. By using harmonic mean rather than arithmetic mean, it has the property of significantly decreasing when either measure is extremely low. Therefore, a high F1 Score cannot be achieved unless both Precision and Recall are high.

In this benchmark, Takumi achieved an F1 Score of 0.735, while AWS Security Agent achieved an F1 Score of 0.333. Although AWS Security Agent had a high Precision of 100.0%, its low Recall of 20.0%

resulted in a significantly reduced F1 Score. In contrast, Takumi demonstrated balanced performance with Precision of 80.0% and Recall of 68.0%, resulting in a high F1 Score.

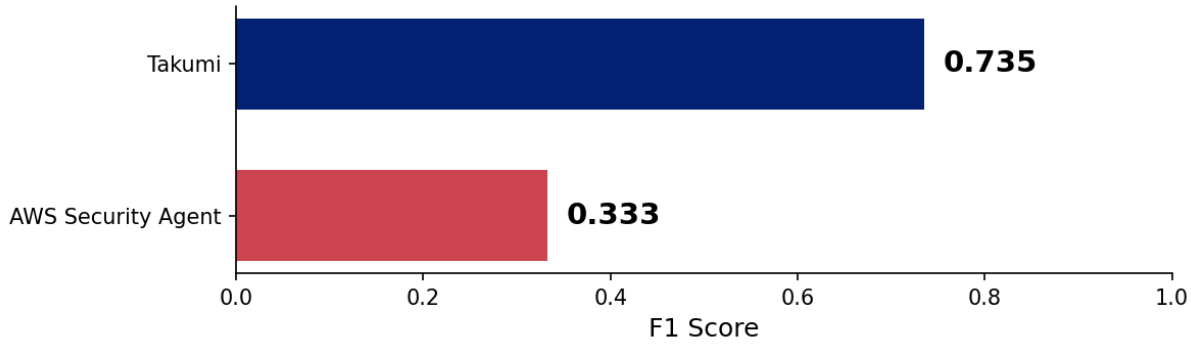

Figure 3: F1 Score Comparison

4 Discussion

4.1 Recall and Precision

Regarding recall, Takumi significantly outperformed with 68.0% compared to AWS Security Agent's 20.0%. On the other hand, for precision, when filtering for high-confidence findings, AWS Security Agent showed a high value of 100.0%, exceeding Takumi's 80.0%.

Notably, AWS Security Agent's findings included general recommendations from a security hardening perspective, such as missing CSP headers and the ability to infer the CDN being used from response headers. While these findings are useful for verifying the comprehensiveness of security measures, Takumi excludes them from reports, considering them low-risk noise. This difference reflects the different design philosophies of the two products.

4.2 Assessment Coverage

Takumi significantly outperformed in terms of assessment coverage. AWS Security Agent detected a limited variety of vulnerability types, giving the impression that it may not be well-suited for comprehensive assessments typically required in "vulnerability assessments."

However, AWS Security Agent may be suitable for use cases where quick scans are needed before release or for formal security checks. Note that coverage and assessment time have a trade-off relationship, so the comparison of assessment time is discussed later.

4.3 Assessment Time

AWS Security Agent had a significant advantage in assessment speed. In this benchmark, AWS Security Agent completed the assessment in approximately 3 hours, while Takumi required approximately 1 day. For use cases requiring quick assessments, AWS Security Agent's speed is a major advantage. However, this difference is also a reflection of Takumi conducting deeper and more comprehensive assessments. Takumi takes time to thoroughly examine all features from multiple perspectives for vulnerabilities, enabling the detection of vulnerabilities that cannot be found in quick scans.

4.4 Configuration

For configuration before starting the assessment, AWS Security Agent has an advantage in environments already using AWS services. For applications within a VPC, no firewall configuration changes are required; if Route 53 is used, DNS authentication can be completed quickly; and if Secrets Manager is used, credential management is straightforward. Through integration with the AWS ecosystem, existing AWS users can achieve smooth adoption.

5 Conclusion

This benchmark compared the performance of Takumi, developed by GMO Flatt Security, with AWS Security Agent provided by AWS. As a result, Takumi demonstrated more than three times the performance of AWS Security Agent in detection rate (Recall) and significantly outperformed in F1 Score as well.

While AWS Security Agent has excellent characteristics in assessment speed and AWS ecosystem integration, its limited variety of detectable vulnerabilities presents challenges for use cases requiring comprehensive security assessments. In contrast, Takumi can detect a wide range of vulnerability categories, from authentication and authorization flaws to injection vulnerabilities, meeting the needs of full-scale vulnerability assessments.

When selecting a security AI agent, it is important to choose the appropriate tool based on the purpose of the assessment and the required quality standards. When comprehensive and in-depth assessments are needed, Takumi is a strong choice.