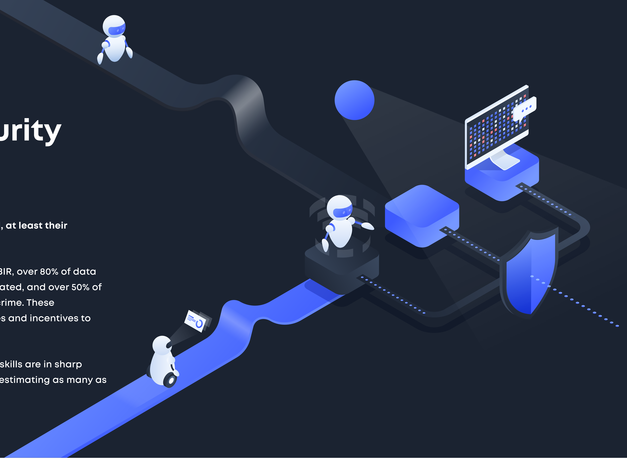
testim

Cybersecurity Spotlight



Cybercriminals never sleep, well, at least their bots don't.

According to the 2020 Verizon DBIR, over 80% of data breaches were financially motivated, and over 50% of breaches were from organized crime. These organizations have the resources and incentives to persist.

At the same time, cybersecurity skills are in sharp demand with Frost and Sullivan estimating as many as 1.8 million unfilled jobs by 2022.

Enter cybersecurity organizations, using AI and new technologies to help organizations keep these risks at bay. The global market for cybersecurity tools is expected to grow by over 10% per year to $248B by 2023.

Cybercriminals aren't relaxing the pressure. Some are building exploit kits that they sell on the dark web. Others leverage massive computing power and artificial intelligence to search for and attack known vulnerabilities in new and innovative ways.

To serve the needs of this market, cybersecurity tools need to be easy-to-use to augment inexperienced cyber skills. They also need to be constantly available to recognize and counter threats whenever criminals attack. And perhaps most importantly, the cybersecurity tools need to keep improving to stay ahead of the criminals who seek to exploit weaknesses faster, before they are remediated.

Software quality impacts security.

For many cybersecurity organizations, your web application is your service. If it goes down or doesn't function properly, your clients' businesses may be exposed. Perhaps it's the opportunity a malicious bot needs to find an entry point.

Cybersecurity companies using Testim

Microsoft FORTER THETARAY

Complicating Challenges

Criticality

Your customers depend on your service for their security. What could be more important? If your new release breaks a feature, you need to diagnose, build a fix, and release it before your customers are impacted. Your testing can't miss functional errors, or slow down your recovery efforts.

Innovation

As a security firm, organizations depend on you to invent new features that keep them protected. As a business, you need to stay ahead of your competitors who are constantly evolving. Moving fast means releasing new features and updating your service. Testing can't be the bottleneck.

Collaboration

Modern DevTest teams strive to automate and accelerate their release cadence. Collaborating on new features and how to test them is critical to ensuring quality when teams are flying fast. Testing needs to be seamlessly integrated into every stage of the development lifecycle, eliminating friction and elevating teamwork.

How Testim helps

Fast authoring

Users can record user flows in minutes, configure, or add custom steps to quickly create test coverage for new features. Testim's intuitive UI enables all agile team members to participate in QA and minimizes context switching for your developers. Users can customize tests through the configuration of validations Features like loops simplify repeating your test across similar UI elements. If code is preferred, tests can be exported and edited in your IDE and stored in GitHub.

Stable tests

Testim tests don't break because of minor changes that impact CSS locators or element text. Instead, our AI-based Smart Locators find the element based on related attributes and pass the test. Less time spent fixing tests means more time for building code coverage. Your developers also gain confidence that regressions will be caught.

Integrated

Test branches, whether codeless or coded can be synchronized with your application code. Tests can be set up to automatically execute on different milestones like smoke tests on code commits, functional tests on pull requests, and E2E suites on release candidates. Test results should surface through your preferred collaboration tools like Slack or Jira.

Troubleshooting

When tests fail, Testim gives you the tools needed to accelerate diagnosis including comparison screenshots, console logs, and HAR files at the failed step. Users can troubleshoot failed tests they didn't write, removing dependencies on specific resources. Aggregated errors help find recurring problems and test failure insights reports help inform process improvements.