CASE STUDY

Michigan State University implements language security technology to address runtime threat gap

PREVOTY

In recent years, Michigan State University (MSU) set out to make information security a critical priority, inviting the former CISO for Farmers Insurance Group, Rob McCurdy, to return to his alma mater as its first-ever CISO. Rob is tackling new initiatives at the university – from building a security team from the ground up to championing the latest approaches for protecting a thriving community.

MSU at a Glance

MSU is a sprawling microcosm with a complex IT environment and large user base:

- A top-tier public research university
- Division I sports heavyweight
- 7th largest school in the U.S. in terms of enrollment
- Tens of thousands of faculty, staff, alumni, and customers for its multiple campuses, healthcare facilities, hotels, event centers

Application Security in Higher Education

"Attackers know application security is not taught in many curricula. Attackers know security may not be a priority for application go-lives. Attackers know it takes them seconds to minutes to compromise an application, but it takes the industry months to identify it."

- Rob McCurdy, CISO, MSU

Every college and business within MSU has its own IT department, but Rob leads the only security team for the university. Striking a balance between preserving and securing an open, fragmented IT application environment – where end users are encouraged to freely share information – is a business challenge unique to higher education.

The Challenges of Protecting a "Business of Businesses"

1. Fragmented IT units

Centralized security must be consultative – not critical – to help developers succeed.

"From one college to another, it's a completely different language and a constant learning curve. A hotel's needs differ from that of a research unit. Solutions must be agile enough to support these siloed IT environments. "

2. Regulatory pressures

(PCI, HIPAA, FERPA) often complicate the need to make information accessible.

"As a public university, free-flowing information is a cornerstone. Security controls mustn't impede research/education or break something, as we are seen as a service provider to the community."

3. Every solution must be defensible and high-performing

While satisfying interdepartmental requirements.

"Security is newer within MSU, still going live with new services. We catch departments at varying stages of development. If we're not integrated from the start, we will likely delay the project and impact the business."

One of MSU Information Security's major goals was to reduce application layer risks across the board. Using a layered, defense-in-depth approach that has statistically been proven to improve overall security in porous application environments, MSU Information Security implemented a consistent blend of security monitoring, vulnerability management, intrusion prevention, and testing, which centralized security without disrupting each department.

The last critical component was runtime response; how would MSU quickly ramp up security for applications in advanced stages of development -- or even in production?

1. Org-wide Education/Awareness
2. Policies/Secure Software
3. Development Life Cycle (**SSDLC**)
4. Static code analysis during development (**SAST**)
5. Dynamic code analysis prior to go-live (**DAST**)
6. Web Application Firewall (**WAF**) for detection/prevention
7. Interactive testing (**IAST**)
8. Runtime security, such as with Runtime Application Self-Protection (**RASP**)

Runtime Application Security and RASP

"Runtime application self-protection (RASP) is important for data flow analysis. It has a more complete view. Static testing can only find a percentage. Unlike a WAF, which looks for an attack's 'signatures' or abnormal behavior patterns, RASP looks for the logic and data flows."

Prevoty's runtime security has the unique capability to neutralize active application threats in production, without relying on pattern matching or definition-based detection. MSU implemented Prevoty's Runtime Application Self-Protection (RASP) solution -- a lightweight, language security technology with unmatched speed, proven scalability, and a simplified deployment model. Since MSU cannot always use a centralized appliance, the organization brought the Prevoty solution entirely on-premise using pre-built SDKs available for Java, C#, Ruby, Node.js, PHP, Python and Go. Content, query, and token-based application attacks such as cross-site scripting (XSS), cross-site request forgery (CSRF), and SQL injections (SQLi) are automatically prevented as the application runs and reported in real-time.

"Being at runtime, Prevoty has a lot more visibility and context. It has a higher correct rate than testing or firewalls and the fullest possible picture of what's actually being executed in real-time. For instance, Prevoty can see how the application will handle a potential SQLi. It doesn't just see that there's an extra special field in a character that shouldn't be there; it figures out how the application is going to treat that post and neutralizes it."

Intelligent, Automated Security with Prevoty

No More Remediation Stalls
"We still do testing and QA, but Prevoty allows us -- even if the app's already in production --- to put protections in place while we remediate the code. It buys us time to fix it and bake it into our release cycle, which is easier to absorb. It's both breach avoidance and cost avoidance."

No Performance Delays
"We haven't identified any noticeable performance impact with Prevoty, and due to how it's implemented, it won't be a hassle to scale with it horizontally."

To learn more or request a demo, contact info@prevoty.com