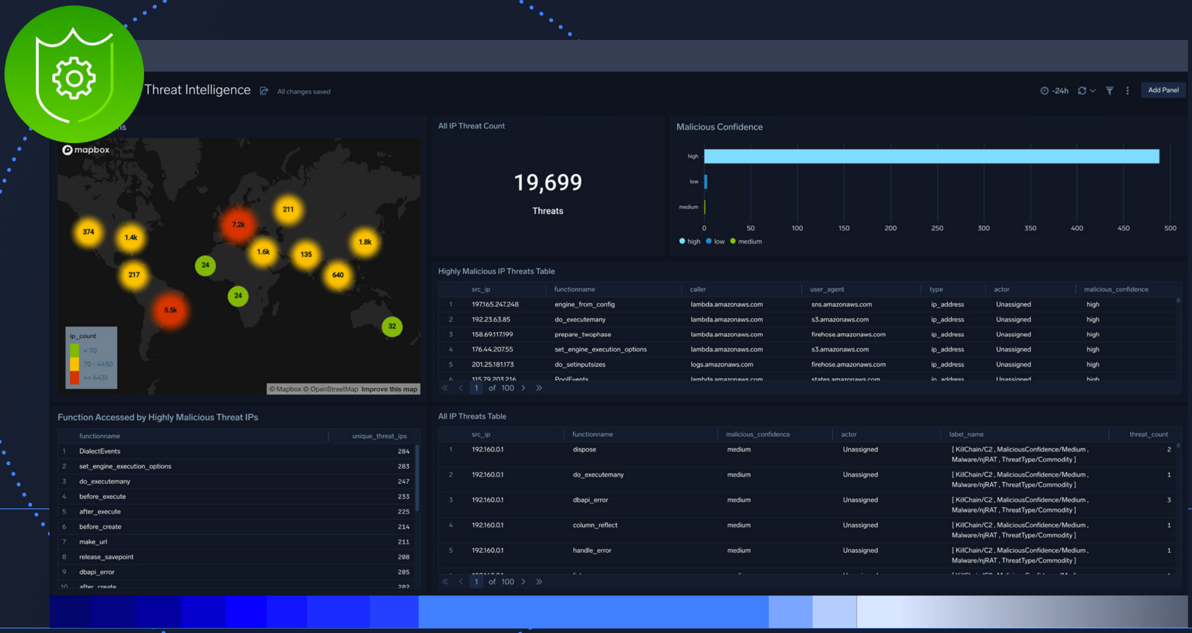
APPLICATION SECURITY GUIDE

How to keep your apps safe and accelerate release cycles



All IP Threat Count

19,699

Threats

Highly Malicious IP Threats Table

	src_ip	functionname	caller	user_agent	type	actor	malicious_confidence
1	197.165.247.248	engine_from_config	lambda.amazonaws.com	sns.amazonaws.com	ip_address	Unassigned	high
2	192.23.63.85	do_executemany	lambda.amazonaws.com	s3.amazonaws.com	ip_address	Unassigned	high
3	158.69.117.199	prepare_twophase	lambda.amazonaws.com	firehose.amazonaws.com	ip_address	Unassigned	high
4	176.44.207.55	set_engine_execution_options	lambda.amazonaws.com	s3.amazonaws.com	ip_address	Unassigned	high
5	201.25.181.173	do_setinputsizes	logs.amazonaws.com	firehose.amazonaws.com	ip_address	Unassigned	high
6	115.79.203.216	PoolEvents	lambda.amazonaws.com	states.amazonaws.com	ip_address	Unassigned	high

« ‹ 1 of 100 › »

Function Accessed by Highly Malicious Threat IPs

	functionname	unique_threat_ips
1	DialectEvents	284
2	set_engine_execution_options	283
3	do_executemany	247
4	before_execute	233
5	after_execute	225
6	before_create	214
7	make_url	211
8	release_savepoint	208
9	dbapi_error	205
10	after_create	202

All IP Threats Table

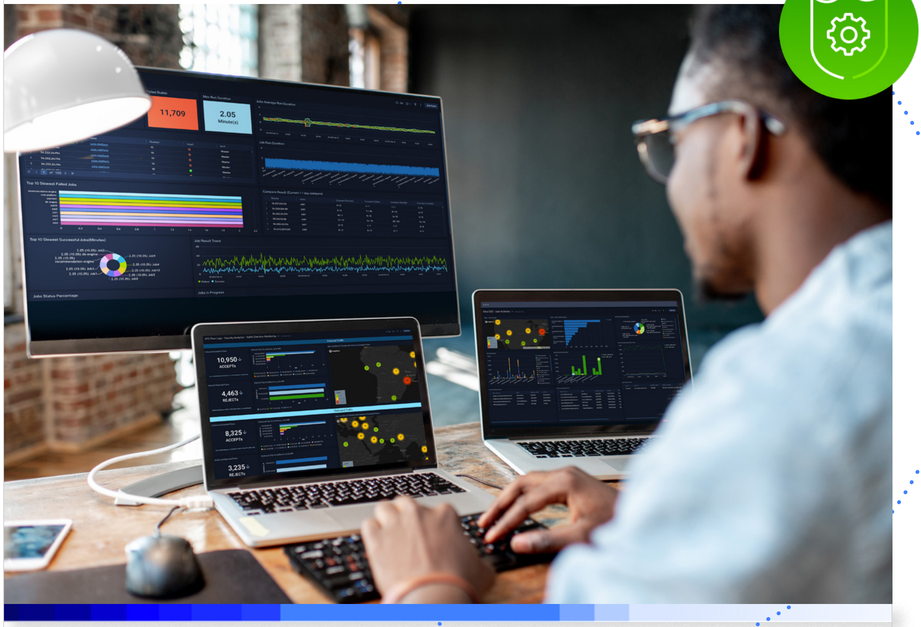
	src_ip	functionname	malicious_confidence	actor	label_name	threat_count
1	192.160.0.1	dispose	medium	Unassigned	[KillChain/C2 , MaliciousConfidence/Medium , Malware/njRAT , ThreatType/Commodity]	2
2	192.160.0.1	do_executemany	medium	Unassigned	[KillChain/C2 , MaliciousConfidence/Medium , Malware/njRAT , ThreatType/Commodity]	1
3	192.160.0.1	dbapi_error	medium	Unassigned	[KillChain/C2 , MaliciousConfidence/Medium , Malware/njRAT , ThreatType/Commodity]	3
4	192.160.0.1	column_reflect	medium	Unassigned	[KillChain/C2 , MaliciousConfidence/Medium , Malware/njRAT , ThreatType/Commodity]	1
5	192.160.0.1	handle_error	medium	Unassigned	[KillChain/C2 , MaliciousConfidence/Medium , Malware/njRAT , ThreatType/Commodity]	1

« ‹ 1 of 100 › »

sumo logic

Applications are critical to the success of any business

Keeping them secure takes work. In fact, over 40% of all breaches now occur via web application assets, and this percentage is rapidly increasing. Integrating security throughout the software development lifecycle is essential to address cybersecurity issues before they happen.



40%

of all breaches now occur via web application assets

Application security monitors a company's software offerings to ensure they are not vulnerable or infiltrated by malicious code at any point in the continuous integration/ continuous deployment (CI/CD) process and production cycle.

DevSecOps teams are often already incorporating solutions like code scanning, posture management and workload protection into their processes, but more is needed. Siloed solutions don't adequately address the modern developer's increasingly complex application security needs.

An IT team must see its full security posture to give applications the protection they require. These practices start by collecting security and event log data from every infrastructure, application and the network supporting the application.

Other components of the CI/CD lifecycle to consider when improving application security include:

Coding — Manage access to sources of code and building environments

Building and testing — Ensure code does not introduce vulnerabilities before it is executed

Deploying and running — Identify outage causes quickly to limit downtime

Monitoring — Examine employee access and overall application usage for ongoing threat detection

Sumo Logic offers full-stack visibility throughout the CI/CD lifecycle, so you can monitor security, increase reliability and speed up processes all in one place.

For a deeper dive on integrating security throughout the software development life cycle, check out our free Accelerate and secure your SDLC with DevSecOps ebook.

Coding

Code environment and repository access should be monitored from a central location, enabling DevSecOps teams to see data from all applications in one place for complete visibility. This enhances a team's capabilities to glean extra information from logs by adding contextual relationships to data and distinguishing static code.

Build tools and code repositories can become entry points for malicious code. With poisoned pipeline execution (PPE), attackers insert unauthorized code into these environments, which runs as part of the CI/CD lifecycle to infect the larger application. Visualizing who is in these environments, when and where they are accessing them, and what changes they are making helps identify problematic user behavior. Sumo Logic's consolidated log data uses additional context to enrich access information, helping identify unauthorized access and code insertion.

Building and testing

Quality assurance (QA) at the creation stage will improve the software development lifecycle by identifying any vulnerabilities before introducing new code. Reviewing output from test logs and building pipelines ensures greater compliance with product expectations over the entire application lifespan.

Third-party risks are also now inherent in code-writing for many app development processes. Studies have found that a majority of open-source code includes vulnerabilities, which then get baked into apps from the creation stage. Tools that allow these vulnerabilities to be detected can mitigate them, but monitoring these tools to know when and where they are detected takes time that many teams simply do not have. That's why combining data from all the different security tools is vital.

Deploying and running

Monitor the entire app script output to identify misconfigurations and determine their causes. Catching minor issues at their source and having complete visibility of the whole tech stack simplifies identifying if a cyberattack is the cause of an outage or if there is a different origin. The faster you make these determinations, the quicker any potential damage can be contained and repaired. This reduces app downtime and complexity needed in rebuilding affected technical assets.

Monitoring

The CI/CD pipeline is a point of vulnerability within a tech stack since it is the center of everyday development operations. Monitoring who has access to this sensitive area and managing the users within this space is vital. Access to pipeline components should be controllable by multiple factors for optimum security. These include:

→ **Role-based access allows staff with particular job descriptions to have access only to what they need.**

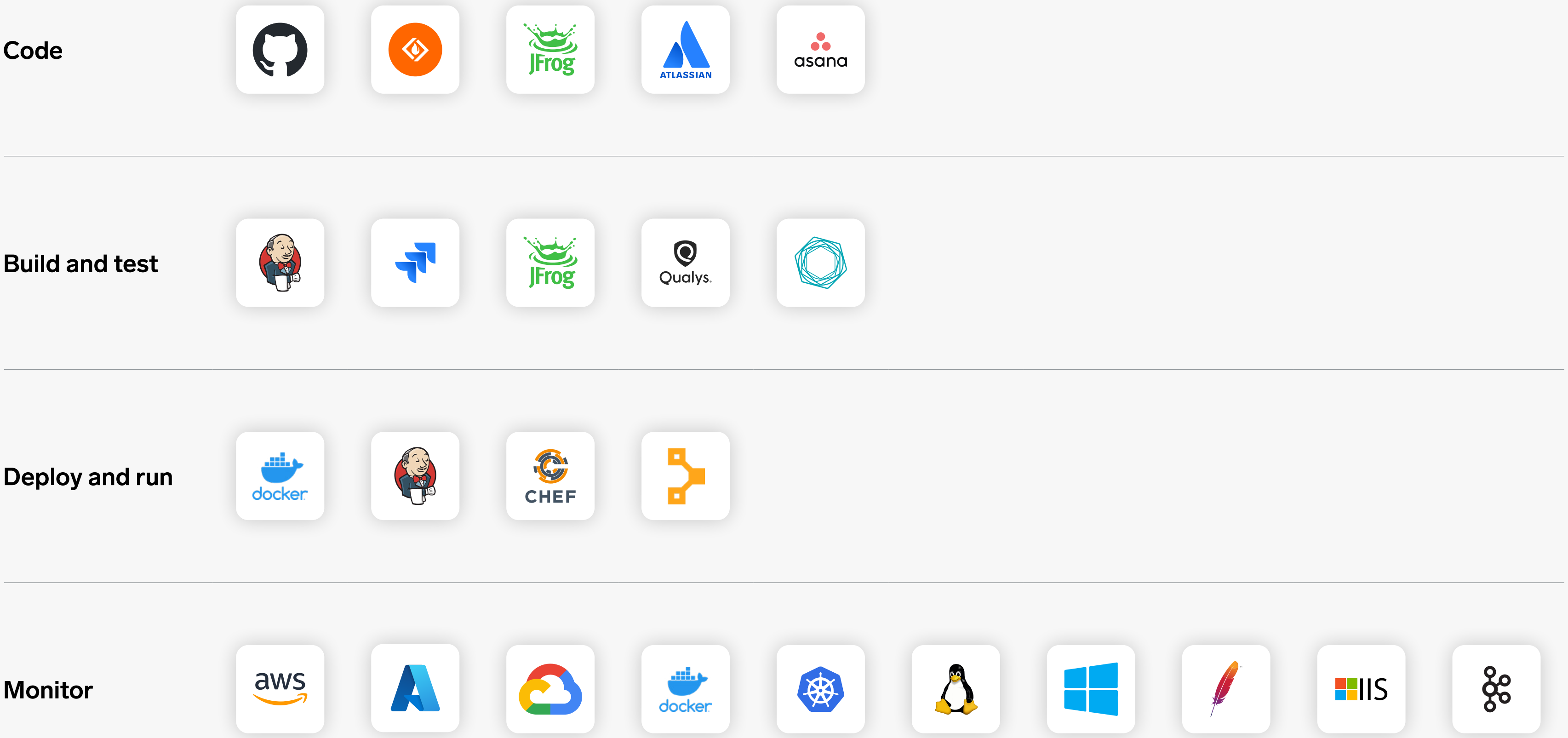
→ **Task-based access, so team members have access only when working on specific tasks.**

→ **Time-based access, so no one has access longer than they need it.**

→ **Pipeline-based access, so each pipeline only has access to necessary data sources.**

Regularly auditing access to all elements within the pipeline enhances app security.

Examples of integrations Sumo Logic supports across the CI/CD lifecycle



Code

Build and test

Deploy and run

Monitor

Learn more about Sumo Logic's hundreds of integrations supporting app development, security and dozens of other use cases here.

PUTTING IT ALL TOGETHER

A global bank accelerates delivery with Sumo Logic

A digital banking solution that operates in 31 countries with over 15,000 employees, wanted to improve its software delivery processes to take advantage of market opportunities and increase customer satisfaction. They implemented Sumo Logic's services to guide their strategy with data-driven insights.

With these goals in mind, the bank was able to:

Increase speed to 100 releases per day
Sumo Logic's platform enabled decision-makers to plan for and create a data-backed, accelerated, 100-per-day release cycle that improved application performance and customer satisfaction. Sumo Logic's elastic architecture and custom dashboards were well suited to analyze the large amount of data inherent to this release level.

Improve customer experience
The bank's DevOps team wanted more data on how their customers experienced the banking platform. Sumo Logic provided a suite of dashboards covering their online banking applications to provide measurable, real-time data on customer experience.

Implement multi-cloud strategy

The bank wanted to migrate its critical digital assets from 70+ small servers to multiple cloud-based environments. The assets moved processed 2,000 digital banking transactions per second. According to the bank's head of engineering the project "was 100% smooth with no issues."

Develop new integration

Teams at the bank used Facebook Workplace to collaborate, so their DevOps team leveraged Sumo Logic's API to ensure insights from Sumo Logic dashboards are shared effortlessly in Workplace for staff discussions.

Sumo Logic unlocks business potential in the following ways:

100

releases per day

2000

digital banking transactions per second

Multi-cloud

moved 70+ servers to multi-cloud environments

To learn more about this case study, find the details here.

Ready to dive in?

Your step-by-step guide to implementing application security best practices

Monitoring app security can be time-consuming and challenging without the right tools. Even in small organizations, it's common to run at least 20 security applications to protect valuable assets. But manually monitoring all the data sources is virtually impossible without full-stack visibility. Having the complete picture of security threats and vulnerabilities in real-time means teams aren't slowed down by manually checking each area individually.

What are the basic process steps involved in application security? On the following page is a list of some common best practices, along with where Sumo Logic's platform can help implement them.

1

Build threat modeling into every app development

Monitor security tool output throughout the development process with Sumo Logic’s visualizations.

2

Apply security to each component within every application

This granular level of protection allows for greater coverage than implementing broader, less precise measures. Bring detailed security data into Sumo Logic’s easy-to-use platform to get the most benefit from it.

3

Automate security installations and configurations

To avoid missing vital coverage due to human error. Sumo Logic promotes DevSecOps best practices with unified log management and analysis to best support your team.

4

Focus on unique value adds and outsource other applications

Let the DevSecOps team devote their time to creating and protecting applications only their business can provide. Sumo Logic’s extensive integrations mean our platform can monitor many security tools you already use.

5

Assume infrastructure is insecure and protect applications accordingly

Cloud, hybrid and even on-premises servers are also susceptible to threats, so it’s important to design systems that protect against infrastructure vulnerabilities.

6

Monitor consistently

With cybersecurity breaches, it is now a matter of when, not if, a business will fall victim to an attack. Never assume applications are fully protected. Never stop monitoring.

7

Use benchmarks to know where application security stands

Benchmark against industry standards and other internal software protection levels.

Secure and reliable apps at your fingertips

Our cloud-native platform is designed for the modern IT landscape with the flexibility and convenience you need. The user-friendly interface makes navigation quick and easy for your security and development teams to integrate into their workflows. Analytics help reduce security breaches; those that occur are quickly found and mitigated.

Application security is an important use case of overall Cloud Security Analytics, the ideal operating posture for managing data. In addition to **application security** discussed here, Sumo Logic Cloud Security Analytics cover other critical use cases that include:

→ **Security data lake —** To store and manage information

→ **Audit and compliance —** To meet security regulations and follow best practices

→ **Threat detection and investigation —** To identify problems quickly

When you're ready to take your application security to the next level, contact us or start your free 30-day trial.

Sumo Logic.
The infinite power of log analytics.