new relic

DATA SHEET

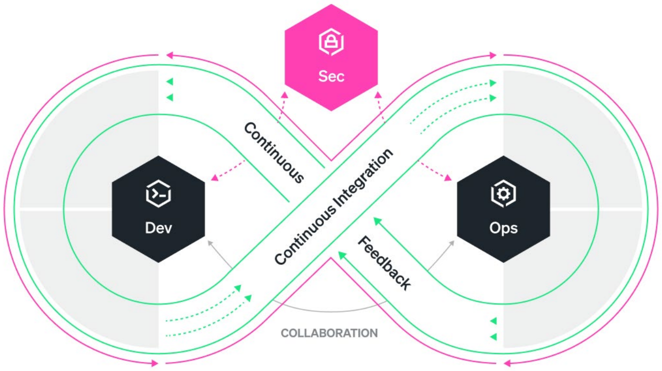
New Relic Interactive Application Security Testing (IAST)

Accurate, fast IAST. Ship code faster with unmatched detection accuracy of security risks.

Today's application security testing practices are inaccurate and disjointed, resulting in false positives, missed release cycles, and increased security costs. To build more secure applications DevOps teams need a solution that provides complete visibility across the application lifecycle, eliminates false positives, and makes it easy to detect and fix real security risks.

New Relic IAST goes beyond current approaches, providing visibility and context to security findings, unmatched detection accuracy, and proof of exploit via dynamic assessment capabilities that pinpoint the source of vulnerabilities by simulating real-world attacks—with guided remediation for faster resolution. In addition, New Relic IAST is fully integrated with New Relic Vulnerability Management, enabling DevOps teams to continuously find, fix, and verify high-risk vulnerabilities across the software developer lifecycle (SDLC).

As part of the New Relic observability platform, New Relic IAST enables DevOps and security teams to accurately and continuously monitor, test, and remediate security risks across the SDLC at scale, and ship code faster.



BENEFITS

See everything and eliminate blind spots

Gain visibility into the entire application stack and associated relationships with context-driven insights to eliminate blind spots and validate the status of remediation efforts.

Enhance security testing accuracy

Eliminate false positives with fast, accurate detection, risk-based prioritization, and automated vulnerability validation.

Focus on what matters with proof-based exploit validation

Find, fix, and verify exploitable vulnerabilities for faster remediation with dynamic assessment capabilities that eliminate the need for code changes.

Accelerate remediation efforts

Guided remediation and guardrails enable developers to avoid critical mistakes by pinpointing code location, stack trace, HTTP trace, encountered URLs, and exploit mechanism and parameters, and more.

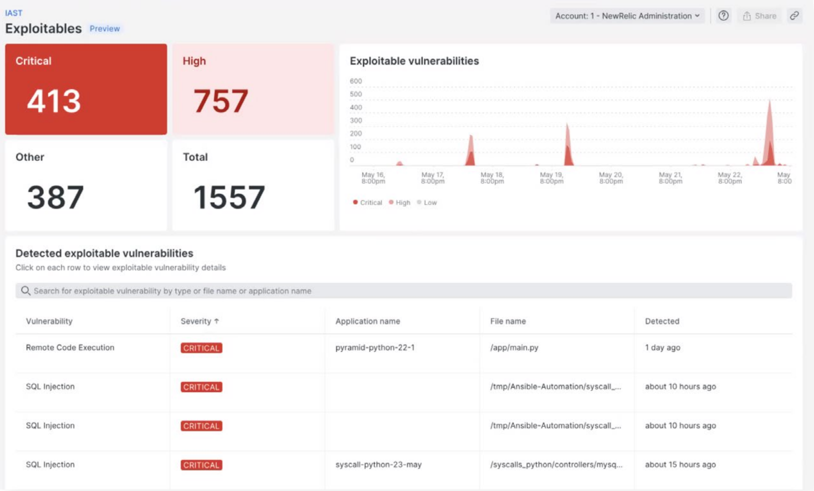
Scale at will

Easy deployment via existing APM agent and seamless integration with continuous integration and continuous delivery (CI/CD) pipelines and ticketing systems helps prevent disruption of existing processes and workflows.

CAPABILITIES

Gain visibility into the entire application stack

See all protected and unprotected applications to eliminate blind spots and pinpoint hidden threats, and continuously monitor and validate the status of remediation efforts to ensure applications are always protected.



Critical	High
413	757
Other	**Total**
387	1557

Exploitable vulnerabilities

Detected exploitable vulnerabilities

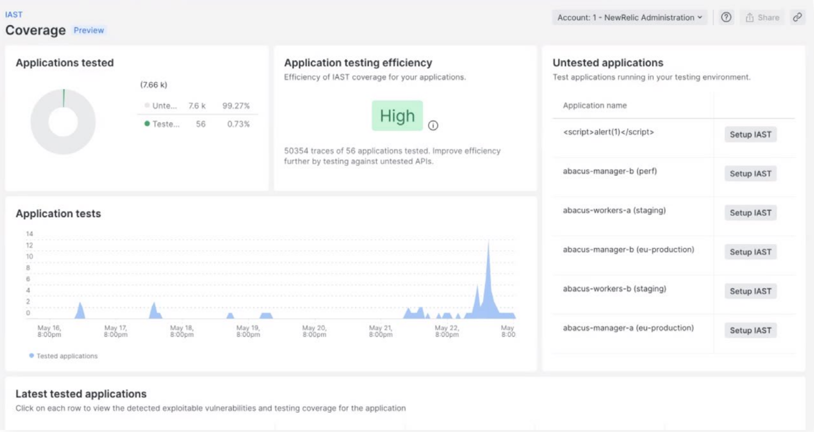
Click on each row to view exploitable vulnerability details

Search for exploitable vulnerability by type or file name or application name

Vulnerability	Severity ↑	Application name	File name	Detected
Remote Code Execution	CRITICAL	pyramid-python-22-1	/app/main.py	1 day ago
SQL Injection	CRITICAL		/tmp/Ansible-Automation/syscall_...	about 10 hours ago
SQL Injection	CRITICAL		/tmp/Ansible-Automation/syscall_...	about 10 hours ago
SQL Injection	CRITICAL	syscall-python-23-may	/syscalls_python/controllers/mysq...	about 15 hours ago

Pinpoint exactly where vulnerabilities exist in real time with near zero false positives

Identify vulnerabilities across all layers of the application stack and reduce false positives with fast, accurate detection, enabling developers to focus on real security threats with risk-prioritized vulnerability lists for faster remediation.



Applications tested

(7.66 k)

● Unte...	7.6 k	99.27%
● Teste...	56	0.73%

Application testing efficiency

Efficiency of IAST coverage for your applications.

High

50354 traces of 56 applications tested. Improve efficiency further by testing against untested APIs.

Untested applications

Test applications running in your testing environment.

Application name	
<script>alert(1)</script>	Setup IAST
abacus-manager-b (perf)	Setup IAST
abacus-workers-a (staging)	Setup IAST
abacus-manager-b (eu-production)	Setup IAST
abacus-workers-b (staging)	Setup IAST
abacus-manager-a (eu-production)	Setup IAST

Application tests

Latest tested applications

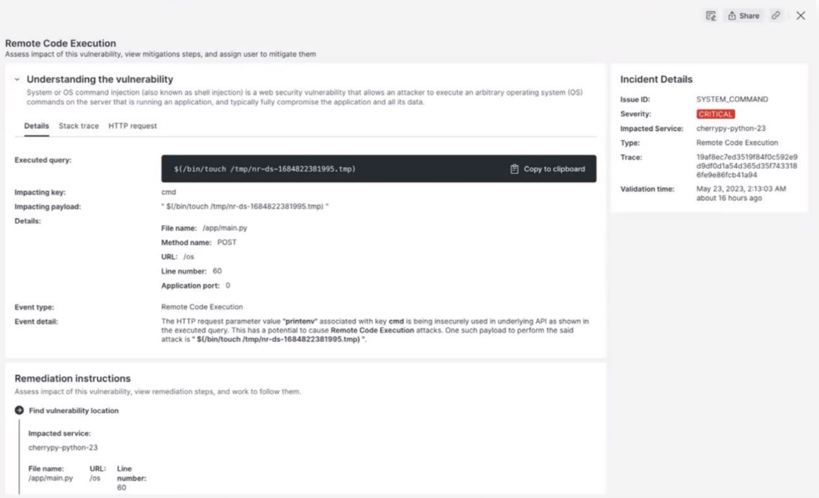
Click on each row to view the detected exploitable vulnerabilities and testing coverage for the application

> “New Relic IAST empowers our developers to code with confidence by automating work and providing a comprehensive view of security risks, including real-time feedback, accuracy, and context-aware security analysis—all in the context of our observability practice and without impeding the development process.”

Agustín Paroli
Head of IT Operations at D24

Identify vulnerabilities with proof of exploit

Save time with dynamic assessment capabilities that identify the source of vulnerabilities by simulating real-world attacks. Then validate them with proof of exploit so that developers can focus on verified vulnerabilities and ship more secure code.



Remote Code Execution
Assess impact of this vulnerability, view mitigations steps, and assign user to mitigate them

Understanding the vulnerability
System or OS command injection (also known as shell injection) is a web security vulnerability that allows an attacker to execute an arbitrary operating system (OS) commands on the server that is running an application, and typically fully compromise the application and all its data.

Executed query:

```
$(/bin/touch /tmp/nr-ds-1684822381995.tmp)
```

Impacting key: cmd
Impacting payload: " $(/bin/touch /tmp/nr-ds-1684822381995.tmp) "
Details:
File name: /app/main.py
Method name: POST
URL: /os
Line number: 60
Application port: 0

Event type: Remote Code Execution
Event detail: The HTTP request parameter value **"printenv"** associated with key **cmd** is being insecurely used in underlying API as shown in the executed query. This has a potential to cause **Remote Code Execution** attacks. One such payload to perform the said attack is " **$(/bin/touch /tmp/nr-ds-1684822381995.tmp)** ".

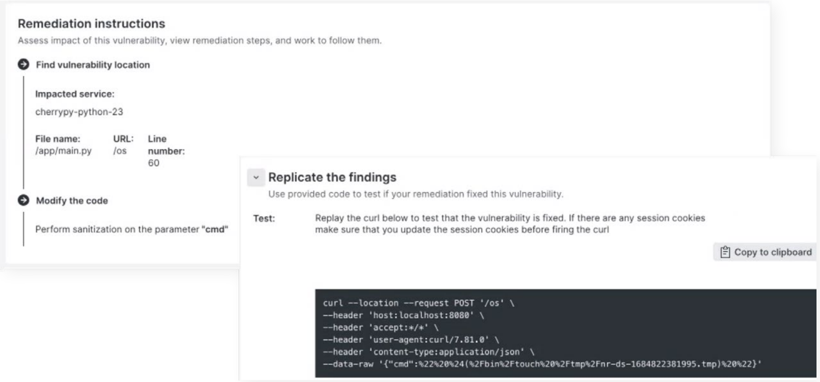
Remediation instructions
Assess impact of this vulnerability, view remediation steps, and work to follow them.

Find vulnerability location
Impacted service:
cherrypy-python-23
File name: /app/main.py **URL:** /os **Line number:** 60

Incident Details
Issue ID: SYSTEM_COMMAND
Severity: CRITICAL
Impacted Service: cherrypy-python-23
Type: Remote Code Execution
Trace: 19af8ec7ed3519f84f0c592e9d9df0d1a54d365d35f7433186fe9e86fcb41a94
Validation time: May 23, 2023, 2:13:03 AM about 16 hours ago

Guided remediation for fast, effective elimination of security risks

Automatically prioritize software flaws like SQL injection, command execution, and other OWASP Top 10 standards, and then eliminate them before they can be exploited. With guided remediation and guardrails from New Relic experts, developers can avoid critical mistakes that could lead to a potential security incident.



Remediation instructions
Assess impact of this vulnerability, view remediation steps, and work to follow them.

Find vulnerability location
Impacted service:
cherrypy-python-23
File name: /app/main.py **URL:** /os **Line number:** 60

Modify the code
Perform sanitization on the parameter **"cmd"**

Replicate the findings
Use provided code to test if your remediation fixed this vulnerability.

Test: Replay the curl below to test that the vulnerability is fixed. If there are any session cookies make sure that you update the session cookies before firing the curl

```
curl --location --request POST '/os' \
--header 'host:localhost:8080' \
--header 'accept:*/*' \
--header 'user-agent:curl/7.81.0' \
--header 'content-type:application/json' \
--data-raw '{"cmd":%22%20%24(%2Fbin%2Ftouch%20%2Ftmp%2Fnr-ds-1684822381995.tmp)%20%22}'
```

Next Steps

- Read More
- Get Started
- Contact Us

new relic.