



Building secure player experiences at Riot Games

David Rook

1

Senior Security Engineer at Riot Games

2

Leading Riots global Application Security program

3

Also lead Riots bug bounty program

4

12 years Information Security. 10 full time Applicaton Security

5

Gamer (but not a good one!)









Application Security and Riot Engineering



Automation



Rioters



Bug Bounty







SKT 0

2016 WORLD CHAMPIONSHIP FINALS

	Kills	Deaths	Assists	Items	Gold
CROWN	0/1/0	2/1	1/1	1/1	1000
FAKE	0/1/0	1/2	1/1	1/1	1000

WORLDS PATCH 6.18

FAKE

STAPLES CENTER
OCT 29 @ WORLDS









GENE SISKEL THEATRE

JOINT at&t

2016
LEAGUE OF LEGENDS
WORLD CHAMPIONSHIP








CHASE
CHICAGO

BJERGSEN
logitech
htc
CYBERPOWERPC
GEICO

SUBWAY





 no-unsafe-innerhtml.js	Validation check added
 no-unsafe-script-innertext.js	Validation check added
 no-unsafe-script-src.js	Validation check added
 no-unsafe-script-text.js	Validation check added
 no-unsafe-script-textcontent.js	Validation check added
 no-unsafe-settimeout.js	New rules!
 no-unsafe-write.js	New rules!


```
* Cloning gh.riotgames.com:appsec/sightstone to /Users/drook/Documents/GH_scan_results/appsec-sightstone ( 1 Files )...  
  * Clone Complete, Starting ESLint Scan...  
    * Scan Complete, Found 1 Issues  
    * Writing Report to /Users/drook/Documents/GH_scan_results/appsec-sightstone-report.txt...
```





AppSec Slack Bot BOT 3:02 PM

Your new repo riotclient-data-mocking-rso-login looks like it's using NodeJS

We think these security resources are awesome for helping engineers build secure Node products:

<http://expressjs.com/en/advanced/best-practice-security.html>

<https://nodesecurity.io/>

<https://github.com/helmetjs>

If you'd like to speak to an AppSec Engineer for more detailed advice please reach out to us via appsecv@riotgames.com or the #ask-infosec Slack channel



Slack API Tester APP 4:27 PM

Hey drook, I've noticed that your repository at /Users/drook/Documents/Scanner/src/RiotGames.AppSec.Scanner/my-repository uses the following dependencies that have been identified as having vulnerabilities. They are as follows:

node-serialize 0.0.3

Code Execution through IIFE (<https://nodesecurity.io/advisories/311>)

serialize-to-js 0.4.0

Code Execution Through IIFE (<https://nodesecurity.io/advisories/313>)

You're being messaged because you are the last committer to have submitted a commit to this project.

M My Team

R Riot Games AppSec

- Reports
- Agents >
- User Management >
- Settings >
- Manage Subscription
- Contact Support

+ Create new team

← Team reports



Issues 6

Security 5

Libraries 22

Licenses 4

Vulnerabilities 5

Updates 1

Licenses 0

Search Vulnerabilities

5 vulnerabilities

Direct only High risk only Has vulnerable methods

Create Issue

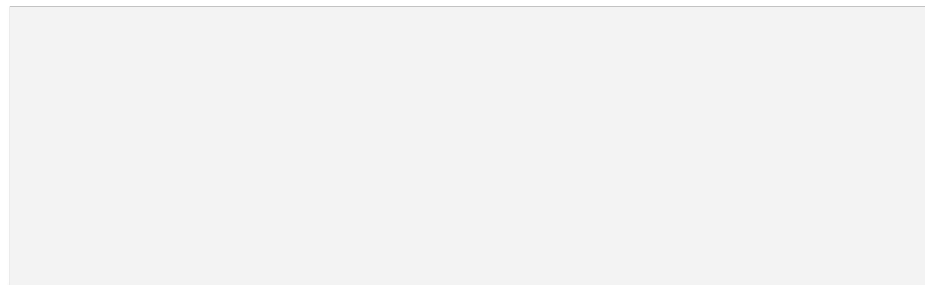
Ignore



Scan	Vulnerability	Library	Version	Severity	Vulnerable Method	Reference	Branch/Tag	Select
969993	Timing Attack	sinatra	1.4.5	Medium Risk	--	No CVE - Premium Data	master	<input type="checkbox"/>
969993	Side-Channel Attack	sinatra	1.4.5	Low Risk	--	No CVE - Premium Data	master	<input type="checkbox"/>
969993	Reflected Cross-Site Scripting (XSS)	sinatra	1.4.5	Medium Risk	--	No CVE - Premium Data	master	<input type="checkbox"/>
969993	Denial of Service (DoS)	rack	1.5.2	Medium Risk	--	CVE-2015-3225	master	<input type="checkbox"/>
969993	IP Spoofing Attack via HTTP Request Header	rack	1.5.2	Medium Risk	--	No CVE - Premium Data	master	<input type="checkbox"/>



Mobile Store

[Overview](#)[Engagements](#)[Repositories](#) 2[Environments](#) 2[Settings](#)[New Review](#)

⚡ Technologies (6)

Spring

Framework

Java

Language

JavaScript

Language

Android

Operating System

iOS

Operating System

Apache HTTPD

Web Server

🔍 Issues

There are no issues.

📄 Metadata

Technical Contact

Audience

Players

Lives where

AWS

Internet accessible

True

VPN

True

Lines of Code

10000

Views

1-10

Inputs

1-10

Platform

Mobile

Lifecycle

Validate

Origin

Internally Developed

User Records

None

LoC

10000

Views

1-10

Inputs

1-10

Mobile Store

 Overview

 Engagements

 Repositories **2**

 Environments **2**

 Settings

Internal GitHub

Internal GitHub

Mobile Store

[🏠 Overview](#)[📅 Engagements](#)[📖 Repositories 2](#)[🌐 Environments 2](#)[⚙️ Settings](#)[+ New Environment](#)

Development Environment

[👍 Approved for Security Testing](#)

Korea3

[✎ Edit Environment](#)[👤 Add Credentials](#)

Pre-Production Environment

[👍 Approved for Security Testing](#)

Korea2

[✎ Edit Environment](#)[👤 Add Credentials](#)

Mobile Store

[Overview](#)[Engagements](#)[Repositories](#) 2[Environments](#) 2[Settings](#)

i Pending Engagement requests that haven't been approved

There are no pending engagements.

i Approved Engagements that have been approved

There are no approved engagements.

i Open Engagements that are in progress

There are no open engagements.

✓ Closed Engagements that have been completed

Feb. 10, 2017 - Feb. 15, 2017

i Security Review





Application Security

We aim to arm every software engineer with the tools and knowledge they need to build safe and secure experiences for Players and Rioters

We respect the Do-Not-Disturb Signs

Owners are responsible for ensuring that the product produced by the team is of a high quality

We will make sure to provide a day for the team so that we can be one another

We have biweekly 1:1s with each other if there is a reasonable agenda

DAN







Web Application Security Checklist

Validation of Data

- Are you using a [whitelisting](#) approach when validating all player supplied data?
- Are you validating all player supplied URLs?
- When you display player supplied data do you use the correct output encoding to avoid Cross Site Scripting vulnerabilities?
- When you are interacting with a SQL database are you using prepared statements to avoid [SQL Injection](#) vulnerabilities?
- When you display player supplied data do you use the correct output encoding to avoid [Cross Site Scripting](#) vulnerabilities?

Cryptography

- If you are storing player data have you implemented the best practices outlined in [RFC 157](#)?
- Where passwords are used are you securely transmitting and storing them?
- Does your TLS configuration implement the best practices outlined in [RFC 0157a](#)?

Access Control

- If you have an external facing API have you implemented rate limiting and authentication?
- Are admin panels only accessible from whitelisted IP's?
- Are you using session specific tokens for form submissions to prevent [Cross Site Request Forgery](#)?
- Do players authenticated session automatically expire after a certain period of idle time?
- Have you implemented brute force protections for all authentication points?

Validation of Data

SQL

Using Django's `querysets`, the resulting SQL will be properly escaped by the underlying database driver. In addition, Django also gives developers power to write raw SQL queries.

If you absolutely must write raw SQL for your application make sure you use parameterized queries to avoid potential SQL Injection vulnerabilities.

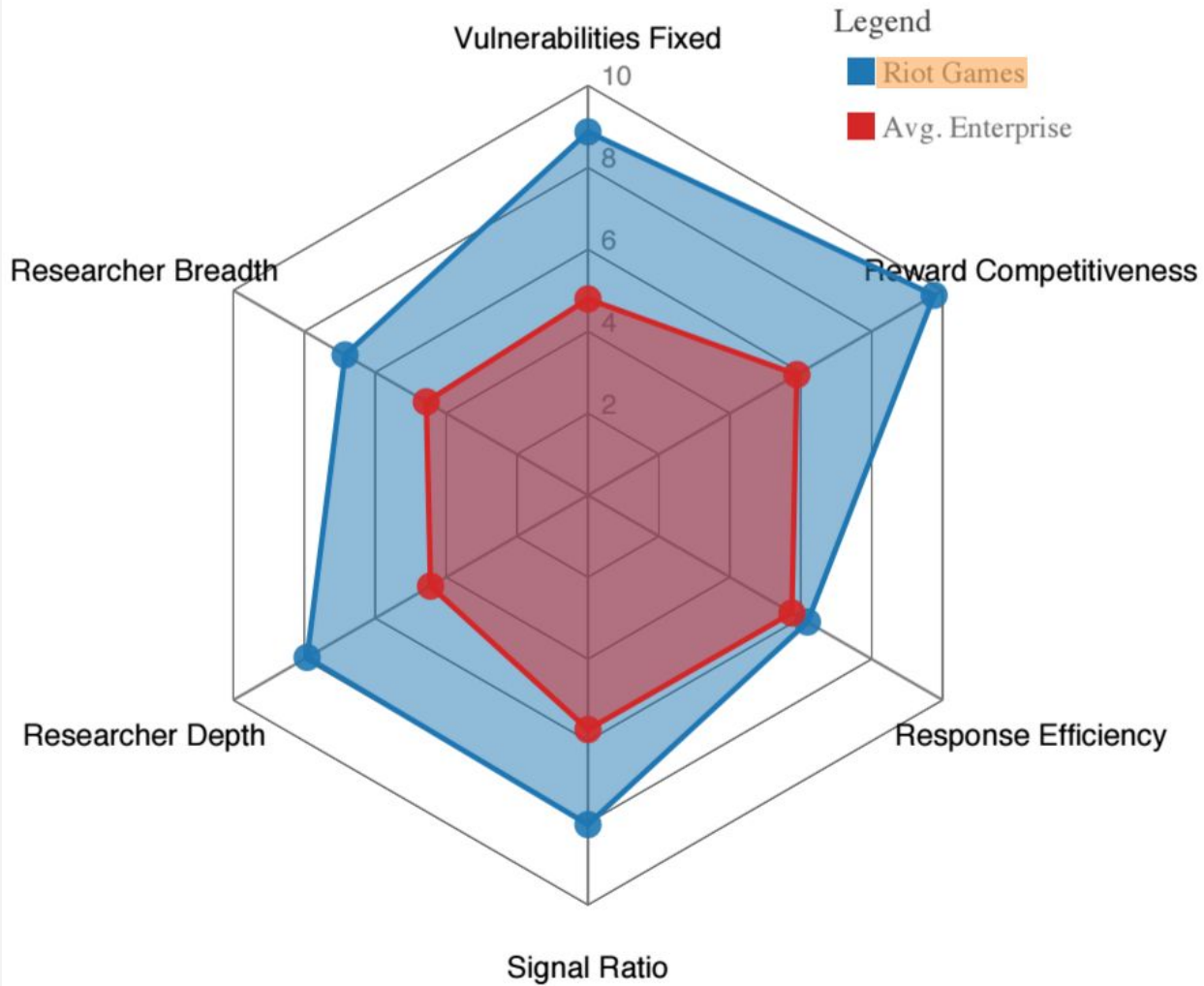
Good Example:

```
Person.objects.raw('SELECT * FROM myapp_person WHERE last_name = %s', [lname])
```

Bad Example:

```
Person.objects.raw('SELECT * FROM myapp_person WHERE last_name = ' + lname)
```









Questions?

@davidrook