Building a better database

How GitHub Structures their Advisory Database to Drive Developer Outcomes

@darakian - 25th March 2024
Who am I?
Hi, I’m Jon (aka Darakian) 👋

• GitHub since 2021
• Security since 2018
• Tech since 2008
• Nerd since time immemorial
• https://github.com/darakian/
We are the GitHub Security lab
We want to secure your software

GitHub Security Lab

Securing the world's software, together

GitHub Security Lab’s mission is to inspire and enable the community to secure the open source software we all depend on.

Follow @GHSecurityLab
What do I do?

I’m a librarian 📚

- I curate advisories for the advisory database
- Ensure correctness, completeness and conciseness
- Quality over quantity
- https://github.com/advisories
- https://github.com/github/advisory-database
First and foremost what are we doing here?
Fixing code!

- People don’t want to ship vulnerable code
- People don’t want their time wasted by fuzzy matching
- People do understand that vulns are nuanced
- People do appreciate a best effort job when the effort is obvious
- People will forgive you for (understandable) mistakes
First and foremost what are we doing here?

What are good outcomes?

- Minimize time from public disclosure to code fix deployed
- Minimize false positives to save developer time
- Maximize utility per advisory
- Maximize human curation impact
- Avoid the tar pit
The developer perspective

but which ones do I care about?

Surely not all of them
**Scope**

*We can do anything so long as we don’t do everything*

---

### Supported ecosystems

Unfortunately, we cannot accept community contributions to advisories outside of our supported ecosystems. Our curation team reviews each community contribution thoroughly and needs to be able to assess each change.

Generally speaking, our ecosystems are the namespace used by a package registry. As such they’re focused on packages within the registry which tend to be dependencies used in software development.

Our supported ecosystems are:

- Composer ([registry](https://packagist.org))
- Erlang ([registry](https://hex.pm))
- GitHub Actions ([registry](https://github.com/marketplace?type=actions))
- Go ([registry](https://pkg.go.dev))
- Maven ([registry](https://repo.maven.apache.org/maven2))
- npm ([registry](https://www.npmjs.com/))
- NuGet ([registry](https://www.nuget.org))
- pip ([registry](https://pypi.org))
- Pub ([registry](https://pub.dev))
- RubyGems ([registry](https://rubygems.org))
- Rust ([registry](https://crates.io))
- Swift ([registry](namespaced by dns))

If you have a suggestion for a new ecosystem we should support, please open an issue for discussion.
Scope

We can do anything so long as we don’t do everything

Supported ecosystems

Unfortunately, we cannot accept community contributions to advisories outside of our supported ecosystems. Our curation team reviews each community contribution thoroughly and needs to be able to assess each change.

Generally speaking, our ecosystems are the namespace used by a package registry. As such they’re focused on packages within the registry which tend to be dependencies used in software development.

Our supported ecosystems are:

- Composer (registry: https://packagist.org)
- Erlang (registry: https://hex.pm/)
- GitHub Actions (registry: https://github.com/marketplace?type=actions)
- Go (registry: https://pkg.go.dev)
- Maven (registry: https://repo.maven.apache.org/maven2)
- npm (registry: https://www.npmjs.com/)
- NuGet (registry: https://www.nuget.org/)
- pip (registry: https://pypi.org/)
- Pub (registry: https://pub.dev/)
- RubyGems (registry: https://rubygems.org/)
- Rust (registry: https://crates.io/)
- Swift (registry: namedpaced by dns)

If you have a suggestion for a new ecosystem we should support, please open an issue for discussion.
CPE is not bijective

Four different roots

**Known Affected Software Configurations**

**Configuration 1**

- **cpe:2.3:a:nodejs:undici:*:*:*:*:node.js:*:***
  - Up to (excluding)
  - 4.17.5

**Configuration 2**

- **cpe:2.3:a:nodejs:undici:*:*:*:*:node.js:*:***
  - Up to (excluding)
  - 5.26.2

**Configuration 3**

- **cpe:2.3:a:npmsjs:semver:*:*:*:*:node.js:*:***
  - From (including)
  - To (excluding)
  - 5.7.2

- **cpe:2.3:a:npmsjs:semver:*:*:*:*:node.js:*:***
  - From (including)
  - To (excluding)
  - 6.0.0

- **cpe:2.3:a:npmsjs:semver:*:*:*:*:node.js:*:***
  - From (including)
  - To (excluding)
  - 7.0.0

**Configuration 4**

- **cpe:2.3:a:mathjs_project:mathjs:*:*:*:*:mathjs:*:***
  - Up to (excluding)
  - 3.17.0


[https://nvd.nist.gov/vuln/detail/CVE-2023-45143](https://nvd.nist.gov/vuln/detail/CVE-2023-45143)


## CPE is not bijective

Index for the benefit of the scanner

### Known Affected Software Configurations

<table>
<thead>
<tr>
<th>Configuration 1 (hide)</th>
<th>Package</th>
<th>Affected versions</th>
<th>Patched versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpe:2.3:a:lodash:lodash:<em>:</em>:<em>:node.js:</em>:*</td>
<td>lodash (npm)</td>
<td>&lt; 4.17.5</td>
<td>4.17.5</td>
</tr>
</tbody>
</table>

https://github.com/advisories/GHSA-fvqr-27wr-82fm

<table>
<thead>
<tr>
<th>Configuration 1 (hide)</th>
<th>Package</th>
<th>Affected versions</th>
<th>Patched versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpe:2.3:a:npmjs:semver:<em>:</em>:<em>:node.js:</em>:*</td>
<td>undici (npm)</td>
<td>&lt; 5.26.2</td>
<td>5.26.2</td>
</tr>
</tbody>
</table>

https://github.com/advisories/GHSA-wqq4-5wpv-mx2g

<table>
<thead>
<tr>
<th>Configuration 1 (hide)</th>
<th>Package</th>
<th>Affected versions</th>
<th>Patched versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpe:2.3:a:npmjs:semver:<em>:</em>:<em>:node.js:</em>:*</td>
<td>semver (npm)</td>
<td>&gt;= 7.0.0, &lt; 7.5.2</td>
<td>7.5.2</td>
</tr>
<tr>
<td>cpe:2.3:a:npmjs:semver:<em>:</em>:<em>:node.js:</em>:*</td>
<td>semver (npm)</td>
<td>&gt;= 6.0.0, &lt; 6.3.1</td>
<td>6.3.1</td>
</tr>
<tr>
<td>cpe:2.3:a:npmjs:semver:<em>:</em>:<em>:node.js:</em>:*</td>
<td>semver (npm)</td>
<td>&lt; 5.7.2</td>
<td>5.7.2</td>
</tr>
</tbody>
</table>

https://github.com/advisories/GHSA-c2qf-rxjj-qgw

<table>
<thead>
<tr>
<th>Configuration 1 (hide)</th>
<th>Package</th>
<th>Affected versions</th>
<th>Patched versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpe:2.3:a:mathjs:mathjs:<em>:</em>:<em>:node.js:</em>:*</td>
<td>mathjs (npm)</td>
<td>&lt; 3.17.0</td>
<td>3.17.0</td>
</tr>
</tbody>
</table>

https://github.com/advisories/GHSA-pv8x-p9hq-j328
Anatomy of an attack
A chain of supply
Anatomy of an attack
A web of supply
Anatomy of an attack
Where bugs manifest in the web
Anatomy of an attack
The alert chain
Signal to noise
Alert fatigue is the enemy
Signal to noise
What are we doing here?

Description
marcor package in PyPi 0.1 through 0.13 included a code-execution backdoor.

Severity
CVSS Version 3.1
Base Score: 9.2 CRITICAL

CVSS Version 2.0
Base Score: 6.8
Vector: AV:N/AC:L/AV:N/AC:L/AV:N/AC:L/AU:N/CI:N/II:N/A:N

Note: NVD Analysts use publicly available information to associate vector strings and CVSS scores. We also display any CVSS information provided within the CVE List from the CNA.

References to Advisories, Solutions, and Tools
By selecting these links, you will be leaving NIST webspace. We have provided these links to other web sites because they may have information that would be of interest to you. No inferences should be drawn on account of other sites being referenced, or not, from this page. There may be other web sites that are more appropriate for your purpose. NIST does not necessarily endorse the views expressed, or concurs with the facts presented on these sites. Further, NIST does not endorse any commercial products that may be mentioned on these sites. Please address comments about this page to nvd@nist.gov.

<table>
<thead>
<tr>
<th>Hyperlink</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://pyppi.dobabino.com/simple/request">http://pyppi.dobabino.com/simple/request</a></td>
<td>Third Party Advisory</td>
</tr>
<tr>
<td><a href="https://github.com/joaofreitas/marcador/issues/5">https://github.com/joaofreitas/marcador/issues/5</a></td>
<td>Issue Tracking</td>
</tr>
<tr>
<td><a href="https://pyppi.org/project/marcador/">https://pyppi.org/project/marcador/</a></td>
<td>Third Party Advisory</td>
</tr>
</tbody>
</table>

Signal to noise
What are we doing here?

Weakness Enumeration

<table>
<thead>
<tr>
<th>CWE-ID</th>
<th>CWE Name</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVD-CWE-noinfo</td>
<td>Insufficient Information</td>
<td>NIST</td>
</tr>
</tbody>
</table>

Known Affected Software Configurations

**Configuration 1**

- **CPE:** cpe:2.3:a:python:pypi:*:*:*:*:*:*:*:*
  - From (including): 0.1
  - Up to (including): 0.13

Signal to noise
What are we doing here?

code execution backdoor #5

We found a malicious backdoor in versions 0.1~0.13 of this project, and its malicious backdoor is the request package. Even if the request package was removed by pypi, many mirror sites did not completely delete this package, so it could still be installed. When using pip3 install marcador==0.13 [http://pypi.doubanio.com/simple --trusted-host pypi.doubanio.com], the request malicious plugin can be successfully installed.

https://github.com/joajfreitas/marcador/issues/5
Signal to noise
What are we doing here?

Just so I understand. You are using the http://pypi.doubanio.com/simple mirror when installing marcador. The malicious package is present in this mirror? Is it also present in the official pypi mirrors?

I see no problem in removing those versions from pypi just want to understand a bit better the thread model here :)

request package was removed by pypi, many mirror sites did not completely delete this package, so it could still be installed. When using pip3 install marcador==0.13[ http://pypi.doubanio.com/simple --trusted-host pypi.doubanio.com], the request malicious plugin can be successfully installed.
We cannot expect better unless we enable better
How do we do that?
Where are we in the web?

- Our readers: Developers
  - We can inspect their source code
  - We cannot inspect their build systems
  - We cannot inspect their deploy environment
How do we do that?
What tools are at hand?

• We have source code to inspect
  • Manifest and lock files are common
  • We can know exactly what packages get used
  • We can know roughly what versions get used
  • We can model this in a database and index security claims
How do we do that?

The plan

• Index vulns on packages
  • Zero false positives by design
  • Quick delivery in the developer workflow
  • Common language to discuss vulns
• Security claims are verifiable
• Security claims are contestable
Anatomy of an advisory
The alert pipe
Anatomy of an advisory
The alert pipe
Anatomy of an advisory

The alert pipe
Anatomy of an advisory
And it scales
Alerts must flow
Fastest patch in the west! Other geos also supported!
Fastest patch in the west! other geos also supported!

Log4J 2.x - hourly downloads after update released

https://www.sonatype.com/resources/log4j-vulnerability-resource-center
The long view
Open source

It’s a trust building exercise
We are not perfect nor do we pretend to be

Does the advisory database cover other maven repositories? #2900

[joshbressers](https://github.com/joshbressers) opened this issue on Oct 31, 2023

As best as I can tell, most of the current Java packages cover Maven Central and not other maven repositories

For example the Atlassian maven repo
https://packages.atlassian.com/content/repositories/atlassian-public/com/atlassian/
contains confuence Java packages where Maven Central does not
https://repo.maven.apache.org/maven2/com/atlassian/

If we look at the MVN Repository site, we can see the top maven repositories
https://mavenrepository.com/repos
(there are shockingly more of these than I expected)

Thanks in advance

[darakian](https://github.com/darakian) on Oct 31, 2023

The short answer is *sorta*. As of today our data should be considered to refer to objects on maven central only and if the package names and versions happen to be useful when read in the context of another registry then that's a happy accident. Longer term we’ve got a conversation going with OSV here [osf/osv-schema#208](https://github.com/osf/osv-schema/pull/208) on how to properly address the data which is happy accident today,
We get it wrong but we make it right

[GHSA-f8vr-r385-rh5r] hyper and h2 vulnerable to denial of service #2057

seanmonstar commented on Apr 12, 2023

Updates

- Affected products
- Description
- Source code location

Comments

The code lives in the h2 library, a fix will only require a new h2 version. A hyper version will not need to be published, since hyper’s dependency range allows the new version to be used once it exists.

https://github.com/github/advisory-database/pull/2057
We can’t do it all

The siren song of the tar pit

Unable to improve advisory database for C / C++ packages #2963

maswilson opened this issue on Nov 21, 2023

Indeed it is documented in the README that contributions are not accepted for advisories outside the supported ecosystems. But some of the most high-impact vulnerability bulletins that need improvements are in C and C++ packages that don’t have an “ecosystem” as such. They are part of all the ecosystems.

I would really like to be able to improve @NSA-mq29-tgif-claw in light of all the confusion seen in madware8688 (comment). But there’s no way to do this.

What could possibly be done to improve these bulletins?

Neustradus on Nov 21, 2023

An important ticket!

Darshan on Nov 22, 2023

They are part of all the ecosystems.

That’s kinda the problem actually. Our ecosystems provide a one to one mapping between some package namespace and an advisory. We don’t have false positives as a result and not having false positives (via dependency matching) results in more actionable advisories and better outcomes for developers receiving alerts.

What could possibly be done to improve these bulletins?

It’s an open question and one that we’re thinking about.

For what it’s worth we do have plenty of advisories which are about C/C++ code which has been bundled into a package in one of our ecosystems

eg. https://github.com/advisories?query=type%3A%2C%2CC%2B%2B%20%7C%7C%20ecosystem%3Apip

https://github.com/github/advisory-database/issues/2963
We’re on a journey
join us won’t you?
Write advisories?
Call out your artifacts

CVE-2024-22243: Spring Framework URL Parsing with Host Validation

HIGH | FEBRUARY 21, 2024 | CVE-2024-22243

Description
Applications that use `UriComponentsBuilder` to parse an externally provided URL (e.g., through a query parameter) AND perform validation checks on the host of the parsed URL may be vulnerable to a open redirect attack or to a SSRF attack if the URL is used after passing validation checks.

Affected Spring Products and Versions
Spring Framework
- 6.1.0 - 6.1.3
- 6.0.0 - 6.0.16
- 5.3.0 - 5.3.31
- Older, unsupported versions are also affected

Mitigation
Upgrade Spring Framework as follows:
- 6.1.x users should upgrade to 6.1.4
- 6.0.x users should upgrade to 6.0.17
- 5.3.x users should upgrade to 5.3.32
No other steps are necessary.

Credit
The issue was identified and responsibly reported by Sean Presc from Motorola Solutions.

GitHub Advisory Database / GitHub Reviewed / CVE-2024-22243

Spring Web vulnerable to Open Redirect or Server Side Request Forgery

Vulnerability details  Dependabot alerts  29

<table>
<thead>
<tr>
<th>Package</th>
<th>Affected versions</th>
<th>Patched versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>/ org.springframework:spring-web (Maven)</td>
<td>&gt;= 6.1.0, &lt; 6.1.4</td>
<td>6.1.4</td>
</tr>
<tr>
<td></td>
<td>&gt;= 6.0.0, &lt; 6.0.17</td>
<td>6.0.17</td>
</tr>
<tr>
<td></td>
<td>&gt;= 5.3.0, &lt; 5.3.32</td>
<td>5.3.32</td>
</tr>
<tr>
<td></td>
<td>&lt;= 5.2.25.RELEASE</td>
<td></td>
</tr>
</tbody>
</table>

Description
Applications that use UriComponentsBuilder to parse an externally provided URL (e.g., through a query parameter) AND perform validation checks on the host of the parsed URL may be vulnerable to a open redirect attack or to a SSRF attack if the URL is used after passing validation checks.

References
### Read advisories?

### Call out errors

<table>
<thead>
<tr>
<th>Status</th>
<th>Title</th>
<th>Author</th>
<th>Label</th>
<th>Projects</th>
<th>Milestones</th>
<th>Reviews</th>
<th>Assignee</th>
<th>Sort</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 Closed</td>
<td>Update GHSA-r4q3-7g4q-x89m.json CVE-2024-22233</td>
<td>prabhuclosed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[GHSA-r4q3-7g4q-x89m] Spring Framework server Web DoS Vulnerability</td>
<td>LukaszGrzesik</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[GHSA-r4q3-7g4q-x89m] Spring Framework server Web DoS Vulnerability</td>
<td>tolimadismerged</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[GHSA-r4q3-7g4q-x89m] Spring Framework server Web DoS Vulnerability</td>
<td>schmidt-fumerged</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[GHSA-r4q3-7g4q-x89m] Spring Framework server Web DoS Vulnerability</td>
<td>fnxptmerged</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[GHSA-r4q3-7g4q-x89m] Spring Framework server Web DoS Vulnerability</td>
<td>Yukitnuclosed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[GHSA-r4q3-7g4q-x89m] Spring Framework server Web DoS Vulnerability</td>
<td>reavmerged</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[GHSA-r4q3-7g4q-x89m] Spring Framework server Web DoS Vulnerability</td>
<td>arunekomerged</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Want something we don’t provide?
Build it. If it’s good they will come, we might help.
**GitHub Advisory Database**

Security vulnerability database inclusive of CVEs and GitHub originated security advisories from the world of open source software.

### GitHub reviewed advisories

<table>
<thead>
<tr>
<th>All reviewed</th>
<th>14,490</th>
</tr>
</thead>
</table>

- **Composer**: 2,242
- **Erlang**: 22
- **GitHub Actions**: 11
- **Go**: 1,214
- **Maven**: 3,951
- **npm**: 3,147
- **NuGet**: 510
- **pip**: 2,083
- **Pub**: 7
- **RubyGems**: 743
- **Rust**: 667
- **Swift**: 29

### Unreviewed advisories

- **All unreviewed**: 187,394

---

- ![CC-BY-4.0 License](image)
- ![Language support](image)
- ![About GitHub Advisory Database](image)
Help us improve our data, it’s available for free, for everyone, forever.
Questions?

We Can Do It!

Help us improve our data, it's available for free, for everyone, forever.
Global API

Want to use our data for your own tools?

Global security advisories

Use the REST API to view global security advisories.

List global security advisories

Works with GitHub Apps

Lists all global security advisories that match the specified parameters. If no other parameters are defined, the request will return only GitHub-reviewed advisories that are not malware.

By default, all responses will exclude advisories for malware, because malware are not standard vulnerabilities. To list advisories for malware, you must include the `type` parameter in your request, with the value `malware`. For more information about the different types of security advisories, see "About the GitHub Advisory database."

Parameters for "List global security advisories"

Code samples for "List global security advisories"

```bash
GET /advisories

cURL, JavaScript, GitHub CLI

curl -L
  -H "Accept: application/vnd.github+json"
  -H "Authorization: Bearer YOUR-TOKEN"
  -H "X-Github-Api-Version: 2022-11-28"
  https://api.github.com/advisories
```

Repo API

Want to get at raw user provided data?

Repository security advisories

Use the REST API to view and manage repository security advisories.

List repository security advisories for an organization

Works with Github Apps

Lists repository security advisories for an organization.

To use this endpoint, you must be an owner or security manager for the organization, and you must use an access token with the `repo` scope or `repository_advisories_write` permission.

Parameters for "List repository security advisories for an organization"

Headers

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>accept</td>
<td>string</td>
</tr>
</tbody>
</table>

Setting to `application/vnd.github+json` is recommended.

Path parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>org</td>
<td>string Required</td>
</tr>
</tbody>
</table>

The organization name. The name is not case sensitive.

Query parameters

Code samples for "List repository security advisories for an organization"

GET /orgs/org/security-advisories

cURL | JavaScript | Github CLI

```
```

Response

<table>
<thead>
<tr>
<th>Example response</th>
<th>Response schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status: 200</td>
<td></td>
</tr>
</tbody>
</table>

\[
\{
  "ghsa_id": "GHSB-abc1234-efgh",
  "cve_id": "CVE-2019-0000",
  "url": "https://api.github.com/repos/repo/a-package/security-advisories",
\}
\]
Private Vulnerability Reporting

Want to not have 0 days dropped on you?

Configuring private vulnerability reporting for a repository

Owners and administrators of public repositories can allow security researchers to report vulnerabilities securely in the repository by enabling private vulnerability reporting.

Who can use this feature

Anyone with admin permissions to a public repository can enable and disable private vulnerability reporting for the repository.

About privately reporting a security vulnerability

Security researchers often feel responsible for alerting users to a vulnerability that could be exploited. If there are no clear instructions about contacting maintainers of the repository containing the vulnerability, security researchers may have no other choice but to post about the vulnerability on social media, send direct messages to the maintainer, or even create public issues. This situation can potentially lead to a public disclosure of the vulnerability details.
Dismissal Rules
Want to fine tune what alerts you see?

Using alert rules to prioritize Dependabot alerts
You can use Dependabot alert rules to filter out false positive alerts or alerts you’re not interested in.

Who can use this feature
People with write permissions can view Dependabot alert rules for the repository. People with admin permissions to a repository, or the security manager role for the repository, can enable or disable Dependabot alert rules for the repository, as well as create custom alert rules.

Note: Dependabot alert rules are currently in beta and are subject to change.