# Diffing Hex Packages

# Johanna Larsson

@joladev

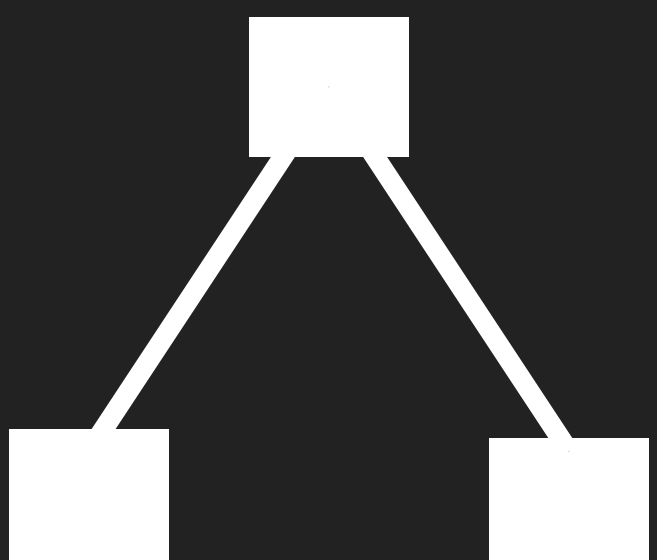

Duffel

# Agenda

- History of dependencies
- Dependencies today
- Attacks in the wild
- Mitigations
- diff.hex.pm
- Takeaways

# Introduction

# Dependencies

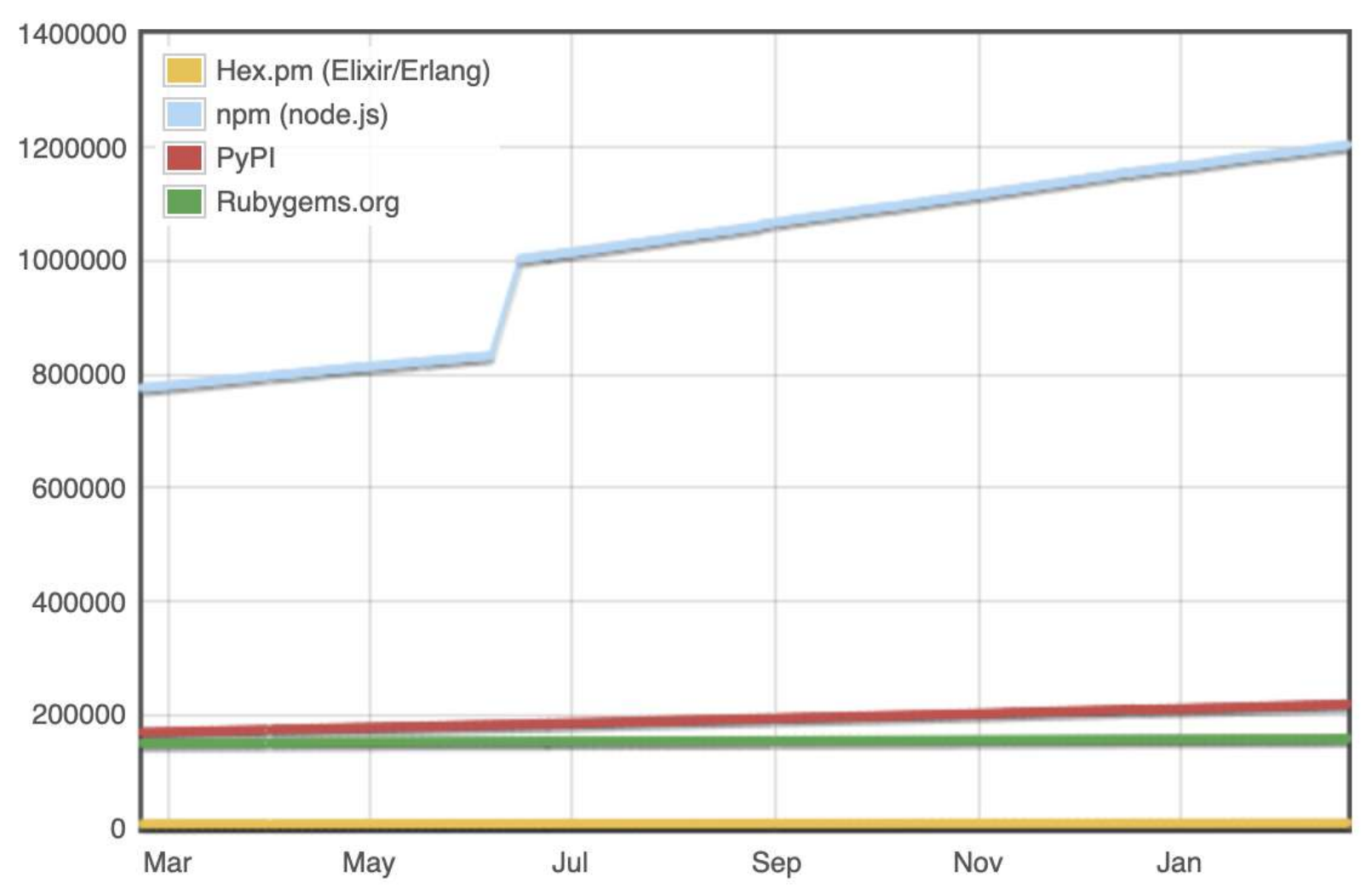


# Vendoring

# The Rise of the Package Repository

# The Dominance of

http://www.modulecounts.com/



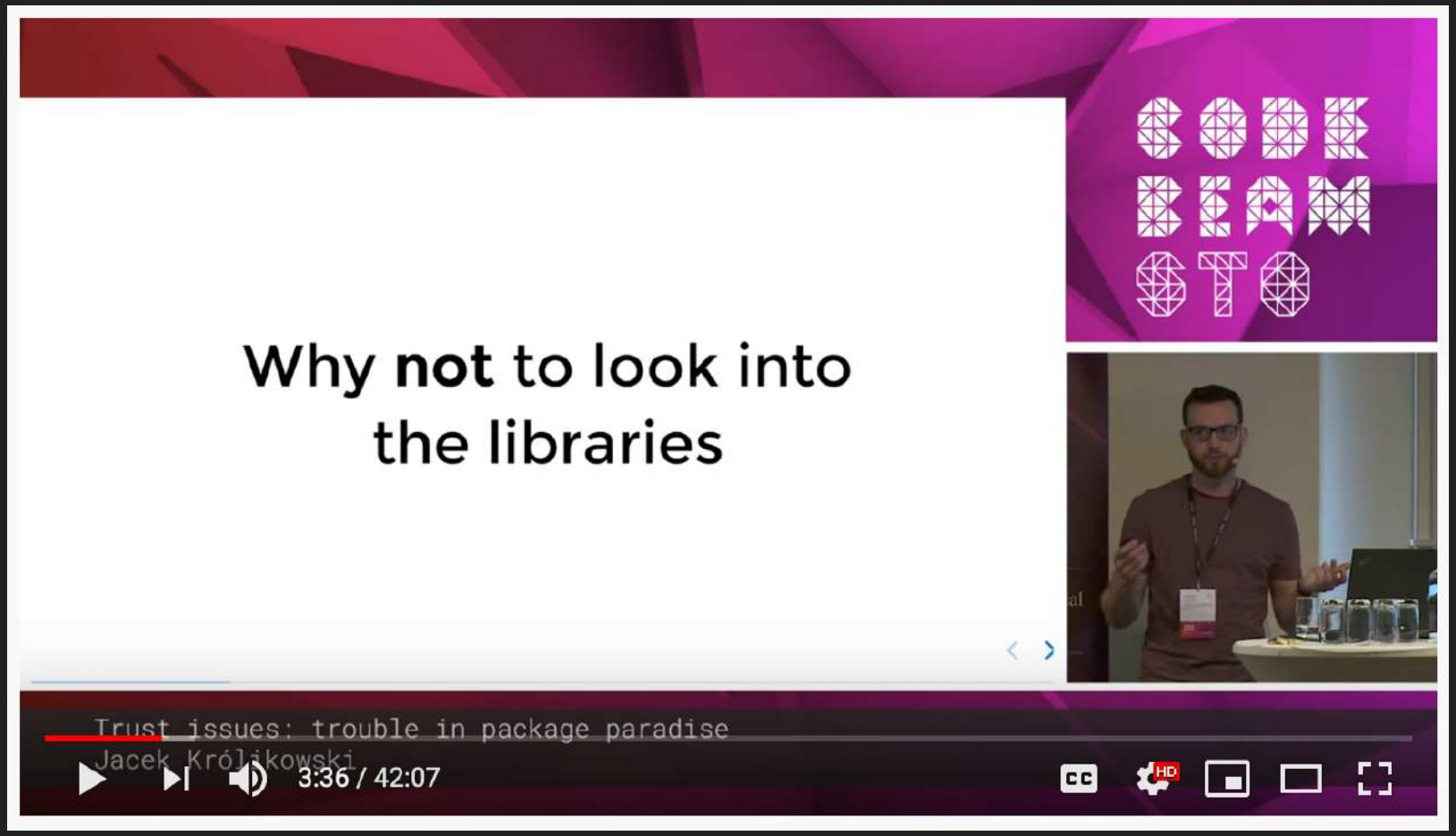
# I'm harvesting credit card numbers and passwords from your site. Here's how.

David Gilbertson Follow

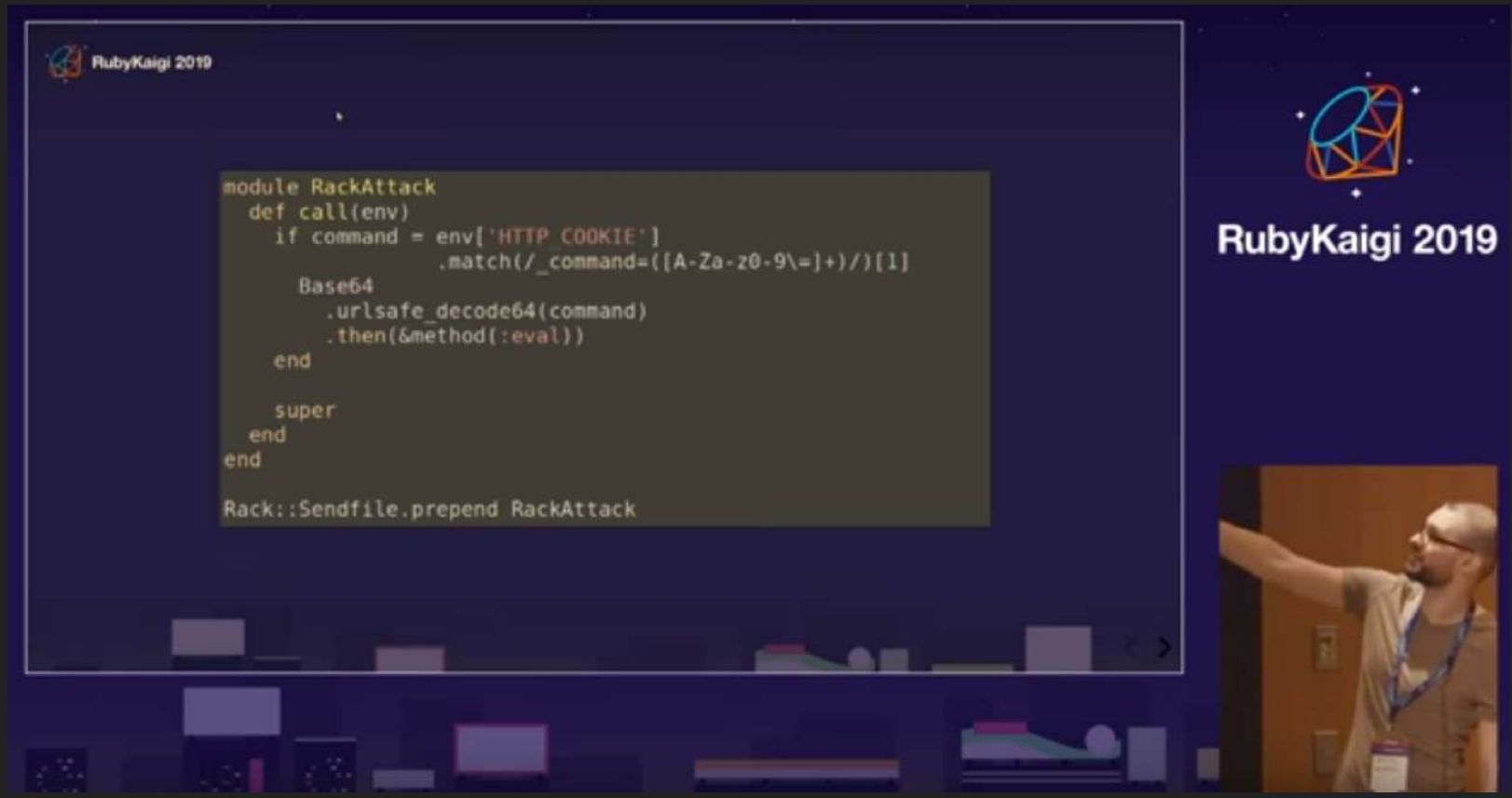
Jan 6, 2018 · 10 min read

The following is a true story. Or maybe it's just based on a true story. Perhaps it's not true at all.

https://medium.com/hackernoon/im-harvesting-credit-card-numbers-and-passwords-from-your-site-here-s-how-9a8cb347c5b5




Jacek Królikowski - Trust issues: trouble in package paradise | Code BEAM STO 19




How to take over a Ruby gem / Maciej Mensfeld @maciejmensfeld

# Transitive dependencies and risk

*“ Installing an average npm package introduces an implicit trust on 79 third-party packages and 39 maintainers, creating a surprisingly large attack surface.*

Small World with High Risks:

A Study of Security Threats in the npm Ecosystem

*“Some maintainers have an impact on hundreds of thousands of packages. As a result, a very small number of compromised maintainer accounts suffices to inject malware into the majority of all packages.*

Small World with High Risks:

A Study of Security Threats in the npm Ecosystem

# Attacks

# Distributing Malicious Code

- Typosquatting
- Malicious dependency
- Compromised credentials
- Offering to maintain package

Intentional or accidental

For profit or to harm

# Run time

vs

# Compile time

# eslint-scope

*“ On installation, the malicious packages downloaded and executed code from pastebin.com which sent the contents of the user's .npmrc file to the attacker. An .npmrc file typically contains access tokens for publishing to npm.*

https://eslint.org/blog/2018/07/postmortem-for-malicious-package-publishes

# event-stream

Scanned the computer for bitcoin wallets and sent credentials to the attacker's server.

Only ran in production mode, not testing, and only for wallets with enough bitcoin in them for it to be worth it.

Took 2 months before it was discovered.

https://medium.com/intrinsic/compromised-npm-package-event-stream-d47d08605502

# bootstrap-sass

Uploaded new malicious version and yanked the previous one to force updates.

Monkey patches rails to eval a special header if it's found in a request, allowing remote code execution.

Was discovered the same day.

https://snyk.io/blog/malicious-remote-code-execution-backdoor-discovered-in-the-popular-bootstrap-sass-ruby-gem/

# Mitigations


- Run by volunteers

- Typosquatting
- Limited republishing/unpublishing

# What can you do?

- Static analysis
- Hardening infrastructure
- Auditing dependencies

# Diffing

# Can't I just use Github?

mix hex.package diff

# diff.hex.pm

# Brief history

https://diff.jola.dev

# Web based package diffs #848

Closed **joladev** opened this issue on Sep 8, 2019 · 6 comments

**joladev** commented on Sep 8, 2019 • edited ▾ Contributor +😀 ···

Hi! I wanted to bring this issue up for discussion. I'll try to describe the importance of package diffs, existing solutions, some possible architectures and some possible solutions that might fit hex.pm. By web-based diffs I mean highlighted outputs from `git diff` in a browser, with shareable links.

We first started seeing npm packages get hijacked, but recently RubyGems has been having issues (rest_client, strong_password, many others). By hijacking I mean the the type of attack where someone gets access to the credentials of the author of a package and uploads malicious versions. Some examples of scenarios:

1. You merge the new updates and deploy, now you're running infected code in production. They can then mine cryptocurrency or inject HTTP handlers that respond to certain payloads, even giving access to servers and databases.


Search...

## About Hex

About
Blog
Sponsors
GitHub
Twitter

## Help

Documentation
Specifications
Report Client Issue
Report General Issue
Contact Support

## Policies and Terms

Code of Conduct
Terms of Service
Privacy Policy
Copyright Policy
Dispute Policy

# prior art


RubyGems


npm

# Technical Background

# hex_core

```
1 > hex_repo:get_names(Config).
2 {ok, {200, ...,
3     #{packages => [
4         #{name => <<"package1">>},
5         #{name => <<"package2">>},
6         ...
7     ]}}}
```

https://github.com/hexpm/hex_core

# ETS

```elixir
def get_versions(key) do
  case :ets.lookup(@store_module, key) do
    [{_key, versions}] -> {:ok, versions}
    _ -> {:error, :not_found}
  end
end

def get_names() do
  :ets.select(@store_module, [{{:"$1", :_}, [], [:"$1"]}])
end
```

https://erlang.org/doc/man/ets.html

# git_diff

```
1 [
2   %GitDiff.Patch{
3     chunks: [
4       %GitDiff.Chunk{
5         from_num_lines: "42",
6         from_start_line: "42",
7         header: "@@ -481,23 +483,24 @@ class Cursor
8         context: "class Cursor extends Model {", #
```

https://github.com/mononym/git_diff

# LiveView

LiveDemo!

https://github.com/phoenixframework/phoenix_live_view

# Problems seen

# Stream everything #46

Merged **ericmj** merged 2 commits into `master` from `emj/stream` 7 days ago

Conversation 8 | Commits 2 | Checks 5 | Files changed 17

**ericmj** commented 19 days ago — Member

This PR reduces peak memory usage for `/diff/udia/0.0.1..0.1.0` from 1400MB to 130MB.

**ericmj** force-pushed the `emj/stream` branch from **a6d22ce** to **020ae89** 19 days ago

GCP load balancer closes connections after 30s

# Takeaways

`hex_core` **is really cool**

Talk about how to work with dependencies in your workplace in a way where it doesn't hurt productivity

# Improve Tooling

Contribute back to the community, report vulnerabilities

# Thank you!