Securing the software supply chain

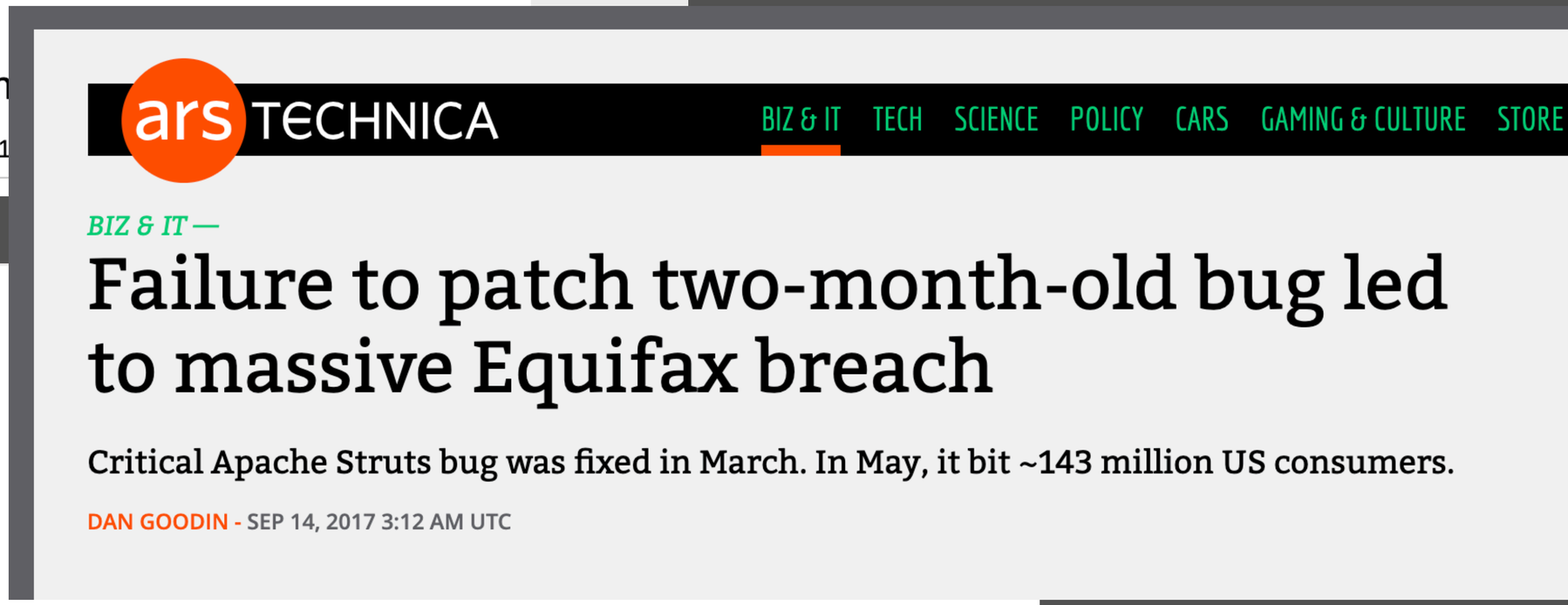
Filippo Valsorda

Google

We all use

other people's code


The Register®
Biting the hand that feeds IT
DATA CENTRE
SOFTWARE
SECURITY
DEVOPS
BUSINESS
PERSONAL TECH
SCIENCE
EMERGENT TECH
Software
How one developer just broke Node, Babel and thousands of projects in 11 lines of JavaScript
Code pulled from NPM – wh
By Chris Williams, Editor in Chief 23 Mar 201
MOST REA
1

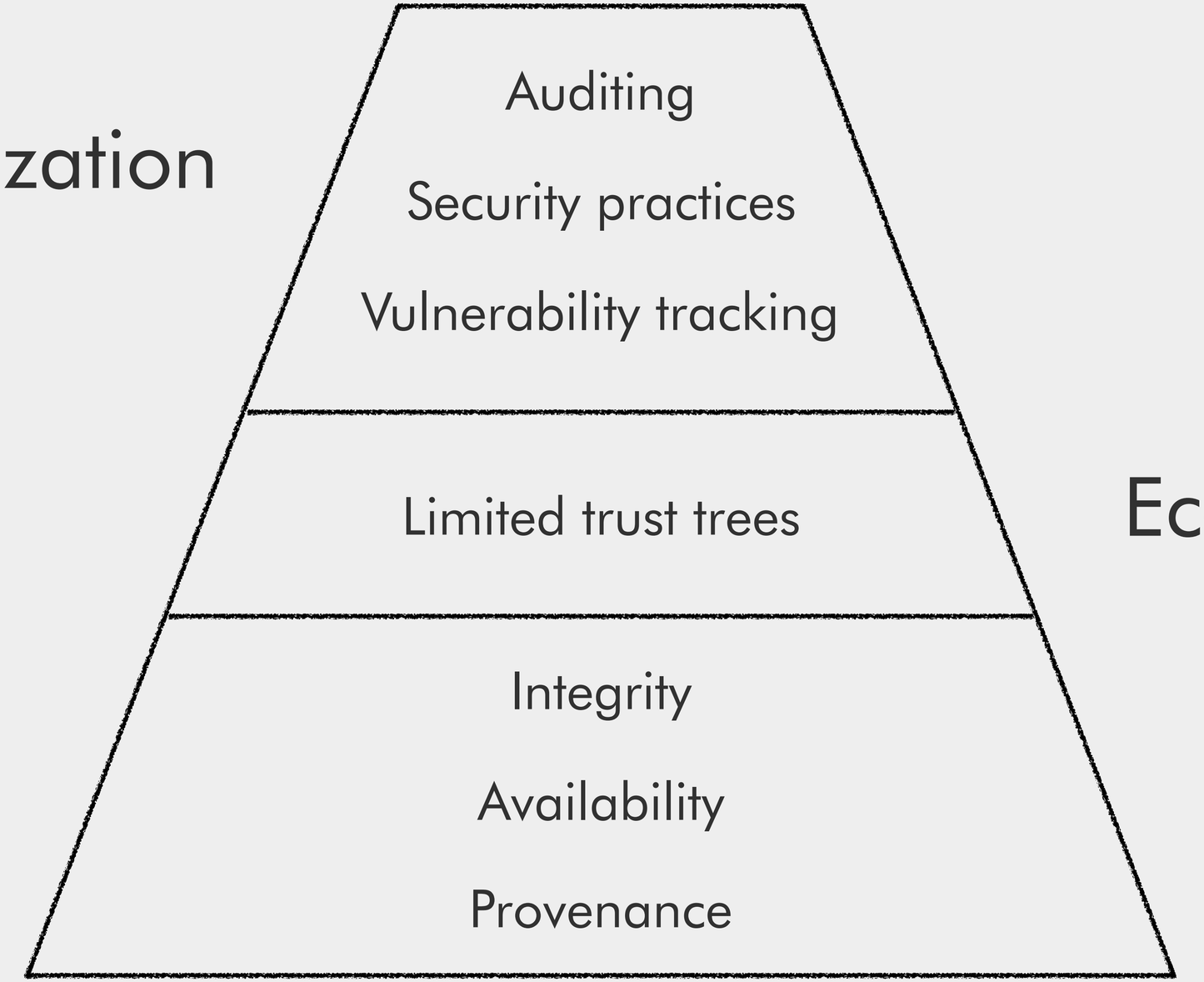

ars TECHNICA
BIZ & IT
TECH
SCIENCE
POLICY
CARS
GAMING & CULTURE
STORE
BIZ & IT —
Failure to patch two-month-old bug led to massive Equifax breach
Critical Apache Struts bug was fixed in March. In May, it bit ~143 million US consumers.
DAN GOODIN - SEP 14, 2017 3:12 AM UTC


the npm blog
Blog about npm things.
Details about the event-stream incident

Three supply chain security players

1. Language — enables trust
2. Ecosystem — propagates and limits trust
3. Organization — manages and mitigates trust

Organization



Ecosystem

Language

Three supply chain security players

1. Language — enables trust
2. Ecosystem — propagates and limits trust
3. Organization — manages and mitigates trust

Three supply chain security players

1. Language — enables trust
    1. Provenance — what code do we depend on?
    2. Availability — where do we get it?
    3. Integrity — has it been tampered with?

Provenance

Trying to establish:

- a universal name
- a permanent version

Python: PyPi and pipenv

Rust: crates.io and cargo

Node: NPM

Ruby: rubygems.org and Bundler

Go Modules

```
module github.com/FiloSottile/mostly-harmless/dcbot

require (
   crawshaw.io/sqlite v0.1.1
   github.com/pkg/errors v0.8.0
   github.com/sirupsen/logrus v1.2.0
   golang.org/x/sync v0.0.0-20181108010431-42b317875d0f
)

go 1.14
```

Availability

Making sure the code is still available in the future.

ars TECHNICA

BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE STOR

BIZ & IT —

Rage-quit: Coder unpublished 17 lines of JavaScript and “broke the Internet”

Dispute over module name in npm registry became giant headache for developers.

SEAN GALLAGHER - MAR 25, 2016 2:10 AM UTC

Go Module Proxies and Mirror

```
GOPROXY=https://proxy.golang.org

https://proxy.golang.org/
github.com/sirupsen/logrus/@v/
v1.4.2.zip
```

Integrity

Protecting code from tampering.

No trust in proxies.

No trust in a central entity.

No trust on first use.

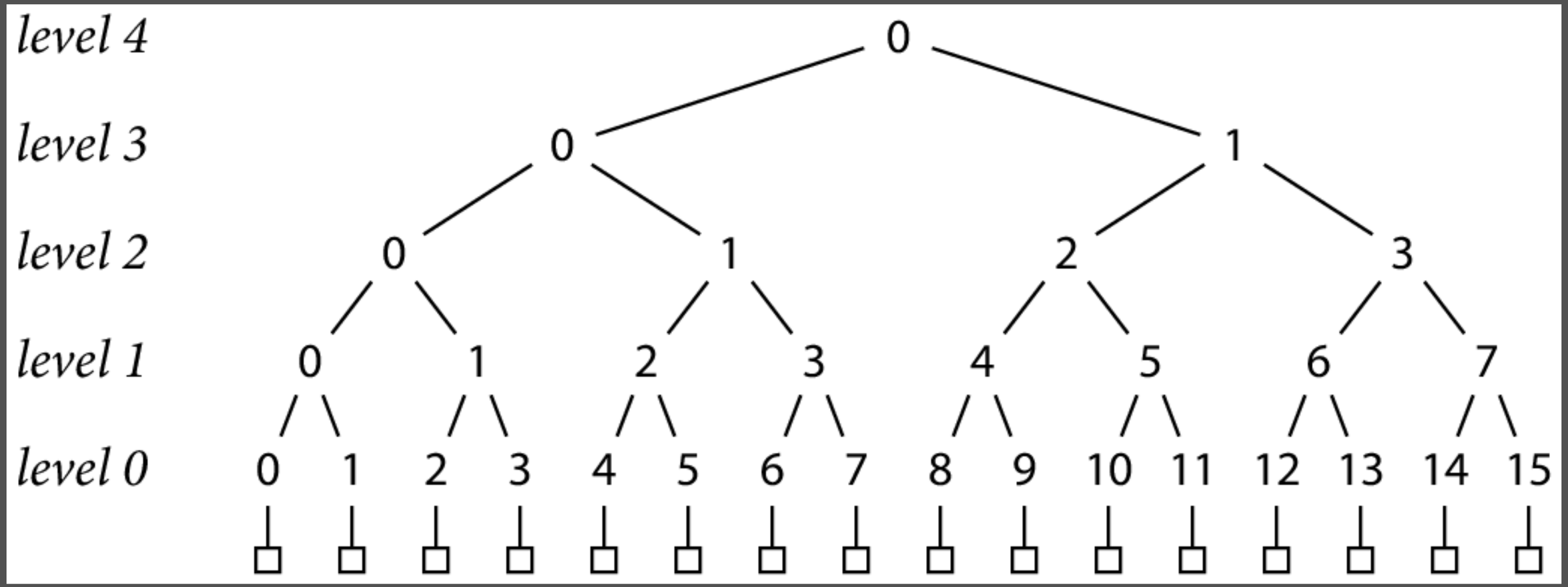
No key management for authors.

The Go Checksum Database

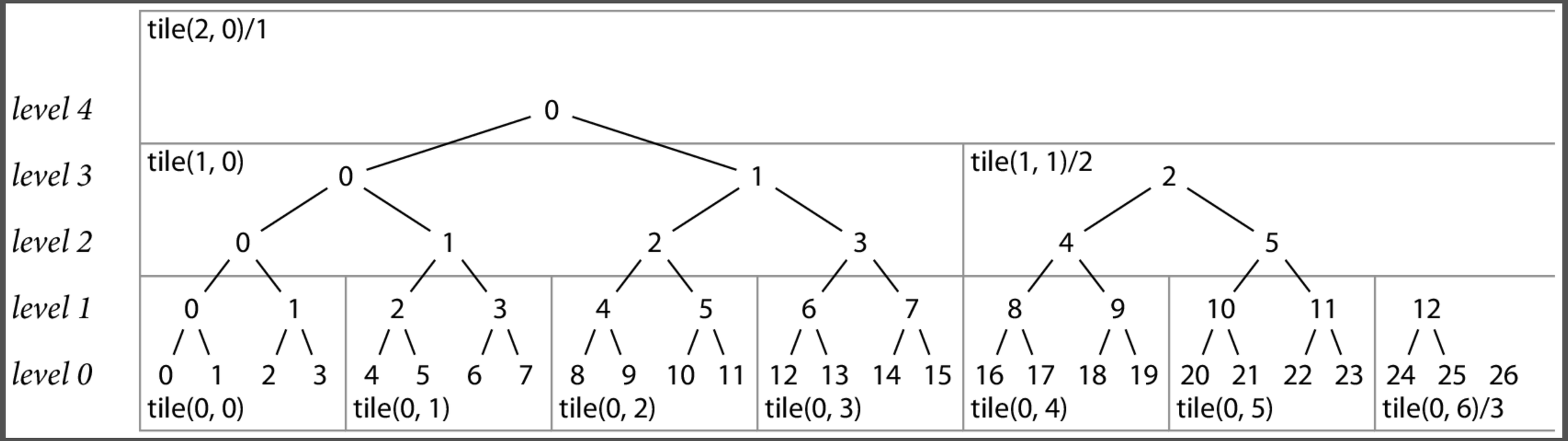
Solve
"is everyone looking at the same code"
as a proxy for
"is this the right code"

```
$ curl https://sum.golang.org/lookup/github.com/sirupsen/logrus@v1.4.2

15937
github.com/sirupsen/logrus v1.4.2
h1:SPIRibHv4MatM3XXNO2BJeFLZwZ2LvZgfQ5+UNI2im4=
github.com/sirupsen/logrus v1.4.2/go.mod
h1:tLMulIdttU9McNUspp0xgXVQah82FyeX6MwdIuYE2rE=

go.sum database tree
737311
a6tjmEcOB7ayb8j3fTHRrOcZ8DMe+iNsWpp8CJMFoqY=

— sum.golang.org Az3gro/R/8oqJRyMC3biFoTrXFFr+nJ6PAhkGIqnsVUbnBNA3vOPxm/
RPGUIm+ejFqe37G9IrT0z+F2hPaYMvaaYSgo=
```

Merkle trees for accountability



Tiles for caching



The Go Checksum Database

- A public append-only log of module version checksums
- kept accountable by Merkle tree proofs verified on the client
- served as cacheable efficient tiles.

Three supply chain security players

1. Language — enables trust
    1. Provenance — what code do we depend on?
    2. Availability — where do we get it?
    3. Integrity — has it been tampered with?

Three supply chain security players

1. Language — enables trust
2. Ecosystem — propagates and limits trust
3. Organization — manages and mitigates trust

Three supply chain security players

1. Language — enables trust
2. Ecosystem — propagates and limits trust
3. Organization — manages and mitigates trust

Importing a dependency delegates a degree of trust to it and to its transitive dependencies.

Importing a dependency delegates a degree of trust to it and to its transitive dependencies.

A healthy ecosystem fights this like technical debt.

ars TECHNICA

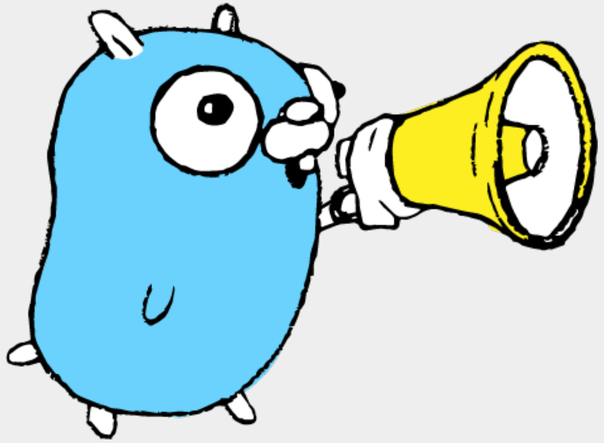
BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE ST

POISONING THE WELL —

Widely used open source software contained bitcoin-stealing backdoor

Malicious code that crept into event-stream JavaScript library went undetected for weeks.

DAN GOODIN - NOV 26, 2018 10:55 PM UTC



Go proverb #8

A little copying is better
than a little dependency.

Go build dependencies

```
$ go list -m all
github.com/FiloSottile/mostly-harmless/dcbot
crawshaw.io/iox v0.0.0-20181124134642-c51c3df30797
crawshaw.io/sqlite v0.1.1-0.20181106130822-19c189e3c5ce
github.com/davecgh/go-spew v1.1.1
github.com/konsorten/go-windows-terminal-sequences v1.0.1
github.com/pkg/errors v0.8.0
github.com/pmezard/go-difflib v1.0.0
github.com/sirupsen/logrus v1.2.0
github.com/stretchr/objx v0.1.1
github.com/stretchr/testify v1.2.2
golang.org/x/crypto v0.0.0-20181203042331-505ab145d0a9
golang.org/x/sync v0.0.0-20181108010431-42b317875d0f
golang.org/x/sys v0.0.0-20181213200352-4d1cda033e06
```

Go build dependencies

```
$ go list -deps -f "{{if not .Standard}}{{.ImportPath}}{{end}}"
crawshaw.io/sqlite
crawshaw.io/sqlite/sqliteutil
github.com/pkg/errors
golang.org/x/sys/unix
golang.org/x/crypto/ssh/terminal
github.com/sirupsen/logrus
github.com/sirupsen/logrus/hooks/syslog
golang.org/x/sync/errgroup
github.com/FiloSottile/mostly-harmless/dcbot
```

Three supply chain security players

1. Language — enables trust
2. Ecosystem — propagates and limits trust
3. Organization — manages and mitigates trust

Three supply chain security players

3. Organization — manages and mitigates trust
    1. Vulnerability tracking — past vulnerabilities
    2. Security practices — future vulnerabilities
    3. Auditing — current vulnerabilities

Vulnerability tracking

Identify and patch public vulnerabilities.

ars TECHNICA

BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE STO

BIZ & IT —

Failure to patch two-month-old bug led to massive Equifax breach

Critical Apache Struts bug was fixed in March. In May, it bit ~143 million US consumers.

DAN GOODIN - SEP 14, 2017 3:12 AM UTC

Security practices

How are new vulnerabilities going to get introduced, discovered, and handled?

Tests, fuzzing and CI

Security reporting

Maintenance status

Sustainability

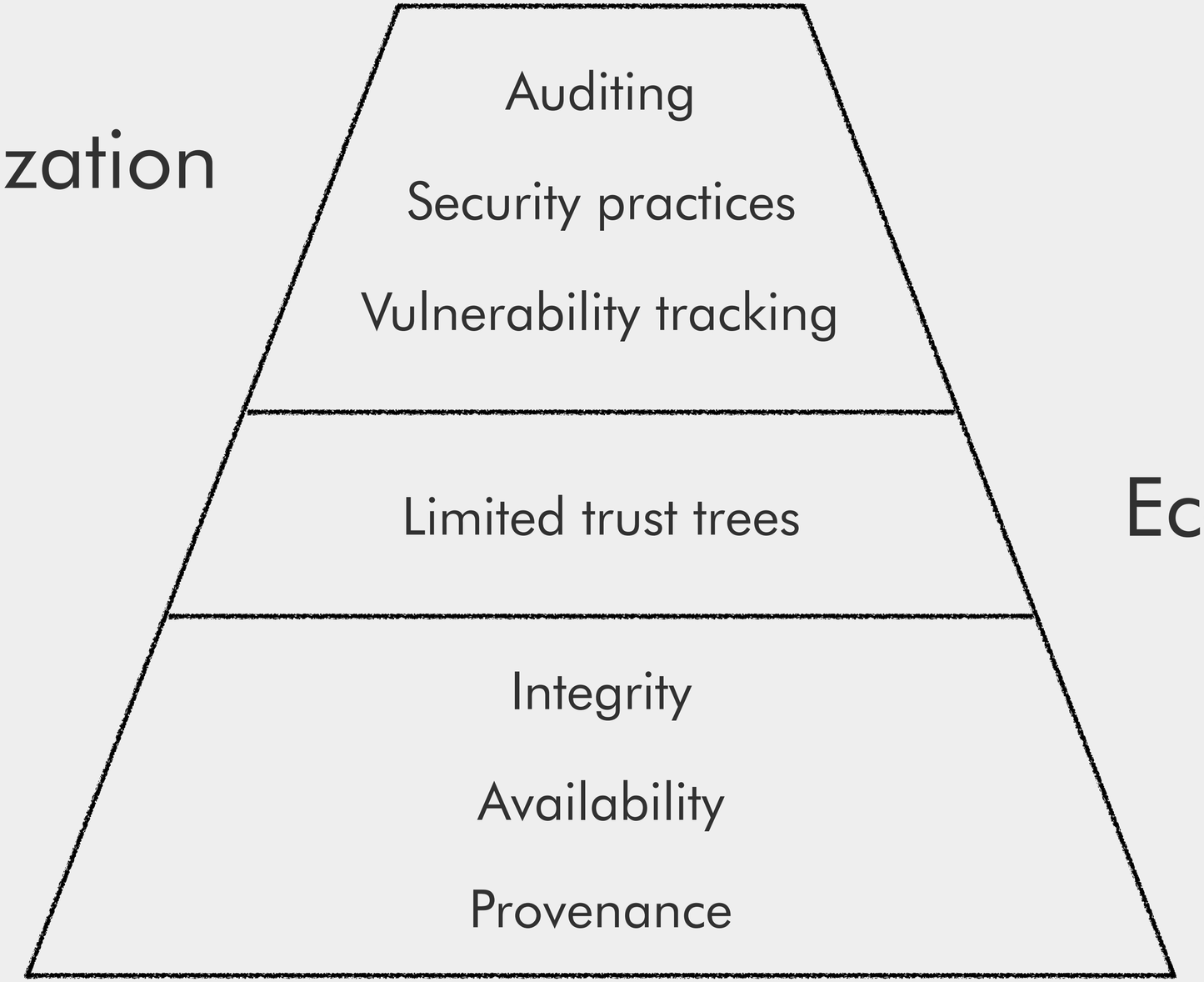
pkg.go.dev

Search for a package

Auditing

Actively looking for malicious code and vulnerabilities.

Organization



Ecosystem

Language

Questions?

The Go Checksum Database
https://golang.org/design/25530-sumdb

"Our Software Dependency Problem" by Russ Cox
https://research.swtch.com/deps

Filippo Valsorda, Google
@FiloSottile — filippo@golang.org