
KubeCon
CloudNativeCon
Europe 2020
Virtual
HELM
KEEP CLOUD NATIVE
CONNECTED

Securing Container Delivery with TUF

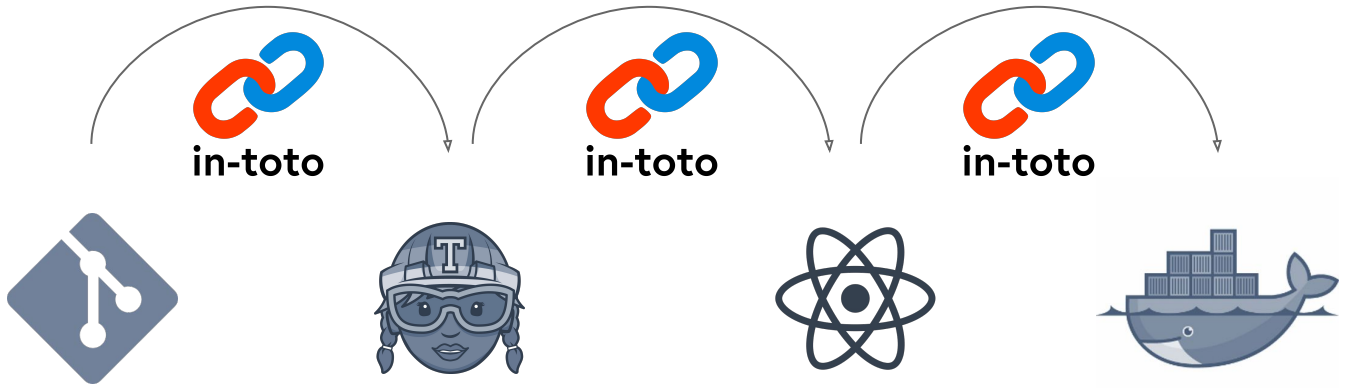
Lukas Pühringer, NYU
August 2020

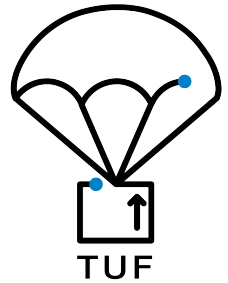
KubeCon | CloudNativeCon

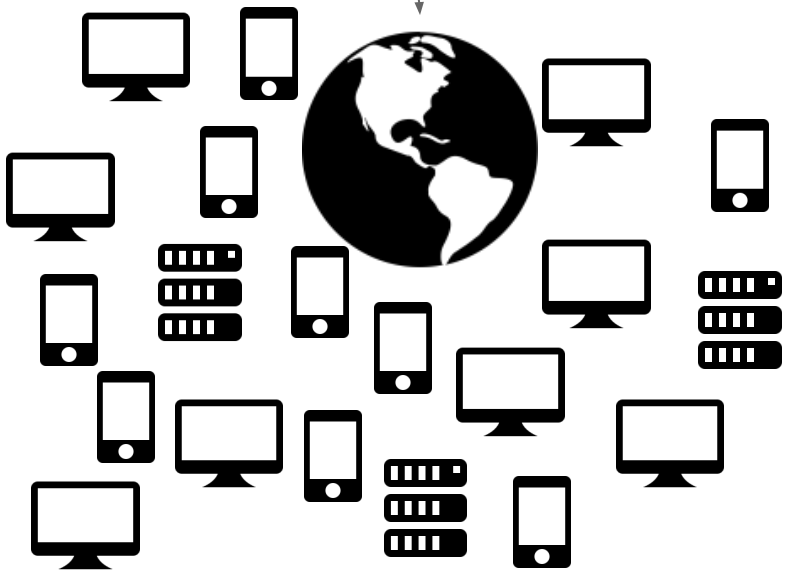
Europe 2020

Virtual

Software supply chain security



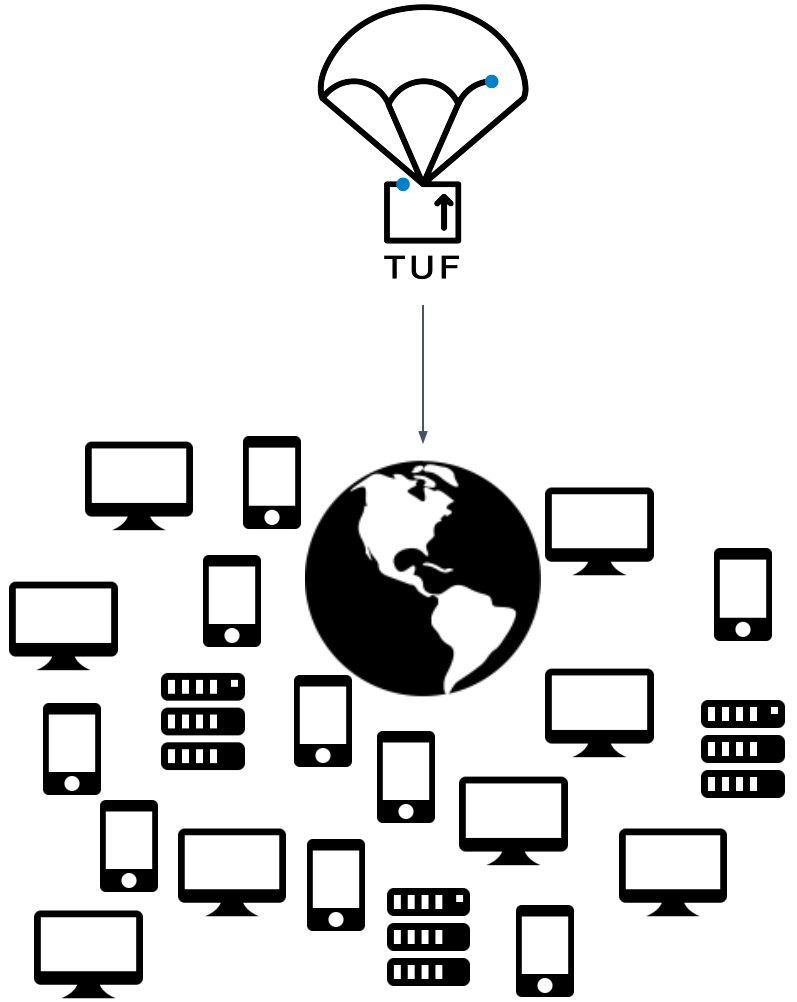




A verifiable trace from the first lines of code until installation or deployment

Software supply chain security

We will look at this part today:



Why do we care about updates?

Important for security

An update fixes CVE-2020-9859

"An application may be able to execute arbitrary code with kernel privileges."

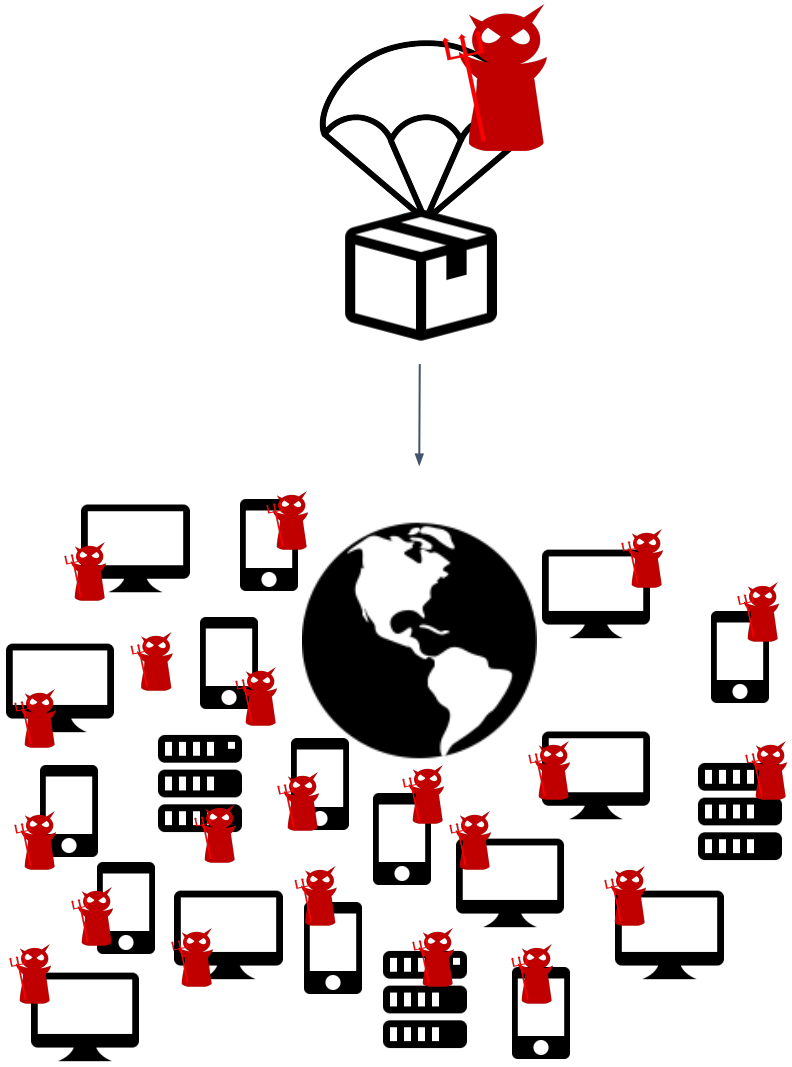
source: https://tidbits.com/2020/06/01/kernel-vulnerability-causes-apple-to-update-all-operating-systems/

That’s why security experts agree...

… “all software and systems should be kept up to date”.

Source: Reeder, R., Ion, I., Consolvo, S.: 152 simple steps to stay safe online: Security advice for non-tech-savvy users. IEEE Security & Privacy (2017)

But also a very attractive target



. . . not only in theory

Adobe

Opera

APACHE HTTP SERVER PROJECT

Windows

php pear "PHP Extension and Application Repository"

CCleaner

debian

RubyGems

source forge

FreeBSD

Linux Mint from freedom came elegance

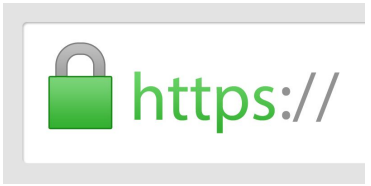
FOSSHUB

Linux

Can’t we just sign it…?

SSL/TLS

- single online key?

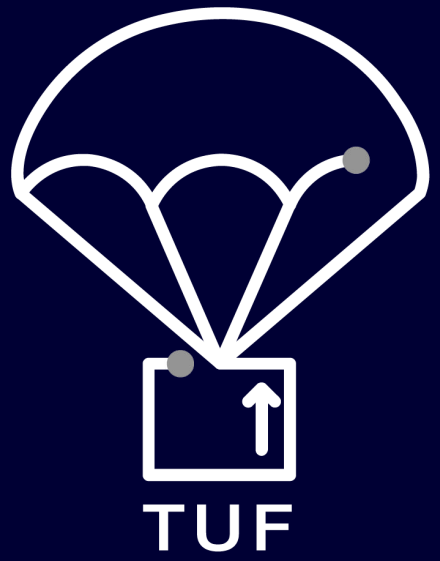


GnuPG

- offline keys!
- distribution/revocation?
- key signing parties?
- thresholds?
- usability?

Needs more than signatures

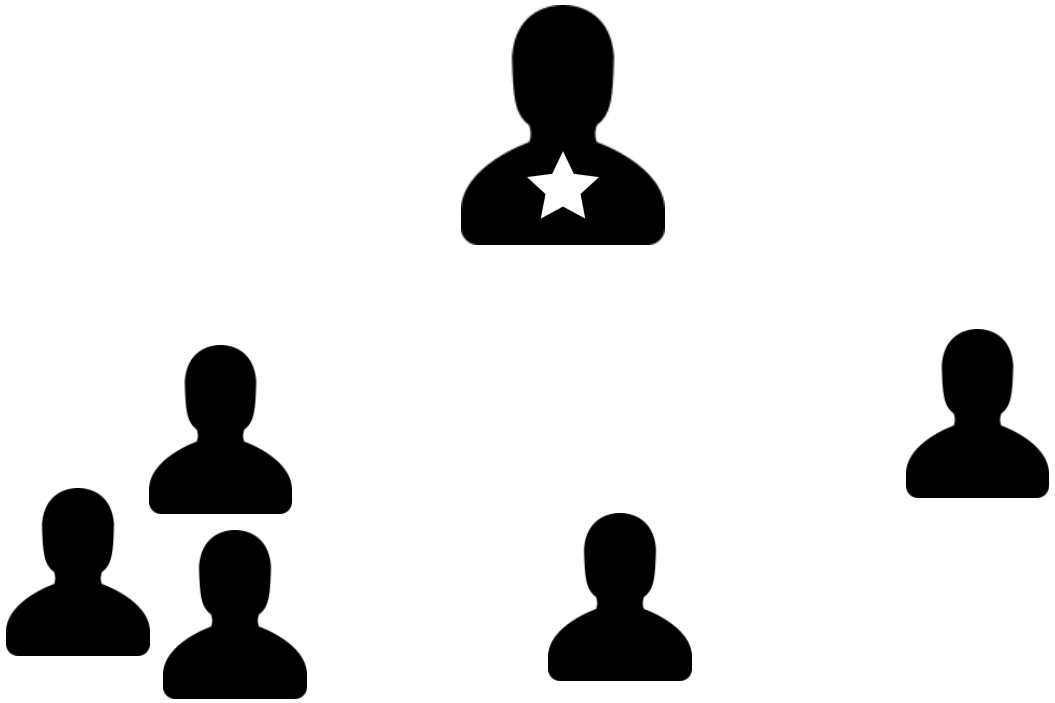
source: poorlylockedbikes.com



Protect against many attacks,
reduce the impact of a compromise,
and allow recovery.

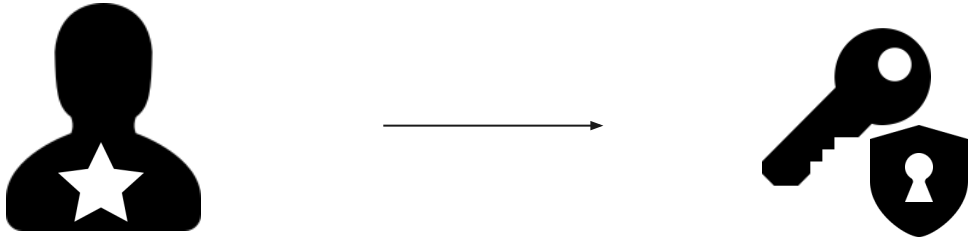
TUF principles I: reduce impact

separation of responsibilities



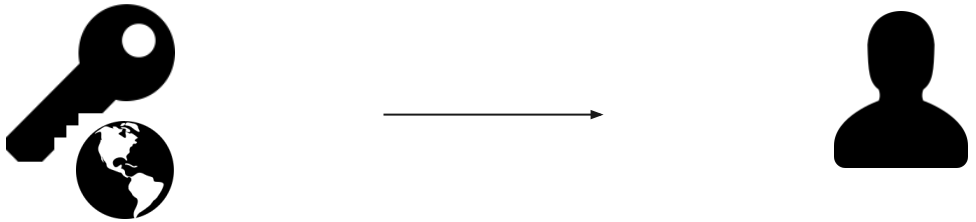
TUF principles I: reduce impact

high-impact responsibilities get secure offline keys



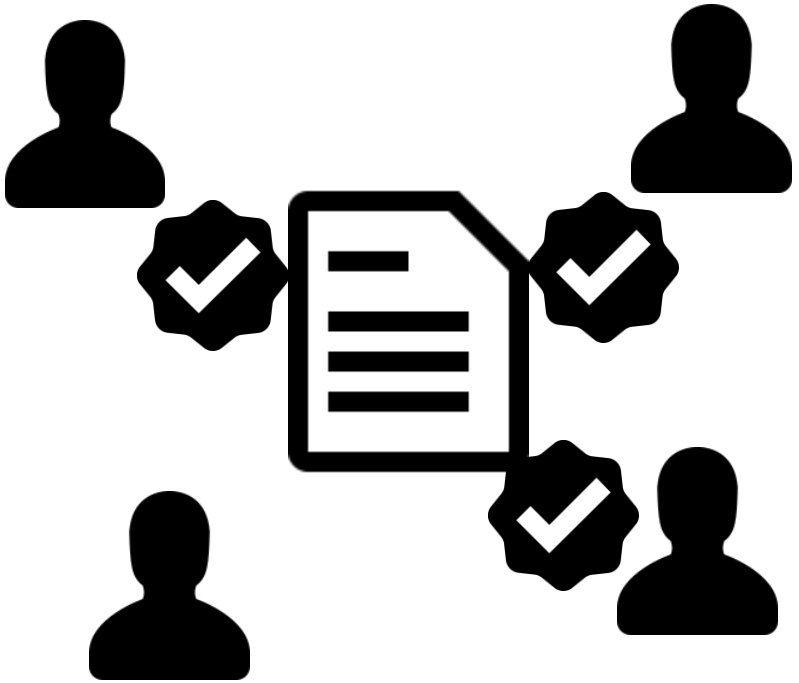
TUF principles I: reduce impact

online keys get low-impact responsibilities



TUF principles I: reduce impact

signature thresholds

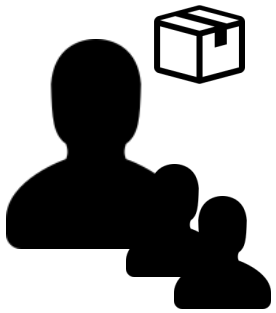


TUF principles 2: allow recovery

explicit and implicit key revocation

more in a minute ...

TUF responsibilities or “roles”

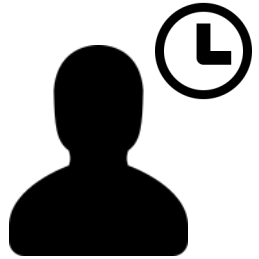


targets

(integrity)

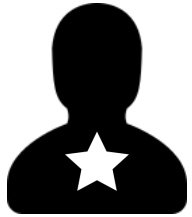
snapshot

(consistency)



timestamp

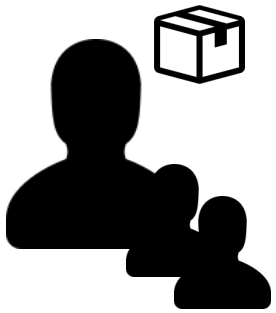
(freshness)



root

(root of trust)

TUF roles or “everything is a file”

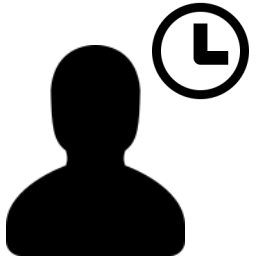


targets.json

()

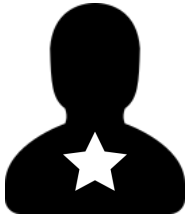
snapshot.json

()



timestamp.json

()



root.json

()

targets.json


```
{
  "BeautifulSoup-3.2.2-py2-any.whl": {
    "hashes": {
      "sha512": “467430a6…”
    },
    "length": 31
    },
    // … other target files
}
```

[note: simplified metadata format for emphasis]

targets.json


```
{
  "BeautifulSoup-3.2.2-py2-any.whl": {
    "hashes": {
      "sha512": “467430a6…”
    },
    "length": 31
    },
    // … other target files
}
```

responsible for integrity of target files

[note: simplified metadata format for emphasis]

targets.json


```
{
  "BeautifulSoup-3.2.2-py2-any.whl": {
    "hashes": {
      "sha512": "467430a6…"
    },
    "length": 31
    },
    // … other target files
}
```

target file hashes

to protect against arbitrary software attack

[note: simplified metadata format for emphasis]

targets.json


```
{
  "BeautifulSoup-3.2.2-py2-any.whl": {
    "hashes": {
      "sha512": “467430a6…”
    },
    "length": 31
    },
    // … other target files
}
```

target file lengths

to protects against endless data attack

[note: simplified metadata format for emphasis]

snapshot.json


```
{
  "targets.json": {
    "version": 12
  }
  // … delegated targets metadata
}
```

[note: simplified metadata format for emphasis]

snapshot.json


```
{
  "targets.json": {
    "version": 12
  }
  // … delegated targets metadata
}
```

responsible for consistency of targets metadata … and thus target files

[note: simplified metadata format for emphasis]

snapshot.json


```
{
  "targets.json": {
    "version": 12
  }
  // … delegated targets metadata
}
```

targets metadata version

to protect against mix and match attacks

[note: simplified metadata format for emphasis]

timestamp.json


```
{
  "expires": “2020-08-18T14:00:00Z“,
  "snapshot.json": {
    "version": 1
  }
  "version": 1
}
```

[note: simplified metadata format for emphasis]

timestamp.json


```
{
  "expires": “2020-08-18T14:00:00Z",
  "snapshot.json": {
    "version": 1
  }
  "version": 1
}
```

responsible for freshness of snapshot metadata

… and thus targets metadata and target files

[note: simplified metadata format for emphasis]

timestamp.json


```
{
  "expires": “2020-08-18T14:00:00Z“,
  "snapshot.json": {
    "version": 1
  }
  "version": 1
}
```

snapshot metadata version

[note: simplified metadata format for emphasis]

timestamp.json


```
{
  "expires": “2020-08-18T14:00:00Z",
  "snapshot.json": {
    "version": 1
  }
  "version": 1
}
```

short expiration period

(implicit revocation)

to protect against freeze attack

[note: simplified metadata format for emphasis]

timestamp.json


```
{
  "expires": “2020-08-18T14:00:00Z“,
  "snapshot.json": {
    "version": 1
  }
  "version": 1
}
```

timestamp metadata version

to protect against rollback attacks

[note: simplified metadata format for emphasis]

root.json


```
{
  "keys": {
    "59a4df8…": {
      "keytype": “ed25519“,
      "public": “adcd0a3…“
    },
    // … more keys
  },
  "targets": {
    "keyids": [“59a4df8…“, … ],
    "threshold": 2
  },
  // … snapshot, timestamp, root
}
```

root of trust

[note: simplified metadata format for emphasis]

root.json


```
{
  "keys": {
    "59a4df8…": {
      "keytype": “ed25519“,
      "public": “adcd0a3…“
    },
    // … more keys
  },
  "targets": {
    "keyids": [“59a4df8…“, … ],
    "threshold": 2
  },
  // … snapshot, timestamp, root
}
```

public key store

[note: simplified metadata format for emphasis]

root.json


```
{
  "keys": {
    "59a4df8…": {
      "keytype": “ed25519“,
      "public": “adcd0a3…“
    },
    // … more keys
  },
  "targets": {
    "keyids": [“59a4df8…“, … ],
    "threshold": 2
  },
  // … snapshot, timestamp, root
}
```

assign keys to roles

[note: simplified metadata format for emphasis]

root.json


```
{
  "keys": {
    "59a4df8…": {
      "keytype": “ed25519“,
      "public": “adcd0a3…“
    },
    // … more keys
  },
  "targets": {
    "keyids": [“59a4df8…“, … ],
    "threshold": 2
  },
  // … snapshot, timestamp, root
}
```

assign keys to roles and defines signature thresholds

[note: simplified metadata format for emphasis]

root.json


```
{
  "keys": {
    "59a4df8…": {
      "keytype": “ed25519“,
      "public": “adcd0a3…“
    },
    // … more keys
  },
  "targets": {
    "keyids": [“59a4df8…“, … ],
    "threshold": 2
  },
  // … snapshot, timestamp, root
}
```

if a key is compromised

[note: simplified metadata format for emphasis]

root.json


```
{
  "keys": {
    "59a4df8…": {
      "keytype": “ed25519“,
      "public": “adcd0a3…“
    },
    "873aaff…": // … new key
  },
  "targets": {
    "keyids": [“59a4df8…“, … ],
    "threshold": 2
  },
  // … snapshot, timestamp, root
}
```

if a key is compromised, we add a new one

[note: simplified metadata format for emphasis]

root.json


```
{
  "keys": {
    "59a4df8…": {
      "keytype": “ed25519“,
      "public": “adcd0a3…“
    },
    "873aaff…": // … new key
  },
  "targets": {  “59a4df8…”
    "keyids": [“873aaff…“, … ],
    "threshold": 2
  },
  // … snapshot, timestamp, root
}
```

[note: simplified metadata format for emphasis]

if a key is compromised, we add a new one, and update the role

(explicit revocation)

no worries, there is tooling ...

Implementations and adoptions

Google

Microsoft Azure

notary

QUAY by CoreOS

Flynn

Advanced Telematic SYSTEMS

DATADOG

Airbiquity OTAmatic™

KOLIDE

CLOUDFLARE

AUTOMOTIVE GRADE LINUX

IBM

PEP 458 (WIP)

- reference implementation
- usability enhancements
- performance optimizations
- exciting new features

Thank you!

KubeCon | CloudNativeCon Europe 2020 Virtual

theupdateframework.io

theupdateframework@googlegroups.com