



Uptane Virtual Industry Conference

Securing Software Updates and Supply Chains on Connected Vehicles

2022-10-13





Agenda

Part I: Coming of Age: The Past and Present of the Uptane Standards

- *What Uptane is, what it does, and how it works (13:00-13:25)*
- *Uptane prevents or deflects specific attacks (13:25-13:55)*
- *Fundamental security assumptions and best practices (13:55-14:30)*

Break (14:30-15:15)

Part II: The Road Ahead: Emerging Challenges for Uptane

- *International standards and national and regional regulations (15:15-15:45)*
 - ISO/SAE 21434 and ISO 24089
 - UN ECE 29 R155 and R156
- *Emerging critical issues (15:45-16:20)*
 - Aftermarket ECUs
 - Supply Chain Security
- *Adoption of Uptane by a Major OEM (16:20-16:40)*
- *Closing Thoughts (16:40-16:45)*



Part I: Coming of Age

The Past and Present of the Uptane Standard

13:00-14:30 CEST



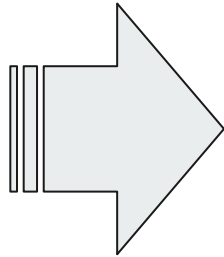
What Uptane is, What it Does and How it Works

Justin Cappos
New York University
(CEST 13:00-13:25)



Who Cares about Hacking Cars?

2015: Guys in tracksuits



Present: Attackers with nation-state level resources



Attacker Goals

Read the contents of updates to discover confidential information, reverse-engineer firmware, or compare two firmware images to identify security fixes and hence determine the fixed security vulnerability.

Deny installation of updates to prevent vehicles from fixing software problems.

Disrupt ECUs in the vehicle, denying use of the vehicle or of certain functions.

Control ECUs within the vehicle, and possibly the vehicle itself.





Uptane Goals

- Prevent known attacks on software update systems
- Provide compromise resilience and security by design
- Minimize damage from a compromised signing key or repository



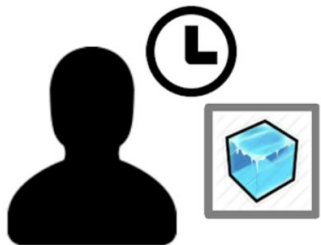


Separation of Roles



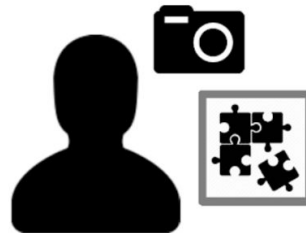
Root

(Root of Trust)



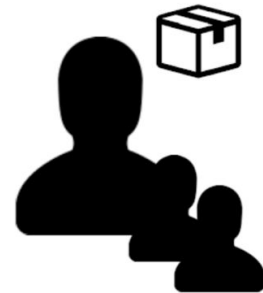
Timestamp

(Freshness)



Snapshot

(Consistency)



Targets

(Authenticity)



Offline and Online Keys on Repos both fail

The OEM needs to tell ECUs which software is authentic and should be installed

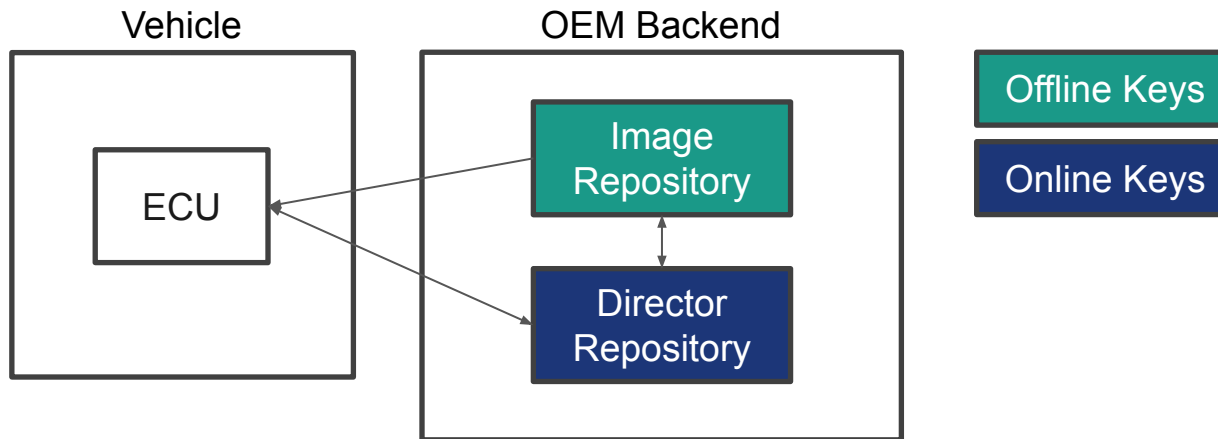
If the keys for authenticity are kept **online** (are on the repository, even in a HSM, etc.):

- A repository hack compromises all users

If the keys for saying which software should be installed are **offline** (e.g., Yubikey kept in a locked desk drawer):

- It is completely unusable because the key needs to be used repeatedly.

Offline and Online Keys



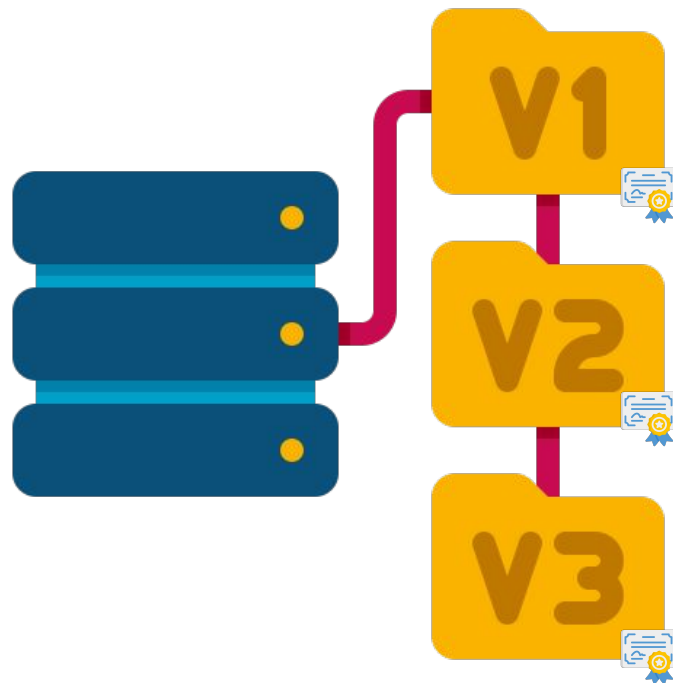
Uptane uses two repositories to provide OEMs with both **security** and **flexibility**!



Image Repository

Authenticity of software images

- 1) Human managed
- 2) Offline keys
- 3) Infrequent updates
- 4) Provides flexible delegation for image signing





Director Repository

Which software should be installed

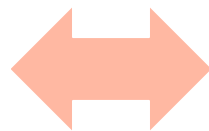
- 1) Automated
- 2) Online Keys
- 3) Frequent Requests
- 4) Generates signed vehicle specific manifest
- 5) Let's OEM control what images are installed
- 6) Works in coordination with a vehicle configuration database





How much work should an ECU do?

If all ECUs must do a lot
of security verification,
few can be protected



If all ECUs do very little
security verification,
protections are limited



Full Verification of Secondaries



Higher computational
cost




Robust security
resilient to
compromised
Primary



Partial Verification of Secondaries



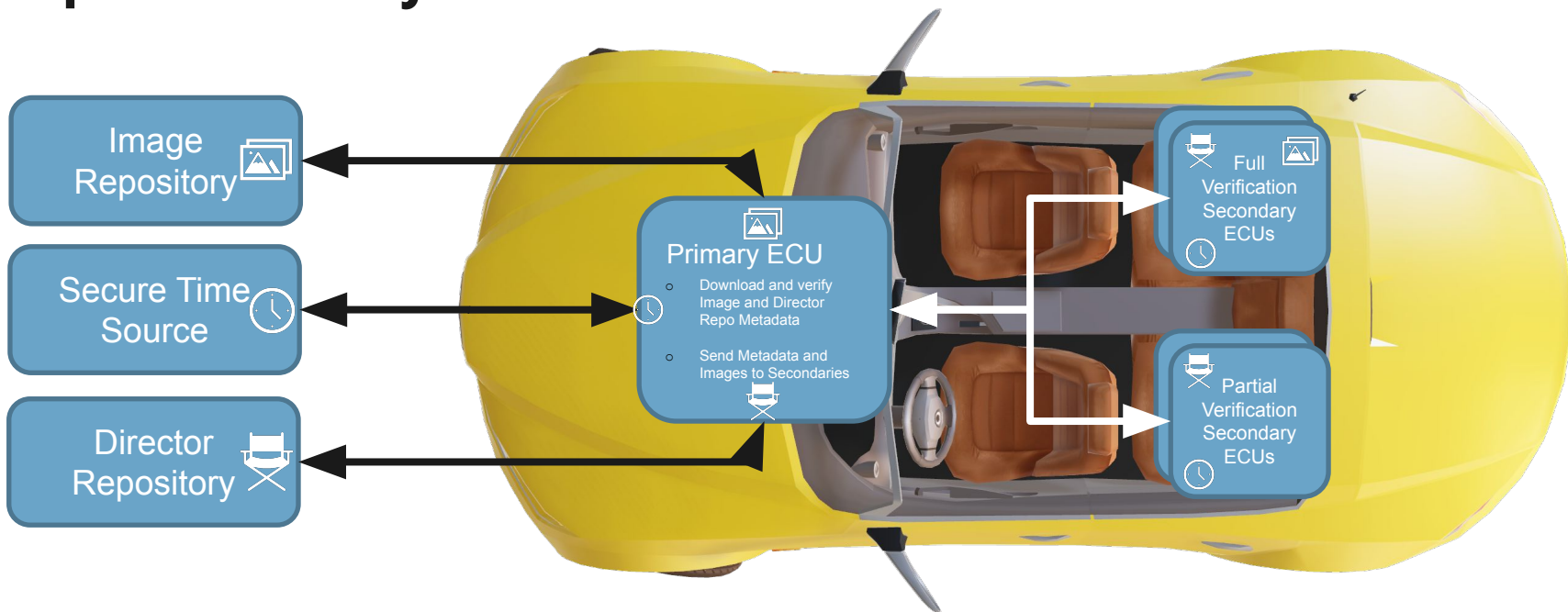
Reduced resilience
to compromised
Primary



Fewer signature
checks for
constrained systems

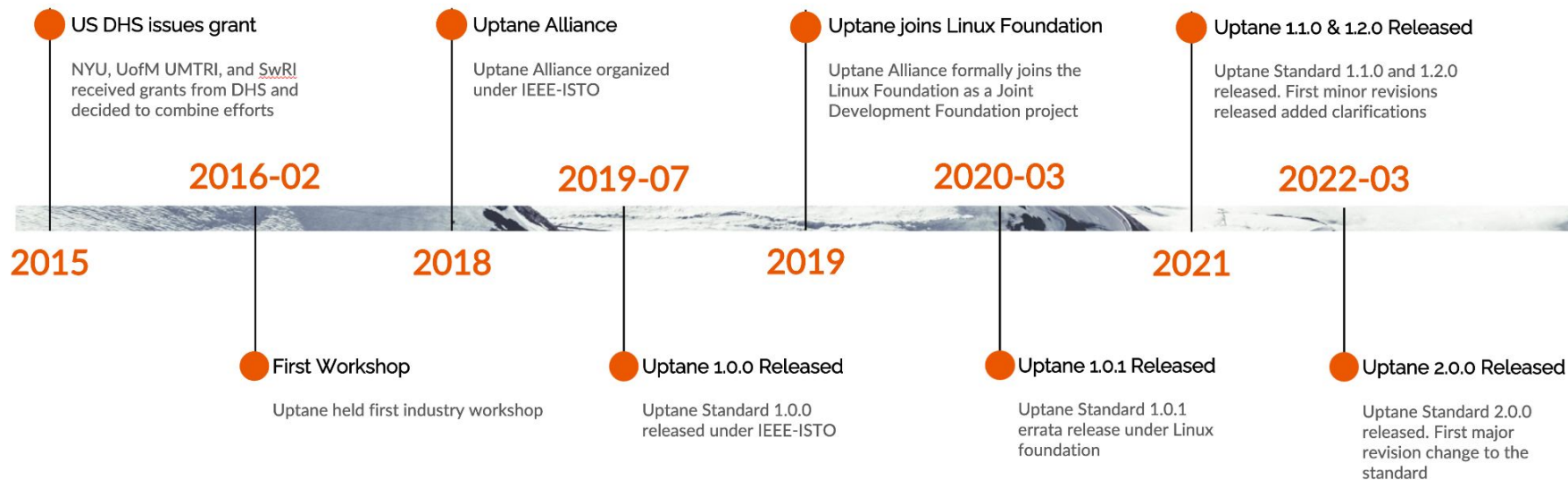


Uptane Ecosystem





Timeline for Uptane Development

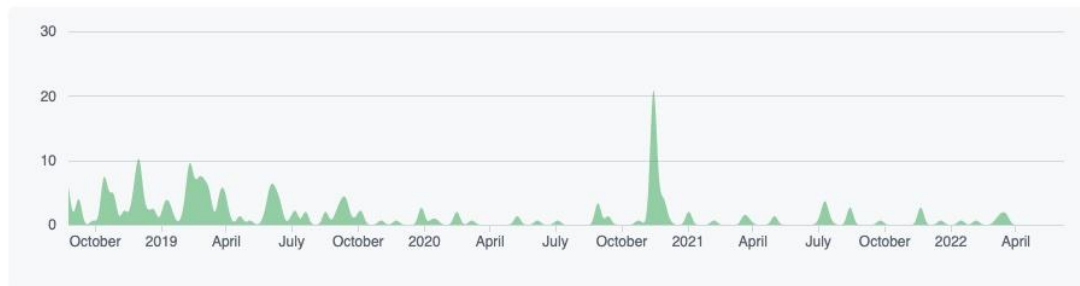




V.2.0.0: Uptane Standard for Design and Implementation

- Released March 18, 2022
- Preceded by two V.1.x minor releases and one patch release
- Versioned companion volume, *Deployment Best Practices*, added in 2021
- The Uptane Standard by the numbers:
 - 4 years
 - 400+ commits
 - 15 contributors

Contributions to master, excluding merge commits and bot accounts





What Has Changed in the Standard ?

Technical modifications *(Not a comprehensive list)*

- Moved specification of protocols, operations, usage, and formats to flexible POUFs
- Ensured Standard specifies only security critical items
- Removed recommended use of an Uptane-specific Time Server, allowing users to identify a secure source of time
- Required ECUs to check that the length of the update image matches the length listed in the metadata
- Clarified relationship between Primaries and Secondaries if a vehicle has multiple Primaries
- Clarified that, if Primary has received multiple hashes for a given update image binary via the Targets role, then it SHALL verify every hash for this image.



What Has Changed in the Standard ?

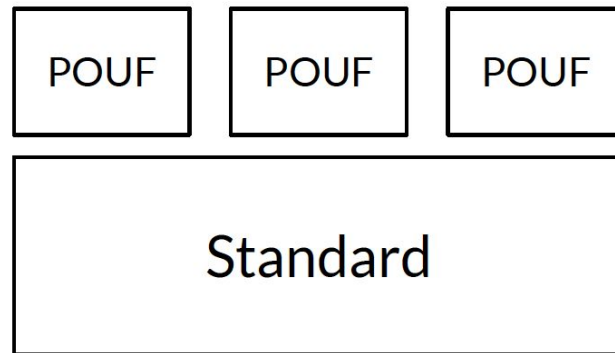
Editorial and Policy Changes

- Established formal policy for approving release of new versions of the Standard
- Clarified potential confusion over use of certain terms, including “secondary storage” and “unique” (as applied to signing keys)
- Established Uptane style sheet to ensure consistency of capitalization, spelling, and punctuation use across the Standard
- Selected “SHALL” as the word to describe mandated actions in the Standard to ensure consistency
- Restricted use of imperatives in the Standard to instances where they are required for interoperation or to limit potentially insecure behavior.
- Clarified processes for Standard release and issue resolution



Uptane POUFs (Protocols, Operations, Usage, and Formats)

- A profile layer on top of the Uptane Standard
- Allows for interoperable Uptane implementations
- Describes an implementation
 - Choices made from the Uptane Standard and Deployment Considerations
 - Networking information, file storage and data definitions

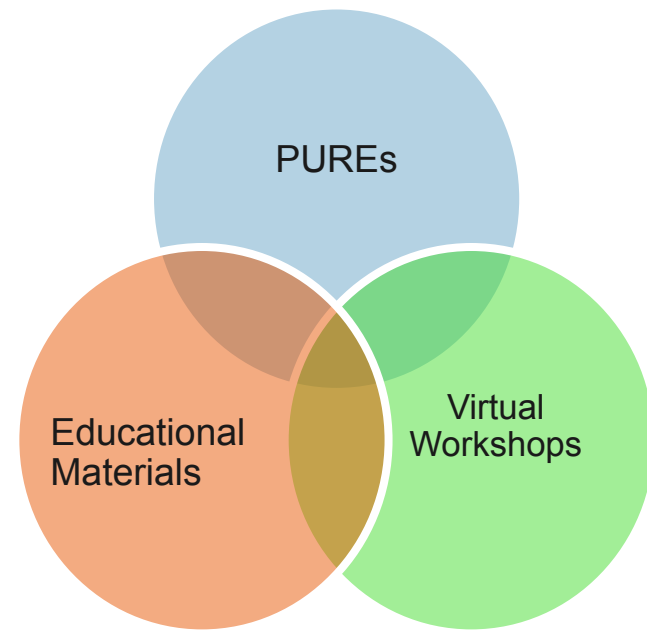




Outreach Beyond Standards

Recent Uptane achievements in addition to releases:

- Establishing policy for accepting proposed contributions to the Standard
- Publishing whitepapers, videos, and tutorials to address new or emerging areas of concern in cybersecurity
- Sharing Uptane stories across borders





PUREs

- Modeled on TAPs from The Update Framework
- A formal method for the community to propose additions or modifications of the Uptane Standard
- Two PUREs approved to date



Proposed Uptane Revisions and Enhancements (PUREs)

Accepted

- [PURE 1: Title: PURE Purpose and Guidelines](#)
- [PURE 2: Title: Offline Updates](#)

Draft

Rejected

License

This work is currently licensed and distributed under the [Apache License, Version 2.0](#).





Educational Materials

- Whitepapers, Videos, Tutorials, etc.
- Communicating emerging issues in automotive cybersecurity
- Promoting awareness of cybersecurity issues to the automotive community
- Addressing software supply chain issues
- Topics for upcoming whitepapers: Compliance with regulations and standards, Security issues in the use of aftermarket materials, Transitioning to Uptane





Industry Workshops

- Offering virtual workshops to reach a global audience at only a fraction of the cost of in-person meetings
- Effective option for a Covid-impacted world
- Two workshops have already been held, one for North America in May 2020 and another for Europe in September 2021
- Soliciting community input on how and when to hold Industry Workshops



Uptane Mitigates Specific Attacks

Marina Moore
New York University
(CEST 13:25-13:55)



Uptane protects against four categories of attacks

- Read updates through **Eavesdropping attacks**
- Deny updates through **Drop-request attacks, Slow retrieval attacks, Freeze attacks, Partial bundle installation attacks**
- Deny functionality through **Rollback attacks, Endless data attacks, Mixed-bundles attacks, Mix-and-match attacks**
- Loss of control leading to **Arbitrary software attacks**



Freeze Attack

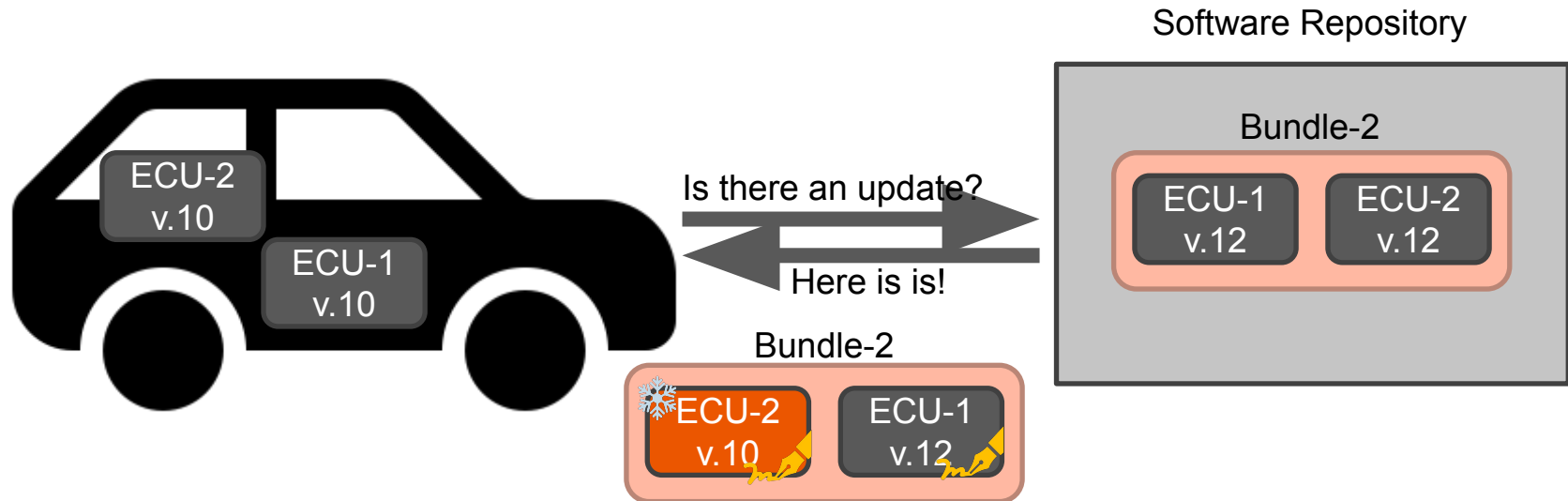




Uptane Protections: Freeze Attack

- The Timestamp metadata includes a timestamp with a short expiration date
- Vehicle can detect that the timestamp is invalid

Partial Freeze Attack





Uptane Protections: Partial Freeze Attack

- Snapshot metadata lists all current targets metadata
- Timestamp signs hash of snapshot



Rollback Attack



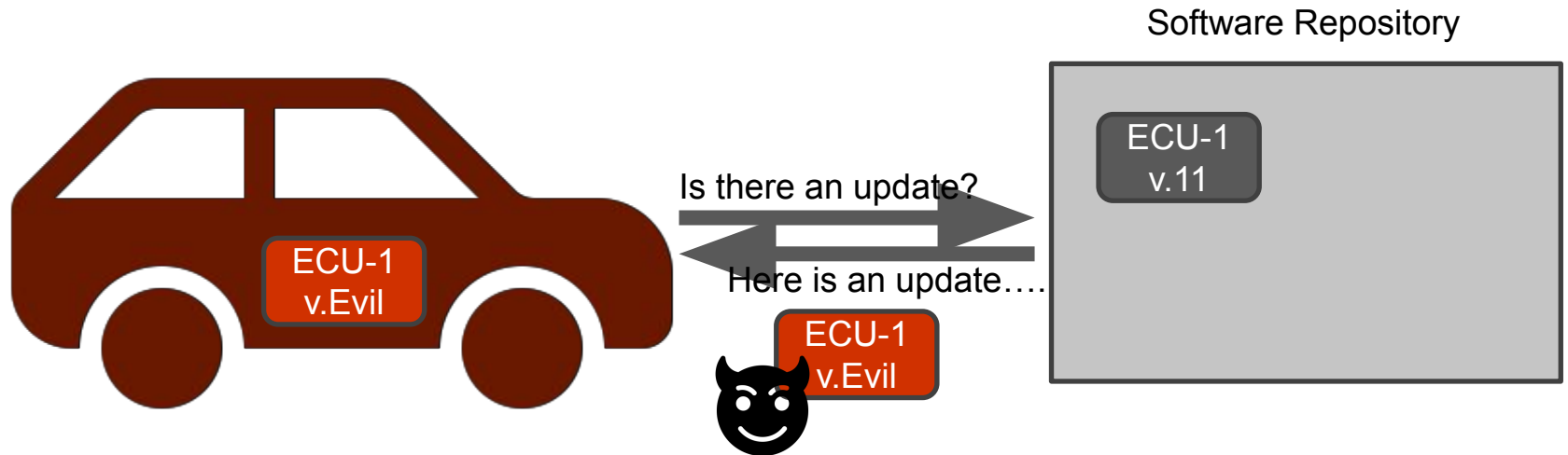


Uptane Protections: Rollback Attack

- Snapshot metadata lists all current targets metadata
- Vehicle checks that all versions numbers are strictly increasing



Arbitrary Software Attack





Uptane Protections: Arbitrary Software Attack

- Targets metadata signs the contents of all updates
- Signed by both repositories
 - Image repository uses offline keys
 - Director repository directs updates

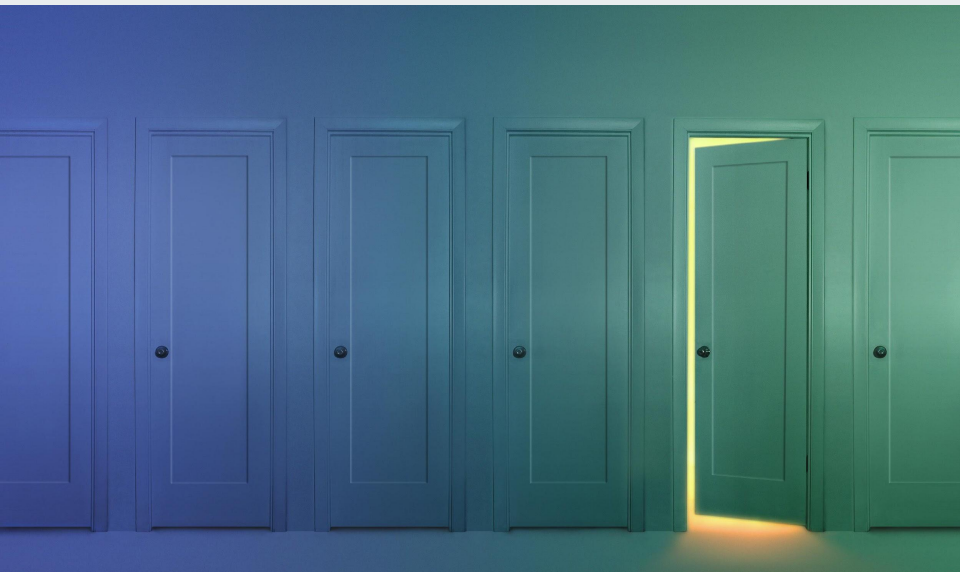
Fundamental Security Assumptions and Best Practices

Phil Lapczynski, Renesas Electronics
(CEST 13:55-14:30)



Does Uptane Consider Device Security Issues?

- 1) Secure interfaces (serial, JTAG, etc)
- 2) Strong security algorithms & assumptions
- 3) Sufficient entropy (validated TRNG is best)
- 4) Protecting keys
- 5) Prevent unwanted data leaks
- 6) Securing the time source



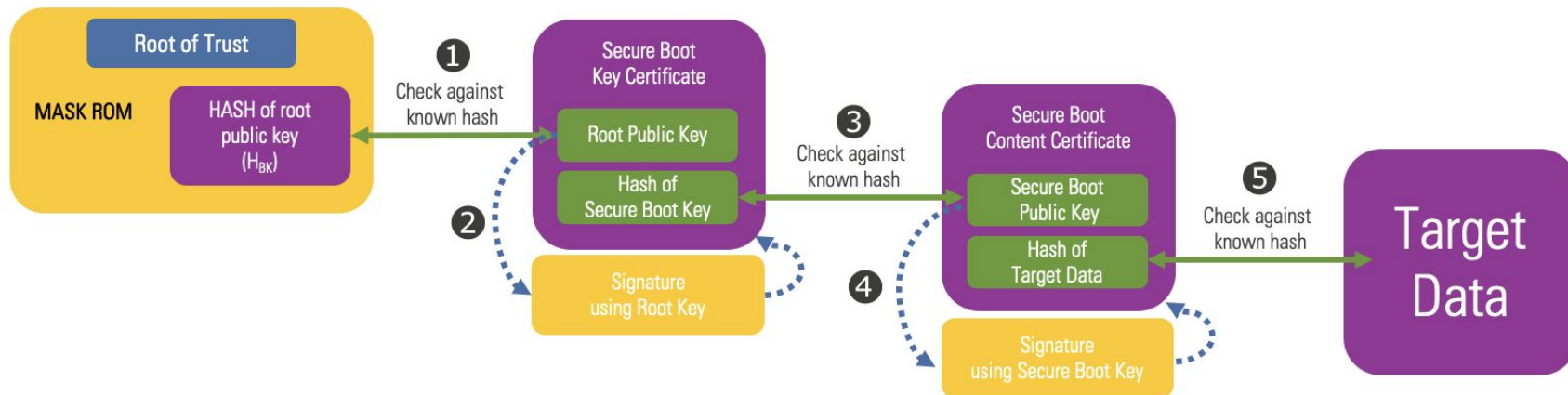


Secure Device Provisioning

How do you initially provision software (including Uptane) on to a device?

- Devices need a mechanism to securely program the initial software and root keys
- Usually root keys are fused in device or are set in OTP flash
- How to protect those keys?

Hardware Assisted Secure Boot



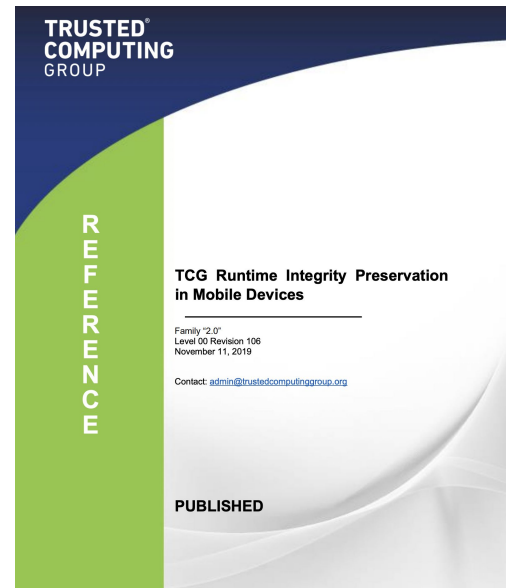
Example Secure Boot Sequence



Hardware Assisted Runtime Integrity

Once secure boot is complete, how to maintain integrity during runtime?

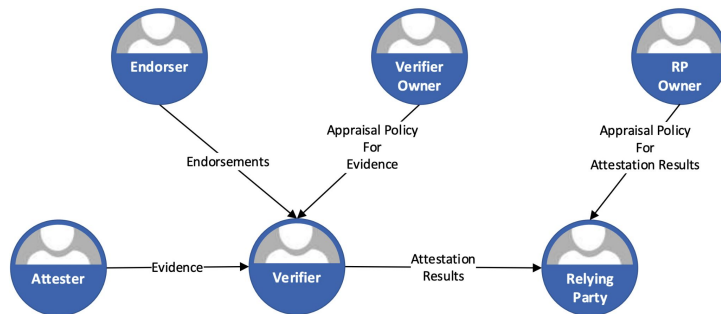
- Runtime integrity tries to answer this question



Hardware Assisted Device Attestation

How to determine when a device can be trusted?

- TCG DICE Attestation Architecture
- IETF Remote Attestation Procedures Architecture (RATS)



Remote Attestation Procedures Architecture
draft-ietf-rats-architecture-22

Status: [IESG evaluation record](#) [IESG writeups](#) [Email expansions](#) [History](#)

Versions:

00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22

draft-birkholz-attestation-terminology 00 01 02
draft-birkholz-rats-architecture 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22
draft-thaler-rats-architecture 01
draft-ietf-rats-architecture 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22

Document

Type	Active Internet-Draft (rats WG)
Authors	Henk Birkholz , Dave Thaler , Michael Richardson , Ned Smith , Wei Pan
Last updated	2022-09-28
Replaces	draft-thaler-rats-architecture , draft-birkholz-rats-architecture
Stream	Internet Engineering Task Force (IETF)
Intended RFC status	Informational
Formats	txt html xml fontdata pdf bibxml
Reviews	OSDIR Last Call review (of -21) has info SECDIR Last Call review (of -21) has info GENART Last Call review (of -21) Ready with Nits

TRUSTED COMPUTING GROUP

SPECIFICATION

DICE Attestation Architecture

Version 1.00
Revision 0.23
March 1, 2021

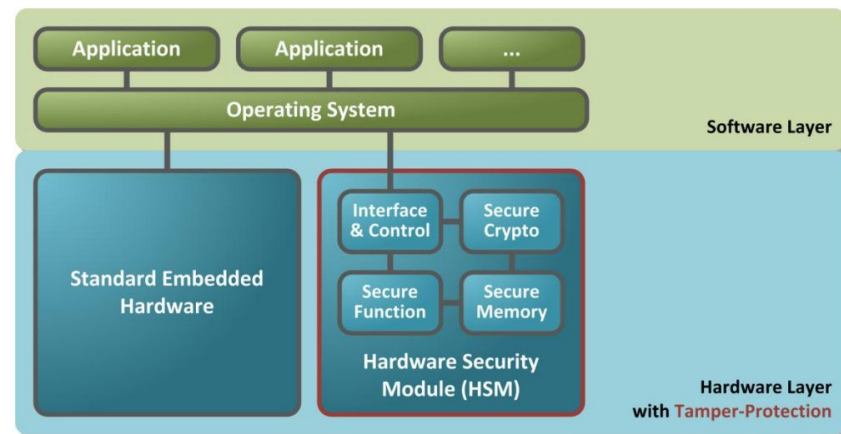
Contact: admin@trustedcomputinggroup.org

PUBLISHED



Hardware Protected Security Environments

- Secure segmentation
- Crypto acceleration
- Secure storage of keys
- Secure boot
- [SAE J3101](#)



ECU Hardening

- Hardware resistance to fault injection attacks
- Secure coding practices to resist FI attacks
- Tamper protection
- Constant time algorithms
- Security testing of update system





Tool Hardening

Make tools unappealing for attackers

- No secret keys in tool
- No secret algorithms in tool
- Authenticate user roles
- Authenticate communication with ECU
- Authenticate communication with backend
- Use end-to-end encryption of binaries (Tool doesn't need the unencrypted binary image)
- Take advantage of certificates



Break

Resume at **15:15**



Part 2: The Road Ahead

Emerging Challenges for Uptane

15:15-16:45 CEST



International Standards and National and Regional Regulations

15:15-15:45 CEST



ISO/SAE 21434 and ISO24089

Suzanne Lightman, NIST

UN ECE WP.29 R155 and R156

Nick Russell, Blackberry

Emerging Critical Issues

15:45-16:20 CEST



Aftermarket Issues

Cameron R. Mott, SWRI



Aftermarket - Current Scenario

Aftermarket companies (think of Mopar and AutoCare) are providing services and equipment that are outside of the OEM sphere

- Following end-of-life support from OEMs
- Adding functionality to a vehicle through aftermarket vendors

Owners/car enthusiasts are customizing cars

- Following the right to repair
- Successfully reverse-engineering
- Configuration adjustments





Concerns

Aftermarket concerns

- Responsibility for component operation
- Integration with existing components

OEM concerns

- Responsibility for safe operation of entire vehicle
 - Including right to repair
 - After end-of-life (minor)
- IP protection

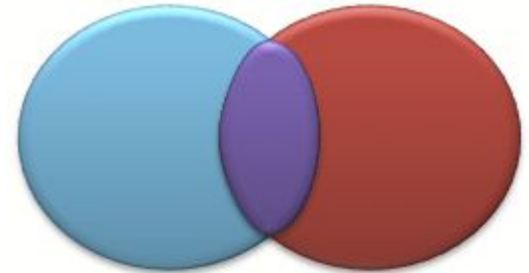
Shared concerns

- Secure the vehicle from both electronic and physical intrusion



Alternatives

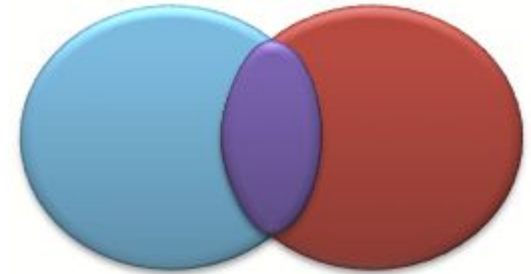
1. Aftermarket/owner operates independently
 - a. OEM and aftermarket/owner operate mutually exclusive ECUs
 - b. May not have their own Primary
2. Responsibility (keys, code) is shifted at specific times
 - a. End of life
 - i. Ownership of update servers would need to be delegated (modify Uptane Standard)
 - b. Upon customization of critical safety functions
 - i. Perhaps a digital “void warranty” if safety critical firmware is modified
 - c. Authorized custom shop is given a key role that allows specific adjustments
3. Aftermarket/customer is integrated
 - a. Leverage existing Director/Image servers
 - i. Aftermarket may be an optional supplier
 - b. Operate their own servers
 - i. Authorize additional servers for specific functionality/ECUs





Alternatives

1. Aftermarket/owner operates independently
 - a. OEM and aftermarket/owner operate mutually exclusive ECUs
 - b. May not have their own Primary
2. Responsibility (keys, code) is shifted at specific times
 - a. End of life
 - i. Ownership of update servers would need to be delegated (modify Uptane Standard)
 - b. Upon customization of critical safety functions
 - i. Perhaps a digital “void warranty” if safety critical firmware is modified
 - c. Authorized custom shop is given a key role that allows specific adjustments
3. Aftermarket/customer is integrated
 - a. Leverage existing Director/Image servers
 - i. Aftermarket may be an optional supplier
 - b. Operate their own servers
 - i. Authorize additional servers for specific functionality/ECUs





Next steps

1. Identify/verify concerns of different stakeholders
2. Rank/identify new alternatives
3. Recommend modifications to Uptane standard

Software Supply Chain Security

Aditya Sirish A. Yelgundhalli, NYU



Software Bill of Materials (SBOM)

SBOMs have emerged as key building blocks in software supply chain security

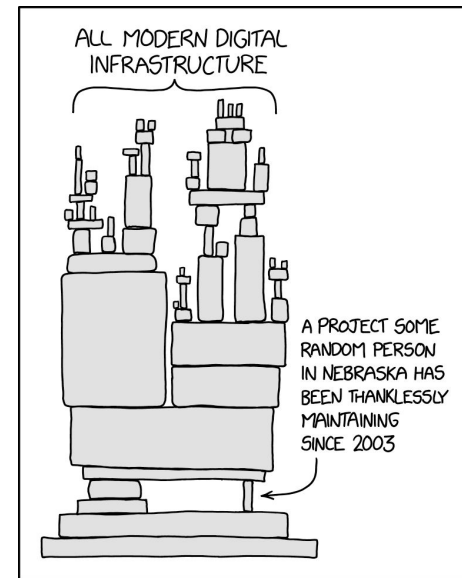
It is a nested inventory, a list of ingredients that make up software artifacts

Regulations like Executive Order 14028 on improving the nation's cybersecurity call for them

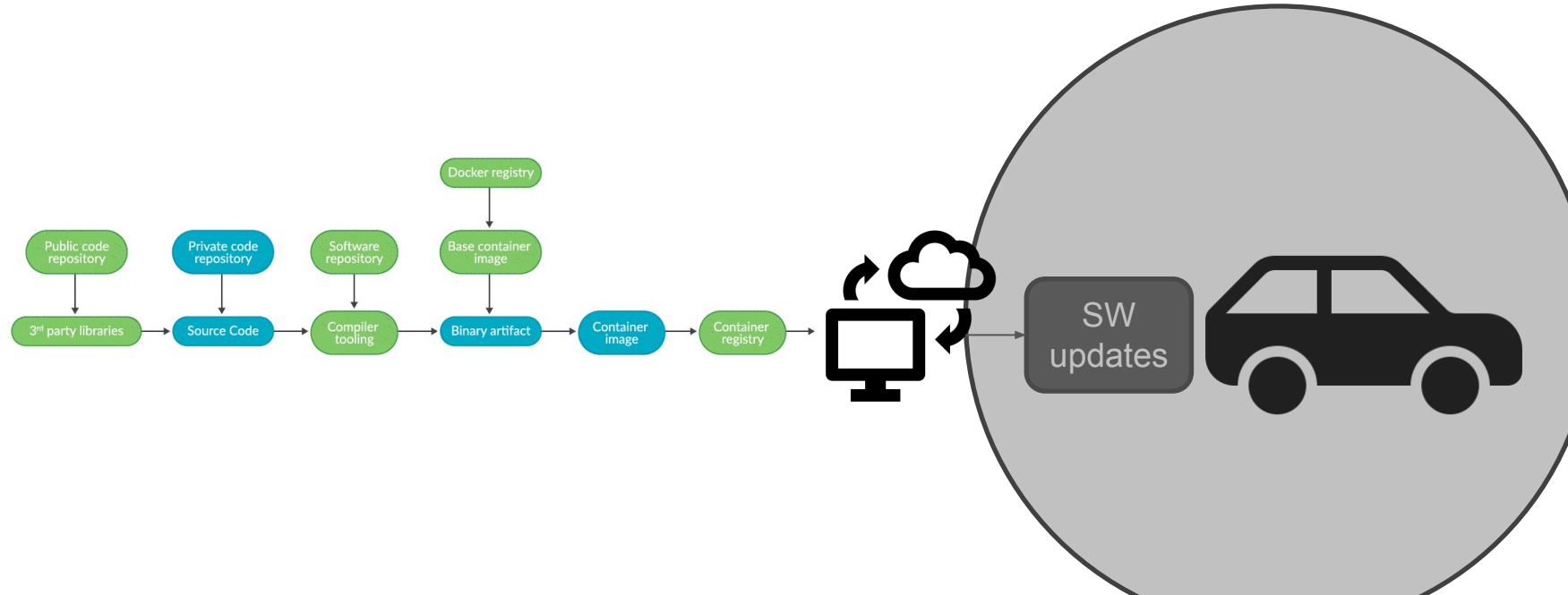
References:

[CISA SBOM-A-RAMA](#)

[NTIA - The Minimum Elements For a Software Bill of Materials \(SBOM\)](#)



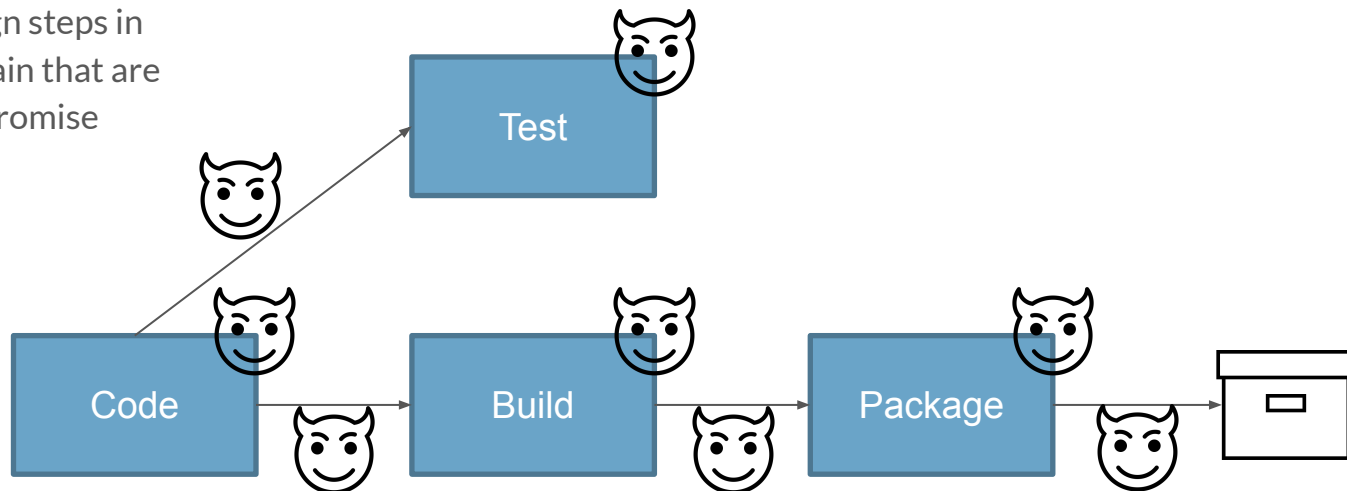
Small Part of Overall Software Supply Chain





Securing the Software Supply Chain

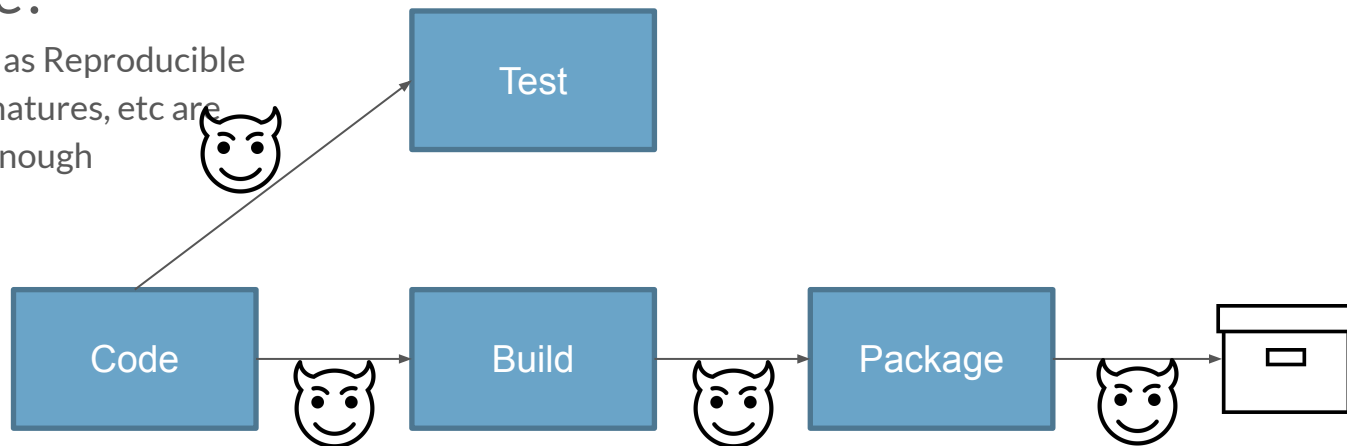
Aim to verifiably sign steps in software supply chain that are vulnerable to compromise



Gaps Between Steps in Supply Chain?

Compliance?

Spot solutions such as Reproducible Builds, Commit Signatures, etc are necessary but not enough

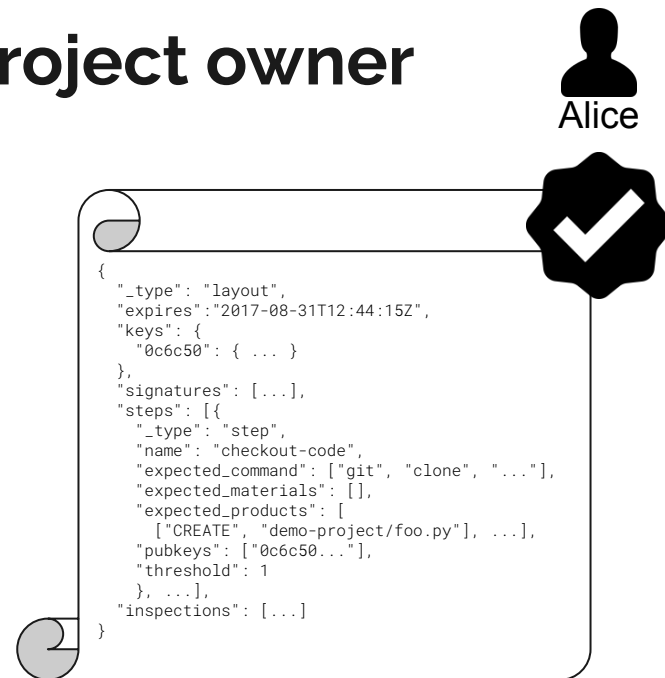
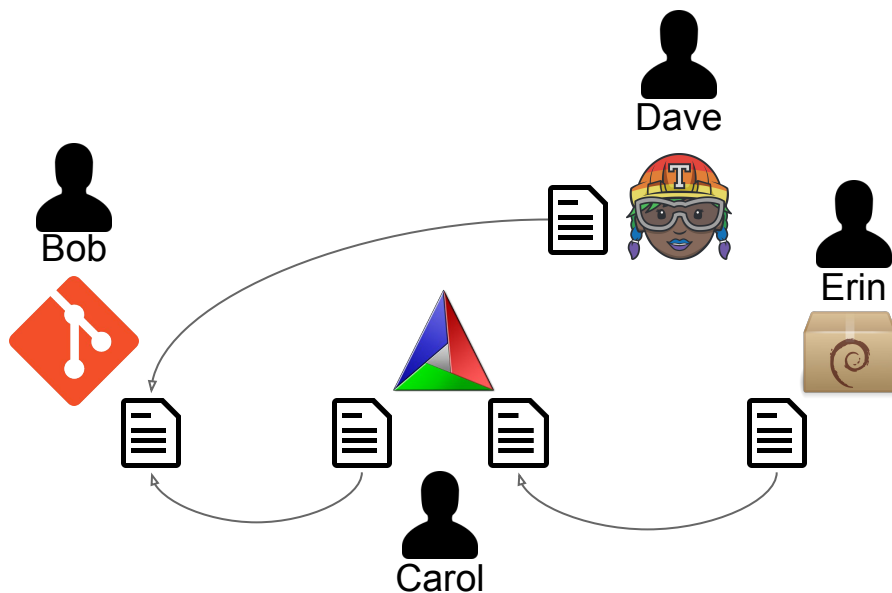




in-toto

- Verifiably define the steps of the software supply chain
- Verifiably define the authorized actors
- Guarantee everything happens according to definition and nothing else

in-toto -- Layout -- Signed by project owner



in-toto -- Links -- Signed evidence for each step

```
$ in-toto-run -- ./do-the-supply-chain-step
```



```
{
  "_type": "Link",
  "name": "code",
  "byproducts": {
    "stderr": "", "stdout": ""
  },
  "command": [...],
  "materials": {},
  "products": {
    "foo": {"sha256": "..."}
  },
  "return_value": 0,
  "signatures": [...]
}
```



```
{
  "_type": "Link",
  "name": "build",
  "byproducts": {
    "stderr": "", "stdout": ""
  },
  "command": [...],
  "materials": {...},
  "products": {
    "foo": {"sha256": "..."}
  },
  "return_value": 0,
  "signatures": [...]
}
```



```
{
  "_type": "Link",
  "name": "build",
  "byproducts": {
    "stderr": "", "stdout": ""
  },
  "command": [...],
  "materials": {},
  "products": {
    "foo": {"sha256": "..."}
  },
  "return_value": 0,
  "signatures": [...]
}
```

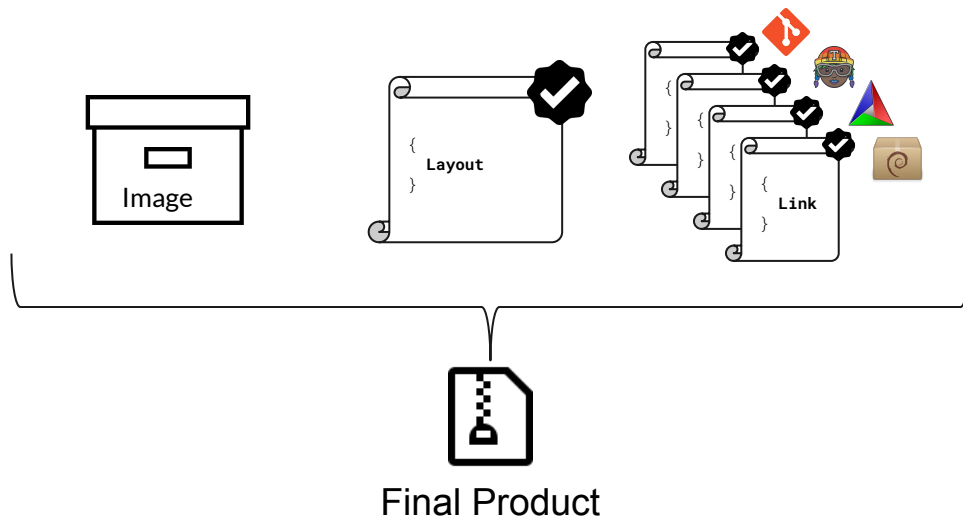


```
{
  "_type": "Link",
  "name": "build",
  "byproducts": {
    "stderr": "", "stdout": ""
  },
  "command": [...],
  "materials": {},
  "products": {
    "in-toto/.git/HEAD": {"sha256": "..."}
  },
  "return_value": 0,
  "signatures": [...]
}
```

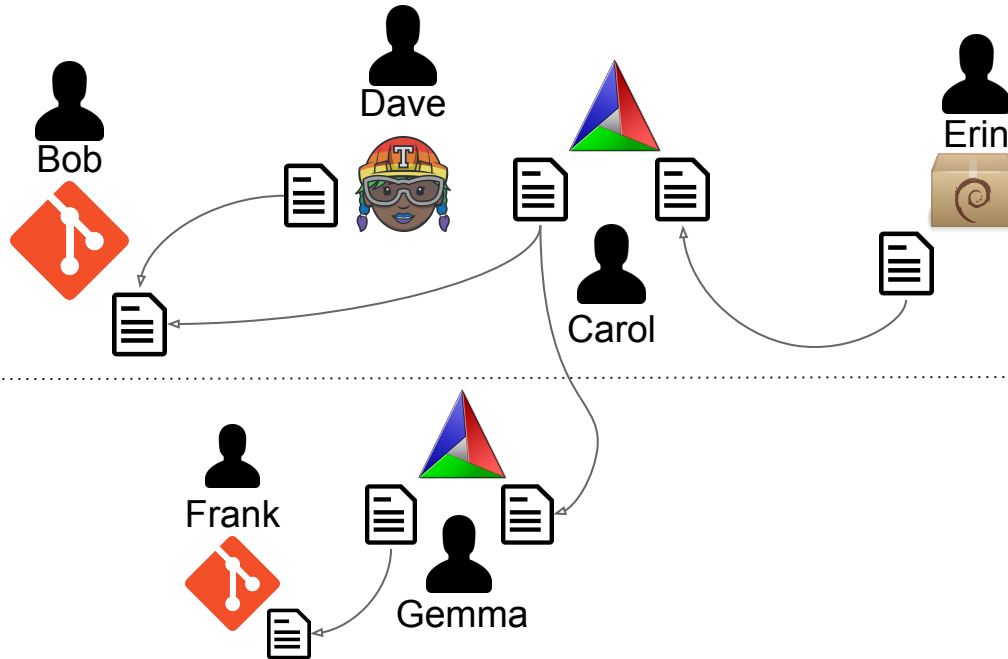



in-toto Verification

```
$ in-toto-verify --layout <layout> --key <pub key>
```



What about vendor supply chains?

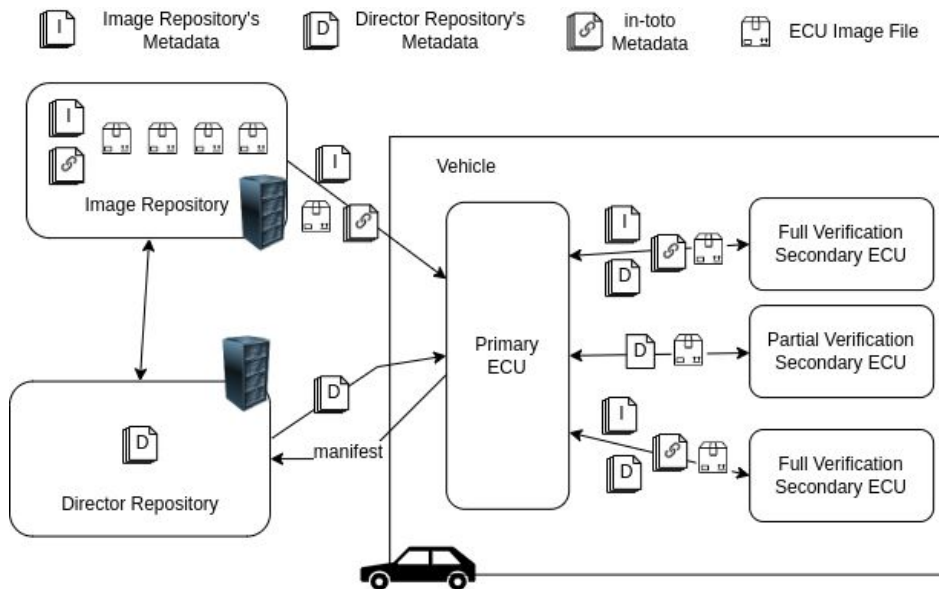




in-toto Integrations and Adoptions



Scudo = in-toto + Uptane





Scudo = in-toto + Uptane

- Securely distributes in-toto metadata for all images to vehicles for verification before images are installed
- Detailed ECU responsibilities for in-toto verification on vehicles
- Includes options where a powerful ECU can verify the metadata pertaining to a less powerful ECU
- Provides a stop-gap option for vehicles with no sufficiently powerful ECUs
- Includes support for vendor supply chains



Scudo = in-toto + Uptane

Successful integrations of in-toto and TUF in use in production:

<https://www.datadoghq.com/blog/engineering/secure-publication-of-datadog-agent-integrations-with-tuf-and-in-toto/>

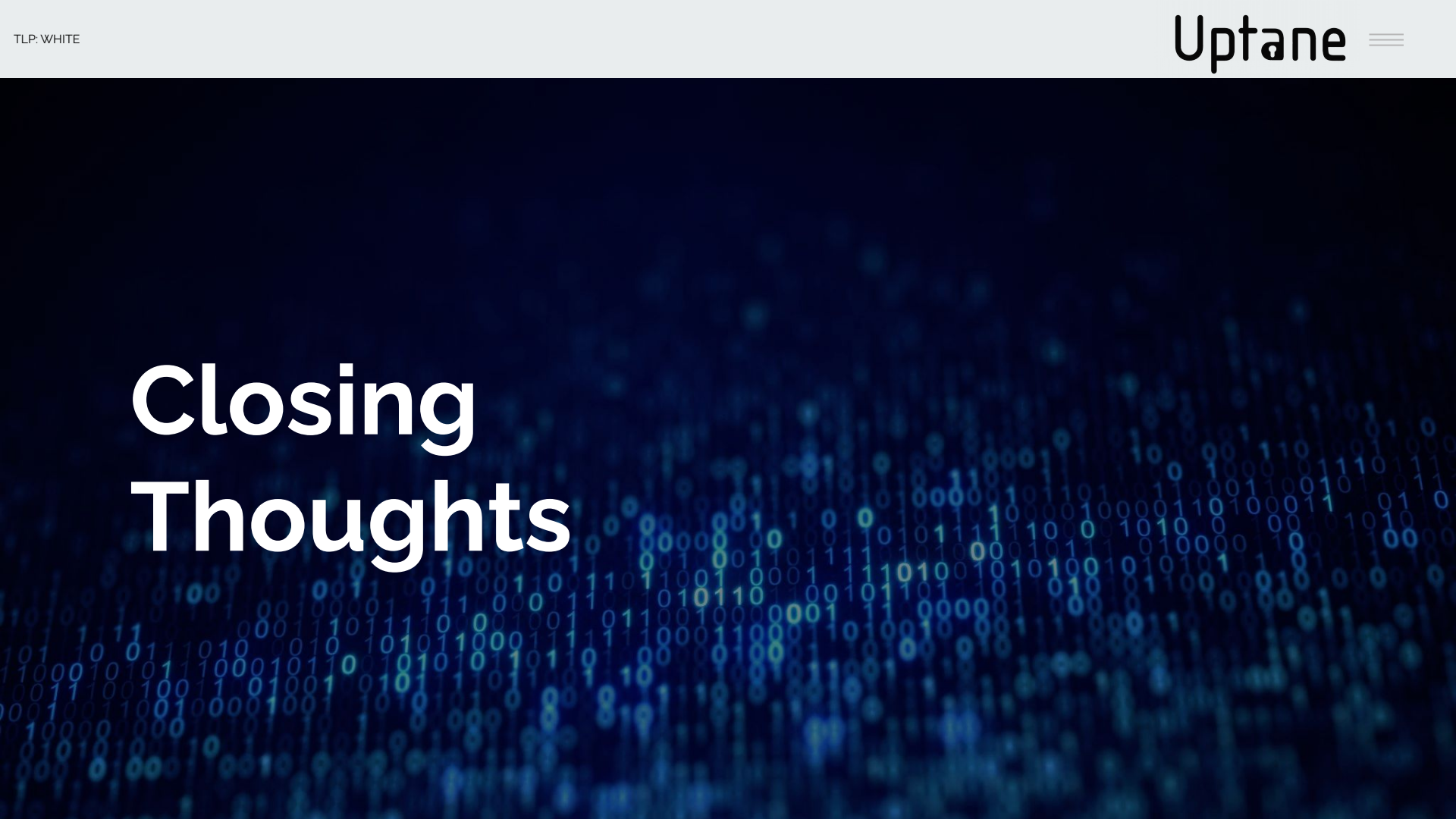
Integrated in-toto with Uptane considers the nuances of the auto industry:

<https://uptane.github.io/papers/scudo-whitepaper.pdf>

More advanced specification of Scudo available as an upcoming Uptane PURE:

<https://github.com/uptane/pures/pull/9>

Closing Thoughts





Conclusions

- At this time, the Uptane Standard is mature, the requirements are well thought through (but not perfect), and it has been deployed in real-world systems.
- Uptane can be used as guidance to develop and deploy secure SOTA. Using some Uptane ideas is better than not using them at all.
- But, how Uptane requirements map to security properties is not explicitly documented, and it requires extensive understanding of Uptane to establish this mapping.



Conclusions

- Uptane will continue to refine and improve the specification, increasingly focusing on motivation and education
 - Mapping of threats to Uptane modules/requirements to understand what individual Uptane modules/requirements contribute to overall system security (similar to a TARA)
 - Provide strategies to transition from existing SOTA systems to Uptane systems (or improve existing systems with ideas from Uptane)
 - Refine guidance in the Deployment Best Practices
 - Focus on aftermarket devices/systems



Uptane Roadmap Planning

1st quarter 2023	2nd quarter 2023	3rd quarter 2023	4th quarter 2023	1st quarter 2024	2nd quarter 2024
Release V.2.1 of Standard and Deployment Best Practices	Hold in-person community meeting (North America)	Hold virtual workshop (Europe)	Release V.2.2 of Standard/ Deployment	Hold virtual community meeting	Release V.3.0.0 of Standard/ Deployment
Release whitepaper on transitioning to Uptane		Release whitepaper on compliance with regulations and standards		Release whitepaper on aftermarket materials	Hold in-person community meeting (North America)
Hold virtual community meeting focused on establishing a roadmap for the project					

Please contact us if you are interested to join, contribute and/or learn more:

<https://uptane.github.io/participate.html>



Thank you.

