Trusted Types deployment
at Meta, Microsoft & Google

Bartosz Niemczura, Jun Kokatsu, Krzysztof Kotowicz

Meta Platforms

- **Rollout Approach**
  - Breaking up large applications into smaller pieces/endpoints
  - Report-Only used for identifying violations; switching to enforcement after a few weeks with no violations. Filtering out known false-positives (e.g. malware or extensions)
  - Manual approach for addressing violations (removing old/legacy code; updating the code to use React or XHP; migrating to use proper Trusted Types with specific policies)

- **Status Update:**
  - TT used by default for new products/new domains
  - In progress: rolling out TT for existing products/domains; currently about 50% of applications enforce TT.
  - Major applications: facebook.com, instagram.com and others in progress

- **Challenges:**
  - Upgrading/Patching Open Source Libraries
  - Problems with one-page applications
  - Problem with widgets causing problems on multiple sites
  - Wider browser support of Trusted Types

Google - rollout prerequisites

- **Static checks** for TT compliance during JS/TS compilation
  (prevents production breaks)

  GitHub - google/tsec

- Safe **TT builder** library
  (sanitizing, escaping, HTML construction, building from literals)

  GitHub - google/safevalues

- **Strict CSP** compliance (no server-side XSS, few eval calls, no unsafe-inline)
- CSP **report collection pipeline** (deduplication, noise removal)

Google - TT for new applications

1. Report-only CSP
2. Fix TT violations in the frameworks / stacks
3. Enforce TT in test suites
4. Add TT enforcement to baseline requirements for a stack

Google - TT for existing applications

1. Report-only CSP
    - identify violations (duh!)
    - identify **already compliant applications (83%)**
2. Group applications into rollout waves
3. For every wave:
    a. Fix violations - **few!**
        i. Only **31 code changes for 880 application migrations** on the most common app stack
        ii. **184 distinct issues** identified in all migrations so far
    b. Enforce TT in test suites
    c. CSP report-only → monitor → external

Google - status

Several app stacks enforce TT by default:

- **~25%** all security tier 0,1 and 2 applications
- **~80** sensitive domains under `google.com`
- **24%** of `text/html` traffic to `*.google.com`
- **0** production breakages

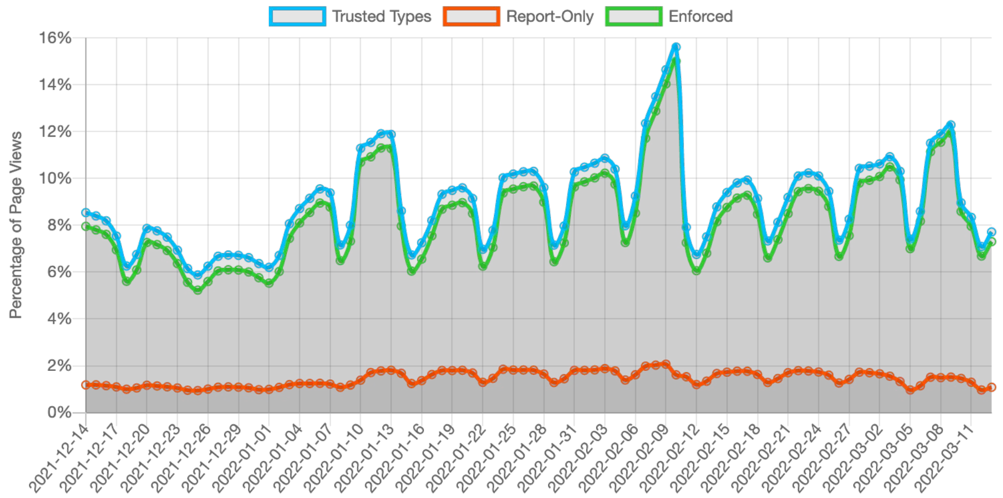
TT migrations drive other XSS reduction efforts:

- # of XSS halved in 2020 and 2021
- **0 DOM XSS** in all TT applications

Code organically gets more secure:

- Violations surface sketchy code
- TT drive engineers towards more secure libraries

Trusted Types enforcement globally



2% → 10% pageviews

over last 12 months

Source: https://mitigation.supply/

Google - challenges

Upstream patches sometimes blocked on multi-browser support:

> https://github.com/highlightjs/highlight.js/pull/3281
> *"we wait for the spec to mature and to be implemented by additional browser engines"*
>
> *https://github.com/microsoft/TypeScript-DOM-lib-generator/pull/1246*
> *"This needs multiple implementor interest."*

Migrations could be simplified with **new features** - fromLiteral, Sanitizer API

Trusted Types deployment in Microsoft Edge

WebUI (browser internal pages):

- Trusted Types enabled by default with policy enforcement (e.g. edge://settings).
  - Perfect Types (i.e. trusted-types ‘none’) by default.
- Onboarding new JS libraries require Trusted Types support.
- Using tsec for compile time validation for most WebUI packages.
  - Security team’s PR approval is required for changes in exemption_list.json. This is how we monitor for changes that might bypass Trusted Types such as pass-through TT policy.

Trusted Types deployment in MS websites

- Some Microsoft websites are being integrated to Edge (e.g. Bing Collections iframed in WebUI).
    - Similar to what Chrome does today (e.g. chrome://whats-new).
- Strict CSP and Trusted Types are used as a security validation mechanism for such integration (verified via CSP Embedded Enforcement).
    - E.g.:
      ```
      <iframe src="https://xyz.bing.example"
       csp="script-src 'nonce-required' 'strict-dynamic';
                base-uri 'self';
                require-trusted-types-for 'script';
                trusted-types default;">
      </iframe>
      ```
    - This is basically a [SecureContext=Injection].

Challenges around Microsoft Edge

- Missing some primitives such as *fromLiteral* and *setHTML*.
    - Makes contributing to 1st or 3rd party libraries a little difficult.
    - Required to create more TT policies.
- Since Trusted Types is used as a security validation mechanism (and there is not much control over other MS org codebase), unfixed (or by design) Trusted Types bypasses becomes a concern
    - E.g. XHR document response might be used to bypass Trusted Types by teams without our knowledge.
    - Shared this information in Cursed Types.
- Developer education.
    - Not many educational materials on how to (beautifully) fix TT violations.

Trusted Types deployment in MS Teams

- **Status:**
  - Work in progress. We started looking more closely at TT in Nov 2021.
  - *trusted-types* directive with policy names is enforced in parts of the app.
  - *require-trusted-types-for* directive is reported from selected dev and test environments.

- **Approach used:**
  - We rely heavily on CSP and flight changes via Report-Only header. This allows us to make gradual progress without risking outages.
  - Part of the app (our own code) is integrated with tsec. This is to prevent specific new debt (e.g. DOMParser.parseFromString).

- **Challenges:**
  - Missing best practices for Trusted Types policy names.
    - The integration in VS Code introduced policy names such as *amdLoader*, *standaloneColorizer*, *notebookRenderer* without any ability to customize. This is a nightmare to maintain in a large ecosystem. MS Teams is a hub for many apps, we even bring in parts of VS Code. It should be a best practice to at least prepend package name (e.g. @microsoft/vscode#amdLoader).
  - Missing type information in TypeScript.
  - Can CSP Embedded Enforcement have a Report-only mode?