[These sections were copied verbatim from the existing charter. Please propose updates.]

Mission

The **mission** of the Web Application Security Working Group is to develop security and policy mechanisms to improve the security of Web Applications, and enable secure cross-origin communication.

Scope

Modern Web Applications are composed of many parts and technologies. They may transclude, reference or have information flows between resources at the same, related or different origins. Due to the historically coarse-grained nature of the security boundaries and principals defined for such applications, they can be very difficult to secure.

In particular, application authors desire uniform policy mechanisms to allow application components to drop privileges and reduce the chance they will be exploited, or that exploits will compromise other content, to isolate themselves from vulnerabilities in content that might otherwise be within the same security boundaries, and to communicate securely across security boundaries. These issues are especially relevant for the many web applications which incorporate other web application resources (mashups). That is, they comprise multiple origins (i.e., security principals).

Areas of scope for this working group include:

Vulnerability Mitigation

Vulnerabilities are inevitable in sufficiently complex applications. The WG will work on mechanisms to reduce the scope, exploitability and impact of common vulnerabilities and vulnerability classes in web applications, especially script injection / XSS.

Attack Surface Reduction

The WG will design mechanisms to:

- Allow applications to restrict or forbid potentially dangerous features which they do not intend to use
- Govern information and content flows into and out of an application
- Allow applications to isolate themselves from other content which may contain unrelated vulnerabilities
- Sandbox potentially untrusted content and allow it to be interacted with more safely
- Uniquely identify application content such that unauthorized modifications may be detected and prevented
- Replace or augment injection-prone APIs in the browser with safer alternatives using strategies such as sanitization, strict contextual autoescaping, and other validation and encoding strategies currently employed by server-side code.

Secure Mashups

Several mechanisms for secure resource sharing and messaging across origins exist or are being specified, but several common and desirable use cases are not covered by existing work, such as:

- Allowing child IFRAMEs to protect themselves from "clickjacking"
- Providing labeled information flows and confinement properties to enable secure mashups. This is especially relevant for, e.g. applications communicating between security principals with different user-granted permissions (e.g. geolocation)

Manageability

Given the ad-hoc nature in which many important security features of the Web have evolved, providing uniformly secure experiences to users is difficult for developers. The WG will document and create uniform experiences for several undefined areas of major utility, including:

- Treatment of Mixed HTTPS/HTTP Content and defining Secure/Authenticated Origins for purposes of user experience, content inclusion/transclusion and other information flows, and for features which require a verifiably secure environment
- Providing hinting and direct support for credential managers, whether integrated into the user-agent or 3rd-party, to assist users in managing the complexities of secure passwords
- Application awareness of features which may require explicit user permission to enable.

The Web Security Model

The WG may be called on to advise other WGs or the TAG on the fundamental security model of the Web Platform and may produce Recommendations towards the advancement of, or addressing legacy issues with, the model, such as mitigating cross-origin data leaks or side channel attacks.

In addition to developing Recommendation Track documents in support of these goals, the Web Application Security Working Group may provide review of specifications from other Working Groups, in particular as these specifications touch on chartered deliverables of this group (in particular CSP), or the Web security model, and may also develop non-normative documents in support of Web security, such as developer and user guides for its normative specifications.

Deliverables

(With notes as to status of each)

Also see status tables at:
https://github.com/w3c/WebAppSec#readme
and
https://www.w3.org/2011/webappsec/

Which might be redundant with:
https://www.w3.org/groups/wg/webappsec/publications
or
https://www.w3.org/PM/Groups/repositories.html?gid=49309
or
https://www.w3.org/PM/Groups/chairboard.html?gid=49309

Content Security Policy Level 3
FPWD; Editor's draft last update Nov 2020

Content Security Policy: Embedded Enforcement
Editor's draft, last update Oct 2018

Mixed Content
Level 1 CR 2106; Level 2 Editor's draft, last update Nov 2020

Upgrade Insecure Requests
CR in 2015; last update 2016

Secure Contexts
CR 2016; last update Feb 2020

Clear Site Data
Editor's draft; last update Nov 2017

Referrer Policy
CR 2017; last update July 2020

Credential Management API
Editor's draft, last update Oct 2019

Subresource Integrity Level 2
REC 2016; Editor's draft, last update Jan 2020

Suborigins
Last update May 2017

Origin ~~Wide~~ Policy
Still in WICG

Permissions API
Editor's draft, last update July 2020

Permissions Policy
Editor's draft, last update Dec 2020

Not in the current charter, but in the status table on GH:

Confinement with Origin Web Labels
FPWD; Editor's Draft 8 November 2017
(is it dead?)

Content Security Policy Pinning
Last update 2016, marked obsolete
(should it be marked as obsolete in the status table?)

Entry Point Regulations
Last update 2016, marked obsolete
(should it be marked as obsolete in the status table?)

Not in the current charter nor the status table:

Trusted Types

Change Password URL

Fetch Metadata

CORS for Developers
Last update Oct 2016

User Interface Security and the Visibility API
Last update June 2016
Should this point to Intersection Observer?

Content Security Policy: Embedded Enforcement
Last update Oct 2018