



AUTOMATED THREATS

Web applications

The OWASP Automated Threats to Web Applications Project is creating information and other resources to help web application owners defend against automated threats

Issue

There is a significant body of knowledge about application vulnerability types, and some general consensus about identification and naming. But issues relating to the misuse of valid functionality, which may be related to design flaws rather than implementation bugs, are less well defined. Yet these problems are seen day-in, day-out, by web application owners.

Excessive abuse of functionality is commonly mistakenly reported as application denial-of-service (DoS) attacks such as HTTP-flooding or application resource exhaustion, when in fact the DoS is a side-effect. Some examples are blog & comment spam, fake account creation, password cracking, web scraping, etc. Most of these problems seen regularly by web application owners are not listed in any OWASP Top Ten or other top issue list or dictionary.

This has contributed to inadequate visibility, and an inconsistency in naming such threats, with a consequent lack of clarity in attempts to address the issues.

OWASP Project

The OWASP Automated Threats to Web Applications Project has completed a review of reports, academic and other papers, news stories and vulnerability taxonomies/listings to identify, name and classify these attacks – threat events to web applications that are undertaken using automated actions. The initial aim is to produce an ontology providing a common language for developers, architects, operators, business owners, security engineers, purchasers and suppliers/vendors, to facilitate clear communication and help tackling the issues.

The project also intends to identify symptoms, mitigations and controls in this problem area. Like all OWASP outputs, everything is free and published using an open source license.

Use Cases

The ontology and supporting materials are expected to be useful for:

- Defining application security requirements
- Sharing intelligence within a sector
- Exchanging threat data between CERTs
- Labelling penetration test findings
- Documenting service acquisition needs
- Characterising vendor services

These are documented further on the project site.

Web Application Owner Survey

The project would like to receive real-world experience on the prevalence and naming of such threats. Please provide your experience overleaf, especially if you are responsible for the ongoing operation of web applications.

If you need to you can complete the survey multiple times – for different sectors or different applications – ask at the OWASP booth for more forms. You can be completely anonymous or provide contact details as you prefer.



“Can you please contribute your experience using the survey form overleaf, or using the online version?

Feel free to speak to me during this week’s AppSec EU conference about this project.”

Colin Watson
Project leader
colin.watson@owasp.org

OWASP Automated Threats to Web Applications

Web application owner survey

Also available online at <http://goo.gl/forms/9zKz56aAp5>

Which of the following threats affect your web application(s), and how frequently?

There may be overlaps. The names are not finalised - please provide suggestions and comments opposite, and/or by email.

Threat	Do not know term (?)	Never (N)	Rarely 1-2 per year (R)	Every quarter (Q)	Every month (M)	Every week (W)	Every day (D)	Many times/day (+)	Continuously (>)
Example never seen Description for the threat appears immediately beneath									
Credential Cracking Identify valid log in credentials by trying different values for usernames and/or passwords									
CAPTCHA Cracking Solve anti-automation tests									
Fake Account Creation Create accounts for subsequent misuse									
Credential Stuffing Mass log in attempts used to verify the validity of stolen username/password pairs									
Account Aggregation Use by an intermediary application to collect together and interact with accounts related to many other applications									
Payment Card Fraud Buy goods or obtain cash from stolen payment cards									
Card Verification Small purchases used to verify the validity of bulk stolen payment card data									
Card Cracking Identify missing payment card details by trying different values for expiry date and security code									
Data Harvesting Read application content and data, copying it elsewhere									
Content Aggregation Use of an intermediary application to collect together and consume content from many application sources, republishing it as content on the web									
Cheating Violate explicit or implicit assumption(s) about the application's use to achieve unfair individual gain, often associated with deceit and loss to some other party									
Click Fraud False click throughs									
Impression Fraud False content impressions									
Man in the Browser Compromise of web browser									
Code Alteration Modify application source code, or executing code, or configuration									
Content Spam Information addition that appears in content, or alters metrics or statistical data									
Application Consumption Misuse of the application to perform calculations, or process data, or perform other actions against other applications, hosts, or in the physical world									
Fingerprinting Requests used to illicit information about the supporting web, application and database server and framework types and versions									
Footprinting Probing and exploration to identify constituents and properties of the application									
Vulnerability Scanning Application crawling and fuzzing in an attempt to identify weaknesses and possible vulnerabilities									
Reverse Engineering Exercise an application, or part of an application, with the intent to gain insight how it is constructed and operates									
HTTP Flood DoS High rate or number of HTTP requests									
HTTP Slow DoS Partial HTTP request headers sent, or fragmented request bodies sent, or slow response read									
Web Application (Layer 7) DoS Denial of service achieved by targeting resources of the application and database servers									
User DoS Individual users locked out or unable to register/use the application									

Your comments and suggestions

Are any automated attacks that your application is affected by missing? For anything missing, how often does the threat materialise? Can you suggest names you are familiar with instead?

What industry sector does the application/your experience relate to?

E.g. Financial, Government, Health, Retail, Technology.

Can you tell us anything about the web application(s) this information refers to?

For example, whether users can log in, whether payments are collected, the general type (e.g. a conventional website, an API, a single page application), types of users (e.g. citizens, employees, customers, clients), primary user geographic locations (e.g. worldwide, Asia, Iceland) and its scale/scope.

What is your role in relation to the application(s)?

E.g. Business owner, Operations, DevOps lead, CIO, CTO, CISO.