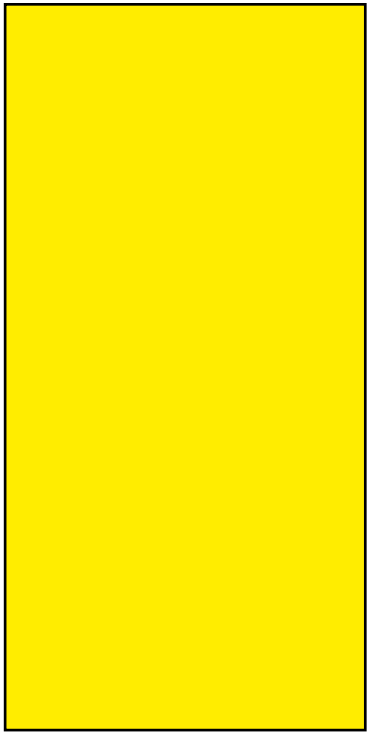


The Automated Threat Handbook

Web Applications

The Automated Threat Handbook provides actionable information and resources to help defend against automated threats to web applications.

Author

Colin Watson

Other Project Contributors

???, ???

Version v0.72 DRAFT published 6th July 2015

Contents

Preface 1

Terminology 2

Introduction 3

Research 5

The Ontology 7

Figure 1: Threat Eventss, ordered by ascending identity code 8

Figure 2: Threat Events, ordered by ascending name 9

Figure 3: WASC Threat Classification view of the Threat Events 10

Figure 4: Mitre CAPEC view of the Threat Events 11

Use Case Scenarios 17

Project Details 20

Handbook Roadmap 21

Automated Threat Event Reference 22

Preface

Most web applications are not under a constant state of compromise, regardless of whether weaknesses and vulnerabilities are present. However, attackers are still using the software in a manner that causes significant pain to the owners/operators, and sometimes also the users.

Previous work on OWASP AppSensor (application-specific attack detection and response) has identified 50 or so types of detection points, and I had speculated about which detection points would be most beneficial to implement first. All AppSensor detection points should have an extremely low false positive attack detection rate so that normal usage is never flagged as malicious, but I wondered which detection points might identify attackers sooner than others - before some potential vulnerability could be targeted. What I needed was a list of threats (probably automated threats) that were not just attempting to exploit individual implementation bugs or misconfigurations. In other words, what are attackers actually doing most of the time?

And here I came across a blocker - there did not seem to be a clear categorisation or quantification of the actual automated threats most web application owners have to deal with day to day. These are also mostly not included in “breach” statistics and discussions, even though breaches of security are occurring. Instead, there is a greater focus on individual types of weaknesses and vulnerabilities, root cause analysis of data confidentiality breaches, and capabilities from vendors about product/services. Some business owners are submerged in technical details that lead to a lack of comprehension about the relationships between security requirements, security activities during development, deployment and operation, and the operational impact of attacks. It also seems to be the case there is too great a focus on individual weaknesses/vulnerabilities in technical assurance activities, especially where the severity rating of each issue in isolation fails to provide the overall picture. For example, it is common for a number of individual low or medium severity issues to contribute to a much more significant business impact.

I was also concerned about the potential misuse of valid functionality, as this is an aspect where early design decisions have a significant effect on operational risk.

In order to quantify these threats, it is necessary to be able to name them. This did not seem to exist in the usual dictionaries and classifications. Therefore, I decided to produce an ontology of automated threats from the perspective of defenders.

I was concerned about the size of the task and consequently, to contain the scope somewhat, I decided to focus solely on web applications. The first output - an ontology - is provided in this document. And now I am moving on to produce other materials for web application defenders.

Colin Watson
6th July 2015

Terminology

This handbook uses terminology based on the following sources:

1. Risk Taxonomy, Technical Standard, The Open Group, 2009
   http://pubs.opengroup.org/onlinepubs/9699919899/toc.pdf
2. NISTIR 7298 rev 2, NIST
   http://nvlpubs.nist.gov/nistpubs/ir/2013/NIST.IR.7298r2.pdf
3. OSI model, Wikipedia
   http://en.wikipedia.org/wiki/OSI_model
4. TCP/IP model, Wikipedia
   http://en.wikipedia.org/wiki/Internet_protocol_suite
5. Architecture of the World Wide Web, Volume One, W3C
   http://www.w3.org/TR/webarch/
6. Help and FAQ, W3C
   http://www.w3.org/Help/

Action

An act taken against an asset by a threat agent. Requires first that contact occurs between the asset and threat agent (Ref 1).

Application

Software that performs a business process, i.e. not system software. A software program hosted by an information system (Ref 2).

Application layer

"Layer 7" in the OSI model (Ref 3) and "application layer" in the TCP/IP model (Ref 4).

Threat

Anything that is capable of acting in a manner resulting in harm to an asset and/or organization; for example, acts of God (weather, geological events, etc.); malicious actors; errors; failures (Ref 1).

Threat Agent

Any agent (e.g., object, substance, human, etc.) that is capable of acting against an asset in a manner that can result in harm (Ref 1).

Threat Event

Occurs when a threat agent acts against an asset (Ref 1).

Web

The World Wide Web (WWW, or simply Web) is an information space in which the items of interest, referred to as resources, are identified by global identifiers called Uniform Resource Identifiers (URI) (Ref 5). The first three specifications for Web technologies defined URLs, HTTP, and HTML (Ref 6).

Web application

An application delivered over the web.

Introduction

Background

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 caused by design flaws rather than implementation bugs) are less well defined. Yet these problems are seen day-in, day-out by web application owners. Some examples commonly referred to are:

- Account enumeration
- Aggregation
- Click fraud
- Comment spam
- Content scraping
- etc.

Excessive abuse of functionality is commonly misreported as application denial-of-service (DoS) attacks, such as HTTP flooding or application resource exhaustion, when in fact the DoS is a side-effect. Most of these problems seen regularly by web application owners are not listed in any OWASP Top Ten or in any 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.

Requirements

The aim was to produce an ontology that would provide a common language for developers, architects, operators, business owners, security engineers, purchasers, and suppliers/vendors, in order to facilitate clear communication and help tackle these 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 licence.

Objectives

The objectives defined in early 2015 were:

- Provide a definition of the term "automated threat"
- Create a common vocabulary of automated threats and their relationships to each other that maintains consistency with existing literature.

This would involve creating a listing of vendor-neutral, technology-agnostic terms that describe real-world automated threats to web applications, at a level of abstraction suitable for application owners. The ontology and other supporting materials need to be practical and useful for a range of activities throughout a secure software development lifecycle (S-SDLC).

Research

Scope

The focus for the project is the abuse of functionality - misuse of inherent functionality and related design flaws, some of which are also referred to as business logic flaws. There is no coverage of implementation bugs. It is neither the case that implementation bugs are not the target of attacks, nor that their exploitation cannot be automated, but there is much more knowledge published in that area with a greater agreement on terminology. The intention was that all the threats must require the web to exist for the threat to be materialised; thus attacks that can be achieved without the web are out of scope.

The threat events are scenarios which are seen commonly by real operating web applications, and are multi-step and/or highly iterative and/or multiple weaknesses involved, and not primarily about events that relate to the tool-based exploitation of single-issue vulnerabilities of individual web applications. Essentially the ontology needs to be a list of concise answers to the operational question “what is happening right now?”.

The summary definition created to describe this is “Threat events to web applications undertaken using automated actions”.

The terms threat, threat event, web, applications and automated are defined in the glossary towards the end of this main text.

Some examples that are out of scope for this ontology are:

- Native mobile apps (but web application endpoint threats are in scope)
- Threats pre deployment (e.g. design, development, testing, deployment)
- Threats that affect web application businesses, but that are not undertaken using the web (e.g. in e-commerce: return fraud, wear & return fraud, not delivered fraud, price arbitrage, nearby address fraud, cross-merchant no-receipt returns, friendly fraud)
- Other layer 7 protocols including e.g. FTP, SMTP
- Host addressing and identification
- Attacks targeting network infrastructure
- Network, HTTP and SSL/TLS denial of service
- Physical and environmental attacks against components supporting web applications.

Therefore, attacks like phishing, pharming, and trojan distribution are excluded.

Literature review

Work began on the project in late January 2015. Over 150 sources of information were identified, read and relevant threat information extracted. The full list of academic papers, blog posts, briefings, conference presentations, dictionaries, news stories, reports, technical papers and white papers is too long to include in this handbook but is published on the OWASP wiki:

https://www.owasp.org/index.php/OWASP_Automated_Threats_to_Web_Applications#tab=Bibliography

This created over 600 data points describing a mixture of threats, attacks and some vulnerabilities. Updates were periodically posted to the project pages on the OWASP wiki.

Analysis

In order to distil the data points to a more manageable scope, the information was first converted into a large-scale diagram. This attempted to remove duplication and highlight interrelationships. The diagram can be found on the OWASP wiki:

https://www.owasp.org/index.php/File:Automated-threats.pdf

Anything relating to exploitation of implementation bugs was excluded. Forty or so clusters of threats were extracted from this diagram, and this was reduced further to a slightly smaller number of candid threat event names. Work then began to identify inter-relationships, similarities, overlaps and unique aspects. This process was undertaken over 1-2 months and reduced the number of recommended threat event names to just over twenty. Further de-duplication reduced the final count to nineteen. See below for a discussion of some of the candidate names that did not make the list.

Peer review and comparison with other dictionaries, taxonomies and lists

The project was announced in the OWASP Foundation’s Connector newsletter sent to 60,000+ recipients in April 2015. It was also highlighted in a two-side colour flyer included in every delegate’s bag at AppSec EU 2015 in Amsterdam. A limited amount of peer review has been undertaken over about a month with:

- Professional colleagues
- Web application owners
- Web application developers
- Delegates at AppSec EU 2015 via an online and printed survey form
- One-to-one interviews with participants of the OWASP Project Summit 2015 in Amsterdam
- Others who found the project by search, or from coverage relating to a presentation to be given at AppSec USA in San Francisco in September 2015.

The Ontology

The peer review led to clearer scope, suggestions for additional threats, and changes to both the names and descriptions of the threat events. Further peer review would be welcome.

Three OWASP projects were reviewed at an early stage:

- The OWASP Top 10 [Web Application] Risks is the most well known OWASP output, but is a high-level awareness document with the aim to educate developers, designers, architects, managers, and organisations about the consequences of the most important web application security weaknesses; it highlights common and higher impact risks caused by both design flaws and implementation bugs; abuse of functionality is not a current top 10 item; no names from the OWASP Top 10 are included in the ontology
- The OWASP Top Ten [Web Application] Proactive Controls is a list of security techniques that should be included in every software development project; it is focused on reducing the incidence of weaknesses and vulnerabilities, but does not particularly address automated threats
- The OWASP WASC (Web Application Security Consortium) Web Hacking Incidents Database Project (WHID) classifies publicly known incidents using attack methods, weaknesses and outcomes. As such, it excludes incidents that were not reported, and thus is lacking in data relating to misuse of functionality. Some of the application denial of service incidents may include data that relates to other threat events described in the ontology.

The OWASP wiki includes many categorisations, one of which is “attack”. The named items point to some automated threats, and were reviewed in the research stage. During the literature review and subsequent analysis and finalisation of the ontology, two reference sources were referred to again and again:

- Common Attack Pattern Enumeration and Classification (CAPEC) is a dictionary and classification taxonomy of known attacks on software. Its primary classification structures are Domains of attack (3000) and Mechanism of Attack (1000). While CAPEC includes many closely related threat events, and many detailed description of attacks, the dictionary does not provide coverage of all the automated threats identified in this ontology; the best match is often the category CAPEC-210 Abuse of Functionality; see Appendix B for a mapping of CAPEC category and attack pattern IDs to the ontology
- The Web Application Security Consortium (WASC) Threat Classification classifies weaknesses and attacks that can lead to the compromise of a website, its data, or its users; this was a useful source of automated threat information, but apart from authentication threats, most of the relevant concerns fall within a single classification (WASC-42 Abuse of Functionality).

But none of the above, nor Mitre’s Common Weakness Enumeration (CWE) which is the most comprehensive dictionary of software weaknesses, provide the coverage and owner-viewpoint that this project aims to create.

Introduction

The research and analysis and discussions with peers, completed over five months, whittled down the threat actions to a smaller core list of nineteen, as described above.

The names used, combined with their defining characteristics, are taken from existing usage whenever possible. However, terminology is not used consistently within the literature sources reviewed, and also in some cases it was necessary to use a more generic term that captures the wider idea, instead of an individual common name. Furthermore, the intended outcomes of the threat action are usually unknown at the time of the action taking place, and thus outcome-related names were generally rejected. For example, is the creation of a fake account intended for distributing malware in user-generated content, or to manipulate search engine scoring, or to influence other users, or to explore the authenticated parts of the application?

The ontology is a list of threat event scenarios (when a threat agent acts against an asset, partially ordered in time) by software. The threat events cause a divergence from accepted behavior producing one or more undesirable effects on a web application. The list excludes tool-based exploitation of single-issue vulnerabilities.

The list

Full details of the finalised ontology threat events are provided in the cream coloured pages at the end of this handbook. A summary is provided below. Table 1 lists the threeat events ordered by ascending identity code, and Table 2 lists the threat events by ascending name.

The details at the end of this handbook categorise the threat events by:

- Sectors Targeted - Sectors that are targeted more commonly than others for the specific threat event are highlighted in amber; this is currently just the author’s opinion, but the project is seeking information to define this aspect more accurately
- Parties Affected - Whether individuals, groups of people, the application owner and other parties are most often affected adversely by the threat event; the threat event may affect other parties depending upon the application and its data; the parties affected, excluding subsequent further misuse
- Data Commonly Misused - The types of data are web application specific; however, some threat events are more likely to occur for certain data types.

Each threat event is also cross-referenced with:

- Mitre CAPEC - best full and/or partial match CAPEC category IDs and/or attack pattern IDs
- WASC Threat Classification - best match to threat IDs
- Mitre Common Weakness Enumeration - closely related base, class & variant weakness IDs
- Matching pages defining terms classified as attacks on the OWASP wiki.

Figure 1: Automated Threat Events, ordered by ascending identity code

Identity Code	Name	Defining characteristics
OAT-001	Carding	Multiple payment authorisation attempts used to verify the validity of bulk stolen payment card data
OAT-002	Token Cracking	Mass enumeration of coupon numbers, voucher codes, discount tokens, etc
OAT-003	Ad Fraud	False clicks and fraudulent display of web-placed advertisements
OAT-004	Fingerprinting	Elicit information from the web, application and database servers about the supporting software and framework types and versions
OAT-005	Scalping	Obtain limited-availability and/or preferred goods/services by unfair methods
OAT-006	Expediting	Perform actions to hasten progress of usually slow, tedious or time-consuming actions on behalf of a person.
OAT-007	Credential Cracking	Identify valid login credentials by trying different values for usernames and/or passwords
OAT-008	Credential Stuffing	Mass log in attempts used to verify the validity of stolen username/password pairs
OAT-009	CAPTCHA Bypass	Solve anti-automation tests
OAT-010	Card Cracking	Identify missing expiry dates and security codes for stolen payment card data by trying different values
OAT-011	Scraping	Collect application content and/or other data for use elsewhere
OAT-012	Cashing Out	Buy goods or obtain cash utilising validated stolen payment card or other user account data
OAT-013	Sniping	Last minute bid or offer, for goods or services
OAT-014	Vulnerability Scanning	Crawl and fuzz application to identify weaknesses and possible vulnerabilities
OAT-015	Denial of Service	Target resources of the application and database servers, or individual user accounts, to achieve denial of service (DoS)
OAT-016	Skewing	Repeated link clicks, page requests or form submissions intended to alter some metric
OAT-017	Spamming	Malicious and/or more benign information addition, that appears in public or private content, databases or user messages
OAT-018	Footprinting	Probe and explore application to identify its constituents and properties
OAT-019	Account Creation	Create multiple accounts for subsequent misuse
OAT-020	Account Aggregation	Use by an intermediary application to collect together accounts and interact on their behalves.

Figure 2: Automated Threat Events, ordered by ascending name

Identity Code	Name	Defining characteristics
OAT-020	Account Aggregation	Use by an intermediary application to collect together accounts and interact on their behalves.
OAT-019	Account Creation	Create multiple accounts for subsequent misuse
OAT-003	Ad Fraud	False clicks and fraudulent display of web-placed advertisements
OAT-009	CAPTCHA Bypass	Solve anti-automation tests
OAT-001	Carding	Multiple payment authorisation attempts used to verify the validity of bulk stolen payment card data
OAT-010	Card Cracking	Identify missing expiry dates and security codes for stolen payment card data by trying different values
OAT-012	Cashing Out	Buy goods or obtain cash utilising validated stolen payment card or other user account data
OAT-007	Credential Cracking	Identify valid login credentials by trying different values for usernames and/or passwords
OAT-008	Credential Stuffing	Mass log in attempts used to verify the validity of stolen username/password pairs
OAT-015	Denial of Service	Target resources of the application and database servers, or individual user accounts, to achieve denial of service (DoS)
OAT-006	Expediting	Perform actions to hasten progress of usually slow, tedious or time-consuming actions on behalf of a person.
OAT-004	Fingerprinting	Elicit information from the web, application and database servers about the supporting software and framework types and versions
OAT-018	Footprinting	Probe and explore application to identify its constituents and properties
OAT-005	Scalping	Obtain limited-availability and/or preferred goods/services by unfair methods
OAT-011	Scraping	Collect application content and/or other data for use elsewhere
OAT-016	Skewing	Repeated link clicks, page requests or form submissions intended to alter some metric
OAT-013	Sniping	Last minute bid or offer, for goods or services
OAT-017	Spamming	Malicious and/or more benign information addition, that appears in public or private content, databases or user messages
OAT-002	Token Cracking	Mass enumeration of coupon numbers, voucher codes, discount tokens, etc
OAT-014	Vulnerability Scanning	Crawl and fuzz application to identify weaknesses and possible vulnerabilities

Mappings to other lists

The cross-references with the WASC Threat Classification and Mitre CAPEC, defined in the reference section at the back of this handbook, were examined further to determine how those differ from this ontology.

Figure 3: WASC Threat Classification view of the Automated Threat Events

The majority of the threat events are both the weakness WASC-21 Insufficient Anti-automation and the attack WASC-42 Abuse of Functionality. Three also relate to the attack WASC-11 Brute Force. WASC-45 Fingerprinting includes both **OAT-004 Fingeprinting** and **OAT-018 Footprinting.** Both WASC and this ontology have a unqiue category for Denial of Service.

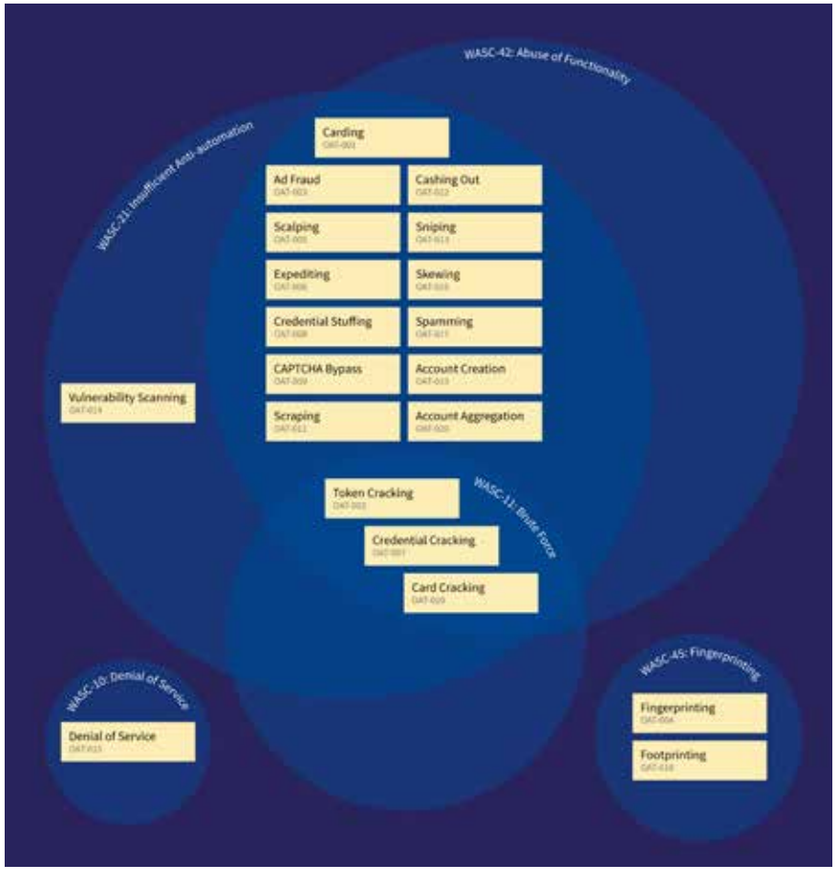
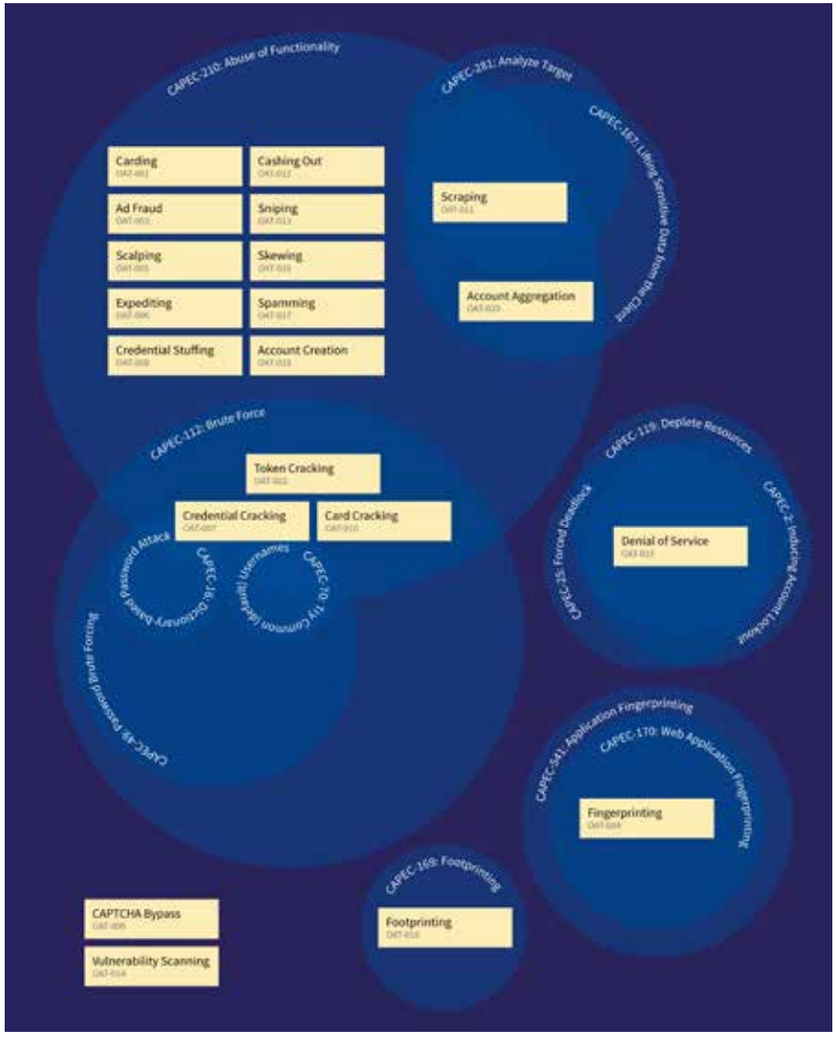


Figure 4: Mitre CAPEC view of the Automated Threat Events

Again, there are many threat events in the CAPEC-210 Abise of Functionality. CAPEC also has additional categosiations for brute force attacks and denial of service. Two threat events, **OAT-009 CAPTCHA Bypas**s and **OAT-014 Vulnerability Scanning**, do not appear to exist within CAPEC.



Notes

Threat event names

In all cases, “automated web application” could be used as a prefix to each name. Thus, for example, **OAT-012 Cashing Out** is concerned only with using web applications to obtain cash or goods; the ontology’s scope excludes cashing out using ATMs. **OAT-015 Denial of Service** is web application denial of service, and not an SSL/TLS or network DoS. When referencing the terms in other contexts, it may be useful to ensure that the web application scope is identified.

Whenever possible, an existing term already used in literature or industry usage was preferred, but in many cases, it was difficult to identify such a term; as such, in some cases a more generic version had to be used. A good example of this is **OAT-013 Sniping**, where auction sniping is the most commonly cited case; it was determined that the characteristics of sniping also occur in threat events against other types of applications, and the selected name was thus made more general.

A handful of threat event names in the ontology are very specific since they are reported to occur frequently (e.g. **OAT-001 Carding**, **OAT-019 Account Creation**). Others are larger buckets (e.g. **OAT-011 Scraping**, **OAT-017 Spamming**, **OAT-014 Vulnerability Scanning**) that cannot be broken down easily without sharding the threat events into a multitude of sector-specific and function-specific examples.

For a while during the development of the ontology, aggregation of user accounts was temporarily included within **OAT-011 Scraping**. However, during final review it was felt the aspects of customer opt in, the intermediarisation and resulting disengagement were sufficiently different from scraping to make it a separate term. Furthermore, it was a threat commonly seen in financial services. The threat event was added back in as **OAT-020 Account Aggregation**. Other threat events described below were removed or consumed in other terms.

The only name newly created is **OAT-006 Expediting**, as there appeared to be a large number of sector-specific threats involving increased multi-step velocity that could otherwise not be aggregated together under a single name.

Threat event identity codes

To enable internal cross-referencing and referencing from elsewhere, each threat event has been given an identification (ID) code. This is a three-digit number prefixed by a hyphen and an abbreviation for OWASP Automated Threat (OAT) e.g. OAT-015. The ID codes were randomly assigned in an attempt to stop the ontology being seen as an ordered list, and also to ensure that neighbouring items are not necessarily related. Other cross-referencing is provided. Currently codes 001 to 020 are used, and it is expected the total number should be many fewer than fifty, unless many sub-items are ever added. Three digits, rather than two, were allotted in case the first digit is used for some other aspect in future, e.g. perhaps mobile application automated threat events could be 1xx, and 2xx for embedded software, etc.

Timing, duration and frequency

The scope focuses on threat events that involve multi-step and/or highly iterative interactions with the application. But by their nature, the identified threat events vary significantly in scale, and their timing, duration and frequency can all vary considerably. This is an area that could be explored further in future work.

Magnitude of impact

Events related to automated threats can have impacts on more than just the application owner. Individuals, third parties and even society can be adversely affected. This ontology does not attempt to provide information on, or rank the threat events in terms of impact, since it will be organisation, data, threat actor and victim-perspective specific. An organisation may choose to use its own risk assessment processes to rank these threats for each operational entity, or each market, or even by individual application.

The perpetrators

During the early stages of the ontology’s creation, it was believed it would be possible to suggest which threat actors might be most likely to initiate the threat event. These threat agents might be groups like competitors, journalists, petty criminals, organised crime, nation states, etc and of course users such as citizens, clients, customers and employees. However, on further inspection, the threat agents appear to be more closely related to the type of data, and thus sector, rather than the particular threat event. Consequently, it is believed threat agents should be re-considered in future sector-specific views of the ontology.

Furthermore, some threat events may be undertaken by, or with the knowledge or implicit support of, application owners. For example, search engine indexing is generally encouraged due to the benefit of increased user traffic (**OAT-011 Scraping**); automated monitoring of web applications may be commissioned (**OAT-011 Scraping**); excessive account creation might contribute to enhanced market reputation when promoting the size of its customer base (**OAT-019 Account Creation**); the application owner with hosted advertisements could receive additional income for false impressions (**OAT-003 Ad Fraud**).

Fraud, legality and cheating

In general, the ontology tries to avoid the use of judgmental words like fraud. But in one case, the industry accepted term for the threat event includes this word: **OAT-003 Ad Fraud**.

In legal terms, whether an action is fraudulent depends on legislation and jurisdiction. Some of the events in this ontology may be illegal actions in some jurisdictions, or may be prohibited in a commercial contract. This will also depend upon the types of data handled, regulation of the application and its owner, and application-specific mandates like terms of use.

Rather than being illegal, some threat events will be considered cheating by other normal users, including **OAT-006 Expediting**, **OAT-005 Scalping**, **OAT-016 Skewing** and **OAT-013 Sniping**. These will be sector, application and culturally specific views, but can undermine user trust and the reputation of the application and its owner.

Terms excluded

A small number of threat events were removed during analysis and review based on discussions with peers and website owners. The primary reason for removal was either being out of scope, or because the term could not be adequately distinguished from another. Other people may have alternative views on these, so the discarded temporary working names and justifications are provided below alphabetically.

Application Consumption was a temporary working name given the misuse of the application to perform calculations, or process data, or perform other actions against other applications, hosts, or in the physical world, i.e. unauthorised real-time consumption of a normal application as if it were an API. Unlike data harvesting, in which information is gathered once or periodically, in consumption, the thought was the application is used on-demand by another system to provide calculated output, send requests to another application, or possibly affect physical assets the application provides direct control over. For example the application might be used to generate images or other files based on user input. In the second case, the application checks user submitted data (e.g. hostname, email address) by undertaking a reverse lookup, pingback or a reputation service check, contributing to a denial of service attack against that other host. In these situations, there seemed to be a close similarity with data harvesting and thus it was eventually concluded to be another example of **OAT-011 Scraping**.

Application Worms, also called cross-site scripting worms, are a combination of two different implementation flaws – cross-site request forgery (CSRF) and cross-site scripting (XSS). Additionally, the automation is undertaken by the web application itself in conjunction with often normal usage by innocent users. Therefore, it was decided this did not fall within the defined scope.

Asset Stripping was considered to encompass the removal of application stored non-data assets using compromised accounts and sessions, including data theft, collecting micro deposits, and collecting refunds. However, this asset removal, extraction or copying from applications used as repositories is no different from other data harvesting at the time of extraction. The only difference is the assets have value in other non-application contexts and may include fiat money, credit, refunds, financial instruments, reputation, virtual assets (e.g. status, score, virtual currency, identity), awards and points, and possible physical assets the application provides control over. But this value is often very subjective. Since these are data, it was considered this threat event was actually part of **OAT-011 Scraping**. The objectives of the attacker and consequences are data and application specific. Additionally, the transfer of money was included within **OAT-012 Cashing Out**. Consequently, Asset Stripping was not included as a separate term.

Attack Platform was at first used to describe the misuse of an application to mount automated attacks against another application or other external information system component. This would include reflected DoS, anti-spam check DoS, amplification DoS, and numerous HTML5 attacks. For example, if the application checks user submitted data (e.g. hostname, email address) by undertaking a reverse lookup, pingback or a reputation service check, contributing to a denial of service attack against that host. Or if an HTML application is compromised to undertake attacks against local and other remote systems. The affected host is not the application itself; instead, the application performs the attack on some other system. Ultimately, like the somewhat related Code Modification below, this was dropped from the ontology

Code Modification relates to when the application logic is changed by modification of the source code, or the executing code, or the configuration, or some combination of these. The kinds of attacks included are malicious software download, malicious software update, advert injection, code tampering, DOM modification, web browser tools, form tampering, malicious software implanting, backdoor addition, shared data manipulation, use of untrusted code, memory modification, AngularJS attack, configuration data modification, exposed reflection, reflection injection, autobinding, and Rich Internet Application (RIA) attacks. The issue is made more significant with the growing use of client-side code. But it was felt these threats were related to lack of integrity checks, particularly during development and distribution, rather than being typical automated threats, and therefore Code Modification is not included in this ontology.

Form Hijacking (e.g. email spam, form to Email spam, SMS spam, use as a spam relay, and unsolicited bulk email) was initially thought to be a core threat event and would have been an ideal candidate for the threat event ontology. But again, it was realised that this is an implementation flaw that leverages vulnerabilities produced when an web server fails to validate input, and thus it does not fit into this ontology.

Man in the Browser (MitB), in which the attacker controls the user's web browser, so that information being transferred can be observed, intercepted and manipulated, was another threat event that was thought at the start of the project would be in the final ontology. The most well-known use case is to undertake financial fraud, and is the result of compromise of the user's device by a banking trojan, such as URLzone, Torpig, and Zeus. However, MitB can also be used for

advert injection, and some simpler variants have been labelled Boy in the Browser (BitB). MitB/BitB are believed to be out-of-scope, since the trojan distribution, and the interception/change of information, are both occurring outside the web application's boundaries.

Reverse Engineering is exercising an application or part of an application with the intent to gain insight into how it is constructed and operates. The purpose may be to understand the inner workings, and may be used to determine business logic such as pricing models, reproduce the application elsewhere, or to assist with vulnerability exploitation and data compromise. It was decided to be an intended outcome of a combination of other threat actions - typically, **OAT-011 Scraping** and **OAT-018 Footprinting**, which include the testing and collection of evidence to determine the underlying logic, structures, algorithms, functions, methods, and secrets of the application. Thus, as an outcome it was decided that Reverse Engineering is not a valid part of the ontology.

Use Case Scenarios

Introduction

The following scenarios and organisation names are completely fictitious.

Scenario: Defining application development security requirements

Cinnaminta SpA intends to build and launch a new multi-lingual and multi-currency e-commerce website. The development will be outsourced and Cinnaminta has been working on the functional design document. Among many other requirements, the application security specification requires that the website must not include any vulnerabilities identified in PCI DSS v3.1 Requirement 6.5, nor any other vulnerabilities that could affect the protection of payment cardholder data. Cinnaminta specifies that the website's payment functions must not be susceptible to the threat events **OAT-001 Carding** or **OAT-010 Card Cracking**, as defined in the ***OWASP Automated Threat Handbook***. In addition, the application must interact with the company's existing fraud detection system to counter **OAT-012 Cashing Out**. The requirements are specified in terms of these threat events, rather than particular product or service categories. Development houses responding to the call for bids use the ontology to focus their answers to these aspects appropriately.

Scenario: Sharing intelligence within a sector

Unlimited Innovations Inc develops and supports patient-facing software solutions to a range of healthcare providers, many of which participate in the National Health Service Cyber Intelligence Sharing Center (NHS-CISC). Unlimited Innovations already builds continuous monitoring capabilities into its software and decides to provide an optional enhancement so that customers could choose to share their misuse event data with each other, to benefit from the combined threat intelligence. Rather than sharing large quantities of low-level data, Unlimited Innovations aggregates information and broadcasts validated and categorised threat data amongst the participating organisations. Automation attacks are classified according to the threat events defined in the ***OWASP Automated Threat Handbook*** so that each receiving party understands the nature of the threat. Even organisations that do not want to take part in this information sharing can benefit, since their own categorised information is made available to internal business management in the form of an easy-to-comprehend monitoring dashboard. The information gathered can also be fed into their other business information management systems to help improve patient service.

Scenario: Exchanging threat data between CERTs

National Computer Emergency Response Teams (CERTs) recognise that sharing of local information can contribute to worldwide prevention of cyber attacks. Despite advances in cooperation between CERTs, anything to increase continuity and interoperability, such as standards for data exchange, is encouraged. CERT Zog is concerned about the sparsity of application-specific data it receives, and also the classification of that data. It has a particular concern about attacks and breaches that affect sectors defined in Zog's 2015 national cyber

security strategy. CERT Zog and its neighbour CERT Tarset agree to tag threat events using the ***OWASP Automated Threat Handbook***, in order to add greater context to existing solutions being used for threat data exchange between them. The programme also collects sector metadata, so that all organisations within these can benefit from the centralised intelligence.

Scenario: Enhancing application penetration test findings

Specialist application security penetration testing firm Cherak Industries Pte Ltd works primarily for financial services companies in the banking and insurance sectors, and is looking to expand its business throughout Asia. Cherak has some innovative pen test result reporting systems which integrate with client software fault and vulnerability tracking systems, and it actively looks for methods to provide additional value to its clients. Cherak has identified that pen test clients would benefit from help in understanding the effects of combinations of vulnerabilities, especially design flaws, and has decided to utilise the ***OWASP Automated Threat Handbook*** to define and explain the automation-related threats. The individual vulnerabilities were scored as normal using CVSSv2 and v3, the matching CWEs identified, and mitigations in place documented. In addition, Cherak uses the threat events defined in the ***OWASP Automated Threat Handbook*** to help create a new section in the executive summary that explains how combinations of the issues found could lead to automation threats and the possible technical and business impacts. For example, an assessment for one client had identified weaknesses in authentication so that there is a risk of **OAT-008 Credential Stuffing**. The defined identifier was provided to the client, so its technical staff could refer to additional information on the OWASP website.

Scenario: Specifying service acquisition needs

Falstone Paradise Inc is concerned about malicious use of their portfolio of hotel and resort websites. The majority of the websites use a shared application platform, but there are some unique applications and a large number of other micro-sites, some of which use generic content management systems such as Wordpress and Drupal. Falstone Paradise has identified that its IT operations team are spending too much time dealing with the effects of automated misuse, such as cleaning up data, resetting customer accounts and providing extra capacity during attacks. Furthermore, the unwanted automation is also causing some instabilities leading to negative feedback from customers. Therefore Falstone Paradise decides to go out to the security marketplace to identify, assess and select products or services that might help address these automation issues for all its websites. Their buying team works with their information technology colleagues to write the detailed requirements in an Invitation to Tender (ITT) document. This describes the types of attacks its web applications are receiving, their frequency of occurrence and their magnitudes. These are defined according to the ***OWASP Automated Threat Handbook***, so that vendors do not misunderstand the requirements, and each vendor's offering can be assessed against the particular automation threat events of concern.

Scenario: Characterising vendor services

Better Best Ltd has developed an innovative technology to help gaming companies defend against a range of automated threats that can otherwise permit cheating and distortion of the game, leading to disruption for normal players. The solution can be deployed on premises, but is also available in the cloud as a service. But Better Best is finding difficulty explaining its solution in the market place, especially since it does not fit into any conventional product category. Better Best decide to use the terminology and threat events listed in the ***OWASP Automated Threat Handbook*** to define their product's capabilities. They hope this will provide some clarity about their offering, and also demonstrate how their product can be used to replace more than one other conventional security device. Additionally, Better Best writes a white paper describing how their product has been successfully used by one of their reference customers Hollybush Challenge Games to protect against **OAT-006 Expediting**, **OAT-005 Scalping**, **OAT-016 Skewing** and **OAT-013 Sniping**.

Project Details

OWASP project

The wiki page for OWASP Automated Threats to Web Applications Project is:

https://www.owasp.org/index.php/OWASP_Automated_Threats_to_Web_Applications

The project's mailing list is:

https://lists.owasp.org/mailman/listinfo/automated_threats_to_web_applications

Source materials and outputs

The original versions of this handbook are maintained at these locations:

- Print-ready high resolution PDF
  TBD
- Low resolution PDF
  TBD
- Adobe InDesign
  TBD

Other working materials and outputs are:

- Print on demand book
  TBC
- Project flyer, 2-page PDF
  https://www.owasp.org/index.php/File:Automation-project-briefing.pdf
- Survey sheet used at Appsec EU 2015, PDF
  https://www.owasp.org/index.php/File:Automation-questionnaire-1v0.pdf
- Summary threats and attacks extracted during the research phase, large-scale PDF
  https://www.owasp.org/index.php/File:Automated-threats.pdf

Handbook Roadmap

Ongoing improvement

It is hoped that the production of the ontology and handbook will lead to further discussion and debate and encourage additional project participants. A key area where help is required is in gathering data on the prevalence of these threats, where some form of data collection initiative is required.

People can contribute by posting ideas, suggestions, and other inputs to the project's public mailing list (see opposite):

Enhancements

It is also intended to develop sector-specific guides that include:

- Highest risk threat events
- Attacker motivations.

Retail and financial service sectors appear to be good candidates to begin with.

Currently, each threat event is defined on a single page (see the listing at the end of this handbook). It is intended to augment the current information with the following details on the reverse sides of the threat event pages:

- Symptoms
- Mitigations
- For builders
- For defenders
- Threat identification metrics.

It would also be useful to summarise the developer-relevant information into a new Automated Threat Cheat Sheet, and contribute that to the OWASP Cheat Sheet Series.

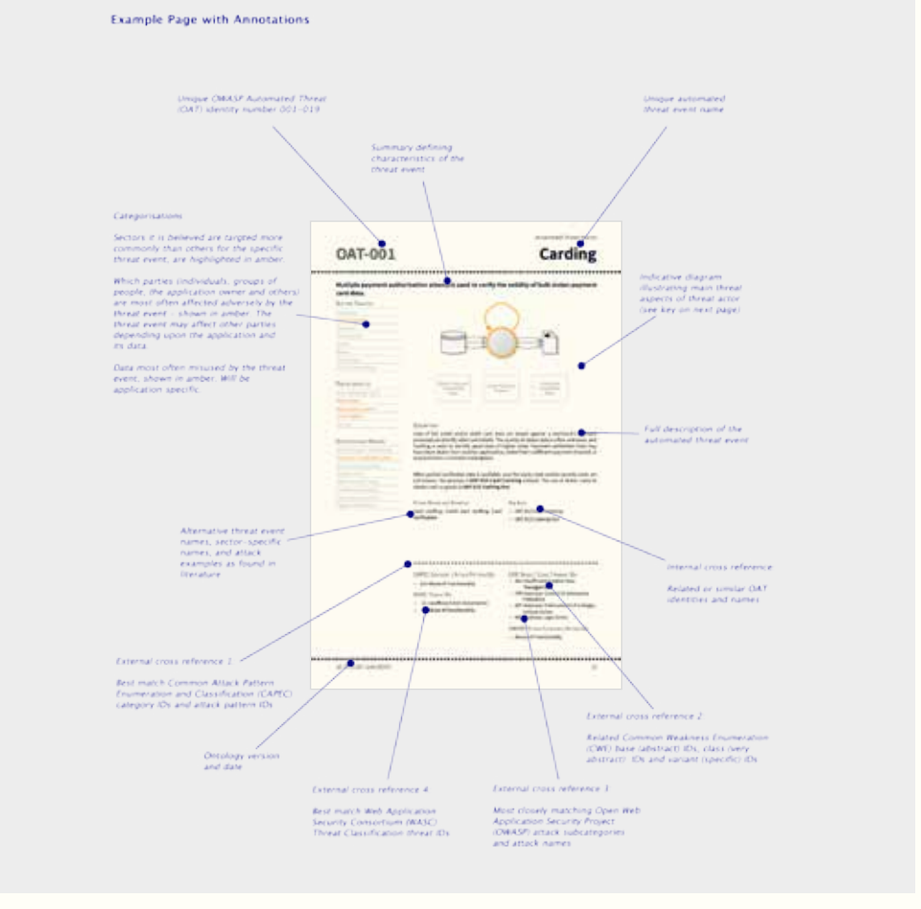
The author also hopes this ontology with its industry cross-referencing may be of help in contributing to Mitre's Common Weakness Risk Analysis Framework (CWRAF) and Common Attack Pattern Enumeration and Classification (CAPEC). Also, in the future, the terms might be useful for helping to describe some application events in the Mitre/DHS Structured Threat Information eXpression (STIX).

The Ontology

The following pages define each automated threat event in detail. The threat events are ordered by ascending identity code.

Key

Each threat event defined in the ontology is laid out on identically laid out pages. The annonated example below gives additional information about the various components. Further information is provided in the previous pages of this document.

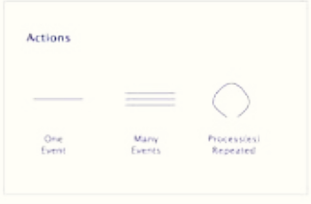


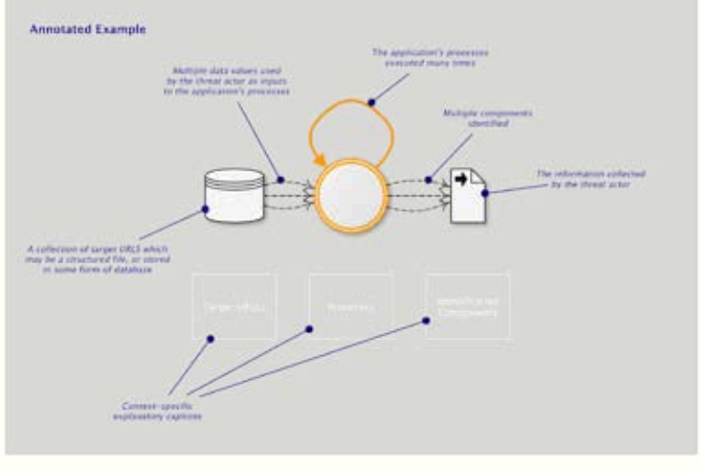
External cross-reference information sources:

1. Common Attack Pattern Enumeratyiona nd Classification (CAPEC), v2.6, The Mitre Corporation, July 2014
   https://capec.mitre.org
2. Common Weakness Enumeration (CWE), v2.8, The Mitre Corporation, July 2014
   http://cwe.mitre.org
3. Category: Attack, Open Web Application Security Project (OWASP)
   https://www.owasp.org/index.php/Category:Attack
4. The WASC Threat Classification, v2.0, Web Application Security Consortium, January 2010
   http://projects.webappsec.org/w/page/13246978/Threat%20Classification

Key

Each threat event includes an indicative diagram. The key below explains the meaning of the symbols used and an annotated example.





OAT-001 Carding

Multiple payment authorisation attempts used to verify the validity of bulk stolen payment card data.

Sectors Targeted

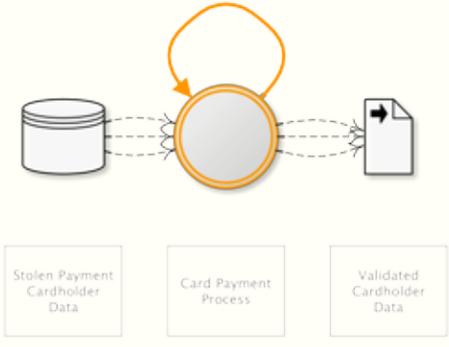
- Education
- Entertainment
- Financial
- Government
- Health
- Retail
- Technology
- Social Networking

Parties Affected

- Few Individual Users
- Many Users
- Application Owner
- Third Parties
- Society

Data Commonly Misused

- Authentication Credentials
- Payment Cardholder Data
- Other Financial Data
- Medical Data
- Other Personal Data
- Intellectual Property
- Other Business Data
- Public Information



Description

Lists of full credit and/or debit card data are tested against a merchant's payment processes to identify valid card details. The quality of stolen data is often unknown, and Carding is used to identify good data of higher value. Payment cardholder data may have been stolen from another application, stolen from a different payment channel, or acquired from a criminal marketplace.

When partial cardholder data is available, and the expiry date and/or security code are not known, the process is **OAT-010 Card Cracking** instead. The use of stolen cards to obtain cash or goods is **OAT-012 Cashing Out**.

Other Names and Examples

Card stuffing; Credit card stuffing; Card verification

See Also

- OAT-010 Card Cracking
- OAT-012 Cashing Out

CAPEC Category / Attack Pattern IDs

- 210 Abuse of Functionality

WASC Threat IDs

- 21 Insufficient Anti-Automation
- 42 Abuse of Functionality

CWE Base / Class / Variant IDs

- 799 Improper Control of Interaction Frequency
- 837 Improper Enforcement of a Single, Unique Action

OWASP Attack Category / Attack IDs

- Abuse of Functionality

Carding

OAT-001

Multiple payment authorisation attempts used to verify the validity of bulk stolen payment card data.

Symptoms

- ???

OAT-002

Token Cracking

Mass enumeration of coupon numbers, voucher codes, discount tokens, etc.

Sectors Targeted

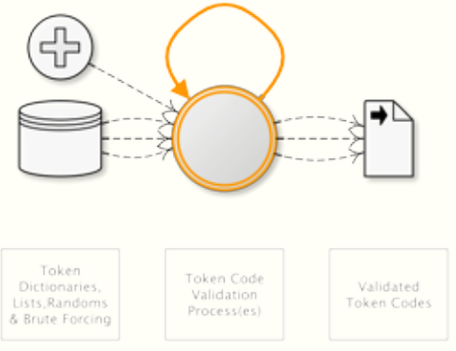
- Education
- Entertainment
- Financial
- Government
- Health
- Retail
- Technology
- Social Networking

Parties Affected

- Few Individual Users
- Many Users
- Application Owner
- Third Parties
- Society

Data Commonly Misused

- Authentication Credentials
- Payment Cardholder Data
- Other Financial Data
- Medical Data
- Other Personal Data
- Intellectual Property
- Other Business Data
- Public Information



Description

Identification of valid token codes providing some form of user benefit within the application. The benefit may be a cash alternative, a non-cash credit, a discount, or an opportunity such as access to a limited offer.

For cracking of usernames, see **OAT-007 Credential Cracking** instead.

Other Names and Examples

Coupon guessing; Voucher, gift card and discount enumeration

See Also

- OAT-007 Credential Cracking
- OAT-011 Scraping
- OAT-012 Cashing Out

CAPEC Category / Attack Pattern IDs

- 112 Brure Force
- 210 Abuse of Functionality

WASC Threat IDs

- 11 Brute Force
- 21 Insufficient Anti-Automation
- 42 Abuse of Functionality

CWEs

- 799 Improper Control of Interaction Frequency
- 837 Improper Enforcement of a Single, Unique Action

OWASP Attacks

- Abuse of Functionality
- Brute Force Attack

Token Cracking

OAT-002

Mass enumeration of coupon numbers, voucher codes, discount tokens, etc.

Symptoms

- ???

Ad Fraud

OAT-003

False clicks and fraudulent display of web-placed advertisements.

Sectors Targeted

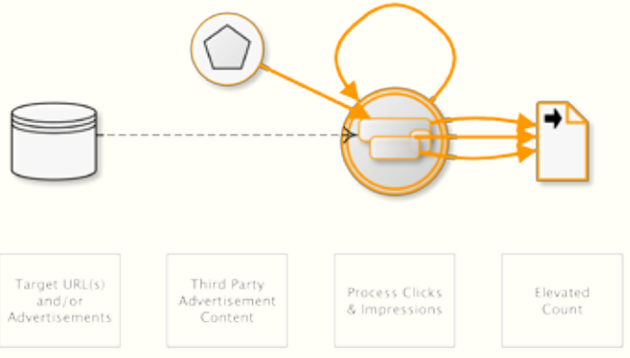
- Education
- Entertainment
- Financial
- Government
- Health
- Retail
- Technology
- Social Networking

Parties Affected

- Few Individual Users
- Many Users
- Application Owner
- Third Parties
- Society

Data Commonly Misused

- Authentication Credentials
- Payment Cardholder Data
- Other Financial Data
- Medical Data
- Other Personal Data
- Intellectual Property
- Other Business Data
- Public Information



Description

Falsification of the number of times an item such as an advert is clicked on, or the number of times an advertisement is displayed. Performed by owners of web sites displaying ads, competitors and vandals.

See **OAT-016 Skewing** instead for similar activity that does not involve web-placed advertisements.

Other Names and Examples

Advert fraud; Adware traffic; Click bot; Click fraud; Hit fraud; Impression fraud; Pay per click advertising abuse; Phoney ad traffic

See Also

- OAT-016 Skewing

CAPEC Category / Attack Pattern IDs

- 210 Abuse of Functionality

WASC Threat IDs

- 21 Insufficient Anti-Automation
- 42 Abuse of Functionality

CWEs

- -

OWASP Attacks

- Abuse of Functionality

Ad Fraud

OAT-003

False clicks and fraudulent display of web-placed advertisements.

Symptoms

- ???

Fingerprinting

OAT-004

Elicit information from the web, application and database servers about the supporting software and framework types and versions.

Sectors Targeted

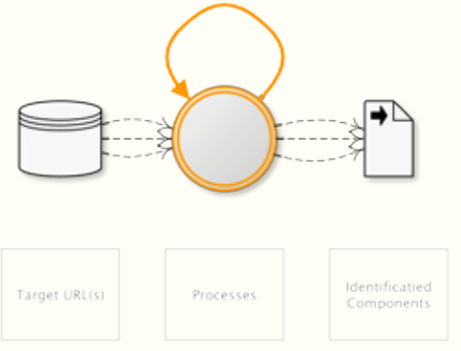
- Education
- Entertainment
- Financial
- Government
- Health
- Retail
- Technology
- Social Networking

Parties Affected

- Few Individual Users
- Many Users
- Application Owner
- Third Parties
- Society

Data Commonly Misused

- Authentication Credentials
- Payment Cardholder Data
- Other Financial Data
- Medical Data
- Other Personal Data
- Intellectual Property
- Other Business Data
- Public Information



Description

Specific requests are sent to the application eliciting information in order to profile the application. This probing typically examine HTTP header names and values, session identifier names and formats, contents of error page messages, URL path case sensitivity, URL path patterns, file extensions, and whether software-specific files and directories exist. Fingerprinting is often reliant on information leakage and this profiling may also reveal some network architecture/topology. The fingerprinting may be undertaken without any direct usage of the application e.g. by quering a store of exposed application properties such as held in a search engine's index.

Fingerprinting seeks to identity application components, whereas **OAT-018 Footprinting** is a more detailed analysis of how the application works.

Other Names and Examples

Google dorking; Google hacking; Shodaning; Target acquisition; Target scanning; Finding potentially vulnerable applications; Reconnaissance; URL harvesting; Web application fingerprinting

See Also

- OAT-011 Scraping
- OAT-018 Footprinting

CAPEC Category / Attack Pattern IDs

- 541 Application Fingerprinting
- 170 Web Application Fingerprinting

WASC Threat IDs

- 45 Fingerprinting

CWEs

- 200 Information Exposure

OWASP Attacks

- -

Fingerprinting

OAT-004

Elicit information from the web, application and database servers about the supporting software and framework types and versions.

Symptoms

- ???

OAT-005

Scalping

Obtain limited-availability and/or preferred goods/services by unfair methods.

Sectors Targeted

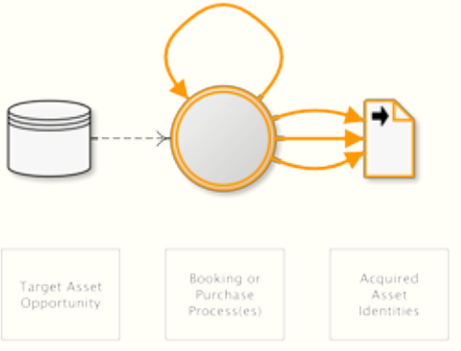
- Education
- Entertainment
- Financial
- Government
- Health
- Retail
- Technology
- Social Networking

Parties Affected

- Few Individual Users
- Many Users
- Application Owner
- Third Parties
- Society

Data Commonly Misused

- Authentication Credentials
- Payment Cardholder Data
- Other Financial Data
- Medical Data
- Other Personal Data
- Intellectual Property
- Other Business Data
- Public Information



Description

Mass acquisition of goods or services using the application in a manner that a normal user would be unable to undertake manually.

Although Scalping may include monitoring awaiting availability of the goods or services, and then rapid action to beat normal users to obtain these, Scalping is not a “last minute” action like **OAT-013 Sniping**, nor just related to automation on behalf of the user such as in **OAT-006 Expediting**. This is because Scalping includes the additional concept of limited availability of sought-after goods or services, and is most well known in the ticketing business where the tickets acquired are then resold later at a profit by the scalpers/touts. This can also lead to a type of user denial of service since the goods or services become unavailable rapidly.

Other Names and Examples

Bulk purchase; Purchase automaton; Purchase bot; Restaurant table/hotel room reservation speed-booking; Queue jumping; Sale stampede; Ticket resale; Ticket scalping; Ticket touting

See Also

- OAT-006 Expediting
- OAT-013 Sniping
- OAT-015 Denial of Service

CAPEC Category / Attack Pattern IDs

- 210 Abuse of Functionality

WASC Threat IDs

- 21 Insufficient Anti-Automation
- 42 Abuse of Functionality

CWEs

- 799 Improper Control of Interaction Frequency
- 837 Improper Enforcement of a Single, Unique Action

OWASP Attacks

- Abuse of Functionality

Scalping

OAT-005

Obtain limited-availability and/or preferred goods/services by unfair methods.

Symptoms

- ???

OAT-006

Expediting

Perform actions to hasten progress of usually slow, tedious or time-consuming actions on behalf of a person.

Sectors Targeted

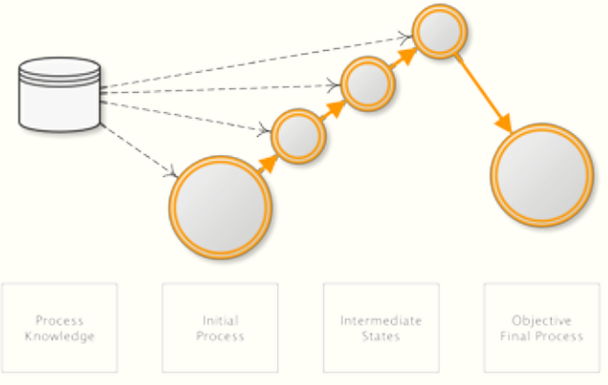
- Education
- Entertainment
- Financial
- Government
- Health
- Retail
- Technology
- Social Networking

Parties Affected

- Few Individual Users
- Many Users
- Application Owner
- Third Parties
- Society

Data Commonly Misused

- Authentication Credentials
- Payment Cardholder Data
- Other Financial Data
- Medical Data
- Other Personal Data
- Intellectual Property
- Other Business Data
- Public Information



Description

Using speed to violate explicit or implicit assumptions about the application's normal use to achieve unfair individual gain, often associated with deceit and loss to some other party.

In contrast to **OAT-016 Skewing** which affects metrics, Expediting is purely related to faster progression through a series of application processes. And **OAT-017 Spamming** is different to Expediting, since the focus of spam is to add information, and may not involve the concept of process progression.

Other Names and Examples

Algorithmic trading; Automated stock trading; Betting automation; Game automation; Gaming bot; Gold farming; Financial instrument dealing; High frequency trading; Purchase automation; Ticketing automation; Trading automation; Virtual wealth generation

See Also

- OAT-005 Scalping
- OAT-013 Sniping
- OAT-016 Skewing
- OAT-017 Spamming

CAPEC Category / Attack Pattern IDs

- 210 Abuse of Functionality

WASC Threat IDs

- 21 Insufficient Anti-Automation
- 42 Abuse of Functionality

CWEs

- 841 Improper Enforcement of Behavioral Workflow

OWASP Attacks

- Abuse of Functionality

Expediting

OAT-006

Perform actions to hasten progress of usually slow, tedious or time-consuming actions on behalf of a person.

Symptoms

- ???

OAT-007

Credential Cracking

Identify valid login credentials by trying different values for usernames and/or passwords.

Sectors Targeted

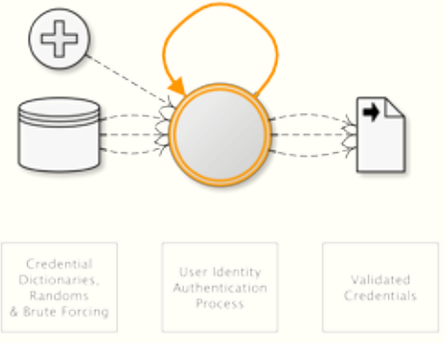
- Education
- Entertainment
- Financial
- Government
- Health
- Retail
- Technology
- Social Networking

Parties Affected

- Few Individual Users
- Many Users
- Application Owner
- Third Parties
- Society

Data Commonly Misused

- Authentication Credentials
- Payment Cardholder Data
- Other Financial Data
- Medical Data
- Other Personal Data
- Intellectual Property
- Other Business Data
- Public Information



Description

Brute force, dictionary (word list) and guessing attacks used against authentication processes of the application to identify valid account credentials. This may utilise common usernames or passwords, or involve initial username evaluation.

The use of stolen credential sets (paired username and passwords) is **OAT-008 Credential Stuffing**.

Other Names and Examples

Brute-force attacks against sign-in; Brute forcing log-in credentials; Brute-force password cracking; Cracking login credentials; Password brute-forcing; Password cracking; Reverse brute force attack; Username cracking; Username enumeration

See Also

- OAT-002 Token Cracking
- OAT-008 Credential Stuffing
- OAT-019 Account Creation

CAPEC Category / Attack Pattern IDs

- 16 Dictionary-based Password Attack
- 49 Password Brute Forcing
- 70 Try Common(default) Usernames and Passwords
- 112 Brute Force

WASC Threat IDs

- 11 Brute Force
- 21 Insufficient Anti-Automation
- 42 Abuse of Functionality

CWEs

- 307 Improper Restriction of Excessive Authentication Attempts
- 799 Improper Control of Interaction Frequency
- 837 Improper Enforcement of a Single, Unique Action

OWASP Attacks

- Abuse of Functionality
- Brute Force Attack

Credential Cracking OAT-007

Identify valid login credentials by trying different values for usernames and/or passwords.

Symptoms

- ???

OAT-008 Credential Stuffing

Mass log in attempts used to verify the validity of stolen username/password pairs.

Sectors Targeted

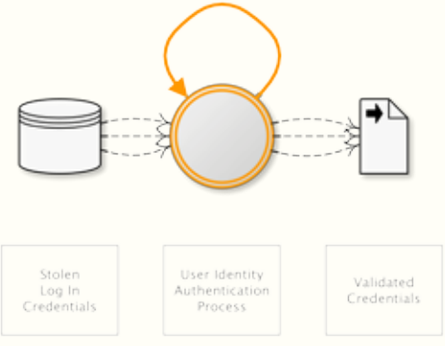
- Education
- Entertainment
- Financial
- Government
- Health
- Retail
- Technology
- Social Networking

Parties Affected

- Few Individual Users
- Many Users
- Application Owner
- Third Parties
- Society

Data Commonly Misused

- Authentication Credentials
- Payment Cardholder Data
- Other Financial Data
- Medical Data
- Other Personal Data
- Intellectual Property
- Other Business Data
- Public Information



Description

Lists of authentication credentials stolen from elsewhere are tested against the application's authentication mechanisms to identify whether users have re-used the same log in credentials. The stolen usernames (often email addresses) and password pairs could have been sourced directly from another application by the attacker, purchased in a criminal marketplace, or obtained from publicly available breach data dumps.

Unlike **OAT-007 Credential Cracking**, Credential Stuffing does not involve any brute-forcing or guessing of values; instead credentials sets used in other applications are being tested for validity.

Other Names and Examples

Account checker attack; Account checking; Account takeover; Account takeover attack; Login Stuffing; Password list attack; Password re-use; Stolen credentials; Use of stolen credentials

See Also

- OAT-007 Credential Cracking
- OAT-019 Account Creation

CAPEC Category / Attack Pattern IDs

- 210 Abuse of Functionality

WASC Threat IDs

- 21 Insufficient Anti-Automation
- 42 Abuse of Functionality

CWEs

- 799 Improper Control of Interaction Frequency
- 837 Improper Enforcement of a Single, Unique Action

OWASP Attacks

- Abuse of Functionality
- Credential Stuffing

Credential Stuffing

OAT-008

Mass log in attempts used to verify the validity of stolen username/password pairs.

Symptoms

- ???

OAT-009

CAPTCHA Bypass

Solve anti-automation tests.

Sectors Targeted

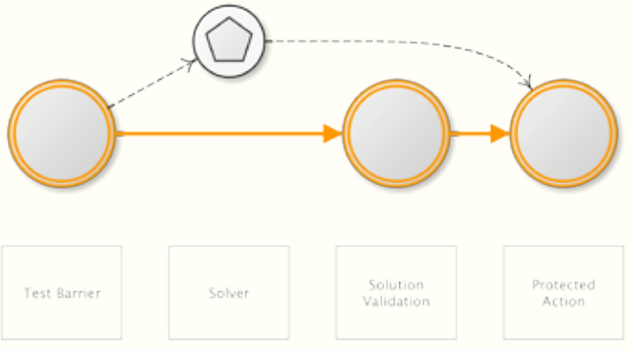
- Education
- Entertainment
- Financial
- Government
- Health
- Retail
- Technology
- Social Networking

Parties Affected

- Few Individual Users
- Many Users
- Application Owner
- Third Parties
- Society

Data Commonly Misused

- Authentication Credentials
- Payment Cardholder Data
- Other Financial Data
- Medical Data
- Other Personal Data
- Intellectual Property
- Other Business Data
- Public Information



Description

Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) challenges are used to distinguish normal users from bots. Automation is used in an attempt to analysis and determination the answer to visual and/or aural CAPTCHA tests and related puzzles. Apart from conventional visual and aural CAPTCHA, puzzle solving mini games or arithmetical exercises are sometimes used. Some of these may include context-specific challenges.

The process that determines the answer may utilise tools to perform optical character recognition, or matching against a prepared database of pre-generated images, or using other machine reading, or human farms.

Other Names and Examples

Breaking CAPTCHA; CAPTCHA breaker; CAPTCHA breaking; CAPTCHA decoding; CAPTCHA solver; CAPTCHA solving; Puzzle solving

See Also

- OAT-006 Expediting
- OAT-011 Scraping

CAPEC Category / Attack Pattern IDs

- -

WASC Threat IDs

- 21 Insufficient Anti-Automation
- 42 Abuse of Functionality

CWEs

- 804 Guessable CAPTCHA
- 841 Improper Enforcement of Behavioral Workflow

OWASP Attacks

- -

CAPTCHA Bypass

OAT-009

Solve anti-automation tests.

Symptoms

- ???

Card Cracking

OAT-010

Identify missing expiry dates and security codes for stolen payment card data by trying different values.

Sectors Targeted

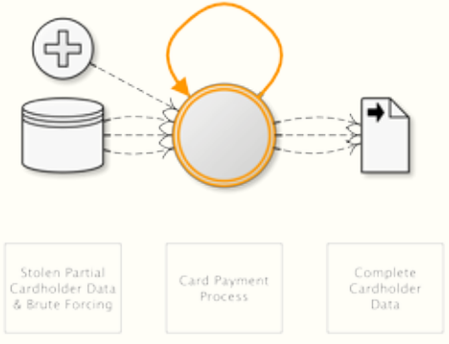
- Education
- Entertainment
- Financial
- Government
- Health
- Retail
- Technology
- Social Networking

Parties Affected

- Few Individual Users
- Many Users
- Application Owner
- Third Parties
- Society

Data Commonly Misused

- Authentication Credentials
- Payment Cardholder Data
- Other Financial Data
- Medical Data
- Other Personal Data
- Intellectual Property
- Other Business Data
- Public Information



Description

Brute force attack agsinst application payment card process to identify the missing values for start date, expiry date and card security code (CSC), referred to in many ways including card validation number 2 (CVN2), card validation code (CVC), card verification value (CV2) card identification number (CID).

When these values are known as well as the Primary Account Number (PAN), **OAT-001 Carding** is used to validate the details, and **OAT-012 Cashing Out** to obtain goods or cash.

Other Names and Examples

Brute forcing credit card information; Card brute forcing; Credit card cracking

See Also

- OAT-001 Carding
- OAT-012 Cashing Out

CAPEC Category / Attack Pattern IDs

- 112 Brute Force
- 210 Abuse of Functionality

WASC Threat IDs

- 11 Brute Force
- 21 Insufficient Anti-Automation
- 42 Abuse of Functionality

CWEs

- 799 Improper Control of Interaction Frequency
- 837 Improper Enforcement of a Single, Unique Action

OWASP Attacks

- Abuse of Functionality
- Brute Force Attack

Card Cracking

OAT-010

Identify missing expiry dates and security codes for stolen payment card data by trying different values.

Symptoms

- ???

Scraping

OAT-011

Collect application content and/or other data for use elsewhere.

Sectors Targeted

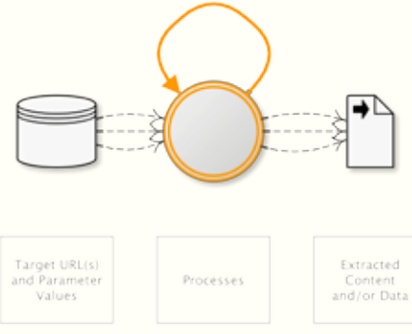
- Education
- Entertainment
- Financial
- Government
- Health
- Retail
- Technology
- Social Networking

Parties Affected

- Few Individual Users
- Many Users
- Application Owner
- Third Parties
- Society

Data Commonly Misused

- Authentication Credentials
- Payment Cardholder Data
- Other Financial Data
- Medical Data
- Other Personal Data
- Intellectual Property
- Other Business Data
- Public Information



Description

Collecting accessible data and/or processed output from the application. Some scraping may use fake or compromised accounts, or the information may be accessible without authentication. The scraper may attempt to read all accessible paths and parameter values for web site pages and web APIs, collecting the responses and extracting data from them. Scraping may occur in real time, or be more periodic in nature. Some Scraping may be to gain insight how it is constructed and operates - perhaps for cryptanalysis, reverse engineering, or session analysis.

When another application is being used as an intermediary between the user(s) and the real application, see **OAT-020 Account Aggregation**. Where the intent is to obtain cash or goods, see **OAT-012 Cashing Out** instead.

Other Names and Examples

API provisioning; Bargain hunting; Comparative shopping; Content scraping; Data aggregation; Database scraping; Harvesting; Meta search scraper; Mining; Mirroring; Pagejacking; Powering APIs; Ripping; Scraper bot; Screen scraping; Search engine bot; Social media bot

See Also

- OAT-012 Cashing Out
- OAT-018 Footprinting
- OAT-020 Account Aggregation

CAPEC Category / Attack Pattern IDs

- 167 Lifting Sensitive Data from the Client
- 210 Abuse of Functionality
- 281 Analyze Target

WASC Threat IDs

- 21 Insufficient Anti-Automation
- 42 Abuse of Functionality

CWEs

- 799 Improper Control of Interaction Frequency

OWASP Attacks

- Abuse of Functionality

Scraping

OAT-011

Collect application content and/or other data for use elsewhere.

Symptoms

- ???

Cashing Out

OAT-012

Buy goods or obtain cash utilising validated stolen payment card or other user account data.

Sectors Targeted

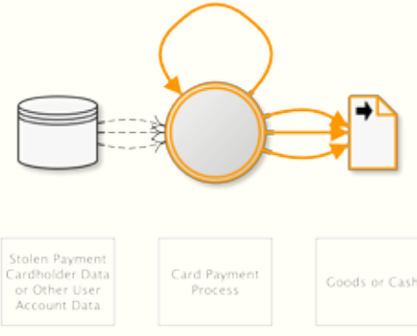
- Education
- Entertainment
- Financial
- Government
- Health
- Retail
- Technology
- Social Networking

Parties Affected

- Few Individual Users
- Many Users
- Application Owner
- Third Parties
- Society

Data Commonly Misused

- Authentication Credentials
- Payment Cardholder Data
- Other Financial Data
- Medical Data
- Other Personal Data
- Intellectual Property
- Other Business Data
- Public Information



Description

Obtaining currency or higher-value merchandise via the application using stolen previously validated payment cards or other account log in credentials. Cashing Out is sometimes may be undertaken in conjunction with product return fraud. For financial transactions, this is usually a transfer of funds to a mule's account. For payment cards, this activity may occur following **OAT-001 Carding** of bulk stolen data, or **OAT-010 Card Cracking,** and the goods are dropped at a reshipper's address. The refunding of payments via non-financial applications (e.g. tax refunds, claims payment) is also included in Cashing Out.

Obtaining other information of value from the application is **OAT-011 Scraping** instead.

Other Names and Examples

Money laundering; Online credit card fraud; Online payment card fraud; Refund fraud; Stolen identity refund fraud (SIRF)

See Also

- OAT-001 Carding
- OAT-011 Scraping
- OAT-010 Card Cracking

CAPEC Category / Attack Pattern IDs

- 210 Abuse of Functionality

WASC Threat IDs

- 21 Insufficient Anti-Automation
- 42 Abuse of Functionality

CWEs

- 799 Improper Control of Interaction Frequency
- 837 Improper Enforcement of a Single, Unique Action

OWASP Attacks

- Abuse of Functionality

Cashing Out

OAT-012

Buy goods or obtain cash utilising validated stolen payment card or other user account data.

Symptoms

- ???

OAT-013

Sniping

Last minute bid or offer, for goods or services.

Sectors Targeted

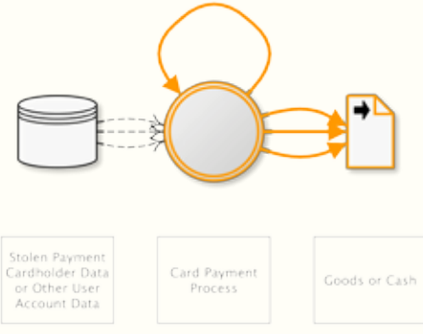
- Education
- Entertainment
- Financial
- Government
- Health
- Retail
- Technology
- Social Networking

Parties Affected

- Few Individual Users
- Many Users
- Application Owner
- Third Parties
- Society

Data Commonly Misused

- Authentication Credentials
- Payment Cardholder Data
- Other Financial Data
- Medical Data
- Other Personal Data
- Intellectual Property
- Other Business Data
- Public Information



Description

The defining characteristic of Sniping is an action undertaken at the latest opportunity to achieve a particular objective, leaving insufficient time for another user to bid/offer. Sniping can also be the automated exploitation of system latencies in the form of timing attacks. Careful timing and prompt action are necessary parts. It is most well known as auction sniping, but the same threat event can be used in other types of application. Sniping normally leads to some disbenefit for other users, and sometimes that might be a form of denial of service.

In contrast **OAT-005 Scalping** is the acquisition of limited availability of sought-after goods or services, and **OAT-006 Expediting** is the general hastening of progress.

Other Names and Examples

Auction sniping; Bid sniper; Front-running; Last look; Last minute bet; Timing attack

See Also

- OAT-005 Scalping
- OAT-006 Expediting
- OAT-015 Denial of Service

CAPEC Category / Attack Pattern IDs

- 210 Abuse of Functionality

WASC Threat IDs

- 21 Insufficient Anti-Automation
- 42 Abuse of Functionality

CWEs

- -

OWASP Attacks

- Abuse of Functionality

Sniping

OAT-013

Last minute bid or offer, for goods or services.

Symptoms

- ???

OAT-014 Vulnerability Scanning

Crawl and fuzz application to identify weaknesses and possible vulnerabilities.

Sectors Targeted

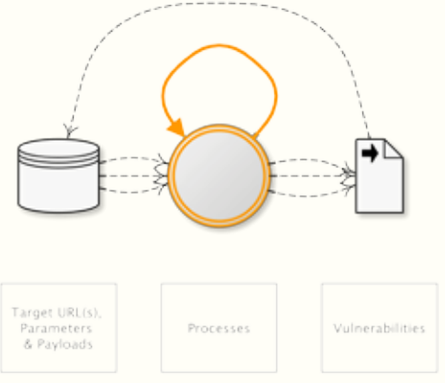
- Education
- Entertainment
- Financial
- Government
- Health
- Retail
- Technology
- Social Networking

Parties Affected

- Few Individual Users
- Many Users
- Application Owner
- Third Parties
- Society

Data Commonly Misused

- Authentication Credentials
- Payment Cardholder Data
- Other Financial Data
- Medical Data
- Other Personal Data
- Intellectual Property
- Other Business Data
- Public Information



Description

Systematic enumeration and examination of identifiable, guessable and unknown content locations, paths, file names, parameters, in order to find weaknesses and points where a security vulnerability might exist. Vulnerability Scanning includes both malicious scanning and friendly scanning by an authorised vulnerability scanning engine. It differs from **OAT-011 Scraping** in that its aim is to identify potential vulnerabilities.

The exploitation of individual vulnerabilities is not included in the scope of this ontology, but this process of scanning, along with **OAT-018 Footprinting**, **OAT-004 Fingerprinting** and **OAT-011 Scraping** often form part of application penetration testing.

Other Names and Examples

Active/Passive scanning; Application-specific vulnerability discovery; Identifying vulnerable content management systems (CMS) and CMS components; Known vulnerability scanning; Malicious crawling; Vulnerability reconnaissance

See Also

- OAT-004 Fingerprinting
- OAT-011 Scraping
- OAT-018 Footprinting

CAPEC Category / Attack Pattern IDs

- -

WASC Threat IDs

- 21 Insufficient Anti-Automation

CWEs

- 799 Improper Control of Interaction Frequency

OWASP Attacks

- -

Vulnerability Scanning

OAT-014

Crawl and fuzz application to identify weaknesses and possible vulnerabilities.

Symptoms

- ???

OAT-015

Denial of Service

Target resources of the application and database servers, or individual user accounts, to achieve denial of service (DoS).

Sectors Targeted

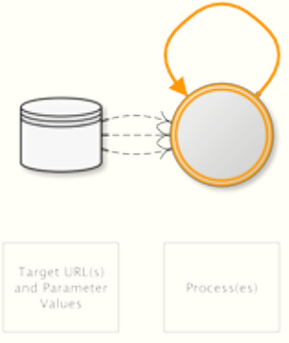
- Education
- Entertainment
- Financial
- Government
- Health
- Retail
- Technology
- Social Networking

Parties Affected

- Few Individual Users
- Many Users
- Application Owner
- Third Parties
- Society

Data Commonly Misused

- Authentication Credentials
- Payment Cardholder Data
- Other Financial Data
- Medical Data
- Other Personal Data
- Intellectual Property
- Other Business Data
- Public Information



Description

Usage may resemble legitimate application usage, but leads to exhaustion of resources such as file system, memory, processes, threads, CPU, human or financial resources. The resources might be related to web, application or databases servers or other services supporting the application such as third party APIs, included third-party hosted content, or content delivery networks (CDNs). The application may be affected as a whole, or the attack may be against individual users such as account lockout.

This ontology's scope excludes other forms of denial of service that affect web applications, namely HTTP Flood DoS (GET, POST, Header with/without TLS), HTTP Slow DoS, IP layer 3 DoS, and TCP layer 4 DoS. Those protocol and lower layer aspects are covered adequately in other taxonomies and lists.

Other Names and Examples

Account lockout; App layer DDoS; Asymmetric resource consumption (amplification); Business logic DDoS; Cash overflow; Forced deadlock; Hash DoS; Inefficient code; Indexer DoS; Large files DoS; Resource depletion, locking or exhaustion; Sustained client engagement

See Also

- OAT-005 Scalping
- OAT-013 Sniping
- OAT-017 Spamming
- OAT-019 Account Creation

CAPEC Category / Attack Pattern IDs

- 2 Inducing Account Lockout
- 25 Forced Deadlock
- 119 Deplete Resources

CWEs

- 399 Resource Management Errors
- 645 Overly Restrictive Account Lockout Mechanism

WASC Threat IDs

- 10 Denial of Service

OWASP Attacks

- Account Lockout Attack
- Cash Overflow
- Denial of Service
- Resource Depletion

Denial of Service

OAT-015

Target resources of the application and database servers, or individual user accounts, to achieve denial of service (DoS).

Symptoms

- ???

Skewing

OAT-016

Repeated link clicks, page requests or form submissions intended to alter some metric.

Sectors Targeted

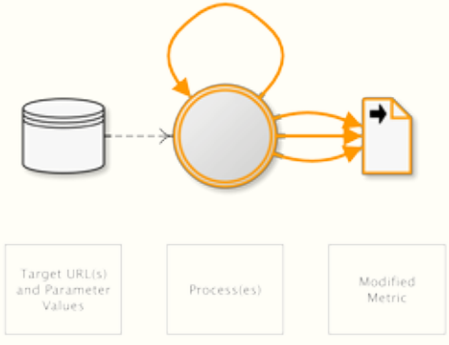
- Education
- Entertainment
- Financial
- Government
- Health
- Retail
- Technology
- Social Networking

Parties Affected

- Few Individual Users
- Many Users
- Application Owner
- Third Parties
- Society

Data Commonly Misused

- Authentication Credentials
- Payment Cardholder Data
- Other Financial Data
- Medical Data
- Other Personal Data
- Intellectual Property
- Other Business Data
- Public Information



Description

Automated repeated clicking or requesting or submitting content, affecting application-based metrics such as counts, and measures of frequency and/or rate. The metric or measurement may be visible to users (e.g. betting odds, likes, market pricing, visitor count, poll results, reviews) or hidden (e.g. application usage statistics, business performance indicators). Metrics may affect individuals as well as application-owner e.g. user reputation, influence others, gain fame, or undermine someone else's reputation.

For malicious alteration of digital advertisement metrics, see **OAT-003 Ad Fraud** instead.

Other Names and Examples

Biasing KPIs; Boosting friends, visitors, and likes; Click fraud; Election fraud; Hit count fraud; Market distortion; Metric and statistic skewing; Page impression fraud; Poll fraud; Poll skewing; Poll/voting subversion; Rating/review skewing; SEO; Stock manipulation; Survey skewing

See Also

- OAT-003 Ad Fraud
- OAT-017 Spamming
- OAT-019 Account Creation

CAPEC Category / Attack Pattern IDs

- 210 Abuse of Functionality

WASC Threat IDs

- 21 Insufficient Anti-Automation
- 42 Abuse of Functionality

CWEs

- 799 Improper Control of Interaction Frequency
- 837 Improper Enforcement of a Single, Unique Action

OWASP Attacks

- Abuse of Functionality

Skewing

OAT-016

Repeated link clicks, page requests or form submissions intended to alter some metric.

Symptoms

- ???

OAT-017

Spamming

Malicious and/or more benign information addition, that appears in public or private content, databases or user messages.

Sectors Targeted

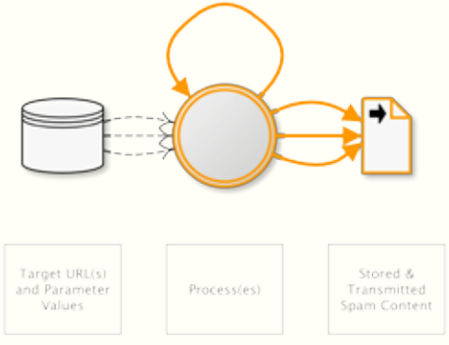
- Education
- Entertainment
- Financial
- Government
- Health
- Retail
- Technology
- Social Networking

Parties Affected

- Few Individual Users
- Many Users
- Application Owner
- Third Parties
- Society

Data Commonly Misused

- Authentication Credentials
- Payment Cardholder Data
- Other Financial Data
- Medical Data
- Other Personal Data
- Intellectual Property
- Other Business Data
- Public Information



Description

Malicious content can include malware, Iframe distribution, photographs & videos, advertisements, referrer spam and tracking/surveillance code. The content might be less overtly malicious but be an attempt to cause mischief, undertake search engine optimisation (SEO) or to dilute/hide other posts.

The mass abuse of broken form-to-email and form-to-SMS functions to send messages to unintended recipients is not included in this threat event, or any other in this ontology, since those are considered to be the exploitation of implementation flaws alone.

For multiple use that distorts metrics, see **OAT-016 Skewing** instead.

Other Names and Examples

Blog spam; Bulletin board spam; Comment spam; Content spam; Content spoofing; Form spam; Forum spam; Guest book spam; Referrer spam; Response form spam; Review spam; SEO spam; Spam crawlers; Spam 2.0; Spambot; Wiki spam; Twitter spam

See Also

- OAT-015 Denial of Service
- OAT-016 Skewing
- OAT-019 Account Creation

CAPEC Category / Attack Pattern IDs

- 210 Abuse of Functionality

WASC Threat IDs

- 21 Insufficient Anti-Automation
- 42 Abuse of Functionality

CWEs

- 506 Embedded Malicious Code
- 799 Improper Control of Interaction Frequency
- 837 Improper Enforcement of a Single, Unique Action

OWASP Attacks

- Abuse of Functionality

Spamming

OAT-017

Malicious and/or more benign information addition, that appears in public or private content, databases or user messages.

Symptoms

- ???

OAT-018

Footprinting

Probe and explore application to identify its constituents and properties.

Sectors Targeted

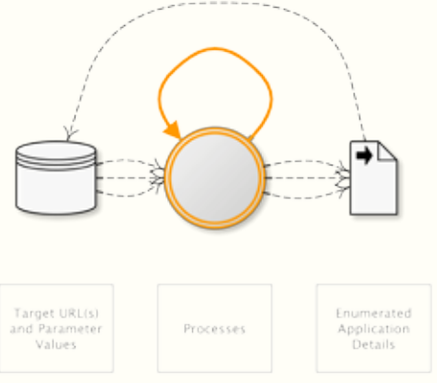
- Education
- Entertainment
- Financial
- Government
- Health
- Retail
- Technology
- Social Networking

Parties Affected

- Few Individual Users
- Many Users
- Application Owner
- Third Parties
- Society

Data Commonly Misused

- Authentication Credentials
- Payment Cardholder Data
- Other Financial Data
- Medical Data
- Other Personal Data
- Intellectual Property
- Other Business Data
- Public Information



Description

Information gathering with the objective of learning as much as possible about the composition, configuration and security mechanisms of the application. Unlike Scraping, Footprinting is an enumeration of the application itself, rather than the data. It is used to identify all the URL paths, parameters and values and process sequences (i.e. to determine entry points, also called the attack surface). As the application is explored, additional paths will be identified which in turn need to be examined.

Footprinting can also include brute force, dictionary and guessing of file and directory names. Fuzzing may also be used to identify further application resources and capabilities. However, it does not include attempts to exploit weaknesses.

Other Names and Examples

Application analysis; API discovery; Application enumeration; Automated scanning; CGI scanning; Crawler; Crawling; Excavation; Forced browsing; Forceful browsing; Fuzzing; Micro service discovery; Scanning; Spidering; WSDL scanning

See Also

- OAT-004 Fingerprinting
- OAT-011 Scraping

CAPEC Category / Attack Pattern IDs

- 169 Footprinting

WASC Threat IDs

- 45 Fingerprinting

CWEs

- 200 Information Exposure

OWASP Attacks

- -

Footprinting

OAT-018

Probe and explore application to identify its constituents and properties.

Symptoms

- ???

OAT-019

Account Creation

Create multiple accounts for subsequent misuse.

Sectors Targeted

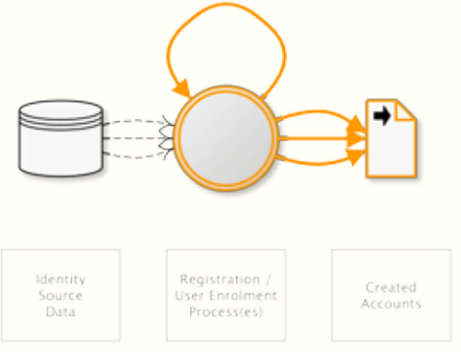
- Education
- Entertainment
- Financial
- Government
- Health
- Retail
- Technology
- Social Networking

Parties Affected

- Few Individual Users
- Many Users
- Application Owner
- Third Parties
- Society

Data Commonly Misused

- Authentication Credentials
- Payment Cardholder Data
- Other Financial Data
- Medical Data
- Other Personal Data
- Intellectual Property
- Other Business Data
- Public Information



Description

Bulk account creation, and sometimes profile population, by using the application's account signup processes. The accounts are used subsequently for misuse such as generating content spam, laundering cash and goods, spreading malware, affecting reputation, causing mischief, and skewing search engine optimisation (SEO), reviews and surveys.

Account Creation generates new accounts, rather than attempting to use existing accounts - see **OAT-007 Credential Cracking** and **OAT-008 Credential Stuffing**.

Other Names and Examples

Account pharming; Fake account; Fake social media account creation; Impersonator bot; Massive account registration; New account creation; Registering many user accounts

See Also

- OAT-007 Credential Cracking
- OAT-008 Credential Stuffing

CAPEC Category / Attack Pattern IDs

- 210 Abuse of Functionality

WASC Threat IDs

- 21 Insufficient Anti-Automation
- 42 Abuse of Functionality

CWEs

- 799 Improper Control of Interaction Frequency
- 837 Improper Enforcement of a Single, Unique Action
- 841 Improper Enforcement of Behavioral Workflow

OWASP Attacks

- Abuse of Functionality

Account Creation OAT-019

Create multiple accounts for subsequent misuse.

Symptoms

- ???

OAT-020 Account Aggregation

Use by an intermediary application to collect together accounts and interact on their behalves.

Sectors Targeted

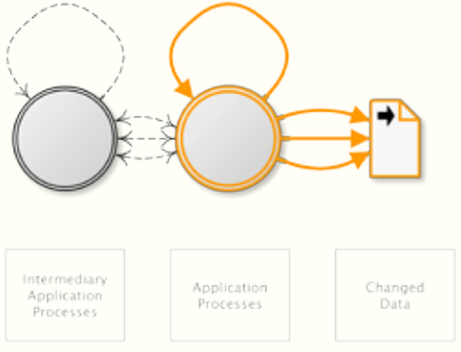
- Education
- Entertainment
- Financial
- Government
- Health
- Retail
- Technology
- Social Networking

Parties Affected

- Few Individual Users
- Many Users
- Application Owner
- Third Parties
- Society

Data Commonly Misused

- Authentication Credentials
- Payment Cardholder Data
- Other Financial Data
- Medical Data
- Other Personal Data
- Intellectual Property
- Other Business Data
- Public Information



Description

Compilation of credentials and information from multiple application accounts into another system. This aggregation application may be used by a single user to merge information from multiple applications, or alternatively to merge information of many users of a single application. Commonly used for aggregating social media accounts, email accounts and financial accounts in order to obtain a consolidated overview, to provide integrated reporting and analysis, and to simplify usage and consumption by the user and/or their professional advisors. May include making changes to account properties and interacting with the aggregated application's functionality.

For other forms of data harvesting, including the distribution of content, see **OAT-011 Scraping**. For hastening progress, see **OAT-006 Expediting** instead.

Other Names and Terms

Account aggregation; Aggregator; Client aggregator; Data aggregation; Financial account aggregator

See Also

- OAT-006 Expediting
- OAT-011 Scraping
- OAT-0 19 Account Creation

CAPEC Category / Attack Pattern IDs

- 167 Lifting Sensitive Data from the Client
- 210 Abuse of Functionality

WASC Threat IDs

- 21 Insufficient Anti-Automation
- 42 Abuse of Functionality

CWEs

- 799 Improper Control of Interaction Frequency

OWASP Attacks

- Abuse of Functionality

Account Aggregation OAT-020

Use by an intermediary application to collect together accounts and interact on their behalves.

Symptoms

- ???