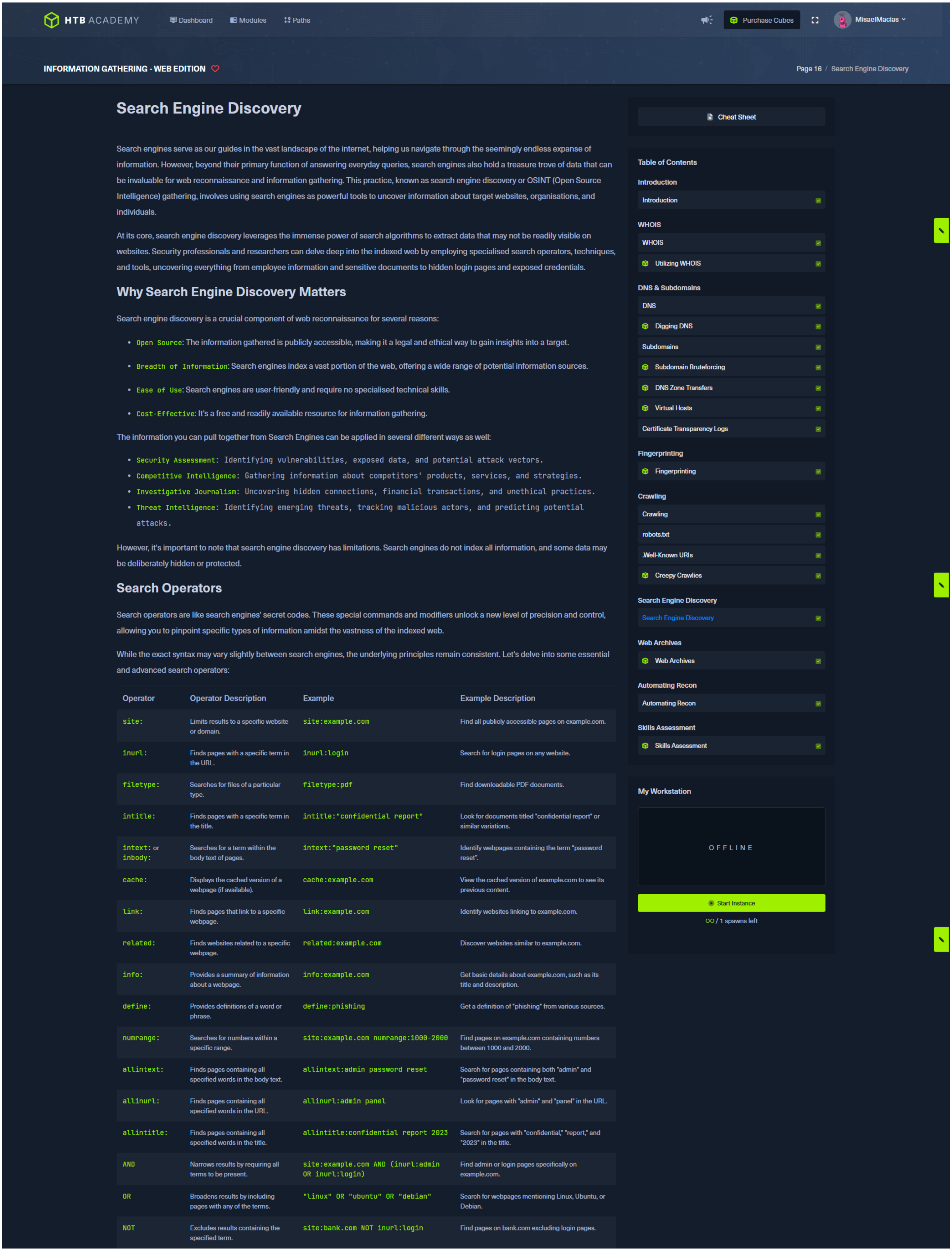


Search Engine Discovery

Search engines serve as our guides in the vast landscape of the internet, helping us navigate through the seemingly endless expanse of information. However, beyond their primary function of answering everyday queries, search engines also hold a treasure trove of data that can be invaluable for web reconnaissance and information gathering. This practice, known as search engine discovery or OSINT (Open Source Intelligence) gathering, involves using search engines as powerful tools to uncover information about target websites, organisations, and individuals.

At its core, search engine discovery leverages the immense power of search algorithms to extract data that may not be readily visible on websites. Security professionals and researchers can delve deep into the indexed web by employing specialised search operators, techniques, and tools, uncovering everything from employee information and sensitive documents to hidden login pages and exposed credentials.

Why Search Engine Discovery Matters

Search engine discovery is a crucial component of web reconnaissance for several reasons:

- `Open Source`: The information gathered is publicly accessible, making it a legal and ethical way to gain insights into a target.
- `Breadth of Information`: Search engines index a vast portion of the web, offering a wide range of potential information sources.
- `Ease of Use`: Search engines are user-friendly and require no specialised technical skills.
- `Cost-Effective`: It's a free and readily available resource for information gathering.

The information you can pull together from Search Engines can be applied in several different ways as well:

- `Security Assessment: Identifying vulnerabilities, exposed data, and potential attack vectors.`
- `Competitive Intelligence: Gathering information about competitors' products, services, and strategies.`
- `Investigative Journalism: Uncovering hidden connections, financial transactions, and unethical practices.`
- `Threat Intelligence: Identifying emerging threats, tracking malicious actors, and predicting potential attacks.`

However, it's important to note that search engine discovery has limitations. Search engines do not index all information, and some data may be deliberately hidden or protected.

Search Operators

Search operators are like search engines' secret codes. These special commands and modifiers unlock a new level of precision and control, allowing you to pinpoint specific types of information amidst the vastness of the indexed web.

While the exact syntax may vary slightly between search engines, the underlying principles remain consistent. Let's delve into some essential and advanced search operators:

Operator	Operator Description	Example	Example Description
`site:`	Limits results to a specific website or domain.	`site:example.com`	Find all publicly accessible pages on example.com.
`inurl:`	Finds pages with a specific term in the URL.	`inurl:login`	Search for login pages on any website.
`filetype:`	Searches for files of a particular type.	`filetype:pdf`	Find downloadable PDF documents.
`intitle:`	Finds pages with a specific term in the title.	`intitle:"confidential report"`	Look for documents titled "confidential report" or similar variations.
`intext:` or `inbody:`	Searches for a term within the body text of pages.	`intext:"password reset"`	Identify webpages containing the term "password reset".
`cache:`	Displays the cached version of a webpage (if available).	`cache:example.com`	View the cached version of example.com to see its previous content.
`link:`	Finds pages that link to a specific webpage.	`link:example.com`	Identify websites linking to example.com.
`related:`	Finds websites related to a specific webpage.	`related:example.com`	Discover websites similar to example.com.
`info:`	Provides a summary of information about a webpage.	`info:example.com`	Get basic details about example.com, such as its title and description.
`define:`	Provides definitions of a word or phrase.	`define:phishing`	Get a definition of "phishing" from various sources.
`numrange:`	Searches for numbers within a specific range.	`site:example.com numrange:1000-2000`	Find pages on example.com containing numbers between 1000 and 2000.
`allintext:`	Finds pages containing all specified words in the body text.	`allintext:admin password reset`	Search for pages containing both "admin" and "password reset" in the body text.
`allinurl:`	Finds pages containing all specified words in the URL.	`allinurl:admin panel`	Look for pages with "admin" and "panel" in the URL.
`allintitle:`	Finds pages containing all specified words in the title.	`allintitle:confidential report 2023`	Search for pages with "confidential," "report," and "2023" in the title.
`AND`	Narrows results by requiring all terms to be present.	`site:example.com AND (inurl:admin OR inurl:login)`	Find admin or login pages specifically on example.com.
`OR`	Broadens results by including pages with any of the terms.	`"linux" OR "ubuntu" OR "debian"`	Search for webpages mentioning Linux, Ubuntu, or Debian.
`NOT`	Excludes results containing the specified term.	`site:bank.com NOT inurl:login`	Find pages on bank.com excluding login pages.

Cheat Sheet

Table of Contents

Introduction

WHOIS

DNS & Subdomains

Fingerprinting

Crawling

Search Engine Discovery

Web Archives

Automating Recon

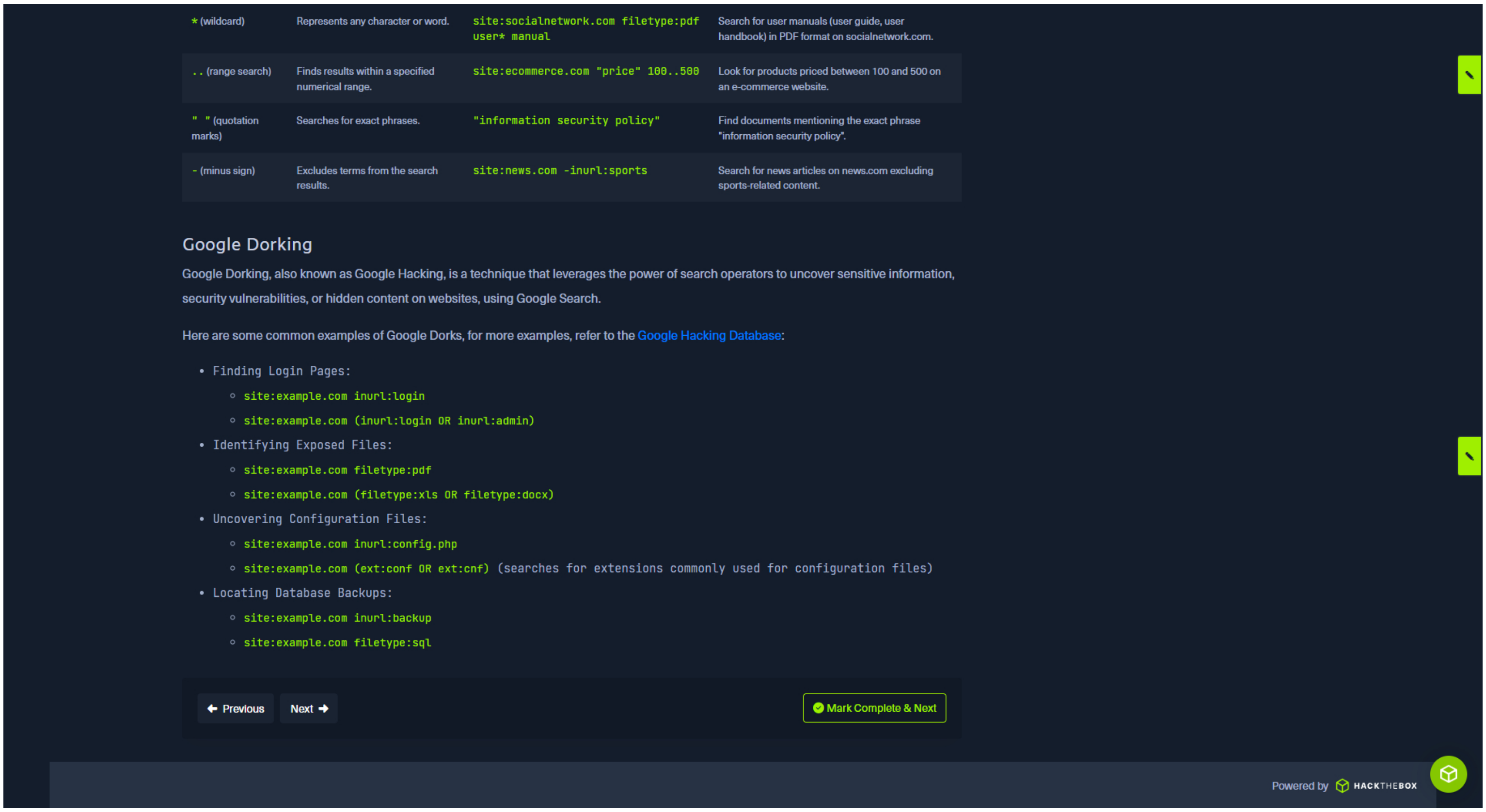
Skills Assessment

My Workstation

OFFLINE

Start Instance

∞ / 1 spawns left



`*` (wildcard)	Represents any character or word.	`site:socialnetwork.com filetype:pdf user* manual`	Search for user manuals (user guide, user handbook) in PDF format on socialnetwork.com.
`..` (range search)	Finds results within a specified numerical range.	`site:ecommerce.com "price" 100..500`	Look for products priced between 100 and 500 on an e-commerce website.
`" "` (quotation marks)	Searches for exact phrases.	`"information security policy"`	Find documents mentioning the exact phrase "information security policy".
`-` (minus sign)	Excludes terms from the search results.	`site:news.com -inurl:sports`	Search for news articles on news.com excluding sports-related content.

Google Dorking

Google Dorking, also known as Google Hacking, is a technique that leverages the power of search operators to uncover sensitive information, security vulnerabilities, or hidden content on websites, using Google Search.

Here are some common examples of Google Dorks, for more examples, refer to the Google Hacking Database:

- Finding Login Pages:
  - `site:example.com inurl:login`
  - `site:example.com (inurl:login OR inurl:admin)`
- Identifying Exposed Files:
  - `site:example.com filetype:pdf`
  - `site:example.com (filetype:xls OR filetype:docx)`
- Uncovering Configuration Files:
  - `site:example.com inurl:config.php`
  - `site:example.com (ext:conf OR ext:cnf)` (searches for extensions commonly used for configuration files)
- Locating Database Backups:
  - `site:example.com inurl:backup`
  - `site:example.com filetype:sql`