OSINT: Open Source Intelligence

BlackPerl

Like Repost

Why is OSINT Important for Investigative Journalism?

Verification
- Check Sources
- Authenticate Media
- Fact-Check Statements

Lead Generation
- Identify Key Players
- Uncover Connections
- Find Documents

Context
- Historical Data
- Geographical Information
- Social Context

Getting Started with OSINT

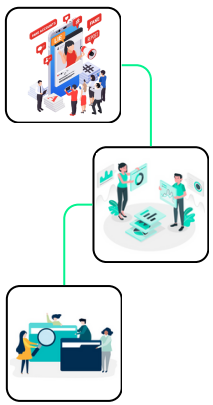
Use the Right Tools
- Search Engines
- Social Media Analysis
- Image Verification
- Public Records
- Mapping and Geolocation

Master Search Techniques
- Boolean Operators - AND, OR, and NOT
- Site Search - site:example.com
- File Type Search - filetype:pdf
- Quotation Marks - quotes

Social Media Investigation
- Profiles and Posts
- Hashtags and Keywords
- Geotagging
- Archived Data

Image and Video Analysis
- Metadata
- Reverse Image Search
- Frame Analysis

Public Records and Databases
- Government Databases
- Corporate Records
- Academic Publications



Privacy and Ethics

Verify Sources
Respect Privacy
Legal Boundaries

Objective: Investigating a Political Scandal

- Background Research
- Social Media Analysis
- Public Records
- Image Verification
- Network Analysis

How to Get Started with Social Media OSINT?

1. Define Your Objectives
2. Choose Relevant Social Media Platforms
3. Use OSINT Tools and Techniques
4. Develop Search Queries
5. Maintain Ethical Standards
6. Stay Updated and Learn Continuously
7. Document and Analyze Findings

Commonly Used Tools for Social Media OSINT

Data Collection and Aggregation Tools
OSINT

Social Media Monitoring and Management Tools
Hootsuite buffer

Analysis and Visualization Tools
IBM Watson Palantir NODEXL

Search Engines and Advanced Search Techniques
Google Advanced Search Yandex

Browser Extensions and Plugins
facebook LinkedIn

Example OSINT Script for Social Media Platforms

Facebook OSINT

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import requests

def facebook_osint(company_name):
 url = f"https://graph.facebook.com/v13.0/{company_name}?
fields=id,name,about,website,phone,email&access_token=
<your_access_token>"
 response = requests.get(url)
 if response.status_code == 200:
 data = response.json()
 print(f"Name: {data.get('name')}")
 print(f"About: {data.get('about')}")
 print(f"Website: {data.get('website')}")
 print(f"Phone: {data.get('phone')}")
 print(f"Email: {data.get('email')}")
 else:
 print(f"Failed to retrieve data from Facebook API. Status code:
{response.status_code}")

if __name__ == "__main__":
 company_name = "examplecompany"
 facebook_osint(company_name)
```

Twitter OSINT

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import tweepy

# Twitter API credentials
consumer_key = "<your_consumer_key>"
consumer_secret = "<your_consumer_secret>"
access_token = "<your_access_token>"
access_token_secret = "<your_access_token_secret>"

def twitter_osint(hashtag):
    auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_token_secret)
    api = tweepy.API(auth, wait_on_rate_limit=True)

    tweets = tweepy.Cursor(api.search, q=hashtag, lang="en",
tweet_mode="extended").items(10)
    for tweet in tweets:
        print(f"Tweet ID: {tweet.id}")
        print(f"Username: {tweet.user.screen_name}")
        print(f"Tweet: {tweet.full_text}")
        print(f"Retweet Count: {tweet.retweet_count}")
        print(f"Favorite Count: {tweet.favorite_count}")
        print(f"Created At: {tweet.created_at}")
        print()

if __name__ == "__main__":
    hashtag = "#examplehashtag"
    twitter_osint(hashtag)
```

OSINT: Open Source Intelligence

BlackPerl

Like Repost

Example OSINT Script for Social Media Platforms

LinkedIn OSINT

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-

from linkedin_api import Linkedin

# LinkedIn credentials
username = "<your_username>"
password = "<your_password>"

def linkedin_osint(company_name):
    linkedin = Linkedin(username, password)
    company = linkedin.get_organization(company_name)
    if company:
        print(f"Name: {company['name']}")
        print(f"Description: {company['description']}")
        print(f"Website: {company['websiteUrl']}")
        print(f"Employee Count: {company['employeeCountRange']}")
    else:
        print("Failed to retrieve data from LinkedIn API.")

if __name__ == "__main__":
    company_name = "example-company"
    linkedin_osint(company_name)
```

Instagram OSINT

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import instaloader

def instagram_osint(username):
    L = instaloader.Instaloader()
    profile = instaloader.Profile.from_username(L.context, username)

    print(f"Username: {profile.username}")
    print(f"Full Name: {profile.full_name}")
    print(f"Biography: {profile.biography}")
    print(f"Followers: {profile.followers}")
    print(f"Following: {profile.followees}")

    print("\nRecent Posts:")
    for post in profile.get_posts():
        print(f"Post ID: {post.mediaid}")
        print(f"Likes: {post.likes}")
        print(f"Comments: {post.comments}")
        print(f"Caption: {post.caption}")
        print(f"Link: https://www.instagram.com/p/{post.shortcode}/")
        print()

if __name__ == "__main__":
    username = "example_username"
    instagram_osint(username)
```

YouTube OSINT

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-

from googleapiclient.discovery import build

# YouTube API credentials
api_key = "<your_api_key>"

def youtube_osint(search_query):
    youtube = build('youtube', 'v3', developerKey=api_key)
    request = youtube.search().list(
        q=search_query,
        part='snippet',
        type='video',
        maxResults=5
    )
    response = request.execute()

    for item in response['items']:
        print(f"Title: {item['snippet']['title']}")
        print(f"Channel: {item['snippet']['channelTitle']}")
        print(f"Description: {item['snippet']['description']}")
        print(f"Link: https://www.youtube.com/watch?v={item['id']
['videoId']}")
        print()

if __name__ == "__main__":
    search_query = "example topic"
    youtube_osint(search_query)
```

Objective:

Facebook: Gather information about a company's public profile on Facebook.

Twitter : Monitor tweets related to a specific hashtag.

LinkedIn: Retrieve information from a LinkedIn company page.

Instagram: Extract posts and engagement metrics from a user's Instagram profile.

Mastering OSINT Evasion Techniques

Google Dorking

Filter	Example	Description
site	site:starbucks.com	Only displays results related to "starbucks.com"
-site	-site:one.starbucks.com	"one.starbucks.com" will not appear in search results
inurl	inurl:mocha	Only webpages with the word "mocha" in its URL will appear
intitle	intitle:coffee	Only displays results which have "coffee" in its title
intext	intext:coffee	Only displays results with "coffee" in the body of the website
filetype	filetype:pdf	Only returns PDF documents

- **site: site:** example.com
- **filetype:** filetype:pdf
- **intitle:** intitle:"index of"
- **inurl:** inurl:admin

Shodan

- **Exposed servers**
- **Open ports**
- **Vulnerable devices**

Maltego

- **Create a new graph**
- **Add entities**
- **Run transforms**

Anonymous Browsing and Communication

- ☑ **Tor and VPNs**
- ☑ **Encrypted Messaging**
- ☑ **Private Email Services**
- ☑ **Secure File Sharing**
- ☑ **Browser Fingerprinting Prevention**
- ☑ **Script and Tracker Blockers**
- ☑ **Virtual Machines (VMs) and Live Operating Systems**
- ☑ **Temporary and Disposable Accounts**
- ☑ **Regular Data Deletion**
- ☑ **VUse Multiple Identities**

Social Media Hygiene

- ☑ **Private Profiles**
- ☑ **Scrubbing Metadata**
- ☑ **Decoy Accounts**
- ☑ **Regular Audits and Cleanup**
- ☑ **Limit Cross-Platform Sharing**
- ☑ **Control Tagging and Mentions**

OSINT: Open Source Intelligence

Social Media Hygiene

- Alias and Pseudonyms
- Mindful Posting Habits
- Deactivate or Delete Inactive Accounts
- Use Social Media Privacy Tools
- Avoid Oversharing in Public Groups

Metadata and Geotagging

- Disable Geotagging
- On Cameras
- Use Metadata Anonymizers
- Automated Processes
- Audit Your Existing Files
- Batch Processing
- Use Anonymous File Formats
- PDFs and Documents
- Be Mindful of File Sharing Services
- Avoid Social Media Uploads
- Encrypt Sensitive Files
- Secure Storage
- Check Metadata on Received Files
- Monitor and Control Third-Party Apps
- Location Sharing

Leveraging the Dark Web

- Tor for Access
- Alternative Networks

Employing Misinformation and Digital Deception

- Deepfakes and AI-Generated Content
- False Narratives

Quick Overview

OSIT

Getting started

Gather Basic Information

- Start with What You Know
- Define Your Requirements
- Gather the Data
- Analyze Collected Data
- Pivot Using New Data
- Validate Assumptions
- Generate a Report

Real Name Searches

- Governmental Resources
- Google Dorks(google dorks)
- People Search Websites

Username Searches

- Reverse Username Lookup
- Google Dorks for Usernames
- Username Search Tools

Email Address Searches

- Google Dorks
- Email Tools
- Compromised Databases

Phone Number Searches

- Social Media Search
- Phone Lookup Services
- PhoneInfoga

Domain Name Searches

- Google Dorks
- Whois and Reverse Whois
- Same IP and Passive DNS

Location Searches

- Geolocation Tools
- IP-Based Geolocation

Image Searches

- Reverse Image Search
- EXIF Data Analysis

SOCMINT

Use Specialized Tools

- Maltego
- Spiderfoot
- Recon-ng