Data mining Tor, social networks, OSINT with AIL Project

E.102

CIRCL Computer Incident Response Center Luxembourg

MISP Project
https://www.misp-project.org/

March 23, 2022

Introduction

Concepts - Deep Web

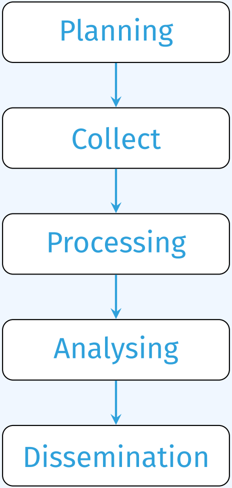
- **Deep Web** is the part of World Wide Web not indexed or directly accessible by standard web search-engines;
- This can be content hidden from **crawlers** by requiring a specific access and this can includes private social media, password-protected forums or content protected by different measures such as paywalls or specific security interface to access the information;
- A large portion of content accessible via Internet is part of the deep web[1].

[1] also called invisible web, hidden web or non-indexed web

Concepts - darknet

- **Darknet** is an overlay network running on top of Internet requiring specific software to access the network and its services;
- Tor, I2P and Freenet are the most commonly used ones. Many are used for hidden services access and some for proxy access to the Internet;
- There are **legitimate use-cases** for such network but also many **illegal or criminal usage**.

Lifecycle of collection and analysis



Collecting, processing and analysing content - web pages

- Building a search engine on the web is a challenging task because:
  - it has to crawl webpages,
  - it has to to make sense of **unstructured data**,
  - it has to **index** these data,
  - it has to provide a way to retrieve data and structure data (e.g. correlation).
- Doing so on Tor is even more challenging because:
  - services don't always want to be found,
  - parts of the dataset have to be discarded.
- in each case, it requires a lot of bandwidth, storage and computing power.

Collecting, processing and analysing content - structured data

- Some data are structured and are easy to process:
  - metadata!
  - API responses.
- Some even provide cryptographic evidences:
  - authentication mechanisms between peers,
  - OpenGPG can leak a lot of metadata
    - key ids,
    - subject of email in thunderbird,
  - Bitcoin's Blockchain is public,
  - pivoting on these data with external sources yields interesting results.

AIL design Objectives

Objectives of the session

- Show how to use and extend an open source tool to monitor web pages, pastes, forums and hidden services
- Explain challenges and the design of the AIL open source framework
- Review different **collection mechanisms** and **sources**
- Learn how to create new modules
- Learn how to use, install and start AIL
- **Supporting investigation using the AIL framework** and including it in cyber threat intelligence lifecycle

AIL Framework

From a requirement to a solution: AIL Framework

History:

- AIL initially started as an **internship project** (2014) to evaluate the feasibility to automate the analysis of (un)structured information to find leaks.
- In 2019, AIL framework is an **open source software** in Python. The software is actively used (and maintained) by CIRCL and many organisations.
- In 2020, AIL framework is now a complete project called **ail project**[2].

[2] https://github.com/ail-project/

Capabilities Overview

Common usage

- **Check** if mail/password/other sensitive information (terms tracked) leaked
- **Detect** reconnaissance of your infrastructure
- **Search** for leaks inside an archive
- **Monitor** and crawl websites

Support CERT/CSIRTs and Law Enforcement activities

- Proactive investigation: leaks detection
  - List of emails and passwords
  - Leaked database
  - AWS Keys
  - Credit-cards
  - PGP private keys
  - Certificate private keys
- Feed Passive DNS or any passive collection system
- CVE and PoC of vulnerabilities most used by attackers

Support CERT/CSIRTs and Law Enforcement activities

- Website monitoring
  - monitor booters
  - Detect encoded exploits (WebShell, malware encoded in Base64...)
  - SQL injections
- Automatic and manual submission to threat sharing and incident response platforms
  - MISP
  - TheHive
- Term/Regex/Yara monitoring for local companies/government

Sources of leaks: Paste monitoring

- Example: `https://gist.github.com/`
  - Easily storing and sharing text online
  - Used by programmers and legitimate users
    → Source code & information about configurations

Sources of leaks: Paste monitoring

- Example: `https://gist.github.com/`
  - Easily storing and sharing text online
  - Used by programmers and legitimate users
    → Source code & information about configurations
- Abused by attackers to store:
  - List of vulnerable/compromised sites
  - Software vulnerabilities (e.g. exploits)
  - Database dumps
    → User data
    → Credentials
    → Credit card details
  - More and more ...

Why so many leaks?

- Economical interests (e.g. Adversaries promoting services)
- Ransom model (e.g. To publicly pressure the victims)
- Political motives (e.g. Adversaries showing off)
- Collaboration (e.g. Criminals need to collaborate)
- Operational infrastructure (e.g. malware exfiltrating information on a pastie website)
- Mistakes and errors

Are leaks frequent?

Yes!
and we have to deal with this as a CSIRT.

- **Contacting companies or organisations** who did specific accidental leaks
- **Discussing with media** about specific case of leaks and how to make it more practical/factual for everyone
- Evaluating the economical market for cyber criminals (e.g. DDoS booters[3] or reselling personal information - reality versus media coverage)
- Analysing collateral effects of malware, software vulnerabilities or exfiltration

→ And it's important to detect them automatically.

[3] https://github.com/D4-project/

Paste monitoring at CIRCL: Statistics

- Monitored paste sites: 27
  - *gist.github.com*
  - *ideone.com*
  - ...

	2016	2017	08.2018
Collected pastes	18,565,124	19,145,300	11,591,987
Incidents	244	266	208

Table: Pastes collected and incident[4] raised by CIRCL

[4] http://www.circl.lu/pub/tr-46