Open-source intelligence gathering in Bosnia-Herzegovina

Espionage Surveillance System

Halis Duraki, duraki@linuxmail.org

End of 2018

Abstract

We designed and developed a stand-alone bot-based OSINT[1] software, built specifically for regional limits of Bosnia-Herzegovina[2]. In this document, we will present **nettis**; an artificial intelligence & neuro-linguistic based, distributed mapping infrastructure, made to map organizational (*legal entity*) network and natural persons. The tool scan and build metadata, based on licenses, open-source data processing and custom-provided roots. The system is powerful in exstensible form, offering custom modules and API (Application Programming Interface). Third-party tools (e.g. *Maltego*) can use this interface to access and register systematic entities. The implementation is developed in Crystal, a Ruby-like, static compiled, strict-type programming language, known for its' computational speed. Persistance is accomplished through several layers, including various notification system and elastic search. A remote-accessible web-based administration panel has been developed, for ease of use in special operations (e.g. *obscure, distant based operations*).

1. Background

Many *open-source intelligence* gathering tools, presented widely today, are not only limited in their true applicative form, but does not share the resources and links between the source, dataset and *marked target*. We built `nettis` as a ground-breaking, machine-learning situated piece of software, offering this feature in natural habbit, occupying machine resource to create a bot-like distributed environment; for which results establish an enormous elastic data in searchable form. Application is created with core system which scan and map several country-defined zones including but not limited to *organizational infrastructure* (e.g. hosts, www, telecommunications, legal entities), and *natural personas* (e.g. identity research, person identification, spirit definitions). Initial development is centered around *positive 0* zone (e.g. statistic linkage between different zones).

Cloud storage[3] services is used to store these information with searchable support, offering pattern-recognition and elastic data examination for further reference. `nettis` is distributed in containerized environment, creating and centering this resources to many different hosts (*to scale on different axis*). Several security protections have been made, including one for type processing, which in return greatly increase the OPSEC[4].

Efficienty is calculated through clear understanding of undefined data, while linguistic operations requires a base of *trained data* and *volumetric approach* for *ad hoc* use. Various registered sources require protection by-pass, and the implementation is naturally registered and examined in the system core. These specific features play a major role in data acquisition. The system is powered and distributed by Kubernetes cluster, creating all-around unique experience for collection processing.

[1] Open-source intelligence.
[2] Abbreviated B&H, a country in the Balkan Peninsula.
[3] Digital data is stored in logical pools.
[4] Operation security.

2. Application

Data-processing application software, like `nettis`, requires deep understanding of evidence discovery. Several abstractions have been made to make this possible. By creating a pass-through linkage between (*known*) data, new materials are ready for TVT[5]. For a fact, when a new identity is discovered without past presence (or similary, *in cached depot*), the target gets higher priority in queue rather then spending bot resources on deep reconnaissance.

Module nodes are equally distributed between operational resources. Core system operations is using PSP[6] to register plausible target profiling. Taking this in concerns, our software has been developed with Unix-like approach ("do one thing and do it well"). Further in this document, you fill find initial scope, an agreement between different zones (or *modules*). The system share knowledge from various protocols and registrants, including operational sources from other govermental agencies (*Valut7, CIA*), and possible exploitable systems.

2.1 Core

At the core-level, `nettis` is a set of a functions for data (*re*)-search and processing. The architectural design is implemented in distributed tone, where several different zone instances are assigned a process. The core then publish the *target*, and subscribed processes execute a worker. `nettis` requires just a world-wide accessible network, or machine with internet connected. A cluster of several `worker` instances greatly increase the operational speed. Since data gathering is fashioned in black-box model, requirement for nodes is at level zero (L0) – e.g. few or no input from operator.

Therefore, we can imagine core as scheduler for intelligence gathering. The subscriber in this case is the entry-point for material confirmation. We can then proceed with corresponding operational research based on data type. Observing closely, we investigated that system of this type have least range of false positives.

[5]Training, validation, tests.
[6]Publish-subscribe pattern.

Figure 1: Level 0, **Source research**

Given at *l0* is zone for *organizational network*; a DNS is taken from ccTLD register (*last known DNS*); core sends an operation for specific link: `www` service process the request. An OCR module reads target from the *whois*. From there both *organizational network* zone, and *personas* are active in their research. The former map the network station and organizational structure (DNS, Hostname, Cluster), while the latter research on homo sapiens target (Clients, Registrants, Employers). This is what we call *l1* or data enumeration.

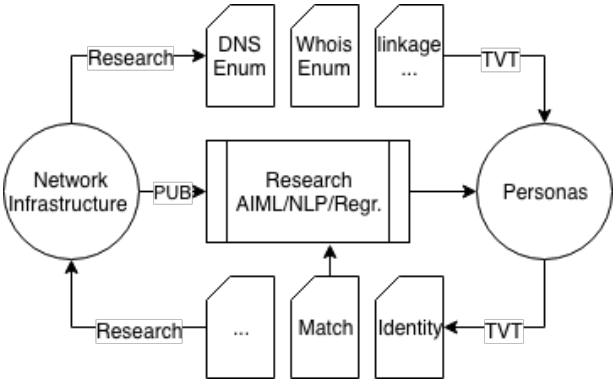

Figure 2: Level 1, **Data enumeration**

2.2 Functionallity

The mass-range scan is defined in operational scope. The purpose of the tool is to gather target data and exploit and exfiliate possible known zone translations. System functionallity is investigated by the core or other services, and as such forwarded to data storage. In a case of machine-learning based algorithms, the data is powered by predefined trained models.

This created a long list of person first and last name, possible DOB (date of birth), registrants and different sources optional but still helpfull for proxitimity vector. The GIGO[7] system requires additional data reconnaissance. This increase or decrease result validity.

In a case of Network Infrastructure, the proximity is defined in power base – the DNS is part of *Host* vector, while host can be both an operational local server, or remote virtual server. For an **Identity** vector, base is different input – e.g. personas last name, friendship end-points or organizational-regional structure.

The point of truth is *computational match* between the two links.

2.3 Source Discovery

As for source discovery, the implementation register same protocol in the range of scan. Information root can be any given match, if same variant is defined (or not defined, *for excludation*). `nettis` use different variants for matching the target. We can take an example below for high-level overview:

```
# possible ccTLD
ZONE_EXTENSION = {
  ".ba"     => 1,  # => Default
  ...
  ".mil.ba" => 5,  # => Military
  ".edu.ba" => 6,  # => Educational
}
```

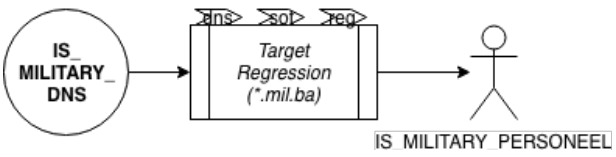

Figure 3: Source Discovery, **Vector Autoregression**

From the same proof, we can rescan with the principle of source data and targeted vector. As seen in Figure 3, the regression is being made from military-operated website, and as such holds a higher level of integrity.

[7]Garbage-in / Garbage-out.

(1)

$$y_t = c + A_1 y_{t-1} + A_2 y_{t-2} + \cdots + A_p y_{t-p} + e_t,$$

Application proceed with research to increase general proof match, and calculate offset given the treshold.

(2)

$$P(a) = \frac{T_a}{K_o}$$

In example (2) the favorable *T(a)* - target, *K(o)* - known combinations. For such system to be managed, we require a proof treshold.

(3)

$$E(p) = INITIAL\_TRESHOLD - P(a)$$

Source discovery is switching target on different zones, the organizational zone is parsing data from either global, ISO, domain-specific or web-wide data. We can use Data_{n+1} for further reference lookup.

> Given the military identity match, push the treshold level further [media-sourced collection (e.g. *news paper*)]. Next-stage layer will examine data using NLP[8] and find possible collaborators, linkage between given targeted identity, and colleagues.

For treshold offset, the constant input and material is used. Some identities require more proof (e.g. *(any) natural person* versus *country president*, since data for the former might not be true). For scan purposes, `nettis` will identify dynamic-based web-applications, store DNS for futher reference, and use it as a *source of truth*.

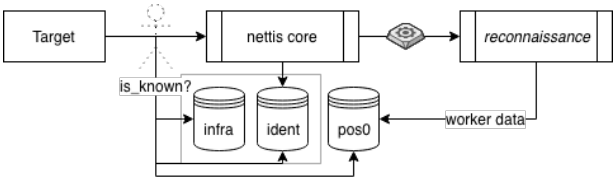

Figure 4: Source Discovery, **Integrity**

[8]Neuro-linguistic programming.

2.4 Modules

2.4.1 Organizational zone

The Organizational zone (*inc*, infrastructure sub-modules) in our application occupy the space for country-wide network scan, including but not limited to: *domains*, *host; interweb-servers*, *intranet*, IoT[9], IoV[10], *telecommunication networks*, *public-domain* & *private-specific* databases, which results in dataset ranging from simple personal computers, to video surveillance cameras and vehicles. `nettis` scan and map these devices in linked abstraction, creating elastic-based, searchable content. This zone is responsible for exfilliation, server discovery, domain mapping, intraweb enum, communication and the list goes on.

2.4.2 Identity zone

Identity (*inc*, personas sub-modules) is responsible for scanning and mapping citizens or residents in the country of Bosnia-Herzegovina. The system is powerful enough to detect invalid matches using techniques presented in Figure 1. To calculate probability of match, NLP service is dynamicly expanded for future bulletproof methods. This module is equiped with various social media based scans, including but not limited to *Facebook*, *Instagram*, *LinkedIn*, *Twitter*. It is also responsible for machine-learning approach on target scan (see `data/define/*` for trained models).

> To identity valuable information, both zone work by registering valid match format, and giving power of each value.

2.4.3 Positive 0 zone

The most complex of `nettis` code is at the **Positive Zero** zone or what we call an "Inner self, Foundation" principle. This piece of code is responsible for connecting various links between above described modules (*Organization, Identity*).

> If **Infrastructure module** got a positive match on telephone or mobile number, it automatically assumes this source is proposed to **Identity module** level. As such, the core system publishes this information to create a match between a *single Identity working for particular Organization*. The linkage between each of the zone give an ultimate point of truth.

To put it simple, the below diagram should present a high-level architecture overview on reconnaissance between modules. At Positive 0, the system combines found links to research target.

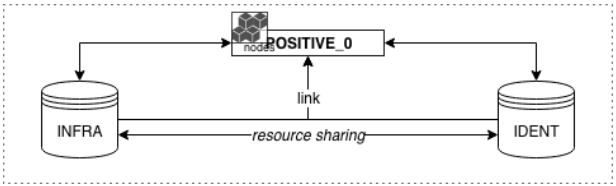

Figure 5: Source Discovery, **Source of truth**

3. Integrity

The integrity of both identity and infrastructure map is mathematically calculated. For some data types, or entities, treshold value is low-based, as to keep positive impact at the valuable collection. For integrity of each incoming data, a question arise – shall the value equal to given model? Who gives the value of each base? To answer these questions, we need to define some constants.

```
# => Base TRESHOLD_LIMIT + n(1)
TRESHOLD_LIMIT      = offset + 1
# => Equals zero to TRESHOLD_LIMIT
BASE_VALUE          = 0
# => Equals to positive identification of
#    treshold integrity
SOURCE_INTEGRITY    = TRESHOLD_LIMIT(n)
# => yet to be defined (since no assumption is made)
TARGET_INTEGRITY    = (KNOWN_DATA || 0)
```

Using above values we can register an integrity at different levels. If we closely examine Source discovery, we can match positive outcome with several sub-matches, as such:

[9]Internet of Things.
[10]Internet of Vehicles.

Organizational network is dompinant to **Identity**, while value of **.gov.ba* (*govermental institution*) holds higher SOURCE_INTEGRITY than **.org.ba* (*organizational institution*). In a similar way, source integrity of a mobile phone number is matched between user provided on sites[11], rather then mapped through third-party tool (e.g. using user-to-phone ratio, see *Instaint*, fig. `High-level architecture overview`).

4. Dataset Versioning

Each and every piece of data (stream) forwarded by either cache[12], or `nettis` stash, holds a versioning information. This is required for calculation of source integrity and target validation. To have a constant input is requirement by `nettis` bot, which creates a basic hash tree (see fig. below), rearranging the position, and giving higher priority to specific targets (those for information is lower than total source integrity, or for old versioned data).

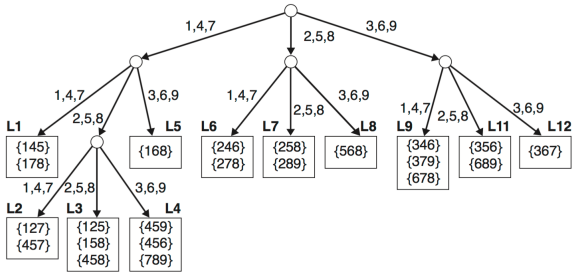

Figure 6: Priority Leveling, **Markle Tree**

5. Data Access

Several *data access points* have been bundled with `nettis`, offering wide-range of use, either internally or by third-party references. Further in this document, you will be presented with different approaches in data access.

5.1 Administration-based

Easiest to use, web-based administration panel, offering (limited) result search. Views available for data stats, and manual data manipulation. Requires no previous knowledge of the system internals. Used by project managers and alike.

5.2 Elastic-based

The most famous and in-depth giving data access point is `ELK` based approach. This system offers a layer of consuming and using dataset from data storage, in a way of combining multiple operative `flags` or `types`. The search query is being run in cluster-like environment where all modules are versioned and offer addition of given flags.

> To combine multiple flags of different source integrity levels, to exfiliate particulare or specific results.

```
# => List all people (identity) from
     a specific organization with
     available mobile numbers.
infra.org.name="*-Security Agency *"
AND ident.has_mobile_number=true

# => List all govermental DNS from
     a specific ccTLD extension and
     personas name
infra.tld.ext="gov.ba"
AND ident.first_name
AND ident.last_name

# => List all military DNS email addresess
infra.tld.ext="mil.ba"
AND ident.email_address

# => Additional flags
data.version=(<yyyymmddhhmmss)
data.integrity=(1-5)
data.integrity.linked=true
...
```

[11] Social-media websites.
[12] Component that store and serve data for faster reuse.

5.3 Native-based

This is technical-level search on database engine. In case of `nettis`, several storage abstractions have been developed, including *SQLite3*, *MariaDB* and *Hadoop* database. For reference on each of the native based search, reffer to *Technical Documentation on nettis storage abstraction*.

5.4 Notifications

Notification system is used to notify end-user, managers, master, operation, or administrators with new data versions and integrity calculations. Once a match has been discovered, a notifications is sent on various provided messaging systems (*Email*, *Push notifications*, *Telegram* etc.).

5.5 API

`nettis API` is submodule for data access; allowing technical heads to manipulate with dataset and integrate their software with this tool. Application Programming Interface is accessible through HTTP/REST protocol, although live stream may be examined through *Websockets*. For reference on API, reffer to *Techincal Documentation on nettis API*.

Acknowledgements

This paragraph is dedicated to original `nettis` authors, contributors, their families and friends, and everyone who helped creating and maintaining this complex piece of software.

References