Guillaume PRIGENT - Founder/CTO diateam

<guillaume.prigent@diateam.net>

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License.

HITBSECCONF2011 - May 17-20 - Amsterdam
Plan

1. Introduction
2. OSINT process
3. Netglub
4. Perspectives
Disclaimer*

*a big fat one, because everybody loves fineprint

- Information is provided as-is, though every effort has been made to ensure the accuracy of the information presented. Author of the presentation is not legally liable under any circumstances for any damages such as but not limited to (including direct, indirect, incidental, special, consequential, exemplary or punitive damages) resulting from the use or application of the presented information.

- Unless explicitly noted in forms such as but not limited to "the XYZ Company says", etc., the opinions expressed in this presentation are solely and entirely my own. They should not be interpreted as representing the positions of any organization (past, present, future, existent, non-existent, public, private, or otherwise) with which I may or may not have been, are or are not, or will or will not be affiliated at some time in the past, present, or future.

- All trademarks and registered names are the property of their respective owners. All the effort has been made to link to the original material used as exhibition items in the presentation, and those items are property of their respective owners.

- This presentation is ©2010, Guillaume Prigent <guillaume.prigent@diateam.net>. Released under:

  Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License.

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License.
Context
Or "What are we talking about?"

Definition
OSINT is a form of intelligence collection management that involves finding, selecting, and acquiring information from publicly available sources and analyzing it to produce actionable intelligence... (From Wikipedia)

Consideration
- Today, OSIF = digital information
- Quantity versus quality dilemma
- Independent of information’s "color"

Not to be confused!
Open Source Information != Open Source Software
Domain mapping
Or "How to link the real world and the digital world?"

<table>
<thead>
<tr>
<th>Real world</th>
<th>Digital world</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emails</td>
<td>IP, hosts, netblocks, AS</td>
</tr>
<tr>
<td>Persons</td>
<td>Whois records / rWhois</td>
</tr>
<tr>
<td>Phone numbers</td>
<td>Forward and reverse DNS</td>
</tr>
<tr>
<td>Addresses</td>
<td>Google fu / Deep web</td>
</tr>
<tr>
<td>Documents</td>
<td>Document metadata</td>
</tr>
<tr>
<td>Patents / Projects</td>
<td>Twitter, Facebook, ...</td>
</tr>
<tr>
<td>Sentences / Words ...</td>
<td>XFN, vCards, hCards</td>
</tr>
<tr>
<td>Habits / Hobbies</td>
<td>Face detection, ...</td>
</tr>
<tr>
<td>Social affinities</td>
<td>ip2geo, Google Earth, ...</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>
What for ?

- Analyze « social networks » to try and predict the potential of a commercial product ;
- Detect « key people » in a group or social network ;
- Use face detection to cross-reference profiling information ;
- Use social networks to obtain economic intelligence ;
- Aggregate fiscal, administration, patent and shareholding information for a « Tender offer » ;
- Perceive the technological environment of a product and identify interdependencies ;
- Perform a technical and organizational reconnaissance before a penetration test ;
- ...
Methodology 1/2
Or "Open Source Intelligence cycle management"
Methodology 2/2
Or "Open Source Intelligence cycle management"

1. Identify the need (the question, on what and/or who);
2. Identify the potential Open Sources for information collection;
3. Acquire or extract the information elements in their context;
4. Analyze and normalize the actually useful information;
5. Visualize the produced dependencies and better understand their structure;
6. Prune, and optionally qualify and enrich the results;
7. Repeat again...
Open Source selection
...in most cases visible Web isn’t enough...

*Forgive me father, for I have sinned

*I know

*Picture courtesy of P.Chappatte - Copyright P.Chappatte 2011*
Open Source selection
...needles in a haystack...

Some sources

- DNS, Whois, http://dnshistory.org, ...
- ...
- © French DoD suppliers
  http://www.achats.defense.gouv.fr/Annuaire-des-fournisseurs,13786
- © French DoD contracts
  https://www.achats.defense.gouv.fr/-Liste-des-marches-conclus,57343-
Transformation 1/5
First chain

Transformation
- Domain to DNS Names: MX/NS/Zone transfer/Bruteforce
- DNS Names to IP Address: Resolve 😊
- IP Address to Netblock: Whois
- Netblock to AS: Code routers, Web
Transformation 2/5
Forward chain

- Domain to DNS Names: MX/NS/Zone transfer/Bruteforce
- DNS Names to IP Address: Resolve 😊
- IP Address to Netblock: Whois
- Netblock to AS: Code routers, Web
Transformation 3/5
Six more transforms...via Whois
Transformation 4/5
Six more transforms...using Search Engines, PGP servers, ...

Website
  
Domains -> DNS Names -> IP Address -> Netblock -> AS
  
Email Address
Person
Phone Number
Transformation 5/5
And more...

Phrase

Domain

DNS Names

Website

IP Address

Netblock

AS

Location

File Document

Email Address

Person

Phone Number

Social Network Profile
"...if you have been living under a rock"

- Visual tool (graph) for dealing with information
- Entities and relationships
- Platform for information integration & correlation
- [http://www.paterva.com](http://www.paterva.com)

Open Source?

- Closed source
- Application Service Provider / Software as a Service
- Privacy ? / Anonymity ?
- ...

@ROELOF: Вес мир СОРТИР, а люди в нём ЗАСРАНЦЫ
Admiral William Studeman, 1992, USA

The plan establishes the goal of creating an integrated community open source architecture. The new architecture must provide, among other things:

- flexible collection,
- networked access to external data bases,
- immediate user and customer feedback, and
- automated, profiled delivery of collected open source information...

...will be an Open Source Information Exchange comprising a central switch and digital communications networks which interconnect...
Development process
...the uggly part...

Netglub lifecycle

- analyze what exists & "state of the art";
- identifying COTS & APIs;
- focus on Open Source;
- identify the technological barriers;
- define a flexible, scalable architecture;
- implement various Proof of Concept (PoC);
- test & compare our results (benchmark);
- reiterate & (one day) complete the tool/framework.
General architecture
Not quite a botnet...
Netglub’s components
Slave(s), Master & GUI

<table>
<thead>
<tr>
<th>Slave’s main features</th>
<th>Master’s main features</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Autonomous network service (daemon)</td>
<td>- Autonomous network service (daemon)</td>
</tr>
<tr>
<td>- Login to &quot;Master&quot; / announcement</td>
<td>- Command &amp; Control federation of slaves</td>
</tr>
<tr>
<td>- &quot;Job&quot; runner &amp; scheduler for &quot;transforms&quot;</td>
<td>- Authentications &amp; Permissions for &quot;Slaves&quot; &amp; Clients (GUI,CLI)</td>
</tr>
<tr>
<td></td>
<td>- &quot;transforms&quot; &amp; &quot;entities&quot; database</td>
</tr>
<tr>
<td></td>
<td>- API XML-RPC for clients</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GUI’s main features</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>- Login to &quot;Master&quot; (XML-RPC)</td>
<td>- ...</td>
</tr>
<tr>
<td>- Relationships visualisation</td>
<td>- ...</td>
</tr>
<tr>
<td></td>
<td>- ...</td>
</tr>
</tbody>
</table>

Netglub : Really Open Source Information Gathering
Step 1/2: Netglub

- User features
- More internal
Work in progress

- Sources selection & Ontologies
- Personnal OSIF / Building datastore
- Qualification (community detection, confidence, time attributes, ...)
- Reusability, Scalability
- Anonymity
- Huge graph layout and real-time interaction
- Maintainability, Distribution
- Machine learning
- ...

---

guillaume.prigent@diateam.net - 2011/05/19
Glubby
...as a component for Netglub NG...

Motivations / Needs

- Real time "force based" graph layout
- Fast render in OpenGL for 3D & 2D
- GPU and/or CPU based
- Library & Open Source, ...

State of the art

- GraphViz (http://www.graphviz.org/)
- Gephi (http://gephi.org/)
- Tulip (http://tulip.labri.fr/)
- Jung (http://jung.sourceforge.net/)
- NetworkX (http://http://networkx.lanl.gov/)
- Igraph (http://igraph.sourceforge.net/)
- UbiGraph (http://ubietylab.net/ubigraph/)
- ...

...as a component for Netglub NG...
What’s GPU?
Or "The return of the vector machine"

Pros
- Fast
- Cheap
- Low-power
- Future

Cons
- Specialized
- Hard to program
- Bandwidth problems
- Rapidly changing
Limitations of GPUs  
...the dark side of the force...

If the GPU is so great, why are we still using the CPU?

You can’t simply "port" existing code and algorithms!

- Data-stream mindset required
- Not suitable to all problems
  - Pointer chasing impossible or inefficient
  - Recursion
- Debugging is hard
  - Hardware is designed without debug bus
  - Driver is closed
- Bottlenecks
- Standard API?
Network representation as graph
Graph-theoretic data structures

*Vertices 5
*Edges
0 1
0 2
0 3
2 3
2 4
Edge list*
*Sometimes called incidence list

Connections
0 1 2 3 ... 0 0 0
-1 0 3 -1 -1
-1 0 -1 2 0
0 0 -1 0 1

Adjacency list
0 → 1, 2, 3
1 → 0
2 → 0, 3, 4
3 → 0, 2
4 → 2

Adjacency matrix
0 1 2 3 4
0 1 1 1 1 0
1 1 0 0 0 0
2 1 0 0 1 1
3 1 0 1 0 0
4 0 0 1 0 0

Laplacian matrix*
*Sometimes called admittance matrix
or Kirchhoff matrix
0 1 2 3 4
0 3 -1 -1 -1 0
1 -1 1 0 0 0
2 -1 0 3 -1 -1
3 -1 0 -1 2 0
4 0 0 -1 0 1

*Sometimes called admittance matrix
or Kirchhoff matrix
Glubby internal graph representation

GPU/CPU structure

- float * A
- float * B
- float * Size
- int * Locked
- int nbnode
- int dimension
- int k
- float maxDisplace
- float gravity
- float speed
- int noCollision
Glubby
Our implementation on CPU/GPU

**Fruchterman-Reingold on GPU&CPU**


http://citeseer.ist.psu.edu/viewdoc/download;?doi=10.1.1.13.8444&rep=rep1&type=pdf

**Barnes-Hut on GPU&CPU**


http://www.nature.com/nature/journal/v324/n6096/abs/324446a0.html


**Burtsher on GPU**


http://www.gpucalculating.net/?q=node/1314
Demonstration

Step 2/2 : Glubby

- UbiGraph (http://ubietylab.net/ubigraph/)
- Glubby : Fruchterman-Reingold on GPU&CPU
- Glubby : Barnes-Hut on CPU
- Glubby : Barnes-Hut + Burtsher on GPU

Disclaimer

It’s still rough around the edges!
No user’s features (or just few for testing) for the moment, that’s not the purpose, it’ll be a library and/or a component for Netglub 2.0 branch (or Netglub NG)!
But it’s awesome... ...or not 😊
Netglub : Really Open Source Information Gathering

http://www.netglub.org

Thanks for your attention.
Any questions? (one at a time & slowly please 😊)

<guillaume.prigent@diateam.net>

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License.