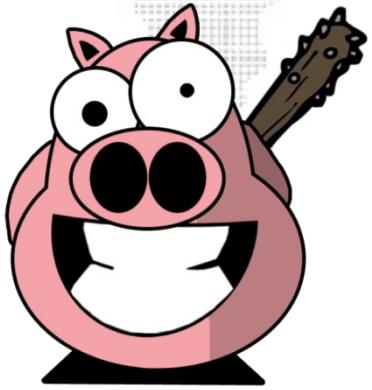
HIGH-TECH BRIDGE®

INFORMATION SECURITY SOLUTIONS

P.I.G.

Passive Information Gathering



26 juillet 2011

Xavier ROUSSEL

Introduction

- The aim of this paper is to present the P.I.G. software, a private tool developed by High-Tech Bridge to optimize the information gathering phase during penetration tests.
- This paper only contains few technical information in order to provide a global view of the software implementation, which may be useful to people willing to automate such a process.

SUMMARY

HIGH-TECH BRIDGE®
INFORMATION SECURITY SOLUTIONS

I. What is P.I.G.?

II. Principle of P.I.G.

1. Architecture
2. Event dispatcher
3. Projects
4. Informations
5. Transforms

III. IHM

IV. Examples

I. What is P.I.G. ?

HIGH-TECH BRIDGE®
INFORMATION SECURITY SOLUTIONS

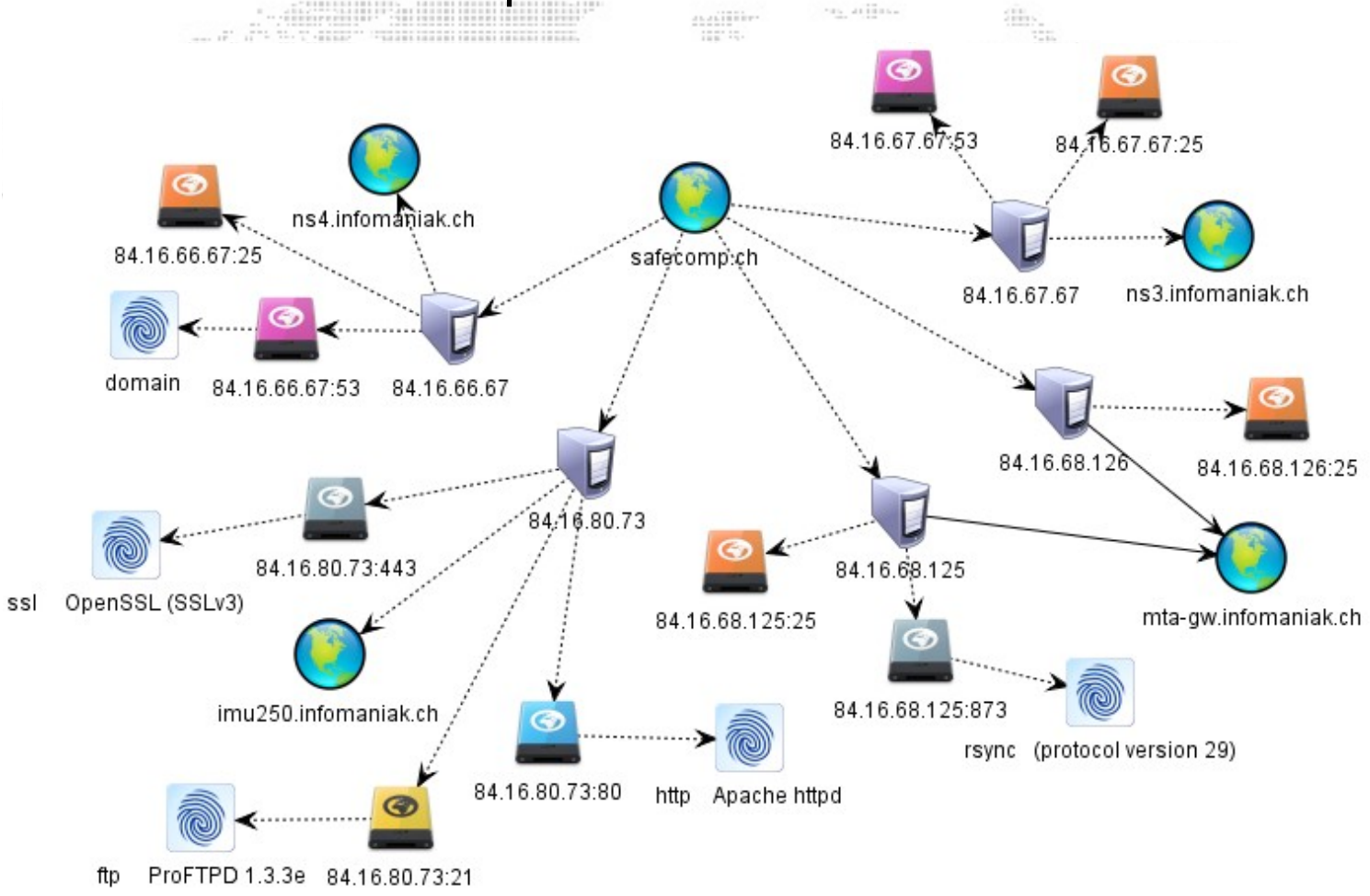
- The PIG tool is a project launched by High-Tech Bridge to optimize the information gathering phase during penetration tests.
- It is developed to be used in a black box penetration test, and therefore without any prior knowledge of the target.
- PIG is an acronym for “Passive Information Gathering”. Indeed, it was the initial goal of the tool, which now also offers some actives functions, such as TCP port scan and dictionary attacks.

II. Principle of P.I.G.

- Inspired by a well know software in IT security : Maltego, developed by Parteva.
- The aim is to deduce information from other information, by using simple or complexes algorithms called “Transforms”.
- Transforms don’t take any parameter except the object on which they apply.
- This constraint significantly reduces the users’ duties, as they only have to select the base object and the desired algorithm to discover new information.

HIGH-TECH BRIDGE
INFORMATION SECURITY SOLUTIONS

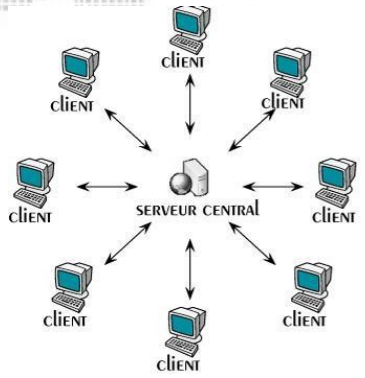
- Example of a quick search by seeking servers related to the domain safecomp.ch:



II.1. Architecture

HIGH-TECH BRIDGE®
INFORMATION SECURITY SOLUTIONS

- Based on a client / server architecture.
- All data are stored on server side.
- Allow simultaneous connections from multiple clients on a single project.
- Keeps clients synchronized with the current state of project information's through an event dispatcher.

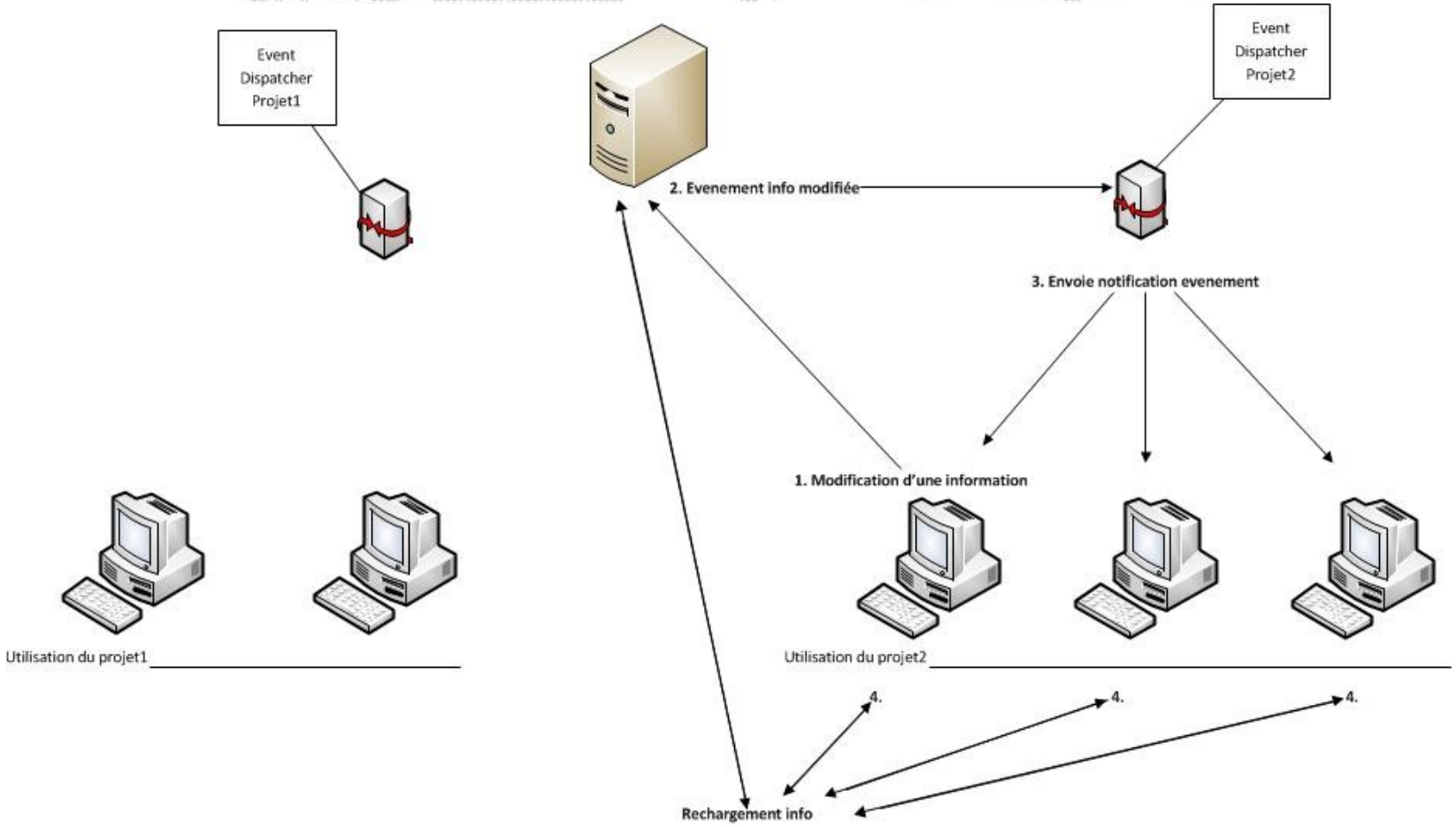


II.1. Architecture

- Objects' implementation is always done on the server side, clients only have pointers to these objects in order to interact (JAVA RMI principle).
- Client and server binaries are developed in JAVA and interact together through the JAVA RMI protocol.
- Auditors can dynamically add new server side modules to add new transforms or objects in any project.
- Management of users accounts.

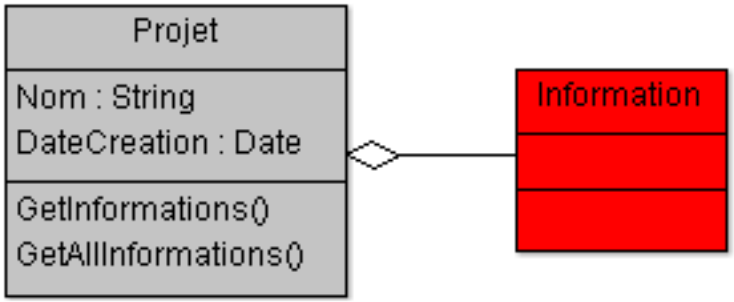
II.2. Event dispatcher

- Notification to clients when information is added, removed or updated in opened project.



II.3. Projects

- Information are organized into projects.
- Projects consist of basic information added by user and all information that have been deduced from transforms.
- Projects are added by PIG's users and can be edited or deleted at any time.



- Informations are Objects that represent a certain type of information:

RangeIp: Such type of information is used to represent an IP range defined by its network address and submask (e.g. 10.0.0.0/24).

Domain: Such type of information is used to represent a domain name (e.g. www.google.fr).

II.4. Informations

Server: Such type of information is used to represent a server by its IP address (e.g. 10.0.0.1).

Service: Such type of information is used to represent a service on a server and is divided into several sub-types for the main services (e.g. FTP, SSH or HTTP).

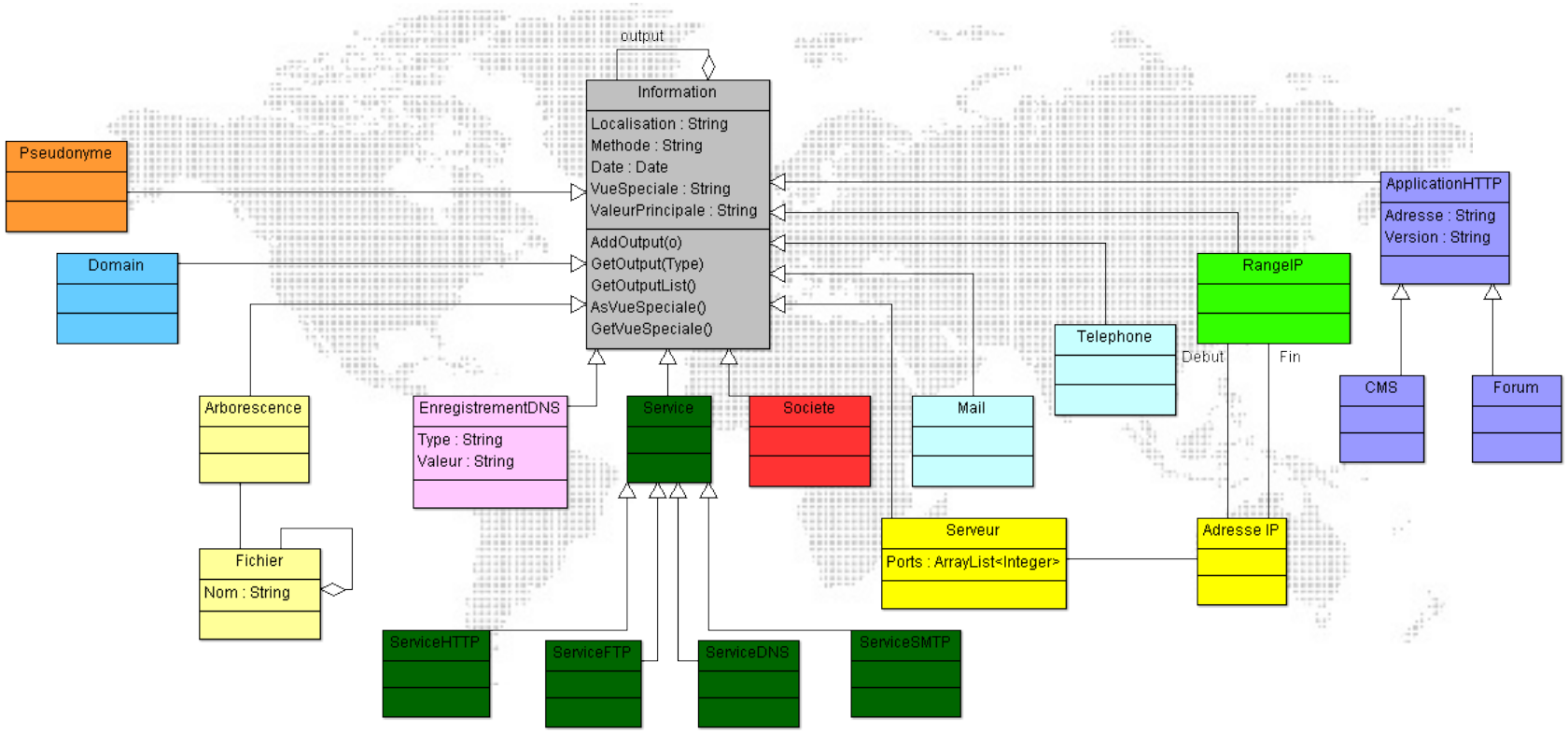
Access: Such information is used to represent a couple user / password when an access to a server is discovered.

II.4. Informations

FingerPrint: Such information is used to represent the results of a FingerPrinting transforms.

- There are many other types of information used to represent phone numbers, mail address, pseudonyms, companies, etc.

II.4. Informations – Class Diagram

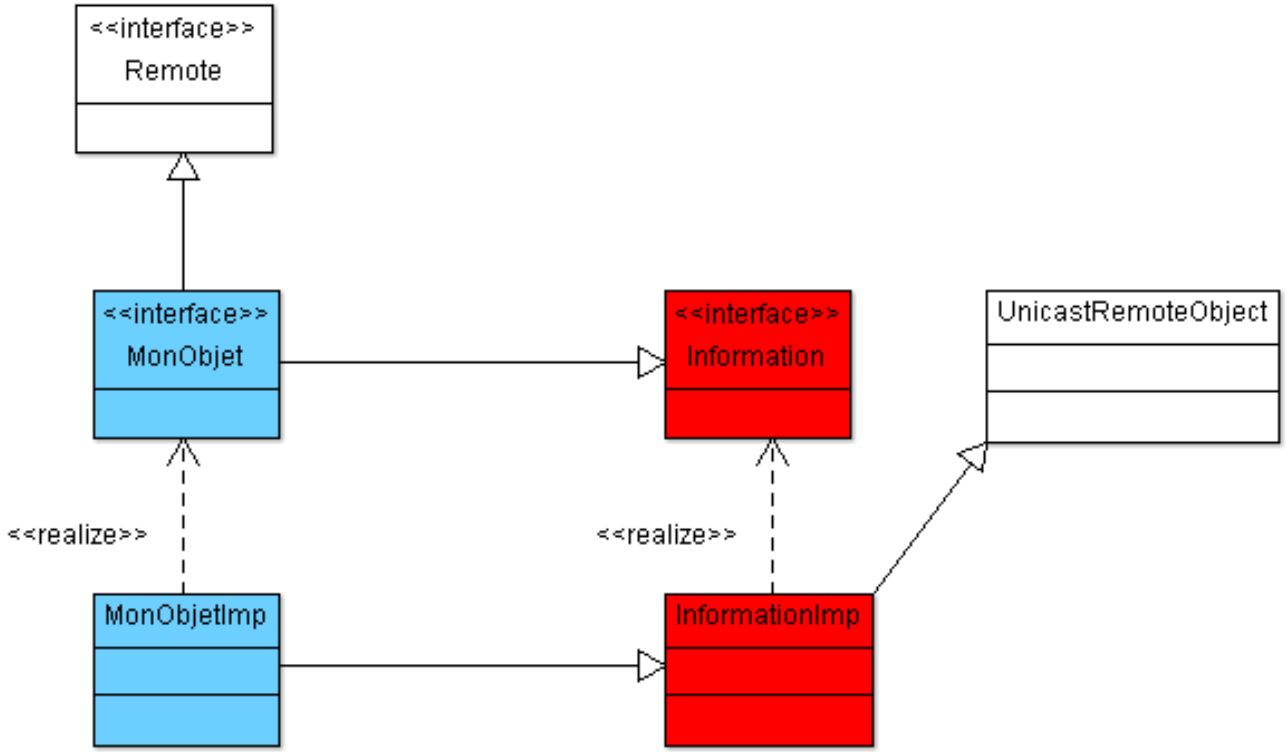


II.4. Informations – RMI Implementation

HIGH-TECH BRIDGE®
INFORMATION SECURITY SOLUTIONS

- RMI have an important role.
- Allow to manipulate remote objects as references.
- Prevent data serialization and reference loss.
- Significantly simplify client / server communication.
- Require additional safety rules to secure objects.

- Modification of class diagram to use RMI.



II.4. Informations – RMI Object Example

HIGH-TECH BRIDGE®
INFORMATION SECURITY SOLUTIONS

```
public interface Domain extends Information {

}

public class DomainImp extends InformationImp implements
    Domain,BaseInformation {
public transient static final String icon = "Network-PC-icon.png" ;

public DomainImp() throws RemoteException {
    super() ;
}

public DomainImp(String _v) throws RemoteException {
    super(_v) ;
}
```

II.4. Informations – RMI Object Example


```
public DomainImp(InformationImp _d) throws RemoteException {
    super(_d) ;
}

@Override
public InformationProperties getProperties() {
    return new InformationProperties(DomainImp.class.getName(),
            Domain.class.getSimpleName(),
            "Classe représentant un domaine ou un domaine.",
            "xxxx.xxxx.xxx") ;
    }
}
```

II.5. Transforms

- Algorithms located on server side that allow to find new information by sending another information as input.
- Can be based on existing tools such as Nmap, used to scan IP ranges and ports, or Hydra, used to perform dictionary attacks.
- Or implement a completely new algorithm for information retrieval (e.g. information gathering through Web Search Engines).

II.5. Transforms levels

HIGH-TECH BRIDGE®
INFORMATION SECURITY SOLUTIONS

- Transforms are categorized into 5 levels.
- Level 1 includes all the passives transforms which never contact the server.
- Level 2 includes all transforms that open a connection on the server which could sometimes be seen as a potential attack.
- Level 3 includes transforms that could clearly be seen as attacks on remote systems.
- Level 4 includes transforms which try to find an access on the server.
- Level 5 includes transforms that can potentially be dangerous for the server if misused.

II.5. Transforms examples

HIGH-TECH BRIDGE®
INFORMATION SECURITY SOLUTIONS

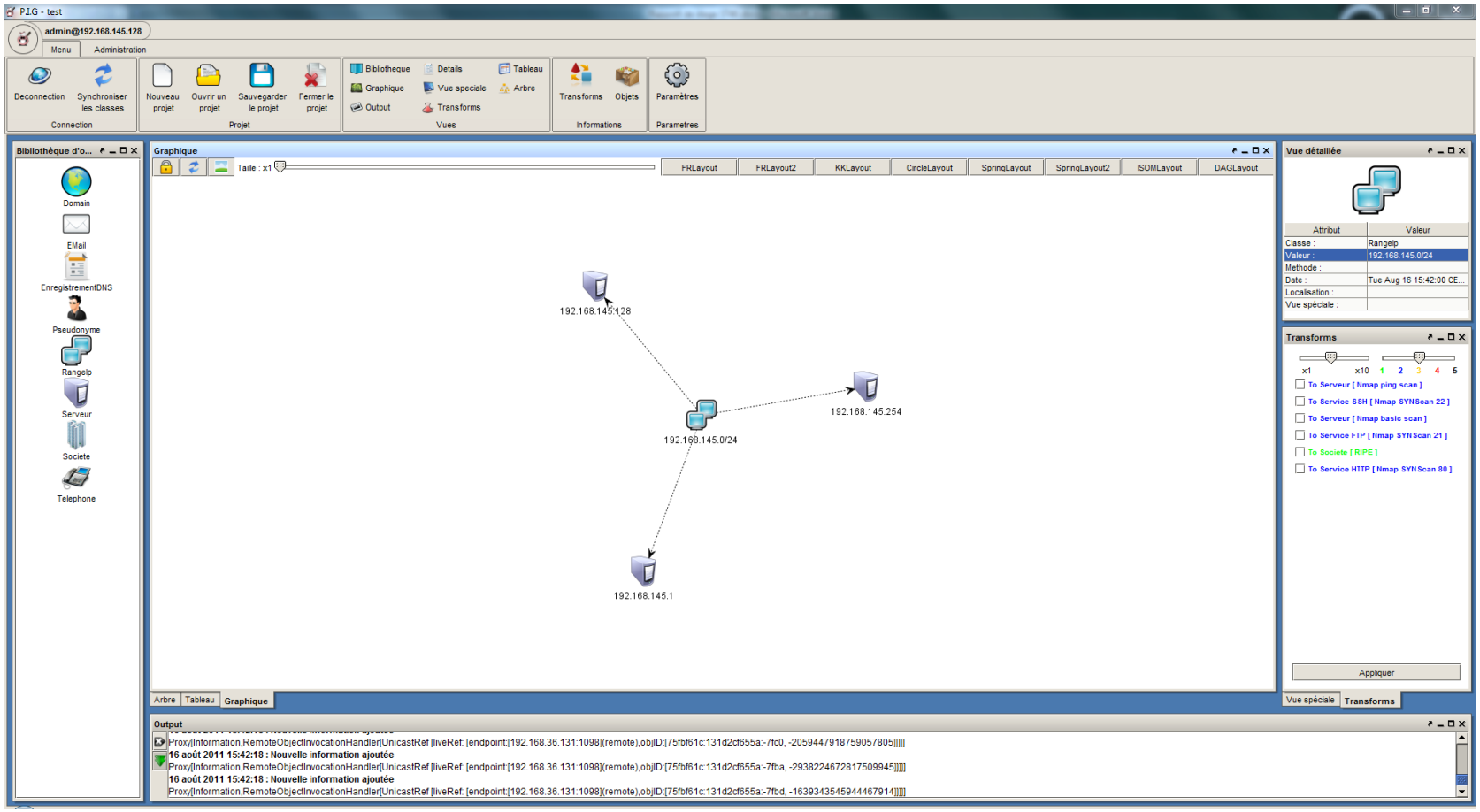
- **Domain to Server Using nslookup** : Search for the IP address or addresses associated with a domain name through NSLOOKUP.
- **NetBlock to Server Using Nmap PingScan** : Search all hosts on an IP range by performing a ping scan.
- **NetBlock to ServiceFTP Using Nmap RangeScan** : Search all FTP services on an ip range by performing a SYN scan.

II.5. Transforms examples

HIGH-TECH BRIDGE®
INFORMATION SECURITY SOLUTIONS

- **NetBlock to Societe Using RIPE** : Search for the name of the company registered on RIPE database though part of the IP range.
- **Server To NetBlock Using RIPE** : Search to which IP netblock a specific server belongs to through the RIPE database.
- **ServiceFTP To Acces Using Hydra DictionnaryAttack** : This transform will attempt to get an FTP access on a remote server through dictionary attacks.

III. IHM – Global view



III. IHM – Global view

HIGH-TECH BRIDGE®
INFORMATION SECURITY SOLUTIONS

- 8 main windows.
- The library window represents all the objects available on the server with the ability to add them on the graph through drag and drop.
- The graph window displays in real-time project's information.
- The table and the tree view are two static views that can be refreshed by the user.
- The output window displays the different actions performed in the project.

III. IHM – Global view

HIGH-TECH BRIDGE®
INFORMATION SECURITY SOLUTIONS

- The detailed view window allows the user to get more information on the selected object.
- The list of available transforms allows the user to run new server side transformations on selected objects.
- The special view window is specific to each objects and allows to get more information than through the detailed view.

III. IHM – Global view

HIGH-TECH BRIDGE®
INFORMATION SECURITY SOLUTIONS

- Provides dynamic management of windows position provided by the component Docking Windows from InfoNode.
- The main menu is based on the JAVA component Flamingo.
- 3 types of views are implemented: 1 dynamic view (synchronized with the server) and 2 static views (refreshed by the user).

III. IHM – Graph

HIGH-TECH BRIDGE®
INFORMATION SECURITY SOLUTIONS

- Based on JUNG (JAVA component).
- Synchronized with the server.
- Get Events from the event dispatcher and update information on client side.
- Allow the user to add and delete information in a given project.
- Ability to apply different layouts on the graph to modify nodes disposition.

III. IHM – Tree

HIGH-TECH BRIDGE®
INFORMATION SECURITY SOLUTIONS

- Based on JUNG (JAVA component).
- Updated by the user.
- Show information in a tree form.

III. IHM – Table

HIGH-TECH BRIDGE
INFORMATION SECURITY SOLUTIONS

- Static view refreshed by the user.
- Ideal for a quick overview of all information found.
- Ability to apply filters on results.

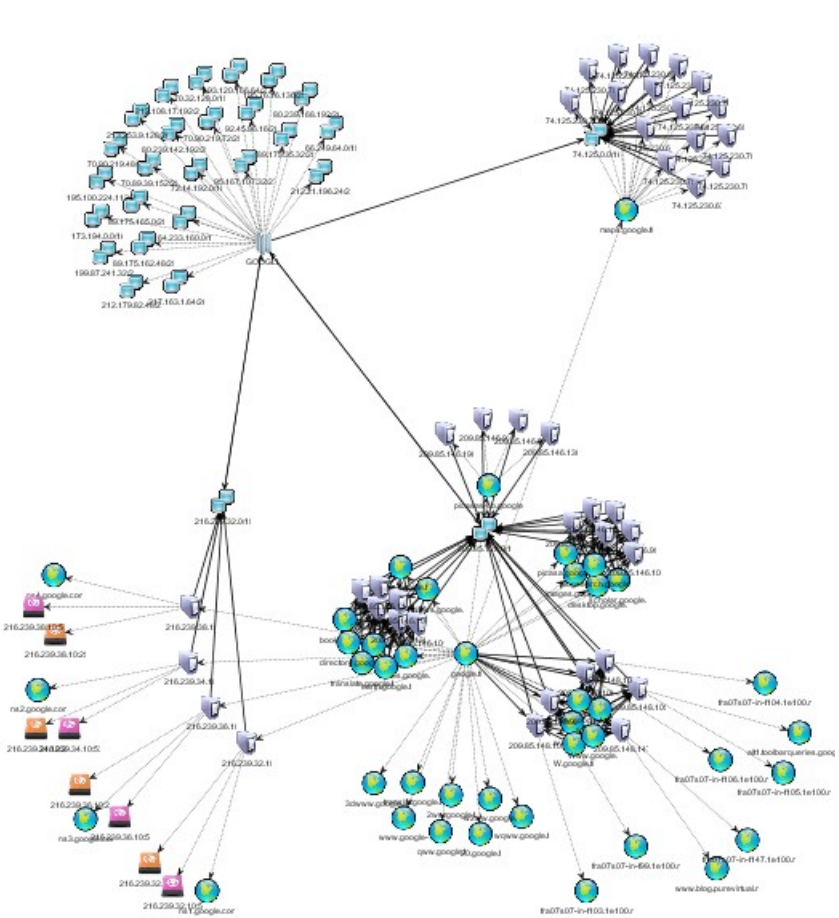
Class ▲	Valeur principale	Date de création	Methode	Localisation	Vue spéciale
DomainImp	htbridge.ch	22 juil. 2011			
DomainImp	fr.htbridge.ch	22 juil. 2011	FromDomainToSubDomainUsingBing		
DomainImp	openbsd.li	22 juil. 2011	FromDomainToSubDomainUsingBing		
DomainImp	www.htbridge.ch	22 juil. 2011	FromDomainToSubDomainUsingBing		
DomainImp	your.privatedns.com	22 juil. 2011	FromServerToDomainUsingReverseLook...		
DomainImp	my.privatedns.com	22 juil. 2011	FromServerToDomainUsingReverseLook...		
ServeurImp	67.205.124.44	22 juil. 2011	FromDomainToServerUsingNsLookup		
ServeurImp	67.205.116.5	22 juil. 2011	FromDomainToDNSServeurUsingNsLook...		
ServeurImp	174.142.253.5	22 juil. 2011	FromDomainToDNSServeurUsingNsLook...		
ServiceDNSImp	67.205.116.5:53	22 juil. 2011			
ServiceDNSImp	174.142.253.5:53	22 juil. 2011			
ServiceSMTPImp	67.205.124.44:25	22 juil. 2011			

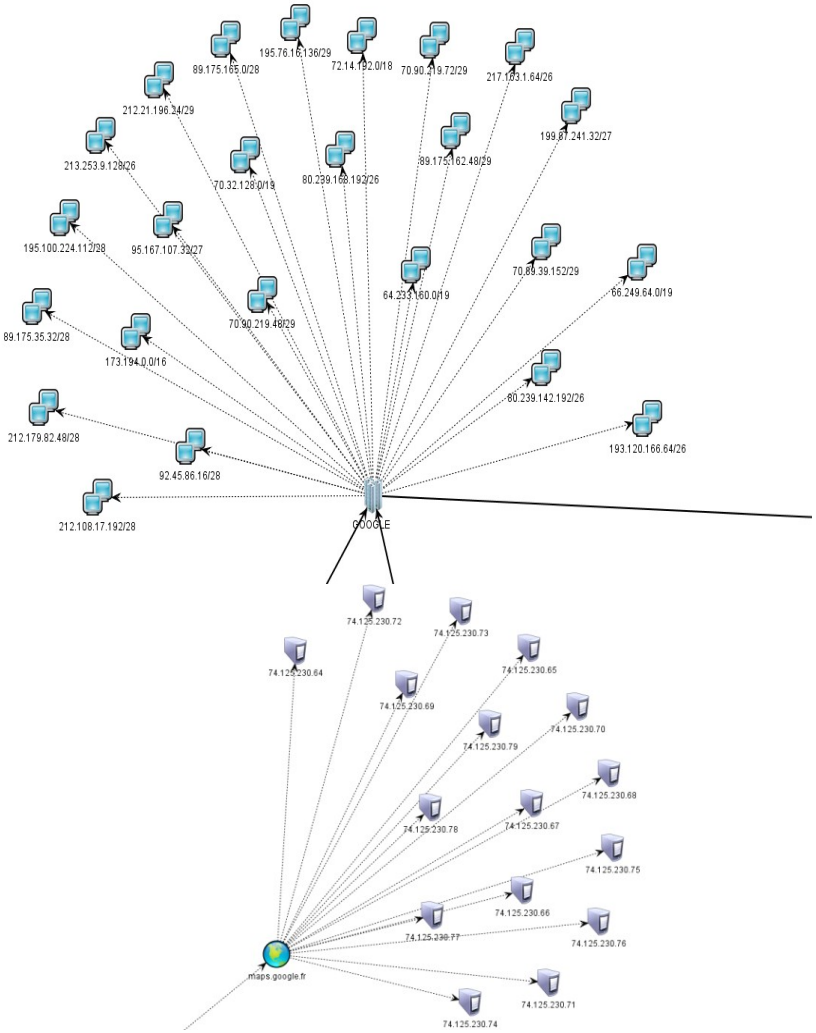
IV. Example – google.fr

HIGH-TECH BRIDGE®
INFORMATION SECURITY SOLUTIONS

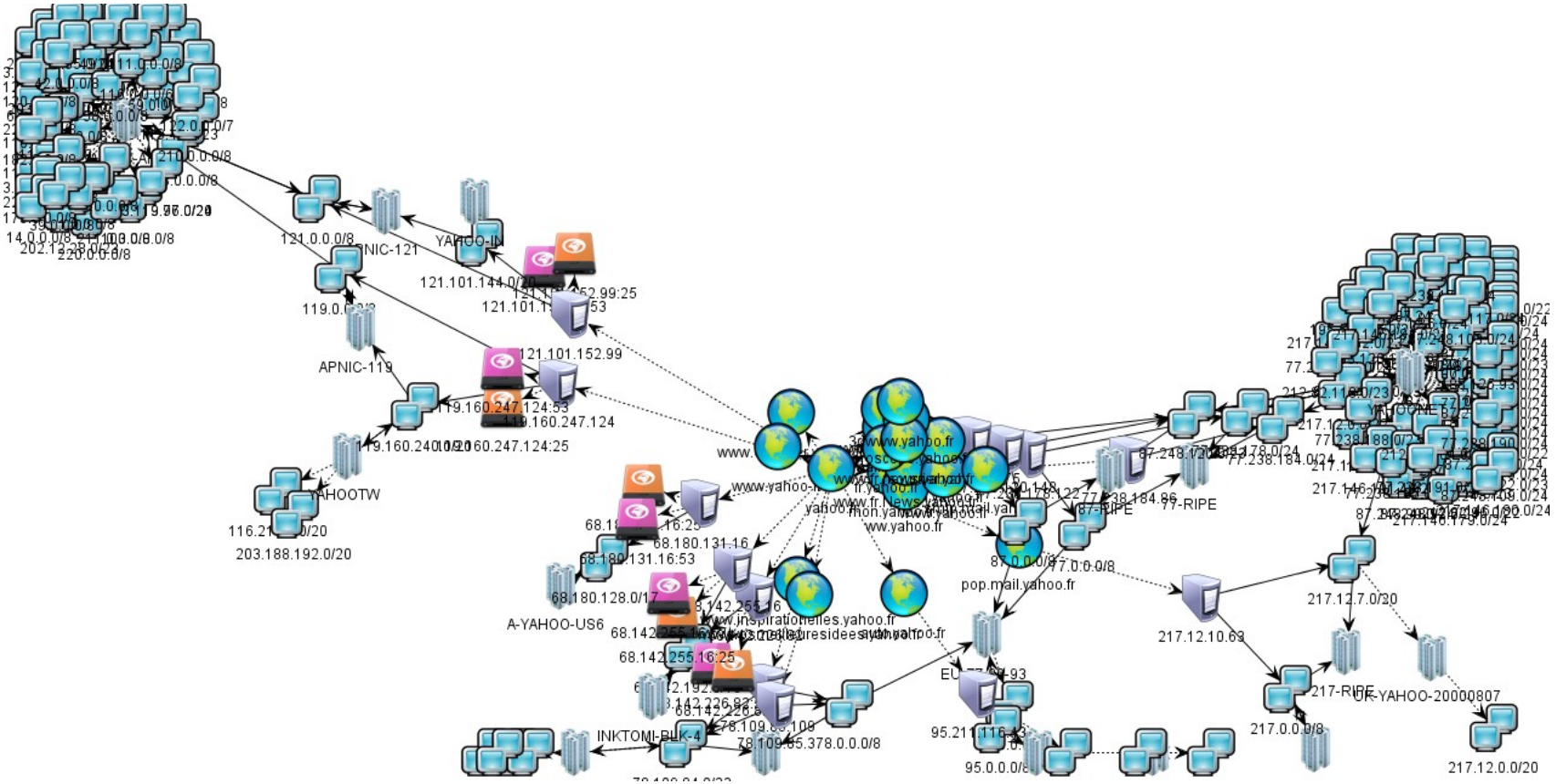
- Information that can be discovered on Google France in less than 20 seconds with only the google.fr domain as input:
  - 27 IP ranges
  - 41 domain names
  - 42 servers

IV. Example – google.fr





IV. Example – yahoo.fr



References

- http://www.java.com
- http://www.paterva.com
- http://www.infonode.net
- http://jung.sourceforge.net
- http://java.net/projects/flamingo