Malware Clustering using impfuzzy and Network Analysis - impfuzzy for Neo4j -

Hi again, this is Shusei Tomonaga from the Analysis Center.

This entry introduces a malware clustering tool “impfuzzy for Neo4j” developed by JPCERT/CC.

Overview of impfuzzy for Neo4j

impfuzzy for Neo4j is a tool to visualise results of malware clustering using a graph database, Neo4j. A graph database is a database for handling data structure comprised of records (nodes) and relations among the records. Neo4j provides functions to visualise registered nodes and relations in a graph.

impfuzzy for Neo4j operates in the following sequence:

1. Calculate the similarity of malware using impfuzzy
2. Generate a graph (network) based on the similarity
3. Conduct network analysis over the graph (clustering)
4. Register and visualise the clustering results on Neo4j database

First, the tool calculates the similarity of malware using impfuzzy; the techniques to estimate the similarity of Windows executables based on a hash value generated from Import API. impfuzzy was introduced in our blog article before, so please take a look for further details.

After that, a graph is generated by connecting nodes that were judged to be similar based on the impfuzzy results. The graph is then analysed using Louvain method [1]. This is one of the methods to cluster network graphs, which outperforms other algorithms in speed. With this analysis, malware is automatically classified into groups.

Finally, the information of analysed malware and its group is registered in Neo4j database.

Figure 1 describes the clustering result of Emdivi malware using impfuzzy for Neo4j.

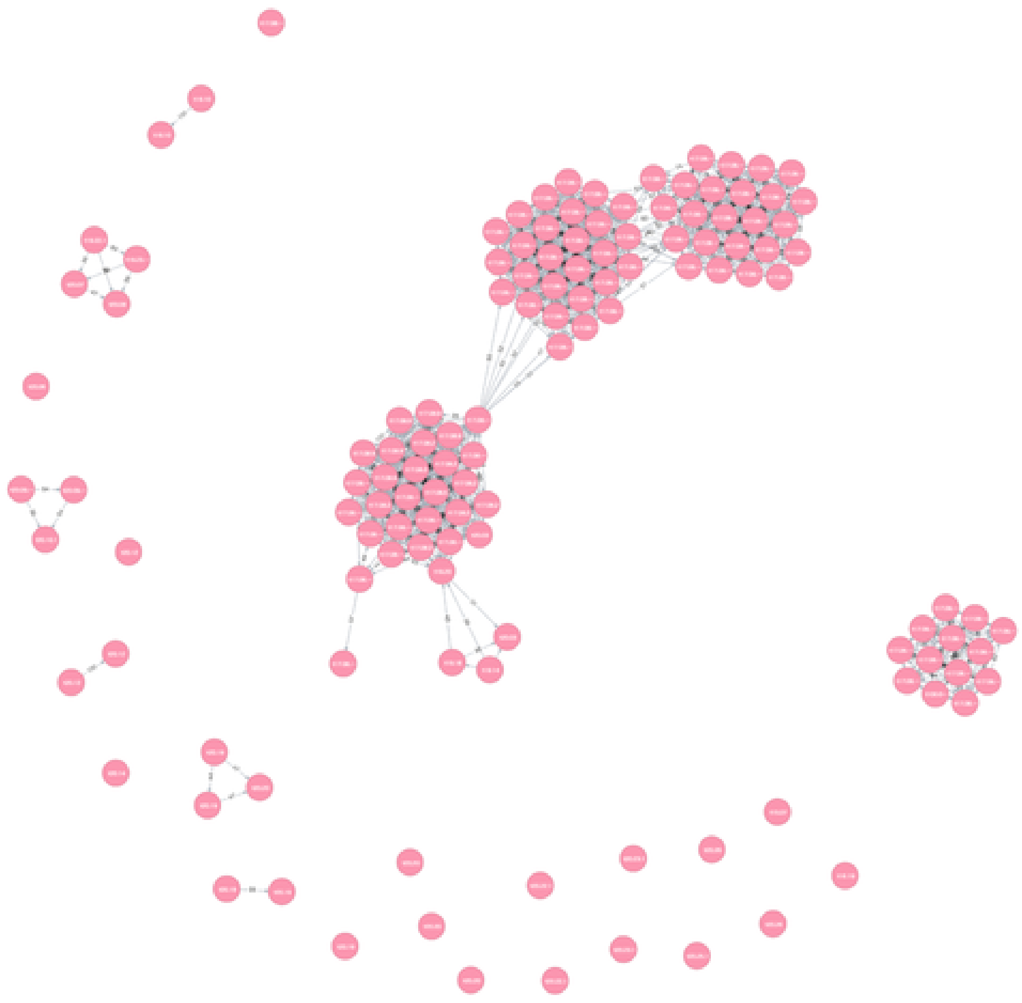


Figure 1: Clustering result of Emdivi by impfuzzy for Neo4j

In this graph, types of malware (pink nodes) that are judged to be similar are connected with lines. From the above visualisation, it is clear that there are several groups of their variants with high similarity.

Since impfuzzy for Neo4j automatically clusters related samples through network analysis, it is possible to extract samples that belong to a specific group. Figure 2 visualises the relationship of a specific group from the example in Figure 1. The numbers on the grey lines (grey edges) between samples indicate the similarity of the malware in the range from 0 to 100 (the higher the number is, the more similar the samples are).

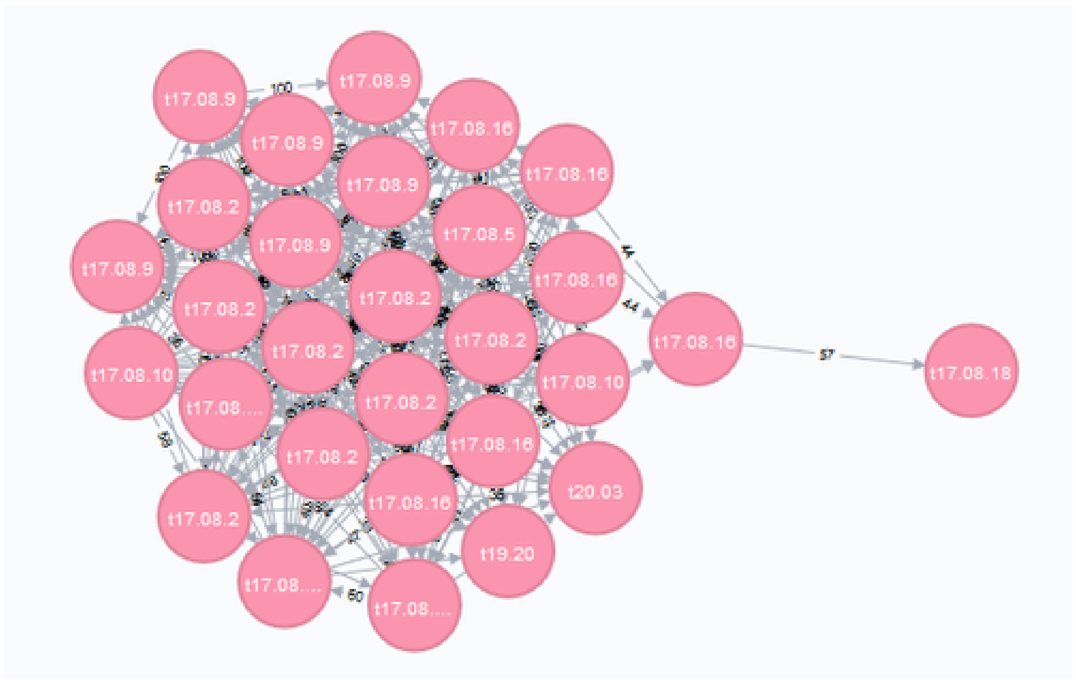


Figure 2: Visualisation results of samples belonging to a specific group

How to obtain and use impfuzzy for Neo4j

The tool is available on GitHub. Please refer to the following webpage:

JPCERTCC/aa-tools GitHub - impfuzzy for Neo4j
https://github.com/JPCERTCC/aa-tools/tree/master/impfuzzy/impfuzzy_for_Neo4j

Here are the instructions for using impfuzzy for Neo4j.

1. Obtain and install Neo4j community edition

Download Neo4j community edition from the following webpage and install it:
https://neo4j.com/download/

2. Download impfuzzy_for_neo4j.py

From the following webpage:
https://github.com/JPCERTCC/aa-tools/tree/master/impfuzzy/impfuzzy_for_Neo4j

3. Install the software required for executing impfuzzy_for_neo4j.py

- Install Python module pyimpfuzzy

```
$ pip install pyimpfuzzy
```

For more information on the install procedures, please see:
https://github.com/JPCERTCC/aa-tools/tree/master/impfuzzy/pyimpfuzzy

- Install Python module py2neo v3

```
$ pip install py2neo
```

For more information on the install procedures, please see:
http://py2neo.org/v3/#installation

- Download Python script pylouvain.py from the following webpage and save it to the same folder as impfuzzy_for_neo4j.py

  https://github.com/patapizza/pylouvain

4. Run Neo4j

   Run Neo4j by GUI or a command line.

5. Configure a password for Neo4j in impfuzzy_for_neo4j.py

   Configure the login password for Neo4j in impfuzzy_for_neo4j.py (change the {password} below).

```
NEO4J_PASSWORD = "{password}"
```

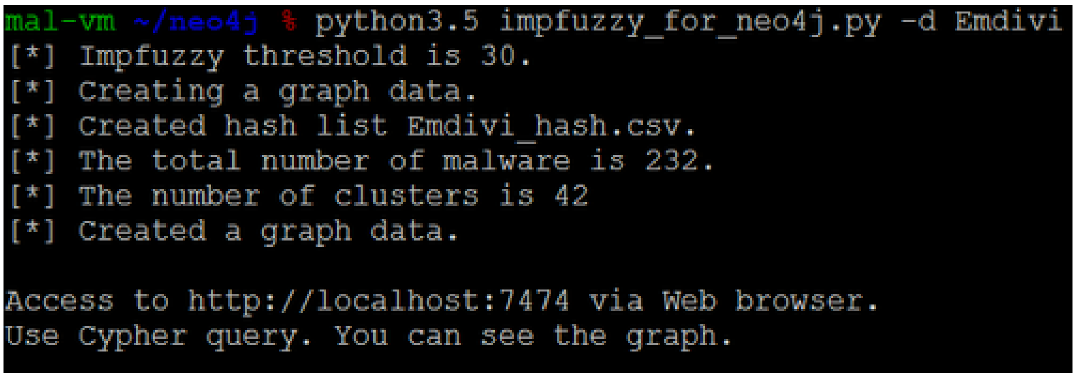
How to use impfuzzy for Neo4j

To use impfuzzy for Neo4j, use these options to specify the input of malware to cluster.

- -f - Specify malware (a file)
- -d - Specify a folder where malware is stored
- -l - Specify a CSV file(*) which lists malware

  (*) The format of CSV files are the following:
  File name, impfuzzy hash value, MD5 hash value, SHA1 hash value, SHA256 hash value

In the following example, malware is stored in the folder ‘Emdivi’ which is passed as a parameter.



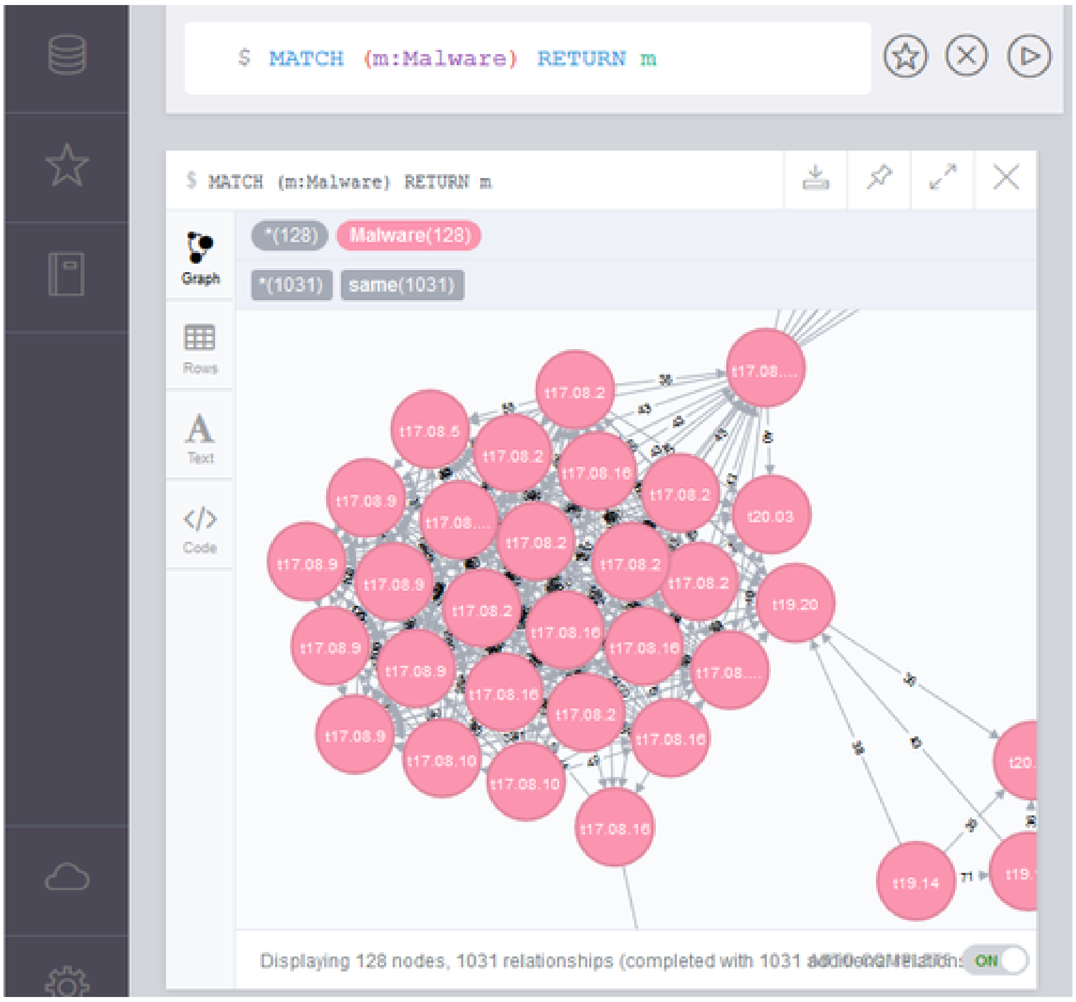
```
mal-vm ~/neo4j % python3.5 impfuzzy_for_neo4j.py -d Emdivi
[*] Impfuzzy threshold is 30.
[*] Creating a graph data.
[*] Created hash list Emdivi_hash.csv.
[*] The total number of malware is 232.
[*] The number of clusters is 42
[*] Created a graph data.

Access to http://localhost:7474 via Web browser.
Use Cypher query. You can see the graph.
```

Figure 3: impfuzzy for Neo4j execution result

Clustering results are registered in Neo4j database. Visualisation is available through the web interface of Neo4j, which is accessible from the URL below (The following is an example of Neo4j installed in a local environment).

```
http://localhost:7474/
```

For visualising a graph of clustering results, a Cypher query (a command to operate Neo4j database) needs to be executed through the web interface. Figure 4 is an example of executing a Cypher query through the web interface.



```
$ MATCH (m:Malware) RETURN m
```

*(128) Malware(128)
*(1031) same(1031)

Displaying 128 nodes, 1031 relationships (completed with 1031 additional relationships)

Figure 4: Example of Cypher query execution

Cypher queries to execute are different depending on what kind of clustering results you would like to visualise. Below are the examples of Cypher queries to visualise different clustering results.

[Example 1] Visualise all clustering results (Figure 1 is the result of the following Cypher query)

```
$ MATCH (m:Malware) RETURN m
```

[Example 2] Visualise a group of samples with a specific MD5 hash value (Figure 2 is an example of the following Cypher query)

```
MATCH (m1:Malware) WHERE m1.md5 = "[MD5 hash value]"
MATCH (m2:Malware) WHERE m2.cluster = m1.cluster

RETURN m2
```

[Example 3] Visualise all clustering results with impfuzzy similarity level over 90

```
$ MATCH (m:Malware)-[s:same]-() WHERE s.value > 90 RETURN m,s
```

Summary

Clustering large amount of malware to distinguish unknown types that needs to be analysed in a quick manner is crucial in malware analysis. We hope that impfuzzy for Neo4j will help such analysis tasks.

In a future entry, we will introduce the clustering and analysis results that we gained through this tool.

- Shusei Tomonaga

(Translated by Yukako Uchida)

Reference

[1] The Louvain method for community detection in large networks
http://perso.uclouvain.be/vincent.blondel/research/louvain.html