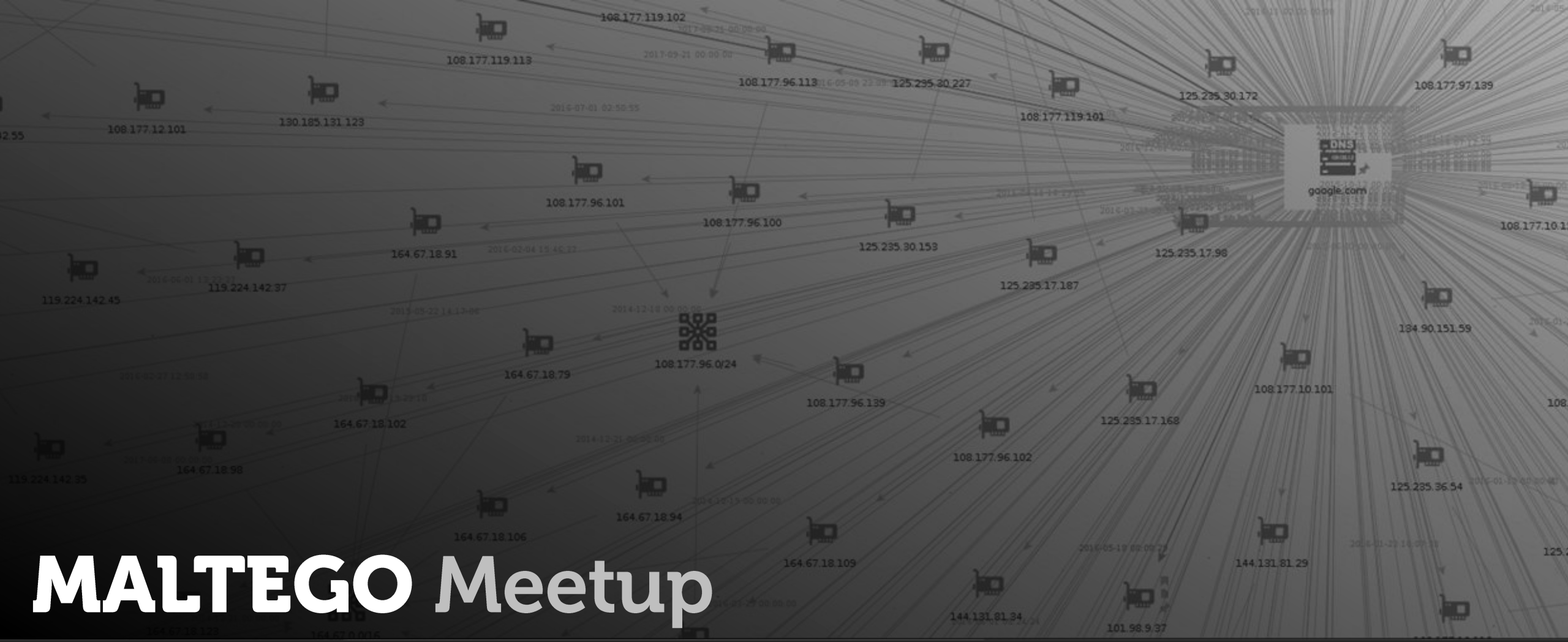


MALTEGO Meetup

PARIS - Wednesday, November 28th 2018

Felix Aimé (@felixaime)

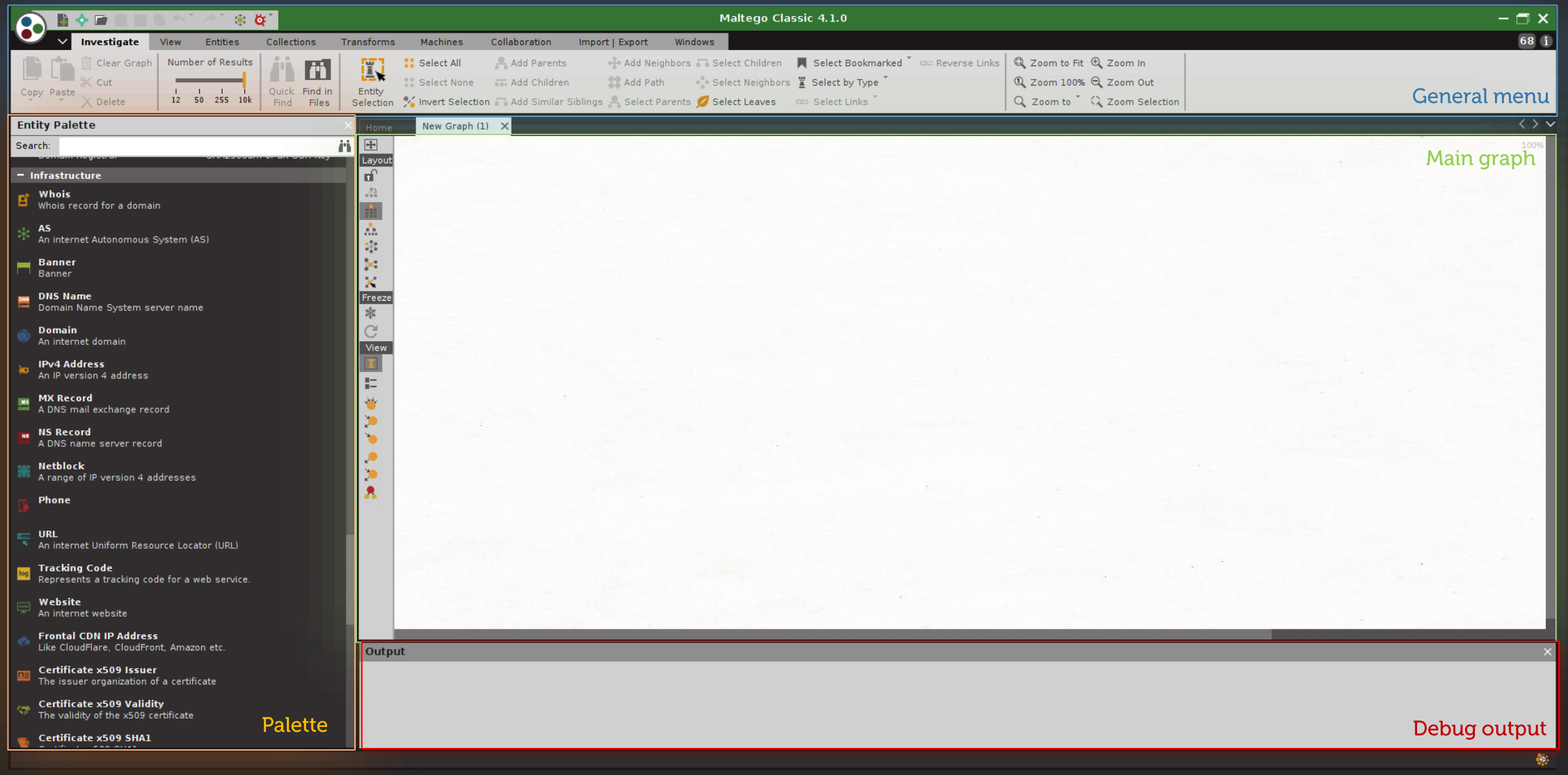
AGENDA

1. Maltego presentation for newbies
2. Transforms surgery
3. Analysts tips and tricks
4. Developpers tips and tricks

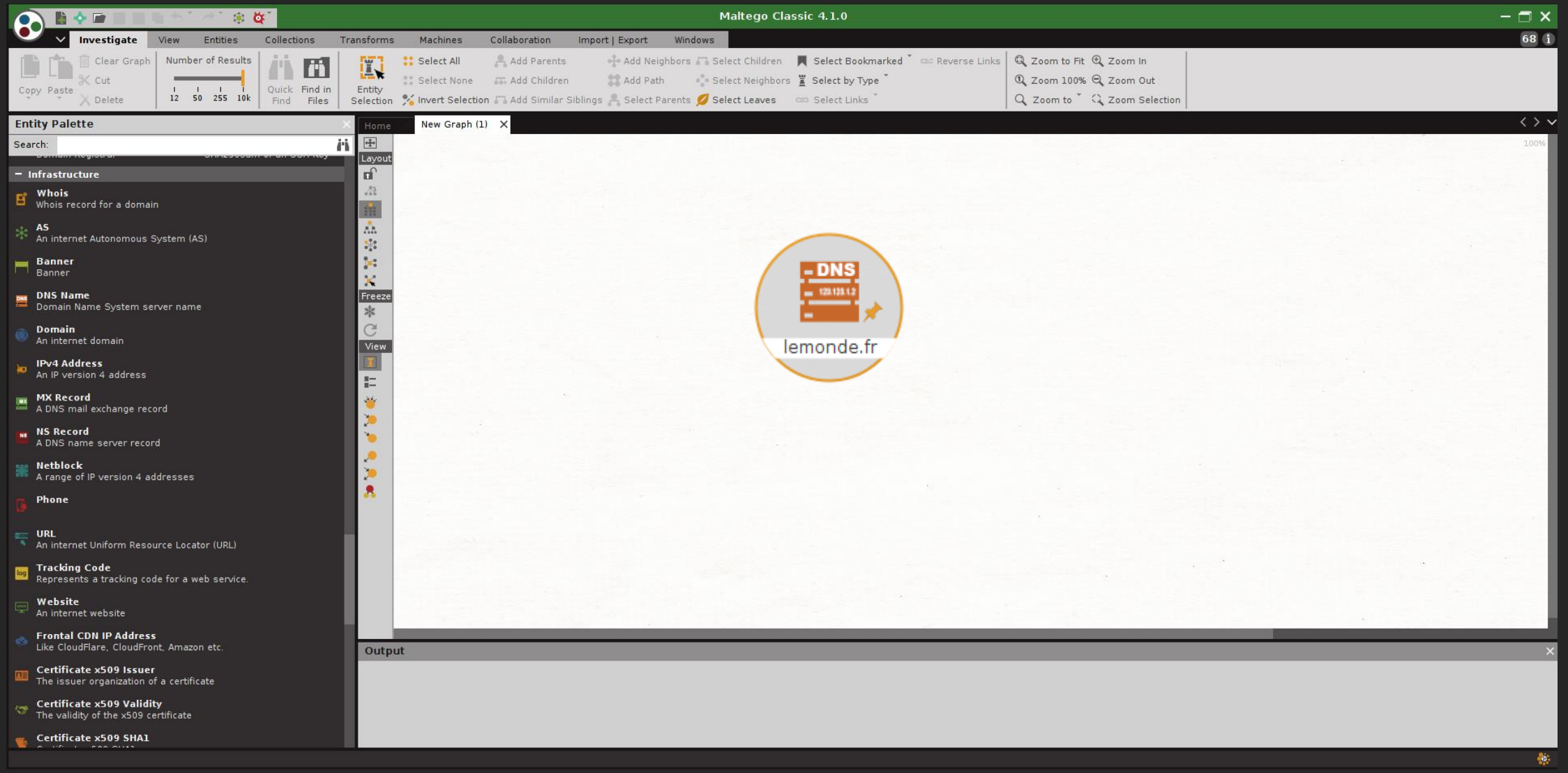
Maltego presentation for newbies :

Maltego it's like IDA,
without scripts it's an empty shell.

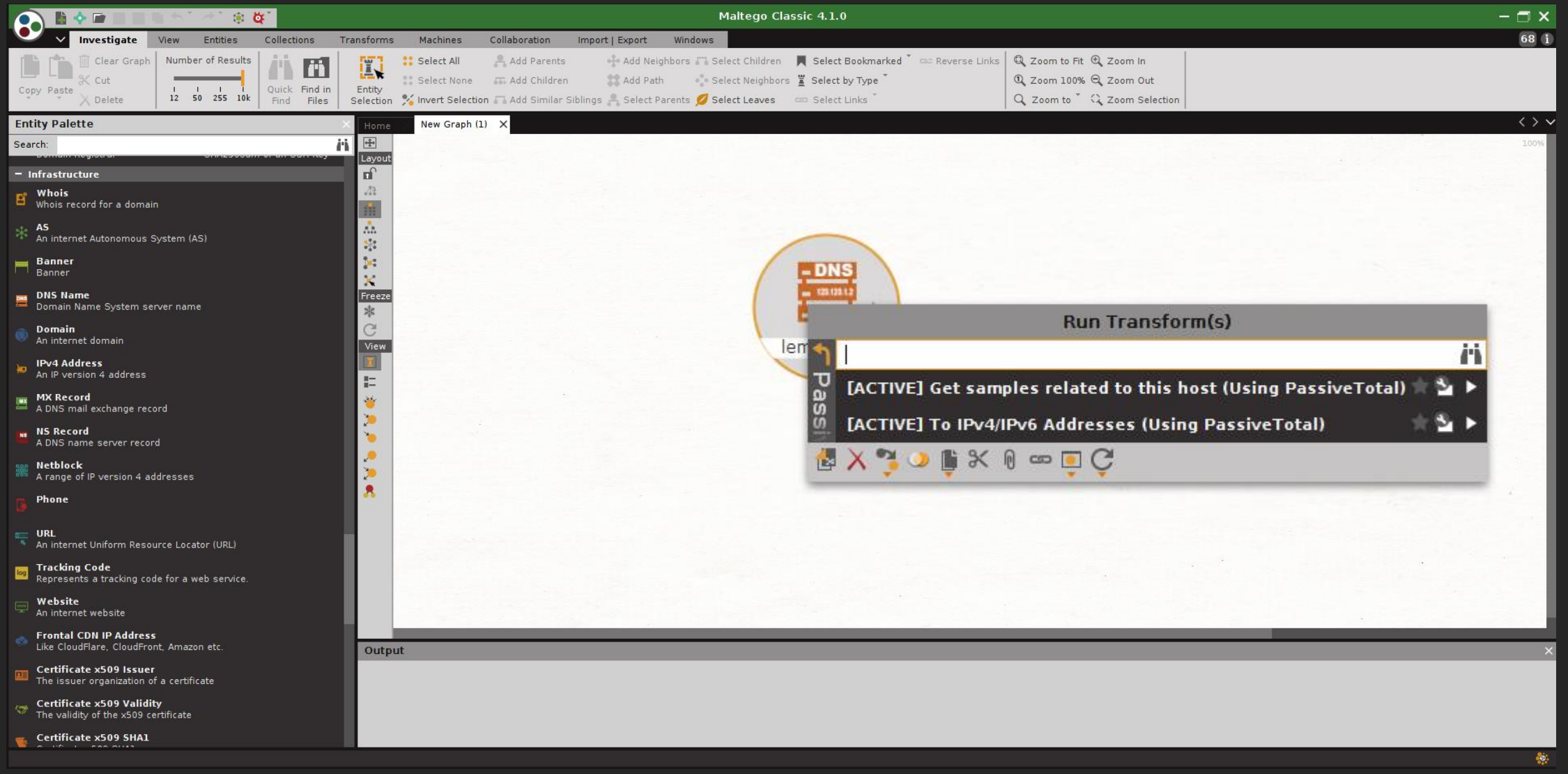
Maltego presentation for newbies: interface



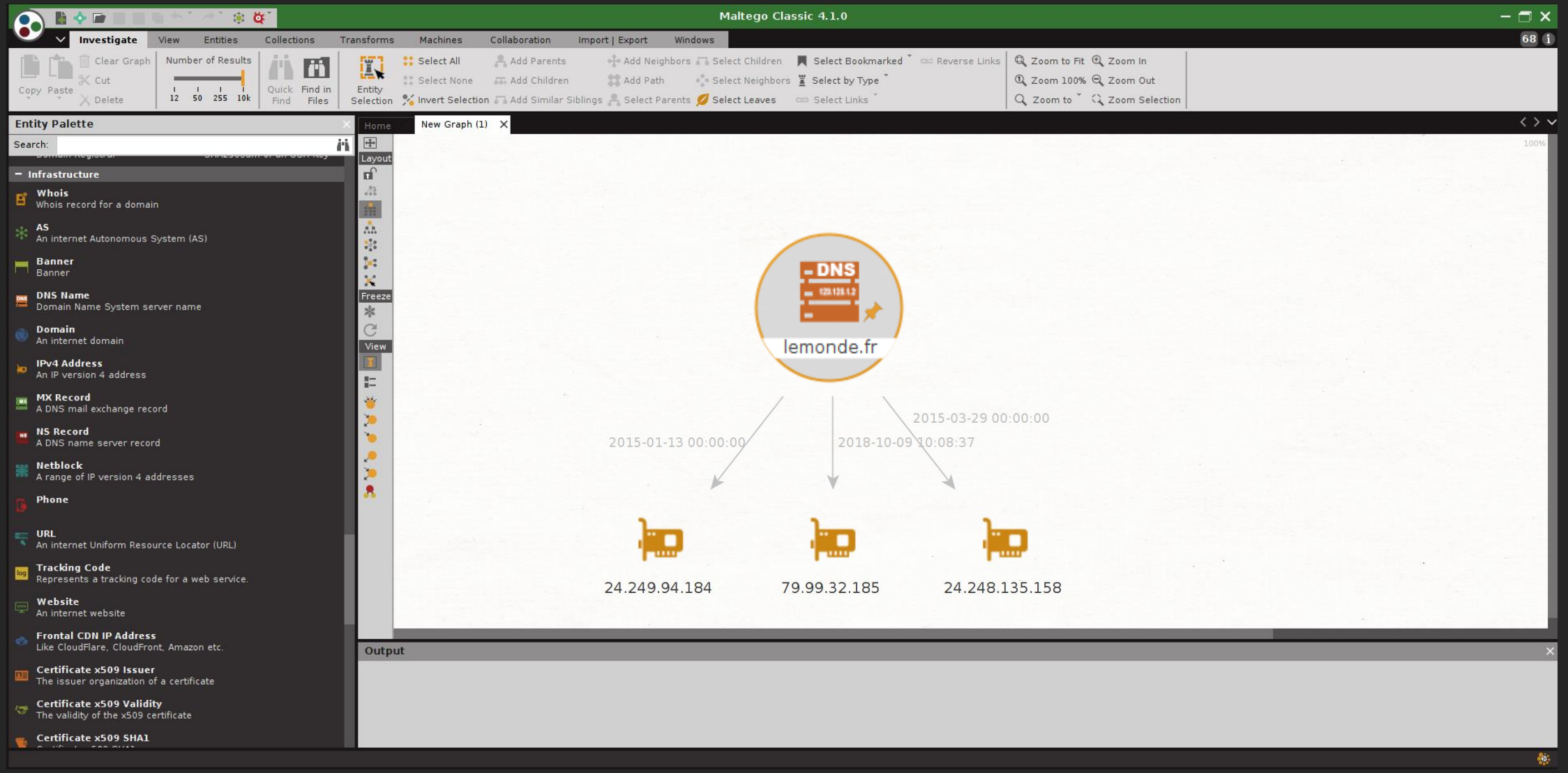
Maltego presentation for newbies: interface



Maltego presentation for newbies: interface



Maltego presentation for newbies: interface



You can create your own entities.

You can create your own transforms.

An entity can be any atomic object.

A transform is a specific operation that can be applied on an entity.

A transform can be applied on different entities.

Demo.

Local Transfroms.

Local transform development: an overview.

A transform is simply a local script which outputs maltego XML.

Local transform development: an overview.

In terminal it looks like that.

Local transform development: an overview.

```
$python3.5 passivedns.py lemonde.fr
<MaltegoMessage>
<MaltegoTransformResponseMessage>
<Entities>
<Entity Type="maltego.IPv4Address">
<Value>195.154.120.129</Value>
<Weight>100</Weight>
<AdditionalFields>
<Field MatchingRule="" Name="link#maltego.link.label" DisplayName="Label">2012-07-05 - 2018-11-25</Field>
</AdditionalFields>
</Entity>
<Entity Type="maltego.IPv4Address">
<Value>93.184.220.20</Value>
<Weight>100</Weight>
<AdditionalFields>
<Field MatchingRule="" Name="link#maltego.link.label" DisplayName="Label">2015-01-22 - 2018-05-18</Field>
</AdditionalFields>
</Entity>
</Entities>
<UIMessages>
</UIMessages>
</MaltegoTransformResponseMessage>
</MaltegoMessage>
```

Local transform development: an overview.

```
$python3.5 passivedns.py lemonde.fr
<MaltegoMessage>
<MaltegoTransformResponseMessage>
<Entities>
<Entity Type="maltego.IPv4Address">
<Value>195.154.120.129</Value>
<Weight>100</Weight>
<AdditionalFields>
<Field MatchingRule="" Name="link#maltego.link.label" DisplayName="Label">2012-07-05 - 2018-11-25</Field>
</AdditionalFields>
</Entity>
<Entity Type="maltego.IPv4Address">
<Value>93.184.220.20</Value>
<Weight>100</Weight>
<AdditionalFields>
<Field MatchingRule="" Name="link#maltego.link.label" DisplayName="Label">2015-01-22 - 2018-05-18</Field>
</AdditionalFields>
</Entity>
</Entities>
<UIMessages>
</UIMessages>
</MaltegoTransformResponseMessage>
</MaltegoMessage>
```

Local transform development: an overview.

```
$python3.5 passivedns.py lemonde.fr
<MaltegoMessage>
<MaltegoTransformResponseMessage>
<Entities>
<Entity Type="maltego.IPv4Address">
<Value>195.154.120.129</Value>
<Weight>100</Weight>
<AdditionalFields>
<Field MatchingRule="" Name="link#maltego.link.label" DisplayName="Label">2012-07-05 - 2018-11-25</Field>
</AdditionalFields>
</Entity>
<Entity Type="maltego.IPv4Address">
<Value>93.184.220.20</Value>
<Weight>100</Weight>
<AdditionalFields>
<Field MatchingRule="" Name="link#maltego.link.label" DisplayName="Label">2015-01-22 - 2018-05-18</Field>
</AdditionalFields>
</Entity>
</Entities>
<UIMessages>
</UIMessages>
</MaltegoTransformResponseMessage>
</MaltegoMessage>
```

Local transform development: an overview.

Paterva has developped many libraries for different languages.

Local transform development: in Python.

```
1  import TransformLib
2  import json
3  import sys
4
5  res = request.get("http://127.0.0.1:8888/pdns/%s" % (sys.argv[1]))
6  t   = TransformLib.MaltegoTransform()
7
8  for result in json.loads(res.content.decode('utf8')):
9      e = t.MaltegoEntity()
10     e.setType("maltego.IPv4Address")
11     e.setValue(result["ip_addr"])
12     t.addEntityToMessage(e)
13
14 t.returnOutput()
```

Local transform development: in Python.

```
1 import TransformLib
2 import json
3 import sys
4
5 res = request.get("http://127.0.0.1:8888/pdns/%s" % (sys.argv[1]))
6 t   = TransformLib.MaltegoTransform()
7
8 for result in json.loads(res.content.decode('utf8')):
9     e = t.MaltegoEntity()
10    e.setType("maltego.IPv4Address")
11    e.setValue(result["ip_addr"])
12    t.addEntityToMessage(e)
13
14 t.returnOutput()
```

Local transform development: Python way.

```
1 import TransformLib
2 import json
3 import sys
4
5 res = request.get("http://127.0.0.1:8888/pdns/%s" % (sys.argv[1]))
6 t   = TransformLib.MaltegoTransform()
7
8 for result in json.loads(res.content.decode('utf8')):
9     e = t.MaltegoEntity()
10    e.setType("maltego.IPv4Address")
11    e.setValue(result["ip_addr"])
12    t.addEntityToMessage(e)
13
14 t.returnOutput()
```

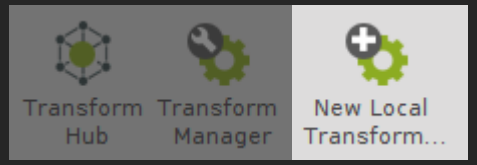
Local transform development: Python way.

```
1  import TransformLib
2  import json
3  import sys
4
5  res = request.get("http://127.0.0.1:8888/pdns/%s" % (sys.argv[1]))
6  t   = TransformLib.MaltegoTransform()
7
8  for result in json.loads(res.content.decode('utf8')):
9      e = t.MaltegoEntity()
10     e.setType("maltego.IPv4Address")
11     e.setValue(result["ip_addr"])
12     t.addEntityToMessage(e)
13
14 t.returnOutput()
```

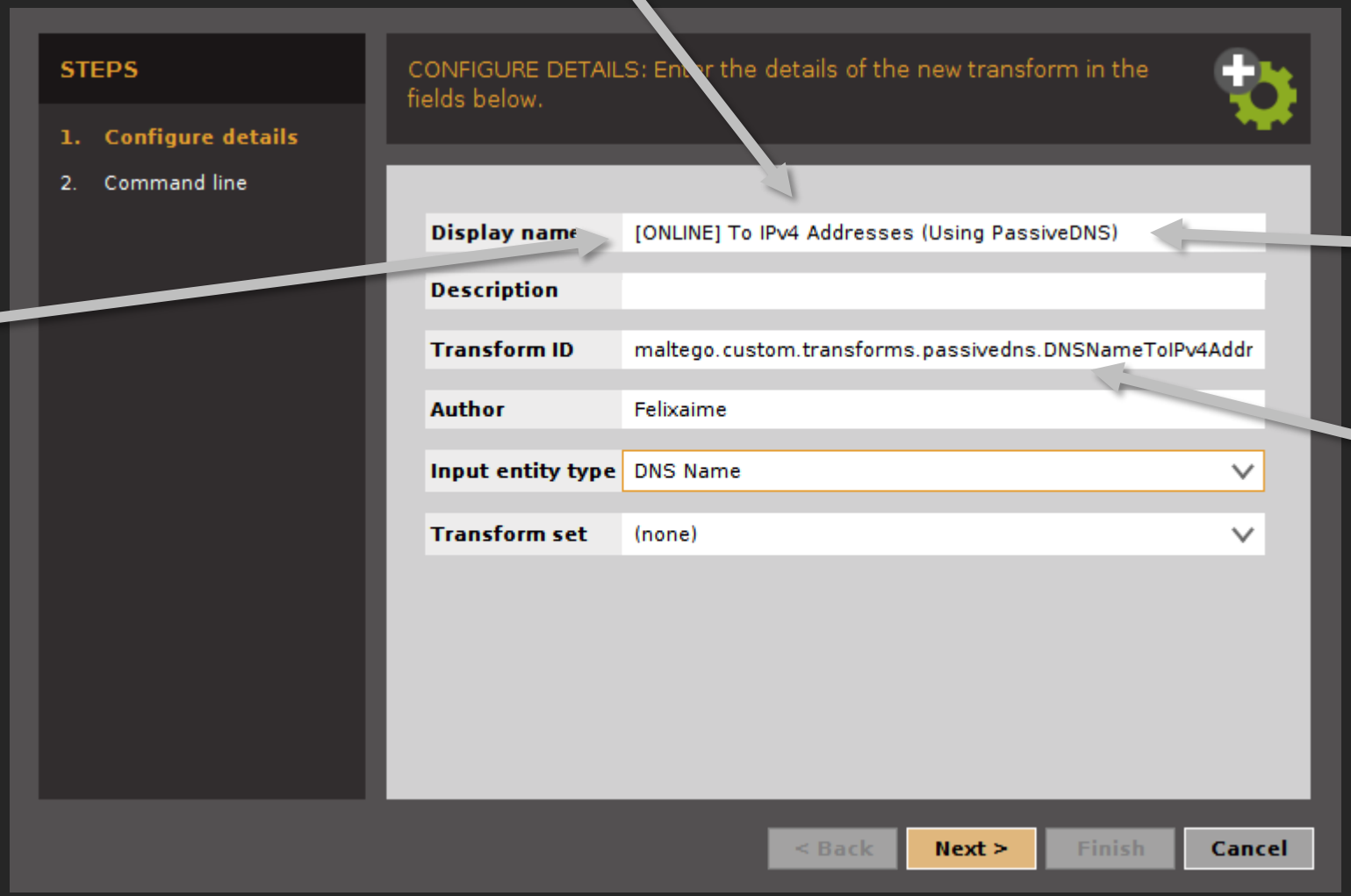
Local transform development: Python way.

```
 1 import TransformLib
 2 import json
 3 import sys
 4
 5 res = request.get("http://127.0.0.1:8888/pdns/%s" % (sys.argv[1]))
 6 t   = TransformLib.MaltegoTransform()
 7
 8 for result in json.loads(res.content.decode('utf8')):
 9     e = t.MaltegoEntity()
10     e.setType("maltego.IPv4Address")
11     e.setValue(result["ip_addr"])
12     t.addEntityToMessage(e)
13
14 t.returnOutput()
```

Local transform development: Installation



Local transform development: Installation

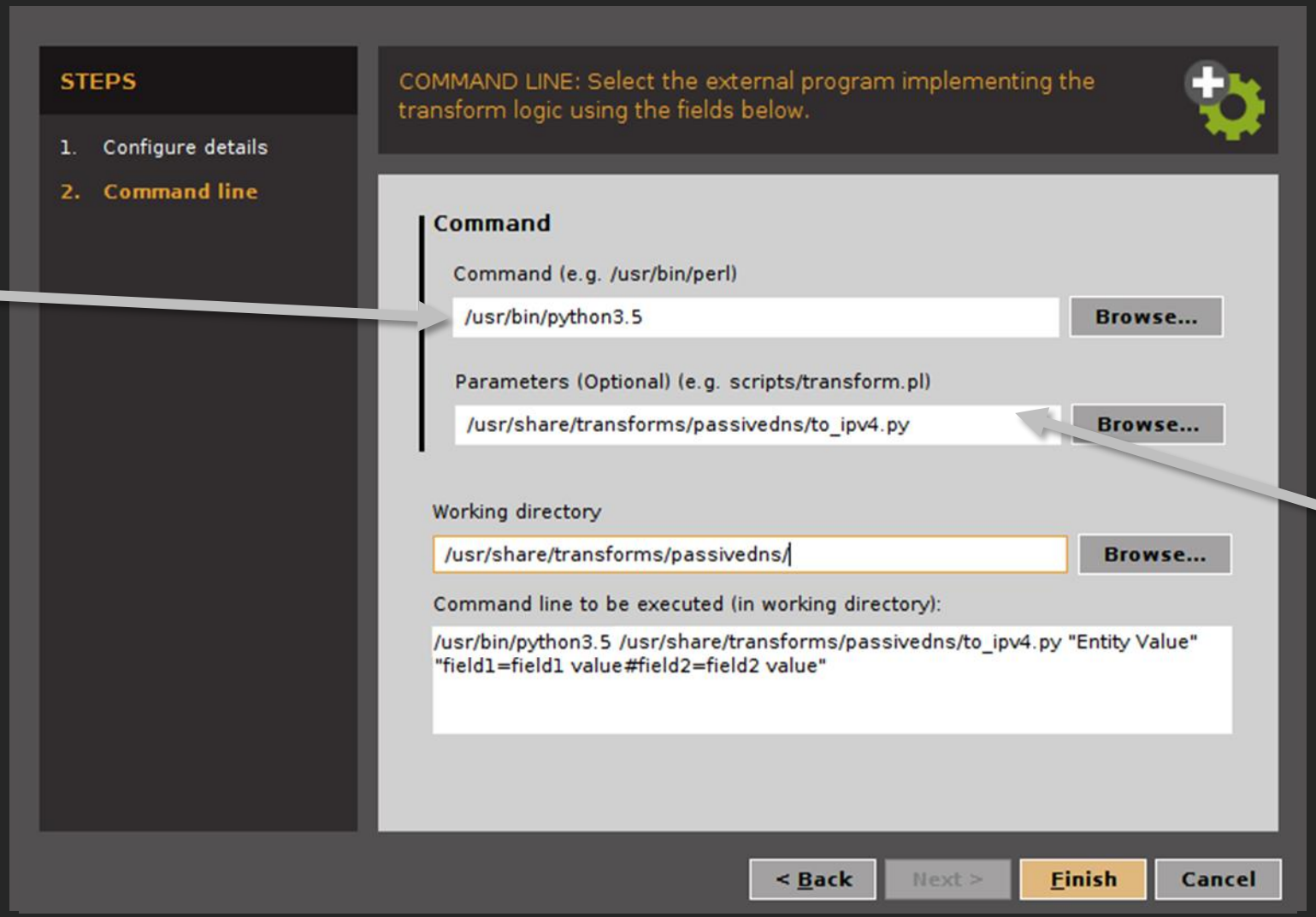


Local transform development: Namespaces

maltego.custom.transforms.<service>.<SrcEntity>To<DstEntity>

DNSNameToIP

Local transform development: Installation



Interpreter path

Full path to the Script transform (can contain arguments)

Local transform development: Where to store transforms?

/usr/share/transforms/<service>/transform.py –method=dnsname_to_ipv4

/home/mike/scripts/transforms/toipv4.py

And that's all folks, you've your PDNS in your Maltego instance.

Transforms surgery.

CASE #1

Directory Explorer & Metadata extraction

Case #1: Directory Explorer & Metadata extraction

2 Transforms (Explorer, Exiftool)

3 Entities (Directory, File, Software, Date)

Case #1: Step one, Directory Explorer

```
import TransformLib
import sys
import os

transform = TransformLib.MaltegoTransform()
current_directory = sys.argv[1]

for item in os.listdir(current_directory):
    if os.path.isdir(os.path.join(current_directory,item)):
        e = TransformLib.MaltegoEntity()
        e.setType("maltego.custom.entities.explorer.directory")
        e.setValue(os.path.join(current_directory,item))
        transform.addEntityToMessage(e)
    else:
        e = TransformLib.MaltegoEntity()
        e.setType("maltego.custom.entities.explorer.file")
        e.setValue(os.path.join(current_directory,item))
        transform.addEntityToMessage(e)

transform.returnOutput()
```

Case #1: Step one, Directory Explorer

```
import TransformLib
import sys
import os

transform = TransformLib.MaltegoTransform()
current_directory = sys.argv[1]

for item in os.listdir(current_directory):
    if os.path.isdir(os.path.join(current_directory,item)):
        e = TransformLib.MaltegoEntity()
        e.setType("maltego.custom.entities.explorer.directory")
        e.setValue(os.path.join(current_directory,item))
        transform.addEntityToMessage(e)
    else:
        e = TransformLib.MaltegoEntity()
        e.setType("maltego.custom.entities.explorer.file")
        e.setValue(os.path.join(current_directory,item))
        transform.addEntityToMessage(e)

transform.returnOutput()
```

Case #1: Step one, Directory Explorer

```
import TransformLib
import sys
import os

transform = TransformLib.MaltegoTransform()
current_directory = sys.argv[1]

for item in os.listdir(current_directory):
    if os.path.isdir(os.path.join(current_directory,item)):
        e = TransformLib.MaltegoEntity()
        e.setType("maltego.custom.entities.explorer.directory")
        e.setValue(os.path.join(current_directory,item))
        transform.addEntityToMessage(e)
    else:
        e = TransformLib.MaltegoEntity()
        e.setType("maltego.custom.entities.explorer.file")
        e.setValue(os.path.join(current_directory,item))
        transform.addEntityToMessage(e)

transform.returnOutput()
```

Case #1: Step one, Directory Explorer

```
import TransformLib
import sys
import os

transform = TransformLib.MaltegoTransform()
current_directory = sys.argv[1]

for item in os.listdir(current_directory):
    if os.path.isdir(os.path.join(current_directory,item)):
        e = TransformLib.MaltegoEntity()
        e.setType("maltego.custom.entities.explorer.directory")
        e.setValue(os.path.join(current_directory,item))
        transform.addEntityToMessage(e)
    else:
        e = TransformLib.MaltegoEntity()
        e.setType("maltego.custom.entities.explorer.file")
        e.setValue(os.path.join(current_directory,item))
        transform.addEntityToMessage(e)

transform.returnOutput()
```

Case #1: Step one, Directory Explorer

```
import TransformLib
import sys
import os

transform = TransformLib.MaltegoTransform()
current_directory = sys.argv[1]

for item in os.listdir(current_directory):
    if os.path.isdir(os.path.join(current_directory,item)):
        e = TransformLib.MaltegoEntity()
        e.setType("maltego.custom.entities.explorer.directory")
        e.setValue(os.path.join(current_directory,item))
        transform.addEntityToMessage(e)
    else:
        e = TransformLib.MaltegoEntity()
        e.setType("maltego.custom.entities.explorer.file")
        e.setValue(os.path.join(current_directory,item))
        transform.addEntityToMessage(e)

transform.returnOutput()
```

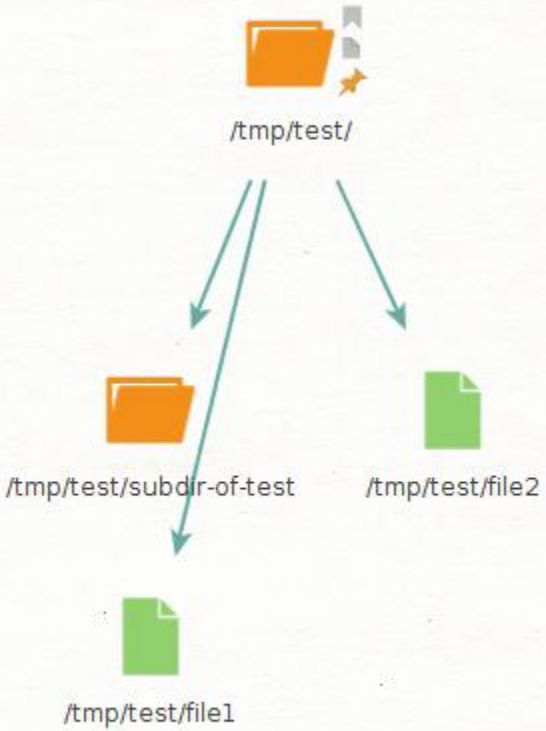
Case #1: Step one, Directory Explorer

```
import TransformLib
import sys
import os

transform = TransformLib.MaltegoTransform()
current_directory = sys.argv[1]

for item in os.listdir(current_directory):
    if os.path.isdir(os.path.join(current_directory,item)):
        e = TransformLib.MaltegoEntity()
        e.setType("maltego.custom.entities.explorer.directory")
        e.setValue(os.path.join(current_directory,item))
        transform.addEntityToMessage(e)
    else:
        e = TransformLib.MaltegoEntity()
        e.setType("maltego.custom.entities.explorer.file")
        e.setValue(os.path.join(current_directory,item))
        transform.addEntityToMessage(e)

transform.returnOutput()
```

Case #1: Step one, Directory Explorer

```
import TransformLib
import sys
import os

transform = TransformLib.MaltegoTransform()
current_directory = sys.argv[1]

for item in os.listdir(current_directory):
    if os.path.isdir(os.path.join(current_directory,item)):
        e = TransformLib.MaltegoEntity()
        e.setType("maltego.custom.entities.explorer.directory")
        e.setValue(os.path.join(current_directory,item))
        transform.addEntityToMessage(e)
    else:
        e = TransformLib.MaltegoEntity()
        e.setType("maltego.custom.entities.explorer.file")
        e.setValue(os.path.join(current_directory,item))
        transform.addEntityToMessage(e)

transform.returnOutput()
```

Case #1: Step one, Directory Explorer



Case #1: Step one, Directory Explorer

To improve it: put the full path in additional fields.

(See the full transform in the resources directory)

Case #1: Step two, Exiftool to Maltego.

```
$ exiftool 9f270b1cef15fa30f3b [...] ba99158ee4d4ba8c815.doc

ExifTool Version Number         : 10.80
File Name                       : 9f270b1cef15fa30f3b [...] ba99158ee4d4ba8c815.doc
Directory                       : .
File Size                       : 848 kB File Modification
Date/Time                       : 2018:08:07 15:46:36+02:00
File Access Date/Time           : 2018:11:26 18:20:10+01:00
File Inode Change Date/Time     : 2018:08:07 15:46:43+02:00
File Permissions                : rw-rw-r–
File Type                       : DOC
File Type Extension             : doc
MIME Type                       : application/msword
Author                          : James
Template                        : Normal.dotm Last
Modified By                     : john
Revision Number                 : 2
Software                        : Microsoft Office Word
Total Edit Time                 : 6.0 minutes
Create Date                     : 2018:03:15 09:09:00
Modify Date                     : 2018:03:15 09:26:00
```

Case #1: Step two, Exiftool to Maltego.

```
$ exiftool 9f270b1cef15fa30f3b [...] ba99158ee4d4ba8c815.doc

ExifTool Version Number         : 10.80
File Name                       : 9f270b1cef15fa30f3b [...] ba99158ee4d4ba8c815.doc
Directory                       : .
File Size                       : 848 kB File Modification
Date/Time                       : 2018:08:07 15:46:36+02:00
File Access Date/Time           : 2018:11:26 18:20:10+01:00
File Inode Change Date/Time     : 2018:08:07 15:46:43+02:00
File Permissions                : rw-rw-r–
File Type                       : DOC
File Type Extension             : doc
MIME Type                       : application/msword
Author                          : James
Template                        : Normal.dotm Last
Modified By                     : john
Revision Number                 : 2
Software                        : Microsoft Office Word
Total Edit Time                 : 6.0 minutes
Create Date                     : 2018:03:15 09:09:00
Modify Date                     : 2018:03:15 09:26:00
```

Case #1: Step two, Exiftool to Maltego.

```
import TransformLib
import subprocess
import sys

transform    = TransformLib.MaltegoTransform()
current_file = sys.argv[1]

p   = subprocess.run( [ "exiftool", current_file ], shell=False, stdout=subprocess.PIPE, timeout=10)
res = (p.stdout).decode("utf-8")

for line in res.splitlines():
    try:
        metadata_key   = line.split(" : ")[0].strip()
        metadata_value = line.split(" : ")[1].strip()

        if "Author" in metadata_key:
            e = TransformLib.MaltegoEntity()
            e.setType("maltego.Alias")
            e.setValue(metadata_value)
            self.transform.addEntityToMessage(e)
        # elif "Creator" in metadata_key...
    except:
        continue
        self.transform.returnOutput()

transform.returnOutput()
```

Case #1: Step two, Exiftool to Maltego.

```
import TransformLib
import subprocess
import sys

transform    = TransformLib.MaltegoTransform()
current_file = sys.argv[1]

p   = subprocess.run( [ "exiftool", current_file ], shell=False, stdout=subprocess.PIPE, timeout=10)
res = (p.stdout).decode("utf-8")

for line in res.splitlines():
    try:
        metadata_key   = line.split(" : ")[0].strip()
        metadata_value = line.split(" : ")[1].strip()

        if "Author" in metadata_key:
            e = TransformLib.MaltegoEntity()
            e.setType("maltego.Alias")
            e.setValue(metadata_value)
            self.transform.addEntityToMessage(e)
        # elif "Creator" in metadata_key...
    except:
        continue
        self.transform.returnOutput()

transform.returnOutput()
```

Case #1: Step two, Exiftool to Maltego.

```
import TransformLib
import subprocess
import sys

transform    = TransformLib.MaltegoTransform()
current_file = sys.argv[1]

p   = subprocess.run( [ "exiftool", current_file ], shell=False, stdout=subprocess.PIPE, timeout=10)
res = (p.stdout).decode("utf-8")

for line in res.splitlines():
    try:
        metadata_key   = line.split(" : ")[0].strip()
        metadata_value = line.split(" : ")[1].strip()

        if "Author" in metadata_key:
            e = TransformLib.MaltegoEntity()
            e.setType("maltego.Alias")
            e.setValue(metadata_value)
            self.transform.addEntityToMessage(e)
        # elif "Creator" in metadata_key...
    except:
        continue
        self.transform.returnOutput()

transform.returnOutput()
```

Case #1: Step two, Exiftool to Maltego.

```
import TransformLib
import subprocess
import sys

transform    = TransformLib.MaltegoTransform()
current_file = sys.argv[1]

p   = subprocess.run( [ "exiftool", current_file ], shell=False, stdout=subprocess.PIPE, timeout=10)
res = (p.stdout).decode("utf-8")

for line in res.splitlines():
    try:
        metadata_key   = line.split(" : ")[0].strip()
        metadata_value = line.split(" : ")[1].strip()

        if "Author" in metadata_key:
            e = TransformLib.MaltegoEntity()
            e.setType("maltego.Alias")
            e.setValue(metadata_value)
            self.transform.addEntityToMessage(e)
        # elif "Creator" in metadata_key...
    except:
        continue
        self.transform.returnOutput()

transform.returnOutput()
```

Case #1: Step two, Exiftool to Maltego.

```
import TransformLib
import subprocess
import sys

transform    = TransformLib.MaltegoTransform()
current_file = sys.argv[1]

p   = subprocess.run( [ "exiftool", current_file ], shell=False, stdout=subprocess.PIPE, timeout=10)
res = (p.stdout).decode("utf-8")

for line in res.splitlines():
    try:
        metadata_key   = line.split(" : ")[0].strip()
        metadata_value = line.split(" : ")[1].strip()

        if "Author" in metadata_key:
            e = TransformLib.MaltegoEntity()
            e.setType("maltego.Alias")
            e.setValue(metadata_value)
            self.transform.addEntityToMessage(e)
        # elif "Creator" in metadata_key...
    except:
        continue
        self.transform.returnOutput()

transform.returnOutput()
```

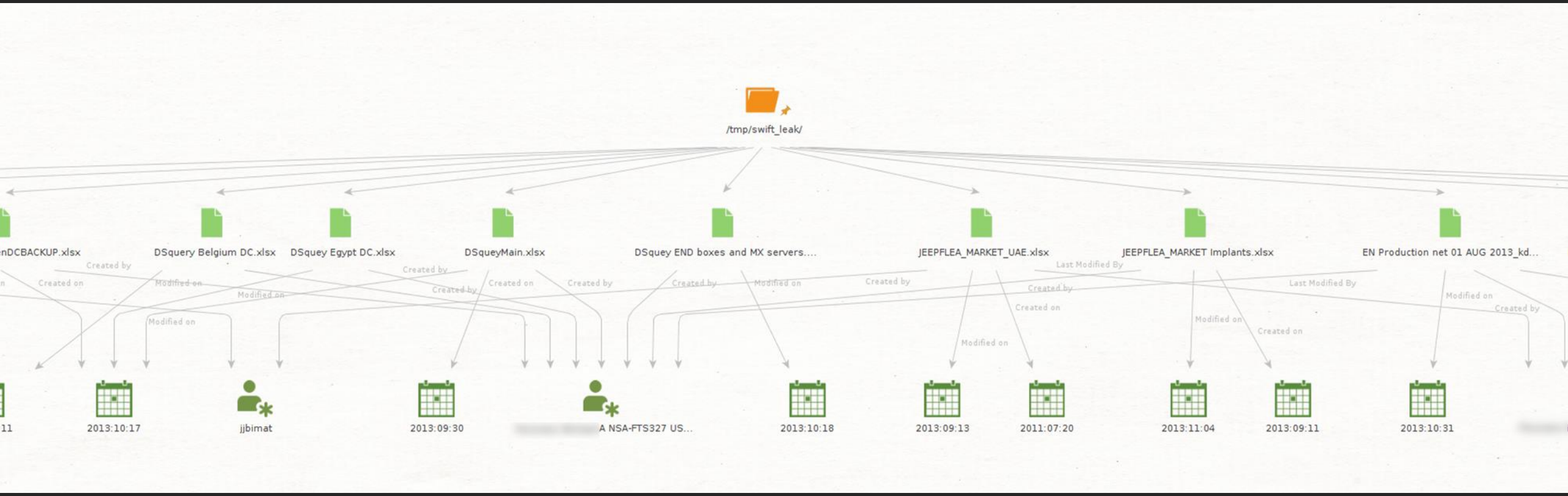
Case #1: Step two, Exiftool to Maltego

To improve it: add as label the metadata key.

(See the full transform in the resources directory)

Demo

./transforms/exiftool/transform.py



EXAMPLE: SIMPLE METADATA EXTRACTION FROM TSB LEAKS

CASE #2

Getting TOR node fingerprint and nickname from an IPv4 Address

Case #2: Web scrapping using requests and LXML.

Onionite

Search results for '[redacted]':

#	Nickname	Bandwidth	Uptime	Country	Flags	Type
1	[redacted]	1.02 MB/s	451d 7h	Netherlands		Relay

v1.1.3 - About - Source code - Report a bug

A thing by @lukechilds

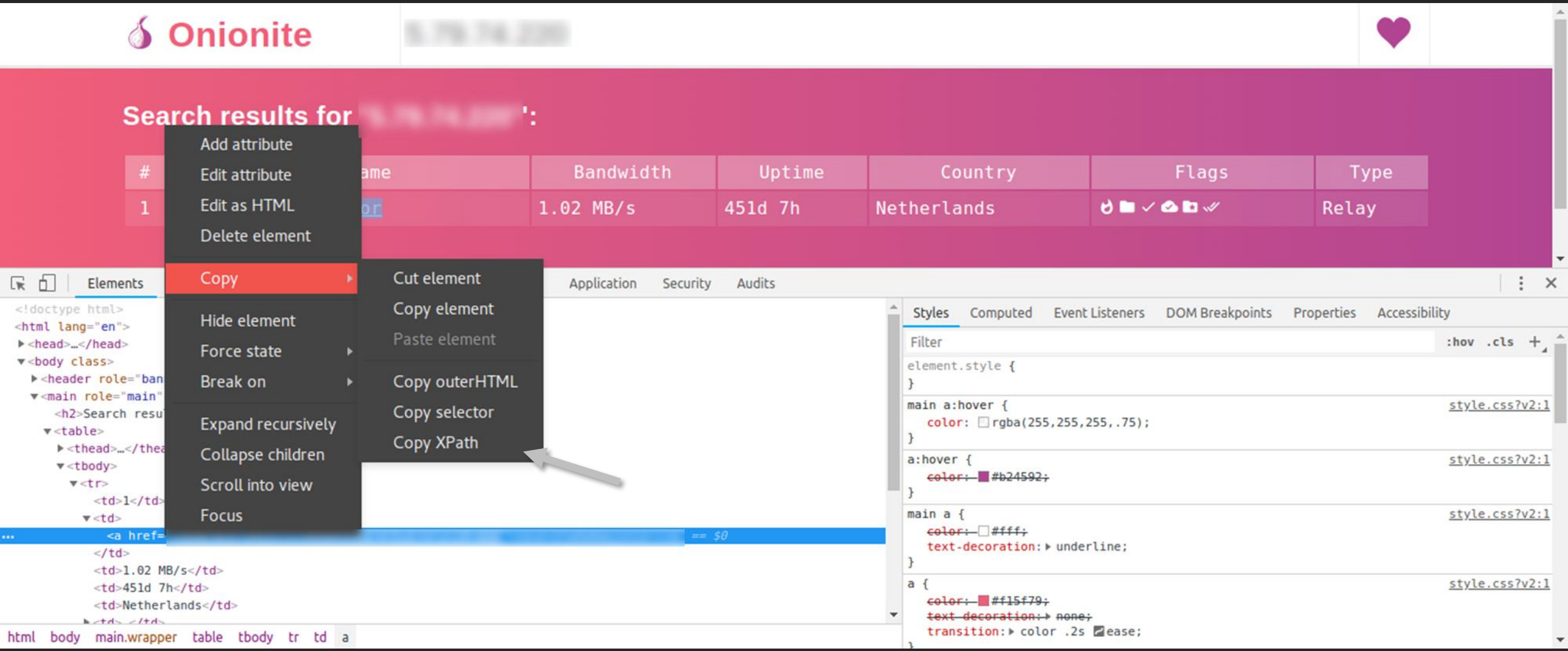
Case #2: Web scrapping using requests and LXML.

```
from lxml import etree
import TransformLib
import requests
import sys

transform = TransformLib.MaltegoTransform()
res       = requests.get("https://onionite.now.sh/?s=%s" % (sys.argv[1]))
tree      = etree.fromstring(res.content, etree.HTMLParser())

for td in tree.xpath("/html/body/main/table/tbody/tr"):

    e = TransformLib.MaltegoEntity()
    e.setType("maltego.custom.entities.infrastructure.TorExitNode")
    e.addAdditionalFields(fieldName="nickname",
                          displayName="Nickname",
                          value=str(td[1][0].text),
                          matchingRule="strict")
    e.setValue(td[1][0].attrib['href'][-40:])
    transform.addEntityToMessage(e)

transform.returnOutput()
```

Case #2: Web scrapping using requests and LXML.

```
from lxml import etree
import TransformLib
import requests
import sys

transform = TransformLib.MaltegoTransform()
res       = requests.get("https://onionite.now.sh/?s=%s" % (sys.argv[1]))
tree      = etree.fromstring(res.content, etree.HTMLParser())

for td in tree.xpath("/html/body/main/table/tbody/tr"):

    e = TransformLib.MaltegoEntity()
    e.setType("maltego.custom.entities.infrastructure.TorExitNode")
    e.addAdditionalFields(fieldName="nickname",
                          displayName="Nickname",
                          value=str(td[1][0].text),
                          matchingRule="strict")
    e.setValue(td[1][0].attrib['href'][-40:])
    transform.addEntityToMessage(e)

transform.returnOutput()
```

Case #2: Web scrapping using requests and LXML.



Case #2: Web scrapping using requests and LXML.

```
from lxml import etree
import TransformLib
import requests
import sys

transform = TransformLib.MaltegoTransform()
res       = requests.get("https://onionite.now.sh/?s=%s" % (sys.argv[1]))
tree      = etree.fromstring(res.content, etree.HTMLParser())

for td in tree.xpath("/html/body/main/table/tbody/tr"):

    e = TransformLib.MaltegoEntity()
    e.setType("maltego.custom.entities.infrastructure.TorExitNode")
    e.addAdditionalFields(fieldName="nickname",
                          displayName="Nickname",
                          value=str(td[1][0].text),
                          matchingRule="strict")
    e.setValue(td[1][0].attrib['href'][-40:])
    transform.addEntityToMessage(e)

transform.returnOutput()
```

Case #2: Web scrapping using requests and LXML.

```
from lxml import etree
import TransformLib
import requests
import sys

transform = TransformLib.MaltegoTransform()
res       = requests.get("https://onionite.now.sh/?s=%s" % (sys.argv[1]))
tree      = etree.fromstring(res.content, etree.HTMLParser())

for td in tree.xpath("/html/body/main/table/tbody/tr"):

    e = TransformLib.MaltegoEntity()
    e.setType("maltego.custom.entities.infrastructure.TorExitNode")
    e.addAdditionalFields(fieldName="nickname",
                          displayName="Nickname",
                          value=str(td[1][0].text),
                          matchingRule="strict")
    e.setValue(td[1][0].attrib['href'][-40:])
    transform.addEntityToMessage(e)

transform.returnOutput()
```

Case #2: Web scrapping using requests and LXML.

```
from lxml import etree
import TransformLib
import requests
import sys

transform = TransformLib.MaltegoTransform()
res       = requests.get("https://onionite.now.sh/?s=%s" % (sys.argv[1]))
tree      = etree.fromstring(res.content, etree.HTMLParser())

for td in tree.xpath("/html/body/main/table/tbody/tr"):

    e = TransformLib.MaltegoEntity()
    e.setType("maltego.custom.entities.infrastructure.TorExitNode")
    e.addAdditionalFields(fieldName="nickname",
                          displayName="Nickname",
                          value=str(td[1][0].text),
                          matchingRule="strict")
    e.setValue(td[1][0].attrib['href'][-40:])
    transform.addEntityToMessage(e)

transform.returnOutput()
```

Case #2: Web scrapping using requests and LXML.

```
from lxml import etree
import TransformLib
import requests
import sys

transform = TransformLib.MaltegoTransform()
res       = requests.get("https://onionite.now.sh/?s=%s" % (sys.argv[1]))
tree      = etree.fromstring(res.content, etree.HTMLParser())

for td in tree.xpath("/html/body/main/table/tbody/tr"):

    e = TransformLib.MaltegoEntity()
    e.setType("maltego.custom.entities.infrastructure.TorExitNode")
    e.addAdditionalFields(fieldName="nickname",
                          displayName="Nickname",
                          value=str(td[1][0].text),
                          matchingRule="strict")
    e.setValue(td[1][0].attrib['href'][-40:])
    transform.addEntityToMessage(e)

transform.returnOutput()
```

Case #2: Web scrapping using requests and LXML.

Title : [ONLINE] To TOR Exit node (Using Onionite)
Input entity : maltego.custom.entities.infrastructure.entities.TorExitNode
namespace : maltego.custom.transforms.infrastructure.onionite.IPv4toExitNode
Command : /usr/bin/python3.5
Parameters : /usr/share/transforms/onionite/transform.py --method=ipv4_to_exitnode

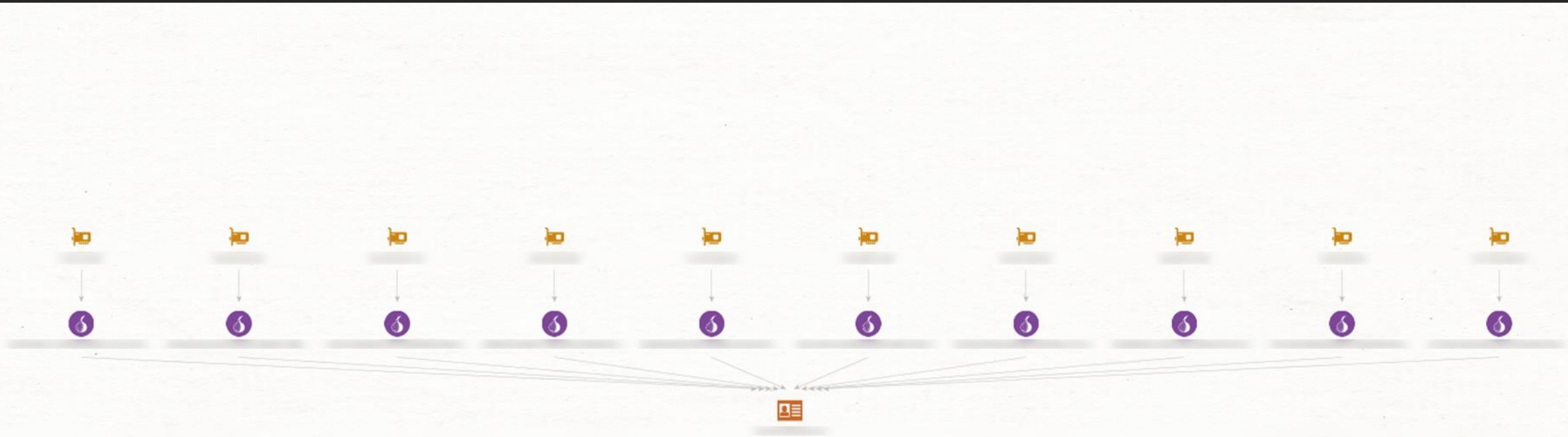
Title : [ONLINE] To NickName (Using Onionite)
Input entity : maltego.custom.entities.infrastructure.entities.TorExitNode
namespace : maltego.custom.transforms.infrastructure.onionite.TorExitNodeToNickname
Command : /usr/bin/python3.5
Parameters : /usr/share/transforms/onionsite/transform.py --method=exitnode_to_nickname

Title : [ONLINE] To TOR Exit nodes (Using Onionite)
Input entity : maltego.custom.entities.infrastructure.TorNicknameToExitNodes
namespace : maltego.custom.transforms.infrastructure.onionite.IPv4toExitNodes
Command : /usr/bin/python3.5
Parameters : /usr/share/transforms/onionite/transform.py --method=nickname_to_exit_nodes

Title : [ONLINE] To IPv4 Addresses (Using Onionite)
Input entity : maltego.custom.entities.infrastructure.entities.TorExitNode
namespace : maltego.custom.transforms.infrastructure.onionite.ExitNodeToIPv4
Command : /usr/bin/python3.5
Parameters : /usr/share/transforms/onionite/transform.py --method=exit_node_to_ipv4

Demo

./transforms/onionite/transform.py

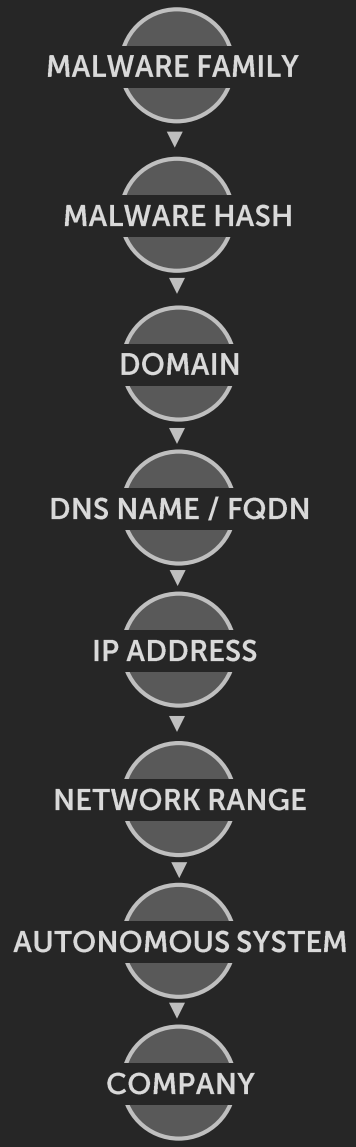


EXAMPLE: USE OF A SAME NICKNAME FOR TOR NODES BY AN INTRUSION SET.

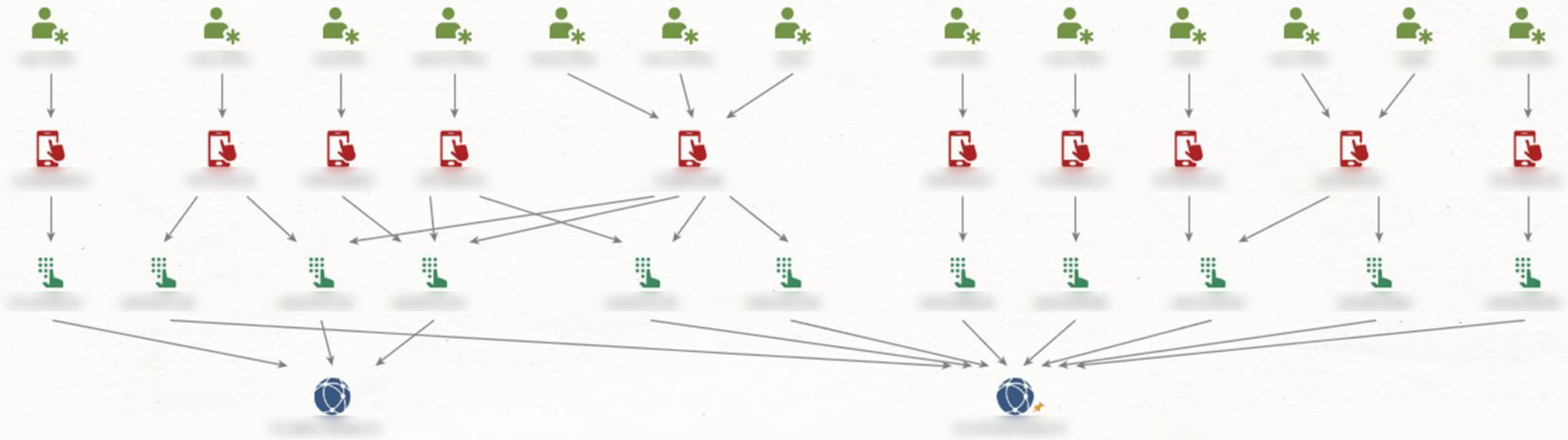
Analyst tips & Tricks

ANALYST TIP #1

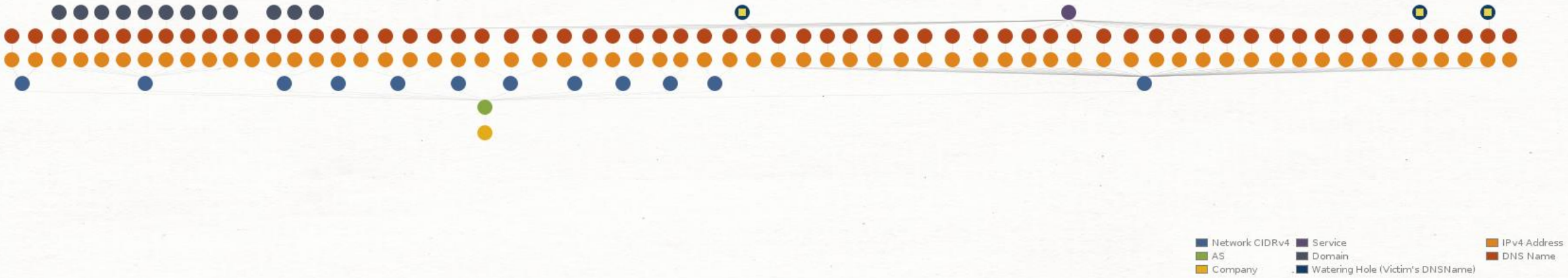
Stack your graph.



* This is just an example.



EXAMPLE: USERS RELATED TO MOBILE IMPLANTS AND THEIR C2s.



EXAMPLE: A WATERING HOLE CAMPAIGN, EACH DNSNAME IS A DELIVERY SERVER

ANALYST TIP #2

Use keyboard shortcuts.

ANALYST TIP #3

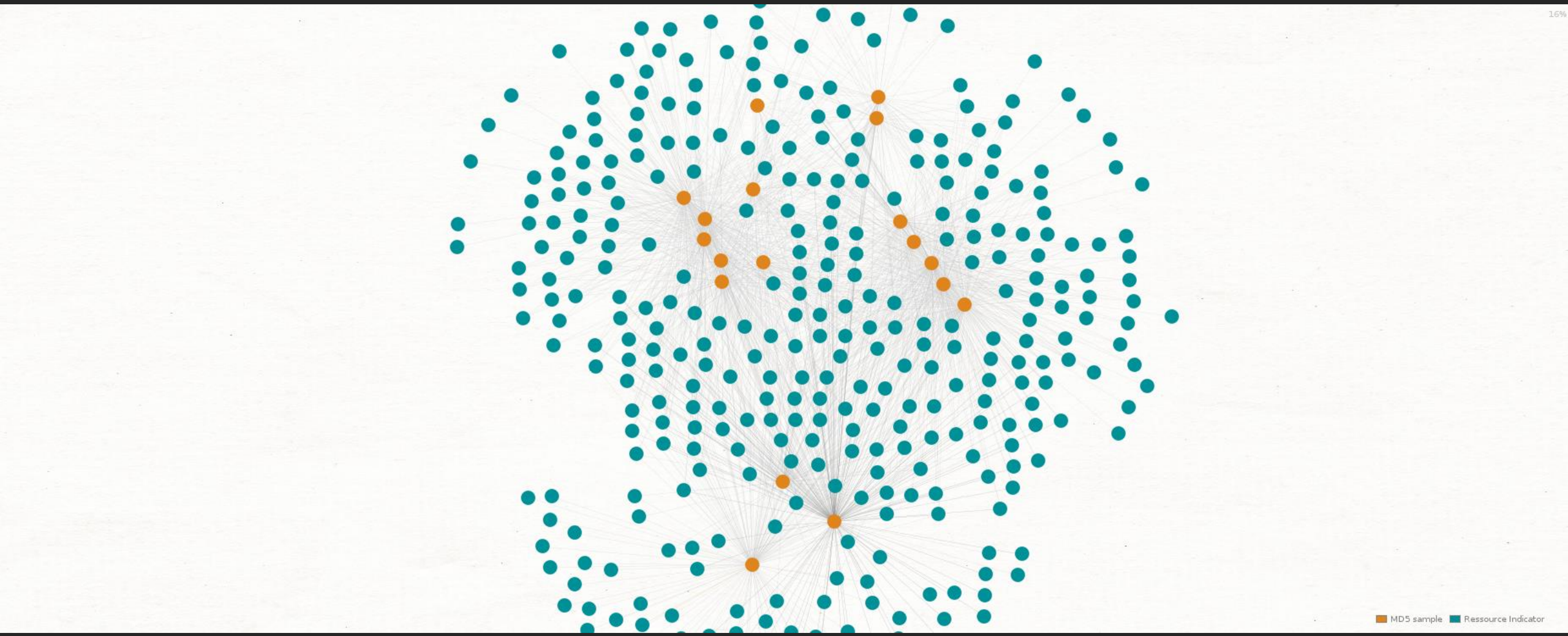
Use physical view for easy-correlation.

EXAMPLE: FOR MALWARE CLUSTERIZATION

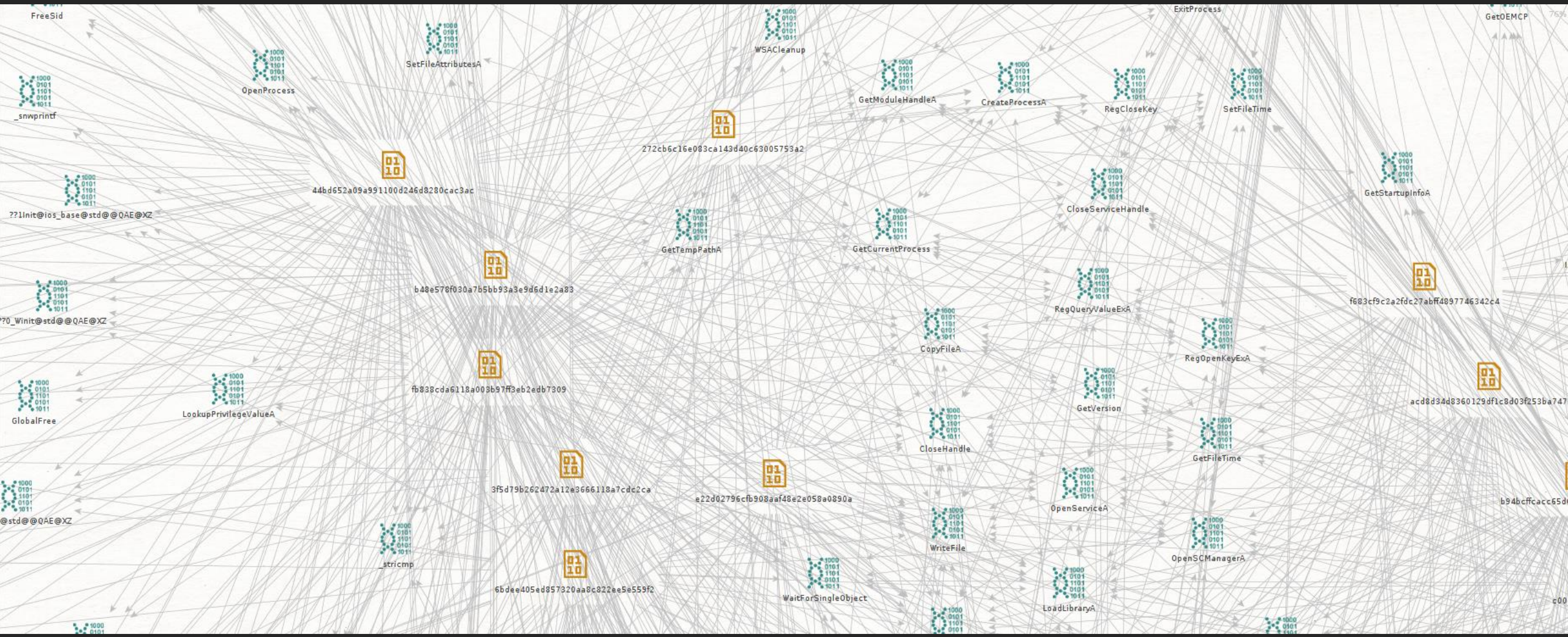
ANALYST TIP #3

Use organic view for
quick and dirty correlation.

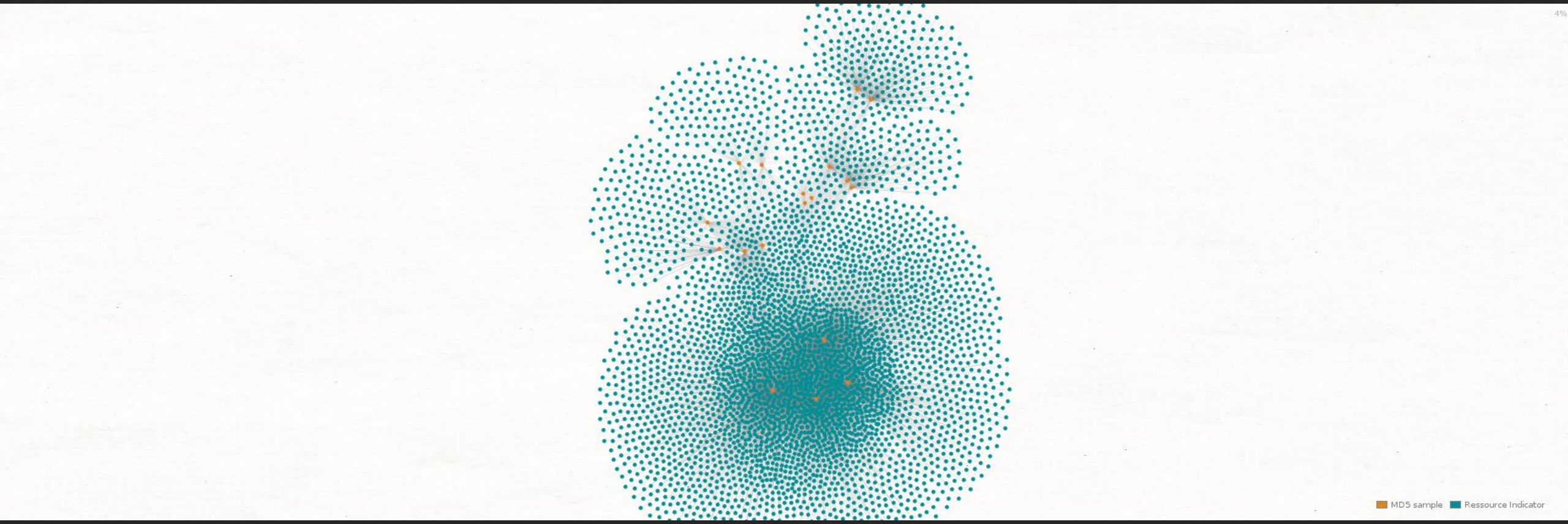
EXAMPLE: FOR MALWARE CLUSTERIZATION



EXAMPLE: PIRPI CLUSTERIZATION BASED ON IMPORTS



EXAMPLE: PIRPI CLUSTERIZATION BASED ON IMPORTS

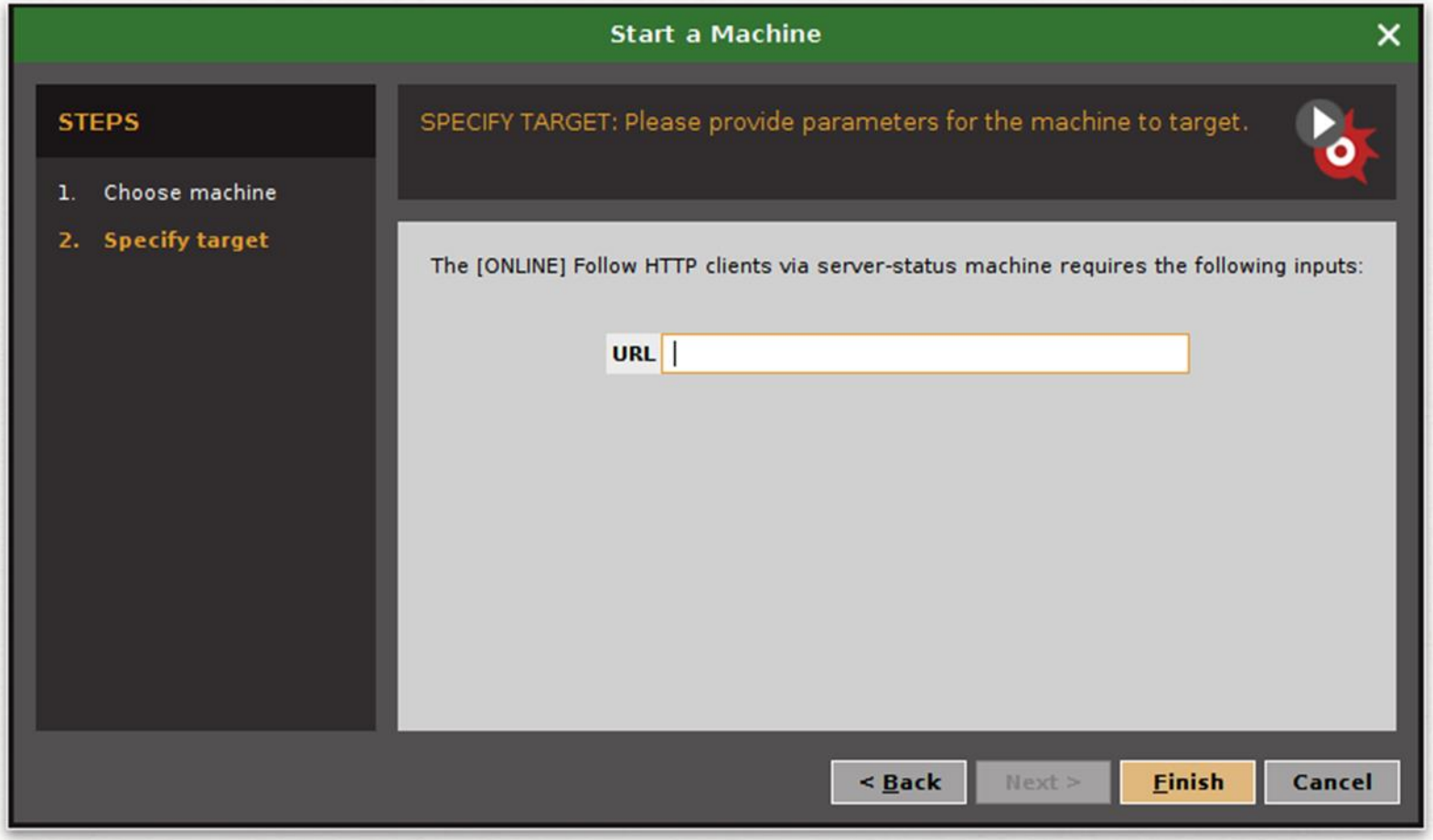


EXAMPLE: PIRPI CLUSTERIZATION BASED ON STRINGS

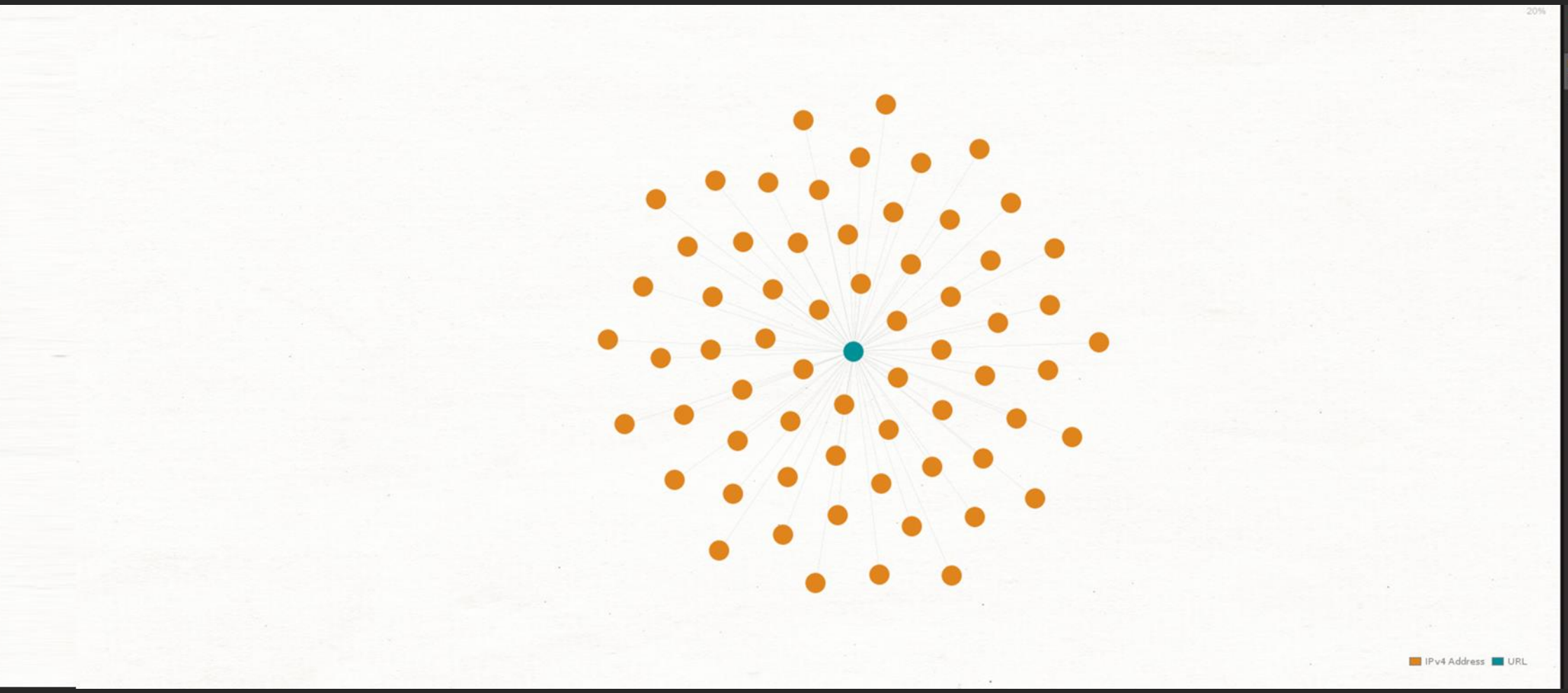
ANALYST TIP #4

For real-time based graph, use machines.

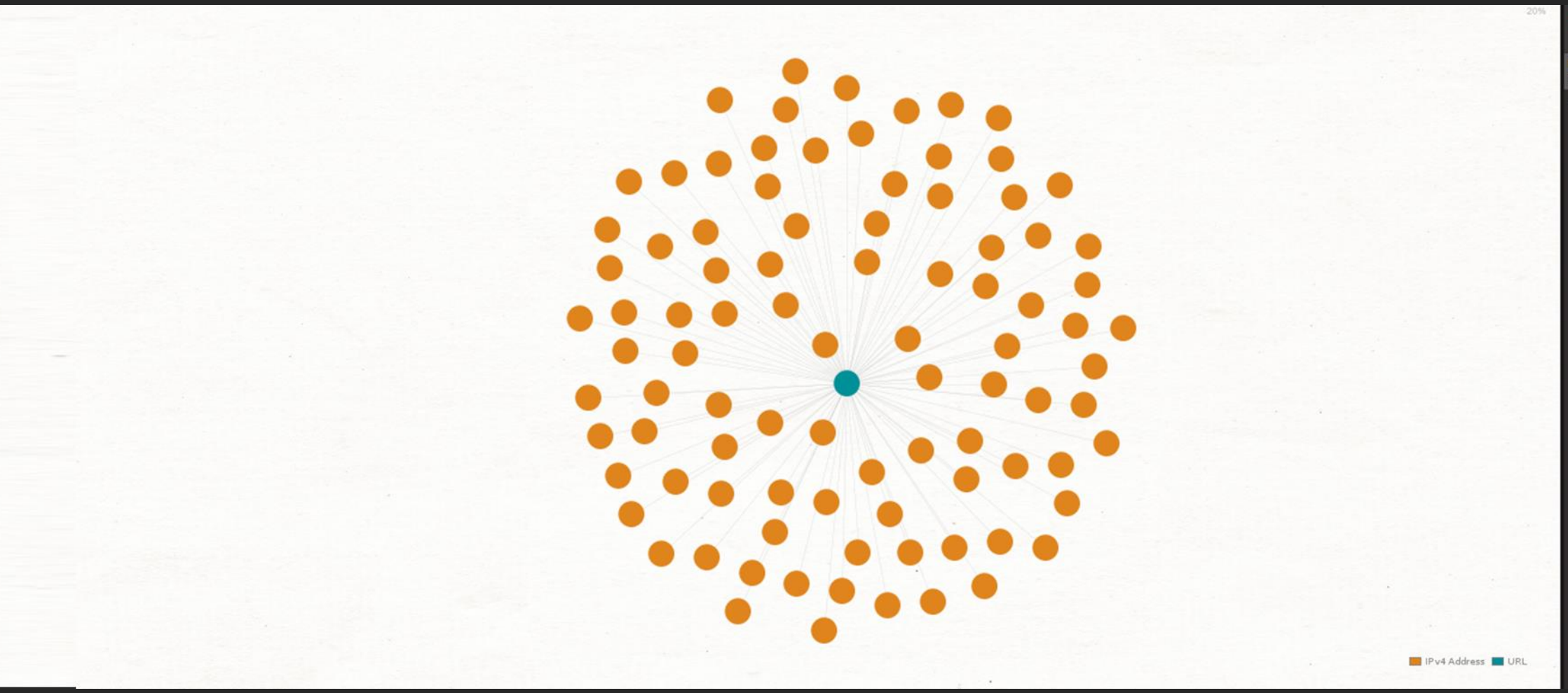
DOCUMENTATION: HTTPS://DOCS.PATERVA.COM/EN/DEVELOPER-PORTAL/MACHINES



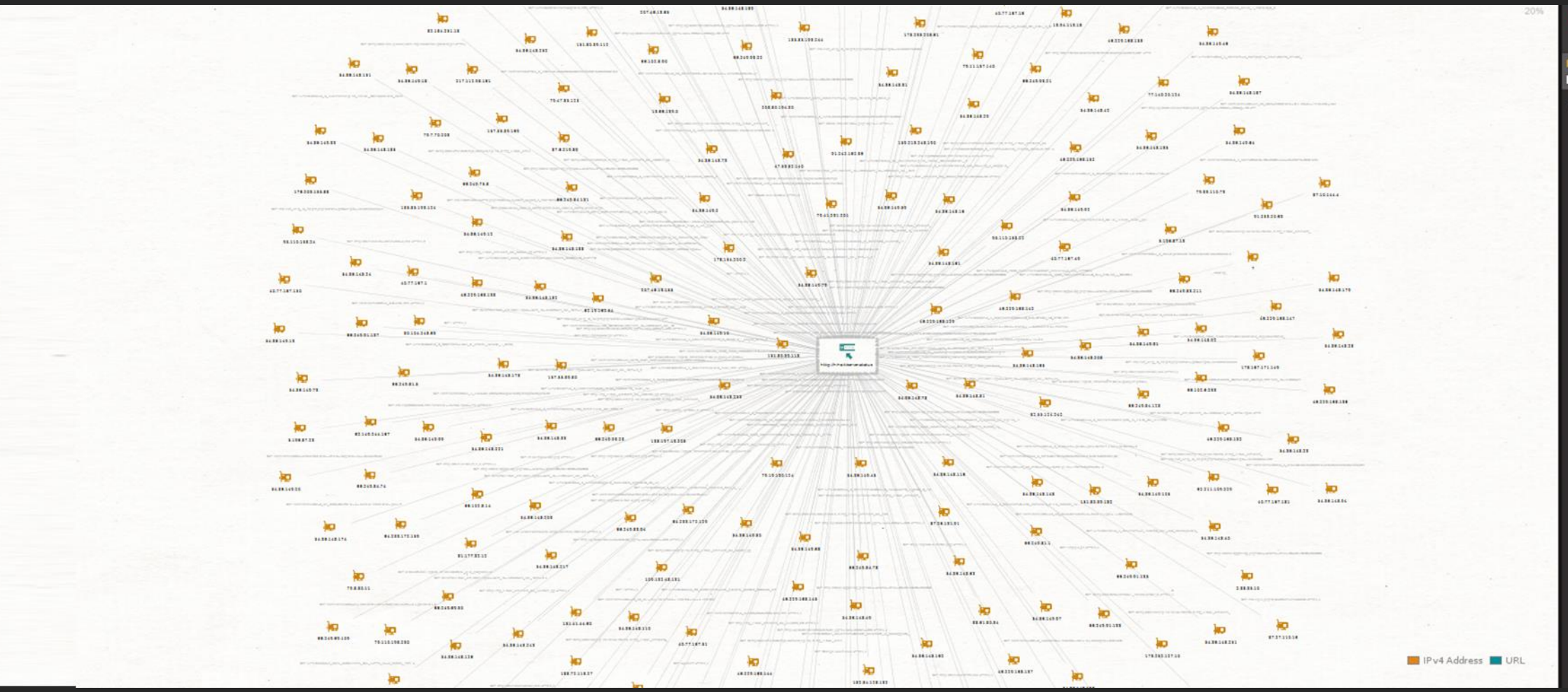
EXAMPLE: GETTING HTTP CLIENTS BY REQUESTING A C2 WITH APACHE SERVER-STATUS MOD ENABLED



EXAMPLE: GETTING HTTP CLIENTS BY REQUESTING A C2 WITH APACHE SERVER-STATUS MOD ENABLED



EXAMPLE: GETTING HTTP CLIENTS BY REQUESTING A C2 WITH APACHE SERVER-STATUS MOD ENABLED



EXAMPLE: GETTING HTTP CLIENTS BY REQUESTING A C2 WITH APACHE SERVER-STATUS MOD ENABLED

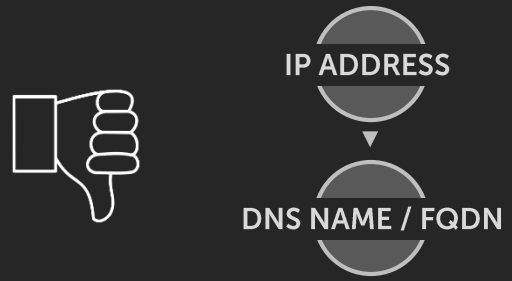
Dev. tips & tricks

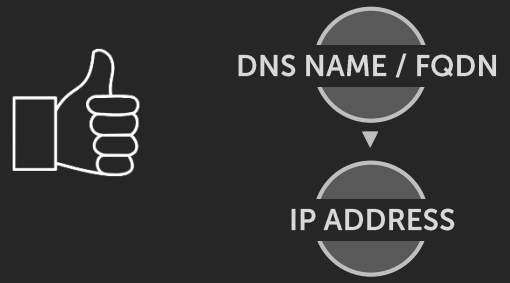
DEVELOPPER TIP #1

Think about link direction


IP ADDRESS

[ONLINE] IP Address to DNSNames


IP ADDRESS
DNS NAME / FQDN

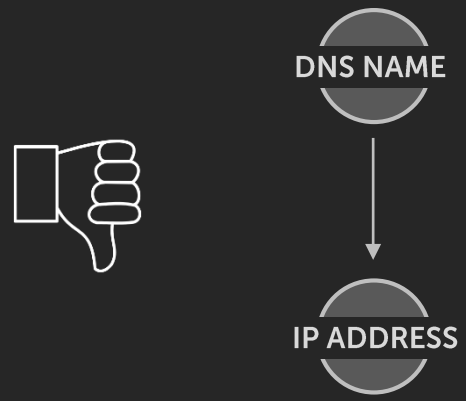

DNS NAME / FQDN
IP ADDRESS

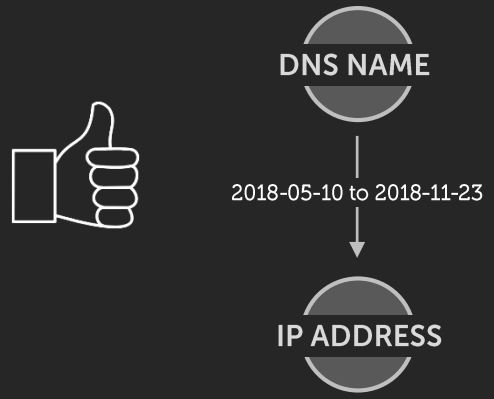
```
t = TransformLib.MaltegoTransform()
e = TransformLib.MaltegoEntity()

e.setType("maltego.DNSName")
e.setValue(domain)
e.addAdditionalFields('link#maltego.link.direction',
                      'link#maltego.link.direction',
                      'loose',
                      'output-to-input')

t.addEntityToMessage(e)
t.returnOutput()
```

DEVELOPPER TIP #2

Think about link label

[ONLINE] IP Address to DNSNames

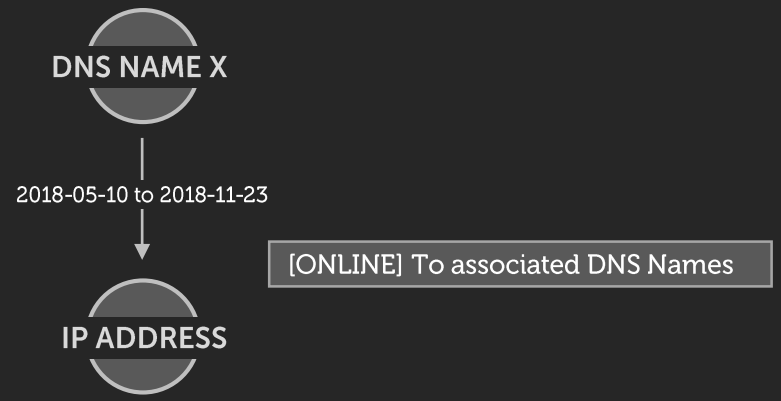

DNS NAME
IP ADDRESS

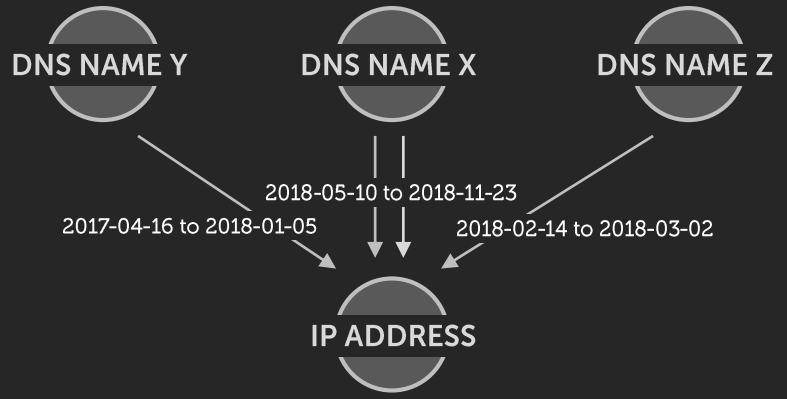

DNS NAME
2018-05-10 to 2018-11-23
IP ADDRESS

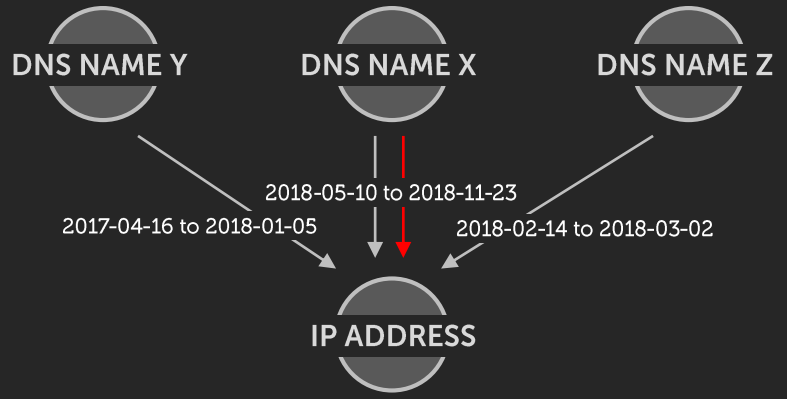
```
t = TransformLib.MaltegoTransform()
e = TransformLib.MaltegoEntity()

e.setType("maltego.DNSName")
e.setValue(domain)
e.setLinkLabel("%s - %s" % (first_seen, last_seen))
e.addAdditionalFields('link#maltego.link.direction',
                      'link#maltego.link.direction',
                      'loose',
                      'output-to-input')

t.addEntityToMessage(e)
t.returnOutput()
```

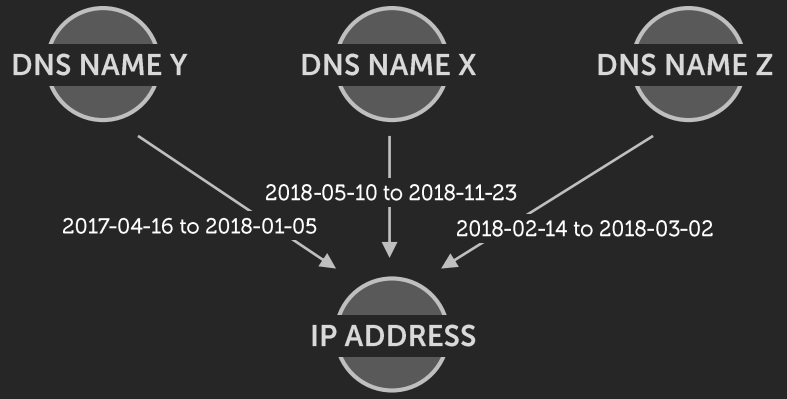
DEVELOPPER TIP #3

Avoid link loopback.


DNS NAME X
2018-05-10 to 2018-11-23
IP ADDRESS
[ONLINE] To associated DNS Names


DNS NAME Y
DNS NAME X
DNS NAME Z
2018-05-10 to 2018-11-23
2017-04-16 to 2018-01-05
2018-02-14 to 2018-03-02
IP ADDRESS

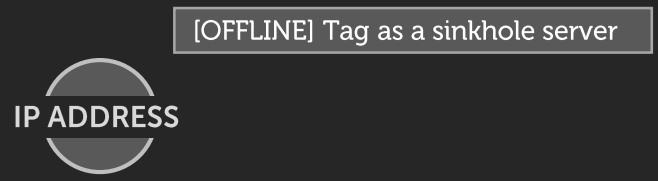

DNS NAME Y
DNS NAME X
DNS NAME Z
2018-05-10 to 2018-11-23
2017-04-16 to 2018-01-05
2018-02-14 to 2018-03-02
IP ADDRESS

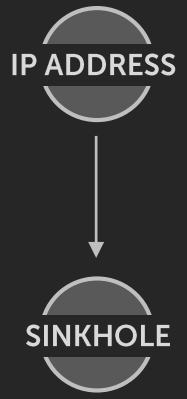

DNS NAME Y
DNS NAME X
DNS NAME Z
2018-05-10 to 2018-11-23
2017-04-16 to 2018-01-05
2018-02-14 to 2018-03-02
IP ADDRESS

```python
def additional_fields_to_dict():
    """
        Translate additional fields to dict.
    """
    rtn = {"parent.entity.value" : ""}
    for i, field in enumerate(sys.argv[3].split("#")):
        field = field.split("=")
        if i == 0:
            rtn["entity.value"] = field[1]
        else:
            rtn[field[0]] = field[1]
    return rtn

parent_fields = additional_fields_to_dict()

if parent_fields["parent.entity.value"] != entity_value:
    e = TransformLib.MaltegoEntity()
    e.setType("maltego.Entity")
    e.addAdditionalFields(fieldName="parent.entity.value",
                          displayName="Parent Entity",
                          value=entity_value)
    e.setValue(entity_value)
    transform.addEntityToMessage(e)
```

DEVELOPPER TIP #4

Use transforms to push data
to your databases.


[OFFLINE] Tag as a sinkhole server
IP ADDRESS


IP ADDRESS
SINKHOLE

Few months later...


[OFFLINE] Qualify this IP Address
IP ADDRESS

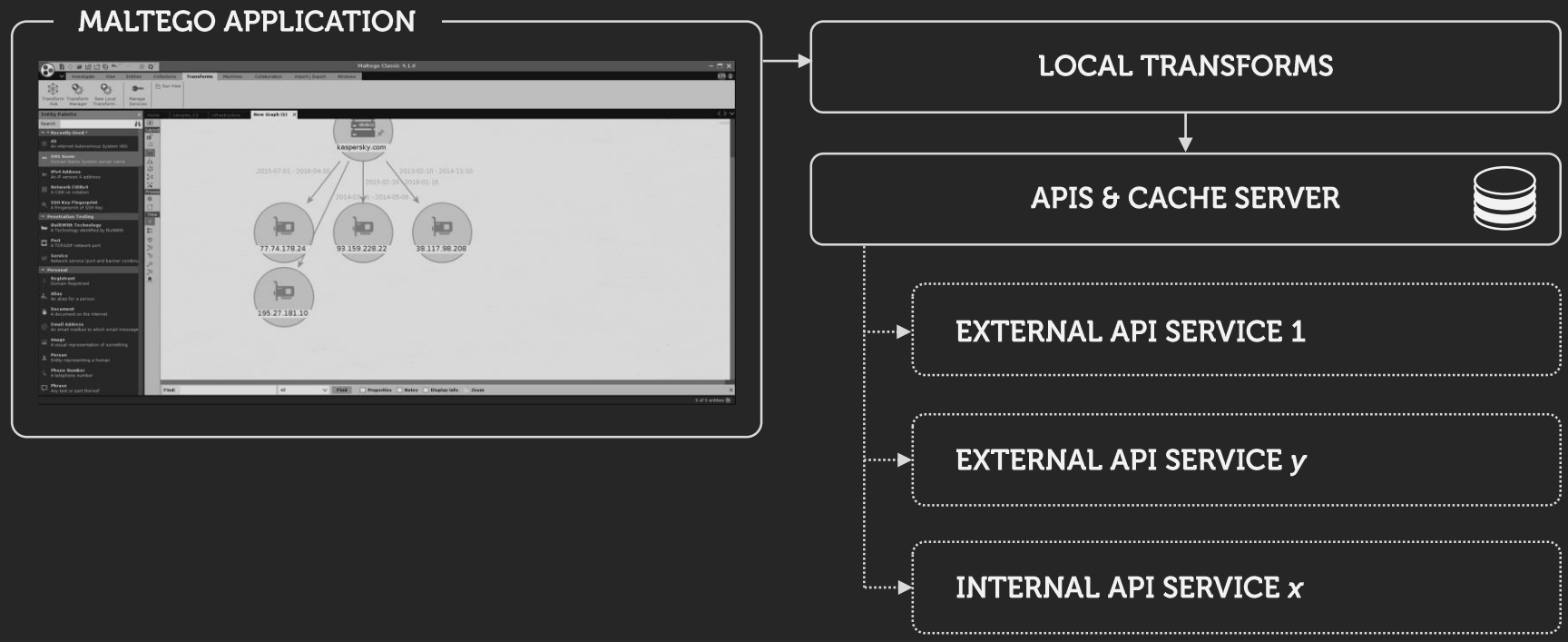

IP ADDRESS
SINKHOLE

DEVELOPPER TIP #5

Think about additional fields and notes.

DEVELOPPER TIP #5

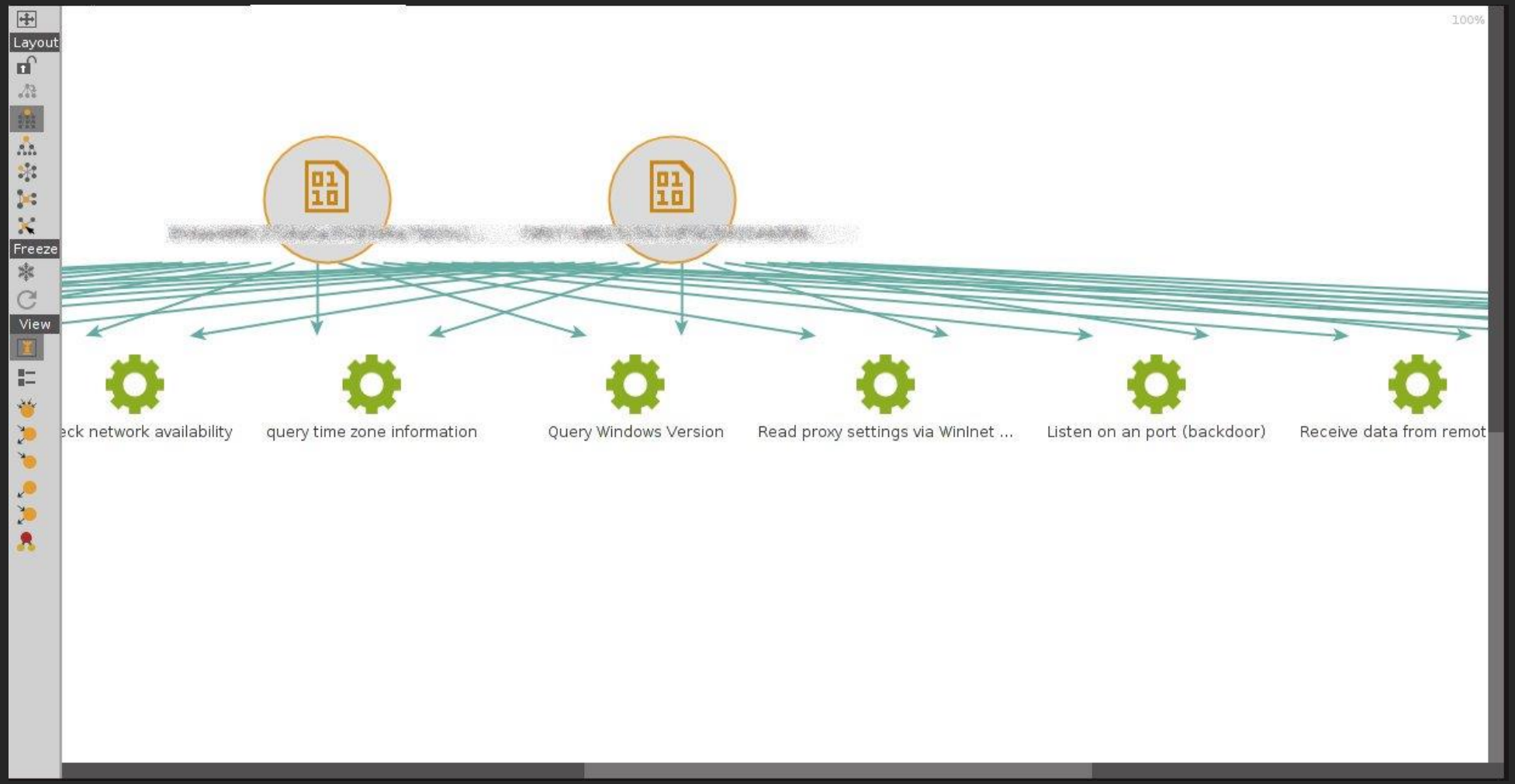
Use a multi-layered architecture.



EXAMPLE: SOME « HOME MADE » INFRASTRUCTURE

DEVELOPPER TIP #5

Don’t hesitate to be creative and integrate your own databases, applications and tools.



EXAMPLE: GETTING FEATURES FROM MALWARES HASHES (IDAscope DB & VTI)

End.