- Showcase
  - Use Cases ━
    - **Use Cases**
    - **Network footprinting with Maltego**
  - Case studies ✚

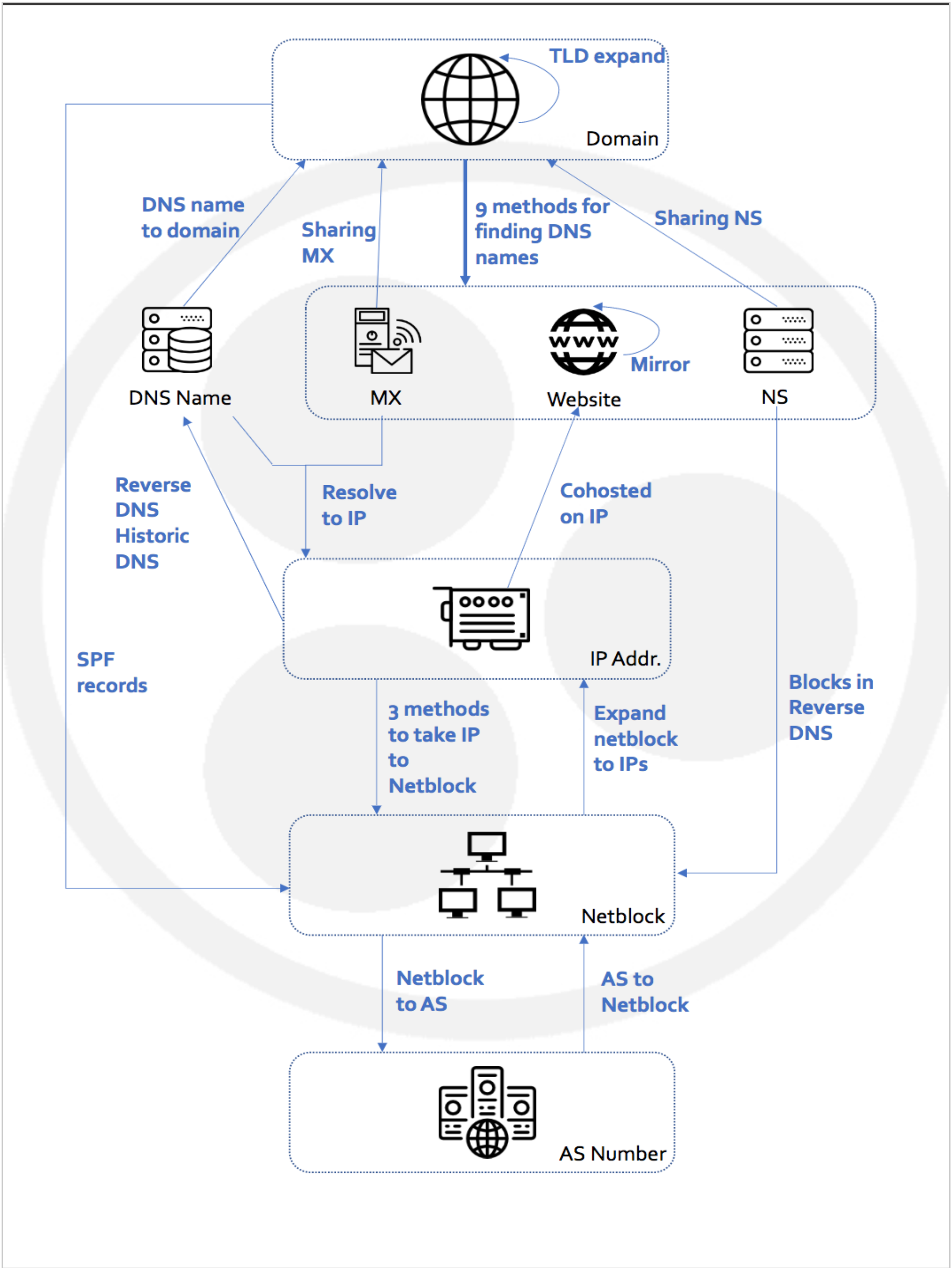
≡ Open navigation

One common task that Maltego is used for is doing infrastructure footprints on an organisation's network. This post will detail a possible methodology used for network footprints as well as demonstrate how they can be performed in Maltego. Finally the post will show how the process is drastically simplified with the use of machines that automates the process of running transforms in Maltego.

NETWORK FOOTPRINTING METHODOLOGY

When performing a footprint on a domain the goal is to find as much information about the domain as possible on an infrastructure level. When dealing with a large footprint it can be quite difficult to know when you have found all possible information that is publicly available for that particular domain. To make the process a little easier we have a structured methodology that we follow when conducting a network footprint in Maltego. This process is outlined in the data model in Image 1 below.

At each level of this data model we want to find as much information as possible relating to the domain in question. Arrows in the data model relates to transforms within Maltego that can be used to find related information either above, below or on the same level of the model. Throughout this blog post I will refer back to this data model. Starting at the top of the model with the target domain you'll see an arrow that points from a domain back to a domain. This transform relates to the TLD (top level domain) expansion of the target. In the real world this means going from (for example) google.com to all the other Google domains (google.net, google.co.uk etc.)



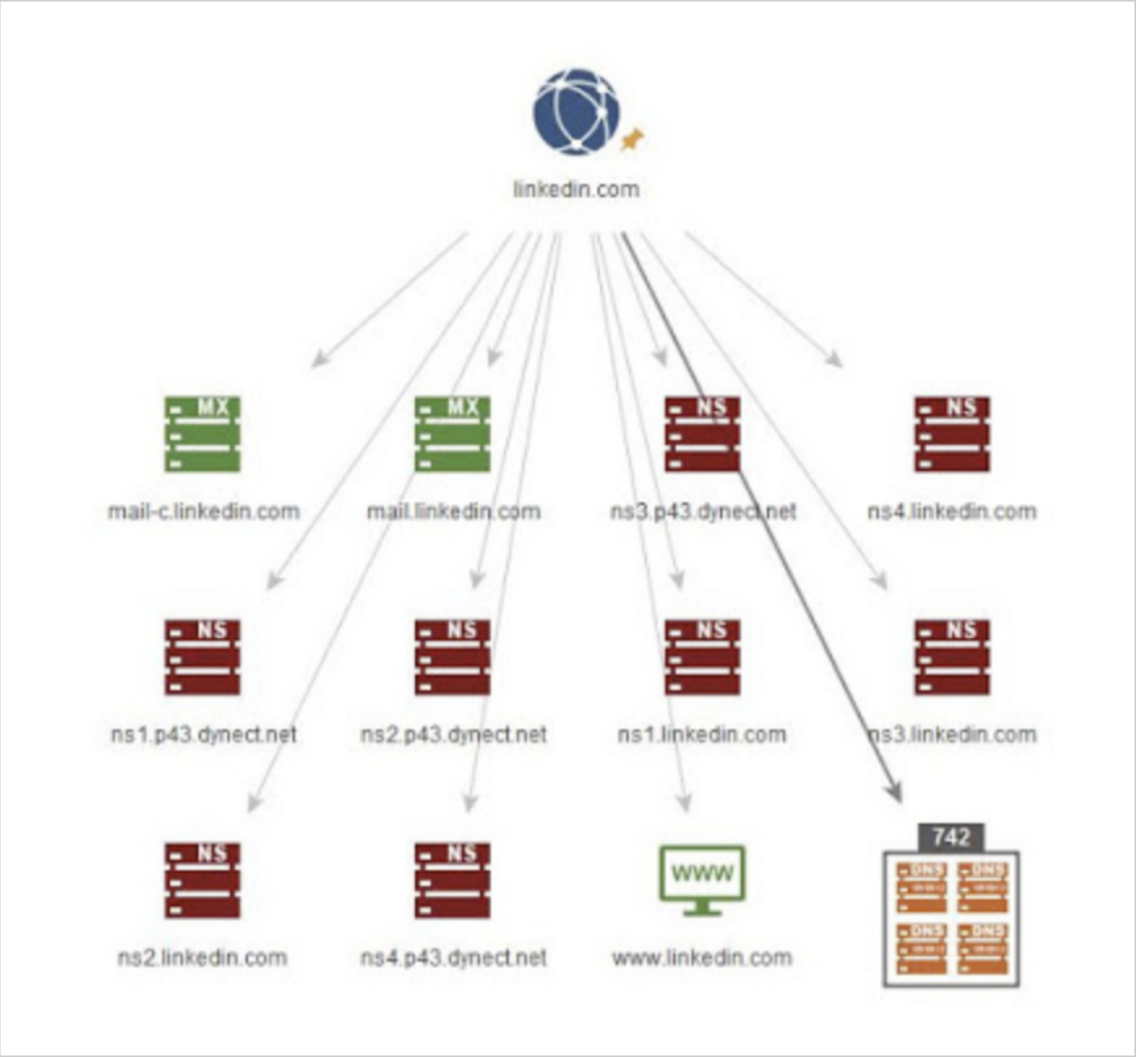
(https://s3-eu-central-1.amazonaws.com/euc-cdn.freshdesk.com/data/helpdesk/attachments/production/15005597516/original/jqalfEHK46FGi_RPJ3gP4H8oXlNgUH8QXQ.png?1532706113)

Image 1

Once the top level domains are enumerated the first step is to try find as many DNS names from that domain's zone file. This includes getting the domain's **MX record** (https://en.wikipedia.org/wiki/MX_record)s, its **NS records** (https://en.wikipedia.org/wiki/Name_server)
and as many A records as possible. In Maltego there are nine transforms for finding DNS names related to a domain. Explaining how each of these transforms work is out of the scope of this post, however, transform explanations can be found

in our transforms guide. In Maltego there is also a transform set named DNS from Domain that includes all nine of these transforms. Running this transform set on the domain linkedin.com results in the graph shown in Image 2 below.

Note that there are 742 entities in the DNS name collection node.From the DNS name level on the data model back in Image 1 you will see that there are three transforms for going back up a level from DNS names to find more related domains. Two of these transforms look for domains that share the same name servers (NS) or mails servers (MX) that have been found from our original domain. The third transform simply extracts the domain from that DNS name.

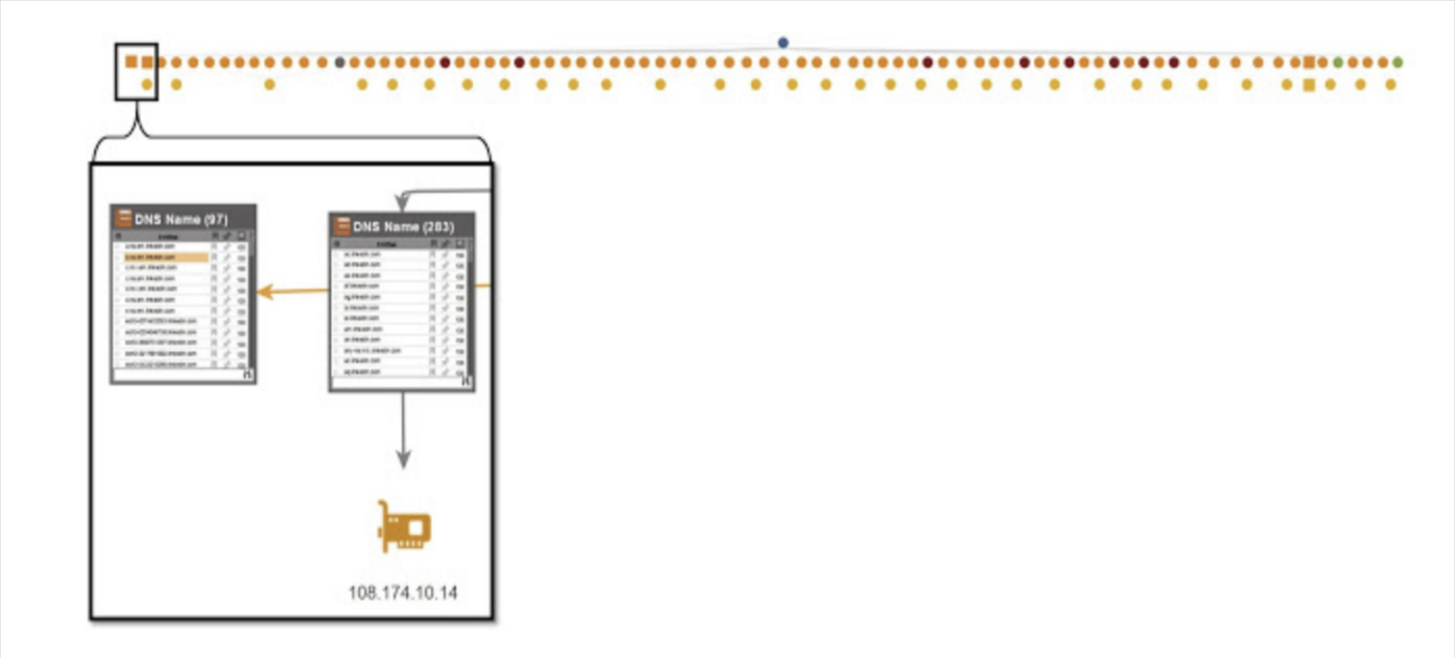


(https://s3-eu-central-1.amazonaws.com/euc-cdn.freshdesk.com/data/helpdesk/attachments/production/15005595796/original/uGaRwaOCln-E7xzjB6UIvGRO0FW2S5mx0A.png?1532702782)

Image 2

When finding shared infrastructure it is important to consider whether the name servers and mail servers are hosted by your target organisation or by an ISP. Looking at the shared infrastructure belonging to an ISP will results in many domains being returned that are hosted by the ISP but not related to your target. Determining if a MX or NS is hosted can be tricky but visiting the website of the related entity mostly helps in making that decision. It is outside the scope of this document to detail this process (but it's mostly just common sense). The next step in going down the data model is to resolve all the DNS names to IP addresses.

Doing so results in the graph the below:



(https://s3-eu-central-1.amazonaws.com/euc-cdn.freshdesk.com/data/helpdesk/attachments/production/15005595952/original/OkaOjJ93wWmTmEpcxyLY9ZX5GvlHsgJrfw.png?1532703086)

Image 3

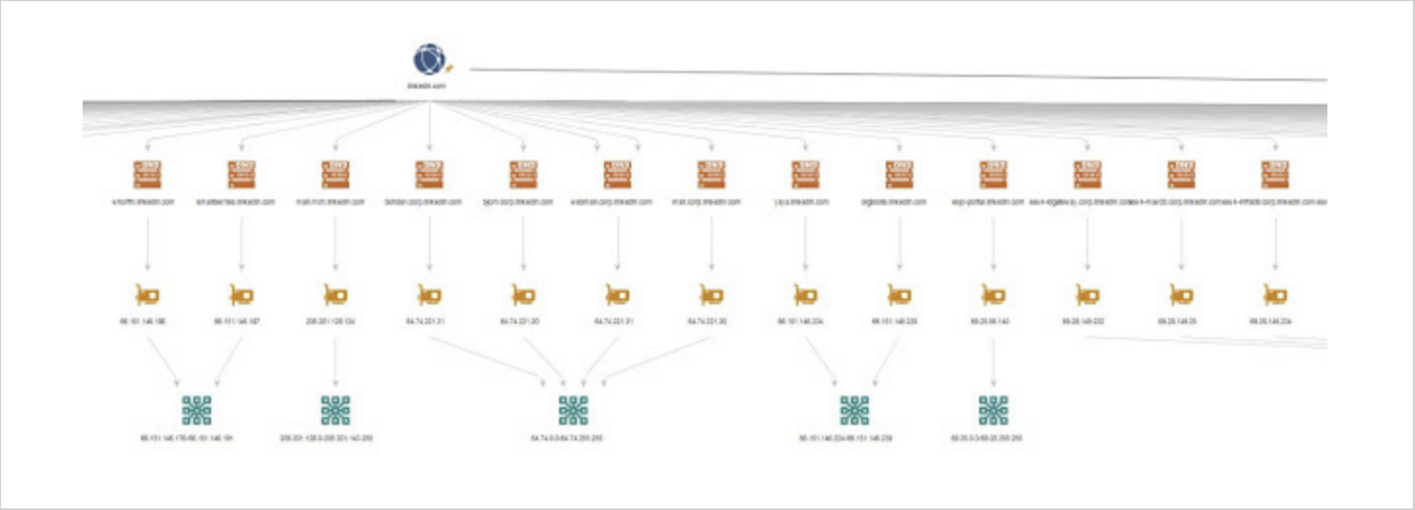
It is interesting to note here that 283 of the DNS names that we found all resolve to a single IP address shown in Image 3 above. From the image it can also be noted that there were 97 DNS names that currently do not resolve to an IP address at all. This might be an indication of old DNS names or DNS names that resolve to internal resources configured on a split DNS system.On the IP address layer of the data model we could now go back up a level to find more DNS names related to the IP addresses. This can be done by looking at historical DNS records collected from passive DNS, reverse DNS and by querying Bing to see what other website have been seen resolving to the same IP address (aka the "IP:" trick).

Continuing down the data model from the IP addresses we next want to find the netblocks that the addresses belong to and determine whether the entire netblocks actually belongs to our target organisation. Finding the correct netblock size can be a tricky process and often requires some trial and error to get right. In Maltego there are three transforms for finding netblocks from an IP address and it is important to understand how each of these work. These three transforms are listed below:

- To Netblocks [Using natural boundaries] - This transform will sort IP addresses into netblock sizes specified by the user.
- To Netblocks [Using routing info] - This transform determines the netblock that an IP address belongs to by looking up its routing table information.
- To Netblocks [Using WHOIS info] - This transform will look up the Netblock for an IP address by querying the registrars.

It is very important to place IP addresses into the correctly sized netblocks. If you make the block size too small you will miss out on IP space belonging to your target organisation. You also do not want to make the netblock too large and include IP space belonging to someone else.

Running the transform To Netblocks [Using WHOIS info] on our example graph from Image 3 results in the following graph:



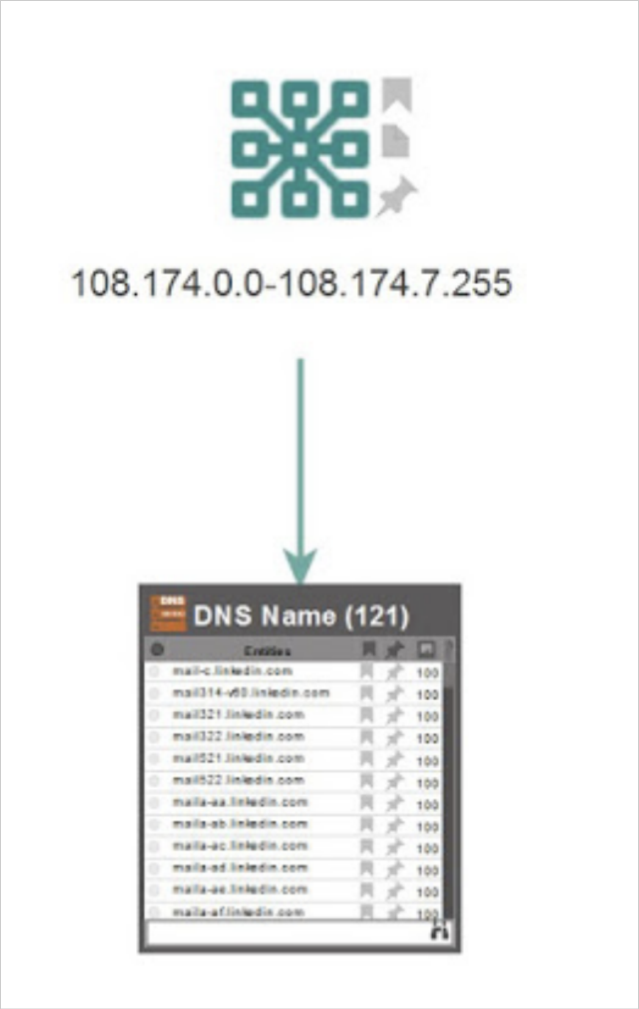
[(https://s3-eu-central-1.amazonaws.com/euc-cdn.freshdesk.com/data/helpdesk/attachments/production/15005596064/original/sRkrfU_O24keK-LvwSuf6RIDdx-tT60c-A.png?1532703228)](https://s3-eu-central-1.amazonaws.com/euc-cdn.freshdesk.com/data/helpdesk/attachments/production/15005596064/original/sRkrfU_O24keK-LvwSuf6RIDdx-tT60c-A.png?1532703228)

Image 4

Image 4 above shows a portion of the resulting graph. Once we have these netblocks it is important to validate that we are still looking at our target's infrastructure and have not included IP space belonging to "innocent bystanding" organisations.

One way of doing this is to run the historical DNS transform on the netblock and then manually inspect whether or not the block belongs to your target by looking at the (reverse) DNS names that you get back. This is done by running the transform To DNS names in netblock [reverse DNS]. Running this transforms on the netblock 108.174.0.0-108.174.7.255 found previously results in 121 DNS names being returned.

Manually inspecting these DNS names it is quite easy to see that they all do belong to our target organisation and we can therefore make the assumption with near certainty that the entire netblock does in fact belong to our target. In this step we have also found more DNS names related to our target and the process can be repeated by resolving the newly found DNS names to IP addresses and then finding the netblock that they belong to.

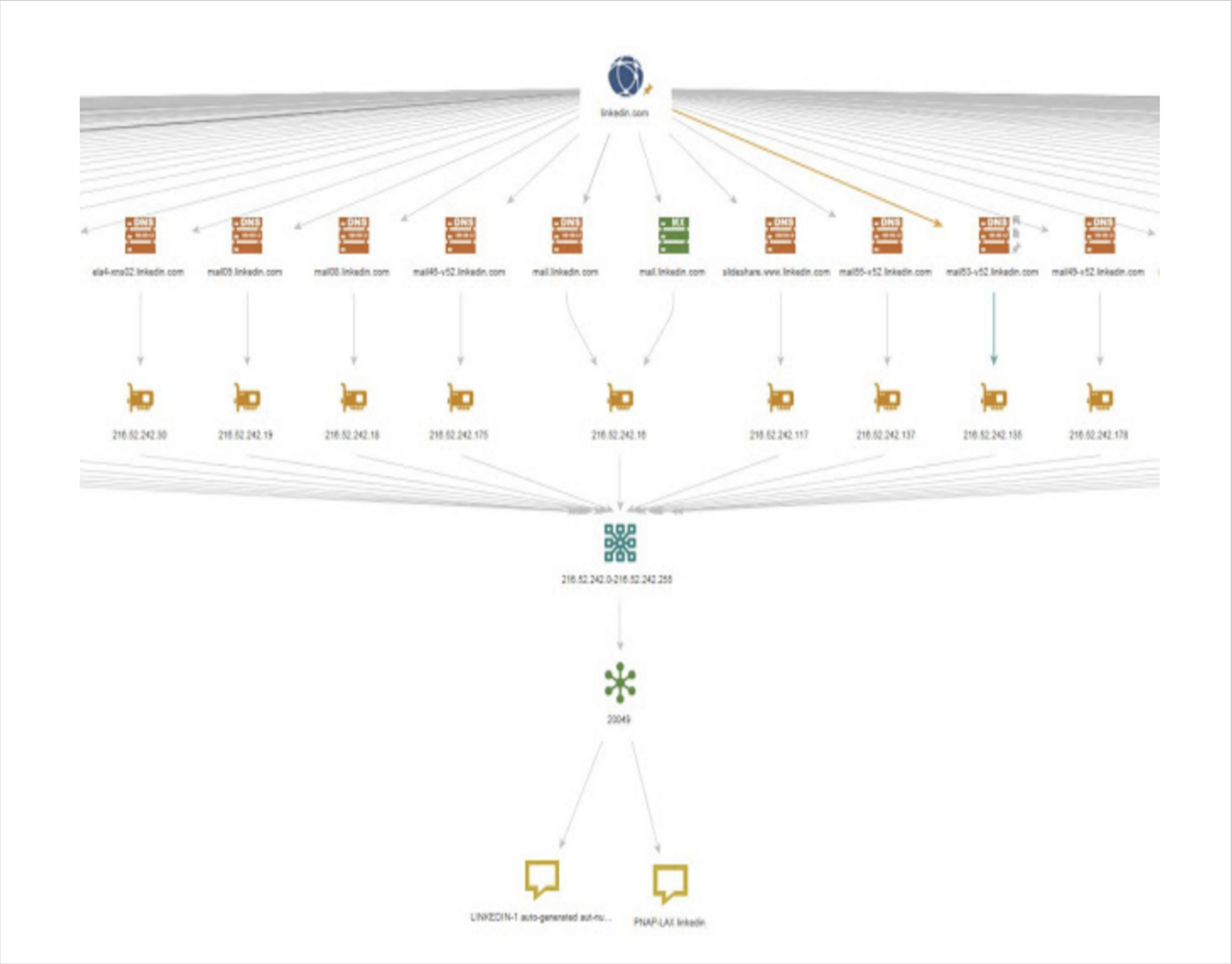


Entities	
mail-c.linkedin.com	100
mail314-v60.linkedin.com	100
mail321.linkedin.com	100
mail322.linkedin.com	100
mail521.linkedin.com	100
mail522.linkedin.com	100
maila-aa.linkedin.com	100
maila-ab.linkedin.com	100
maila-ac.linkedin.com	100
maila-ad.linkedin.com	100
maila-ae.linkedin.com	100
maila-af.linkedin.com	100

[(https://s3-eu-central-1.amazonaws.com/euc-cdn.freshdesk.com/data/helpdesk/attachments/production/15005596138/original/q9PL5J7r4eC-3Aqw4TTxK4IFFuuqqdyikg.png?1532703367)](https://s3-eu-central-1.amazonaws.com/euc-cdn.freshdesk.com/data/helpdesk/attachments/production/15005596138/original/q9PL5J7r4eC-3Aqw4TTxK4IFFuuqqdyikg.png?1532703367)

Image 5

Next from the netblocks we have found we can have a look at the Autonomous Systems (AS-es) that they belong to and determine whether the entire AS is in fact owned by our target organisation. First we run the transform To AS number on the netblocks we have. We then run the transform To Company [Owner] to see who owns the AS. Doing so on our example results in seven AS-es being returned that belong to the LinkedIn organisation. Image 6 below shows a small portion of the graph and the path taken to get to one of these AS-es:

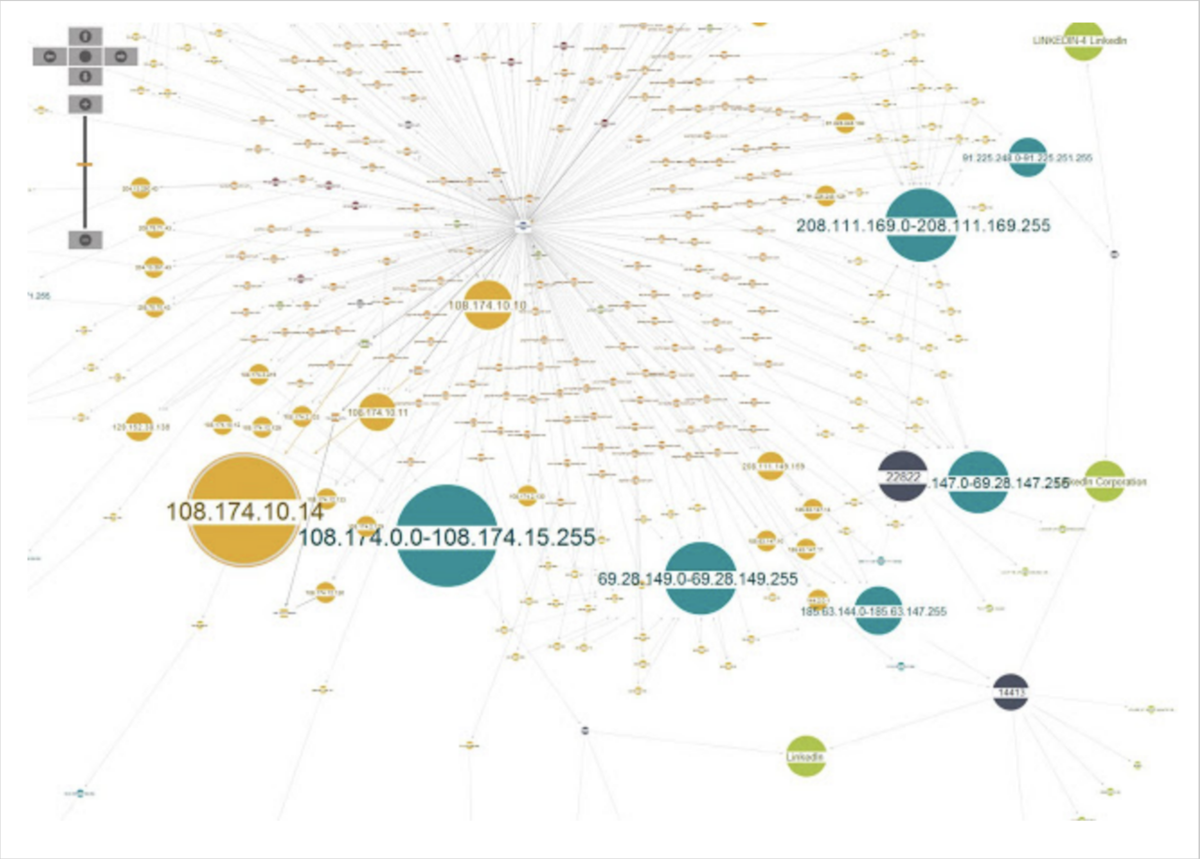


(https://s3-eu-central-1.amazonaws.com/euc-cdn.freshdesk.com/data/helpdesk/attachments/production/15005596234/original/4O9BILatxysm79EBM8IOK2T5qmARSC0rEA.png?1532703560)

Image 6

At this point we have reached the bottom level of the data model from Image 1. The next step would be to take the AS-es we have found belonging to our target organisation and start moving back up the data model to find more related information at each level. First we would get all the netblocks in the AS-es and from these new netblocks we would then find more DNS names by looking at their historical or reverse DNS records. From new DNS names that are found we could potentially find more domains belonging to the target and then start the whole process again on the new domains. Note that this step is not included on the example graph in this post.

An important aspect to realize here is that a network footprint is a cyclical process, not a linear one (and you're never done, you just give up ;)) . The most simple footprint you can do would be to go from the top of the data model to the bottom without moving up the model at any stage as we have done here in this example. However we could continue this footprint by moving back up the data model from the AS-es that we have found belonging to our target.



[(https://s3-eu-central-1.amazonaws.com/euc-cdn.freshdesk.com/data/helpdesk/attachments/production/15005596278/original/X4xWZXcEhffJhRplNNOXKWTsLRf-uMcpEw.png?1532703643)](https://s3-eu-central-1.amazonaws.com/euc-cdn.freshdesk.com/data/helpdesk/attachments/production/15005596278/original/X4xWZXcEhffJhRplNNOXKWTsLRf-uMcpEw.png?1532703643)

Image 7

The final graph from our example in bubble view is shown in Image 7 above. Bubble view will size entities according to the number of incoming links it has from different sources. This makes it easy to identify the most connected parts of the network as well as its outliers.

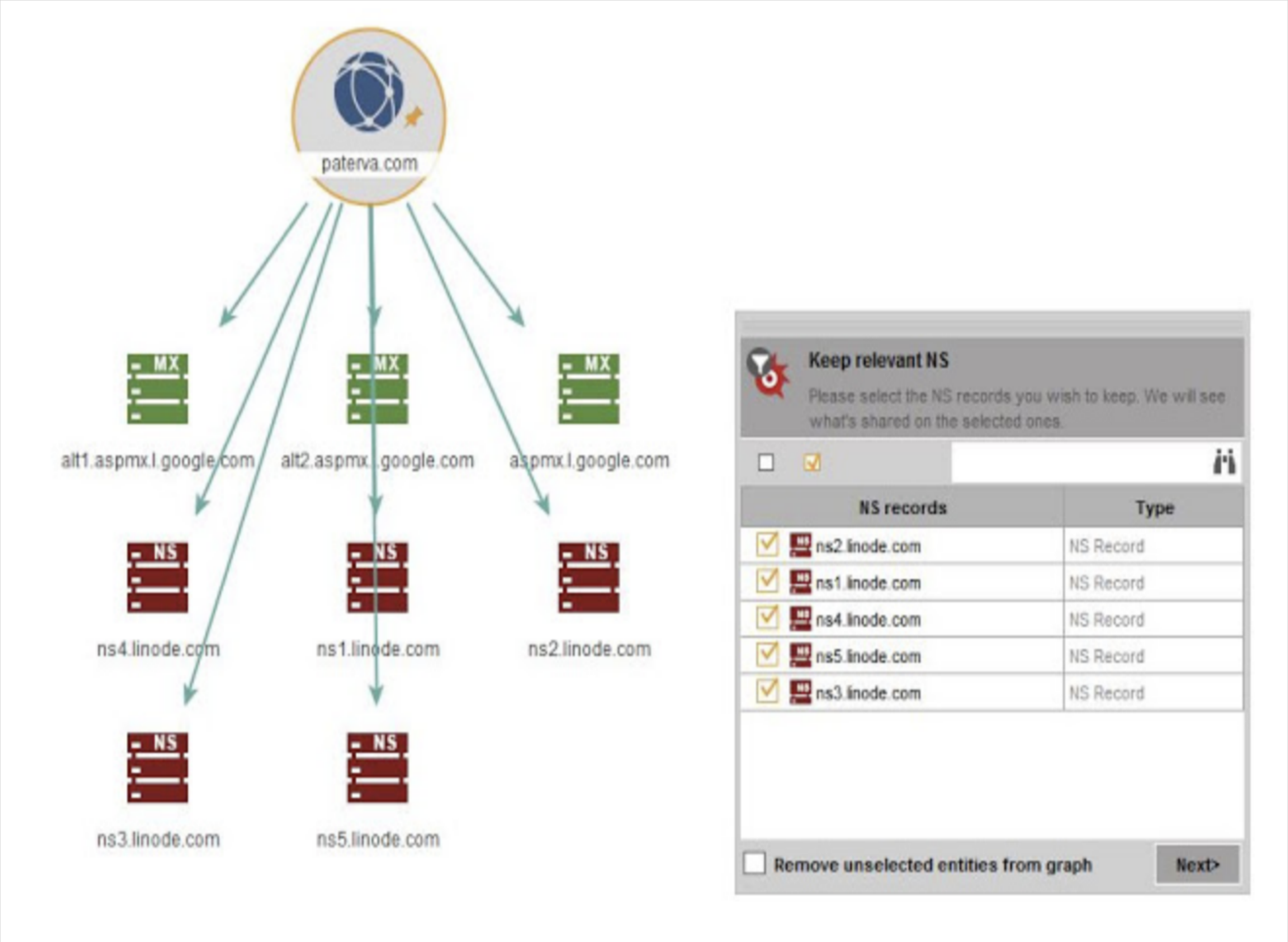
FOOTPRINTING MACHINES

Fortunately, it is not required to remember every step of this footprinting process thanks to the concept of machines in Maltego. Machines allow you to script transforms together and have them run sequentially in an automated fashion. Out-the-box Maltego comes with three machines for network footprinting that roughly follow the process described previously. These three machines are described briefly below. Note that Maltego also ships with a forth machine for footprinting named Footprint XXL. Footprint XXL uses a different method which is useful when footprinting larger networks. However this machine is not within the scope of this blog post as it is aimed at advanced Maltego users footprinting massive multi-national organizations.

Footprint L1:
This is the most basic footprinting machine and runs through the data model from Image 1 straight down from top to bottom without looking at any shared infrastructure or historical DNS records.

FOOTPRINT L2:

This machine will run through the same steps as Footprint L1 above. Additionally this machine will look for additional domains related to the original domain by looking for shared infrastructure of its name servers (NS) and mail servers (MX). The machine will also look for other websites hosted on the same IP addresses. The machine also has user filters - these are popups which are displayed while the machine is running and prompts the user to manually inspected results and decides with ones to continue with. In machine L2's case user filters are used to allow the user to choose which name servers, mail servers and websites are hosted by the target organisation or by an ISP. This is done to prevent the machine from looking for shared infrastructure on DNS names that are not hosted by the target. An example of a user filter when running Footprint L2 on the domain paterva.com is shown in the Image 8 below.



Keep relevant NS

Please select the NS records you wish to keep. We will see what's shared on the selected ones.

NS records	Type
☑ ns2.linode.com	NS Record
☑ ns1.linode.com	NS Record
☑ ns4.linode.com	NS Record
☑ ns5.linode.com	NS Record
☑ ns3.linode.com	NS Record

□ Remove unselected entities from graph

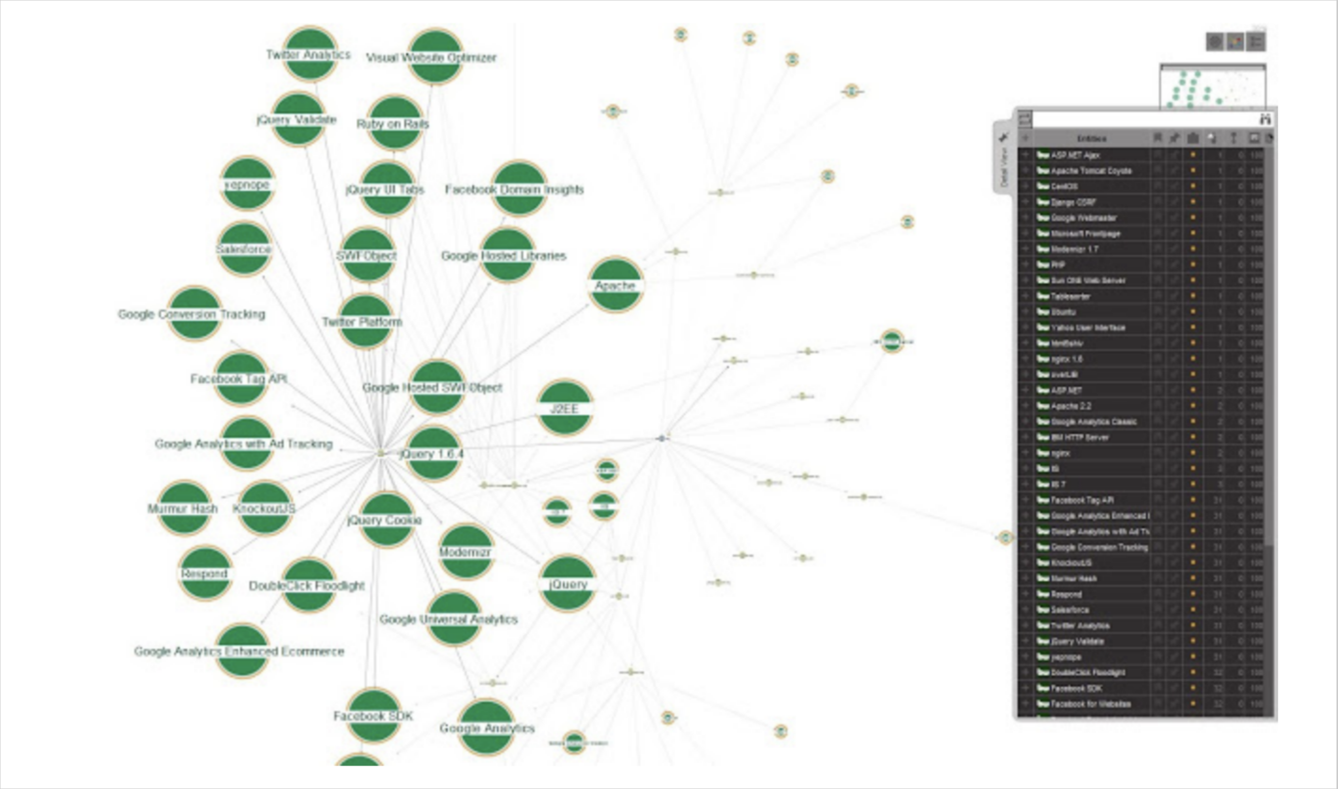
(https://s3-eu-central-1.amazonaws.com/euc-cdn.freshdesk.com/data/helpdesk/attachments/production/15005596592/original/3FwO9odZdYmn9zLM3pgfo_-mZ3mfjUdLOQ.png?1532704074)

Image 8

From visual inspection it is clear that Paterva's mail is hosted by Google and their name servers are hosted by Linode. Therefore you would not want the machine to continue to run transforms that look for shared infrastructure on these entities as you'll follow the rabbit hole all the way to Google's (and Linode's) infrastructure!

Footprint L3:

Footprint L3 runs the same transforms as Footprint L2 but additionally it will look at historical / reverse DNS records on the netblocks that are found in order to find additional DNS names belonging to the target. Again the machine will use user filters to allow the user to specify which of the netblocks are still relevant.Footprint L3 will also run a transform named ToServerTechnologyWebsite on selected website entities on the graph and returns the name of different server technologies that are used on that particular website. Running this transform provides an easy way to identify which technologies are used commonly across many of the target's websites as well as outliers - the (sometimes more outdated) technologies that are only be used on one or two servers.



Entities						
ASP.NET Ajax			●	1	0	100
Apache Tomcat Coyote			●	1	0	100
CentOS			●	1	0	100
Django CSRF			●	1	0	100
Google Webmaster			●	1	0	100
Microsoft Frontpage			●	1	0	100
Modernizr 1.7			●	1	0	100
PHP			●	1	0	100
Sun ONE Web Server			●	1	0	100
Tablesorter			●	1	0	100
Ubuntu			●	1	0	100
Yahoo User Interface			●	1	0	100
MediaElement			●	1	0	100
nginx 1.6			●	1	0	100
swfLib			●	1	0	100
ASP.NET			●	2	0	100
Apache 2.2			●	2	0	100
Google Analytics Classic			●	2	0	100
IBM HTTP Server			●	2	0	100
nginx			●	2	0	100
IIS			●	2	0	100
IIS 7			●	3	0	100
Facebook Tag API			●	31	0	100
Google Analytics Enhanced I			●	31	0	100
Google Analytics with Ad Tr			●	31	0	100
Google Conversion Tracking			●	31	0	100
KnockoutJS			●	31	0	100
Murmur Hash			●	31	0	100
Respond			●	31	0	100
Salesforce			●	31	0	100
Twitter Analytics			●	31	0	100
jQuery Validate			●	31	0	100
yepnope			●	31	0	100
DoubleClick Floodlight			●	32	0	100
Facebook SDK			●	32	0	100
Facebook for Websites			●	32	0	100

(https://s3-eu-central-1.amazonaws.com/euc-cdn.freshdesk.com/data/helpdesk/attachments/production/15005596624/original/3zkm71ufnAYQREvsH2m6cz2cZipCjl018A.png?1532704158)

Image 9

The screenshot in Image 9 above shows the results of the transform ToServerTechnologyWebsite on web servers of redcross.org.

51 websites are found that are related to the domain redcross.org and inspecting the graph you can identify the website technologies that are commonly used. Selecting all the BuiltWith entities on the graph and ordering the detail view in descending order according to the number of incoming links shows which website technologies are the 'odd-ones-out' and are only used on a couple of websites - these are often the more 'interesting' sites...

CONCLUSION

The examples shown in this blog post provides one possible strategy for conducting a network footprint in a structured and repeatable way. The three footprinting machines that come with Maltego out-the-box provide an easy method for applying this strategy to any domain while each machine differs in exploration depth.

🡐 **Previous: Use Cases** (https://docs.maltego.com/support/solutions/articles/15000012022-use-cases) **Next: Case studies > How to: Shodan** 🡒 (https://docs.maltego.com/support/solutions/articles/15000012021-how-to-shodan)

Search all Maltego Guides:

Table of contents

NETWORK FOOTPRINTING METHODOLOGY

^