Badger: The Networked Security State Estimation Toolkit

Edmond Rogers, Will Rogers, and Gabe Weaver

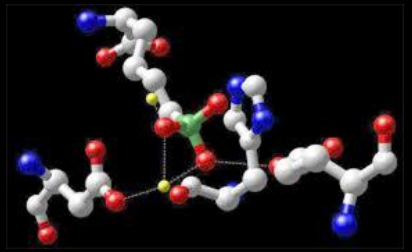
Badger42.org

A **target system network** is a tuple $N = (G, \Sigma, \Delta, T, I)$

- $G = (V, E)$ is a graph
- The graph has type information $T = (T_V, T_E, T_G)$
  - Σ is an alphabet of vertex types
  - Δ is an alphabet of edge types
  - $T_V : V \rightarrow \Sigma$ is a function that maps $v \in V$ to $\sigma \in \Sigma$
  - $T_E : E \rightarrow \Delta$ is a function that maps $e \in E$ to $\delta \in \Delta$
  - $T_G : \Sigma \times \Delta \rightarrow \mathit{String}$ defines the type of the graph
- The graph has identifier information
  - $id_V(V) : V \rightarrow \mathit{String}$ is a function that maps a vertex to an identifier
  - $id_E(E) : E \rightarrow \mathit{String}$ is a function that maps an edge to an identifier
  - $id_G(G) : G \rightarrow \mathit{String}$ is a function that maps the graph to an identifier.

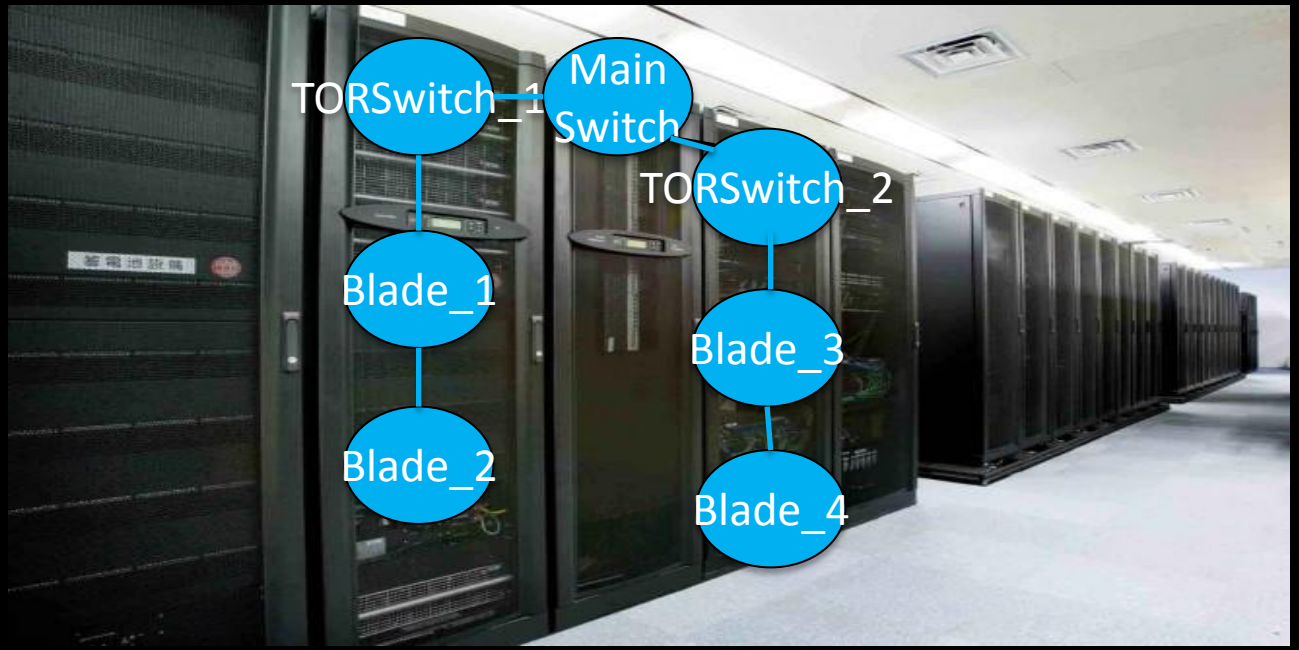
- **Human-readable graphical language**

- **Registry of vertex and edge types**



- **Join atomic values among disparate data sources**

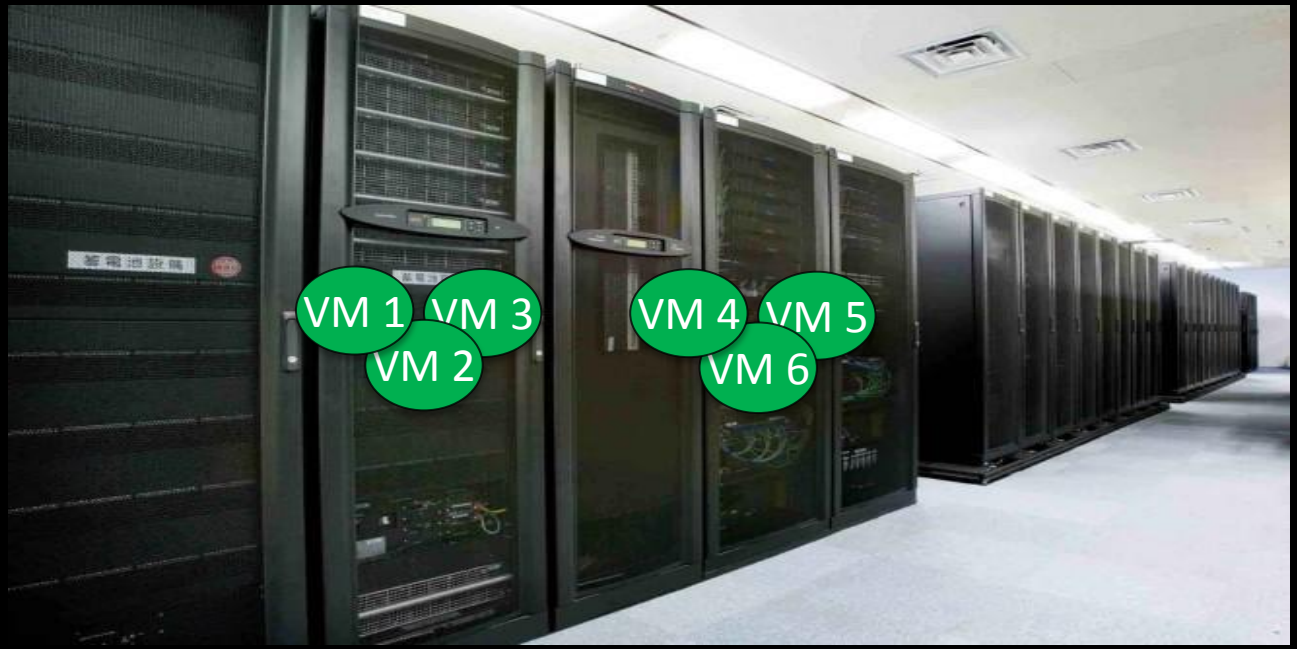
urn:cptl:cloud:cloudspace.network



Cloud Infrastructure Provider (Cloudspace)

Badger42.org

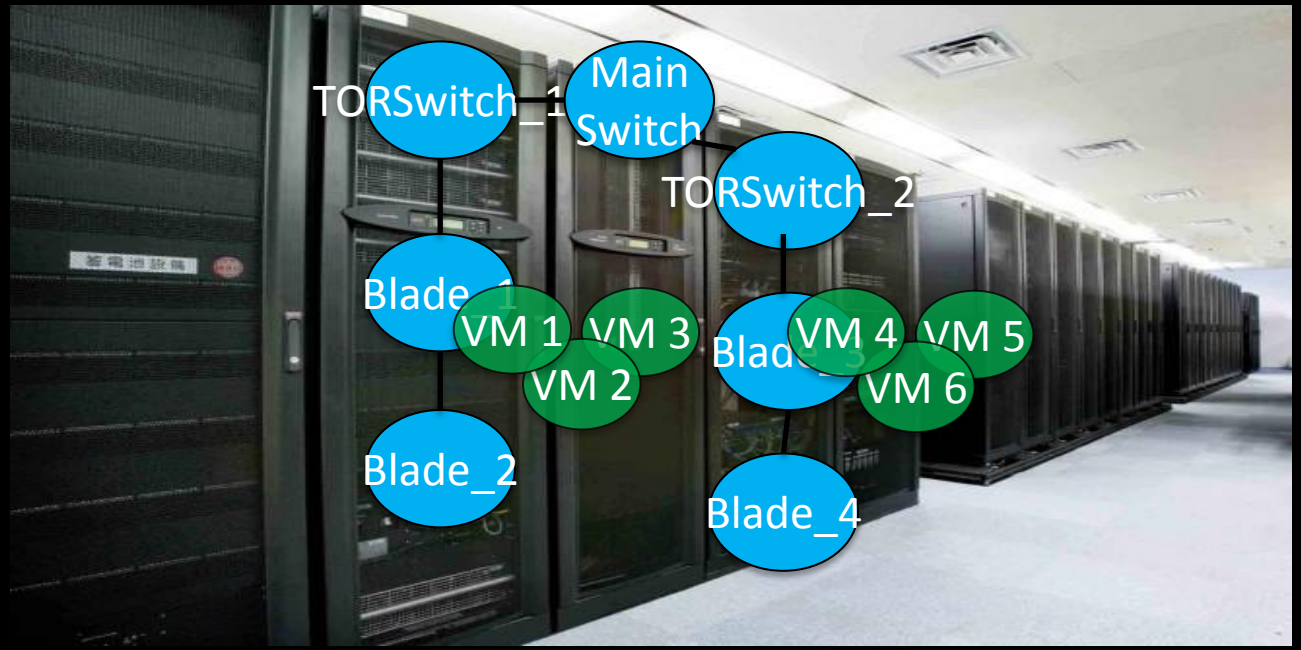
urn:cptl:cloud:streampics.network



Cloud Service Provider (Streampics)

Badger42.org

urn:cptl:cloud:cloudspace-streampics.network



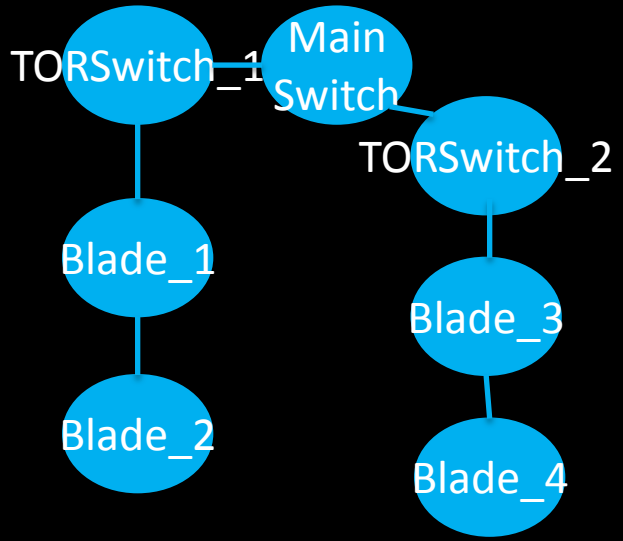
Cloud Service Provider (Streampics)

Cloud Infrastructure Provider (Cloudspace)

Badger42.org

Serialize to GraphML

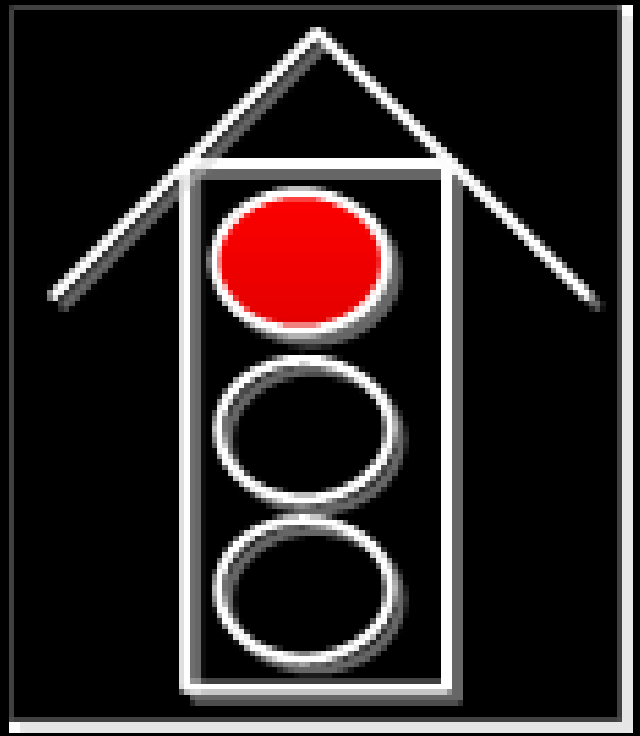
urn:cptl:cloud:cloudspace.network




```
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <key id="label"  for="node" attr.name="label" attr.type="string"/>
  <key id="type"   for="node" attr.name="type"  attr.type="string"/>
  <key id="etype"  for="edge" attr.name="type"  attr.type="string"/>
  <graph id="urn:cptl:cloud:cloudspace.network"
         edgedefault="undirected">
    <!-- We obtain full context of id by concating with containing graph -->
    <node id="MainSwitch_1">
      <data key="label">Main Switch 1</data>
      <data key="type">Switch</data>
    </node>
    <node id="TORSwitch_1">
      <data key="label">Top Of Rack Switch 1</data>
      <data key="type">Switch</data>
    </node>
    <node id="TORSwitch_2">
      <data key="label">Top Of Rack Switch 2</data>
      <data key="type">Switch</data>
    </node>
    <node id="Blade_1">
      <data key="label">Server Blade 1</data>
      <data key="type">Blade</data>
    </node>
    <node id="Blade_2">
      <data key="label">Server Blade 2</data>
      <data key="type">Blade</data>
    </node>
    <node id="Blade_3">
      <data key="label">Server Blade 3</data>
      <data key="type">Blade</data>
    </node>
    <node id="Blade_4">
      <data key="label">Server Blade 4</data>
      <data key="type">Blade</data>
    </node>
    <node id="Blade_5">
      <data key="label">Server Blade 5</data>
      <data key="type">Blade</data>
    </node>
```

Badger42.org

- Red == Bad
- Yellow == Caution
- Green == It’s all good



Badger42.org

Networked Security State Estimation

- Measurement of the state of security
- Somewhat polarizing
- Take it for what its worth . . .

Badger42.org

URN:CPTLID:CAPABILITY:ATTRIBUTE

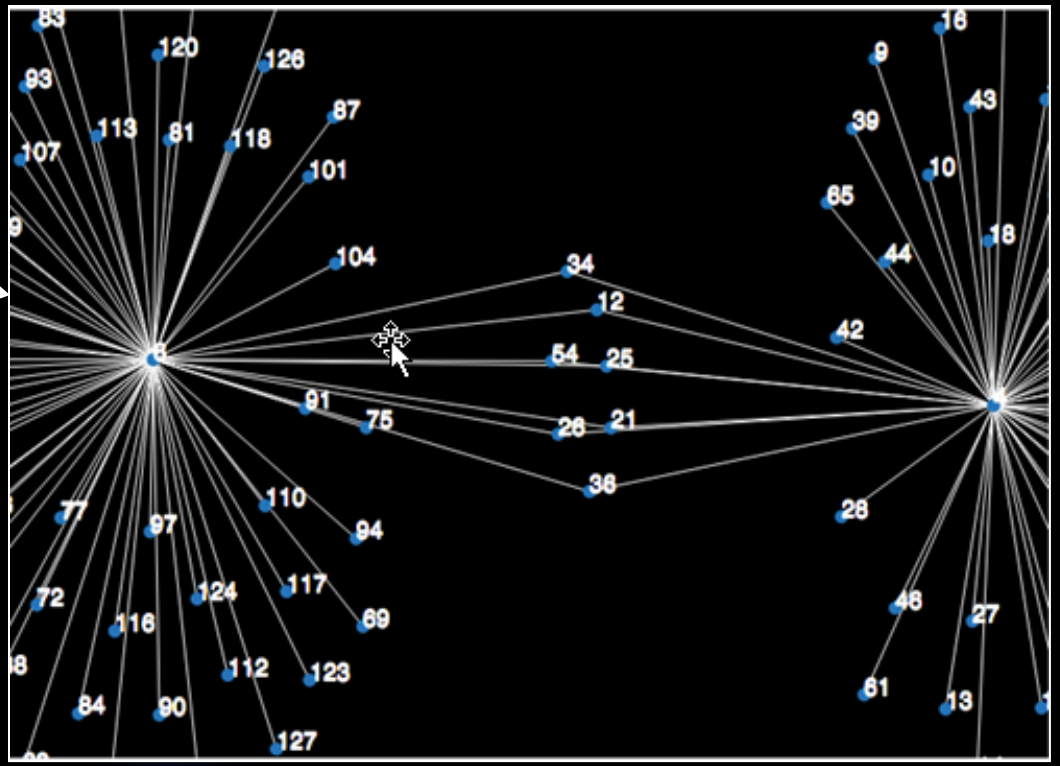
Diagram renders in application

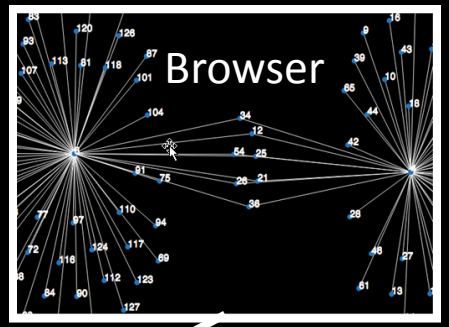
CPTL URN

Construct

Query

Ontology





1

2

3

4

5

6

7

CPTL URN

Construct

Query

Ontology

Badger URN

Capability

Attribute

Badger Server

Security Estimator

State Data

Data

Data

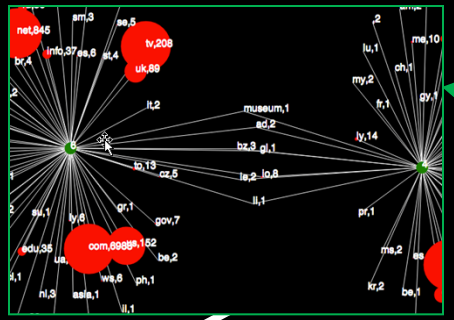
Baseline

Entropy (Kerf)

Other

Process

Data



1

2

3

4

5

6

7

8

CPTL URN

Construct

Query

Ontology

Badger Server

Badger URN

Capability

Attribute

Node Security State

State Data

Data

Data

Baseline

Entropy (Kerf)

Other

Process

Data

badger42.org

github.com/bigezy/badger

Demo Time

Comes a story of a corrupted young mind...

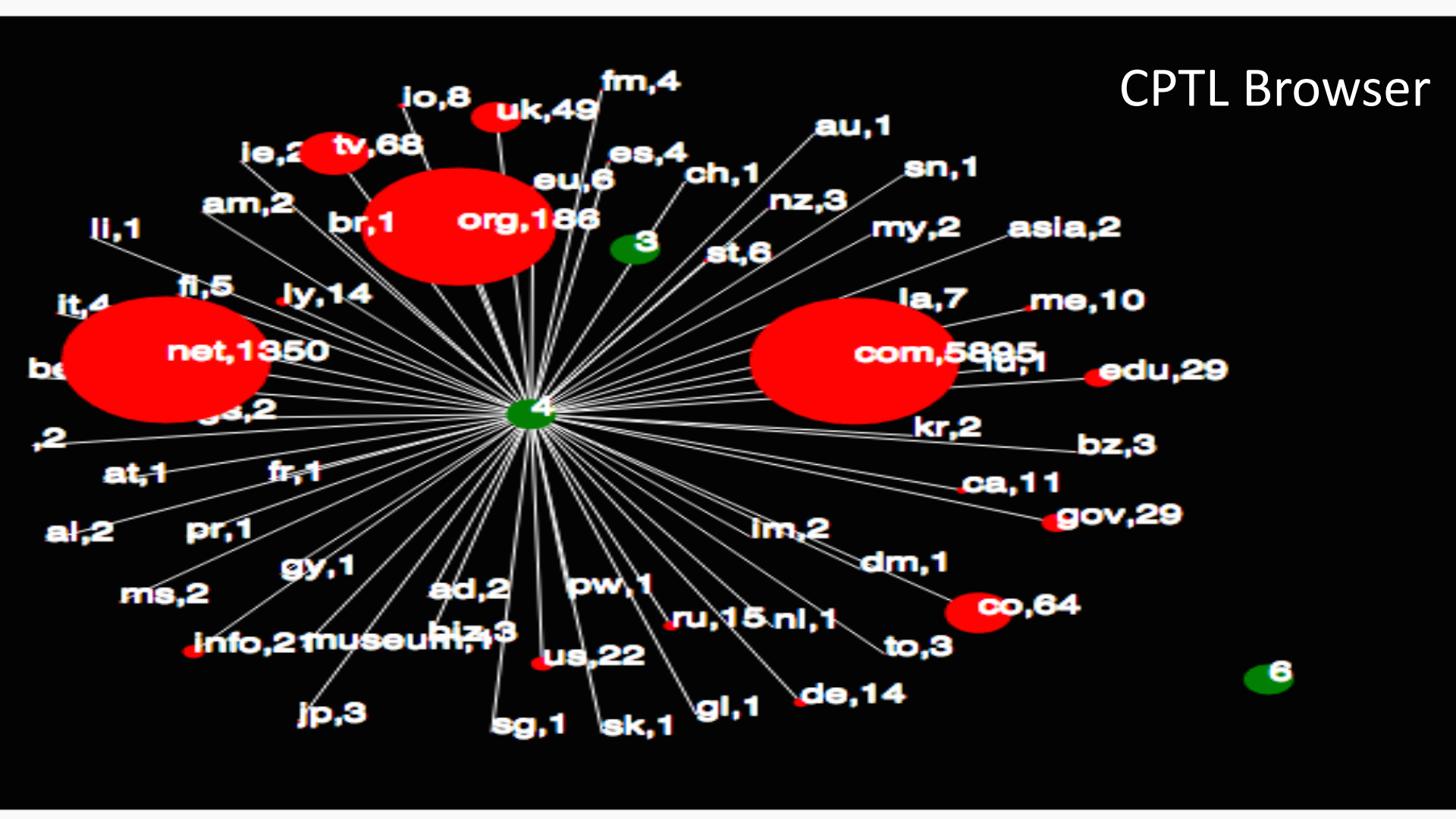
```
import skills, sys, time, demo
from luck import *

now =  time.time()
demo = open('badger' , 'r')

for blackhat in demo:
    print skills.haxor(blackhat)
else:
    print 'WASTED!'
```

REQUEST:badger42.org/GETCAPABILITY?source_vertex_attr_type=urn-cptl-HOST-ipv4

RESPONSE: A list of capabilities formatted as text/json

[{ name: "urn:badger:get_hostip_dest_hostnames", description: "Given an IPv4 address, get the destination hostnames",
source_vertex_attr_type = "urn-cptl-HOST-ipv4",
target_vertex_attr_type = "urn-cptl-HOST-hostname" },

{ name: "urn:badger:get_host_dest_tldcounts", description: "Given an IPv4 address, get the top-level domain counts",
source_vertex_attr_type = "urn-cptl-HOST-ipv4",
target_vertex_attr_type = "urn-cptl-HOST-hostname" }]


CPTL Browser
io,8
uk,49
fm,4
au,1
tv,68
es,4
ch,1
sn,1
eu,6
nz,3
am,2
br,1
org,186
3
my,2
asia,2
ll,1
st,6
fi,5
ly,14
la,7
me,10
net,1350
com,5895
edu,29
4
kr,2
bz,3
at,1
fr,1
ca,11
gov,29
pr,1
im,2
dm,1
gy,1
ad,2
pw,1
co,64
ru,15
nl,1
to,3
info,2
us,22
6
de,14
jp,3
gl,1
sg,1
sk,1

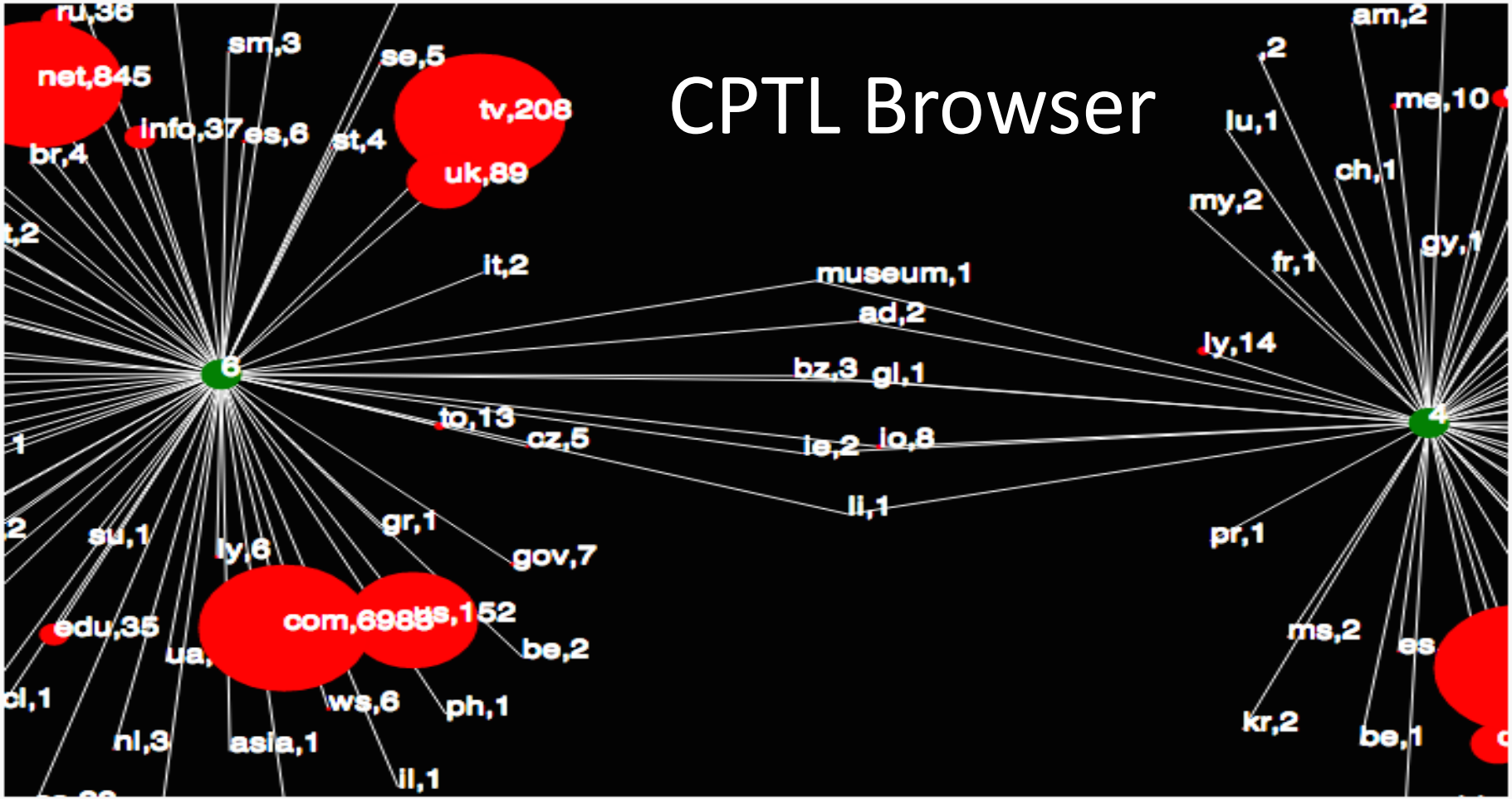
REQUEST:badger42.com/service?name=urn:badger:get_host_dest_tld
counts_selected_vertex_attr_values =192.168.1.100,192.168.1.120

RESPONSE: A graph of the following format:

{[{"id": 1, source1_vertex_attr_type: "urn-cptl-HOST-ipv4"}
{"id": 2, source2_vertex_attr_type: "urn-cptl-HOST-ipv4" {"id": 3,
target_1_vertex_attr_type: "urn-cptl-HOST-tag-tldcount",
target_1_vertex_attr_value: "com,44"}{"id": 4,
target_2_vertex_attr_type: "urn-cptl-HOST-tag-tldcount",
target_2_vertex_attr_value: "com,44"}],

EDGES: [{"source": 1, "target":4}, {"source":2, "target":4}] }

Graph (reload)



map.network-perception.com

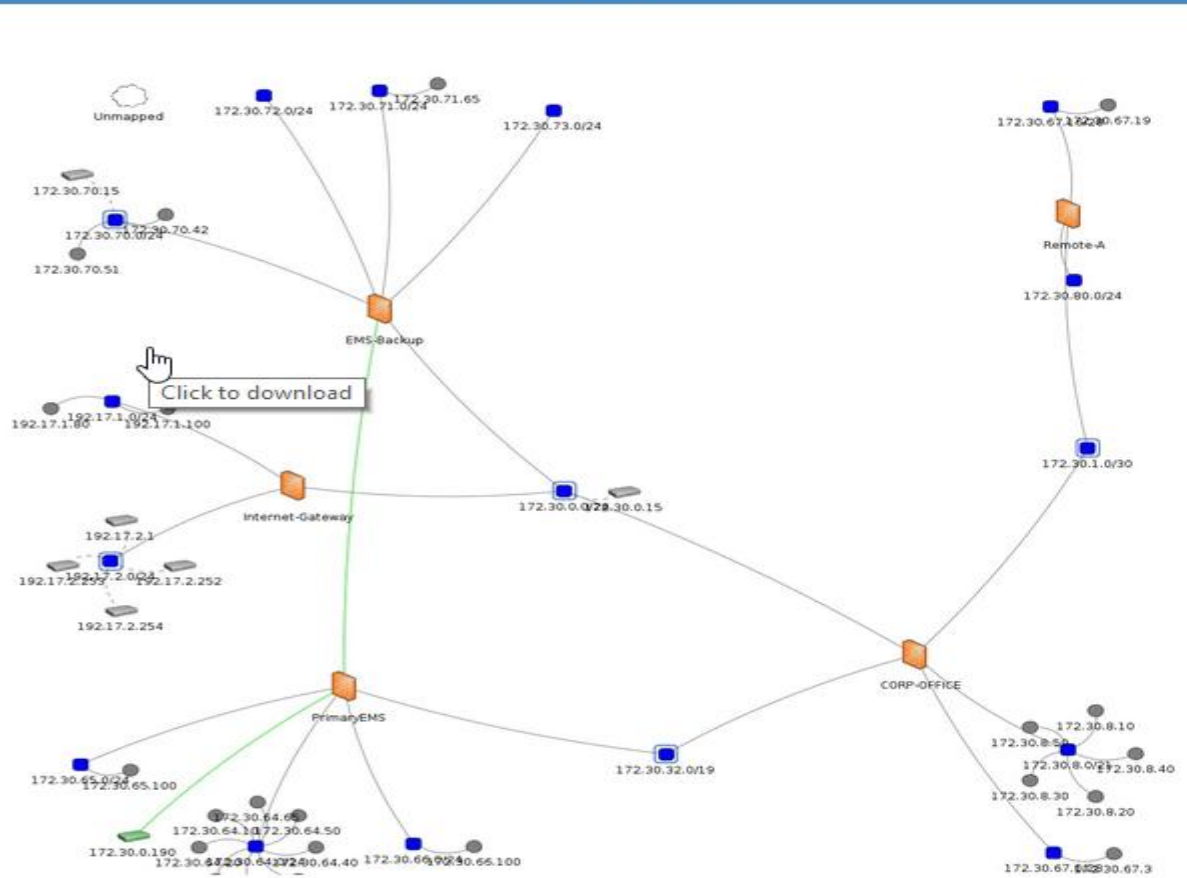
Analysis in progress...

```
Mode: command line
[Cisco Parser][Backup_EMS.txt] done.
DONE 1/5
[Cisco Parser][Primary_EMS.txt] done.
DONE 2/5
[Cisco Parser][Remote_A.txt] done.
DONE 3/5
[Cisco Parser][Internet.txt] done.
DONE 4/5
[Cisco Parser][Main_CORP.txt] done.
DONE 5/5
File(s) successfully parsed.
Performing topology inference...
Inspect routes...
Creating primary networks...
Marking VPN networks...
Creating nodes from group definitions...
Building border cloud...
```

NP-View Topology BETA

Home About Contact

Network Map Generated



Next Step: T

Customize your map • Run a path an

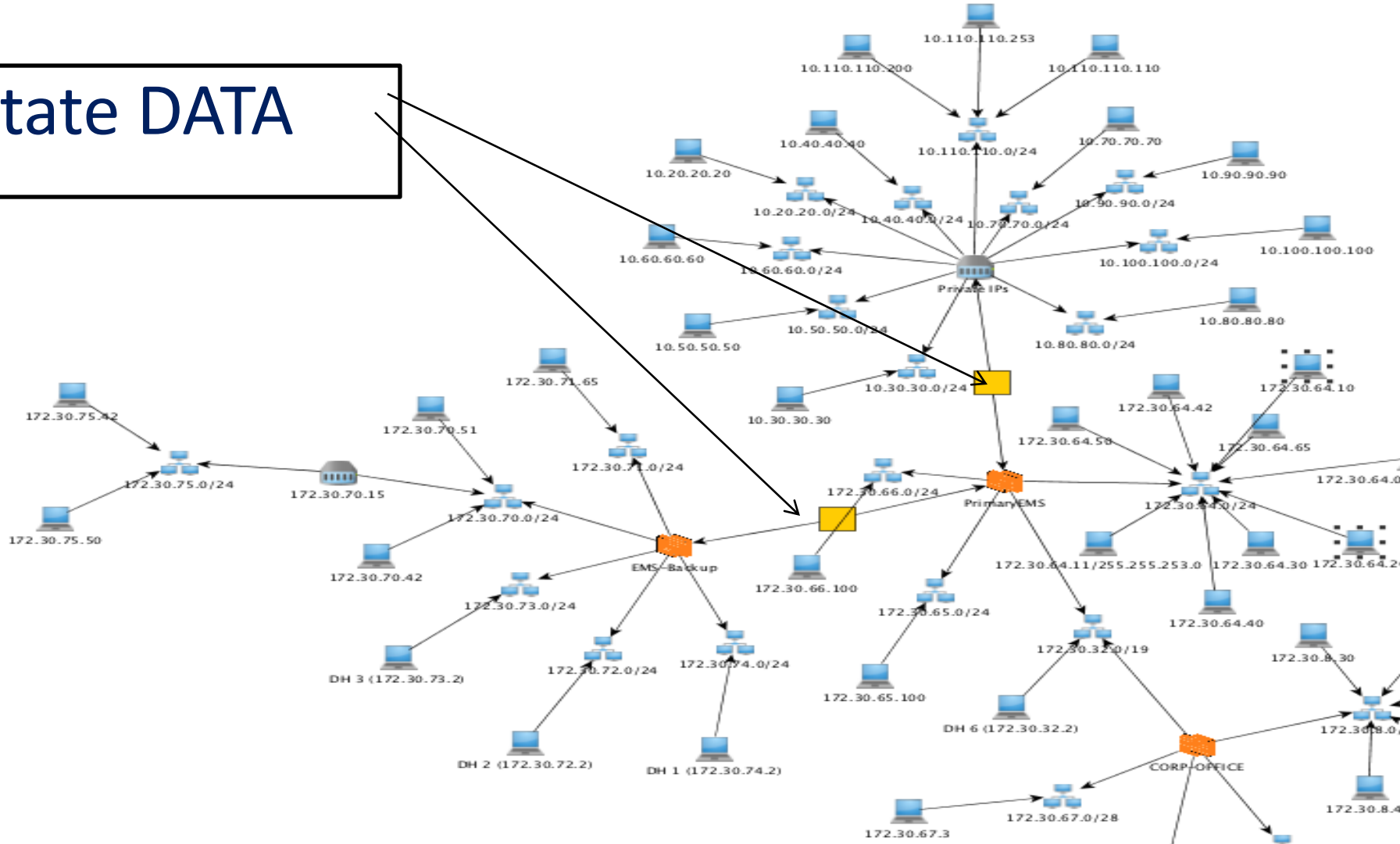
Rate this map

Your response has been recorde

Submit another response

This form was created using Go
Forms.
Create your own

State DATA


10.110.110.253
10.110.110.200
10.110.110.110
10.40.40.40
10.110.110.0/24
10.70.70.70
10.20.20.20
10.90.90.90
10.20.20.0/24
10.40.40.0/24
10.70.70.0/24
10.90.90.0/24
10.60.60.60
10.60.60.0/24
10.100.100.0/24
10.100.100.100
Private IPs
10.50.50.0/24
10.80.80.80
10.50.50.50
10.80.80.0/24
10.30.30.0/24
10.30.30.30
172.30.71.65
172.30.75.42
172.30.70.51
172.30.71.0/24
172.30.64.10
172.30.64.42
172.30.64.58
172.30.64.65
172.30.75.0/24
172.30.70.15
172.30.66.0/24
Primary EMS
172.30.64.0/24
172.30.70.0/24
172.30.75.50
172.30.70.42
EMS - Backup
172.30.64.11/255.255.253.0
172.30.64.30
172.30.73.0/24
172.30.66.100
172.30.65.0/24
172.30.64.40
DH 3 (172.30.73.2)
172.30.72.0/24
172.30.74.0/24
172.30.32.0/19
172.30.65.100
DH 6 (172.30.32.2)
DH 2 (172.30.72.2)
DH 1 (172.30.74.2)
CORP - OFFICE
172.30.67.0/28
172.30.67.3


BADGER42.ORG

Github.com/bigezy/badger

Github.com/ITI/cptl-power