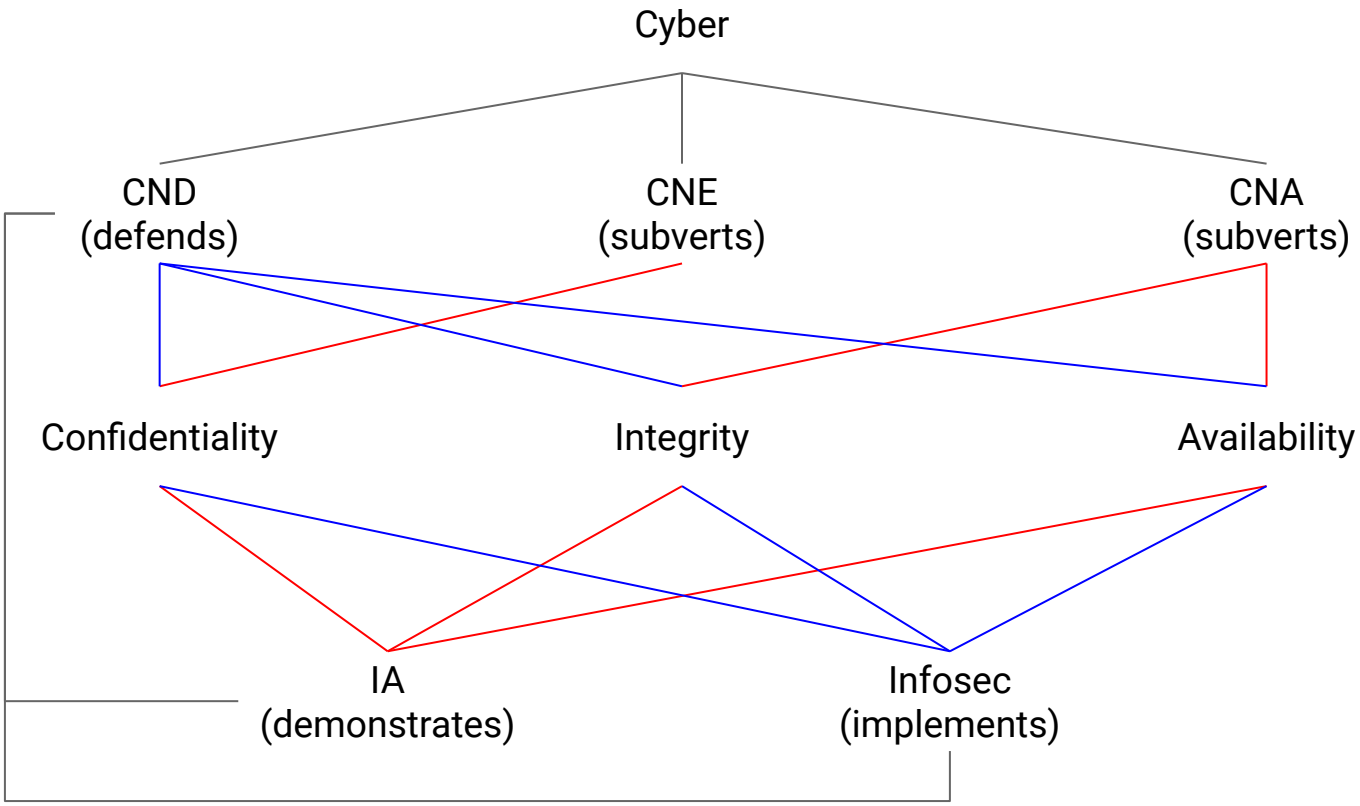
./red_team_project

Security Data Analytics Workshop

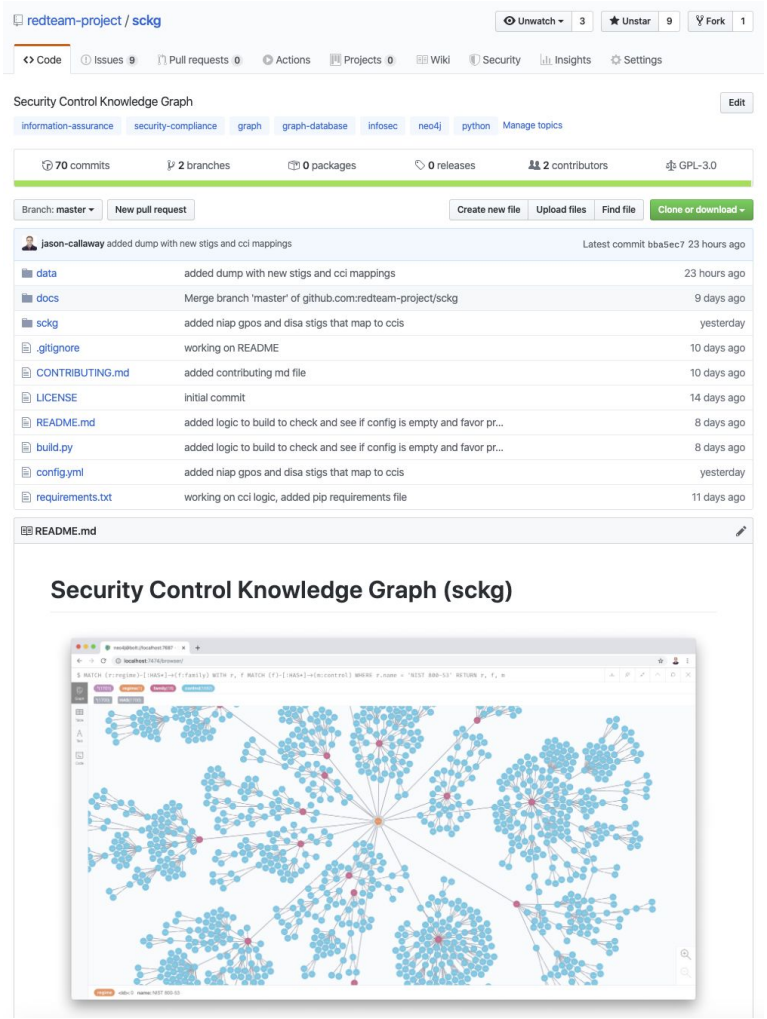

Cyber
CND
(defends)
CNE
(subverts)
CNA
(subverts)
Confidentiality
Integrity
Availability
IA
(demonstrates)
Infosec
(implements)

Red Team Project - Focus Areas

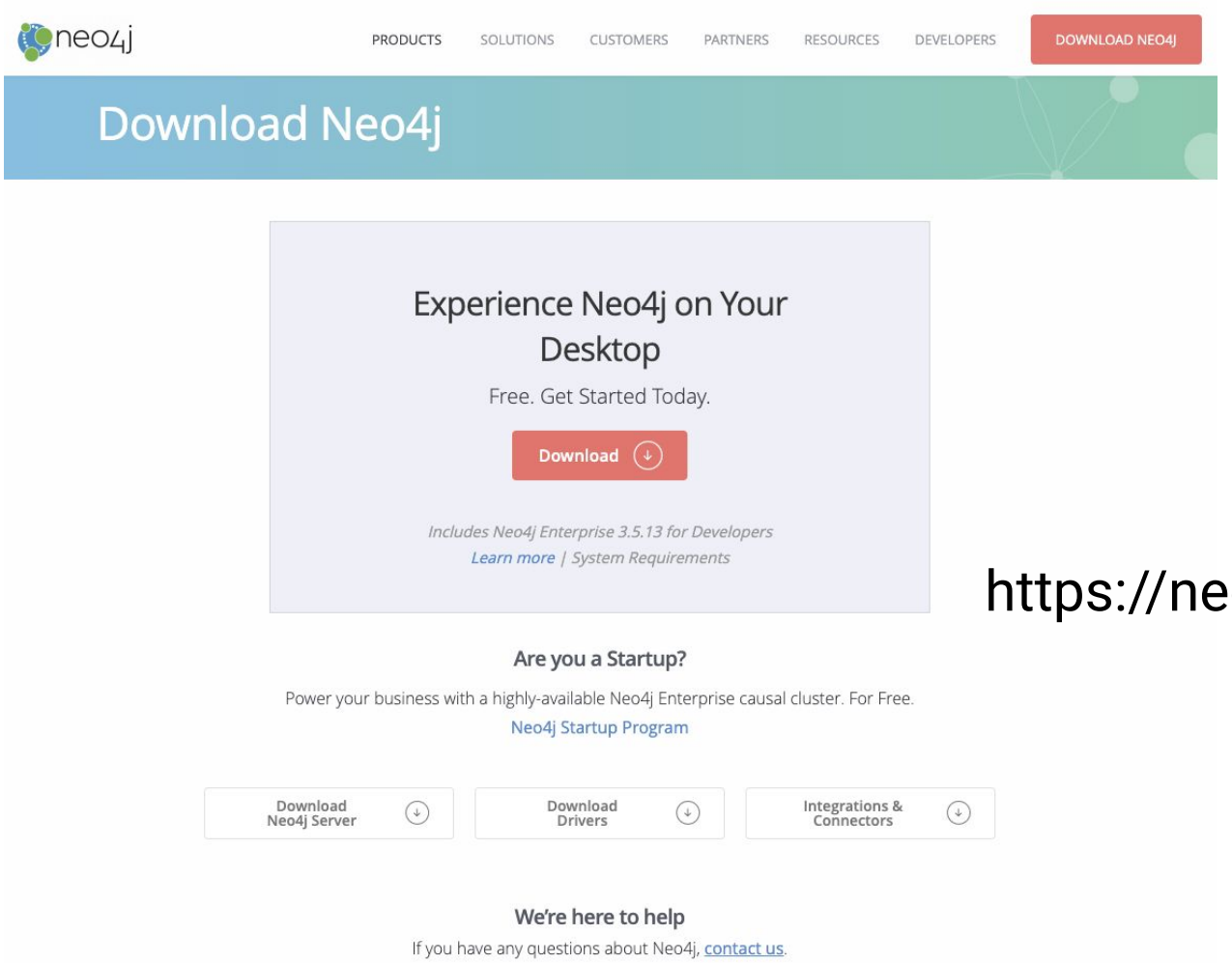
Cyber Range Automation

Binary Risk Quantification

Standards Advancement



https://github.com/redteam-project/sckg



https://neo4j.com/download

NIST Special Publication 800-53
Revision 4

Security and Privacy Controls for Federal Information Systems and Organizations

JOINT TASK FORCE
TRANSFORMATION INITIATIVE

DoD CYBER EXCHANGE PUBLIC

Topics Training PKI/PKE SRGs/STIGs

STIGs Document Library

SECURITY TECHNICAL IMPLEMENTATION GUIDES (STIGS)

- SRG/STIGs Home
- Control Correlation Identifier (CCI)
- Document Library

Home » Security Technical Implementation Guides (STIGs) » STIGs Document Library

Show 25 entries Search:

TITLE	SIZE	UPDATED
Canonical Ubuntu 16.04 STIG - Ver 1, Rel 2	650.28 KB	26 Apr 2019
IBM AIX 7.x STIG - Ver 1, Rel 1	684.81 KB	15 Aug 2019
IBM AIX 7.x STIG - Version 1 - Release Memo	865.71 KB	14 May 2019
Oracle Linux 5 STIG - Ver 1, Rel 13	500.65 KB	26 Jun 2019
Red Hat Enterprise Linux 6 STIG - Ver 1, Rel 24	721.58 KB	31 Oct 2019
Red Hat Enterprise Linux 7 STIG - Ver 2, Rel 5	759.9 KB	31 Oct 2019
Solaris 10 SPARC STIG, Ver 1, Rel 25	1.01 MB	26 Jul 2019
Solaris 10 x86 STIG, Ver 1, Rel 25	1.18 MB	26 Jul 2019
Solaris 11 SPARC STIG - Ver 1, Rel 19	802.97 KB	31 Oct 2019
Solaris 11 X86 STIG - Ver 1, Rel 19	997.93 KB	31 Oct 2019
SUSE Linux Enterprise Server 12 STIG - Ver 1, Rel 3	817.74 KB	31 Oct 2019

STIG TOPICS

- Operating Systems (11)
- UNIX/Linux (11)

NIAP
National Information Assurance Partnership
Common Criteria

About Us Products Protection Profiles Resources FAQ
NIAP Community
Site Search

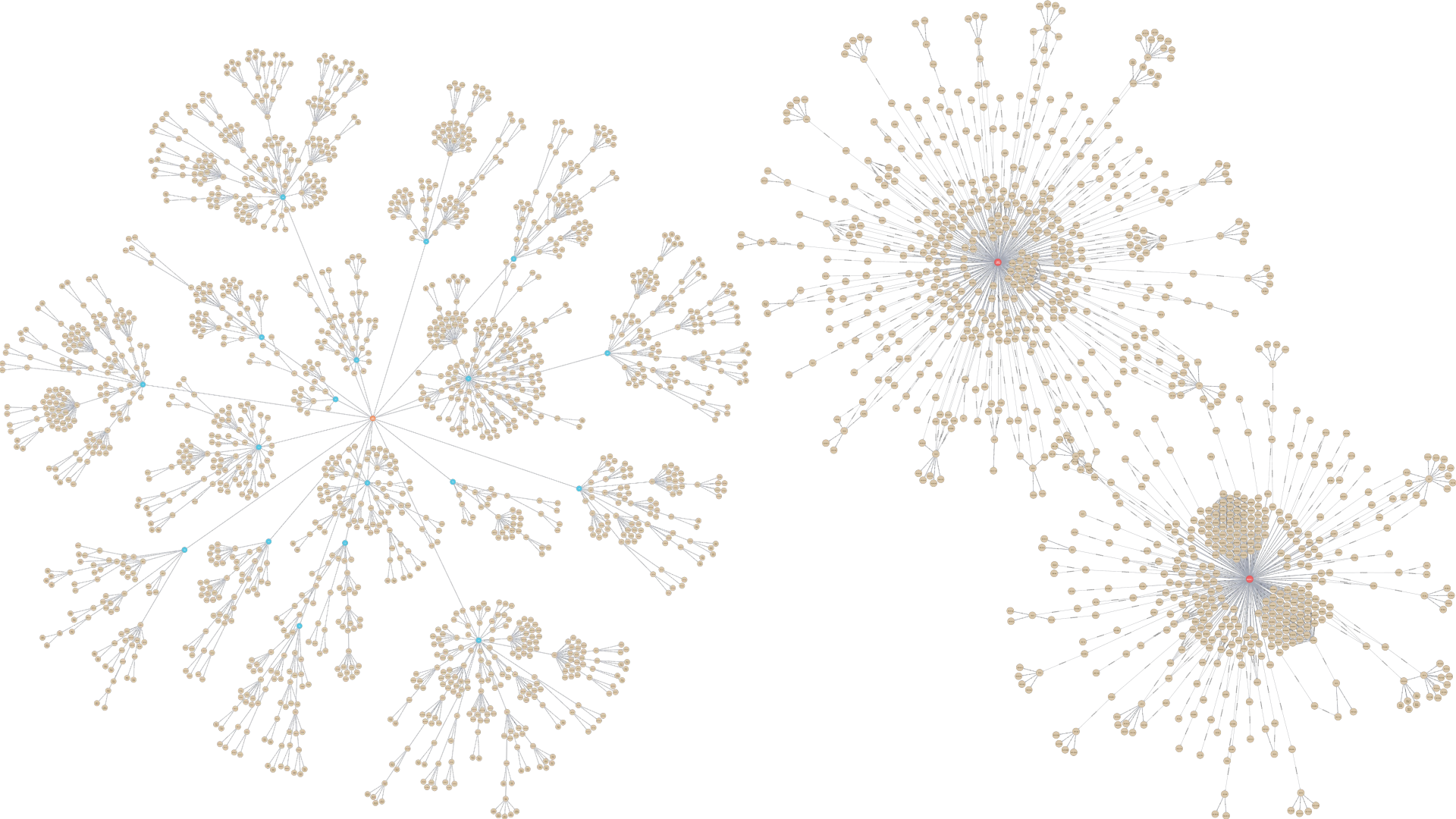
NIAP Oversees Evaluations of Commercial IT Products for Use in National Security Systems
Questions? We're here to help.

NIAP » Protection Profiles » Approved PPs » Details

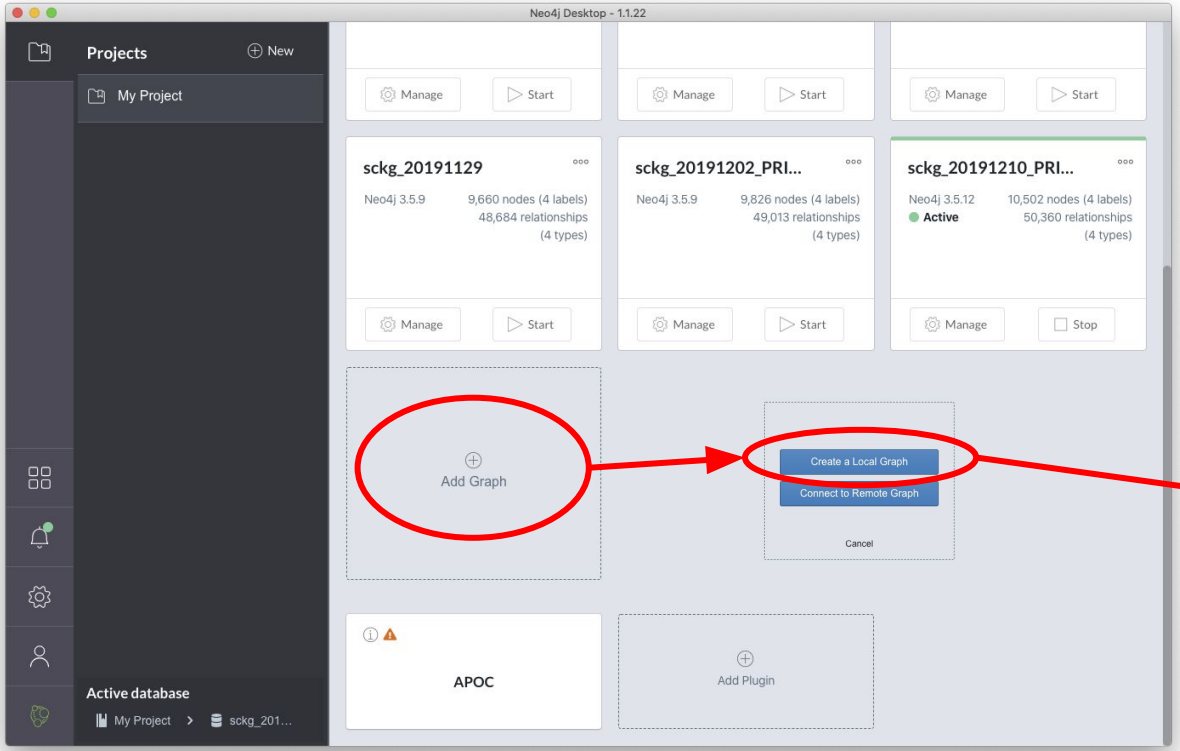
U.S. Government Approved Protection Profile - Protection Profile for General Purpose Operating Systems Version 4.2.1

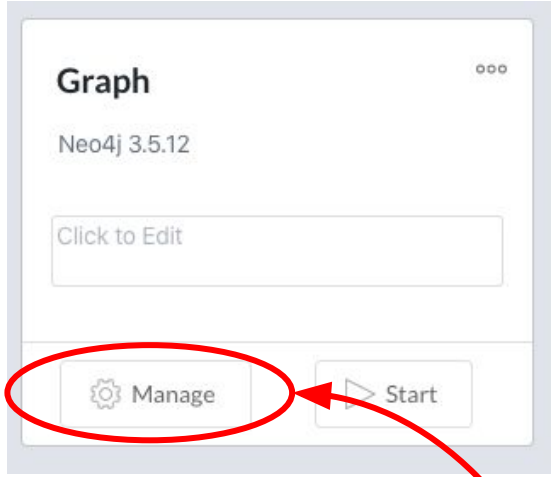
Short Name: pp_os_v4.2.1
Technology Type: Operating System
CC Version: 3.1
Date: 2019.04.22
Preceded By: pp_os_v4.2
Conformance Claim: None

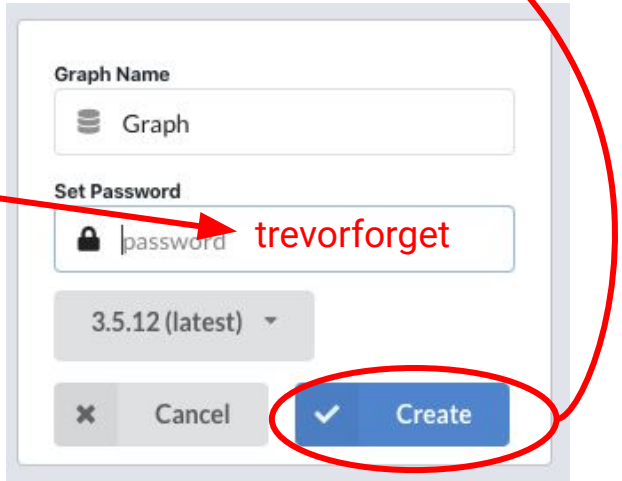
Protection Profile
Protection Profile
Configuration Annex
Configuration Annex

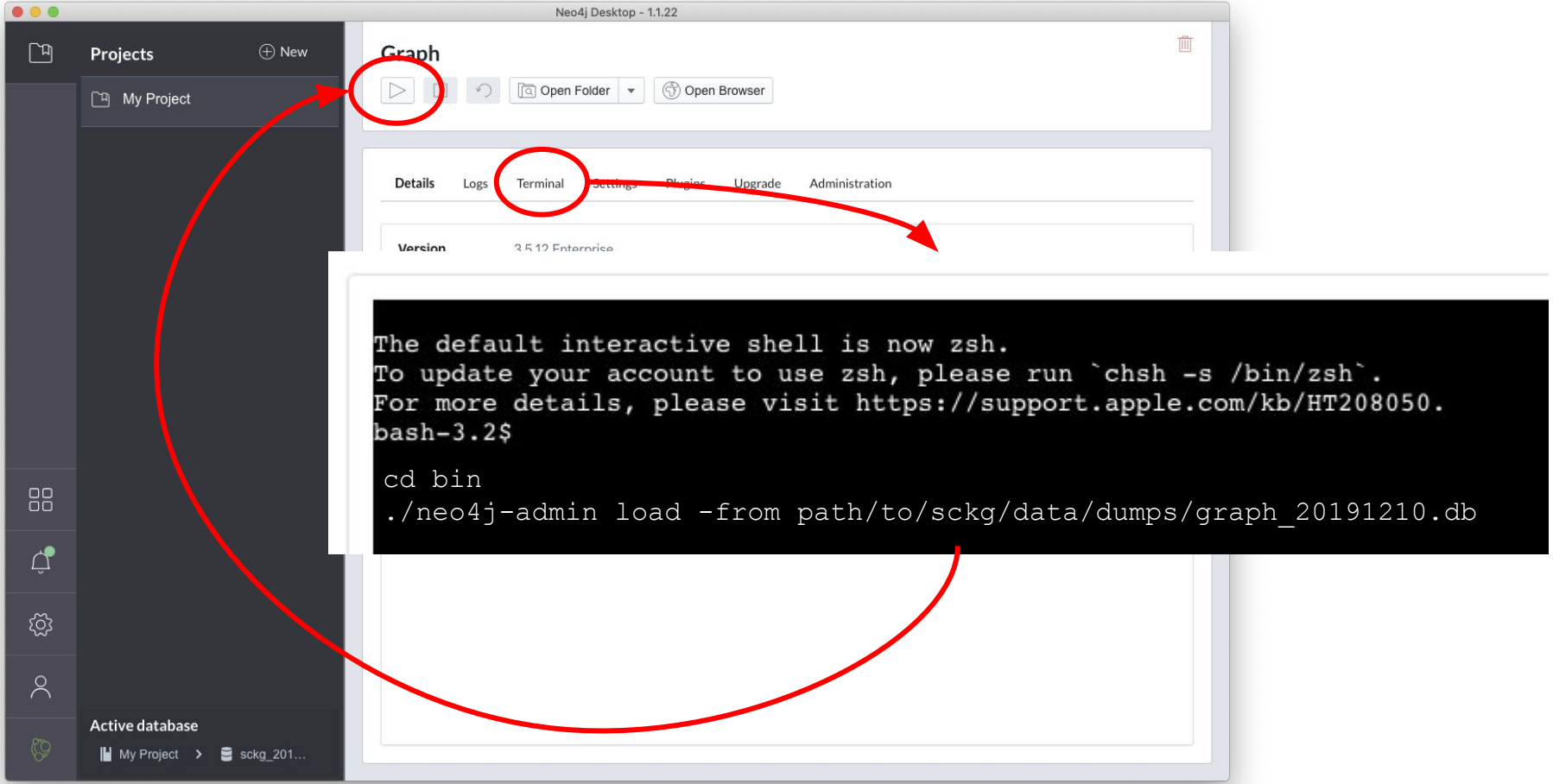


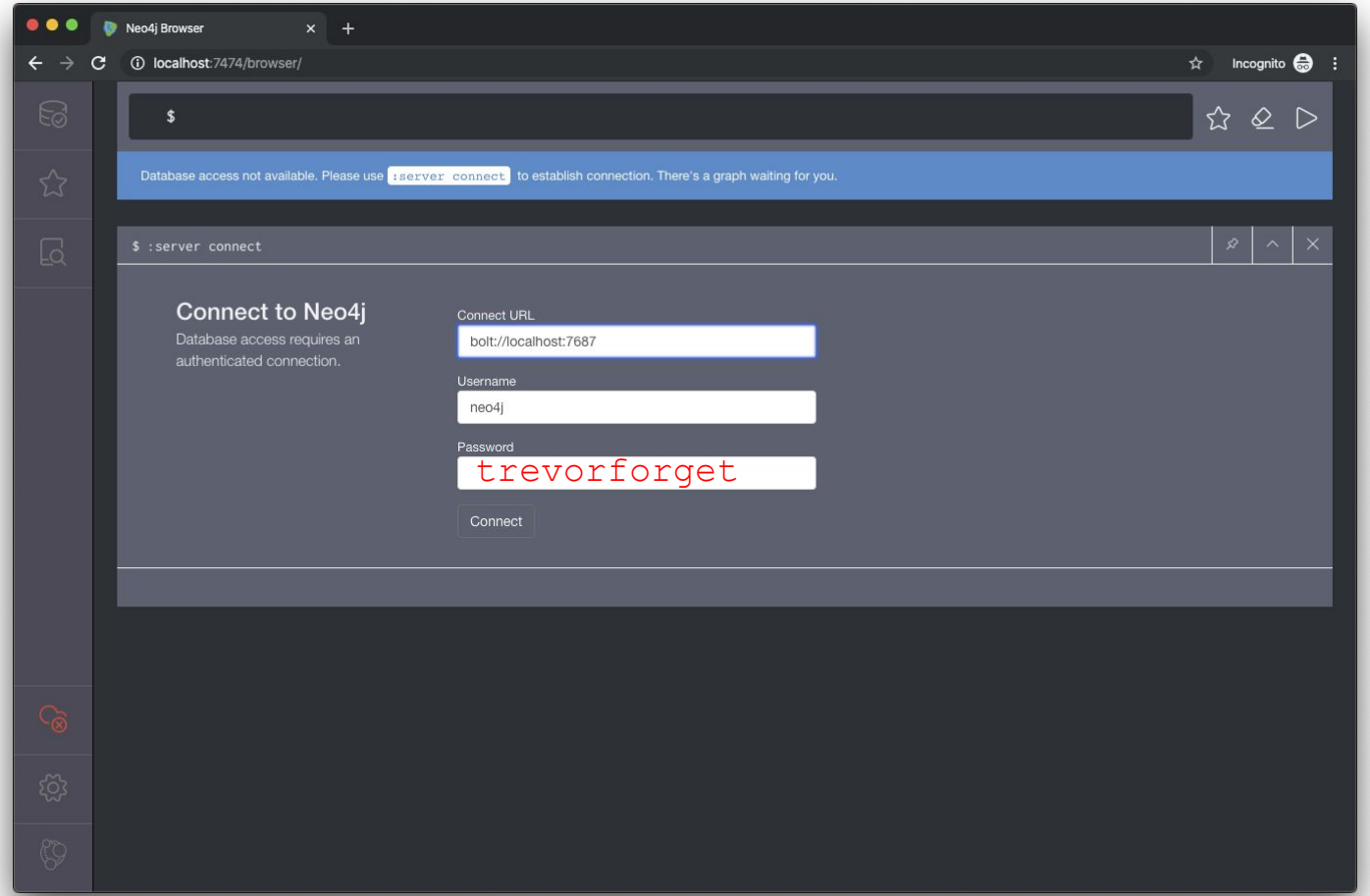
git clone https://github.com/redteam-project/sckg



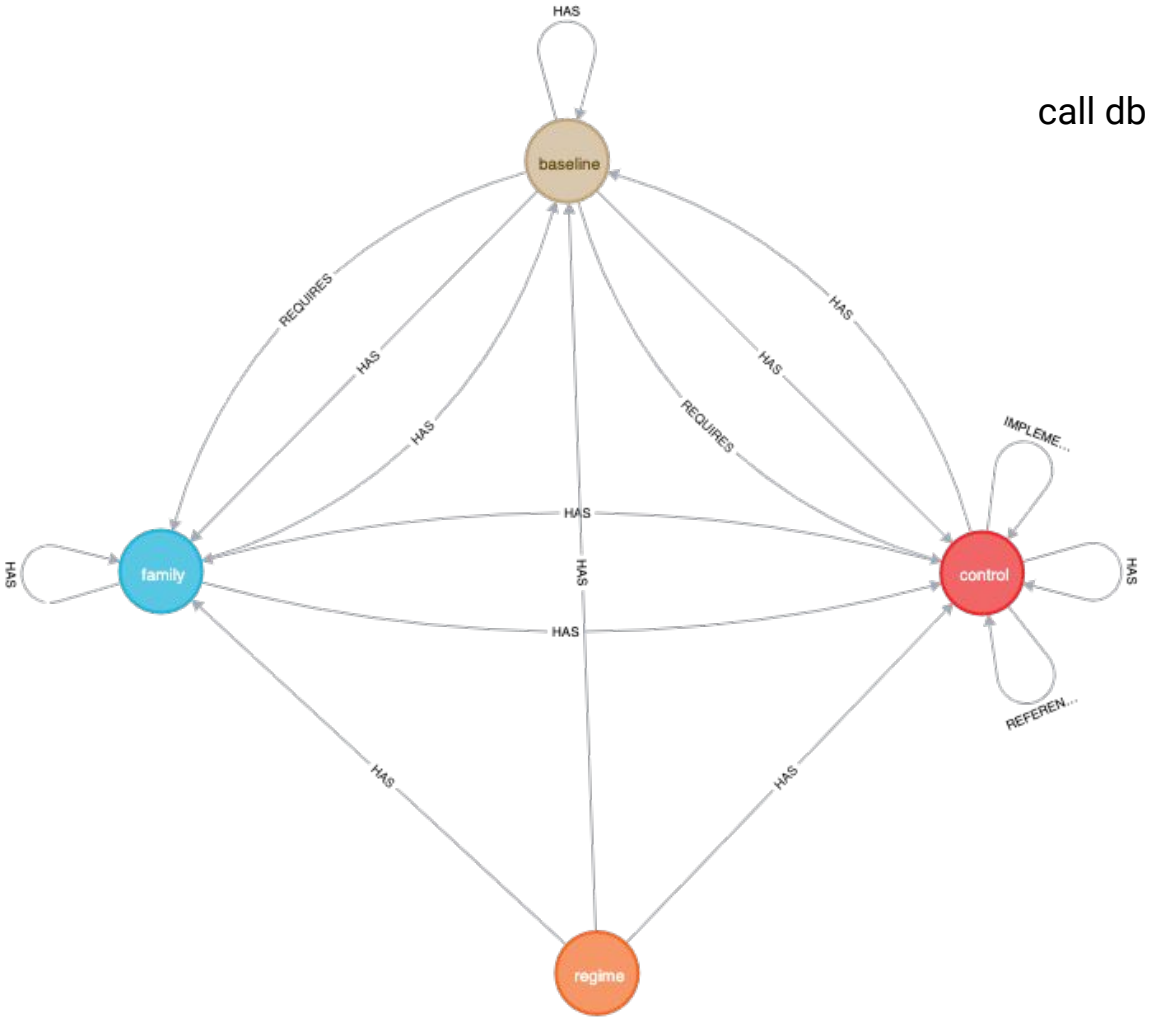




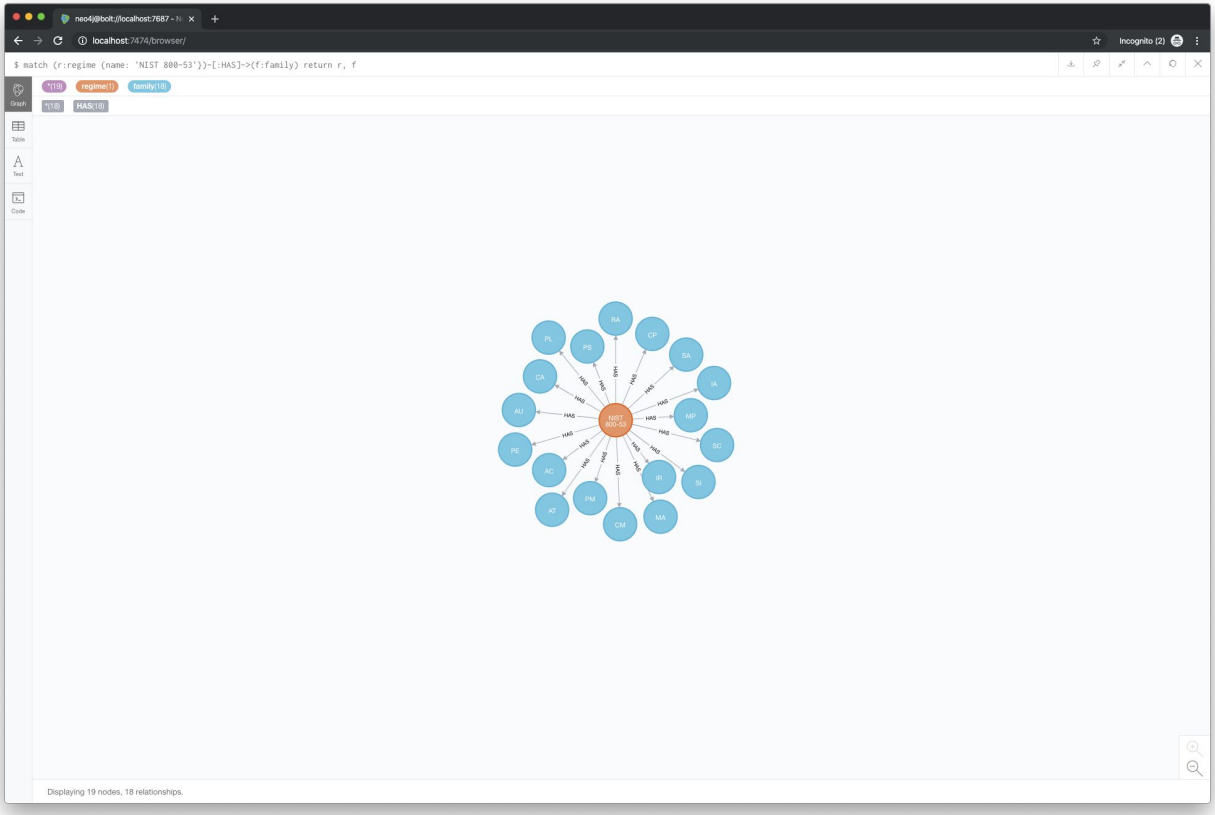

Neo4j Desktop - 1.1.22
Projects
New
My Project
Graph
Open Folder
Open Browser
Details
Logs
Terminal
Upgrade
Administration
Version
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
bash-3.2$
cd bin
./neo4j-admin load -from path/to/sckg/data/dumps/graph_20191210.db
Active database
My Project
sckg_201...


Neo4j Browser
localhost:7474/browser/
Incognito
$
Database access not available. Please use :server connect to establish connection. There's a graph waiting for you.
$:server connect
Connect to Neo4j
Database access requires an authenticated connection.
Connect URL
bolt://localhost:7687
Username
neo4j
Password
trevorforget
Connect

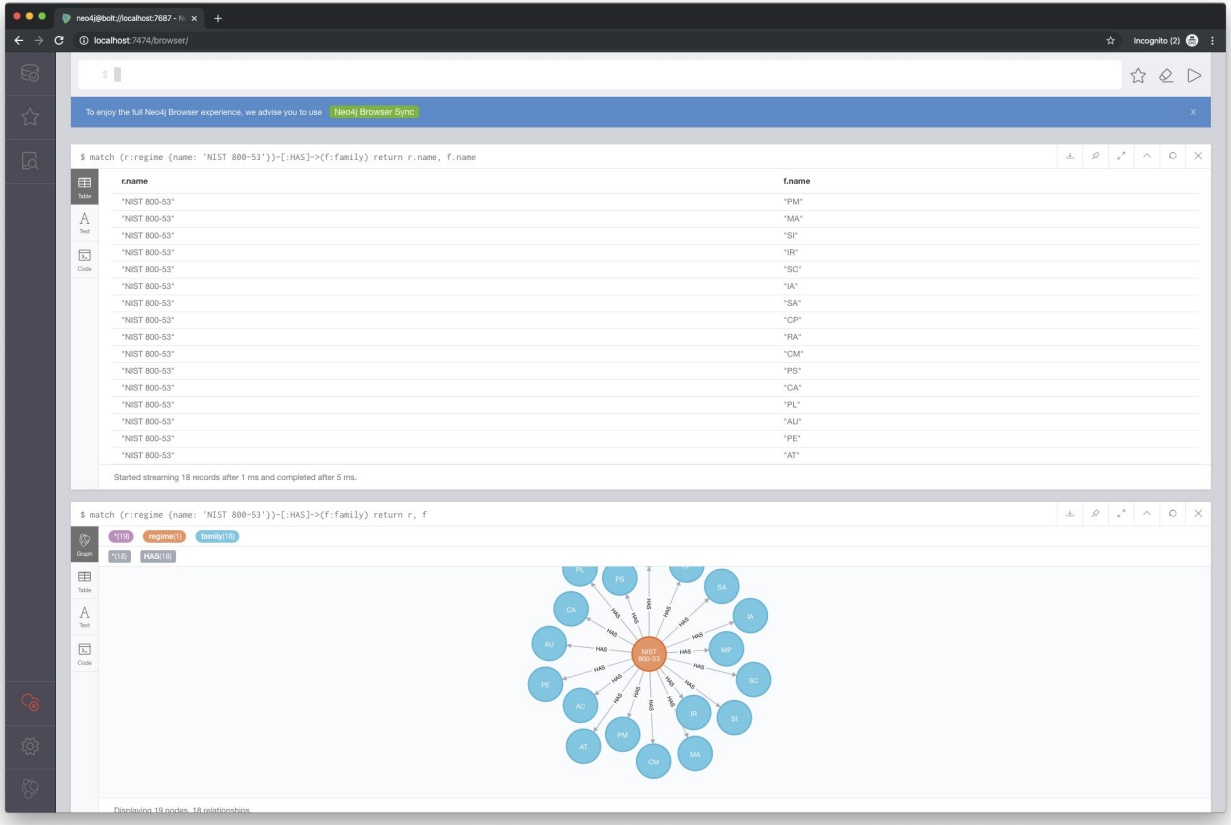
call db.schema()



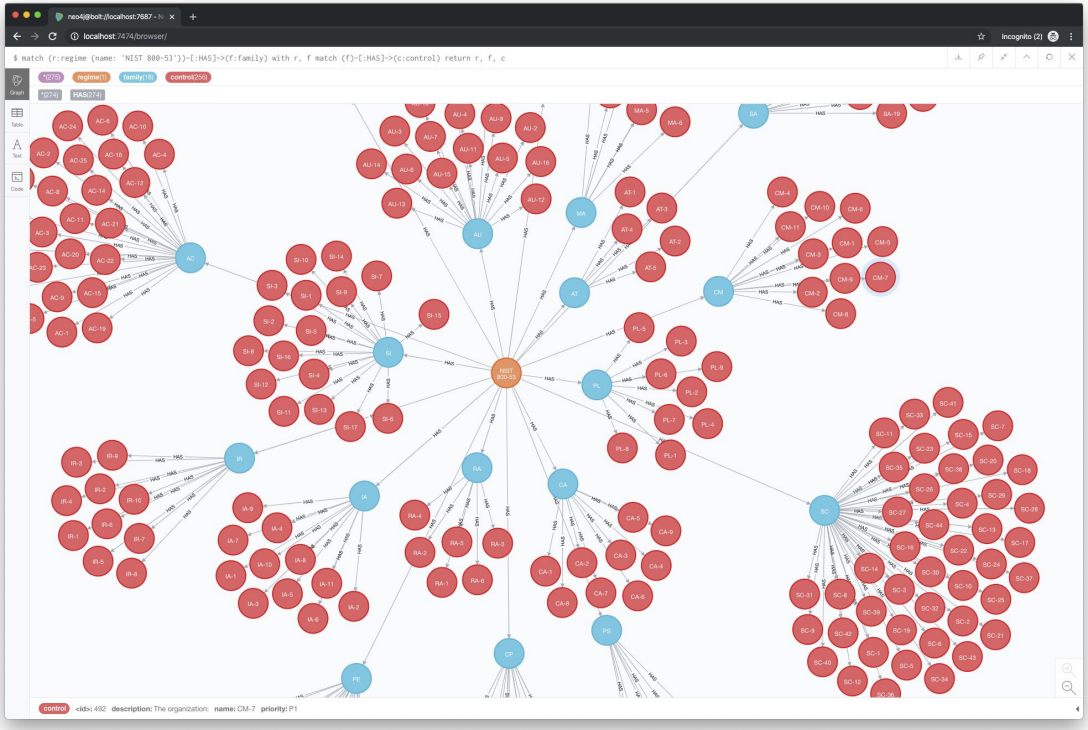
```
match (r:regime {name: 'NIST 800-53'})-[:HAS]->(f:family) return r, f
```



```
match (r:regime {name: 'NIST 800-53'})-[:HAS]->(f:family) return r.name, f.name
```



```
match (r:regime {name: 'NIST 800-53'})-[:HAS]->(f:family)
with r, f
match (f)-[:HAS]->(c:control) return r, f, c
```

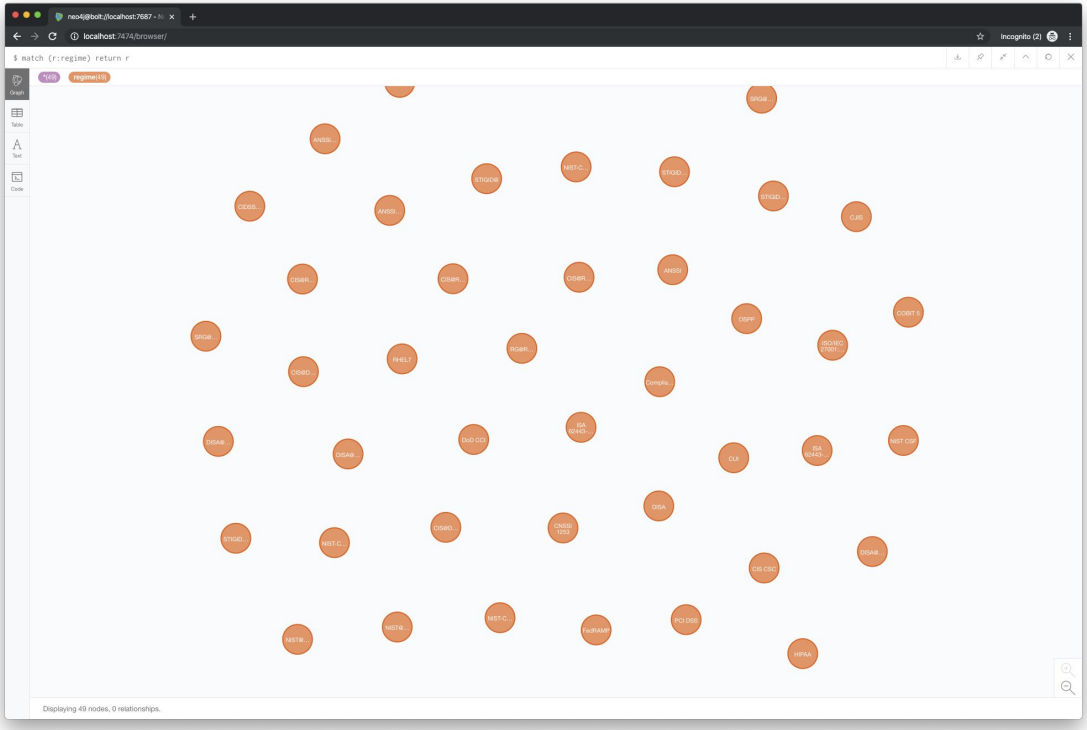


```
match (r:regime {name: 'NIST 800-53'})-[:HAS *]->(f:family)
with r, f
match (f)-[:HAS]->(c:control) return r, f, c
```

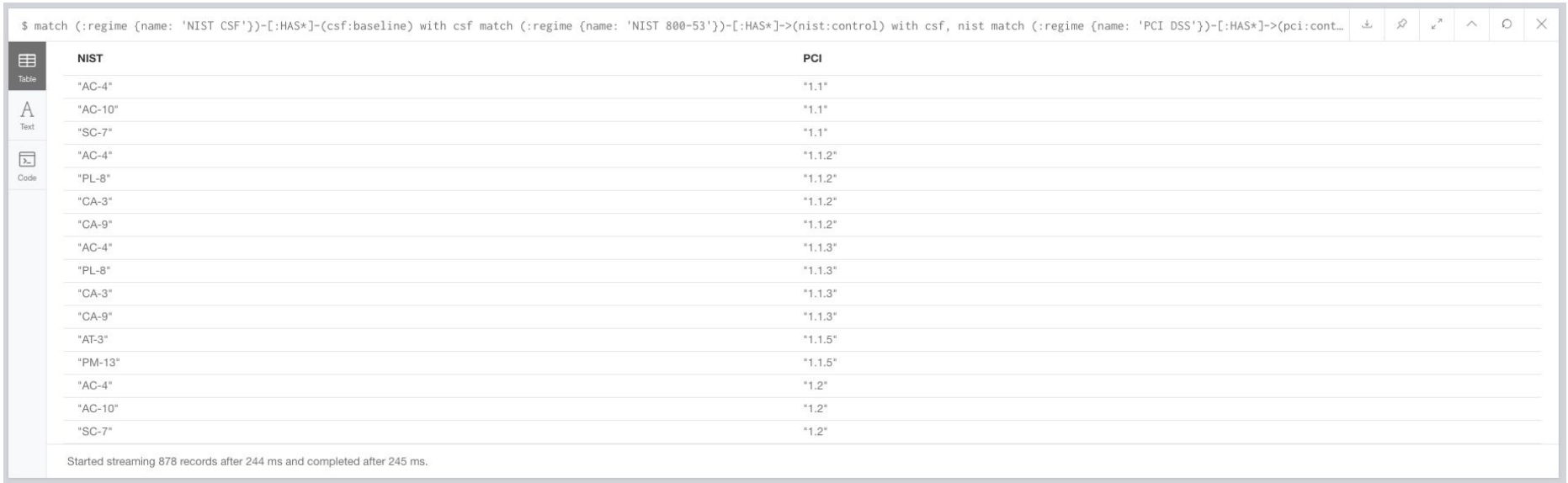
```
:config

{
  "cmdchar": ":",
  "maxHistory": 30,
  "theme": "auto",
  "initCmd": ":play start",
  "initialNodeDisplay": 300,
  "maxNeighbours": 100,
  "showSampleScripts": true,
  "browserSyncDebugServer": null,
  "maxRows": 1000,
  "shouldReportUdc": true,
  "autoComplete": true,
  "scrollToTop": true,
  "maxFrames": 30,
  "editorAutocomplete": true,
  "editorLint": false,
  "useCypherThread": true,
  "enableMultiStatementMode": false
}
```

Note the British spelling!

```
match (r:regime) return r
```

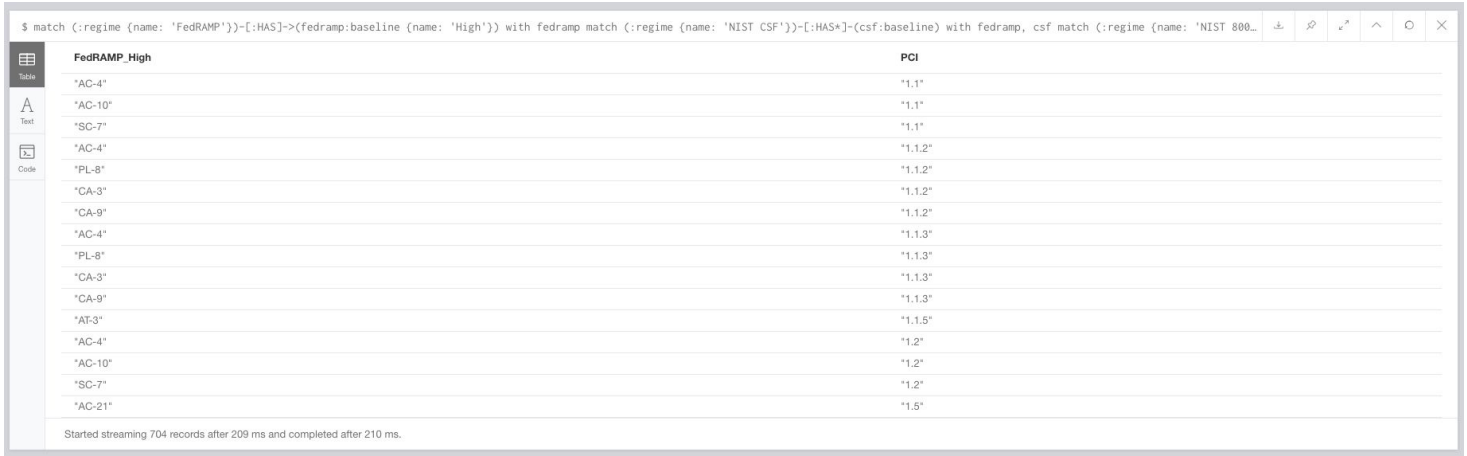


```
match (:regime {name: 'NIST CSF'})-[:HAS*]-(csf:baseline)
with csf
match (:regime {name: 'NIST 800-53'})-[:HAS*]->(nist:control)
with csf, nist
match (:regime {name: 'PCI DSS'})-[:HAS*]->(pci:control)
with csf, nist, pci
match (csf)-[:REQUIRES]->(nist), (csf)-[:REQUIRES]->(pci)
return nist.name as NIST, pci.name as PCI order by PCI
```



NIST	PCI
"AC-4"	"1.1"
"AC-10"	"1.1"
"SC-7"	"1.1"
"AC-4"	"1.1.2"
"PL-8"	"1.1.2"
"CA-3"	"1.1.2"
"CA-9"	"1.1.2"
"AC-4"	"1.1.3"
"PL-8"	"1.1.3"
"CA-3"	"1.1.3"
"CA-9"	"1.1.3"
"AT-3"	"1.1.5"
"PM-13"	"1.1.5"
"AC-4"	"1.2"
"AC-10"	"1.2"
"SC-7"	"1.2"

```
match (:regime {name: 'FedRAMP'})-[:HAS]->(fedramp:baseline {name: 'High'})
with fedramp
match (:regime {name: 'NIST CSF'})-[:HAS*]-(csf:baseline)
with fedramp, csf
match (:regime {name: 'NIST 800-53'})-[:HAS*]->(nist:control)
with fedramp, csf, nist
match (:regime {name: 'PCI DSS'})-[:HAS*]->(pci:control)
with fedramp, csf, nist, pci
match (fedramp)-[:REQUIRES]->(nist), (csf)-[:REQUIRES]->(nist),
(csf)-[:REQUIRES]->(pci)
return nist.name as FedRAMP_High, pci.name as PCI order by PCI
```



FedRAMP_High	PCI
"AC-4"	"1.1"
"AC-10"	"1.1"
"SC-7"	"1.1"
"AC-4"	"1.1.2"
"PL-8"	"1.1.2"
"CA-3"	"1.1.2"
"CA-9"	"1.1.2"
"AC-4"	"1.1.3"
"PL-8"	"1.1.3"
"CA-3"	"1.1.3"
"CA-9"	"1.1.3"
"AT-3"	"1.1.5"
"AC-4"	"1.2"
"AC-10"	"1.2"
"SC-7"	"1.2"
"AC-21"	"1.5"

redteam-project / sckg

Unwatch 3 | Unstar 9 | Fork 1

Code | Issues 9 | Pull requests 0 | Actions | Projects 0 | Wiki | Security | Insights | Settings

Branch: master ▾ | sckg / build.py / Jump to ▾ | Find file | Copy path

jason-callaway added logic to build to check and see if config is empty and favor pr... 0ab6f8f 8 days ago

1 contributor

93 lines (74 sloc) | 2.98 KB | Raw | Blame | History

You're using code navigation to jump to definitions or references. Learn more or give us feedback

```
import importlib
import os
import yaml

from sckg.neo4j import Neo4j

class Build(object):
  """Class for building the Neo4j sckg"""

  def __init__(self):
    """Init build class

    This class will grab config data from config.yml and private.yml, then
    iterate over the regimes from both files to build the graph.
    """
    self.config = {}
    with open('config.yml', 'r') as f:
      self.config = yaml.safe_load(f.read())

    try:
      with open('private.yml', 'r') as f:
        private_regimes = yaml.safe_load(f.read())
        if self.config.get('regimes'):
          self.config['regimes'] += private_regimes['regimes']
        else:
          self.config['regimes'] = private_regimes['regimes']
    except FileNotFoundError as e:
      # It's ok if there's no private config
      pass
```

```python
  def regime_etl(self, regime):
    """Extract / transform / load function for a regime

    Args:
      regime: the dict regime list element from the configs

    Returns:
      None

    Raises:
      None
    """

    # first we have to figure out which module and class we're going to use
    # for this regime. if the regime dict doesn't specify, we'll use the
    # generic class
    etl_config = regime.get('etl', 'generic')
    if etl_config == 'generic':
      module_name = self.config['defaults']['generic']['module']
      class_name = self.config['defaults']['generic']['class']
    else:
      module_name = regime['etl']['module']
      class_name = regime['etl']['class']

    parsable_document = self.config['cwd'] + '/' + \
                        regime['document']['parsable']

    # this part's a little tricky. we're dynamically instantiating the class
    # associated with this regime with the getattr / import_module combo.
    # this way we can have a nicely generic etl function without a bunch of
    # repeated code.
    etl_class = getattr(importlib.import_module(module_name),
                        class_name)
    etl_instance = etl_class(self.config)

    # the next three lines are for the extract, transform, and load methods.
    # extract takes the parsable document and returns a list of rows from the
    # parsable doc
    regime_list = etl_instance.extract(regime, parsable_document)
    # transform takes the list of dicts and converts them into cypher statements
    stmts = etl_instance.transform(regime, regime_list)
    # load takes the cypher statements and sends them to the neo4j server
    etl_instance.load(regime, self.neo4j, stmts)

def main():
  build = Build()

if __name__ == '__main__':
  main()
```

redteam-project / sckg

Unwatch 3 | Unstar 9 | Fork 1

Code | Issues 9 | Pull requests 0 | Actions | Projects 0 | Wiki | Security | Insights | Settings

Branch: master ▾ sckg / config.yml — Find file | Copy path

jason-callaway added niap gpos and disa stigs that map to ccis — 18dc528 yesterday

1 contributor

217 lines (217 sloc) | 7.11 KB — Raw | Blame | History

```yaml
---
# Configuration settings for sckg
defaults:
  generic:
    module: 'sckg.etl.generic'
    class: 'Generic'
  templates:
    cypher:
      path: 'data/templates/cypher/'
regimes:
  - name: '800_53'
    description: 'NIST 800-53'
    document:
      source: 'http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf'
      parsable: 'data/regimes/nist_800-53r4.tsv'
    etl:
      module: 'sckg.etl.800_53'
      class: 'NIST80053'
    meta:
      render_related: False
      families:
        AC: 'Access Control'
        MP: 'Media Protection'
        AT: 'Awareness and Training'
        PE: 'Physical and Environmental Protection'
        AU: 'Audit and Accountability'
        PL: 'Planning'
        CA: 'Security Assessment and Authorization'
        PS: 'Personnel Security'
        CM: 'Configuration Management'
        RA: 'Risk Assessment'
        CP: 'Contingency Planning'
        SA: 'System and Services Acquisition'
        IA: 'Identification and Authentication'
        SC: 'System and Communications Protection'
        IR: 'Incident Response'
        SI: 'System and Information Integrity'
        MA: 'Maintenance'
        PM: 'Program Management'
```

redteam-project / sckg

Unwatch 3 | Unstar 9 | Fork 1

Code | Issues 9 | Pull requests 0 | Actions | Projects 0 | Wiki | Security | Insights | Settings

Branch: master ▾ sckg / sckg / etl / generic.py / Jump to ▾ | Find file | Copy path

jason-callaway generic etl comments done for now 08f815e 10 days ago

1 contributor

398 lines (321 sloc) | 11.8 KB | Raw | Blame | History

You're using code navigation to jump to definitions or references. Learn more or give us feedback

```
import os
from jinja2 import Template

class Generic(object):
  """generic ETL class

     The getters and setters for the ETL classes and child classes live here
     as well.
  """

  def __init__(self, config):
    """Init class for Generic ETL

    Args:
      config: the serialized config.yml data

    Returns:
      None

    Raises:
      None
    """
    self.config = config
    self.template_path = self.config['cwd'] + '/' + \
                         self.config['defaults']['templates']['cypher']['path']

  def extract(self, regime, parsable_document):
    """extract method

    Args:
      regime: the regime dict from config
      parsable_document: path relative to build.py to the regime source doc

    Returns:
      regime_list: a list of lines from the parsable_document

    Returns:
      None
    """
    with open(parsable_document, 'r') as f:
      rows = f.readlines()

    regime_list = self.parse_baseline(rows)

    return regime_list
```

redteam-project / sckg

Unwatch 3 | Unstar 9 | Fork 1

Code | Issues 9 | Pull requests 0 | Actions | Projects 0 | Wiki | Security | Insights | Settings

Branch: master ▾ sckg / sckg / etl / dod_cci.py / Jump to ▾ | Find file | Copy path

jason-callaway removed unneded import from cci etl e4e56fb 10 days ago

1 contributor

79 lines (64 sloc) | 2.16 KB | Raw | Blame | History

You're using code navigation to jump to definitions or references. Learn more or give us feedback

```

from lxml import objectify

from sckg.etl.generic import Generic

class DoDCCI(Generic):

  def __init__(self, config):
    super().__init__(config)

  def extract(self, regime, parsable_document):
    with open(parsable_document, 'r') as f:
      xml = f.read()

    root = objectify.fromstring(xml)

    j = []
    for item in root.cci_items.cci_item:
      id = item.attrib.get('id', '')
      control_list = []
      for reference in item.references.reference:
        control_list.append(reference.attrib.get('index'))
      j.append({
        'id': id,
        'controls': control_list
      })

    return j
```

- Python 3 required
- virtualenv recommended
- `cd sckg`
- `virtualenv venv`
- `source venv/bin/activate`
- `pip install -r requirements`
- Edit secrets/neo4j.yml

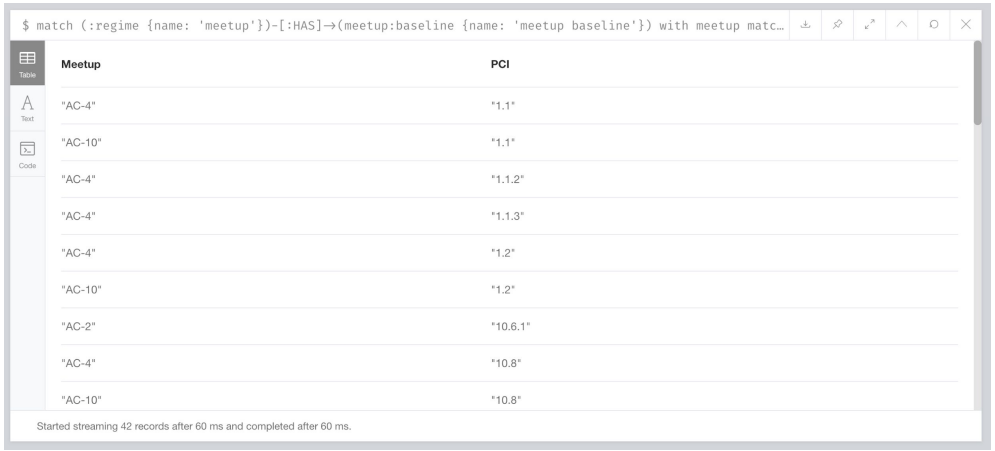
```
---
url: bolt://localhost:7687
username: neo4j
password: trevorforget
```

- Comment out list under regimes object in config.yml EXCEPT 800_53
  - See Issue #13
- Edit private.yml

```
---
regimes:
  - name: 'meetup'
    description: 'meetup regime'
    document:
      source:
'https://gist.github.com/jason-callaway/bd1bfcefc693f770b1463e877
42844aa'
      parsable: 'data/regimes/private/meetup.tsv'
    baseline:
      regime_name: 'meetup'
      baseline_name: 'meetup regime'
      control_regime: '800_53'
      uid_key: 'id'
```

- No etl object means default class and ETL methods
- `python build.py`

```
match (:regime {name: 'meetup'})-[:HAS]->(meetup:baseline {name: 'meetup
baseline'})
with meetup
match (:regime {name: 'NIST CSF'})-[:HAS*]-(csf:baseline)
with meetup, csf
match (:regime {name: 'NIST 800-53'})-[:HAS*]->(nist:control)
with meetup, csf, nist
match (:regime {name: 'PCI DSS'})-[:HAS*]->(pci:control)
with meetup, csf, nist, pci
match (meetup)-[:REQUIRES]->(nist), (csf)-[:REQUIRES]->(nist),
(csf)-[:REQUIRES]->(pci)
return nist.name as Meetup, pci.name as PCI order by PCI
```



Branch: master ▾ sckg / CONTRIBUTING.md

Find file | Copy path

jason-callaway added contributing md file 5404204 10 days ago

1 contributor

11 lines (8 sloc) | 601 Bytes

Raw | Blame | History

Contributing to sckg

There are lots of ways to help!

- Build your own graph and open Issues if you find something's wrong
- Have a good idea? Open an RFE Issue
- Add another regime to the graph. See the hacking page for details on how to get started
- Help us out by starring this repo or by following us on Twitter
- Come to one of our meetups in person or remote via Google Hangout

(This doc is under construction. Check back soon for updates.)