python-stix primer

Ben Schmoker

github.com/bschmoker

What is python-stix?

- Developer friendly
  - Python objects > raw XML
- Re-usable
  - Open-source libraries
- Plug-in ready
  - Integrate with existing tools

Let's get started!

- Install Python 2.7 and dependencies

```
apt-get install python-dev python-pip
apt-get install libxml2-dev libxslt-dev
apt-get install zlib1g-dev

pip install stix
```

Create a STIX document

```
$ cat > write.py
from stix.core import STIXPackage, STIXHeader
header = STIXHeader ()
header.title = "My first document!"

pkg = STIXPackage()
pkg.stix_header = header
print pkg.to_xml() // output XML
```

Generate a STIX Indicator

- The following slides will reference this example code

Create IP Address Indicator

```
$ cat >> write.py
ind = Indicator()
ind.title="malicious IP"
ind.add_indicator_type("IP Watchlist")

// set value
addr = Address()
addr.address_value="10.0.0.0"
addr.category = 'ipv4-addr'
addr.condition = "Equals"

// add to package
ind.add_observable(addr)
stix_package.add_indicator(ind)
```

github.com/STIXProject/python-stix

Add optional fields

```
$ cat >> write.py
// add a type of malicious activity
activity = TTP(title="C2 Behavior")
stix_package.add_ttp(activity)

//link indicator to activity
ind.add_indicated_ttp(TTP(idref = activity.id_) )
```

Parsing STIX

- The following slides will reference this example code

Load a STIX document

```
$ curl http://tiny.cc/samplestix > in.xml

$ python
from stix.core import STIXPackage, STIXHeader
myfile = open('in.xml')
pkg = STIXPackage.from_xml(myfile)
```

Access Data Elements

```
$cat in.xml
<stix:STIX_Package
<stix:Package_Intent>Incident
<stix:Description>Sample breach report
</>

$ cat >> read.py
print pkg.stix_header.description
```

Iterate Lists

```
$cat in.xml
<stix:Incident>
<incident:Title>Breach of Cyber Tech Dynamics
</>

$ cat >> read.py
for inc in pkg.incidents:
   print inc.title
```

Parsing STIX (Advanced)

- The following slides will reference this example code

Examine Observables

```
$curl > in.xml
<stix:Indicator>
<indicator:Observable>
<cybox:Object>
<cybox:Properties>
<FileObj:Hashes>
<cyboxCommon:Hash>d3adb33f
</>

$ cat > read.py
for ind in pkg.indicators:
        for obs in ind.observables:
                for digest in obs.object_.properties.hashes:
                    print digest
```

Dereference Links

```
$cat in.xml
<stix:TTPs>
<stix:TTP id="id_value">
[...]
</>
<stix:Indicator>
<indicator:Indicated_TTP>
<stixCommon:TTP idref="id_value">
</>

$ cat >> read.py
relationship_dict = {}
for ttp in package.ttps.ttps:
          relationship_dict [ttp.id_] = ttp # assign object to dictionary value,
with ID as key

for rel_ttp in indicator.indicated_ttps:
         if rel_ttp.item.idref in ttps: # look up object by ID
                  print relationship_dict[rel_ttp.item.idref].title
```

github.com/STIXProject/python-stix

Further Reading

- Sample code and use cases
  - stixproject.github.io/documentation/idioms
- Python documentation
  - stix.readthedocs.org
- Pandas documentation
  - https://didatica.tech/o-pacote-pandas-python-para-machine-learning/