Cyber Prediction User Manual

Dr. Andrew J. C. Blyth, PhD.[1]

ADISA Research Centre (ARC), ADISA, Thrales End Business Centre, Thrales End Lane, Harpenden, AL5 3NS
andrew.blyth@adisa.global

Document Control:	
Author(s):	Dr. Andrew Blyth (andrew.blyth@adisa.global)
Version:	1.0
Date:	1st August 2019
ADISA Doc. Ref:	/20190801/1.0
Copyright Info:	ADISA 2019

Table of Contents

Cyber Prediction User Manual 1
Dr. Andrew J. C. Blyth, PhD.
1 Software Tools and Libraries 3
2 Useability Instructions 3

1 Software Tools and Libraries

The development of the cyber attack prediction software makes use of the following libraries:

- C++ and the gcc compiler
  - Apple LLVM version 10.0.1 (clang-1001.0.46.4)

- Python 3.7.3, and the libraries and modules used include:
  - xmltodict - version 0.12.0
  - json - version 3.16.0
  - stix2 - version 1.1.2
  - pyqtgraph - version 0.10.0
  - requests - version 1.21.0
  - numpy - version 1.16,2
  - gremlinpython - version 3.2.6

2 Useability Instructions

When invoking tools please note that: a) *python* is a link to the Python 3.7.3 interpreter, b) the command *cls* is a link to the *clear* command, c) that the command *make* is a link to *GNU Make 3.81*, and d) the *g++* compiler is used.

- artificallifeengine
  - This is the Artificial Life Synthetic Engine defined in Figure 1. It is the software component that executed the particle swarm optimisation algorithm (PSO).

    ```
    $ cls
    $ python ale.py ./utfunctions
    ```

- cefjsonrpc
  - This is the CEF particle ingestor and is part of the Event Collector Connector depicted in Figure 1. This tool can be invoked via the following command within the *cefjsonrpc* directory.

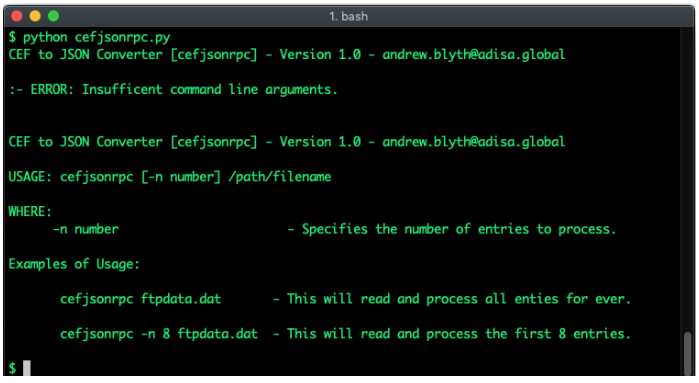

Fig. 1. The CEF Tool

In the above example we are processing the first CEF event contained in the file *./data/cef.dat*.

– datajsonrpc
  - This is the data particle ingestor and is part of the Particle Generator depicted in Figure 1.This tool can be invoked via the following command within the *datajsonrpc* directory.

```
$ cls
$ python datajsonrpc.py ../data/dataparticle.xml
```

– debugger
  - This is the cloud and enterprise service debugging tool that we developed to validate the correction insertion of data into the JanusGraph and the cache via the enterprise service bus.

    It makes use of an JSON RPC interface on the enterprise service bus to achieve this. This tool can be invoked via the following command within the *debugger* directory.

```
$ cls
$ python debugger.py
```

– dnsjsonrpc
  - This is the DNS particle ingestor and is part of the Event Collector Connector depicted in Figure 4. This tool can be invoked via the following command within the *dnsjsonrpc* directory.

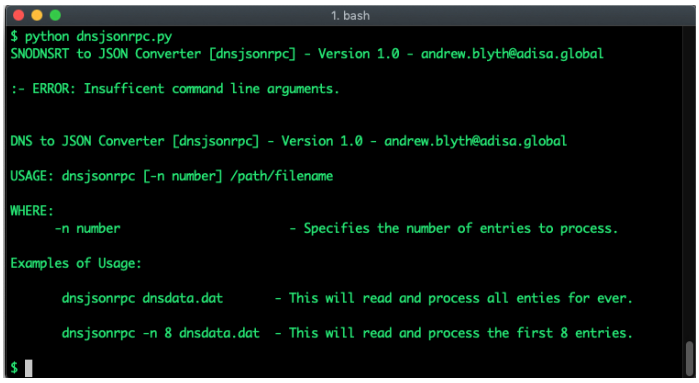

Fig. 2. The DNS Tool

– enterpriseservicebus
  - This is the JSON RPC Interface into the cloud environment and the cache. The following is the list of some of the JSON RPC interfaces supported in the Enterprise Service Bus defined in Python.

```
dispatcher["addParticle"]
dispatcher["addDataParticle"]
dispatcher["getParticle"]
dispatcher["getAllSyslogParticles"]
dispatcher["getAllSnortParticles"]
dispatcher["getAllTCPDParticles"]
dispatcher["getAllSSHDParticles"]
dispatcher["getAllCEFarticles"]
dispatcher["getAllVSFTPDParticles"] )
dispatcher["getAllDNSParticles"]
dispatcher["getAllNCSAParticles"]
dispatcher["getAllNetFlowParticles"]
dispatcher["getAllFTPDParticles"]
dispatcher["getAllEvtXParticles"]
dispatcher["getAllSTIXParticles"]
```

- The enterprise service bus can be invoked via the following command within the *enterpriseservicebus* directory.

```
$ cls
$ python ESB.py
```

– icmpjson
- This is the ICMP particle ingestor and is part of the Event Collector Connector depicted in Figure 1. This tool can be invoked via the following command within the *icmpjson* directory.

```
$ cls
$ Bin/icmpjson
```

– ftpdjsonrpc
- This is the FTP particle ingestor and is part of the Event Collector Connector depicted in Figure 1. This tool can be invoked via the following command within the *ftpdjsonrpc* directory.

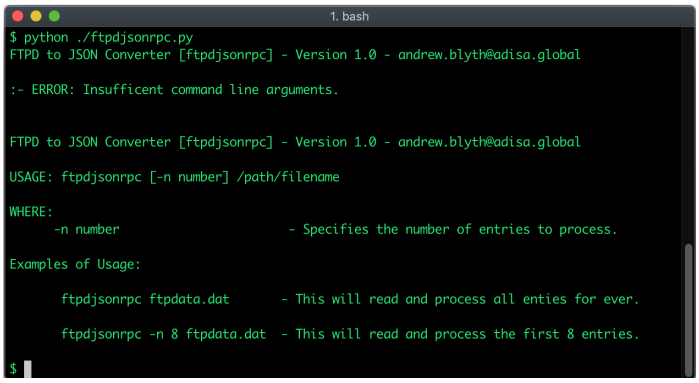

Fig. 3. The FTPD Inserter Tool

– insertjsonrpc
  - This is a generic JSON RPC inserter that will read JSON data on the command line and insert into the cloud environment via the Enterprise Service Bus. In particular, this tool is used to inject IP/TCP/UDP and ICMP particles.

  - This tool can be invoked via the following command within the *insertjsonrpc* directory.

```
$ cls
$ make
```

– ipjson
  - This is the IP particle ingestor and is part of the Event Collector Connector depicted in Figure 1. This tool can be invoked via the following command within the *ipjson* directory.

```
$ cls
$ Bin/ipjson
```

– simulation
  - This is the simulation engine that was used to create the TRL 3 technical demonstrator that was used to validate the project. It reads an XML file that defines when and where the particles in the simulation are to be ingested into the cloud environment. This tool can be invoked via the following command within the *simulation* directory.

```
$ cls ; python simulation.py
```

– mweljsonrpc
  - This is the Microsoft Windows Event particle ingestor and is part of the Event Collector Connector depicted in Figure 1. This tool can be invoked via the following command within the *mweljsonrpc* directory.

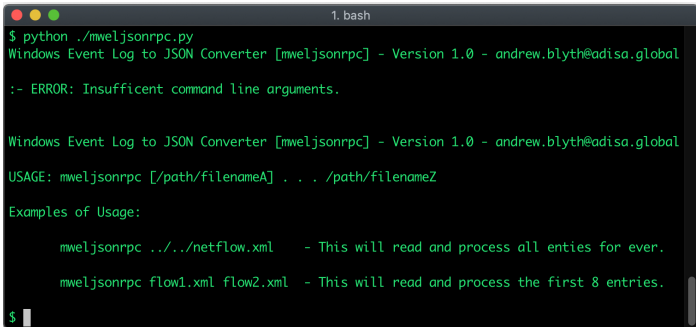

Fig. 4. The Microosft Windows Event Inserter Tool

– tcpjson

- This is the TCP particle ingestor and is part of the Event Collector Connector depicted in Figure 1. This tool can be invoked via the following command within the *tcpjson* directory.

```
$ cls
$ Bin/tcpjson
```

– ncsajsonrpc
- This is the NCSA particle ingestor and is part of the Event Collector Connector depicted in Figure 1. This tool can be invoked via the following command within the *ncsajsonrpc* directory.

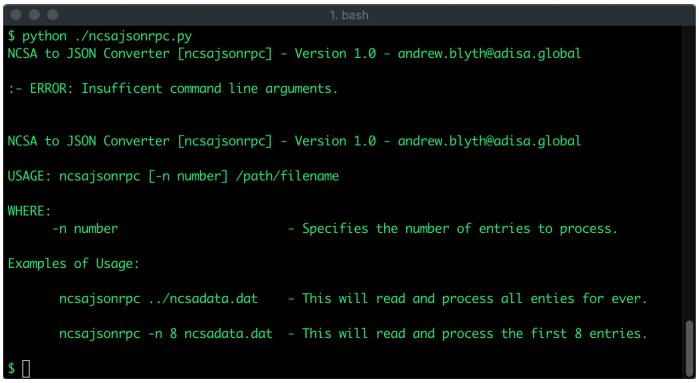

Fig. 5. The NCSA Event Inserter Tool

– netflowjsonrpc
- This is the NETFLOW particle ingestor and is part of the Event Collector Connector depicted in Figure 1. This tool can be invoked via the following command within the *netflowjsonrpc* directory.

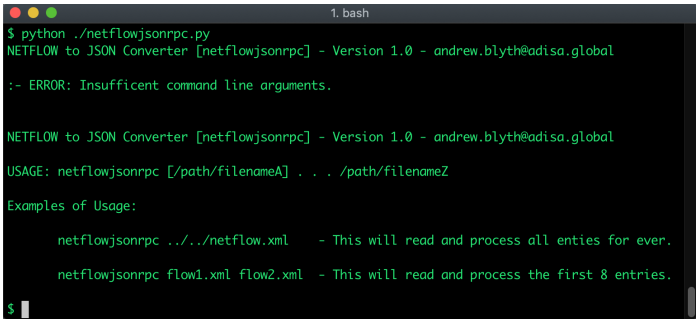

Fig. 6. The Netflow Event Inserter Tool

– snortjsonrpc
  - This is the SNORT particle ingestor and is part of the Event Collector Connector depicted in Figure 4. This tool can be invoked via the following command within the *snortjsonrpc* directory.

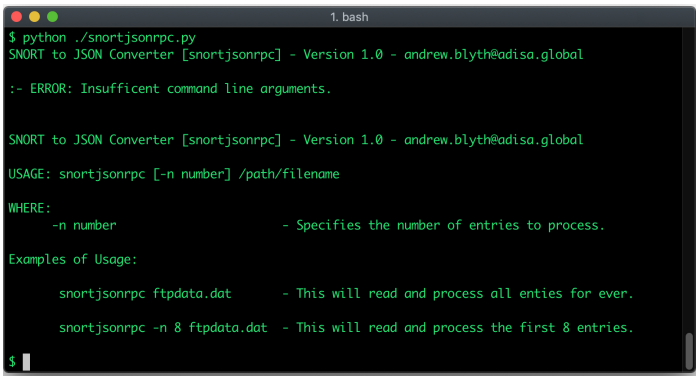

Fig. 7. The Snort Event Inserter Tool

– sshdjsonrpc
  - This is the SSHD particle ingestor and is part of the Event Collector Connector depicted in Figure 1. This tool can be invoked via the following command within the *sshdjsonrpc* directory.

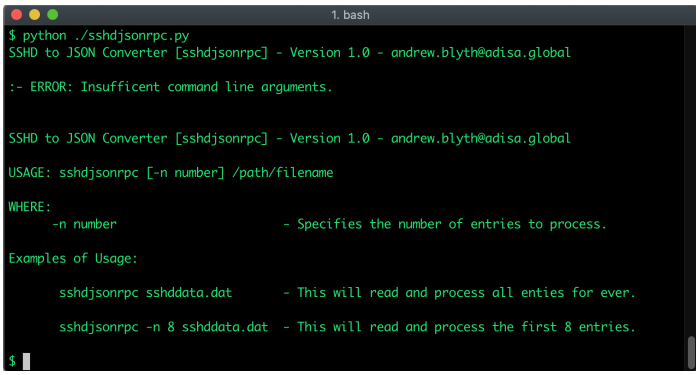

Fig. 8. The SSHD Event Inserter Tool

– syslogjsonrpc
  - This is the SYSLOG particle ingestor and is part of the Event Collector Connector depicted in Figure 1. This tool can be invoked via the following command within the *syslogjsonrpc* directory.

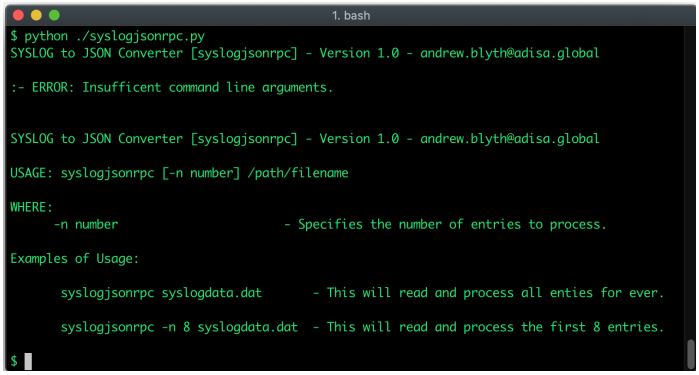

Fig. 9. The Syslog Event Inserter Tool

– tcpdjsonrpc
  - This is the TCPD particle ingestor and is part of the Event Collector Connector depicted in Figure 1. This tool can be invoked via the following command within the *tcpdjsonrpc* directory.

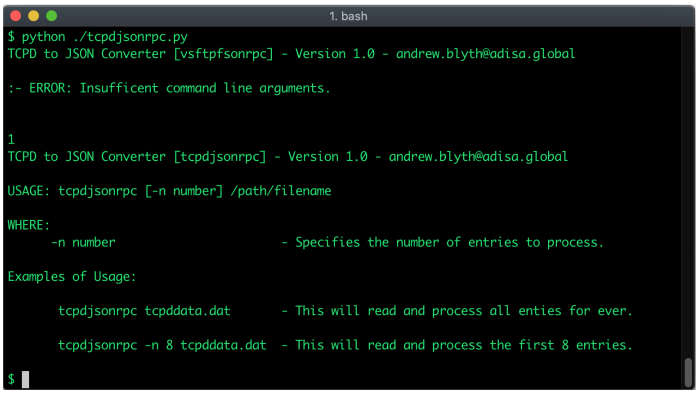

Fig. 10. The TCPD Event Inserter Tool

– threatintelligenceengine
  - This is the threat intelligence engine that manages the STIX and TAXII objects and it is part of the Threat Intelligence Engine depicted in Figure 1.

  - This tool can be invoked to create the JSON STIX/Prediction particle definition via the following command within the *threatintelligenceengine* directory.

```
$ rm -rf *.json
$ cls
$ python stix.py > stix.json
```

- This tool can be invoked to inject the JSON STIX/Prediction particles into the cloud environment via the following command within the *threatintelligenceengine* directory.

```
$ cls
$ python stix.py
```

– udpjson
- This is the UDP particle ingestor and is part of the Event Collector Connector depicted in Figure 1. This tool can be invoked via the following command within the *udpjson* directory.

```
$ cls
$ Bin/udpjson
```

– vizswarm
- This is the set of visualisation tools that allows you to visualise the behaviour of particles; it is part of the Particle Visualisation Engine depicted in Figure 4. The three visualisations developed for this project are:
  * The Cyber Swarm Particle Collider (CSPC). This tool can be invoked via the following command:

```
$ make swarm
```

  * The Syslog particles insertion rate viewer. This tool can be invoked via the following command:

```
$ make log
```

  * The Non-Syslog particles insertion rate viewer.This tool can be invoked via the following command:

```
$ make nlog
```

– vsftpdjsonrpc
- This is the VSFTPD particle ingestor and is part of the Event Collector Connector depicted in Figure 4. This tool can be invoked via the following command:

```
$ cls
$ python vsftpdjsonrpc.py
```