Utilities

Ivan Kirillov

13 July 2012

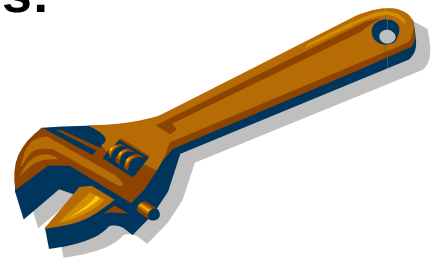
MITRE

Agenda

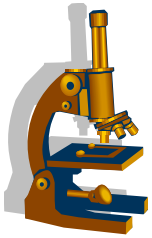
- The MAEC Toolset
- Use Case Example
- Future Tools
- MAEC Community Resources

MAEC Tools Overview

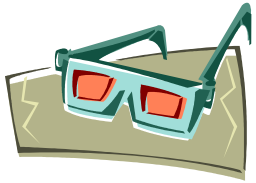
- **Three Different Classes of Tools:**
  - **Content Creation/Manipulation**



  - **Analysis**



  - **Views**



MAEC Tool Outreach and Collaboration

- **Tool vendors/producers supported our development of MAEC translators:**
  - **GFI Sandbox : GFI Software**
  - **ThreatExpert : Symantec**
  - **Anubis : International Secure Systems (ISEC) Lab**
  - **FireEye**
  - **Norman Sandbox : Norman Data Defense Systems**
- **Discussions with tool vendors about adopting MAEC as a native output format (under NDAs)**
- **Open Source projects**
  - **Honeynet.org honeyclient project (thug)**
  - **Cuckoobox sandbox**

Current MAEC Toolset

Tool	Class	Language	Current Support	Author	License
MAEC/CybOX Bindings	Bindings	Python	MAEC v2.1/CybOX 1.0	MITRE	New BSD
Anubis → MAEC	Translator	Python	MAEC v2.1/CybOX 1.0	MITRE	New BSD
MAEC → OVAL	Translator	Python	MAEC v1.1/OVAL v5.7	MITRE	New BSD
ThreatExpert → MAEC	Translator	Python	MAEC v1.1	MITRE	New BSD
GFI Sandbox → MAEC	Translator	Python	n/a - in development	MITRE	New BSD
FireEye → MAEC	Translator	Python	n/a - in development	MITRE	New BSD
Norman Sandbox →MAEC	Translator	Python	n/a - in development	MITRE	New BSD
MAEC→ HTML	Translator	XSL	MAEC v1.1	MITRE	New BSD
MAEC Comparator	Analysis	Python	MAEC v1.1	Blake Hartstein	n/a
PEFile (Python Module)	Native Output	Python	n/a – in development	MITRE	New BSD
CuckooBox (Sandbox)	Native Output	Python	MAEC v1.1	Cuckoo Team	GNU GPL v3
Thug (Honeyclient)	Native Output	Python	MAEC v1.1	Angelo Dell'Aera	GNU GPL v2

MAEC Schema Bindings

- **Permit:**
  - **Creation of new MAEC content**
  - **Manipulation of existing MAEC content**

- **Currently for Python 2.x**
  - **Full CybOX 1.0 draft support**
  - **Created with GenerateDS**
    - **http://cutter.rexx.com/~dkuhlman/generateDS.html**

MAEC Schema Bindings – Content Creation Example I

```
import maec_2_1 as maec
import cybox.file_object_1_2 as file_object

#Create a MAEC Bundle
bundle = maec.BundleType(id='maec-example-bnd-1',schema_version='2.1')

#Create an Action
action = maec.ActionType(id='maec-example-act-1')
#Set the Action Properties
#Set the Action Type
action.set_type('Create')
#Set the Action Name
action_name = maec.cybox.ActionNameType(Defined_Name = 'Create File')
action.set_Action_Name(action_name)
#Set the Action Context
action.set_context('Host')

#Create an Associated Object for the Action
associated_object = maec.cybox.AssociatedObjectType(id='maec-example-obj-1',
association=‘Affected')
#Create the Defined Object for the Associated Object
defined_object = file_object.FileObjectType()
```

MAEC Schema Bindings – Content Creation Example II

```
#Add some attributes to the Defined Object
defined_object.set_File_Name(maec.common.StringObjectAttributeType(datatype='String',
valueOf_='barfoo.exe'))
defined_object.set_File_Path(maec.common.StringObjectAttributeType(datatype='String',
valueOf_='c:\windows\system32'))\
#Set the xsi:type for the Defined Object
defined_object.set_anyAttributes_({'xsi:type':'FileObj:FileObjectType'})
#Add the Defined Object to the Associated Object
associated_object.set_Defined_Object(defined_object)

#Add the Associated Object to the Action
associated_objects = maec.cybox.AssociatedObjectsType()
associated_objects.add_Associated_Object(associated_object)
action.set_Associated_Objects(associated_objects)

#Add the action to the MAEC Bundle
actions = maec.Actions()
actions.add_Action(action)
bundle.set_Actions(actions)

#Export the MAEC Bundle to an output file
outfile = file('example.xml',mode='w')
bundle.export(outfile, 0, namespacedef_='‘namespace declarations go here’)
```

MAEC Schema Bindings – Content Creation Example Output

```
<maec:MAEC_Bundle id="maec-example-bnd-1" schema_version="2.1">
  <maec:Actions>
    <maec:Action context="Host" type="Create" id="maec-example-act-1">
      <cybox:Action_Name>
        <cybox:Defined_Name>Create File</cybox:Defined_Name>
      </cybox:Action_Name>
      <cybox:Associated_Objects>
        <cybox:Associated_Object id="maec-example-obj-1" association_type="Affected">
          <cybox:Defined_Object xsi:type="FileObj:FileObjectType">
            <FileObj:File_Name datatype="String">barfoo.exe</FileObj:File_Name>
            <FileObj:File_Path datatype="String">c:\windows\system32</FileObj:File_Path>
          </cybox:Defined_Object>
        </cybox:Associated_Object>
      </cybox:Associated_Objects>
    </maec:Action>
  </maec:Actions>
</maec:MAEC_Bundle>
```

MAEC Schema Bindings – Content Parsing Example

```
import maec_2_1 as maec

#Parse the input file to get the MAEC Bundle object
input_file = 'example.xml'
maec_bundle = maec.parse(input_file)

#Extract the top-level actions from the MAEC Bundle
maec_actions = maec_bundle.get_Actions()

#Print the association type of each Associated Object in each Action
for action in maec_actions.get_Action():
    associated_objects = action.get_Associated_Objects()
    for associated_object in associated_objects.get_Associated_Object():
        print associated_object.get_association_type()
```

MAEC Content Creation/Translators

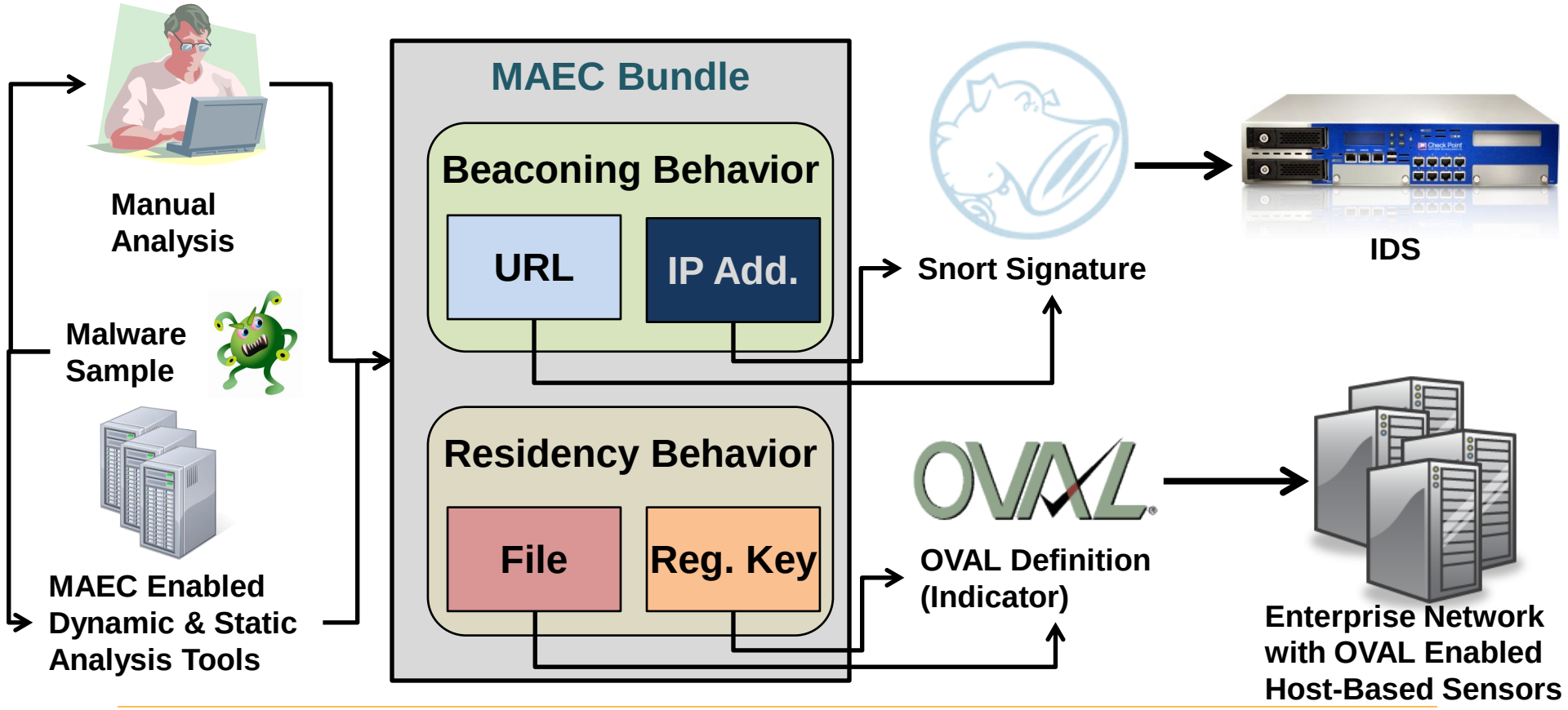
- **Intended to facilitate creation of MAEC content as POC, or for specific use cases**

Tool	Availability*	Input	Output
Anubis → MAEC	Now	Anubis XML	MAEC XML
ThreatExpert → MAEC	07/2012	TE XML	MAEC XML
GFI Sandbox → MAEC	07/2012	GFI XML	MAEC XML
FireEye → MAEC	08/2012	FireEye XML	MAEC XML
PE File (Python Module)	08/2012	PE File	MAEC XML
Norman Sandbox → MAEC	09/2012	Norman XML	MAEC XML
MAEC → OVAL	09/2012	MAEC XML	OVAL XML
MAEC → HTML	09/2012	MAEC XML	HTML

*For MAEC v2.1/CybOX v1.0 support

Making MAEC Operational

- MAEC enables comprehensive malware descriptions, allowing various components of a MAEC bundle to be used as signatures and indicators in the enterprise



MAEC & Host Based Detection

Dynamic Analysis Engine

- Anubis
- GFI Sandbox
- CuckooBox
- Etc.

Engine Output

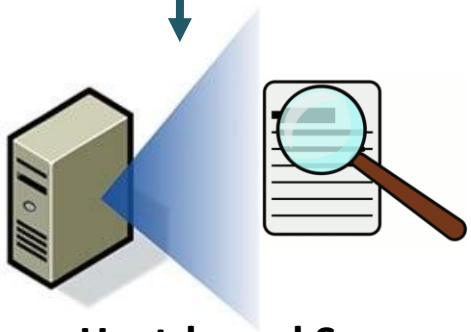
Native MAEC Output

MAEC™

OVAL®

Malware Binary

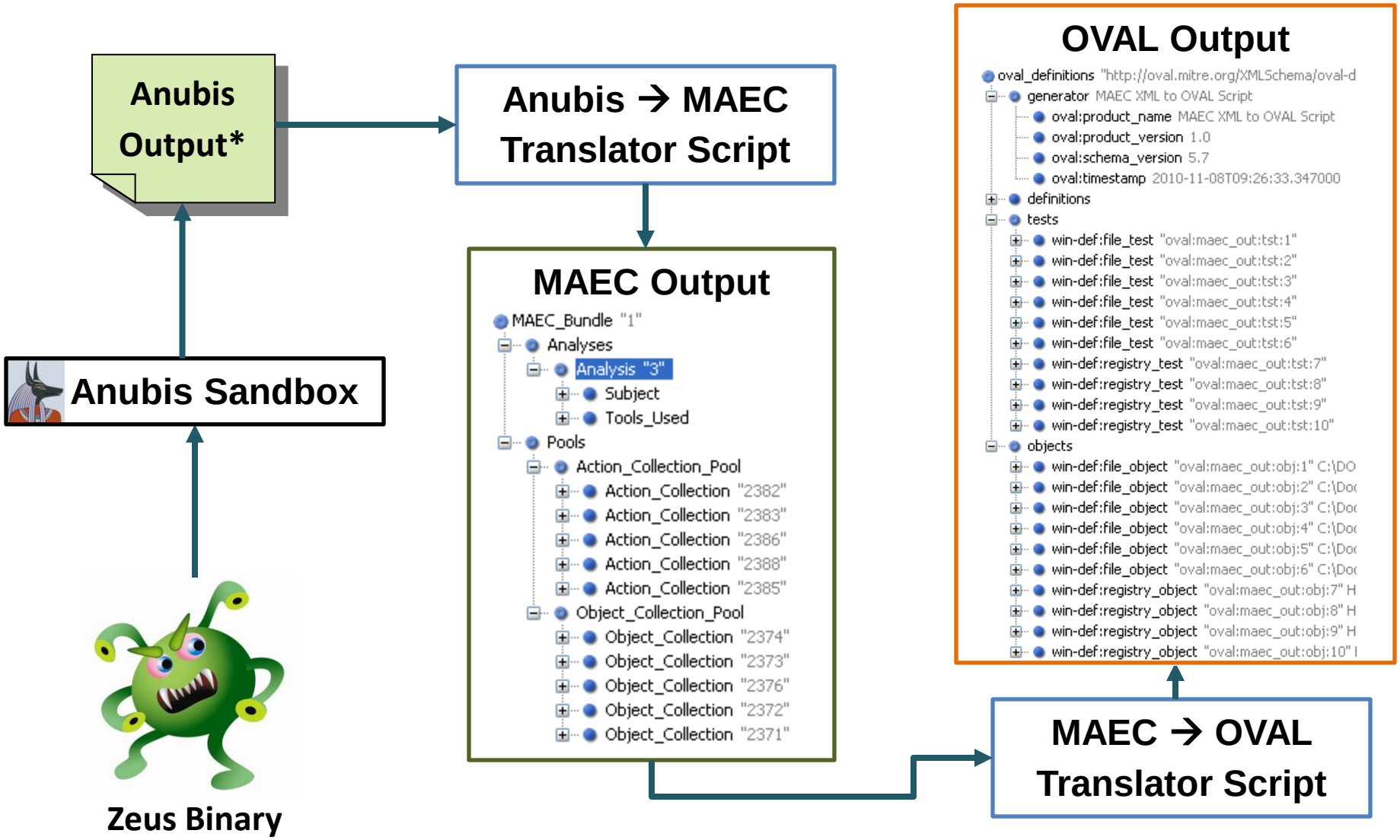
Sandbox → MAEC Translator



Host-based Scanner

MITRE

MAEC & Host Based Detection Example: MAEC & Zeus



*http://anubis.iseclab.org/?action=result&task_id=1167a57d1aa905e949df5d5478ab23bf9

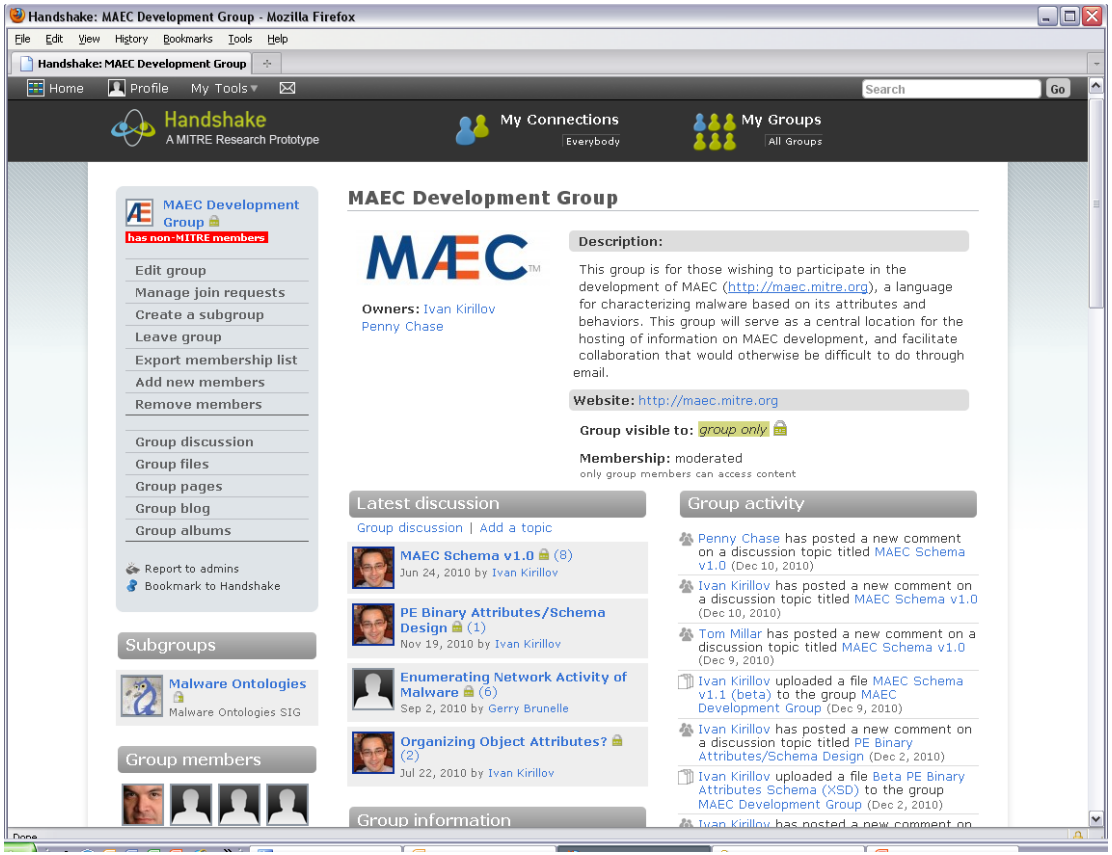
Future MAEC Tools

- **MAEC API**
  - **Allow users to generate valid/usable MAEC content without perfect knowledge of the schema**
  - **Current 'MAEC Helper' is a very simple, early take on this concept**
- **MAEC Bundle Management**
- **MAEC View Construction**

MAEC Community: Github Repository

- **Primary repository for MAEC scripts and tools**
  - **Currently contains:**
    - **Python Bindings**
    - **Anubis → MAEC Translator (MAEC v2.1/CybOX v1.0 draft)**
    - **ThreatExpert → MAEC Translator (MAEC v1.1)**
    - **MAEC → OVAL Translator (MAEC v1.1/OVAL v5.7)**
    - **MAEC → HTML Transform (MAEC v1.1)**
    - **Comparator (MAEC v1.1)**
- **Wiki contains code snippets and other useful information**
- **https://github.com/MAECProject/Tools**

MAEC Community: MAEC Development Group on Handshake



- **MITRE hosts a social networking collaboration environment: https://handshake.mitre.org**
- **Supplement to mailing list to facilitate collaborative schema development**
- **Malware Ontologies SIG Subgroup**

Questions?