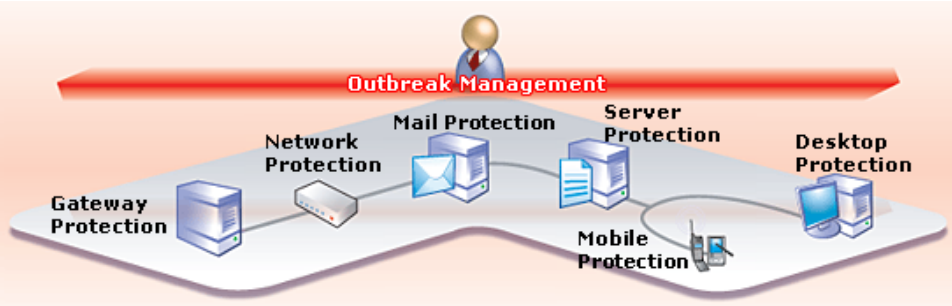
Ivan Kirillov

March 24th, 2011

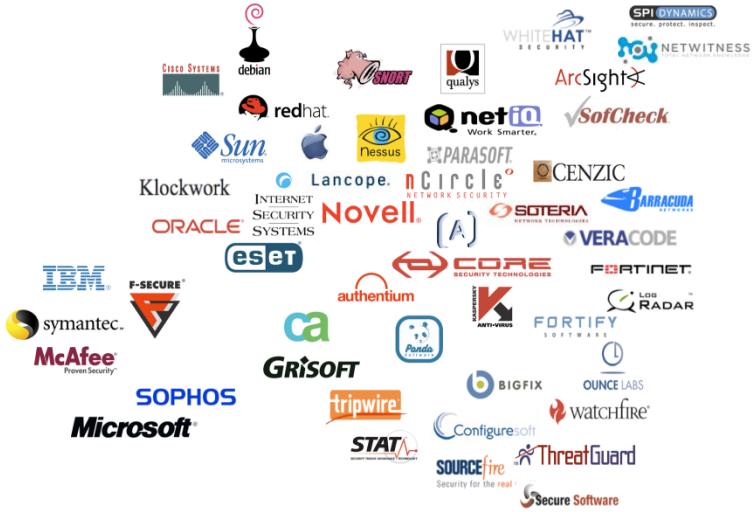
MITRE

Why Do We Need to Develop Standards for Malware?

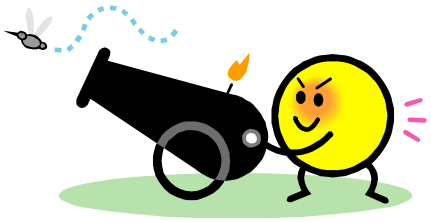
Multiple layers of protection



Lots of products



Inconsistent reports

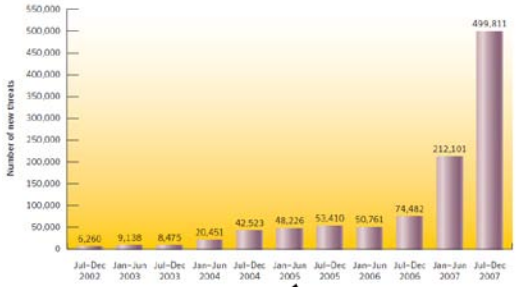


There's an arms race

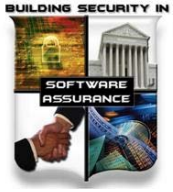
Background

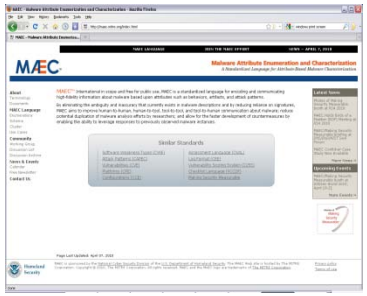
Rise of New Threats

Symantec Global Internet Security Threat Report, Volume XIII, 4/2008



Nimda or I-Worm or Readme?





Oct 2004

Initial CME discussions at VB Conference

Feb 2005

CME Submission Server

Oct 2005

CME public announcement and website

Jan 2007

39 CME IDs assigned

Feb 2007

DHS SwA Forum Malware WG

Dec 2009

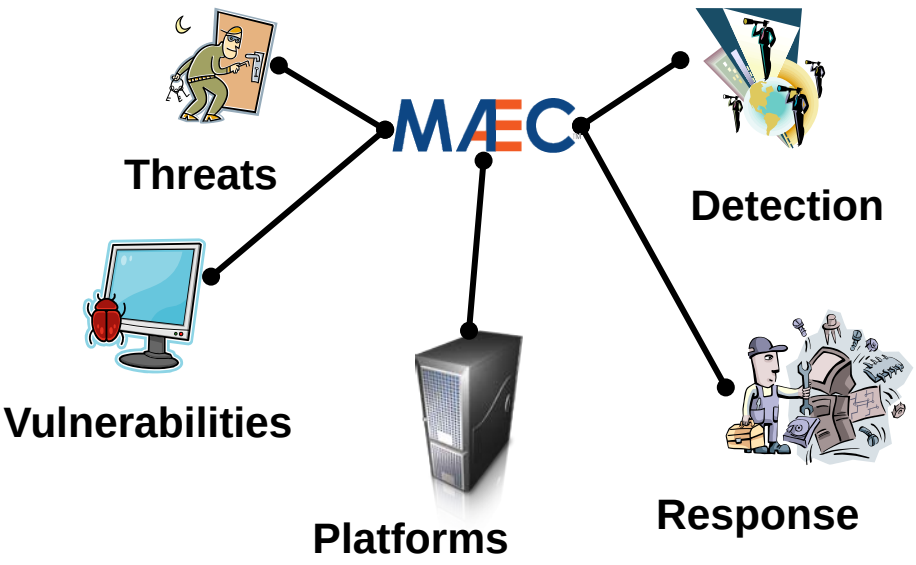
MAEC public website

Jun 2010

Initial MAEC Schema

CME → MAEC™

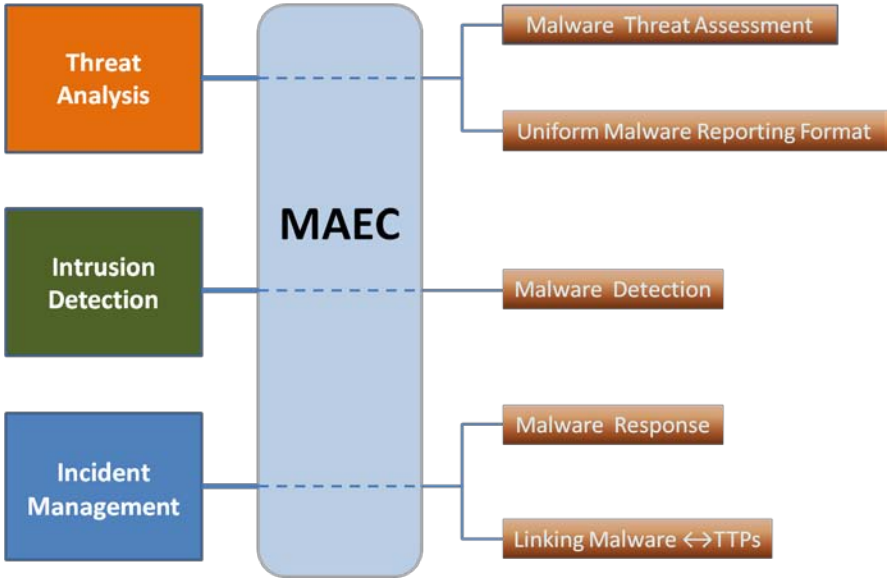
Malware Attribute Enumeration and Characterization (MAEC)



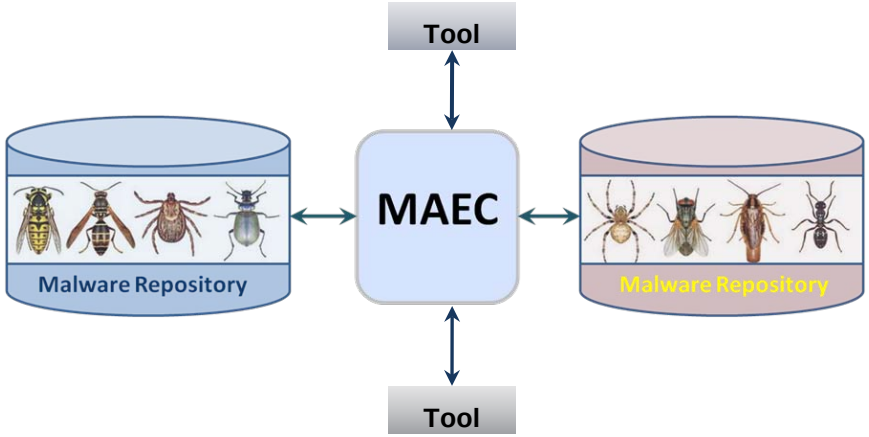
- Language for sharing structured information about malware
  - Grammar (Schema)
  - Vocabulary (Enumerations)
  - Collection Format (Bundle)
- Focus on attributes and behaviors
- Enable correlation, integration, and automation

MAEC Use Cases

- **Operational**

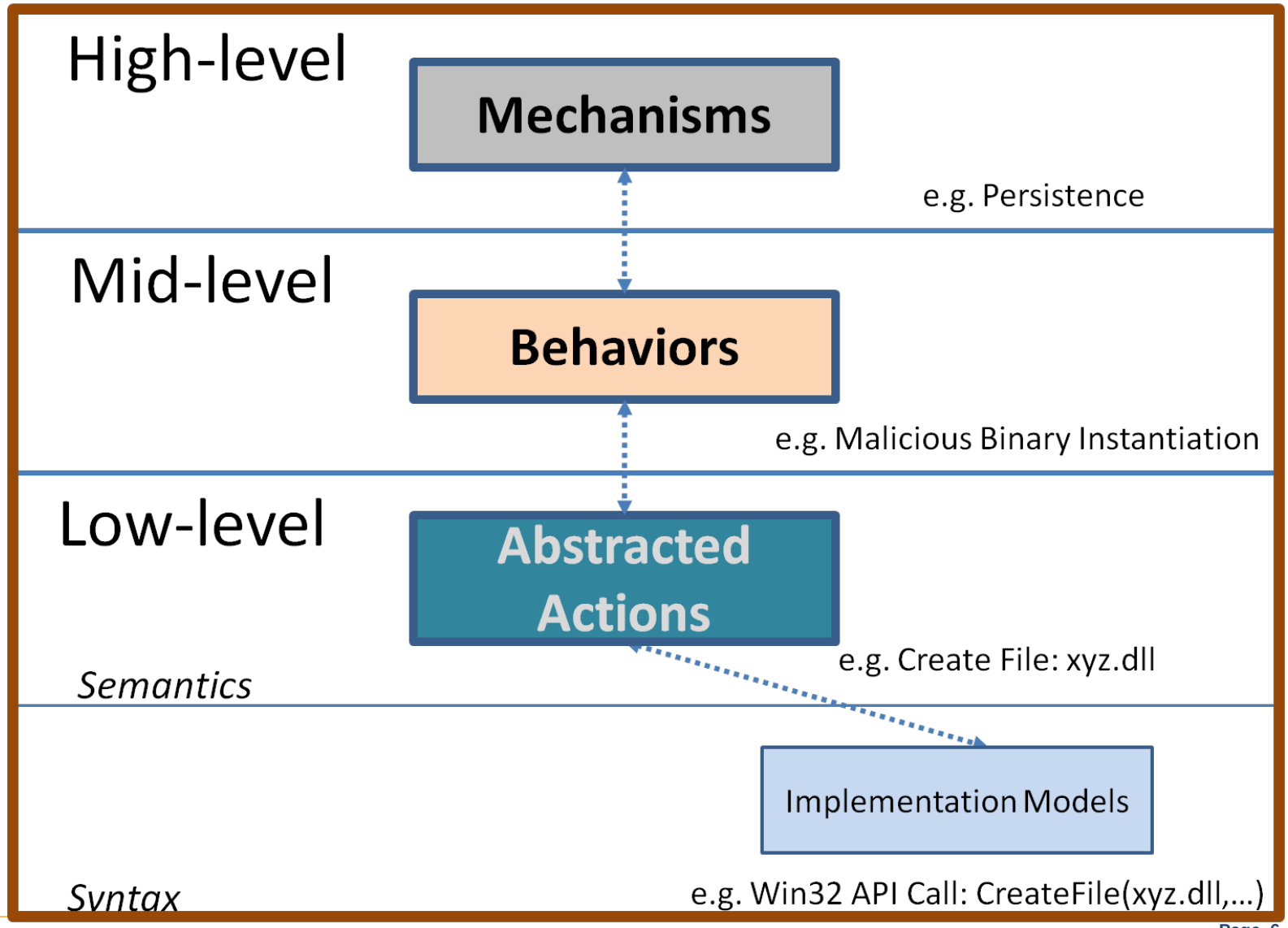


- **Analysis**
  - Help Guide Analysis Process
  - Standardized Tool Output
  - Malware Repositories



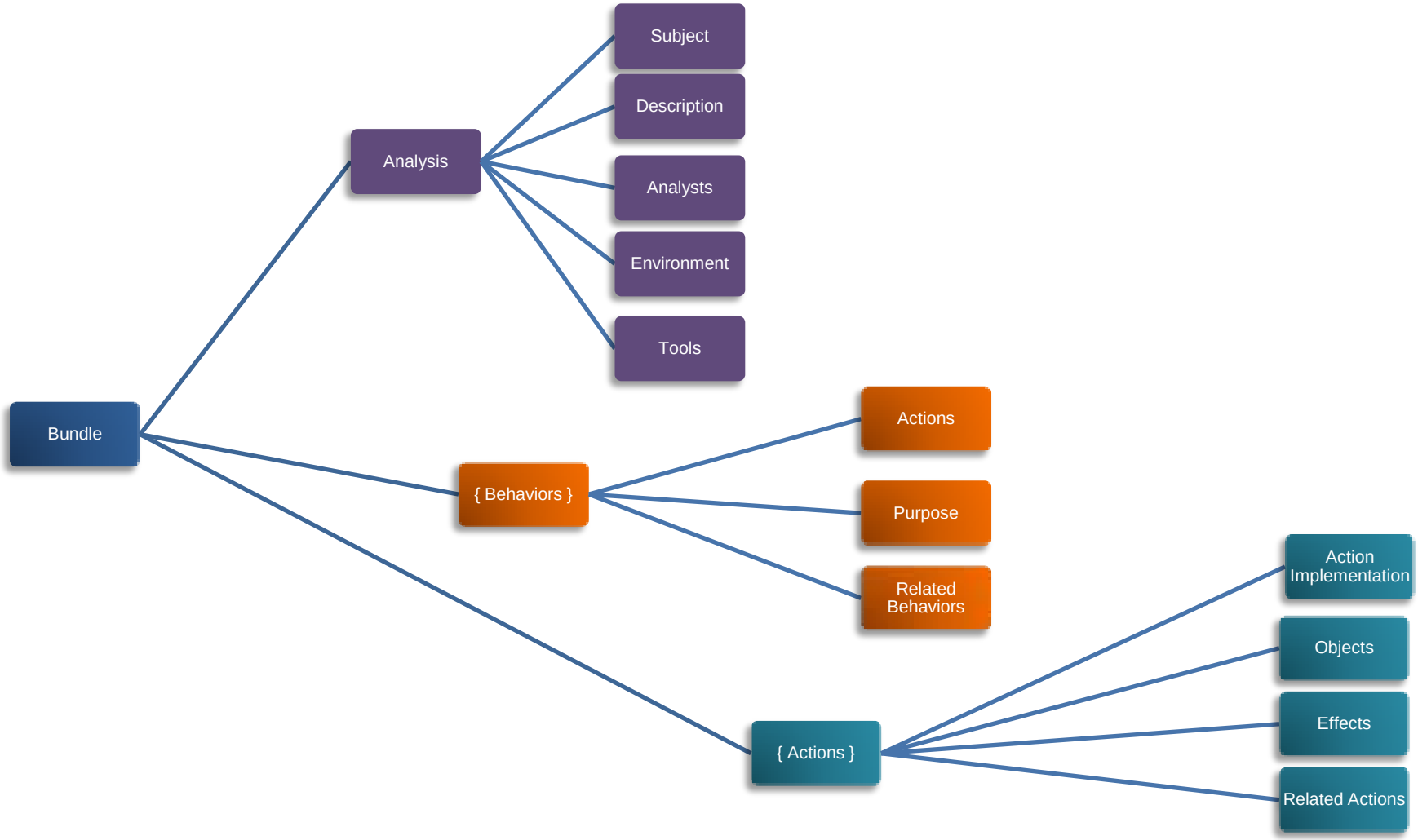
MITRE

MAEC Overview



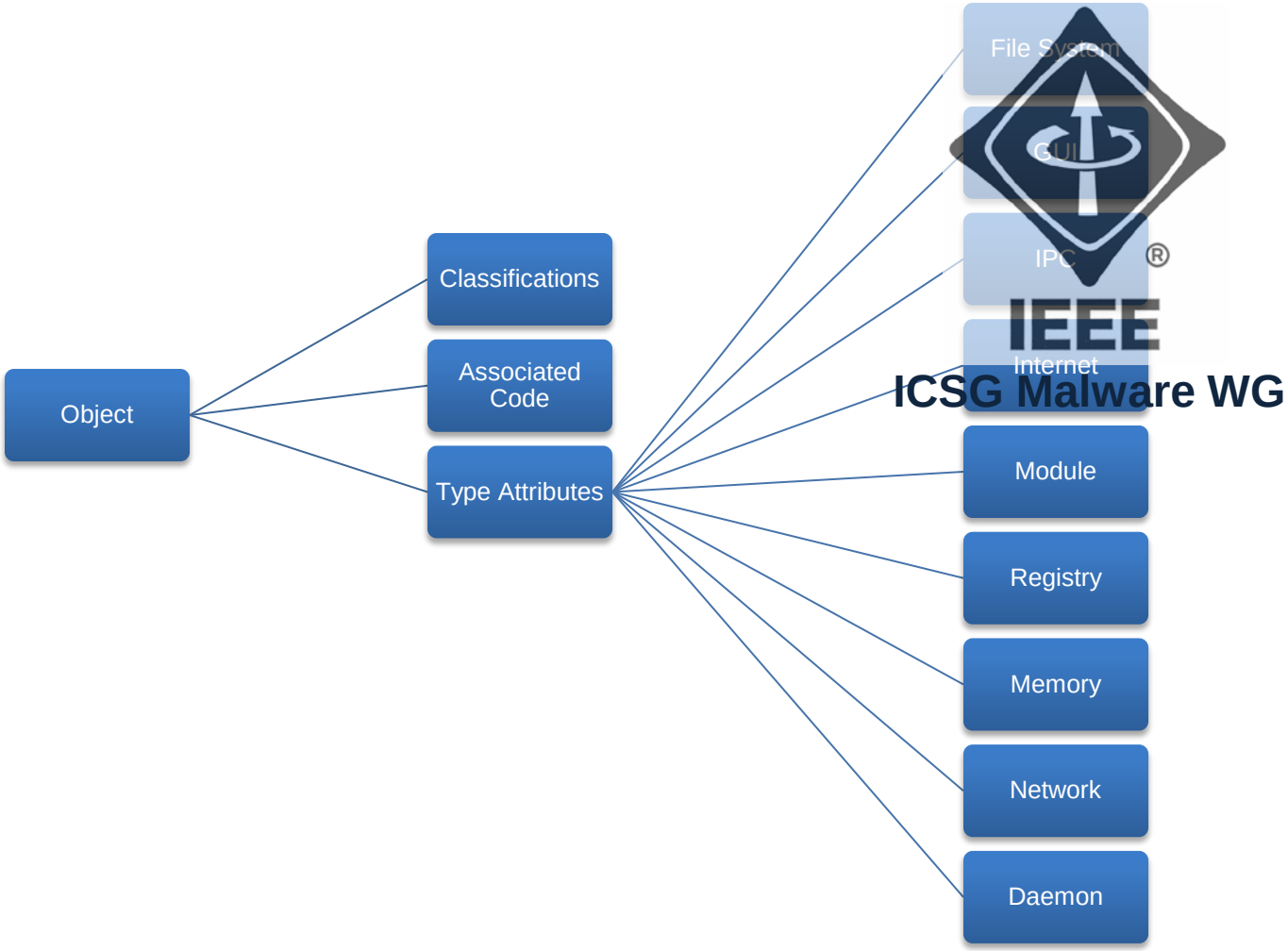
MITRE

MAEC Schema v 1.1 Overview



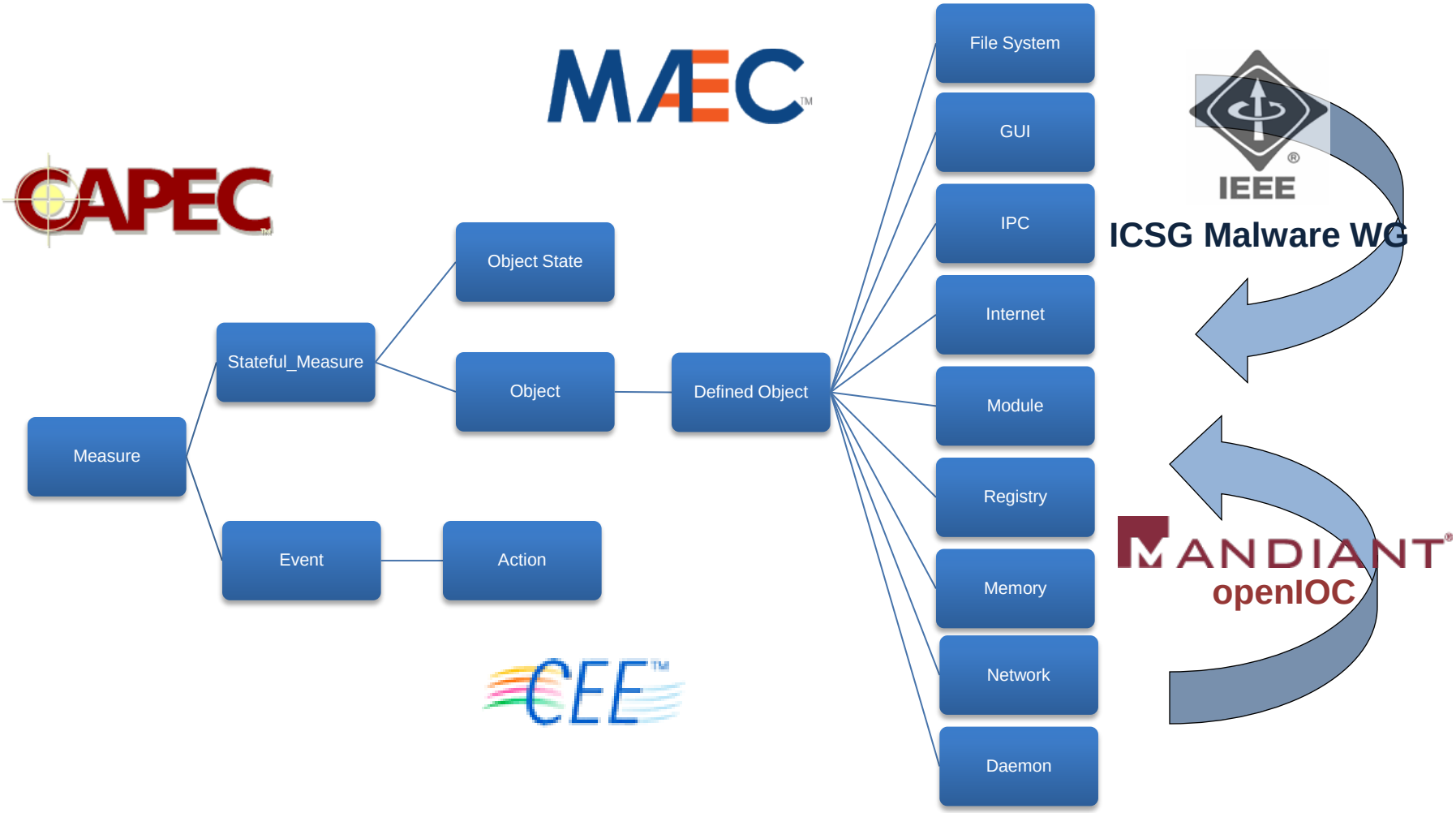
MITRE

MAEC Schema v 1.1 Objects



MITRE

Common Cyber Observables Schema



Community Engagement

- IEEE ICSG Malware Working Group
  - Developed Malware Metadata exchange schema to facilitate the sharing of sample data between AV product vendors
    - Attributes for AV classifications, source (URIs), object properties (file hashes, registry keys), boolean properties (isKernel, isPolymorphic)
  - MAEC currently imports the IEEE ICSG Malware Metadata exchange schema
  - The MAEC team has been invited to join the WG and develop the next version of the schema
- Industry Collaborations
  - Working with Mandiant on MAEC <-> openIOC
  - Tool vendors supported our development of MAEC translators:
    - CWSandbox : GFI Software
    - ThreatExpert : Symantec
    - Anubis : International Secure Systems (Isec) Lab

MAEC Schema Roadmap

- MAEC v 1.0
  - Analysis: Dynamic
  - Operational: Detection (Host-based through OVAL)
  - Schema Level: Host-based observables
- MAEC v 1.1 (current release)
  - Analysis: Static
  - Schema Level: Malware metadata
- Future Schemas
  - Additional attributes (Netflow, Layer 7 protocols)
  - In-Depth Analysis
    - Mid-level behaviors
  - Operational
    - Signature and Indicators of Compromise (IOCs) management
    - Mitigation and response support
  - Expressiveness (operators, constraints, relationships)

Use Case: Host Based Detection

Dynamic Analysis Engine

- Anubis
- CWSandbox
- ThreatExpert
- Etc.

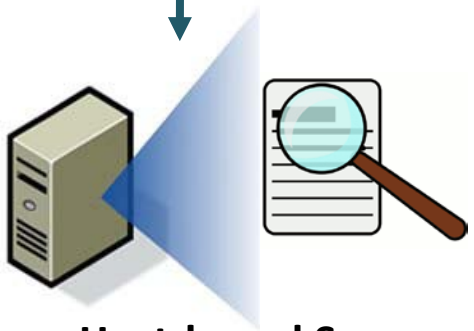
Engine Output

MAEC™

OVAL®

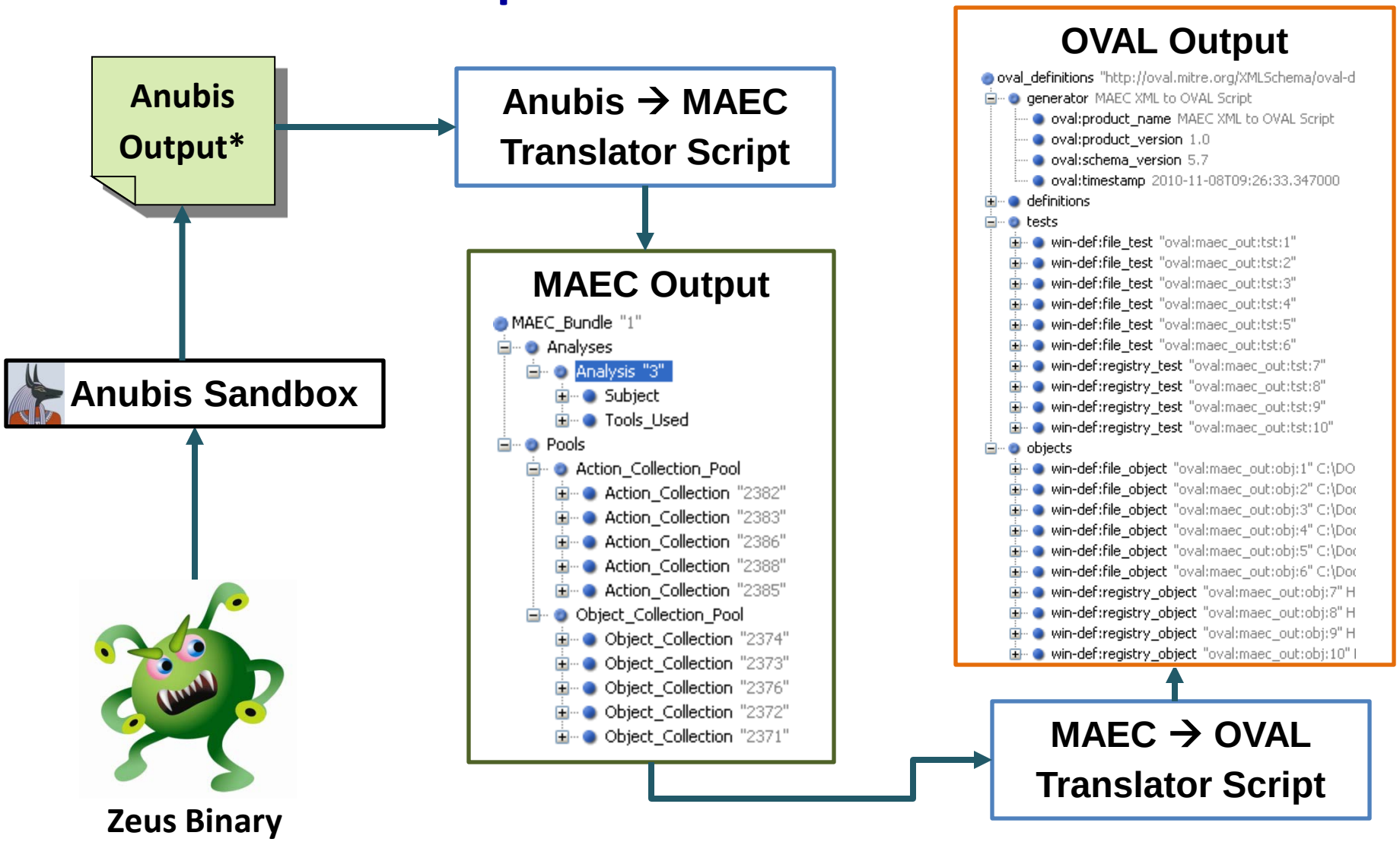
Malware Binary

Sandbox -> MAEC Translator



Host-based Scanner

Real World Example: MAEC & Zeus



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

MAEC & OVAL today

- **Using OVAL in combination with MAEC can augment existing anti-malware capabilities**
  - **Particularly useful for 0-day malware detection**

- **MAEC to OVAL script**
  - **Generates OVAL XML from MAEC XML documents**
  - **Allows for malware detection based on files & registry keys**
  - **Future support will add detection based on ports/IP addresses, processes, DNS cache, and services**
  - **Available on MAEC's Handshake group**
    - **Email the MAEC team at maec@mitre.org for access**

MAEC Requested OVAL Capabilities I

- **Expand capability for malware detection by adding new tests**

- **Windows:mutex test**
  - **Object: <mutex_object>**
    - **name (required) : The name of the mutex.**
  - **State: <mutex_state>**
    - **name : The name of the mutex.**
    - **owner_pid : The ID of the process which owns the mutex.**

MAEC Requested OVAL Capabilities II

- **Windows:file_signature test**
  - **Object: <file_signature_object>**
    - **filepath**
    - **path**
    - **filename**
  - **State: <file_signature_state>**
    - **filepath**
    - **path**
    - **filename**
    - **signature_exists: Whether a signature exists for the file or not.**
    - **signature_verified: Whether the authenticode signature is verified or not.**
    - **certificate_issuer: The issuer of the certificate used to sign the file.**
    - **certificate_subject: To whom the certificate used to sign the file was issued to.**

Requested Future OVAL Capabilities III

- **Add test(s) for memory scanning (heaps, etc.) ?**
  - **Use OVAL for malware forensics**

Questions?

References

- MAEC website: http://maec.mitre.org
- MAEC discussion list: http://maec.mitre.org/community/discussionlist.html
- Handshake Access: email maec@mitre.org