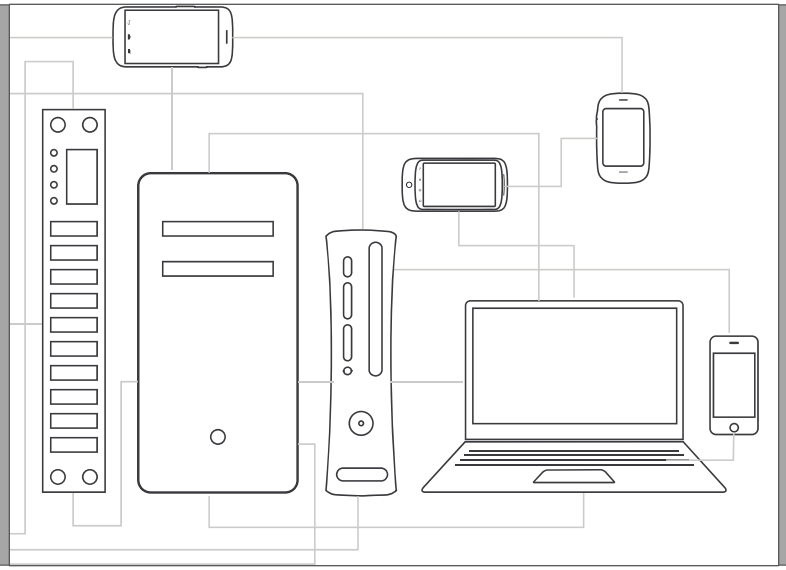
Incident Response with STIX and Autopsy



Brian Carrier
Ann Priestman

BASIS TECHNOLOGY

DHS S&T Project

- Scope: Build unique forensic analysis features into Autopsy that are open source and free for everyone.
- Focus is on law enforcement features.
- Completed:
  - Timeline Analysis (in 3.1.1)
  - Image Gallery (likely in 3.1.2)
- Working on:
  - STIX integration

What Is STIX (and CybOX)?

- STIX™: Structured Threat Intelligence Expression
- CybOX™: Cyber Observable eXpression
- Structured way of storing cyber threat intelligence to enable sharing.
  - Lots of XML.
- Sponsored by US DHS.
- Technical effort lead by MITRE.
  - https://stix.mitre.org/

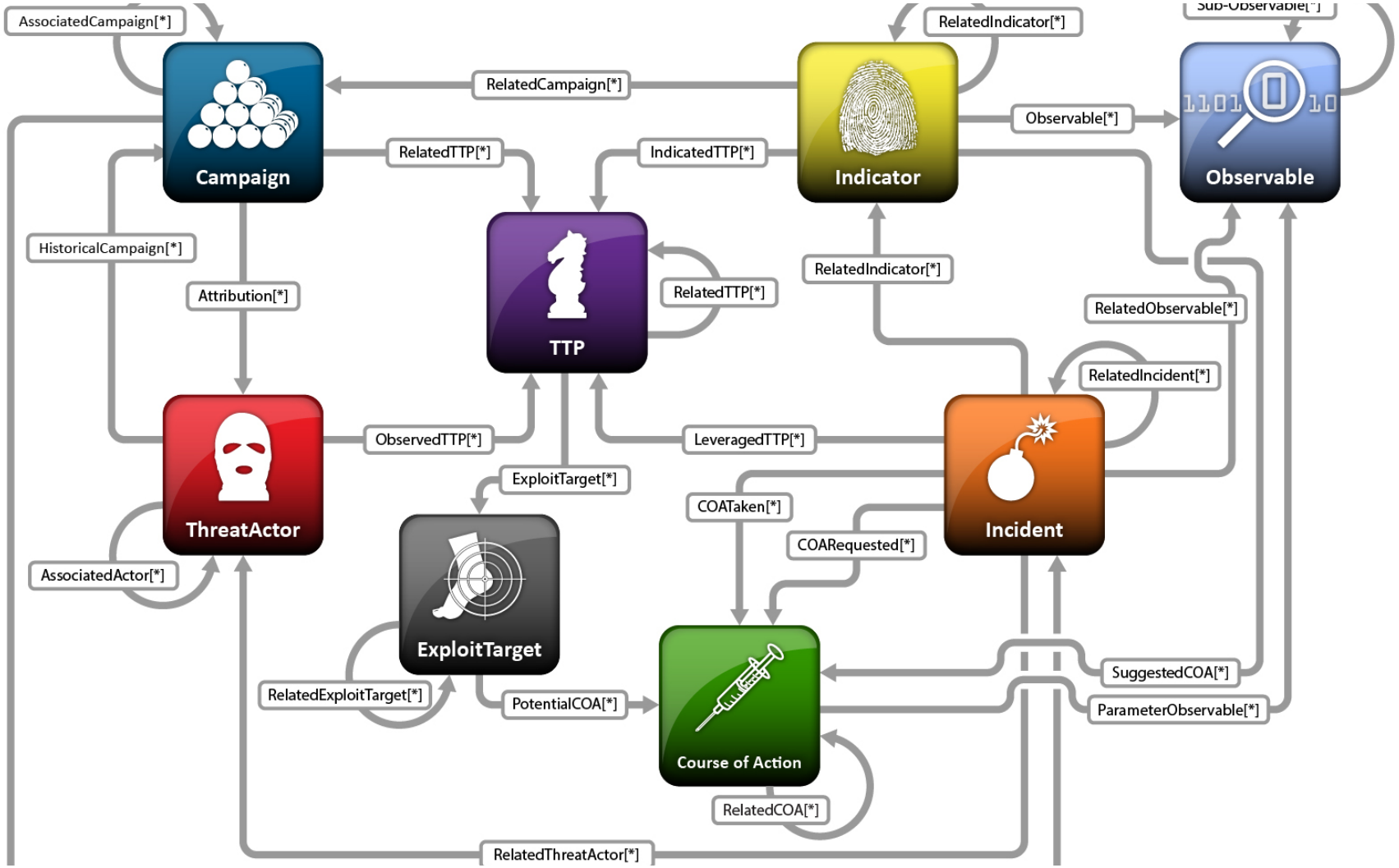
What is “Cyber (Threat) Intelligence”

- What activity are we seeing?
- What threats should I look for on my networks and systems and why?
- Where has this threat been seen?
- What does it do?
- What weaknesses does this threat exploit?
- Why does it do this?
- Who is responsible for this threat?
- What can I do about it?

From: STIX Introduction, Sean Barnum, MITRE

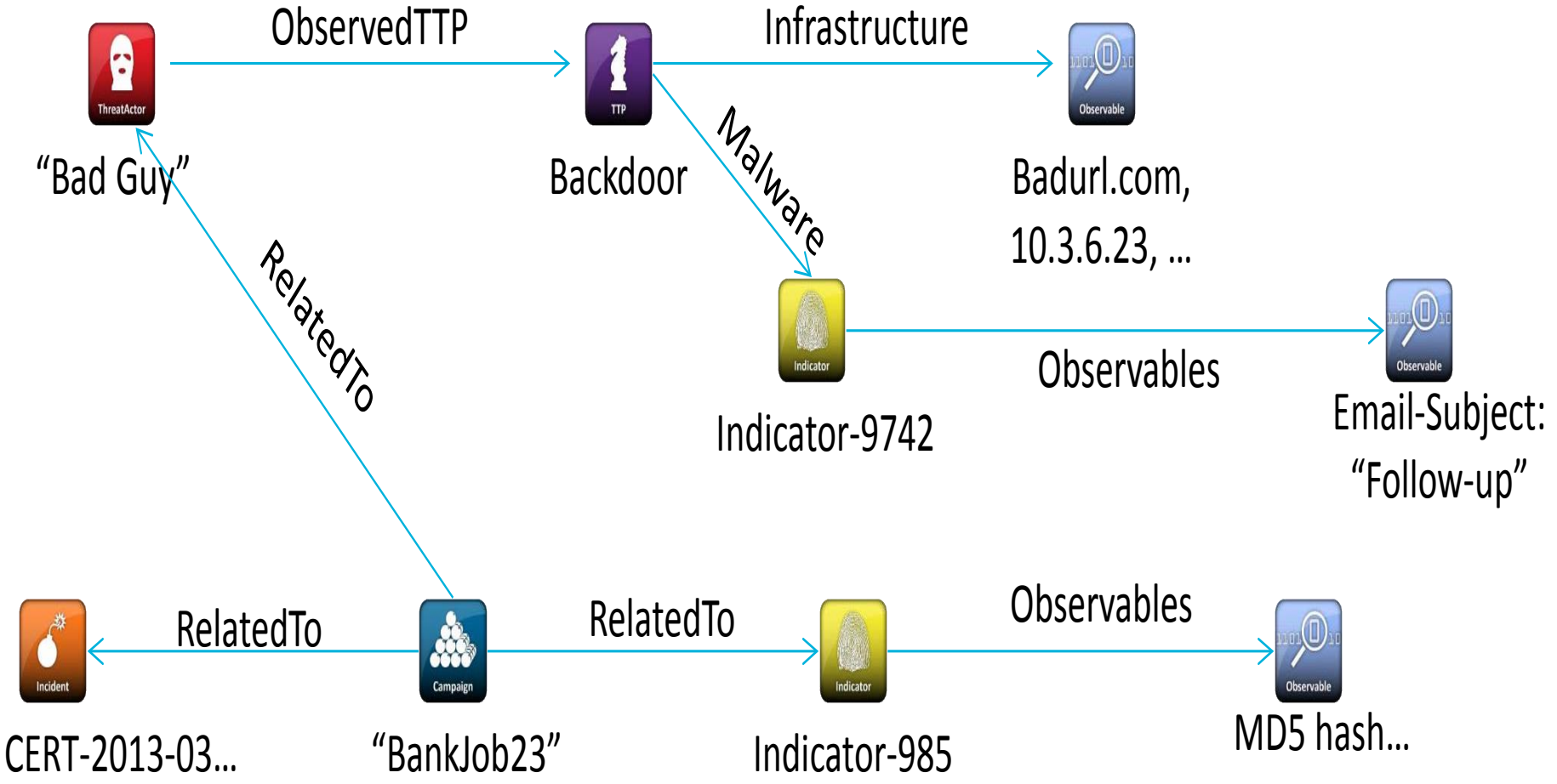
BASIS TECHNOLOGY

STIX Architecture



From: STIX Introduction, Sean Barnum, MITRE

More Concrete STIX Example



From: STIX Introduction, Sean Barnum, MITRE

CybOX Example

```
<cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  … >
  <cybox:Observable id="example:Observable-58115a77-e24a-42b5-bb29-7bd56fa9655f">
    <cybox:Description>This observable specifies a specific file
observation.</cybox:Description>
    <cybox:Object id="example:Object-17e97e7c-d3e6-4138-891b-291576dc5d41">
      <cybox:Properties xsi:type="FileObj:FileObjectType">
        <FileObj:File_Name>bad_file24.exe</FileObj:File_Name>
        <FileObj:File_Path>AppData\Mozilla</FileObj:File_Path>
        <FileObj:File_Extension>.exe</FileObj:File_Extension>
        <FileObj:Size_In_Bytes>3282</FileObj:Size_In_Bytes>
        <FileObj:Hashes>
          <cyboxCommon:Hash>
            <cyboxCommon:Type
                xsi:type="cyboxVocabs:HashNameVocab-1.0">MD5</cyboxCommon:Type>
            <cyboxCommon:Simple_Hash_Value>a7a0390e99406f8975a1895860f55f2f
                </cyboxCommon:Simple_Hash_Value>
          </cyboxCommon:Hash>
        </FileObj:Hashes>…
```

BASIS TECHNOLOGY

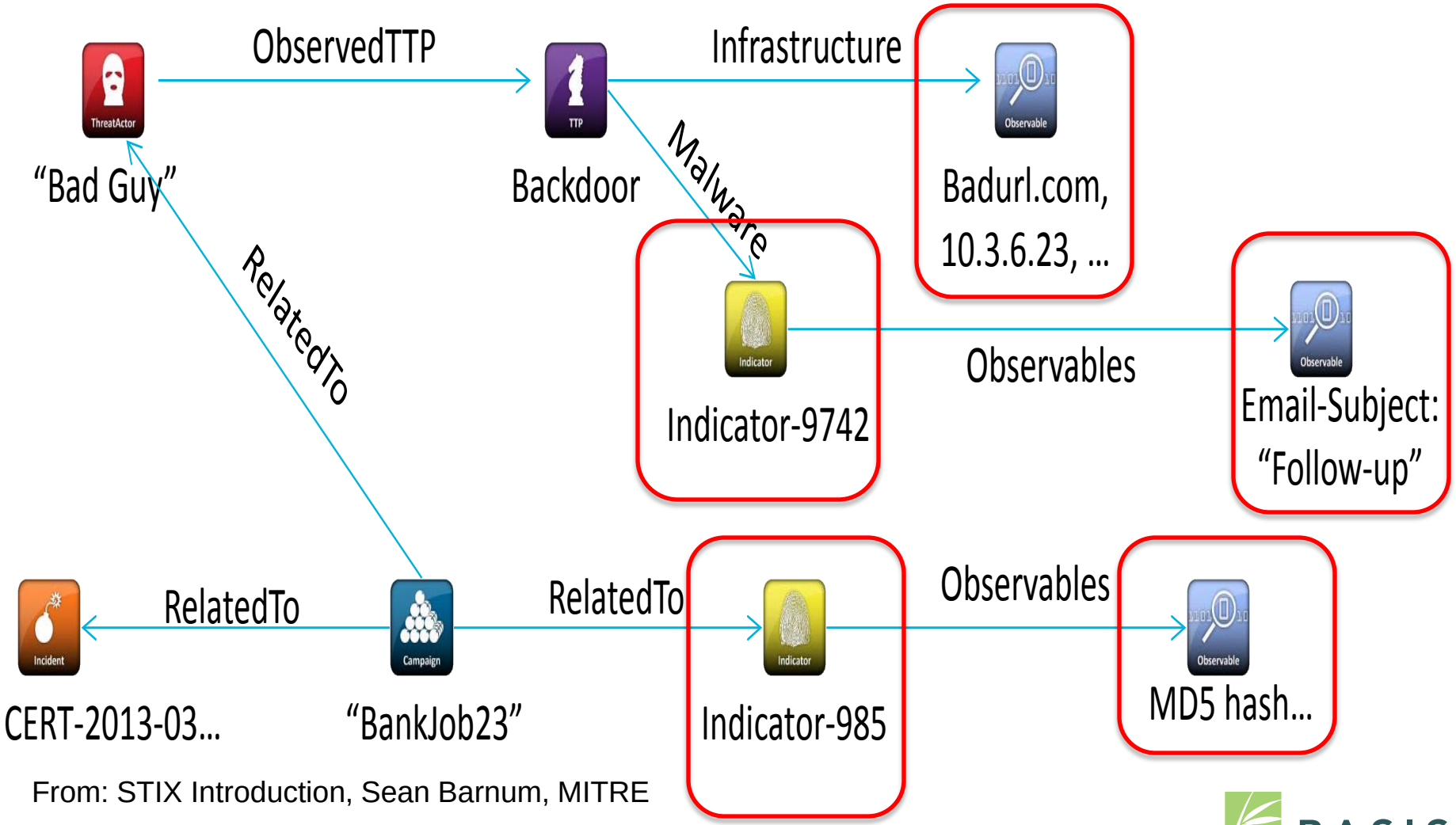
STIX Example

```
<stix:STIX_Package xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ...>
  <stix:STIX_Header><stix:Title>Example file watchlist</stix:Title></stix:STIX_Header>
  <stix:Indicators>
    <stix:Indicator xsi:type="indicator:IndicatorType" id="example:Indicator-61a1...”>
      <indicator:Description>Indicator that contains malicious file hashes.
      </indicator:Description>
      <indicator:Observable
          id="example:Observable-c9ca84dc-4542-4292-af54-3c5c914ccbbc">
        <cybox:Object id="example:Object-c670b175-bfa3-48e9-a218-aa7c55f1f884">
          <cybox:Properties xsi:type="FileObj:FileObjectType">
            <FileObj:Hashes>
              <cyboxCommon:Hash>
                <cyboxCommon:Type xsi:type="cyboxVocabs:HashNameVocab-1.0"
                    condition="Equals">MD5</cyboxCommon:Type>
                <cyboxCommon:Simple_Hash_Value condition="Equals">
                    01234567890abcdef01234567890abcde
                </cyboxCommon:Simple_Hash_Value>
              </cyboxCommon:Hash>...
```

Project Scope

- Use Case: Provide someone with a copy of Autopsy and a set of STIX documents so that they can scan a system to determine if it is compromised or not.
- We are building an Autopsy module to read STIX and search a drive.
- We are not building a module to generate STIX output.

What We Are Focusing On



From: STIX Introduction, Sean Barnum, MITRE

STIX Autopsy Module

BASIS TECHNOLOGY

Module Purpose

- Read the indicators and related observables from a STIX file and determine whether they are present on a system
  - Indicators will contain a logical combination of CybOX observables
  - Example:

    A file matching a given hash OR a given file name

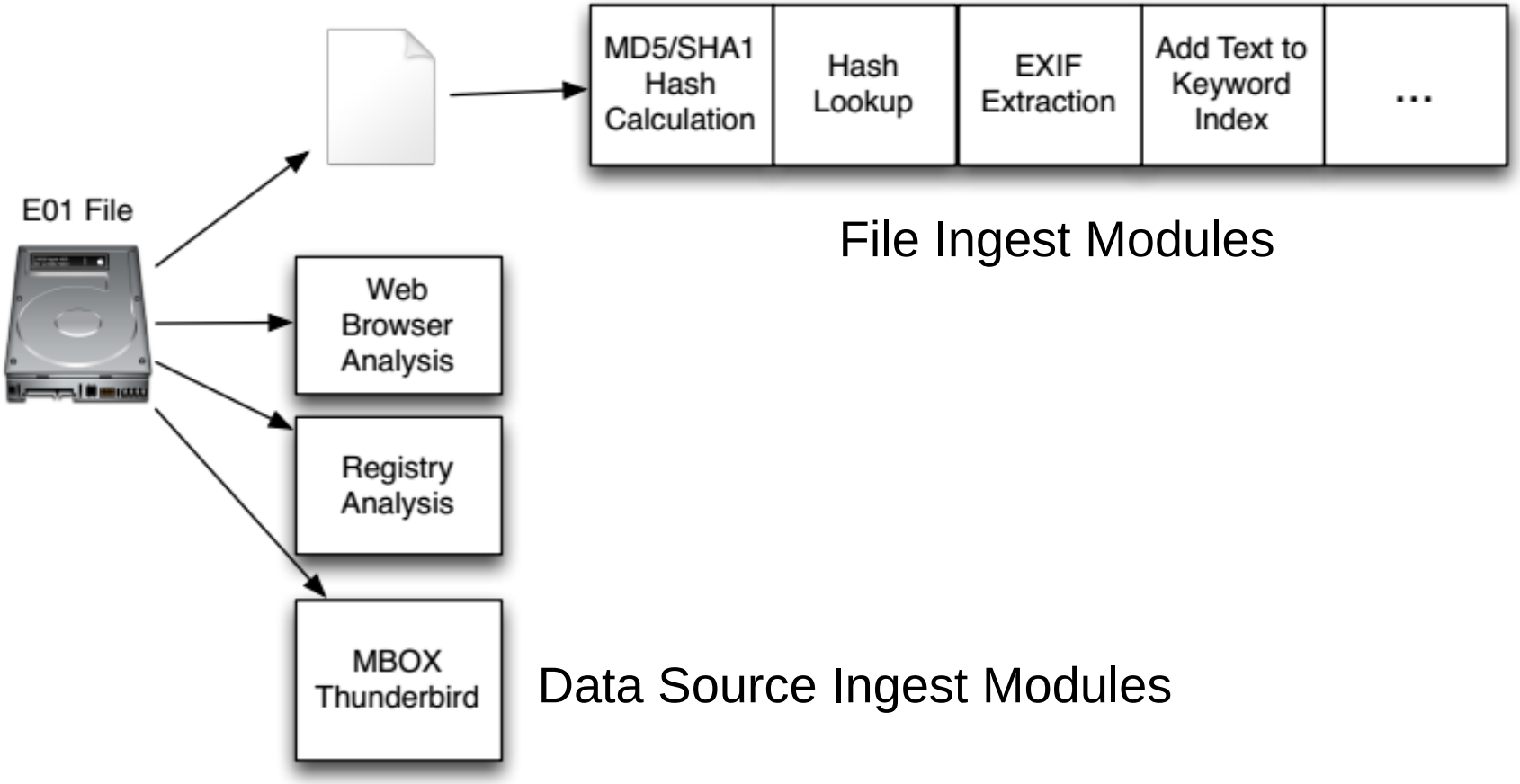
    AND

    A registry key with a given value

Planning the Autopsy Module

- What type of module to create?
- How to read the STIX data?
- Which observables to support?
- How to present the output?

Module Type Considerations



Module Type Considerations (cont.)

- The STIX module depends on the results of other Autopsy modules
- Currently, the module is implemented as a report module so it can be run after the relevant ingest modules finish
  - This is not a perfect solution, as it must be run manually and the generated report file isn't used
- In the future, we may switch to an ingest module set to run after the others finish

Parsing STIX/CybOX Data

- Use a STIX-JAXB package to create Java classes from the STIX/CybOX XML schema
  - https://github.com/PetaSecure/stix-jaxb
  - There were no official Java bindings when we started, but we may switch to those when available
- Use standard JAXB packages to read STIX files

Choosing Observables

- First working on commonly used observables and the ones that we have access to
  - For example, we started with a File Object and the following fields:

File_Name	Size_In_bytes	Modified_Time
File_Path	Hashes	Accessed_Time
File_Extension	Is_masqueraded	Created_Time
File_Format		

Choosing Observables (cont.)

- The next step is adding observables that we need to write parsers and new modules for. Some examples:
  - Can add registry parsing, which would give us Registry Key, Accounts, Network Shares, and others
  - Can add more specific file parsers for archive files, PDF files, and more

Limitations

- When picking CybOX objects/fields to support, we're limited by what Autopsy has access to
- Objects requiring live analysis are out
  - Examples: Memory, Network Packet, Pipe, Port, Process, Win Thread
- Some fields are unreasonable for Autopsy to resolve
  - Examples: Encryption Algorithm/Decryption Key in the general case, various comment fields

Creating Module Output

- Autopsy results tree
  - Stored under Interesting Items
  - Organized by indicator name
  - Includes the observable ID and associated file
- Output file
  - Currently used mostly for debugging
  - Lists observable IDs, state, and information

Release Plan

- Will be putting the current version of the STIX module on a branch of the Autopsy Git repository soon (github.com/sleuthkit/autopsy)
- Planning release around February
  - Exact release plan will be finalized once we determine what the minimum set of observables that people need are

Autopsy Module Example

Process

- Load the image into Autopsy, running relevant ingest modules
  - At present, these are Hash Lookup, File Type Identification, Keyword Search, and Extension Mismatch Detector
- Generate a report, which loads a STIX file (or files) to run against the image
- View the results in the Autopsy tree

Sample STIX Indicator

- Our sample indicator looks for all of the following observables
  - A file with MD5 = 48980ffa1f153667f6c53fcef2039c8f
  - One of the URLs http://www.boston.com/ or http://www.espn.com/
  - At least one of the following:
    - A file with name “badFile.txt”
    - IP Address 192.168.0.15

STIX Indicator (XML)

```
<stix:Indicator xsi:type="indicator:IndicatorType" timestamp="2014-05-08T09:00:00.000000Z"
      id="indicator-8d88d233-1e16-4814-814e-662fb0ac842f">
  <indicator:Title>Sample Indicator 4</indicator:Title>
  <indicator:Description>An indicator example for testing. Searches for a URL, a file with a given hash,
      and either a file with a given name or an IP Address.</indicator:Description>
  <indicator:Observable>
    <cybox:Observable_Composition operator="AND">
      <cybox:Observable idref="Observable-Pattern-1980ce43-8e03-490b-863a-ea404d12242e"/>
      <cybox:Observable idref="Observable-Pattern-275546cf-7722-a923-10cb-ef32e03171ac"/>
      <cybox:Observable id="observable-conference_OR">
        <cybox:Observable_Composition operator="OR">
          <cybox:Observable idref="Observable-Pattern-cc5c00ce-98a6-4cbe-8474-59eaecdb018f"/>
          <cybox:Observable idref="Observable-Pattern-33fe3b22-0201-47cf-85d0-97c02164528d"/>
        </cybox:Observable_Composition>
      </cybox:Observable>
    </cybox:Observable_Composition>
  </indicator:Observable>
</stix:Indicator>
```

MD5

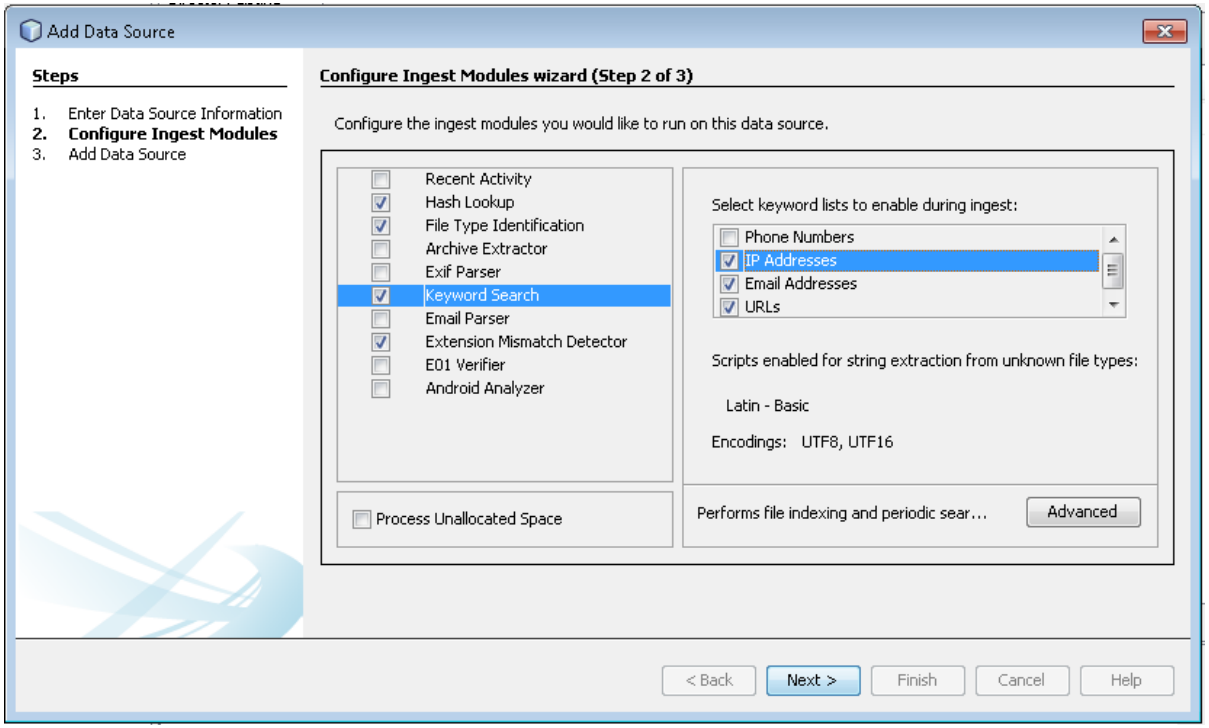
URL

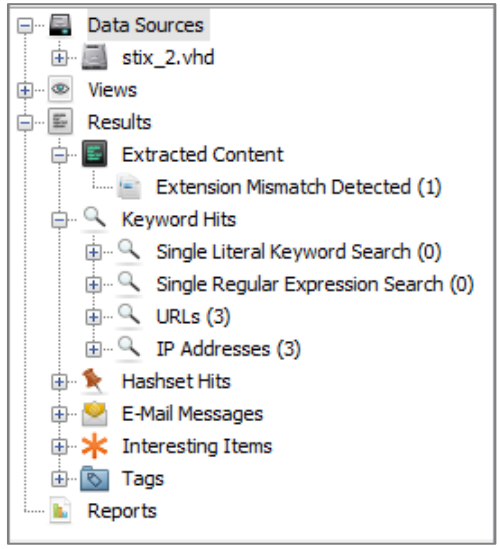
File name

IP

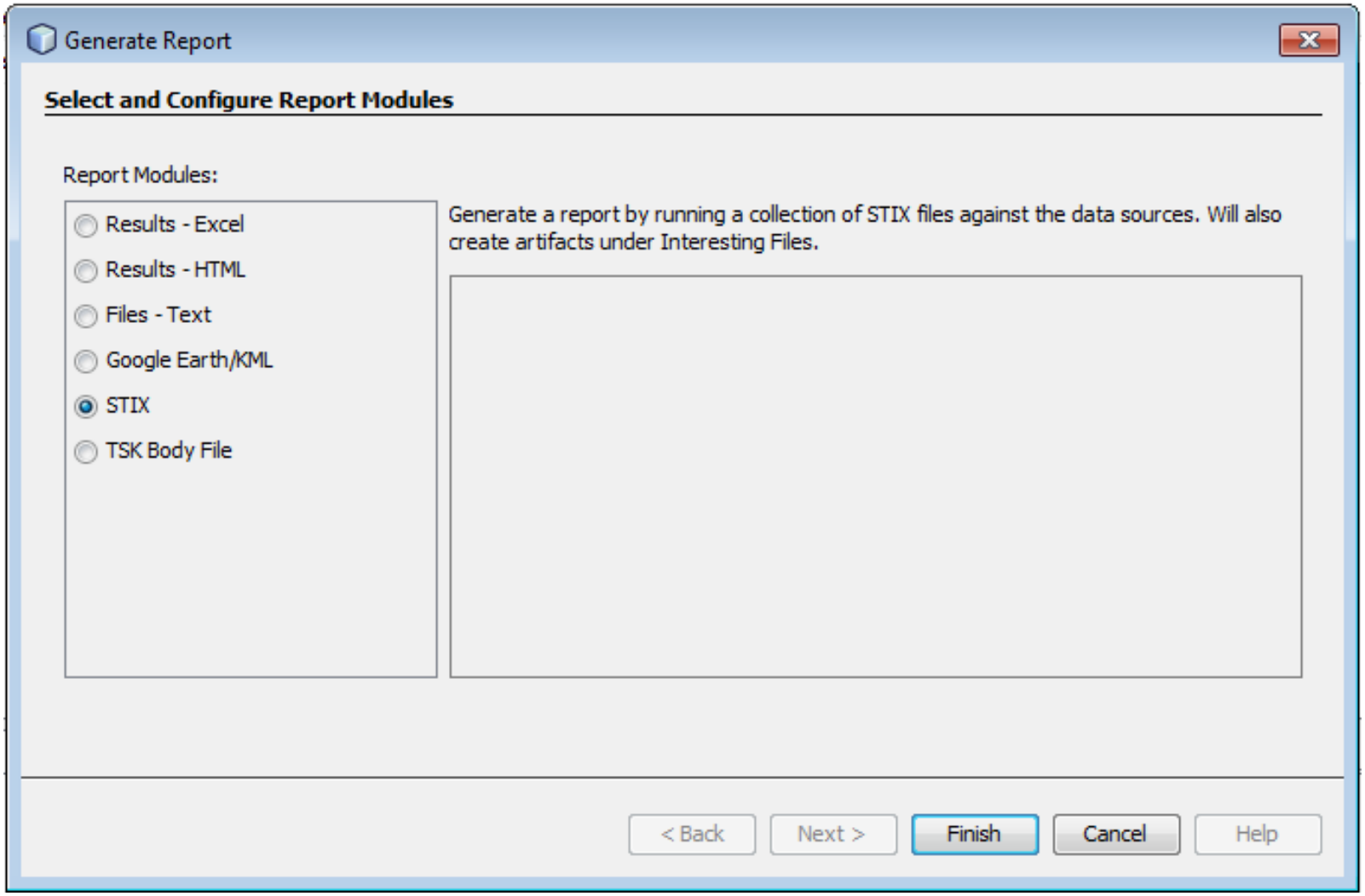
BASIS TECHNOLOGY

Run Ingest Modules

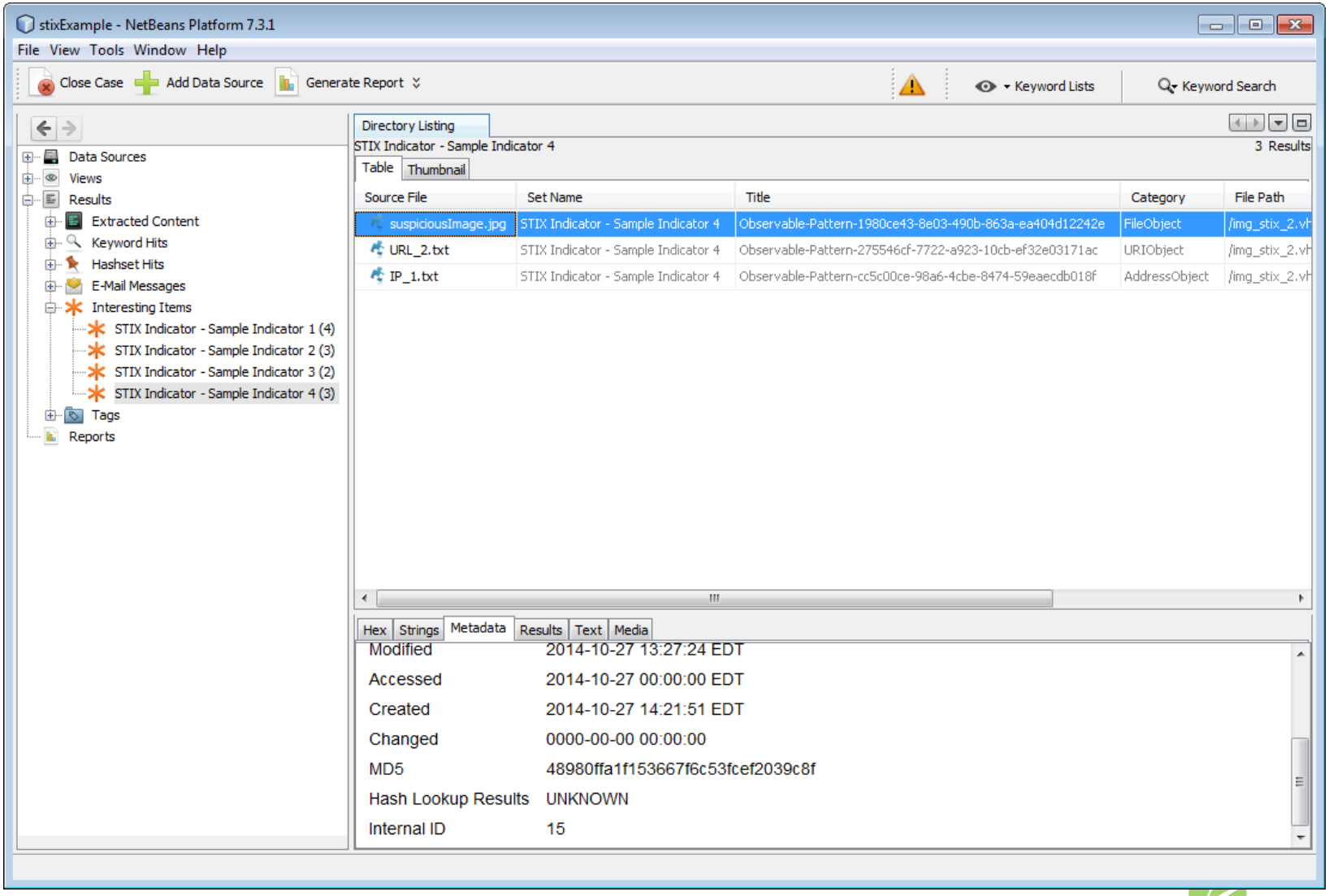




Run the STIX Report Module



Results



Questions?

Discussion (Time Permitting)

- Who is using these structured formats?
- What formats are you using:
  - STIX / Cybox
  - OpenIOC
- What do you want to scan the system for?
- How important is it to make STIX / Cybox output from Autopsy?
- How important is speed?
  - Is our big search at the end OK?
  - Need real-time searches as each file is analyzed?