Cyber Threat Intelligence

Threat Intelligence definition

- intelligent information collection and processing to…
  - support organizations develop a proactive security infrastructure for effective decision making
  - provide data to SecOps tools for **informed** detection and response

Kinds of data (1/3)

- Indicator of Compromise: data that, if found in the systems or networks, are **evidence of a security breach**
  - Network level
    - Domains
    - Emails
    - IP addresses
    - Traffic patterns
  - Host level
    - Pathname
    - Hash
    - Strings
  - Behaviour
  - Email text

Kinds of data (2/3)

- News
  - Dark web
  - Surface web
    - Security bulletins and …
    - websites about hot trends in the dark web
  - Consortia for organizations of same sector
    - Information Sharing and Analysis Centers (ISAC)
- Attack patterns
  - TTP see MITRE ATT&CK

Kinds of data (3/3)

- **Threat actors**
  - Formal or informal “organization” that performs the attacks
- **Campaigns**: a set of attacks performed by a group with specific intent and possibly homogeneous procedures
- **Software/malwares**
  - E.g. WannaCry
- **Results from malware analysis**
- **Vulnerabilities**

- **Relations among the above!**

Data sources “features”

- Public/private/paid
- Community/Companies/Government/Internal
- Complete/partial
- Accurate/inaccurate

Languages and standards

- Sigma
  - represents rules for attack detection in logs
- Yara
  - represents rules for malware identification in files and memory
- STIX
  - represents threat intelligence data and their relationships
- TAXII
  - a transport protocol for STIX, enable easy information exchange between producers and users of STIX data

Sigma rule

- A language that enable easy sharing of detection rules among security professionals and automated tools
- Sigma rues are expressed in yaml format
  - yaml is a different syntax for a json, more suitable for human editing
    - same data types: objects (i.e. maps), array, strings, number, boolean, null
  - however, by standard, Sigma rule converted in json is not standard.

- A single Sigma detection rule can be used across many SIEMs, logging platforms and databases
- Automated conversion form sigma to vendor specific languages
  - E.g. sigma-cli

Sigma example

```
title: Windows Defender Threat Detection Disabled
id: 123e4567-e89b-12d3-a456-426614174001
logsource:
    product: windows
    service: windefend
detection:

    # Detect Windows Defender being disabled or tampered with
    defender_events:
        EventID:
            - 5010
            - 5012
            - 5101
        timeframe: 1h

    # Detect administrative users making configuration changes
    admin_user_events:
        EventID: 4719
        User: Administrator
        timeframe: 1h

    # Trigger only if both conditions occur within the timeframe
    condition: defender_events and admin_user_events
```

A real Sigma example

```yaml
title: WannaCry Ransomware Activity
id: 41d40bff-377a-43e2-8e1b-2e543069e079
status: test
description: Detects WannaCry ransomware
activity
references:
    - https://www.hybrid-
analysis.com/sample/ed01ebfbc9eb5bbea545af4d01b
f5f10716618404804 39c6e5babe8e080e41aa?environme
ntId=100
    - https://x.com/nas_bench/status/1868639048
author: Florian Roth (Nextron Systems), Tom U.
@c_APT_ure (collection), oscd.community,
Jonhnathan Ribeiro
date: 2019-01-16
modified: 2025-10-18
tags:
    - attack.lateral-movement
    - attack.t1210
    - attack.discovery
    - attack.t1083
    - attack.defense-evasion
    - attack.t1222.001
    - attack.impact
    - attack.t1486
    - attack.t1490
    - detection.emerging-threats
logsource:
    category: process_creation
    product: windows
detection:
    selection_img:
        - Image|endswith:
            - '\tasksche.exe'
            - '\mssecsvc.exe'
            - '\taskdl.exe'
            - '\taskhsvc.exe'
            - '\taskse.exe'
            - '\111.exe'
            - '\lhdfrgui.exe'
            # - '\diskpart.exe'
            - '\linuxnew.exe'
            - '\wannacry.exe'
        - Image|contains: 'WanaDecryptor'
    selection_cmd:
        CommandLine|contains: '@Please_Read_Me@.txt'
    condition: 1 of selection_*
fields:
    - CommandLine
    - ParentCommandLine
falsepositives:
    - Unknown
level: critical
```

“tags”

Tag	MITRE ATT&CK	Descrizione
attack.lateral-movement	**Tattica**	Indica che il comportamento rilevato può far parte del movimento laterale, cioè la propagazione del malware all’interno della rete.
attack.t1210	**Exploitation of Remote Services**	WannaCry sfrutta SMBv1 per propagarsi sfruttando vulnerabilità come EternalBlue.
attack.discovery	**Tattica**	Attività legate alla raccolta di informazioni sul sistema o sulla rete (ad esempio, scansione interna).
attack.t1083	**File and Directory Discovery**	Il malware può esplorare i file e le directory locali per individuare target da cifrare.
attack.defense-evasion	**Tattica**	Rappresenta tentativi del malware di nascondere la propria presenza o evitare il rilevamento.
attack.t1222.001	**File and Directory Permissions Modification**	WannaCry può tentare di modificare permessi su file o directory per garantire l’accesso durante la cifratura.
attack.impact	**Tattica**	Indica che l’attività ha un impatto diretto sul sistema o sull’organizzazione (es. perdita di dati).
attack.t1486	**Data Encrypted for Impact**	È la tecnica chiave del ransomware: cifrare i dati per causare danno e chiedere riscatto.
attack.t1490	**Inhibit System Recovery**	WannaCry cancella o disabilita le copie shadow (backup) per impedire il recupero dei file.
detection.emerging-threats	**Categoria personalizzata**	Indica che la regola riguarda una minaccia emergente o recente, utile per threat hunting.

Yara

- A tool and language for creating *rules* to identify malware and suspicious files.
  - Special purpose language (no json or yaml)
- The corresponding of Sigma for malware signatures
  - Express how to detect known malicious patterns in files, memory, or processes
- **Key Features:**
  - Rules consist of strings, patterns, and conditions
  - Supports boolean logic, regex, and hex patterns
- **Integration with STIX/TAXII:**
- YARA rules can be embedded as STIX Indicators by standard
- Distributed via TAXII servers for automated threat intelligence sharing

YARA rule example (1/2)

```
import "pe"

rule WannaCry_Ransomware_Dropper
{

  meta:
    description = "WannaCry Ransomware Dropper"
    reference = "https:__HIGHLIGHT_0__"
    date = "2017-05-12"

  strings:
    $s1 = "cmd.exe /c \"%s\"" fullword ascii
    $s2 = "tasksche.exe" fullword ascii
    $s3 = "icacls . /grant Everyone:F /T /C /Q" fullword ascii
    $s4 = "Global\\MsWinZonesCacheCounterMutexA" fullword ascii

  condition:
    uint16(0) == 0x5a4d and
    filesize < 4MB and
    all of them
}
```

YARA rule example (2/2)

```
…

rule WannaCry_SMB_Exploit
{
  meta:
    description = "WannaCry SMB Exploit"
    reference = "https:__HIGHLIGHT_1__"
    date = "2017-05-12"

  strings:
    $s1 = { 53 4D 42 72 00 00 00 00 18 53 C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
FF FE 00 00 40 00 00 62 00 02 50 43 20 4E 45 54 57 4F 52 4B 20 50 52 4F 47 52 41 4D 20
31 2E 30 00 02 4C 41 4E 4D 41 4E 31 2E 30 00 02 57 69 6E 64 6F 77 73 20 66 6F 72 20 57
6F 72 6B 67 72 6F 75 70 73 20 33 2E 31 61 00 02 4C 4D 31 2E 32 58 30 30 32 00 02 4C 41
4E 4D 41 4E 32 2E 31 00 02 4E 54 20 4C 4D 20 30 2E 31 32 00 00 00 00 00 00 00 88 FF 53
4D 42 73 00 00 00 00 18 07 C0 }

  condition:
    uint16(0) == 0x5a4d and
    filesize < 4MB and
    all of them and
    pe.imports("ws2_32.dll", "connect") and
    pe.imports("ws2_32.dll", "send") and
    pe.imports("ws2_32.dll", "recv") and
    pe.imports("ws2_32.dll", "socket") and
    pe.imports("ws2_32.dll", "closesocket")
}
```

STIX

- **Structured Threat Information eXpression**.
- A standardized language designed for representing, sharing, and analyzing **cyber threat intelligence (CTI)** in a structured, machine-readable format.

- It expresses CTI concepts and relations among them

- By standard is json only
  - but yaml can be used for human readability

Classes of objects represented in STIX 2.1 i.e. STIX Domain Objects (SDO)

Name	Description
Attack Pattern	A type of TTP that describe ways that adversaries attempt to compromise targets.
Campaign	A grouping of adversarial behaviors that describes a set of malicious activities or attacks (sometimes called waves) that occur over a period of time against a specific set of targets.
Course of Action	A recommendation from a producer of intelligence to a consumer on the actions that they might take in response to that intelligence.
Grouping	Explicitly asserts that the referenced STIX Objects have a shared context, unlike a STIX Bundle (which explicitly conveys no context).
Identity	Actual individuals, organizations, or groups (e.g., ACME, Inc.) as well as classes of individuals, organizations, systems or groups (e.g., the finance sector).
Indicator	Contains a pattern that can be used to detect suspicious or malicious cyber activity.
Infrastructure	Represents a type of TTP and describes any systems, software services and any associated physical or virtual resources intended to support some purpose (e.g., C2 servers used as part of an attack, device or server that are part of defence, database servers targeted by an attack, etc.).

Classes of objects represented in STIX 2.1 i.e. STIX Domain Objects (SDO)

Name	Description
Intrusion Set	A grouped set of adversarial behaviors and resources with common properties that is believed to be orchestrated by a single organization.
Location	Represents a geographic location.
Malware	A type of TTP that represents malicious code.
Malware Analysis	The metadata and results of a particular static or dynamic analysis performed on a malware instance or family.
Note	Conveys informative text to provide further context and/or to provide additional analysis not contained in the STIX Objects, Marking Definition objects, or Language Content objects which the Note relates to.
Observed Data	Conveys information about cyber security related entities such as files, systems, and networks using the STIX Cyber-observable Objects (SCOs).
Opinion	An assessment of the correctness of the information in a STIX Object produced by a different entity.

Classes of objects represented in STIX 2.1 i.e. STIX Domain Objects (SDO)

Name	Description
Report	Collections of threat intelligence focused on one or more topics, such as a description of a threat actor, malware, or attack technique, including context and related details.
Threat Actor	Actual individuals, groups, or organizations believed to be operating with malicious intent.
Tool	Legitimate software that can be used by threat actors to perform attacks.
Vulnerability	A mistake in software that can be directly used by a hacker to gain access to a system or network.

Relations that can be represented in STIX

Name	Description
Relationship	Used to link together two SDOs or SCOs in order to describe how they are related to each other.
Sighting	Denotes the belief that something in CTI (e.g., an indicator, malware, tool, threat actor, etc.) was seen.

STIX: Cyber Observable eXpressions (CybOX)

Type	Example Pattern
IP	[ipv4-addr:value = '192.168.1.5']
Domain	[domain-name:value = 'malicious[.]com']
File hash	[file:hashes.'SHA-256' = 'abc123...']
URL	[url:value = 'hxxp://bad.site/download.exe']
Registry	[windows-registry-key:key = 'HKCU\\Software\\BadKey']

- Can be combined:
- [file:hashes.MD5 = 'abcd1234'] OR [domain-name:value = 'malicious.com']

Example Objects


```
{
   "type": "indicator",
   "spec_version": "2.1",
   "id": "indicator--71312c48-925d-44b7-b10e-c11086995358",
   "created": "2017-02-06T09:13:07.243000Z",
   "modified": "2017-02-06T09:13:07.243000Z",
   "name": "CryptoLocker Hash",
   "description": "This file is a part of CryptoLocker",
   "pattern": "[file:hashes.'SHA-256' = '46afeb295883a5efd6639d4197eb18bcba3bff49125b810ca4b9509b9ce
   "pattern_type": "stix",
   "indicator_types": ["malicious-activity"],
   "valid_from": "2017-01-01T09:00:00.000000Z"
}

{
    "type": "malware",
    "id": "malware--81be4588-96a8-4de2-9938-9e16130ce7e6",
    "spec_version": "2.1",
    "created": "2017-02-06T09:26:21.647000Z",
    "modified": "2017-02-06T09:26:21.647000Z",
    "name": "CryptoLocker",
    "description": "CryptoLocker is known to hold files hostage for ransom.",
    "malware_types": ["ransomware"]
}
```

Example Relationship


```
{
   "type": "relationship",
   "id": "relationship--a19fac85-f6f5-47f3-aacd-4bfb54557852",
   "spec_version": "2.1",
   "created": "2017-02-06T09:30:51.987000Z",
   "modified": "2017-02-06T09:30:51.987000Z",
   "relationship_type": "indicates",
   "source_ref": "indicator--71312c48-925d-44b7-b10e-c11086995358",
   "target_ref": "malware--81be4588-96a8-4de2-9938-9e16130ce7e6"
}
```

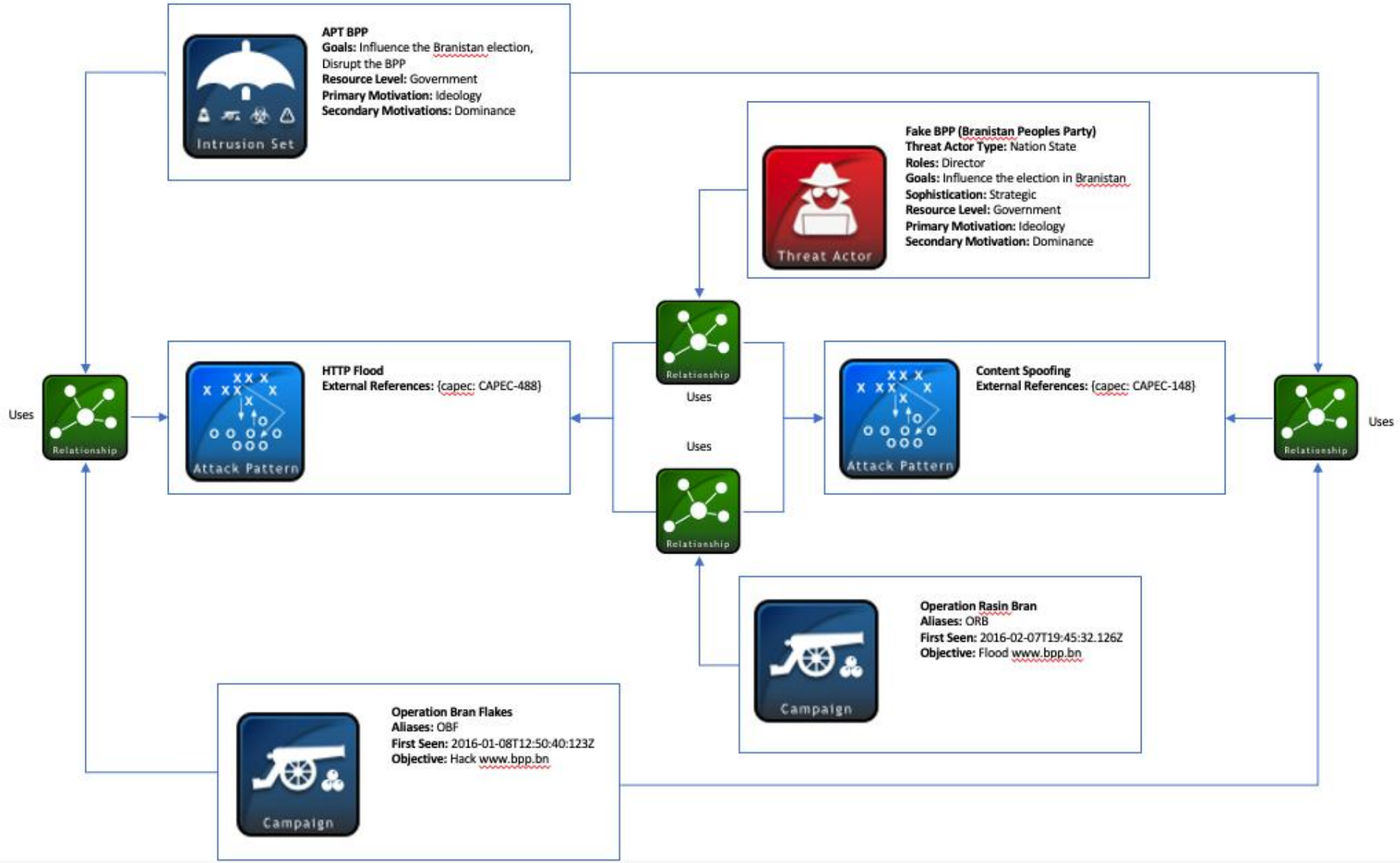
Relationship Types

Relationship Type	Example	Meaning
uses	threat-actor uses malware	Actor or campaign uses a tool, malware, etc.
targets	malware targets vulnerability	Source is directed against target.
compromises	malware compromises infrastructure	Compromises a system or network.
attributed-to	campaign attributed-to threat-actor	Attribution relationship.
mitigates	course-of-action mitigates vulnerability	Describes mitigation or defense.
indicates	indicator indicates malware	Indicator points to another object (e.g., malicious behavior).
delivers	malware delivers payload	Source delivers the target (e.g., downloader malware).
drops	malware drops file	Source drops another malware or file.
exploits	malware exploits vulnerability	Source exploits a weakness.
originates-from	attack-pattern originates-from region	Geographic or logical origin.
communicates-with	malware communicates-with infrastructure	Source connects to target.

Relationship Types

Relationship Type	Example	Meaning
controls	threat-actor controls infrastructure	Control or command-and-control.
hosts	infrastructure hosts malware	Target hosted by source.
authored-by	report authored-by organization	Creation or authorship.
related-to	(any → any)	Generic relationship if no specific one fits.
duplicate-of	(any → any)	Denotes object duplication.
derived-from	(any → any)	Derived or adapted content.
revoked-by	(any → any)	Object superseded by another.

STIX: A graphical example

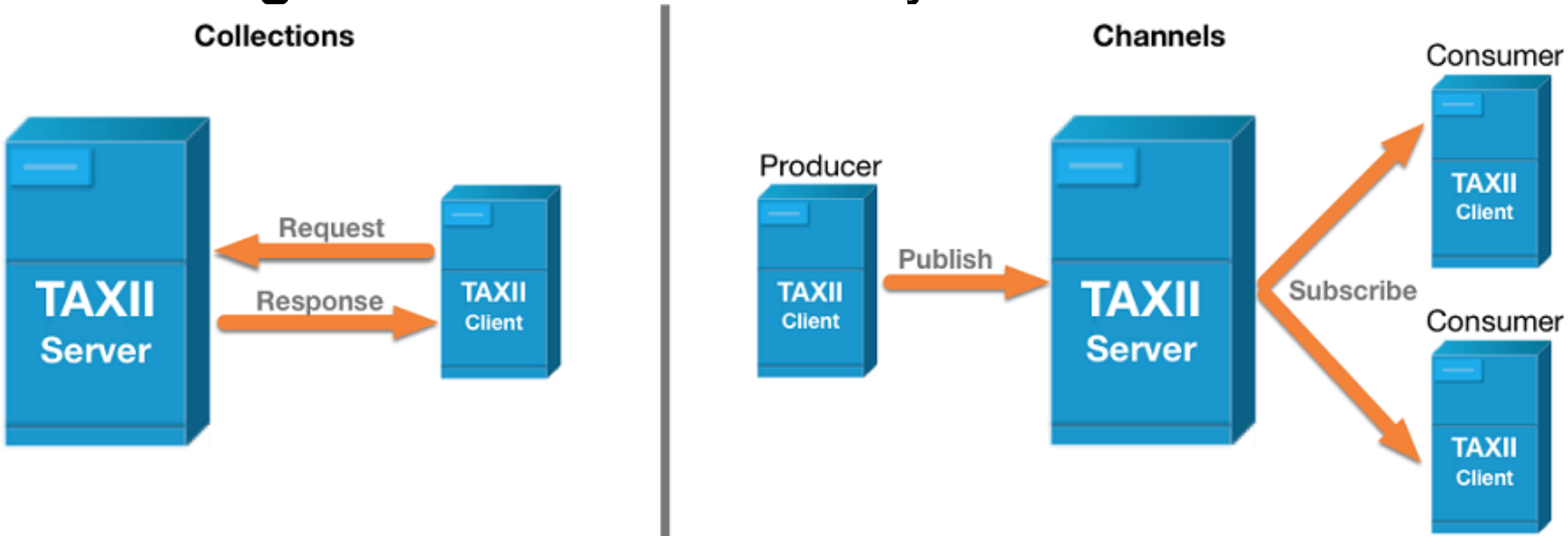


Examples and tools for STIX

- https://oasis-open.github.io/cti-documentation
- https://oasis-open.github.io/cti-stix-visualization/

TAXII

- **TAXII** stands for **Trusted Automated eXchange of Intelligence Information**.
- It's a **transport protocol** — a standardized way to **exchange STIX data** between systems over HTTPS.



TAXII

- **RESTful API**: Uses HTTPS endpoints for retrieving and pushing STIX data.
- **Collections**: Logical “buckets” of STIX data
  - e.g., malware, campaigns
- **Discovery**: Clients can discover available data sources and collections.
- **Security**: Supports authentication, encryption, and access control.
- **Filtering**: Efficiently query specific indicators or timeframes.
- **Versions**
  - **TAXII 1.x**: Older, XML-based, SOAP-like.
  - **TAXII 2.x / 2.1**: Modern JSON + RESTful, aligns with STIX 2.x.
- **Integration**
  - Works with **STIX bundles** for automated threat intelligence sharing.
  - Used in platforms like **OpenCTI, MISP, MITRE ATT&CK TAXII Server**, and commercial CTI systems.
  - Can be used for **real-time SOC threat intelligence feeds**.

Tools: Threat Intelligence Platforms (TIP)

- Platforms for **collecting, analyzing, and sharing threat intelligence**.
  - Integrate multiple sources (open-source, commercial feeds, internal detections) into a **unified view**.
  - STIX/TAXII feeds, APIs, and threat reports.
- Analysis & Enrichment
  - Adds context like geolocation, MITRE ATT&CK techniques, malware families, or reputation scores.
- Integration
  - Feeds SIEMs, firewalls, EDR, and SOC tools with actionable IOCs.
- Sharing
  - Publishes intelligence internally or externally (via TAXII, STIX, or email).
- Examples
  - Open-source: OpenCTI, MISP, CIF.
  - Sveral Commercial tools

MITRE ATT&CK

- ATT&CK is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations.
- Source: MITRE
  - The same of CVE database, and STIX and TAXII

- Objectives
  - Collect and organize *adversary behaviors*
  - Common *taxonomy/language*
  - Applicable to *real environments*
  - At the *right level of abstraction*

ATT&CK concepts (TTP)

- **Tactics**: a **goal** of an attacker during an attack
  - E.g. Exfiltrating sensitive data, gaining persistent access, or disrupting services.
- **Techniques**: the **method** to achieve the goal
  - E.g. Exploiting software vulnerabilities, Phishing emails to gain credentials
  - They have an ID: T1110 (Brute Force)
  - **Subtechniques**: more detailed description at lower abstraction level
    - E.g. T1110.001 (Password Guessing)
- **Procedures**: a specific implementation, it may be unique to a specific *threat actor*
  - E.g. Sending a spear-phishing email pretending to be HR with a malicious PDF attachment.

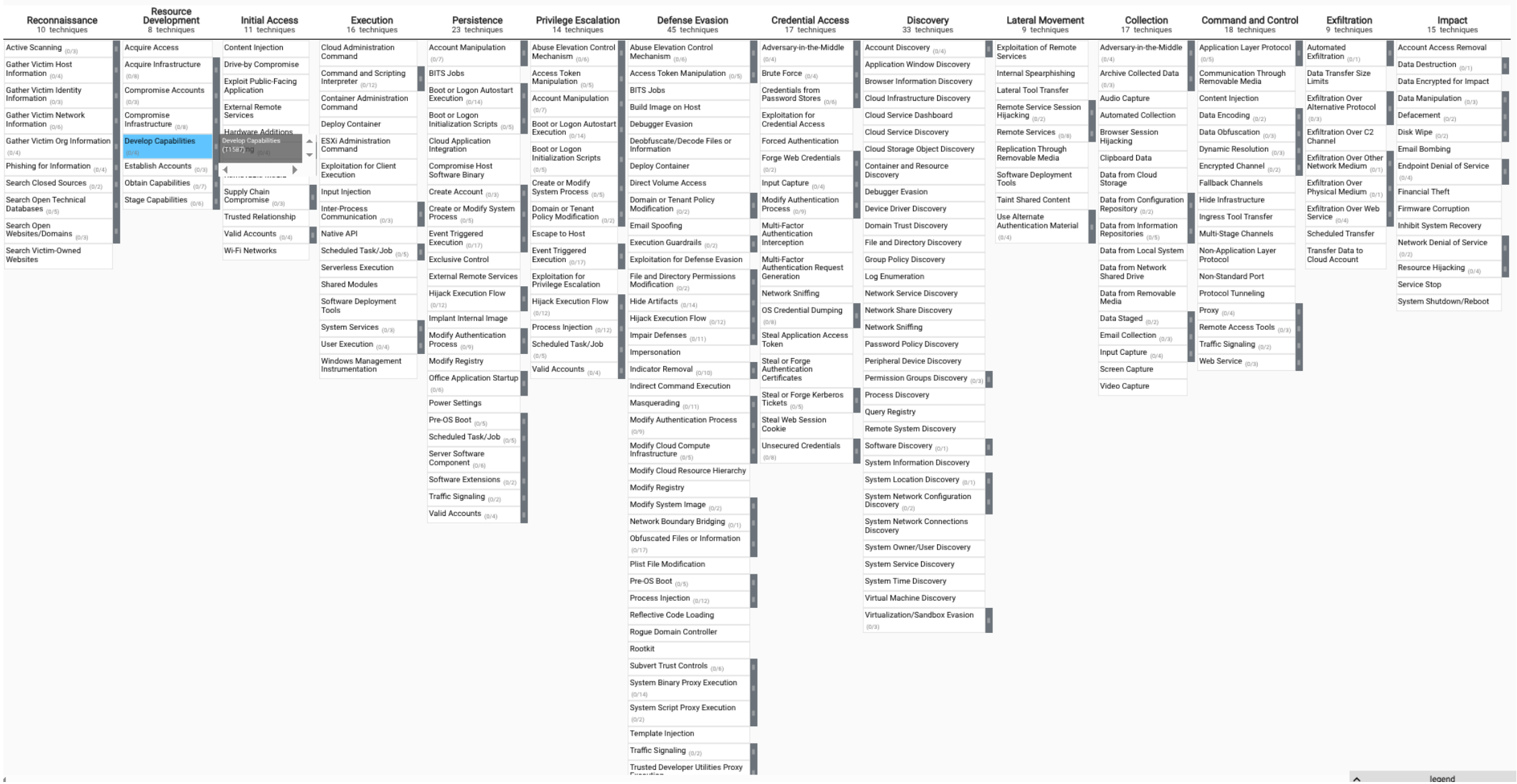
Other concepts

- **Groups**: i.e. *threat actors*
  - Usually identify the “organization” that perform the attacks
    - E.g. a *criminal organization*, an *intelligence service, APT28 (G0007)*
  - Used to group *campaigns* and used software/malware
- **Campaign**: a set of attacks performed by a group with specific intent and possibly homogeneous procedures
  - E.g. Indian Critical Infrastructure Intrusions (C0043)
- **Software/malware**
  - E.g. WannaCry (S0366)

ATT&CK matrix

Tactics →

Techniques ↓



ATT&CK matrices

Currently there are three matrices

- Enterprise
  - Covering Windows, macOS, Linux, PRE, Office Suite, Identity Provider, SaaS, IaaS, Network Devices, Containers, ESXi
- Mobile
  - Covering Android, iOS.
- Industrial Control Systems (ICS)

Tactics & Co.

- 14 tactics, each of them groups 245 techniques in ATT&CK v17 Enterprise

Tactic ID	Tactic Name	Number of Techniques
TA0043	Reconnaissance	10
TA0042	Resource Development	8
TA0001	Initial Access	11
TA0002	Execution	16
TA0003	Persistence	23
TA0004	Privilege Escalation	14
TA0005	Defense Evasion	45
TA0006	Credential Access	17
TA0007	Discovery	33
TA0008	Lateral Movement	9
TA0009	Collection	17
TA0011	Command and Control	18
TA0010	Exfiltration	9
TA0040	Impact	15

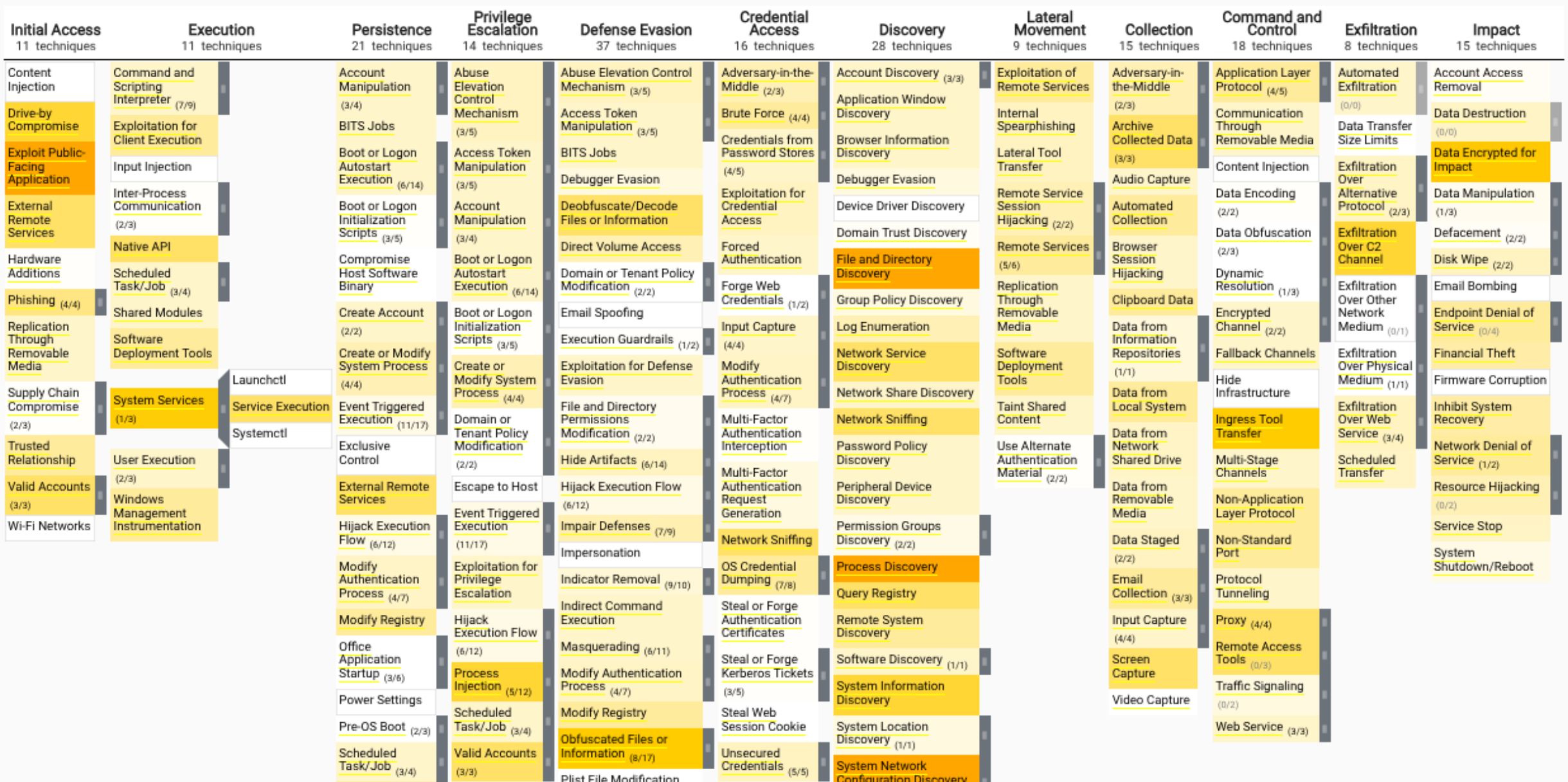
Purpose

- It is not properly a framework but a tool that helps in several tasks:
  - Detections and Analytics
  - Threat Intelligence
  - Adversary Emulation and Red Teaming
  - Assessment and Engineering

ATT&CK matrix navigator

- on-line tool to explore the matrices and to annotate it
- https://mitre-attack.github.io/attack-navigator/
  - Filter
  - Expand subtechniques
  - Annotate
  - Visually evidence information associated with (sub) techniques
    - Detected techniques
    - What is covered by your countermeasures
    - Who is in charge of what
    - etc.

matrix navigator layer example



CAPEC

- Common Attack Pattern Enumeration and Classification
- MITRE
- Publicly available catalog **of common attack patterns** that helps users understand how adversaries exploit weaknesses in apps & co.

- Attack patterns are like *design patterns* for attacks

Some well known examples

- Session Fixation (CAPEC-61)
- Cross Site Request Forgery (CAPEC-62)
- SQL Injection (CAPEC-66)
- Cross-Site Scripting (CAPEC-63)
- Buffer Overflow (CAPEC-100)
- Clickjacking (CAPEC-103)
- Relative Path Traversal (CAPEC-139)

CAPEC vs ATT&CK

- CAPEC is focused on application security
  - describes the common attributes and techniques employed by adversaries to exploit known weaknesses in cyber-enabled capabilities.
    - e.g., SQL Injection, XSS, Session Fixation, Clickjacking
  - Associated with Common Weakness Enumeration (CWE)
- ATT&CK is focused on network defense
  - describes the operational phases in an adversary's lifecycle, pre and post-exploit
    - e.g., Persistence, Lateral Movement, Exfiltration,
    - Oriented to understand advanced persistent threats (APT)

Common Weakness Enumeration (CWE)

- A list of SW (& HW) typical weakness that can became vulnerabilities
  - CVEs are actual vulnerabilities on real software
- MITRE

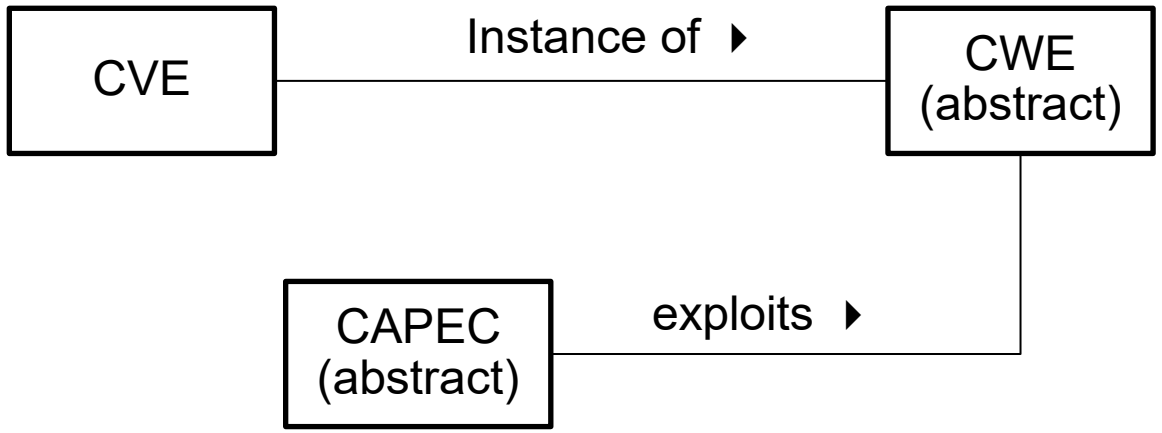
CWE examples

Weaknesses in the 2024 CWE Top 25 Most Dangerous Software Weaknesses

- Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') - *(79)*
- Out-of-bounds Write - *(787)*
- Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') - *(89)*
- Cross-Site Request Forgery (CSRF) - *(352)*
- Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') - *(22)*
- Out-of-bounds Read - *(125)*
- Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
- Use After Free - *(416)*
- Missing Authorization - *(862)*
- Unrestricted Upload of File with Dangerous Type - *(434)*
- Improper Control of Generation of Code ('Code Injection') - *(94)*
- Improper Input Validation - *(20)*
- Improper Neutralization of Special Elements used in a Command ('Command Injection') - *(77)*
- Improper Authentication - *(287)*
- Improper Privilege Management - *(269)*
- Deserialization of Untrusted Data - *(502)*
- Exposure of Sensitive Information to an Unauthorized Actor - *(200)*
- Incorrect Authorization - *(863)*
- Server-Side Request Forgery (SSRF) - *(918)*
- Improper Restriction of Operations within the Bounds of a Memory Buffer - *(119)*

Understanding MITRE standards

CWE	**Weaknesses in the code**
CAPEC	**Attack patterns that leverage a CWE**
CVE	**Documented vulnerabilities of real software, they are instances of a CWEs**
ATT&CK	**High level behavior of an attacker that can leverage all the above**



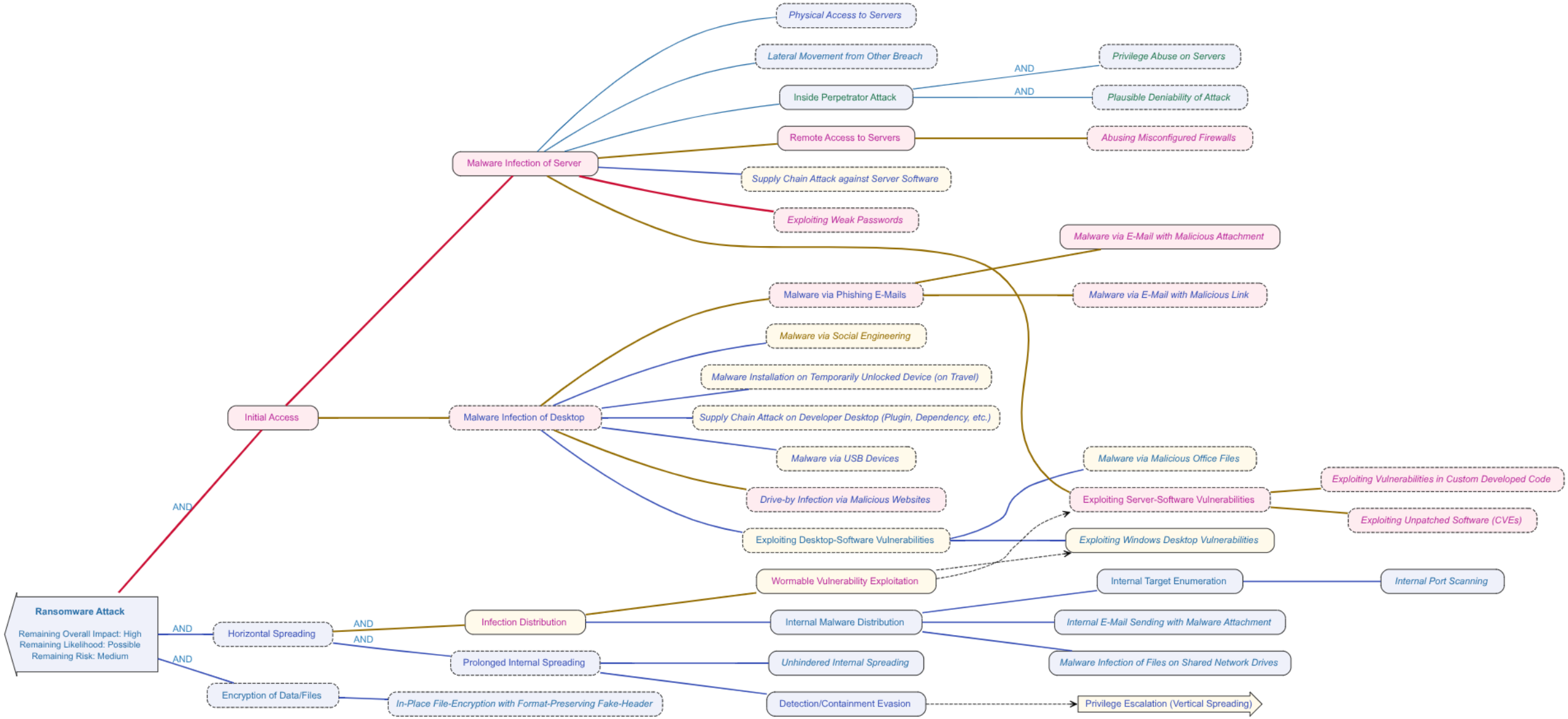
CVE vs. CWE vs. CAPEC vs. TTP

- **CVE (Specific Vulnerability)**
  CVE-2025-12345 – SQL Injection vulnerability in login.php of a web application.
  - Represents a **specific, publicly known vulnerability** in a software system
- **CWE (Underlying Weakness)**
  CWE-89: Improper Neutralization of Special Elements in SQL Commands (“SQL Injection”)
  - Explains the **type of weakness** that caused the CVE.
  - Connects multiple CVEs that share the same root cause.
- **CAPEC (Attack Pattern)**
  CAPEC-66: SQL Injection
  - Describes the **attacker technique** to exploit the weakness (CWE-89) to gain unauthorized access.
  - Provides context for other vulnerabilities with the same attack pattern.
- **TTP:**
  T1190 – Exploit Public-Facing Application
  - Represents **how an adversary leverages the CAPEC attack pattern** in practice.
  - Includes real-world procedural steps, attacker tools, and indicators.

Attack trees

- A conceptual/visual tool to break complex attacks into manageable subgoals
- A modelling tool for cybersecurity analysts
  - They are not cyber threat intelligence
  - They are linked to the specific context or organization, so nothing to share
- structure
  - Root node = the attacker goal
    - e.g., "Exfiltrate customer data"
  - Children of a node = the ways to achieve the node goal
  - Children are combined with OR / AND / SAND operators
    - SAND= sequence-constrained AND
- No standard
- Notable variation: Attack-Defense trees
  - Special nodes to represent countermeasures to block attacks

Attack tree example



Attack trees vs. CTI and risks analysis

- CTI: nodes of attack trees can be labelled with CTI information
  - mostly TTP, CWS, CVE, CAPEC
- STIX can represent attack tree nodes
  - No standard way to represent OR/AND/SAND
- Risks: nodes of attack trees can be labelled with…
    - Impact
    - Frequency
    - Cost
    - Mitigation strategy
    - Controls and their implementation advancement