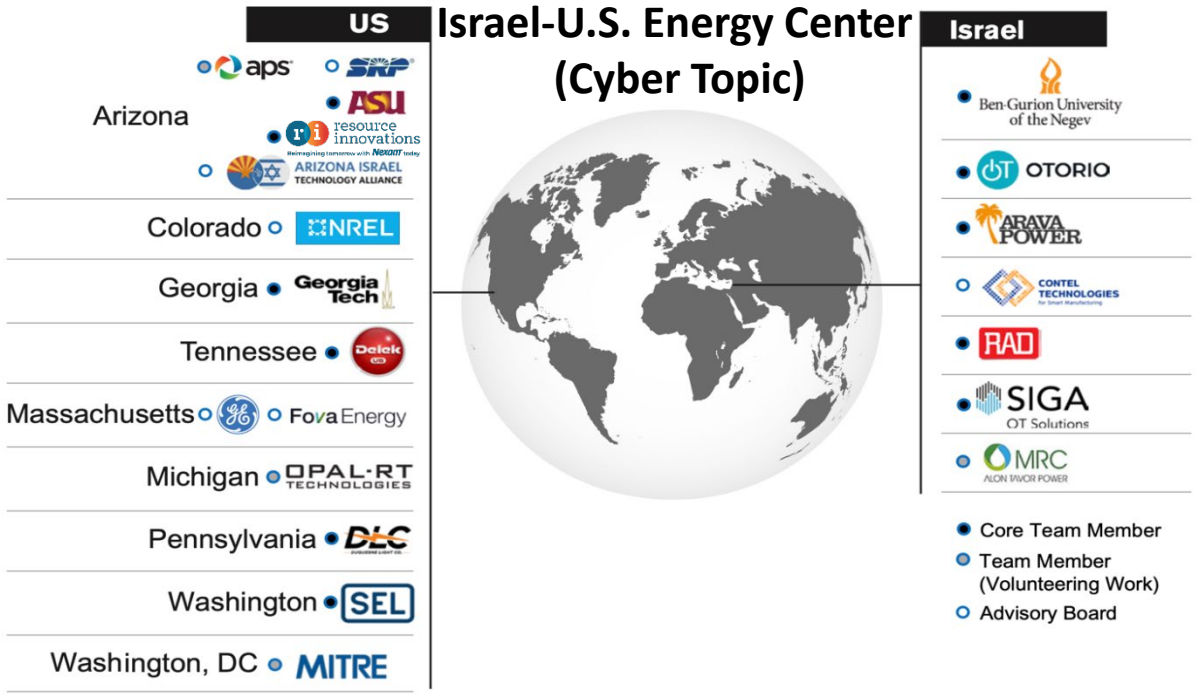
Task 4
Multi-Level Threat Intelligence Knowledge Base



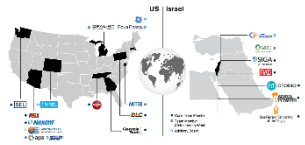
Third Project Review Workshop

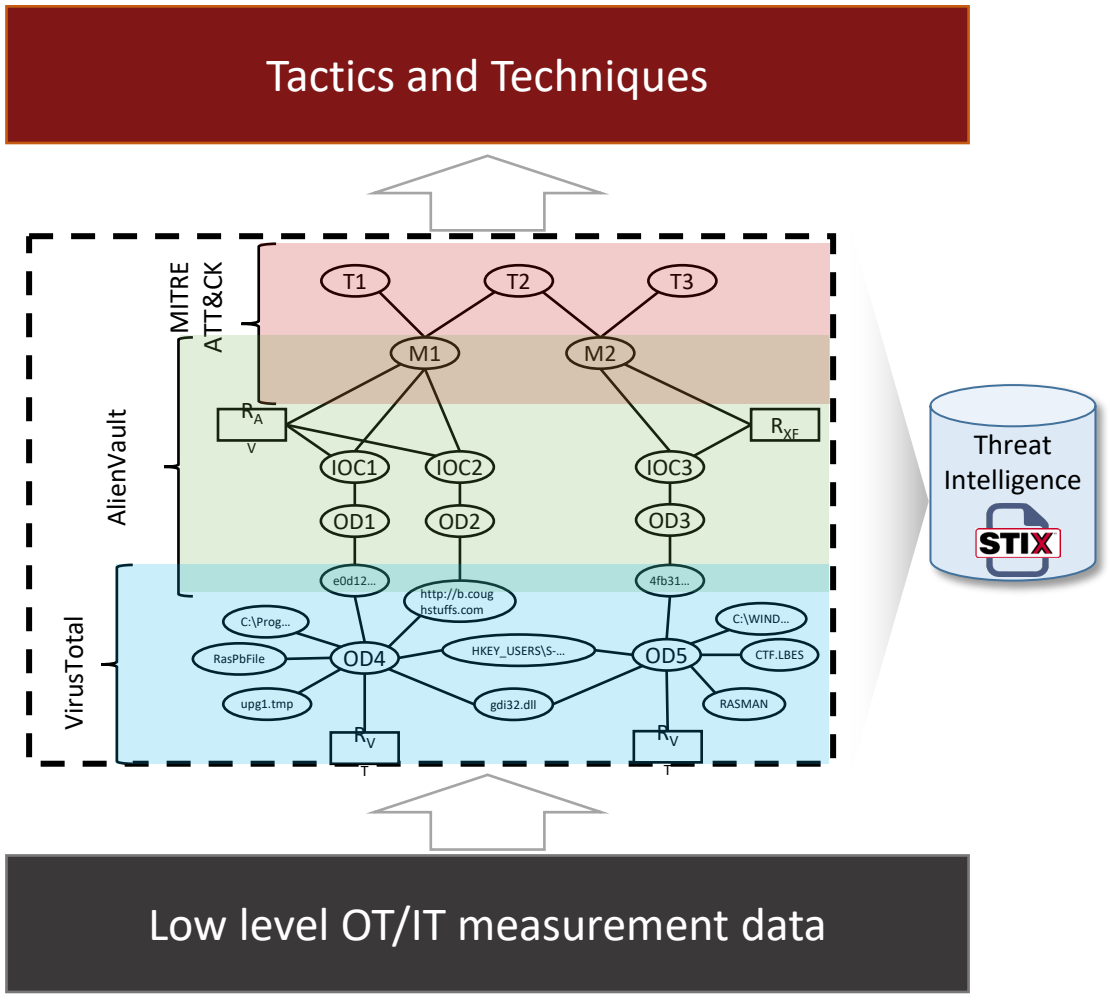
Moti Cohen

BGU

Aug 24, 2022

Multi-Level Threat Intelligence Knowledge Base



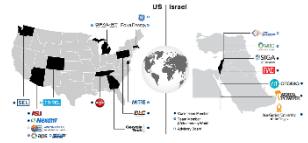


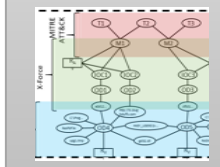
Build machine readable multi-level ICS threat ontology by fusing data from multiple cyber **threat intelligence** sources.

Challenges:

- Few Threat Intelligence sources compared to Enterprise
- Diverse types of observables (vendors/protocols/environments)

Strategy to building the knowledge base





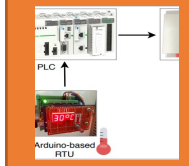
ICS Attack Ontology

Threat Intelligence

Threat Intelligence

CTI feeds and databases

Extracting and fusing publicly reported observations

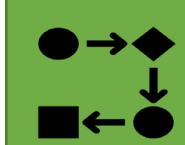


ICS Cyber Labs

Task 3
CALDERA

Generating missing observations

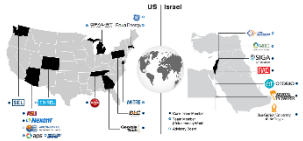
Experts



Task 2
COPE

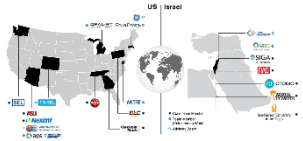
How may an attack technique be reflected in OT/IT measurements?

Data collection status



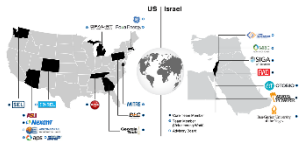
- Information collected and structured for both Enterprise and ICS realms
- Information sources are:
  - MITRE ATT&CK
  - AlienVault OTX
  - VirusTotal
- The information is gathered and stored in a graph database (Neo4j)
- We have updated our VirusTotal collector to work with the latest API (v3) and collect more information
  - Additional file metadata
  - Additional behavior data

Challenges presented in the last workshop



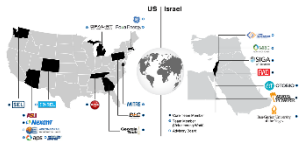
- No publicly available CTI for ICS (OT level)
  - No information to learn from
- STIX2 does not support reporting OT-specific information
  - There are some extensions for IT-specific information
  - Limits our ability to exchange CTI information between relevant parties (research groups, defense teams, etc.)

STIX2 Cyber Observables for ICS



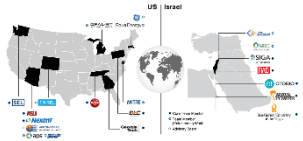
- We have started a work group together with MITRE, Otorio
- Our work was mostly focused on analyzing publicly available information regarding ICS attacks
- Another source of information was publicly available IDS detection rules
- We have extracted samples of relevant information we think would be valuable in a CTI feed
- The result of our work is different extensions of the STIX2 standard that can represent the samples we extracted

ICS reports analysis



- Analyzed 26 ICS malware reports, 1-2 per malware
- We were looking for observable indications
  - Network traffic
  - Physical behavior
  - Device configuration changes
- 10 reports contained some ICS level observables, some specific some more general

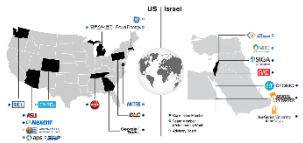
General ICS observable – Backdoor.Oldera



- Reported on network scanning for different ICS-specific services
- Do not specify what exactly was sent

Port number	Software that uses this port
port 44818	Rslinx
port 502	Modbus
port 102	Siemens PLC
port 11234	Measuresoft ScadaPro
port 12401	7-Technologies IGSS SCADA

Industroyer malware

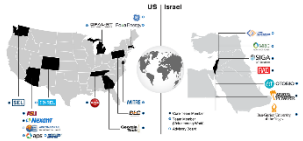


- Campaign to attack the Ukrainian power grid in 2015
- Implemented multiple ICS protocols, specifically IEC 101 and 104, which are used for controlling power systems.
- Example message from ESET's report

object tree

```
startByte1 = 0x68 = 104
blockLength = 0x9 = 9
blockLengthCopy = 0x9 = 9
startByte2 = 0x68 = 104
controlField [ControlField]
  dir = false
  prm = true
  fcb = true
  fcv = true
  functionCode = USER_DATA_CONFIRM_EXPECTED (0x3 = 3)
linkAddress = 0x1 = 1
typeIdentification = C_DC_NA_1 (0x2E = 46)
variableStructureQualifierField [StructureQualifierField]
  sq = false
  number = 0x1 = 1
causeOfTransmissionField [CauseOfTransmissionField]
  testBit = false
  positiveNegativeConfirmBit = false
  causeOfTransmission = ACTIVATION (0x6 = 6)
asduAddress = 0x0 = 0
informationObjectAddress = 0xA = 10
dco [DoubleCommandType]
  se = SELECT (0x1 = 1)
  qualifierOfCommand = NO_ADDITIONAL_DEFINITION (0x0 = 0)
  doubleCommandState = COMMAND_OFF (0x1 = 1)
checksum = 0x34 = 52
stopByte = 0x16 = 22
```

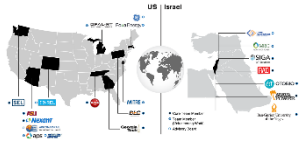
IEC 101/104 extension



- Malware using these protocols: Industroyer, Industroyer2
- Extension Structure:

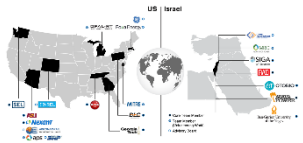
Field	Type	Description
Application Service Data Unit (ASDU) type	byte	Data object type (single, double, etc.)
ASDU address	word	Station address
Cause of Transmission(COT)	byte	Identify the reason for the message being sent
Information Object Address	3 bytes	Identify specific objects inside the station
Command state	On/Off	Relevant for specific ASDU types

Industroyer example – Turn object off



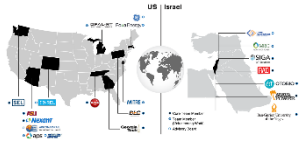
Field	Value	Description
ASDU type	46	Double Command
ASDU address	0	Station address
Cause of Transmission(COT)	6bit	Activation
Information Object Address	10	Specific object in the station
Command state	Off	Setting object 10 to OFF

IDS rules



- We have collected several repositories containing IDS rules for ICS
- These rules span multiple protocols and different suspicious behaviors
- We used these rules to build more STIX2 extensions
- Sources:
  - 60870-5-104 protocol snort rule customization (sans)
  - ICS security tools (ITI)
  - Quickdraw rules (digital bond)
  - URGENT/11 – New ICS Threat Signatures by Nozomi Networks Labs
- ≈ 100 rules (Snort + Suricata)

Modbus rule example

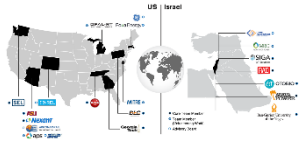


```
alert tcp $MODBUS_CLIENT any -> $MODBUS_SERVER 502 (flow:from_client,established; content:"|00
00|"; offset:2; depth:2; content:"|08 00 04|"; offset:7; depth:3; msg:"SCADA_IDS: Modbus TCP -
Force Listen Only Mode"; reference:url,digitalbond.com/tools/quickdraw/modbus-tcp-rules;
classtype:attempted-dos; sid:1111001; rev:2; priority:1;)
```

- Protocol Identifier (first detection rule) - 00H
- Function Code - Diagnostics (8H), sub-function – listen-only mode (0004H)
- Can force a device into a listen-only mode which would cause a denial of service

* https://github.com/digitalbond/Quickdraw-Snort/blob/master/modbus.rules

Suspicious functions extension

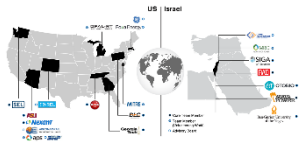


- Extension Structure:

Field	Type	Description
Protocol	Enum	The protocol being used, e.g. Modbus
Function	Enum	
Sub function	Enum	Related to the function

We can also use the actual codes to represent the functions (hexa)

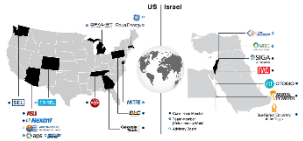
Modbus example revisited



```
alert tcp $MODBUS_CLIENT any -> $MODBUS_SERVER 502 (flow:from_client,established; content:"|00
00|"; offset:2; depth:2; content:"|08 00 04|"; offset:7; depth:3; msg:"SCADA_IDS: Modbus TCP -
Force Listen Only Mode"; reference:url,digitalbond.com/tools/quickdraw/modbus-tcp-rules;
classtype:attempted-dos; sid:1111001; rev:2; priority:1;)
```

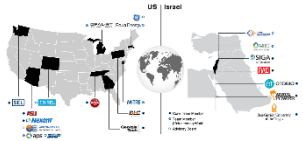
Field	Value
Protocol	Modbus
Function	Diagnostics
Sub function	Listen-only mode

Generalized STIX2 Extensions



- The extensions presented are protocol specific
- We are aiming at generating a more generalized extension(s)
- We are starting an effort to map operational ICS protocols into a common vocabulary
  - This would allow us to provide a unified structure for multiple protocols

Current status and plans



- Current assets
  - Multi-level CTI ontology for enterprise
  - Multi-level Naive Bayes method for techniques inference
  - SHAP based method for explaining anomalies
- Ongoing
  - Discussion with OTORIO regarding technique-observable data generation
  - Populating the ontology using CTI data (enterprise + ICS)
  - Extending the STIX2 framework for ICS
  - Experiment with Energy Dept. IL
- Plans
  - Applying anomaly detection to existing datasets with labeled techniques
  - Populating the ontology using simulated data
  - Populating the ontology using expert-based data
  - Closing the loop with unexplainable anomaly detection