Adaptive and Proactive Security Assessment on Energy Delivery Systems

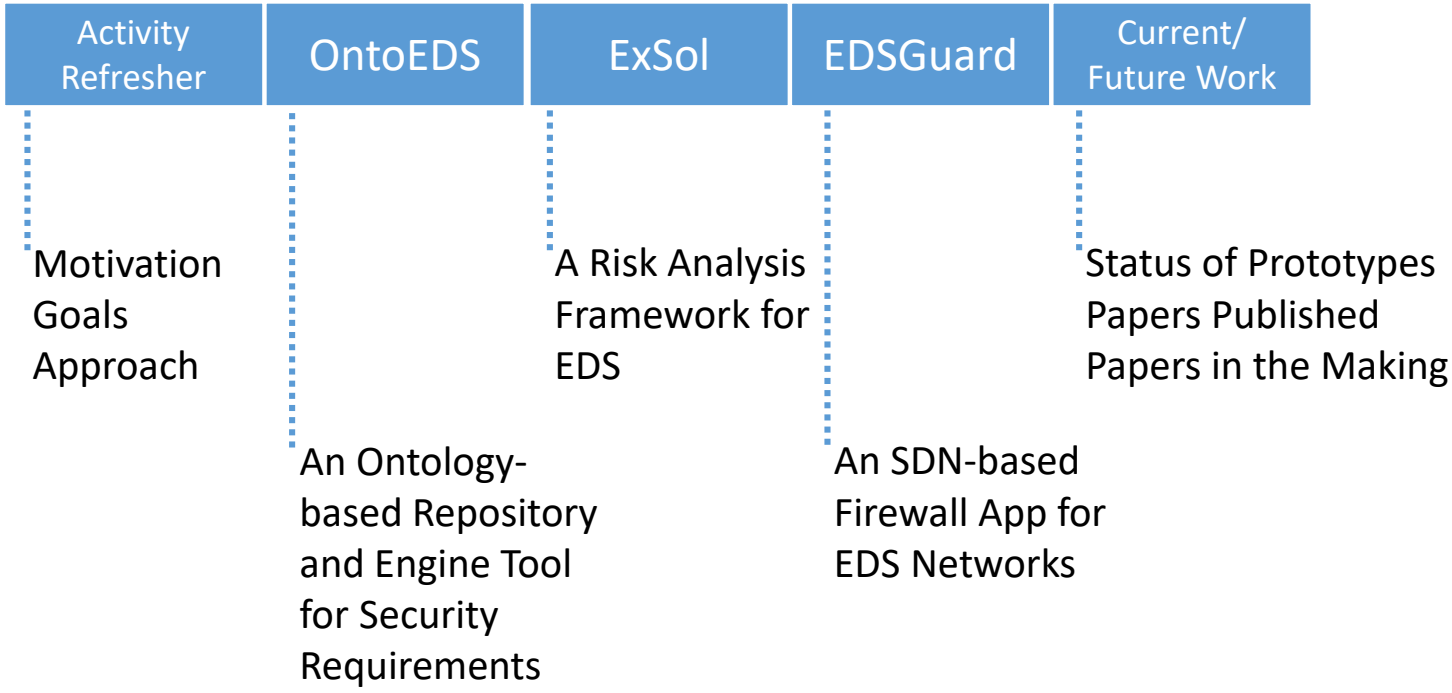
Carlos Rubio-Medrano, Vu Coughlin, Josephine Lamp,
Ziming Zhao, Gail-Joon Ahn and Anna Scaglione

ASU Arizona State University

Center for Cybersecurity & Digital Forensics

CREDC Cyber Resilient Energy Delivery Consortium

Outline


Activity Refresher
OntoEDS
ExSol
EDSGuard
Current/ Future Work
Motivation
Goals
Approach
An Ontology-based Repository and Engine Tool for Security Requirements
A Risk Analysis Framework for EDS
An SDN-based Firewall App for EDS Networks
Status of Prototypes
Papers Published
Papers in the Making

Activity Refresher

Motivation

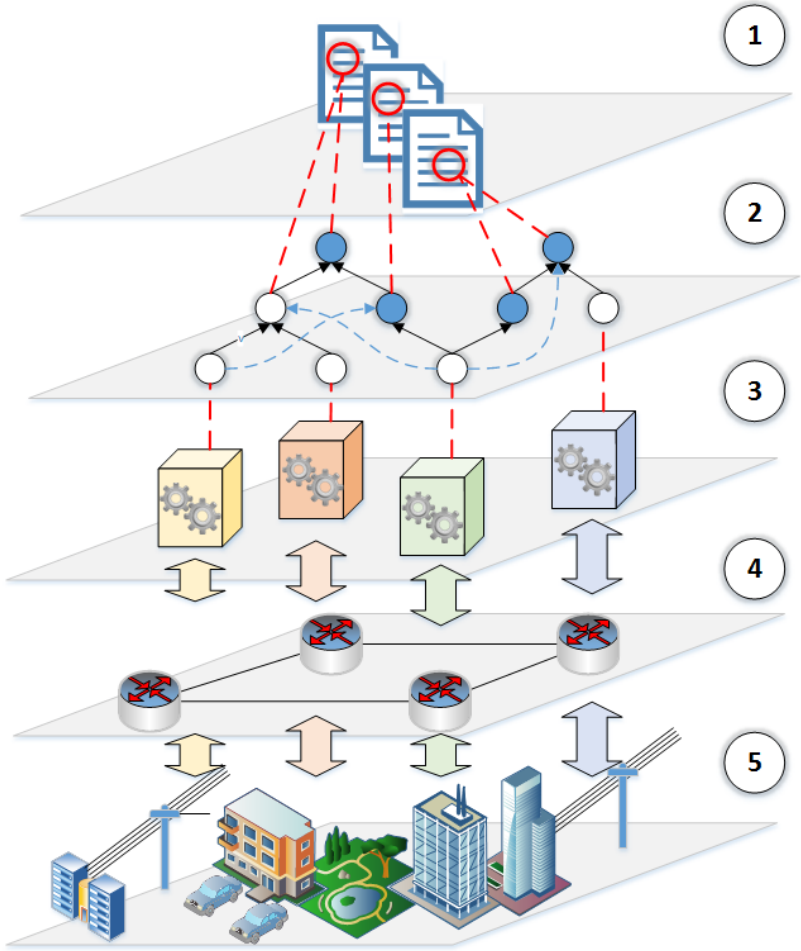
- Security assessment in EDS gets complicated due to:
  - The distributed, highly-interconnected and heterogeneous nature of EDS, e.g., monitoring software, meters, etc.
  - Continuous reconfigurations due to *on-demand* changes,
  - The existence of multiple, large, dense (and sometimes conflicting) documents on security requirements,
    - E.g., subjective interpretations, non-standard implementations, and breakdowns among stakeholders

Goals

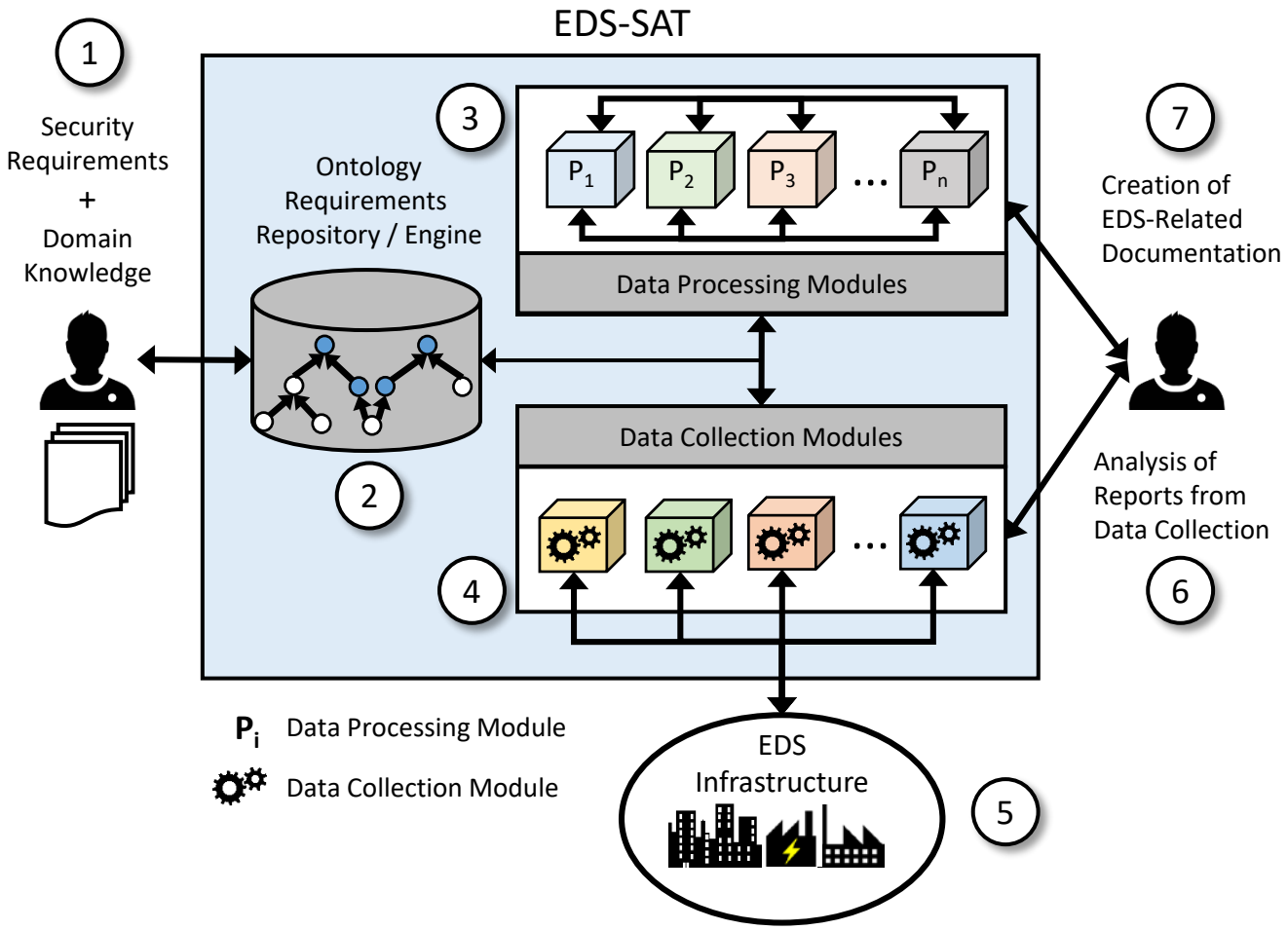
- Assess if particular EDS implementations meet security requirements,
  - Filling in the gap between high-level requirements and field implementations,
- A framework for security assessment and monitoring:
  - Well-defined (theoretically-justifiable),
  - Systematic and automated (repeatable to validate),
  - Practical and configurable (deployable to organizations),
  - Non-intrusive (minor overhead/reconfiguration as possible)

Our Approach (Big Picture)

1. We gather the most relevant documents on best practices for EDS
2. Next, we obtain a description of such best practices by leveraging ontologies
3. We then introduce software-based modules for security monitoring and risk analysis
4. Data from EDS infrastructure (5) is collected and forwarded for further processing



The *EDS-SAT* Security Assessment Framework

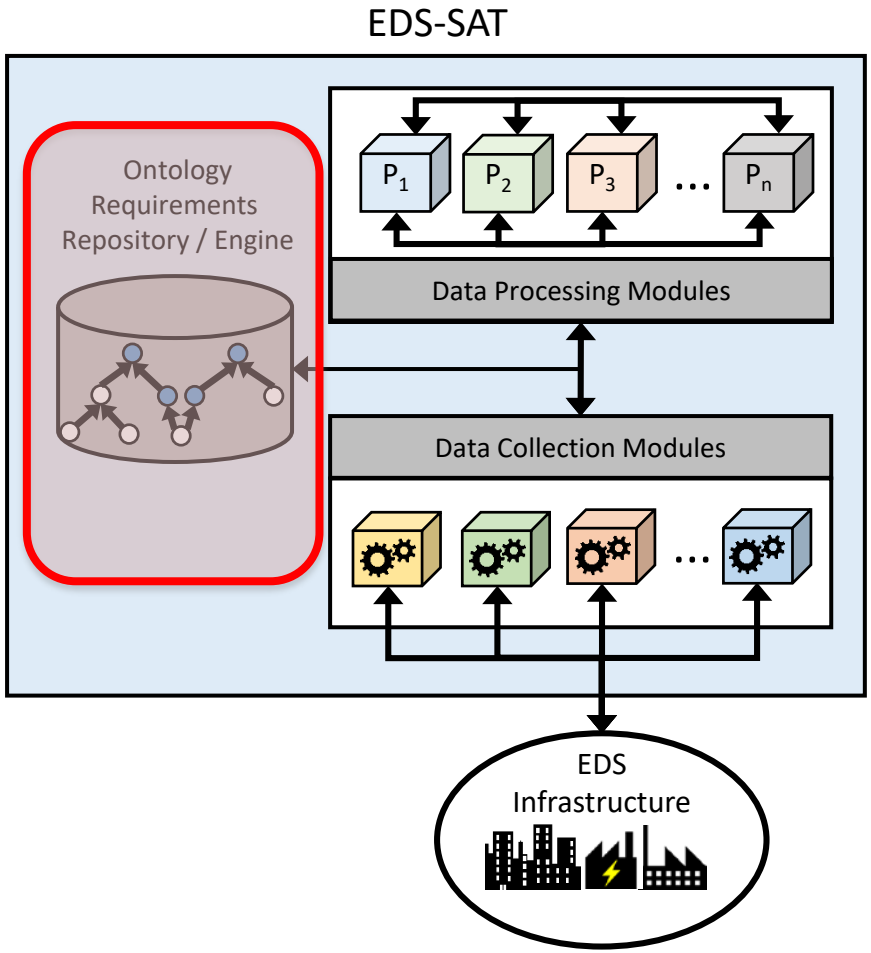


- Encourages the rigorous analysis of security requirements,
- Continuously monitors the security of EDS infrastructure,
- Promotes the development of objective, traceable, justifiable and repeatable security metrics

OntoEDS: Modeling Security Requirements for EDS Using Ontologies

The *OntoEDS* Security Requirements Engine

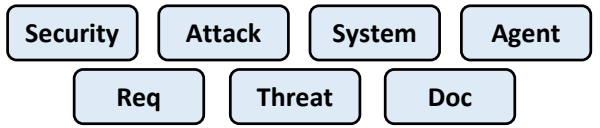
- Unambiguously represents *common vulnerabilities and exposures* (CVEs) *,
- Identifies interdependencies, missing and conflicting information among diverse knowledge sources,
- Supports multiple *dimensions* and *viewpoints*, e. g., relevant information for operators vs vendors

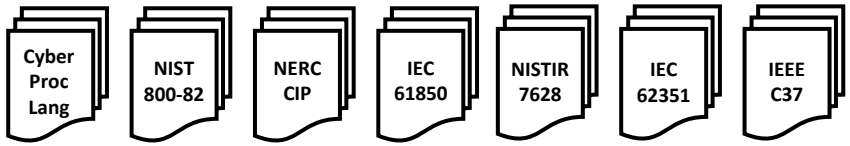


OntoEDS: Modeling Security Requirements

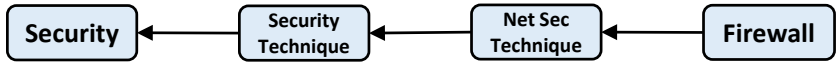
1. Develop supporting foundation structure of ontology
2. Identify and collect key documents
3. For each document, extract key entities from sentences or paragraphs
4. Categorize each entity within the hierarchy structure of the ontology
5. Identify relationships for the defined entity
6. Model the relationships based on predefined characteristics/definitions

Repeat for each paragraph within each document





*"A technique to prevent **integrity** violations of data is the use of **firewalls**, such as application-level firewalls that employ **application filtering**"*
Entities: **Firewall, Integrity, Application Filtering**

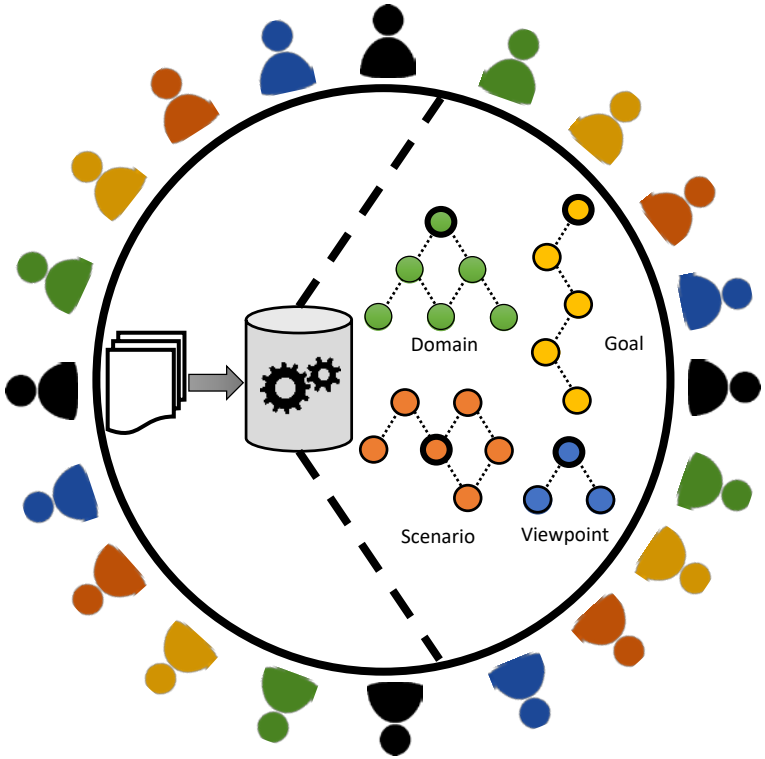


*"A technique to **prevent** integrity violations of data is the use of firewalls, such as application-level firewalls that **employ** application filtering"*
Relationships: **prevent, employ**

OntoEDS: Current State of Ontology

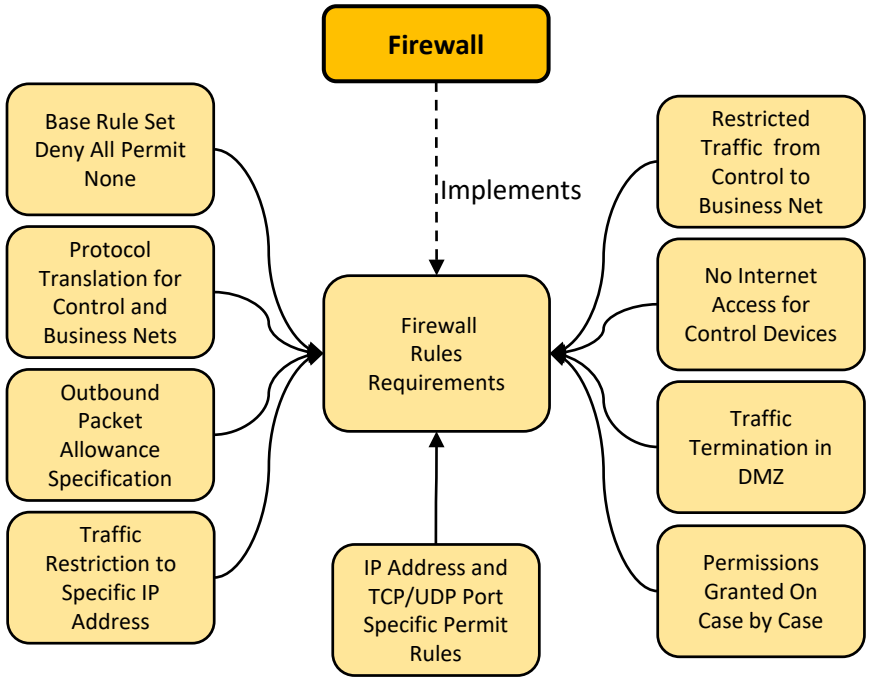
- Comprises more than 300 pages of source documents and includes 600 entities with over 1,700 relationships,

- Currently models the following:
  - Cybersecurity Procurement Language for Energy Delivery Systems developed by the Energy Sector Control Systems Working Group (ESCSWG),
  - NIST 800-82 Special Publication,
  - North American Electric Reliability Corporation Critical Infrastructure Protection (NERC CIP) standards,
  - NISTIR 7628 document,
  - IEEE C37 standards,
  - IEC 61850 and 62351 standards



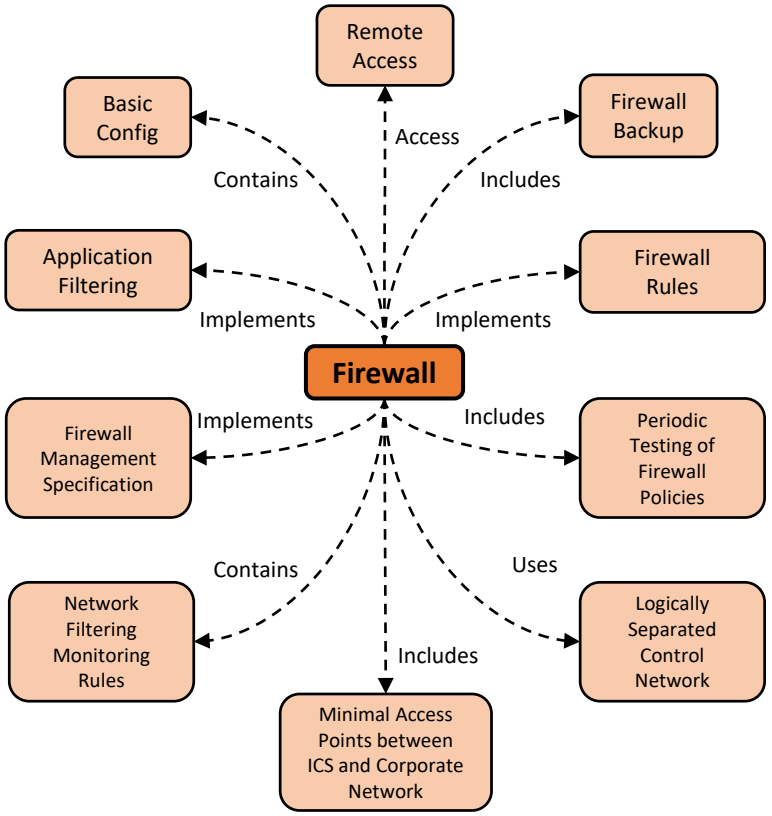
OntoEDS: Analyzing Requirements with Projections

- Goal Projection: Contains objectives the system must achieve to enter into a state of security:
  - Protect system components,
  - Implement security techniques/features,
  - Defend against an attack type,
  - Identify purposes or properties of system components,
  - Protect security principles



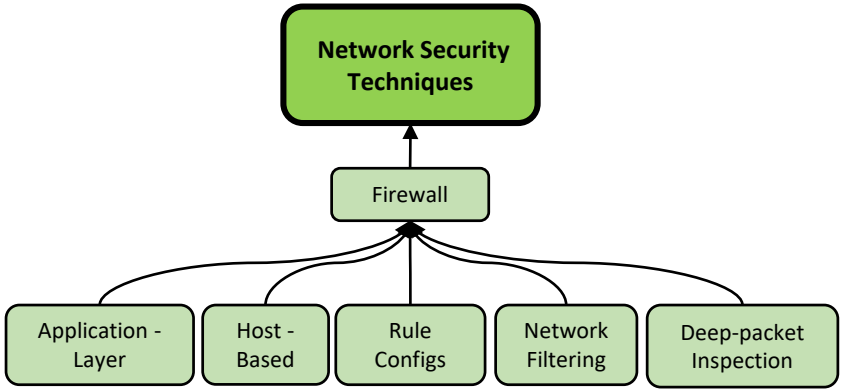
OntoEDS: Analyzing Requirements with Projections (II)

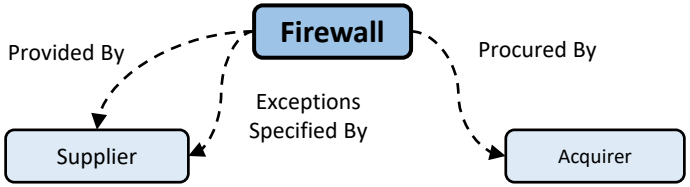
- Scenario Projection: Facts describing a system that include agent behavior and environmental context:
  - Identifies dependencies between the system and its environment,
  - Storyline of events describing system operation,
  - Enables the understanding of a broad picture of ontology elements and their relationships



OntoEDS: Analyzing Requirements with Projections (III)

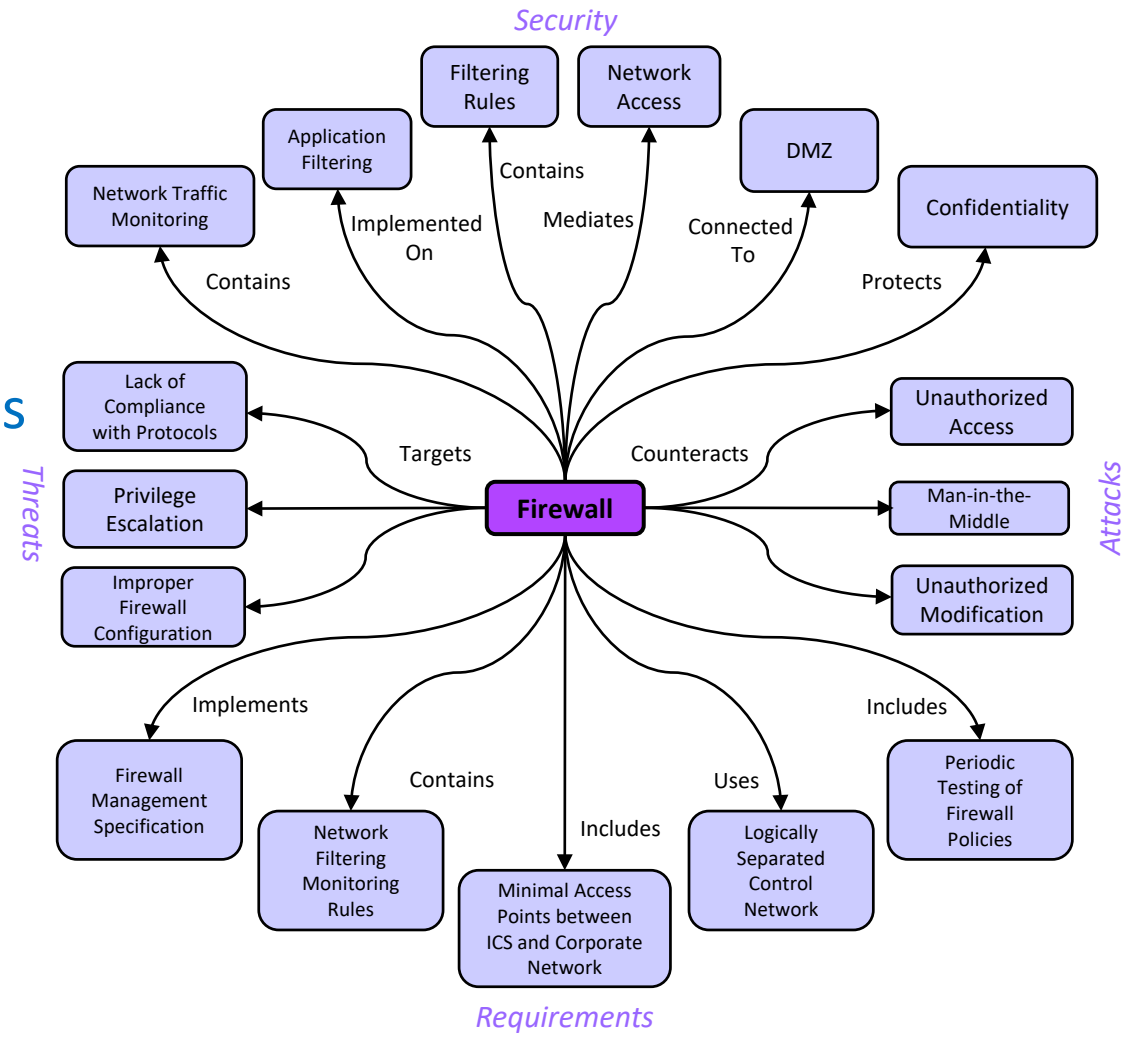
- Domain Projection: Describes a domain taxonomy relative to a specific topic,
  - May support knowledge exploration,
  - Combined with Goal Projection helps identifying inter-dependencies and missing requirements,
- Viewpoint Projection: Retrieves specific responsibilities of an agent,
  - May support knowledge acquisition,





OntoEDS: Analyzing Requirements with Projections (IV)

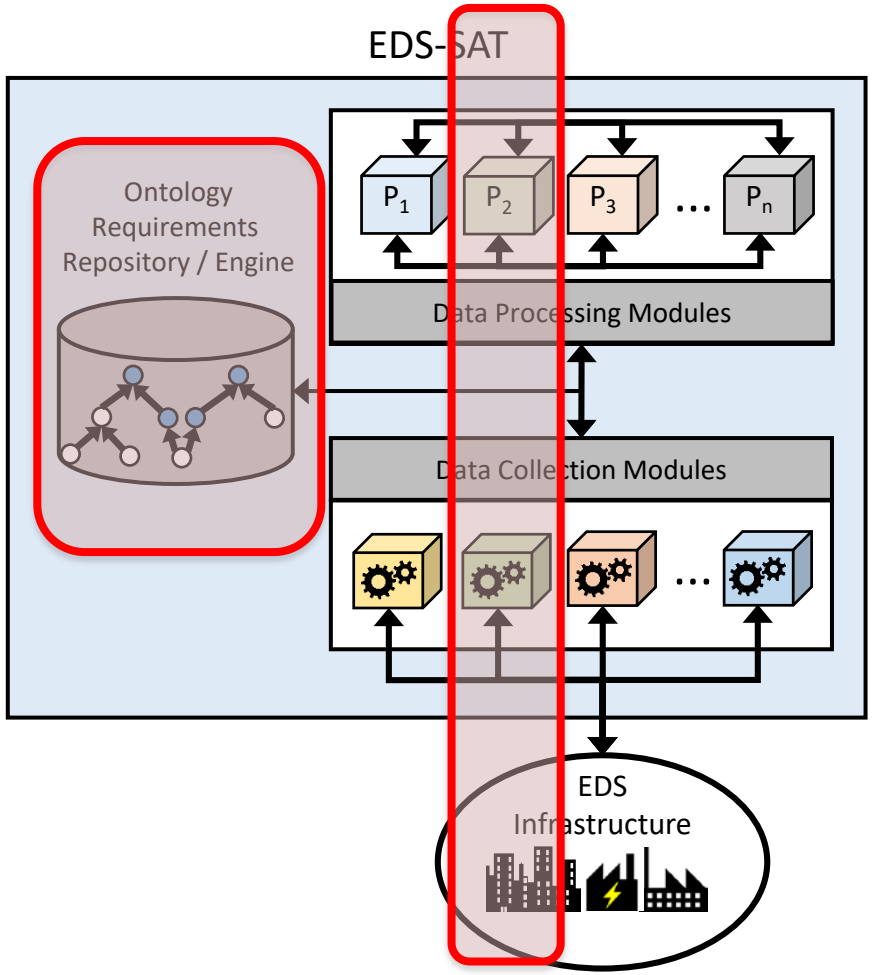
- Risk Analysis Projection: Use a series of goal projections to elucidate threats, attack types, security countermeasures and requirements surrounding an *asset*,
  - Retrieves specific concepts in risk analysis methodologies (to be shown later),



ExSol: A Risk Analysis Framework based on Security Requirements for EDS

The Exploitation-Solution (*ExSol*) Framework

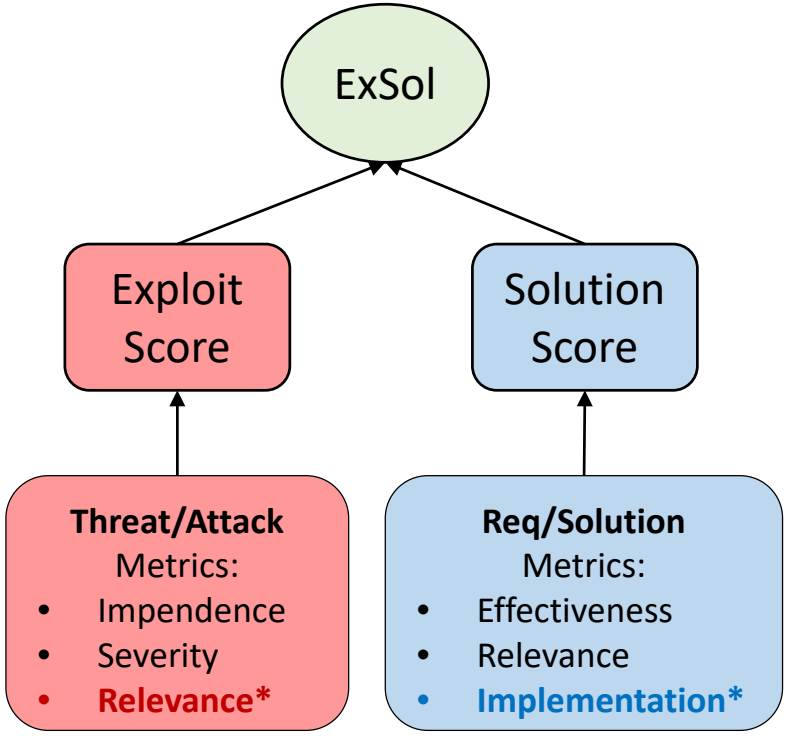
- Leverages *OntoEDS* and *EDS-SAT* for risk analysis and mitigation,
- Elucidates metrics that are cohesively combined in a mathematical model,
- Risk = the probability that a particular threat will exploit a particular vulnerability of a system*



*Vaughn, Rayford B., Ronda Henning, and Ambareen Siraj. "Information assurance measures and metrics-state of practice and proposed taxonomy." In *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, pp. 10-pp. IEEE, 2003.

The ExSol Risk Score

- Combines different metrics into a single score to understand the risk of a system,
- Exploitation metrics and Solution metrics are matched up against one another,
- Each metric's *sub-score* is calculated on a scale from 1 (least) to 5 (greatest),
- Scores determined collaboratively by global and/or local experts,
- Calculated for an asset, but can be done for threats and attacks as well,

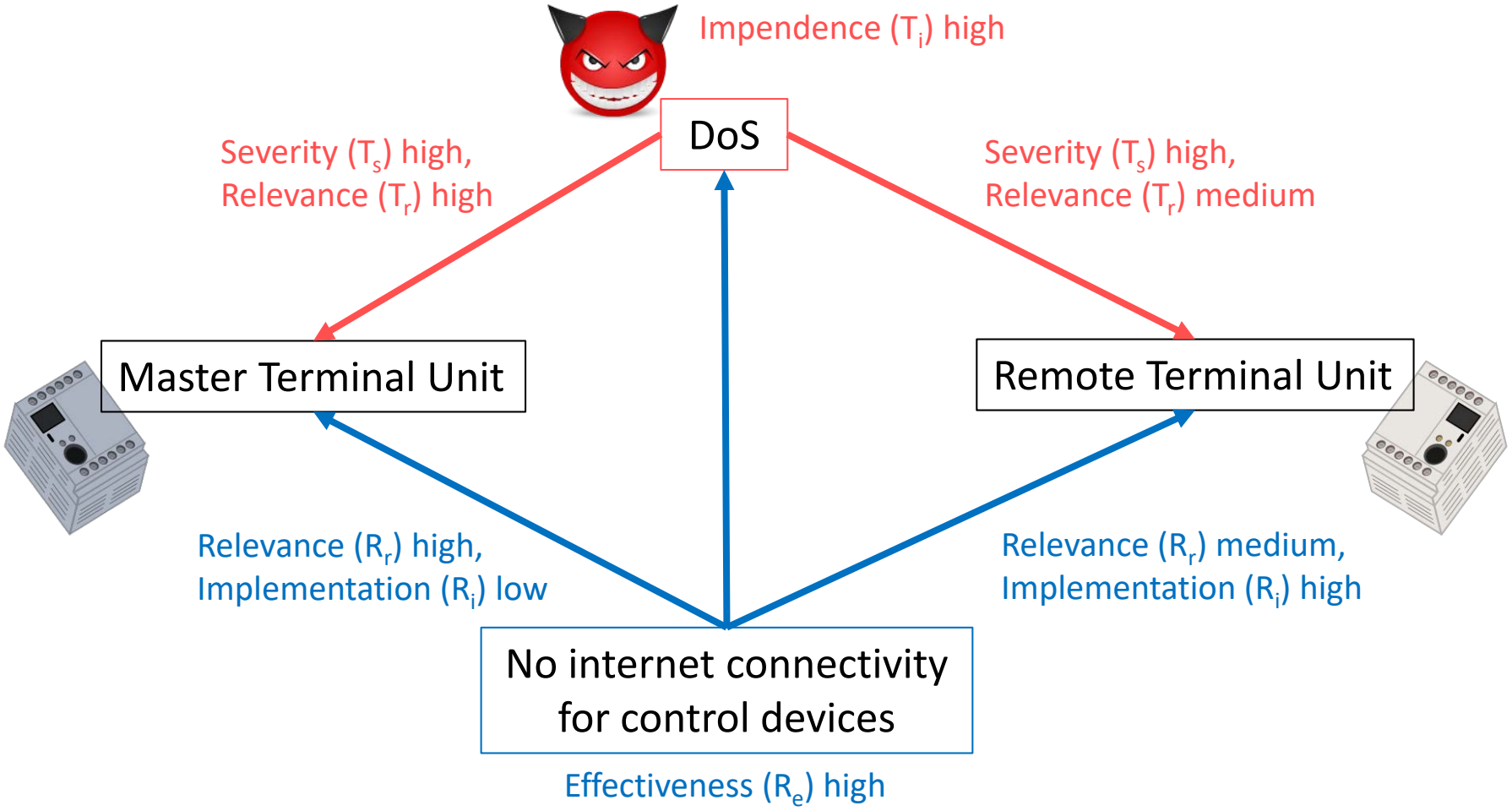


*** Sub-scores calculated using EDS-SAT processing modules**

Exploitation / Solution Score Metrics

	Metric	Definition	Defined By
Exploitation	Impendence (T_i)	Likelihood/Frequency of threat being exploited or attack being performed.	Global / Local Expert
Exploitation	Severity (T_s)	Impact and damage of threat/attack on the asset.	Global / Local Expert
Exploitation	Relevance (T_r)	How applicable or targeted to the asset the threat/attack is.	Local Expert
Solution	Effectiveness (R_e)	Perception on the ability of the requirement to deter/counteract an attack/threat.	Global / Local Expert
Solution	Relevance (R_r)	Applicability of a requirement to the asset being analyzed.	Global / Local Expert
Solution	Implementation (R_i)	Perception on the effectiveness of the implementation of a given the requirement in the system.	Local Expert

ExSol Score Metric Example



ExSol Risk Score Calculation

- Exploitation Sub-score:
  - For each Threat / Attack:
    - (T/A) = T_i * T_r * T_s
- Solution Sub-score:
  - For each Requirement / Security:
    - (R/S) = R_e * R_r * R_i

- ExSol Score = Solution Sub-score – Exploitation Sub-score

ExSol > 0: Good, the greater the better
ExSol = 0: Matched
ExSol < 0: Bad, the lower the worse

↑ Solution	↑ Exploitation	OKAY
↑ Solution	↓ Exploitation	GOOD
↓ Solution	↑ Exploitation	BAD
↓ Solution	↓ Exploitation	OKAY

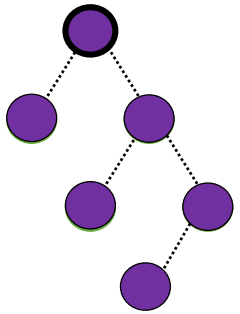
ExSol Calculation Algorithm

1. Retrieve all Threats (T), Attacks (A), Requirements (R) and Security Techniques (S) **related to a given asset** using the Risk Projection,

2. Match T, A, R and S that are *relevant* to each other, creating 4-tuples of the form: <T, A, R, S>,

3. For each TARS-tuple:
   1. Calculate the exploitation and solution sub-scores of each T, A, R and S,
   2. Calculate the ExSol score,

4. Evaluate risk based on the obtained ExSol scores



Risk

Disgruntled Employees

Network Backdoors/ Holes

No Unnecessary Ports

Permissions

$< T_1, A_1, R_1, S_2 > =$

$(80 * 100) - (18 * 180) =$

$4{,}760$

ExSol Risk Score Example: Network Access Point

(T/A)	Disgruntled Employees (T_1)	Unnecessary Ports (T_2)	Network Backdoors/ Holes (A_1)	Spoofing (A_2)
Impendence	3	5	4	1
Severity	2	5	5	2
Relevance	3	4	5	2
Sub-score	**18**	**100**	**180**	**4**

(R/S)	Firewall (S_1)	Permissions (S_2)	Network Segregation (S_3)	Network Segmentation (S_4)	Network Intrusion Detection (S_5)	No Unnecessary Ports (R_1)	No Internet for Control Devices (R_2)	Enable Only Ports Needed (R_3)
Effectiveness	4	4	4	4	2	5	4	4
Relevance	3	5	3	3	3	4	5	4
Implementation	4	5	3	5	4	4	5	4
Sub-score	**48**	**100**	**36**	**60**	**24**	**80**	**100**	**64**

ExSol Risk Score Example: Network Access Point (II)

T_1: Disgruntled Employees
A_1: Network Backdoors/ Holes

R_1 : No Unnecessary Ports
R_2 : No Internet for Control Devices
R_3 : Enable Only Ports Needed
S_2 : Permissions
S_3 : Network Segregation
S_4 : Network Segmentation
S_5: Network Intrusion Detection

1. $< T_1, A_1, R_1, S_2 > = (80 * 100) - (18 * 180) = 4{,}760$
2. $< T_1, A_1, R_2, S_2 > = (100 * 100) - (18 * 180) = 6{,}760$
3. $< T_1, A_1, R_1, S_5 > = (80 * 24) - (18 * 180) = -1320$
4. $< T_1, A_1, R_3, S_5 > = (64 * 24) - (18 * 180) = -1704$
5. $< T_1, A_1, R_2, S_3 > = (100 * 36) - (18 * 180) = 360$
6. $< T_1, A_1, R_1, S_4 > = (80 * 60) - (18 * 180) = 1{,}560$

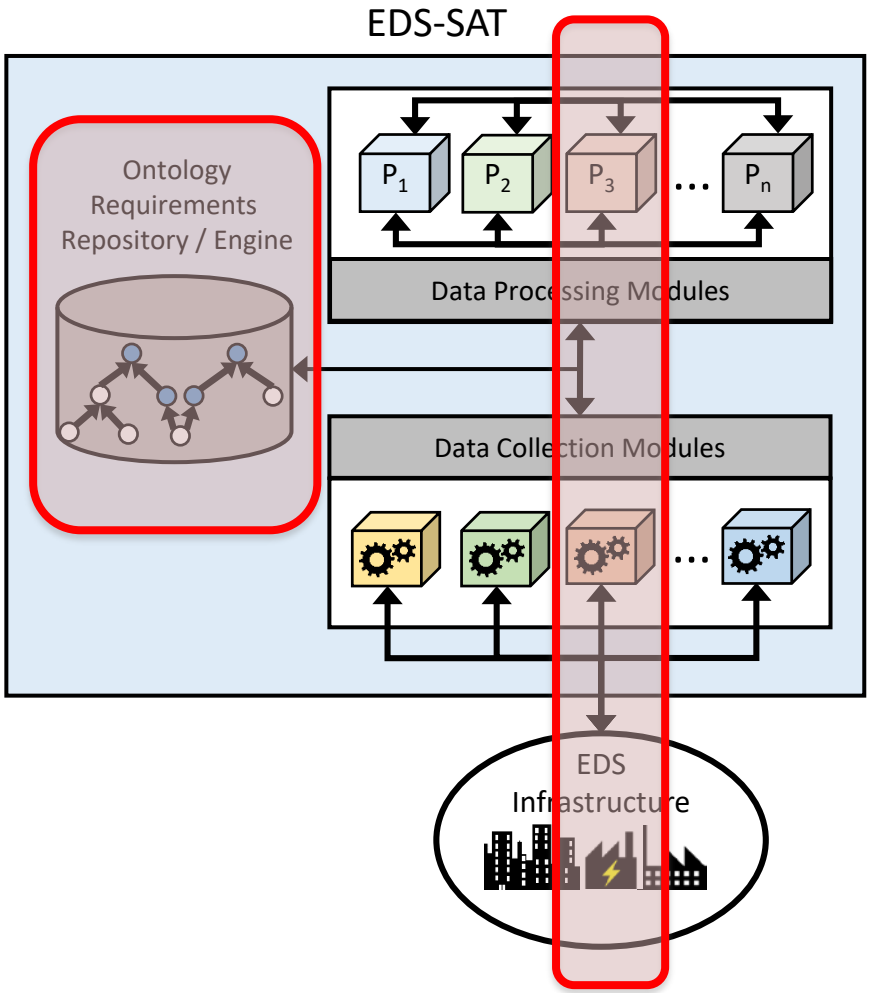
EDSGuard: Enforcing Security Requirements for EDS Networks

The *EDSGuard* SDN-based Firewall App

- Enforces security requirements on EDS firewalls **continuously over time**,

- Leverages:
  - *OntoEDS*,
  - *EDS-SAT*,
  - *Software-defined Networking* (SDN),
  - *State-of-the-art* Firewall Policy Management,

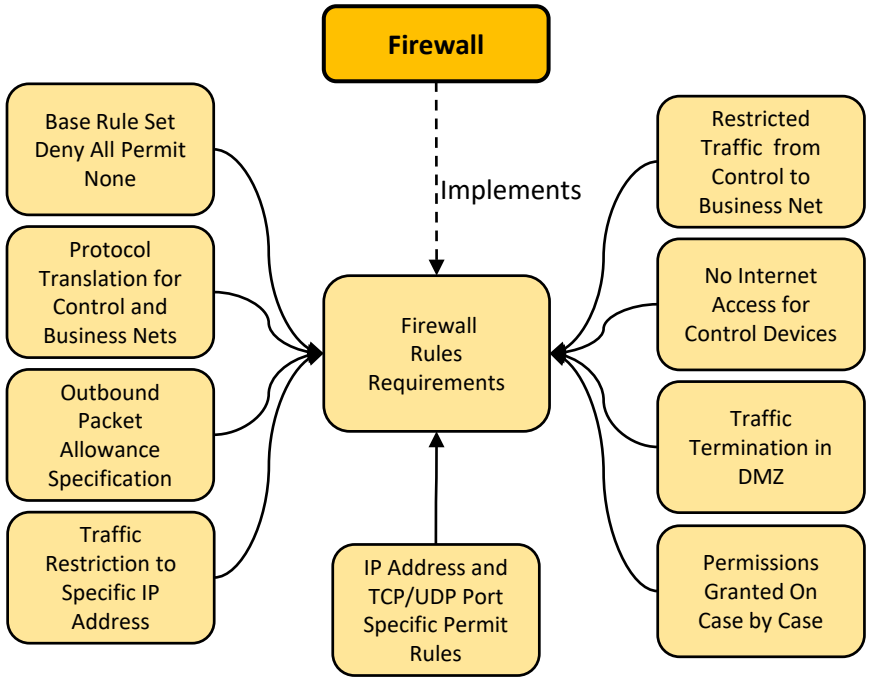
- Intended to deter recent attacks that leveraged erroneous firewall configurations, e.g., Ukraine 2015[1], CrashOverride[2]



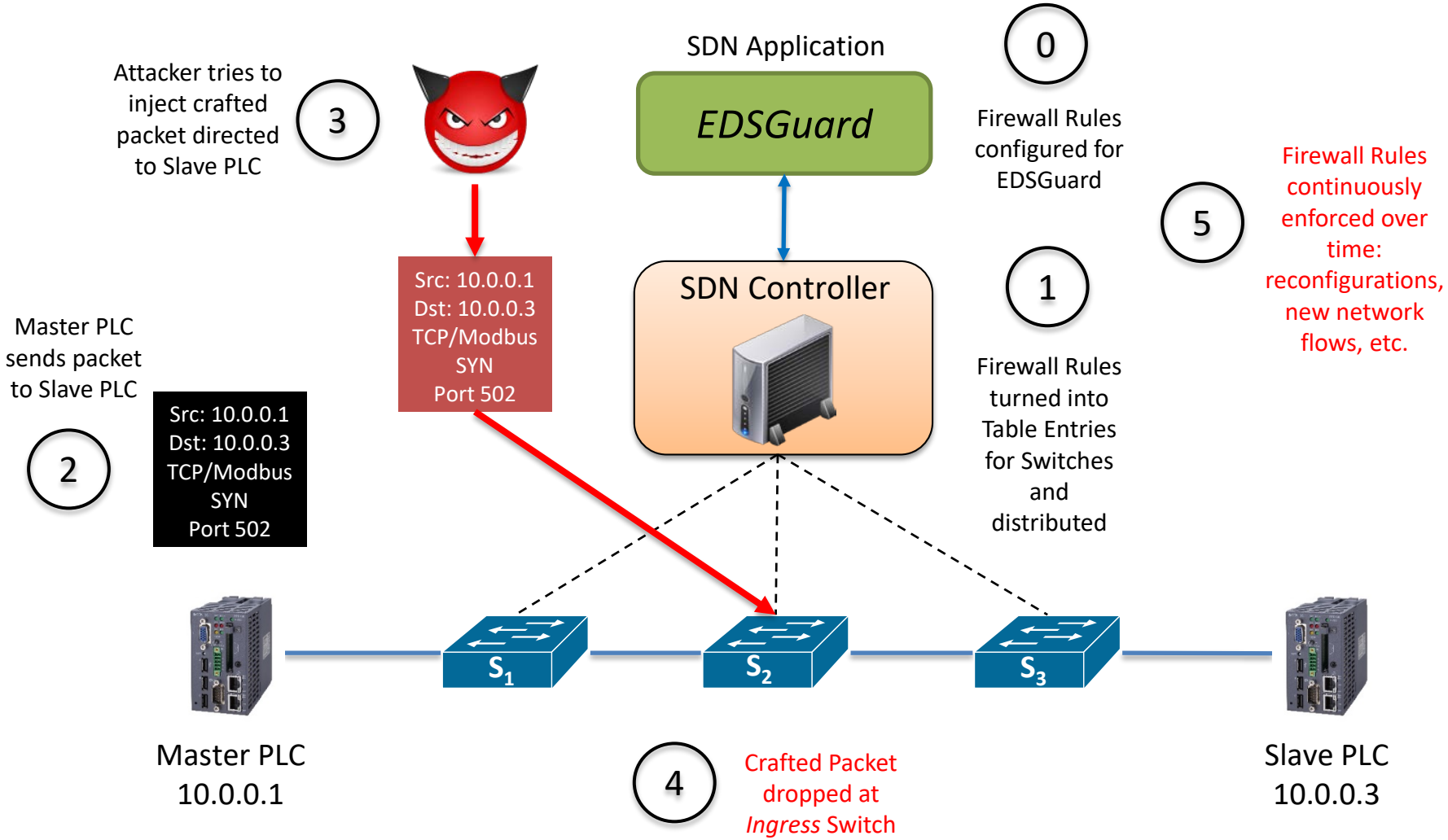
1) R. M. Lee, M. J. Assante, and T. Conway, "Analysis of the Cyber Attack on the Ukrainian Power Grid," *SANS ICS Report*, 2016.
2) Dragos Inc. "CrashOverride: Analyzing the Threat to Electric Grid Operations", Technical Report, 2017.

EDSGuard: Security Requirements

- Extracted from *OntoEDS* using Goal Projections,
- Depicts requirements for Firewall Rules and Network Topology,
- Derived from different documents, e.g., IEC 62351, NIST 800-82, Cybersecurity Procurement Language Document, etc.

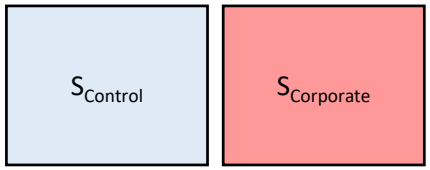


EDSGuard: Overall Approach

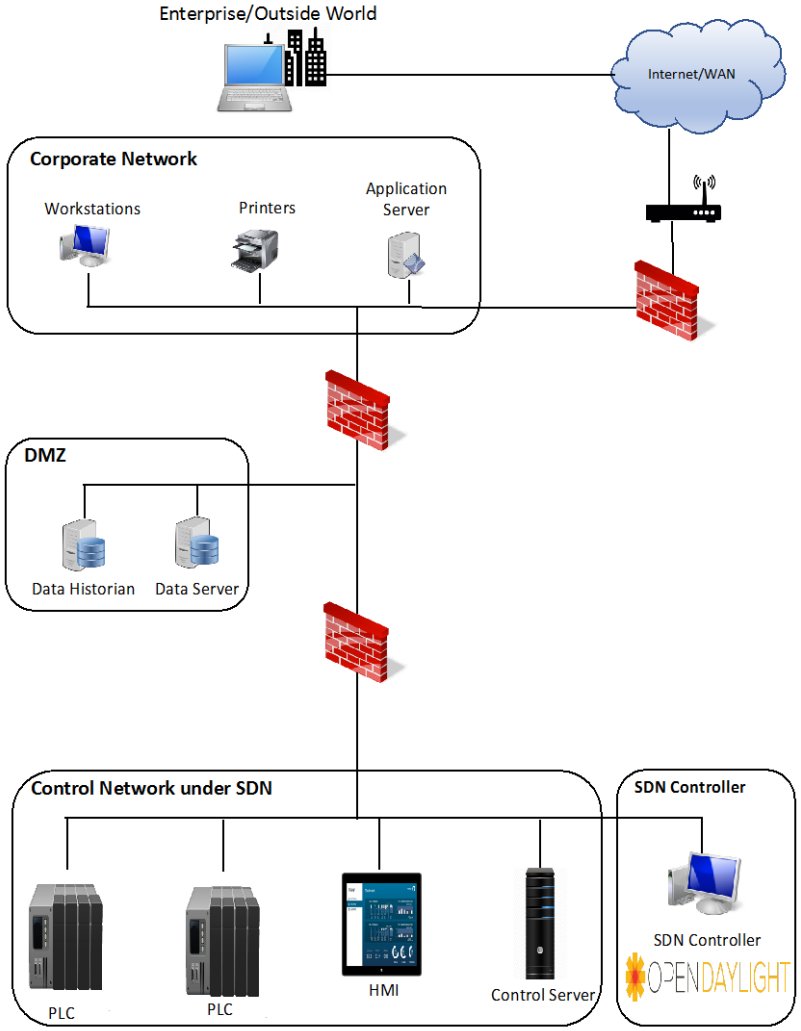


EDSGuard: Requirements Example

- *Traffic should be prevented from transiting directly from the control network to the corporate network,*
- Enforcement based on *authorization spaces*[1]:



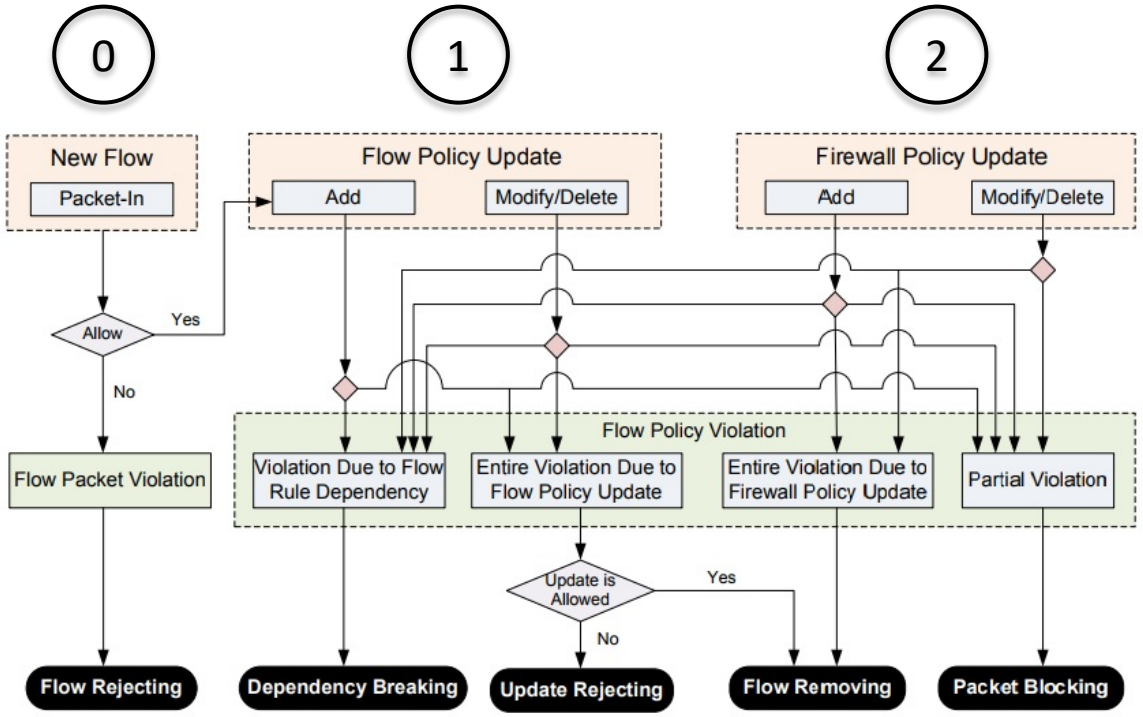
  - Disjoint spaces created for each network,
  - Switch entries derived from them,
  - Future network flows violating spaces detected and removed,



1) Discovery and Resolution of Anomalies in Web Access Control Policies. Hongxin Hu, Gail-Joon Ahn and Ketan Kulkarni. IEEE Transactions on Dependable and Secure Computing (**TDSC**), 2013

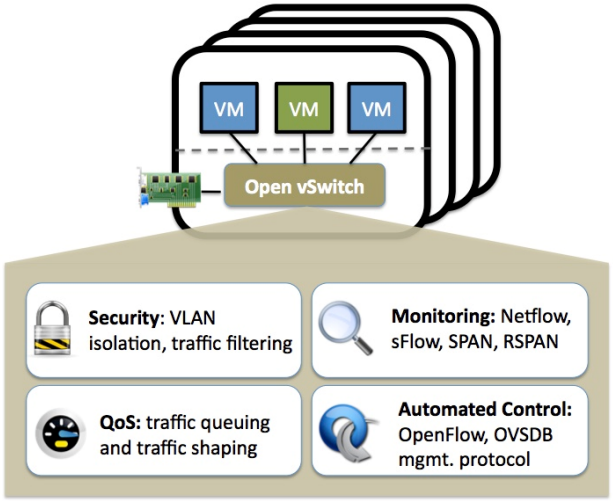
EDSGuard: Detection/Resolutions

- Different detection and resolution strategies available,
  - This way, *EDSGuard* not only detects violations, but can proactively solve them as well,
- *EDSGuard* may then serve as an effective *first-response* countermeasure tool for handling security incidents,



EDSGuard: Experimental Testbed

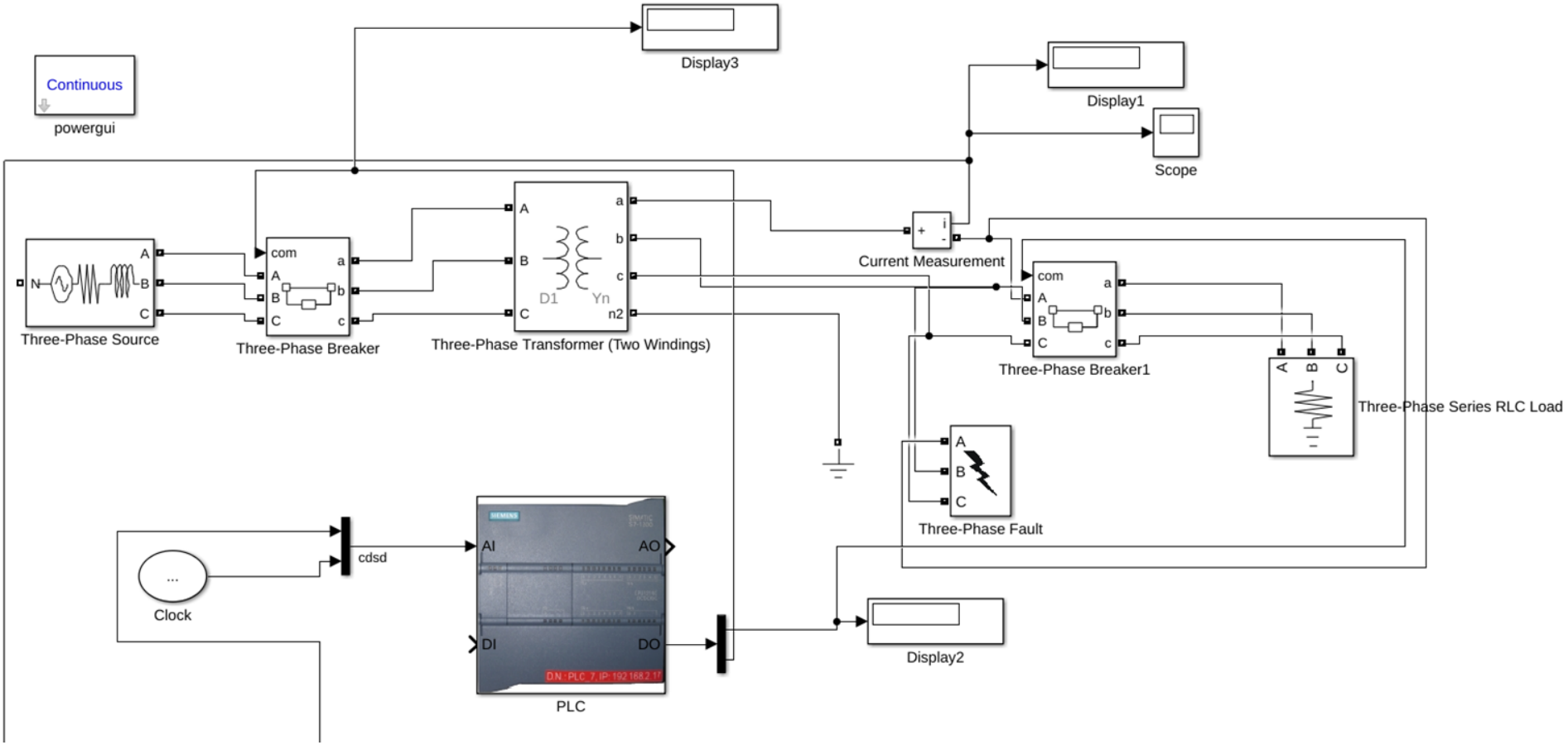
- VM1: Slave_PLC with Matlab simulator + `libmodbus`
- VM2: Master_PLC with `libmodbus` library
- VM3: Attacker with `libmodbus` library



OPEN DAYLIGHT

VirtualBox

EDSGuard: Matlab Simulator

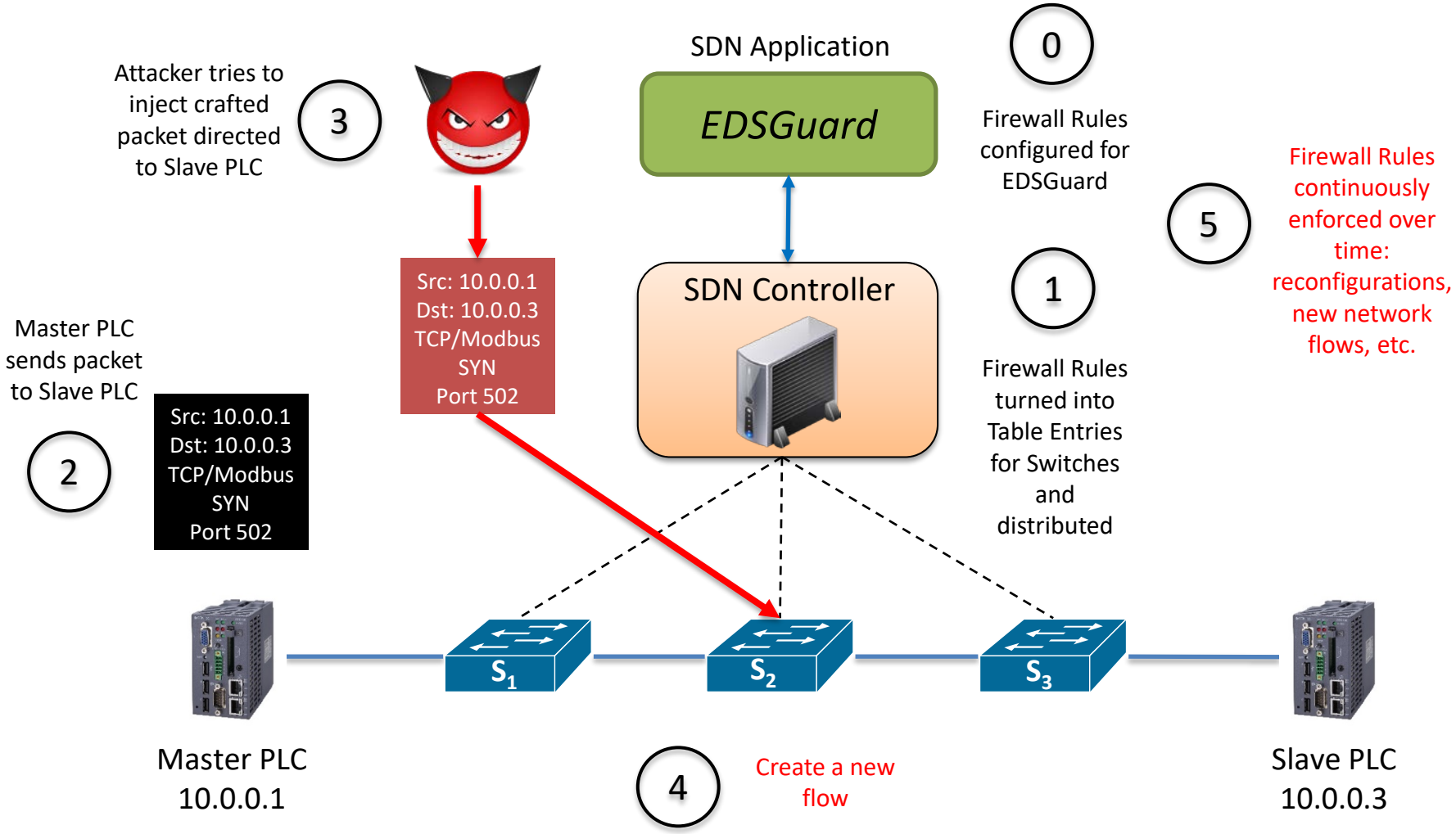


EDSGuard: Firewall Rule Format

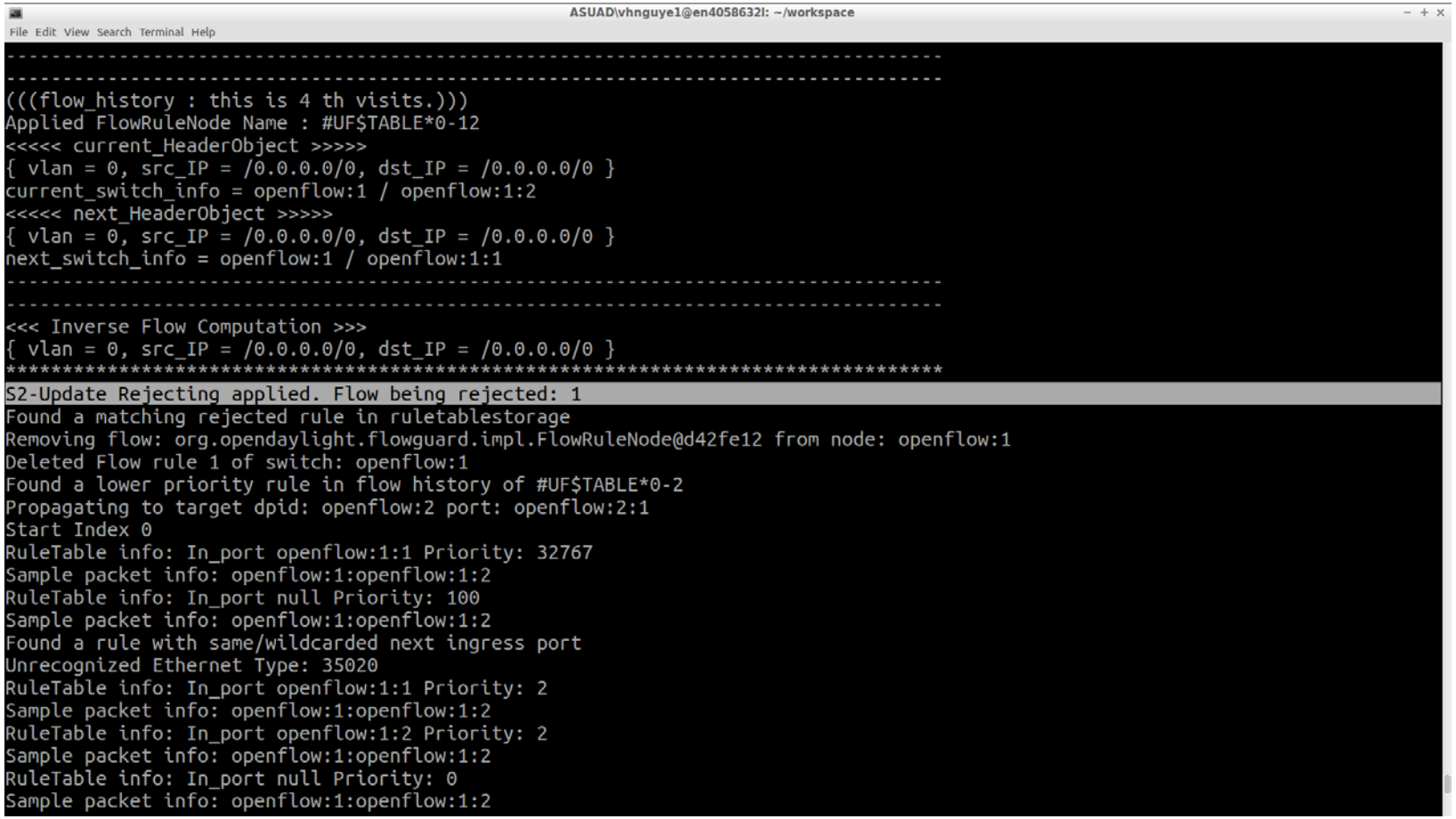
- Rule ID: unique ID for the firewall rules,
- Node: Ppenflow switch appears on controller,
- In Port: the interface of the switch,
- Source and Destination IPs,
- Source and Destination Ports,
- Action: Allow/Deny

```
{
    "fwrule-registry-entry": [
        {
            "ruleId": "1",
            "node": "openflow:1",
            "inPort": "openflow:1:1",
            "priority": "50",
            "sourceIpAddress": "10.0.0.2/32",
            "destinationIpAddress": "10.0.0.1/32",
            "sourcePort": "",
            "destinationPort": "502",
            "action": "deny"
        }
    ]
}
```

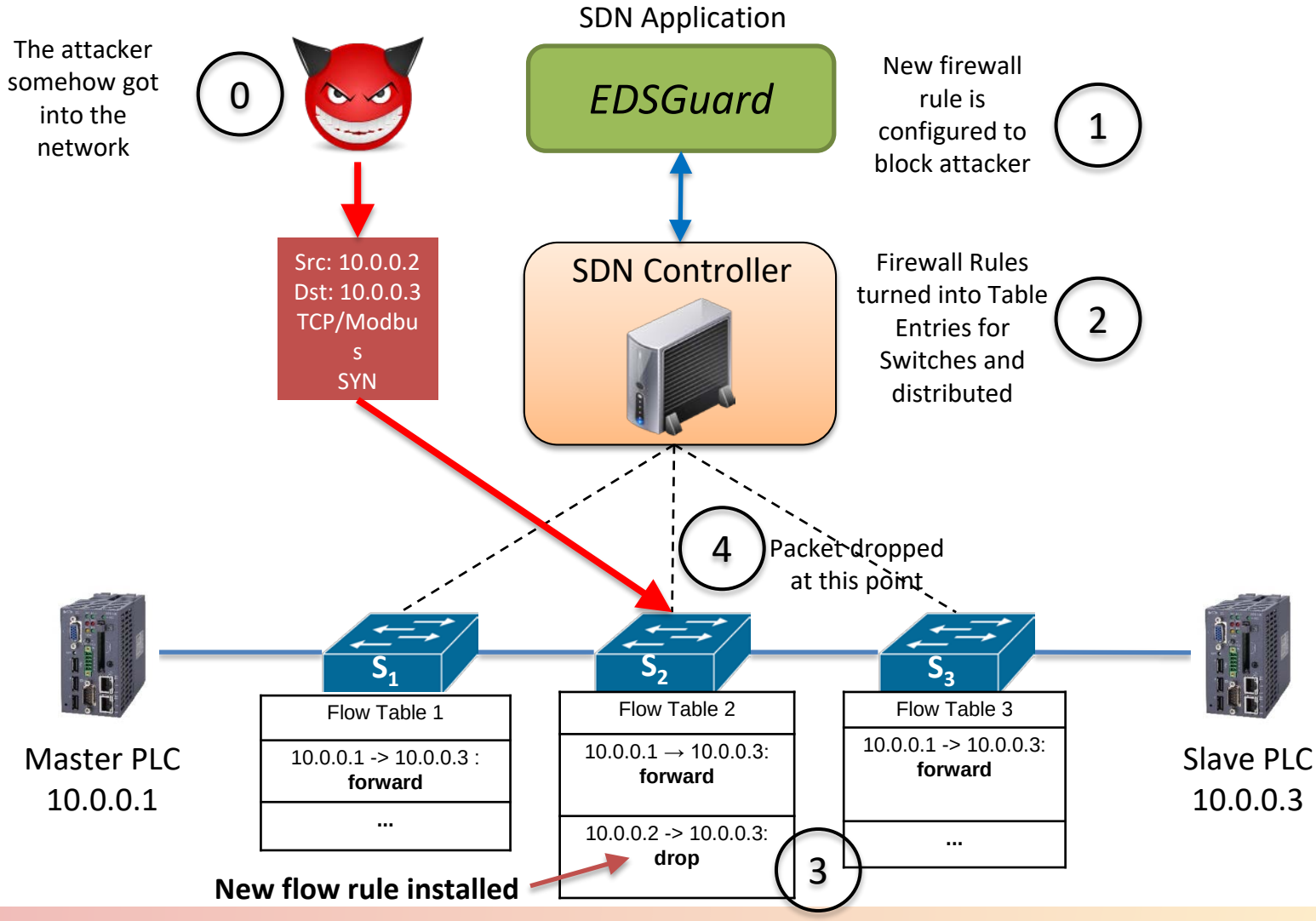
EDSGuard: Flow Update Rejection



EDSGuard: Flow Update Rejection



EDSGuard: Packet Blocking



EDSGuard: Packet Blocking Resolution

Controller retrieves all flows → Controller cross-checks flows with firewall rules for partial violations → Controller installs a flow in the switch to drop the traffic

```
*** s1 ----------------------------------------------------------------------
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x2b00000000000005, duration=82.772s, table=0, n_packets=0, n_bytes=0, priority=100,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x2b0000000000000e, duration=80.750s, table=0, n_packets=2, n_bytes=140, priority=2,in_port=1 actions=output:2,output:3,CONTROLLER:65535
 cookie=0x2b0000000000000d, duration=80.750s, table=0, n_packets=2, n_bytes=140, priority=2,in_port=2 actions=output:1,output:3,CONTROLLER:65535
 cookie=0x2b0000000000000f, duration=80.750s, table=0, n_packets=2, n_bytes=140, priority=2,in_port=3 actions=output:2,output:1,CONTROLLER:65535
 cookie=0x2b00000000000005, duration=82.772s, table=0, n_packets=0, n_bytes=0, priority=0 actions=drop
```

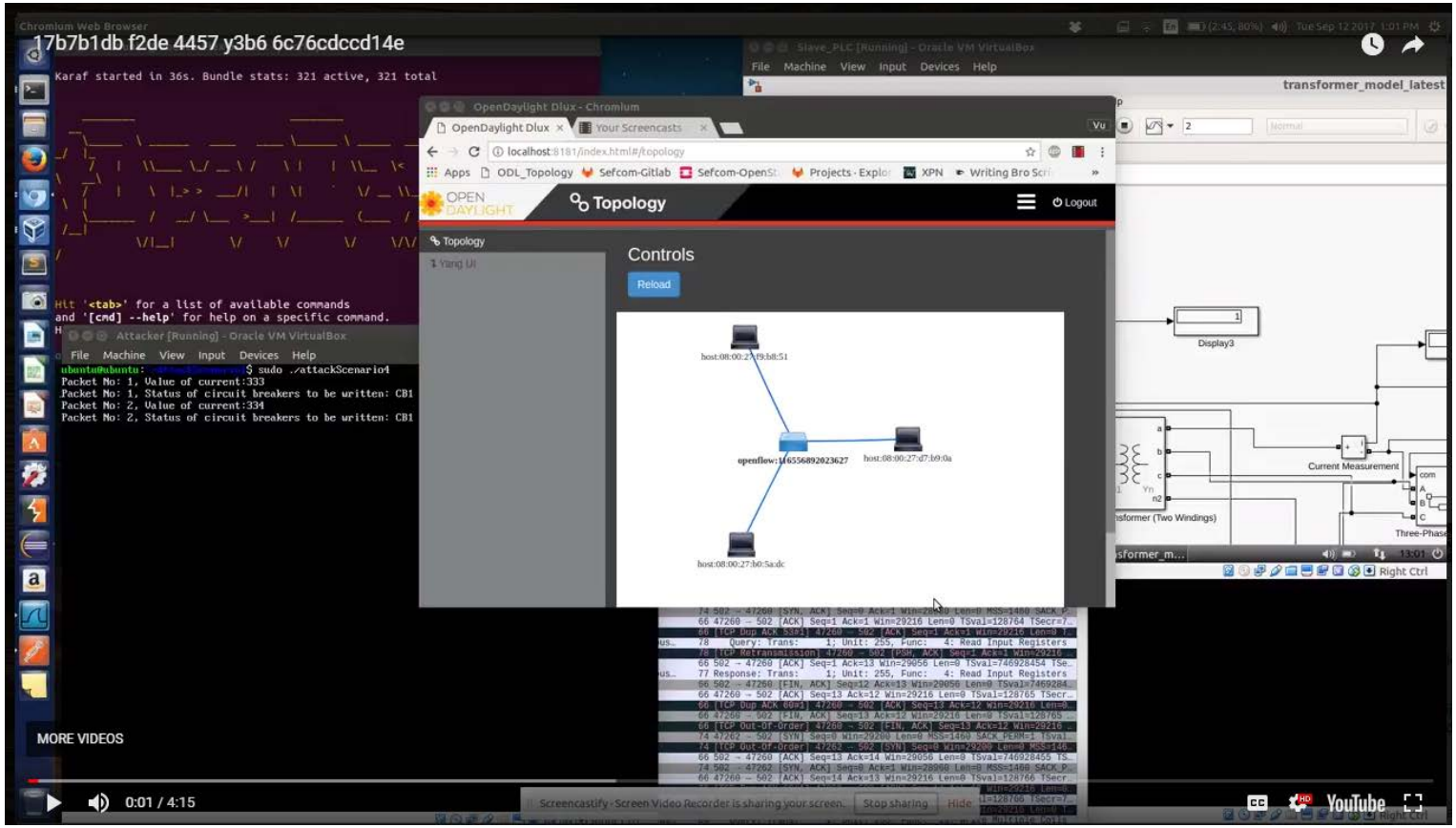
Before resolution

```
*** s1 ----------------------------------------------------------------------
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=808.408s, table=0, n_packets=0, n_bytes=0, priority=32767,ip,in_port=2,nw_src=10.0.0.2,nw_dst=10.0.0.1 actions=drop
 cookie=0x2b00000000000004, duration=971.563s, table=0, n_packets=0, n_bytes=0, priority=100,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x2b0000000000000a, duration=969.560s, table=0, n_packets=5, n_bytes=322, priority=2,in_port=2 actions=output:1,output:3,CONTROLLER:65535
 cookie=0x2b0000000000000c, duration=969.560s, table=0, n_packets=11, n_bytes=854, priority=2,in_port=3 actions=output:2,output:1,CONTROLLER:65535
 cookie=0x2b0000000000000b, duration=969.560s, table=0, n_packets=10, n_bytes=756, priority=2,in_port=1 actions=output:2,output:3,CONTROLLER:65535
 cookie=0x2b00000000000004, duration=971.563s, table=0, n_packets=2, n_bytes=84, priority=0 actions=drop
```

After resolution

EDSGuard: Demo Video on YouTube

- https://youtu.be/1ihcFO0BVLw



Current and Future Work

Current and Future Work

- *OntoEDS*:
  - Paper accepted for publication at IEEE CIC 2017,
- *ExSol*:
  - Working on refining mathematical model and case study,
  - Introducing *reference* ExSol scores for Attacks/Threats for comparison,
  - Paper expected by the end of the Fall 2017 semester,
- *EDSGuard*:
  - Working on initial prototype and experimental setup,
  - Paper expected by the end of the Fall 2017 semester,
- *EDS-SAT*:
  - Introductory Paper published at IEEE MSCPES 2017,
  - Working on incorporating the aforementioned tools as modules,
  - Detailed Paper expected by Second Quarter of 2018,

Thank you all for listening!

- Time for Q & A !

- Contact:
  - ASU Center for Cybersecurity and Digital Forensics:
    https://globalsecurity.asu.edu/cdf
  - Josephine Lamp: jalamp@asu.edu
  - Vu Couhglin: vhnguye1@asu.edu
  - Carlos Rubio-Medrano: crubiome@asu.edu