Continuous Security Monitoring Techniques for Energy Delivery Systems

Armin Rahimi, Adam Hahn, Mathew Merrick

Washington State University

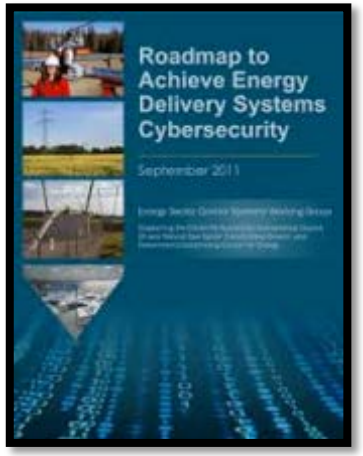
CYBER RESILIENT ENERGY DELIVERY CONSORTIUM

Funded by the U.S. Department of Energy and the U.S. Department of Homeland Security | cred-c.org

Problem

"***Continuous security state monitoring*** of all energy delivery system architecture levels and across cyber-physical domains is widely adopted by energy sector asset owners and operators"

– DOE Roadmap to Achieve Energy Delivery Systems Cybersecurity Year 2020 Goal



What to monitor?

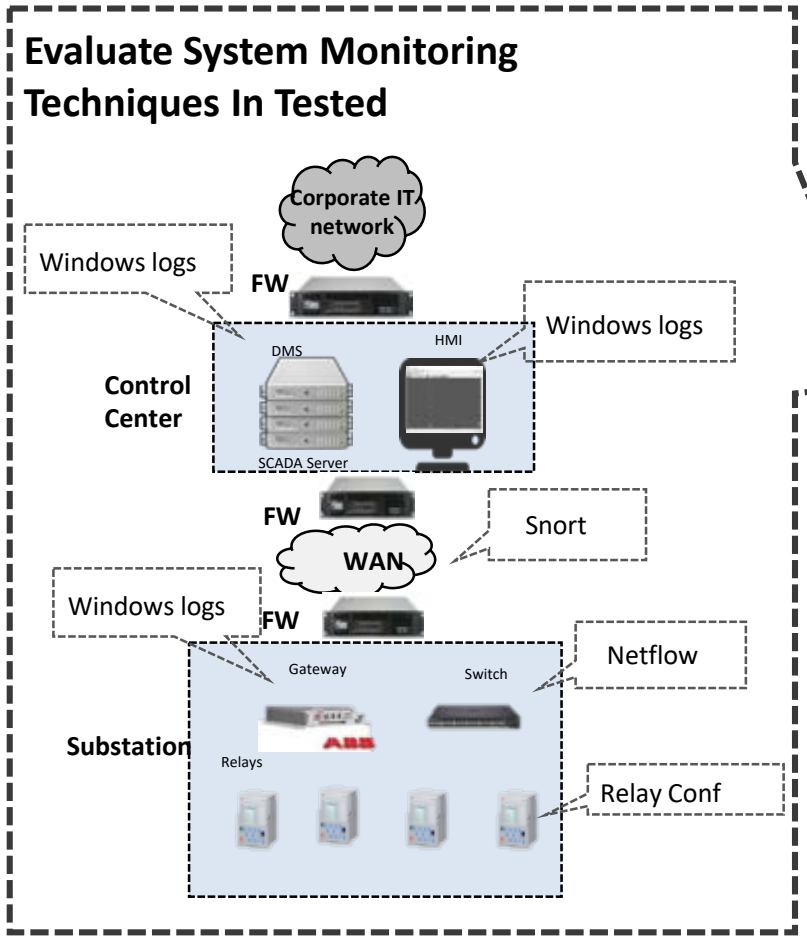
- Unclear what data is available
  - Configuration vs Events
  - System vs Network
  - Granularity
- Unclear how valuable various data is
  - False negatives
  - False positives
  - Information overload

Current Technologies

	Continous monitoring/real time detection	Risk assesment	Asset discovery	Use of AI
Awareness Solutions				
CyberX				
Claroty				
Darktrace				
Dragos Security				
Indegy				
NexDefense				
Nozomi Networks				
SecurityMatters				
Utilidata				
Integrated Solutions	Generic IT product. Not geared towards PG	Generic IT product. Not gear	Generic IT product. Not geared towards	Generic IT product. Not geared
Aperio				
CyberArk				
Nextnine				
Sentryo				
Veracity Industrial Networks				

- Cybex: Continuous real time threat monitoring, asset discovery, use of AI
- Claroty: Continuous monitoring, Risk assessment
- Sentryo: Continuous monitoring, asset discovery, vulnerability management, user defined severity
- Veracity Industrial Networks: Continuous monitoring, security zone creation

What to Monitor?

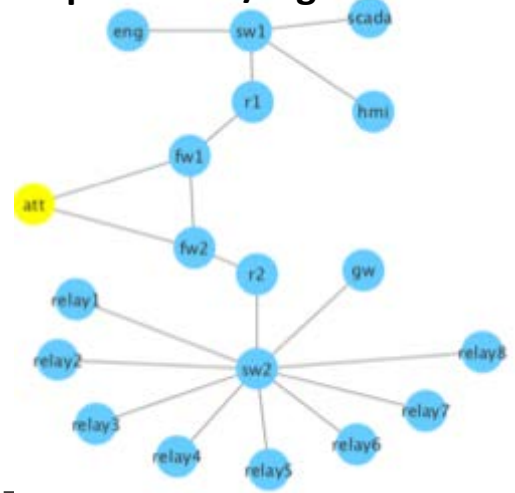


Approach

1. Identify feasible system attack paths
2. Determine coverage of attack monitoring capabilities
3. Develop system-level coverage metrics

MITRE ATT&CK based Tactic and Techniques

Graph Modes/Algorithms



What Mechanisms?

Questions: **1)** What feature set is important to detect malicious activity? → **2)** What mechanisms must be deployed to detect those feature?

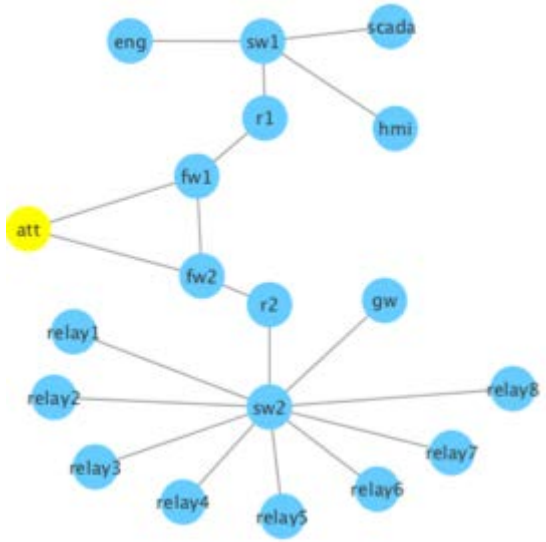
Features*

Mechanisms	Network		Process		File/Registry		Authentication			Power Systems	
	Flows	Payload	Exec	...	Read	Write	Name	Succ	Fail	Meas.	Control
Switch/NF	✓										
IDS	✓	✓								✓	✓
Win Logs			✓	✓			✓	✓	✓		
AV			✓	✓	✓	✓					
App Logs										✓	✓

* *Rough estimate*

System Model

System Model

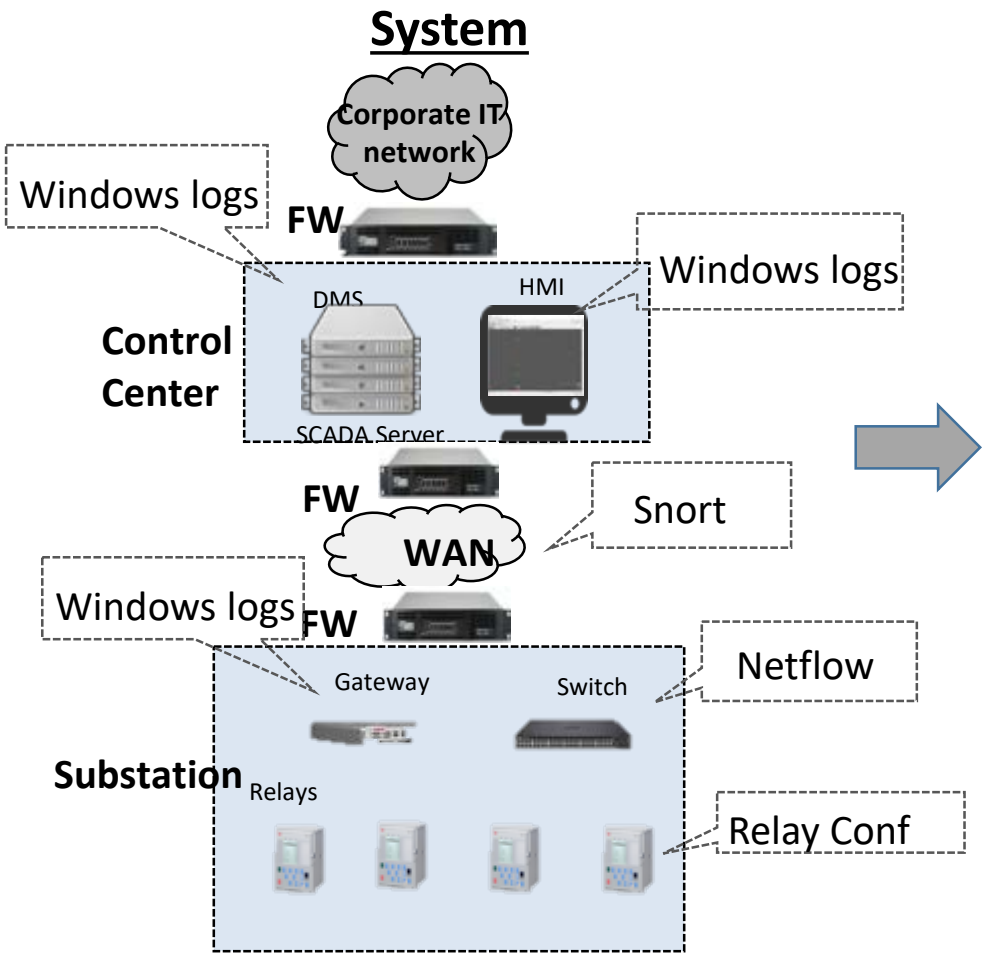


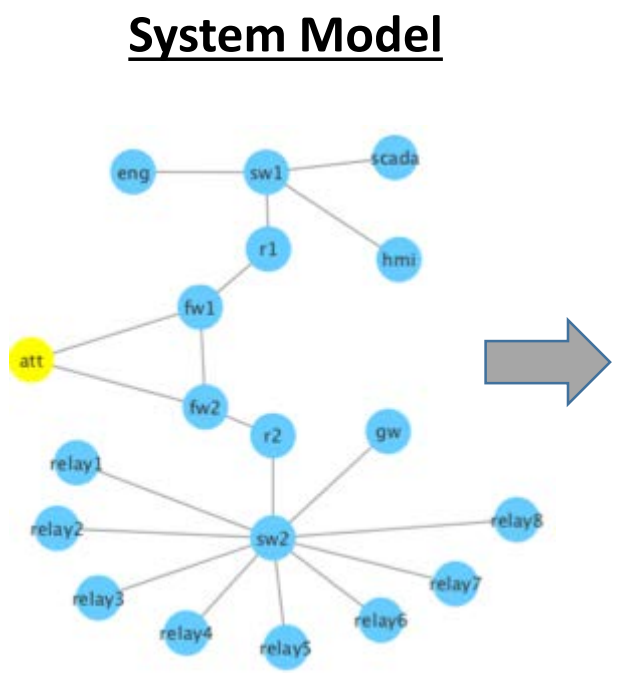
For each node define attributes based on

- Configurations
  - i) operating system
  - ii) enabled services
- Monitoring capabilities
  - Mechanisms (NF, IDS, Event Logs, etc) and features
- Attacks
  - MITRE ATT&CK (Tactics and Techniques)

System Model - Monitoring strategy

Define the set of deployed monitoring mechanisms





Monitoring Strategy

S: $N(G) \rightarrow M$

M = {NF, IDS, Event Log, …}

m_i = {NF, BRO, Snort, Winlog, relay config}

Example

- S(fw1) = {BRO}
- S(fw2) = {BRO}
- S(sw1) = {NF}
- S(sw2) = {NF}
- S(hmi) = {Win Evt}
- S(dms) = {Win Evt}
- S(gw) = {Win Evt}

CREDC

Attack Techniques - MITRE ATT&CK…

Persistence	Privilege Escalation	Defense and Operator Evasion	Credential Access	Discovery	Lateral Movement	Execution	Collection	Exfiltration	Command and Control	Disruption	Destruction
External Remote Service	Exploitation of Vulnerability	Alternate Modes of Operation	Brute Force	Account Enumeration	Default Credentials	API Interaction	Automated Collection	Automated Exfiltration	Commonly Used Port	Alternate Modes of Operation	Alternate Modes of Operation
Firmware	Loadable Module	Block Comm Port	Create Account	Control Process	Exploitation of Vulnerability	Alternate Modes of Operation	Data Staged	Data Compressed	Communication Through Removable Media	Block Comm Port	Block Command Message
Interactive Service	Valid Accounts	Block Reporting Message	Credential Dumping	File and Directory Enumeration	External Remote Service	Command-Line Interface	Data from Local System	Data Encoding	Connection Proxy	Block Command Message	Block Reporting Message
Loadable Module	Web Shell	Code Signing	Credentials in Files	I/O Module Enumeration	Man in the Middle	Exploitation of Vulnerability	Data from Network Service	Data Encrypted	Custom Command and Control Protocol	Block Reporting Message	Command-Line Interface
Modify Control Logic		Exploitation of Vulnerability	Default Credentials	Local Service Enumeration	Remote File Copy	Graphical User Interface	Data from Network Share	Data Transfer Size Limits	Custom Cryptographic Protocol	Command-Line Interface	Device Shutdown
Modify System Settings		File Deletion	Exploitation of Vulnerability	Location Identification	Replication Through Removable Media	Interactive Service	Data from Removable Media	Exfiltration Over Alternative Protocol	Data Encoding	Device Shutdown	Exploitation of Vulnerability
Module Firmware		Inhibit Security Tools/System	Input Capture	Network Connection Enumeration	Taint Shared Content	Loadable Module	Screen Capture	Exfiltration Over Command and Control Channel	Data Obfuscation	Exploitation of Vulnerability	Firmware
Non-Interactive Service		Man in the Middle	Intercept Multi-Factor Authentication	Network Enumeration	Third-party Software	Modify System Settings	Video Capture	Exfiltration Over Other Network Medium	Exfiltration Over Command and Control Channel	Firmware	Man in the Middle
Rootkit		Masquerading	Modify Account	Network Service Enumeration	Valid Accounts	Non-Interactive Service	Web Service	Exfiltration Over Physical Medium	Fallback Channels	Man in the Middle	Masquerading
Scheduled Task		Memory Residence	Network Sniffing	Network Sniffing	Virtual Terminal Services	Scheduled Task		Scheduled Transfer	Multi-Stage Channels	Masquerading	Modify Control Logic
Valid Accounts		Modify Control Logic	Password Manager	Role Identification		Scripting		Virtual Terminal Services	Multiband Communication	Modify Control Logic	Modify Parameter
Web Shell		Modify Event Log	Private Keys	Serial Connection Enumeration		Third-party Software			Multilayer Encryption	Modify Parameter	Modify Physical Device Display
		Modify Event Log Settings				Virtual Terminal Services			Remote File Copy	Modify Physical Device Display	Modify Reporting Message
		Modify HMI/Historian Reporting				Web Shell			Standard Application Layer Protocol	Modify Reporting Message	Modify Reporting Settings
		Modify Parameter							Standard Cryptographic Protocol	Modify Reporting Settings	Modify Tag
		Modify Physical Device Display							Standard Non-Application Layer Protocol	Modify System Settings	Module Firmware
		Modify Reporting Message							Uncommonly Used Port	Modify Tag	Rootkit
		Modify Reporting Settings							Virtual Terminal Services	Module Firmware	Spoof Command Message
		Modify Security Settings							Web Service	Rootkit	Spoof Reporting Message
		Modify System Settings								Spoof Command Message	
		Modify Tag								Spoof Reporting Message	
		Rootkit									
		Spoof Reporting Message									
		Taint Shared Content									

Otis Alexander. “ICS ATT&CK”. The MITRE Corporation. Industrial Control System Security (ICSS) Workshop. Dec 2017.

CARS

- Used in conjunction with ATT&CK
- ATT&CK describes general monitoring strategy
- CARS provides specific signatures within network packets or log events to look for in order to identify threats
- System logs for Remote Desktop Logon

```
[EventCode] == 4624 and
[AuthenticationPackageName] == 'Negotiate' and
[Severity] == "Information" and
[LogonType] == 10
```

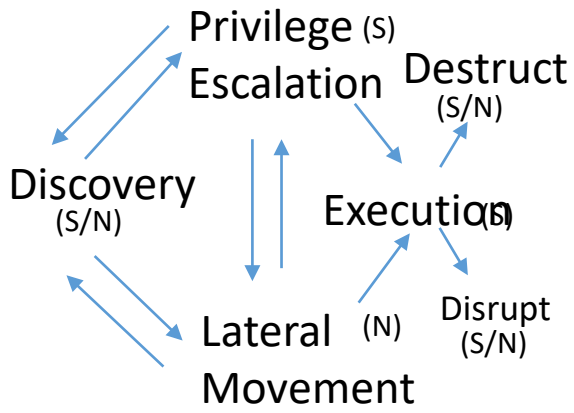
System Model – Attack Techniques

Goal: reduce number of tactics applied for each node…

Tactics

1. Persistence
2. **Privilege Escalation**
3. Defensive and Operator Evasion
4. Credential Access
5. **Discovery**
6. **Lateral Movement**
7. **Execution**
8. Collection
9. Exfiltration
10. Command and Control
11. **Disruption**
12. **Destruction**

FSM of Attack in Phases



Simplified Model

Q: N -> {Discovery, PrivEsc, Execution}

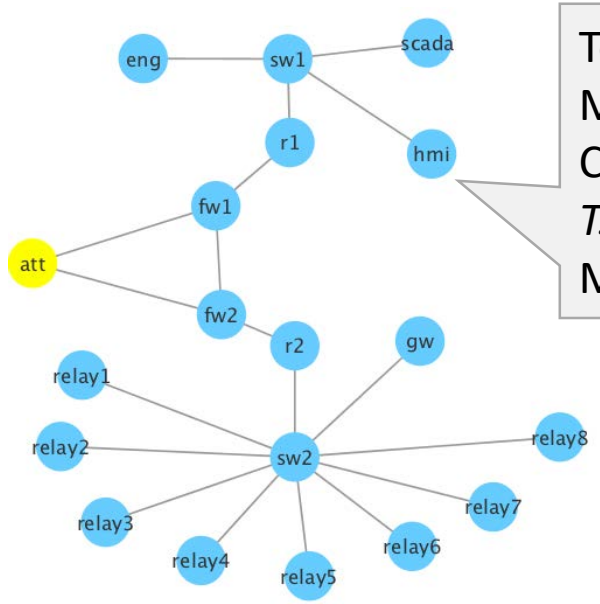
Q: E -> {Discovery, Lateral Movement}a

** Focus on initial exploitation steps*

Calculate Node Scores

Calculate node-based monitoring coverage score

For each node $n \in N$
  For each technique $t \in T$
    if t.os != n.os
      remove t;
    else if t.service $\notin$ n.services
      remove t;
    else if t.CAR $\in$ n.f
      Tscore(t) = Pr(t.CAR \in test sample)
    else
      Tscore(t) = 0
    MonitoredScore(n) = $||tscore||_2$



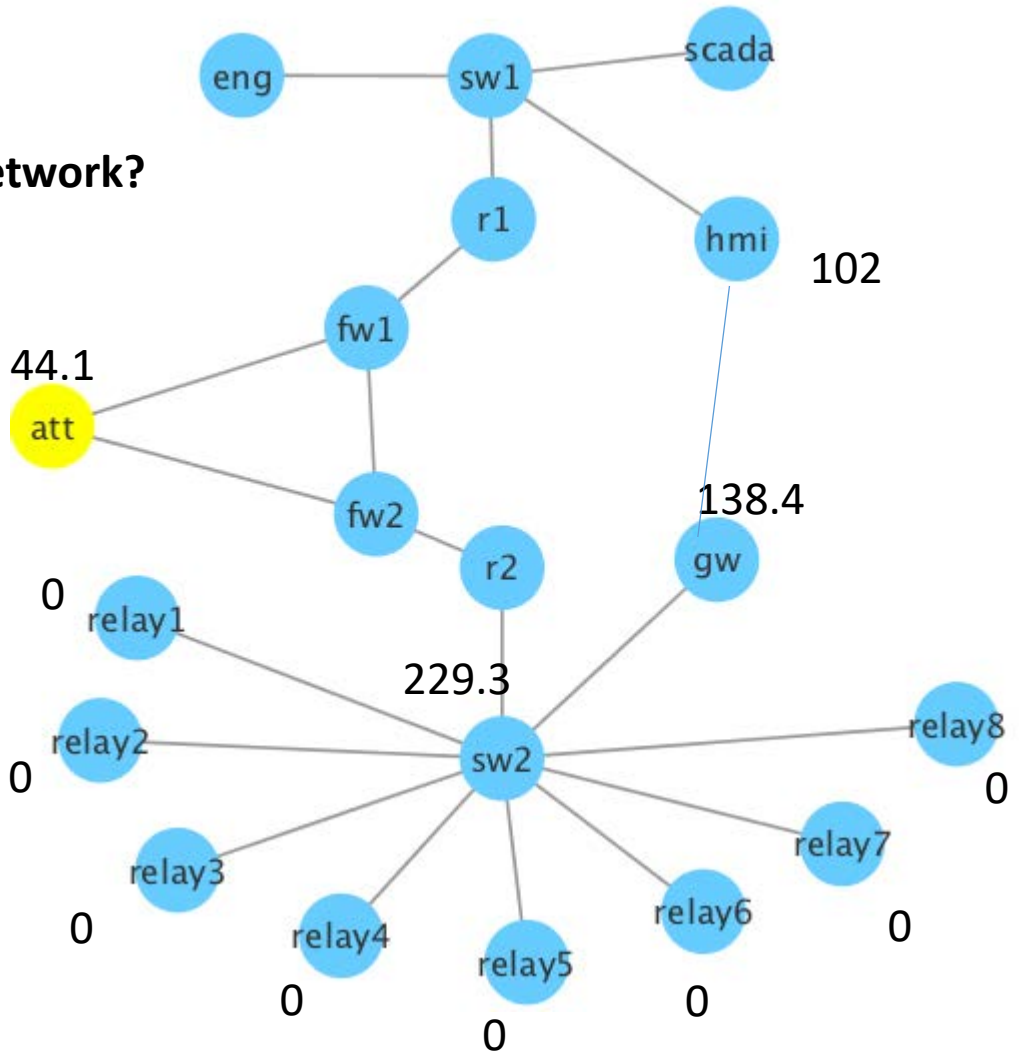
For each node

Techniques, *t*
Monitored features, *f*
Configuration, *c*
Tscore vector = [s(t_1)...s(t_m)]
MonitoredScore = 0

Calculate System Wide Scores

How well monitored is this network?

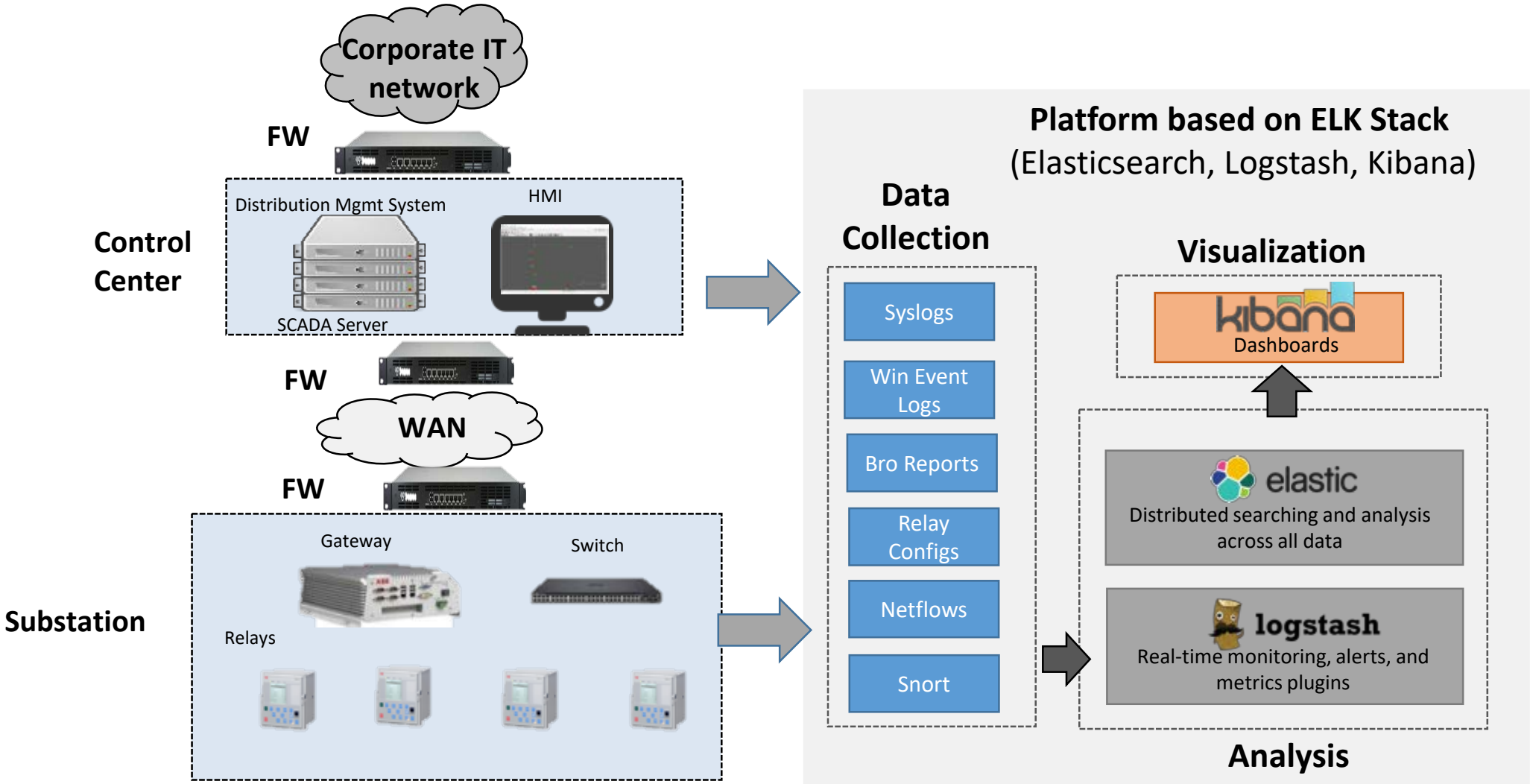
Perform *shortest path* analysis from outside systems to each node



Poorly monitored paths

Well monitored paths

Testbed Evaluation



Data Samples

- One month period
- All the data is from normal day to day usage. No isolated data collection
- 6,199,109 Data points
- 1,534,160 netflow
- 4,657,519 Bro
  - DNS, FTP, HTTP, SMTP, SSL, conn, known_certs, Application
- 7,430 Winlog
  - System
  - Application
  - Security
  - PowerShell

Tables

	Privilege Escalation																			
	Access Token Manipulation	Accessibility Features	AppInit DLLs	Application Shimming	Bypass User Account Control	DLL Search Order Hijacking	Dylib Hijacking	Exploitation of Vulnerability	File System Permissions Weakness	Launch Daemon	New Service	Path Interception	Plist Modification	Scheduled Task	Service Registry Permissions Weakness	Setuid and Setgid	Startup Items	Sudo	Valid Accounts	Web Shell
Attacker	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	44.1
HMI	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	44.1
switch	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0
Gateway	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0
Relays	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0

	Lateral Movement														
	AppleScript	Application Deployment Software	Exploitation of Vulnerability	Logon Scripts	Pass the Hash	Pass the Ticket	Remote Desktop Protocol	Remote File Copy	Remote Services	Replication Through Removable Media	Shared Webroot	Taint Shared Content	Third-party Software	Windows Admin Shares	Windows Remote Management
Attacker	N/A	0	0	N/A	N/A	N/A	0	0	0	N/A	N/A	N/A	0	0	N/A
HMI	N/A	25.15	0	N/A	N/A	N/A	88.5	0	0	N/A	N/A	N/A	0	0	N/A
switch	N/A	0	0	N/A	N/A	N/A	0	0	0	N/A	N/A	N/A	0	0	N/A
Gateway	N/A	138.14	0	N/A	N/A	N/A	0	0	0	N/A	N/A	N/A	0	0	N/A
Relays	N/A	0	0	N/A	N/A	N/A	0	0	0	N/A	N/A	N/A	N/A	0	N/A

- N/A represents attack types we can't currently monitor
- 0 means we can monitor for but it hasn't happened
- Numbers are 1/probability of that event happening in the given time period of 30 days

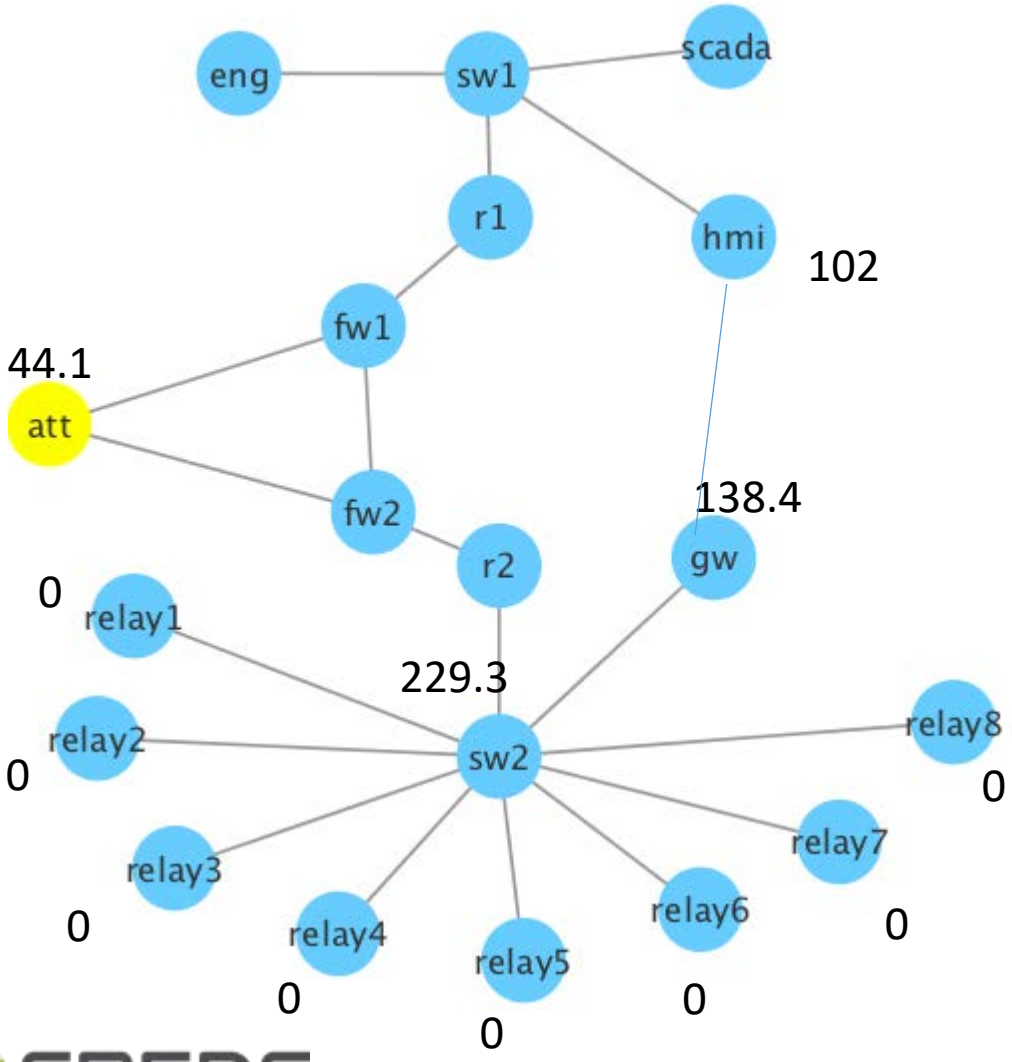
Tables

	Execution																					
	AppleScript	Command-Line Interface	Execution through API	Execution through Module Load	Graphical User Interface	InstallUtil	Launchctl	PowerShell	Process Hollowing	Regsvcs/Regasm	Regsvr32	Rundll32	Scheduled Task	Scripting	Service Execution	Source	Space after Filename	Third-party Software	Trap	Trusted Developer Utilities	Windows Management Instrumentation	Windows Remote Management
Attacker	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	N/A	N/A	0	N/A
HMI	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	N/A	N/A	0	N/A
switch	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	N/A	N/A	0	N/A
Gateway	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	N/A	N/A	0	N/A
Relays	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	N/A	N/A	0	N/A

Normalized probabilities using L2 normalization

- HMI: 102
- Attacker: 44.1
- Switch: 292.3
- Gateway: 138.4
- Relays: 0

Graph Results



Shortest path lengths from attacker:

- To gateway: 284.1
- To HMI: 146.1
- To switch 1: 44.1
- To switch 2: 273.4

- This is not strongly monitored
  - No netflow data from switch 1

- Shortest path for attacker to gateway:
  - 'gw': ['att', 'fw1', 'sw1', 'hmi', 'gw']

Future Works

- Further refine the calculations
- Implement more monitoring features in the ATT&CK list
- Look into automating the graph creation
- Account for what happens when a node goes down
- Account for what happens when an attacker has breached the network

Thanks

armin.rahimi@wsu.edu

https://github.com/wsu-smartcity