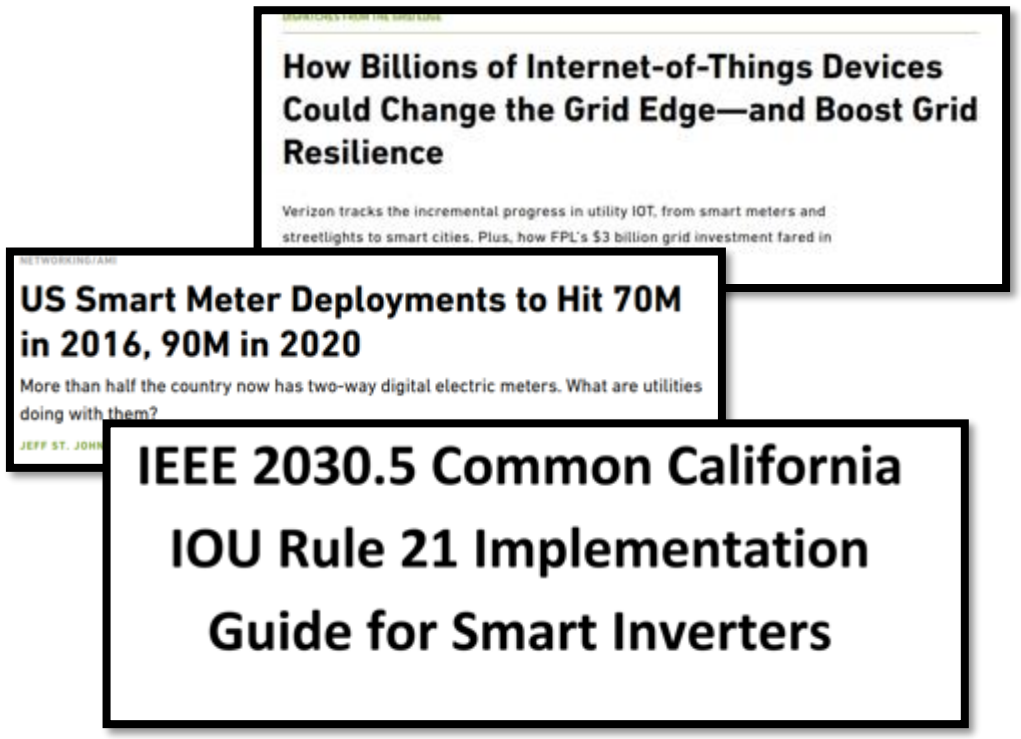
Securing Your ICS Software with the AttackSurface Host Analyzer (AHA)

Adam Hahn
Ali Tamimi
Dave Anderson

Washington State University

Expanding Attack Surface

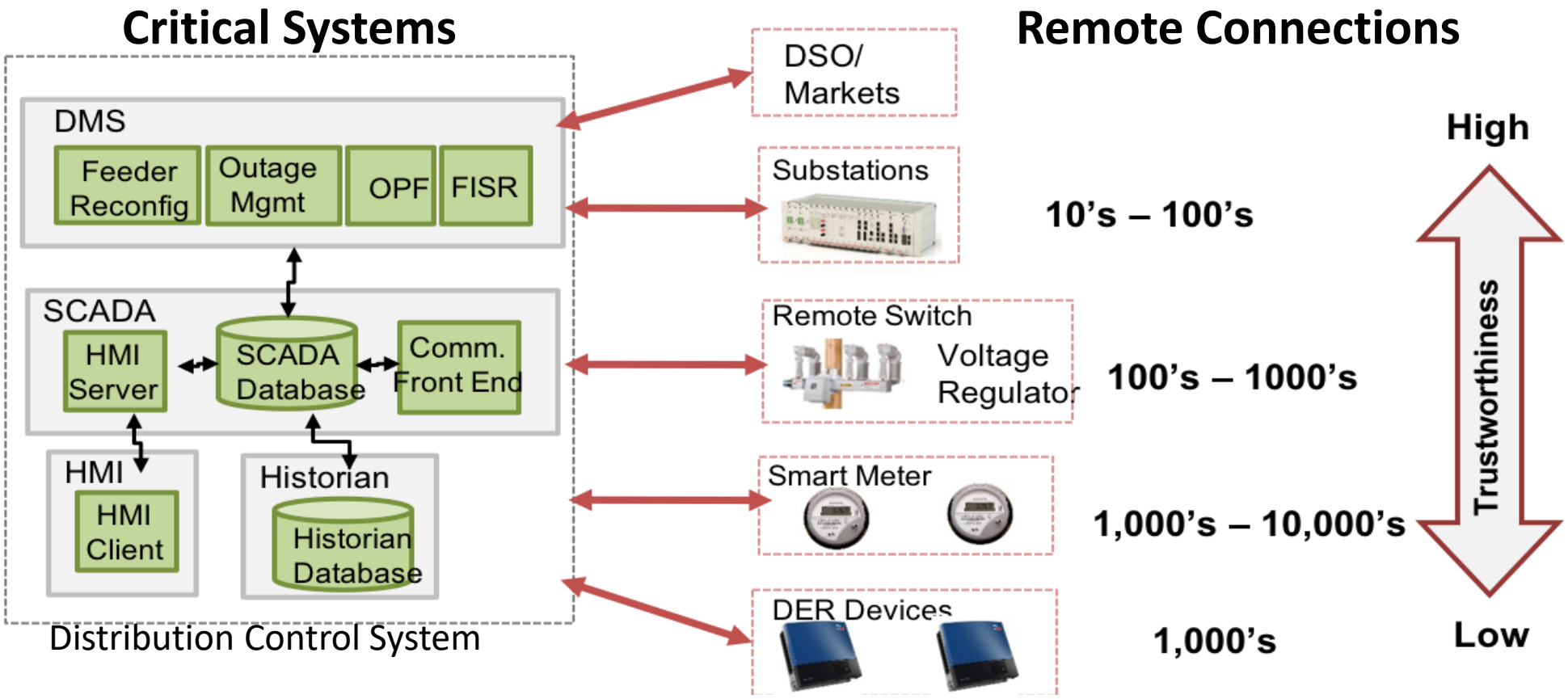
Power Grid



Industrial Control Systems

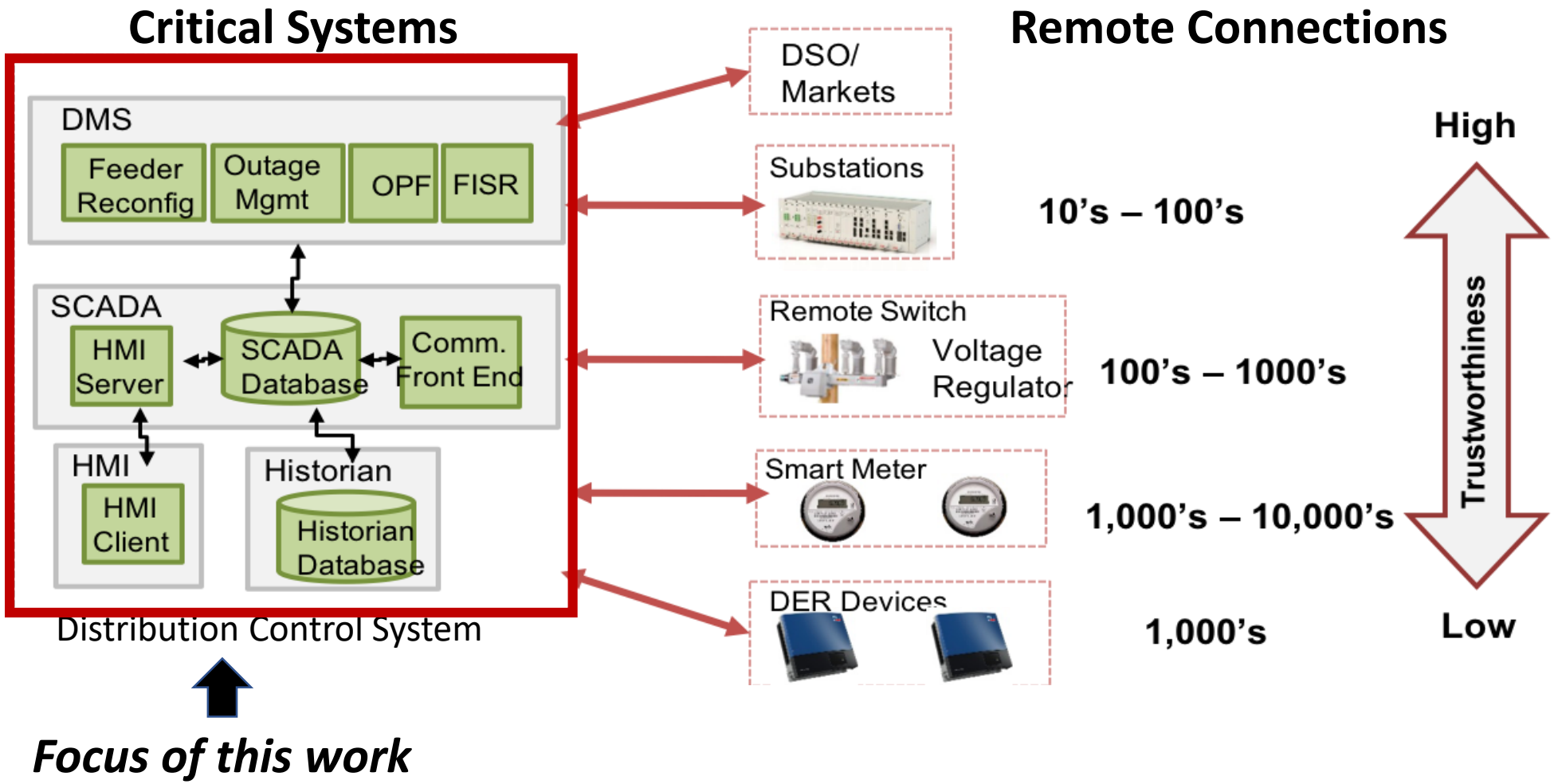
https://www.dhs.gov/sites/default/files/publications/Strategic_Principles_for_Securing_the_Internet_of_Things-2016-1115-FINAL....pdf
https://www.greentechmedia.com/squared/read/how-billions-of-iot-enabled-devices-could-change-the-grid-edge#gs.ZAwnnc4
https://www.pge.com/includes/docs/pdfs/shared/customerservice/nonpgeutility/electrictransmission/handbook/rule21-implementation-guide.pdf
https://www.networkworld.com/article/3243928/internet-of-things/what-is-the-industrial-iot-and-why-the-stakes-are-so-high.html

Grid Attack Surface

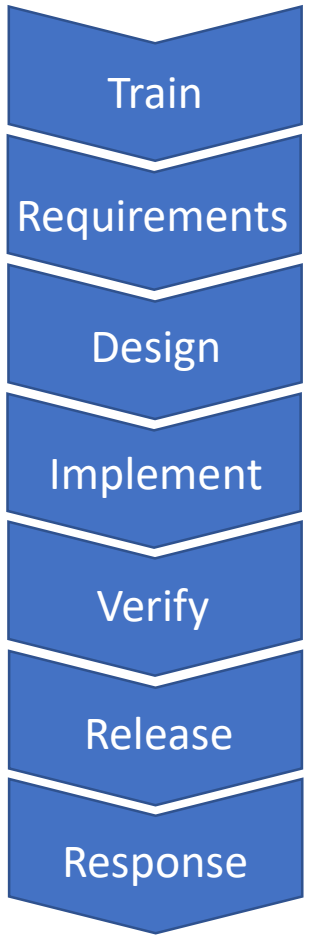


Growing concern for remote attack/exploitation of critical systems!

Grid Attack Surface



Tools in Security Development Lifecycle



Phases | **Tools**

- Establish Design Requirements → DHS CSET
- Attack Surface Analysis Reduction → ???
- Use Threat Modeling → MS Threat Modeling,

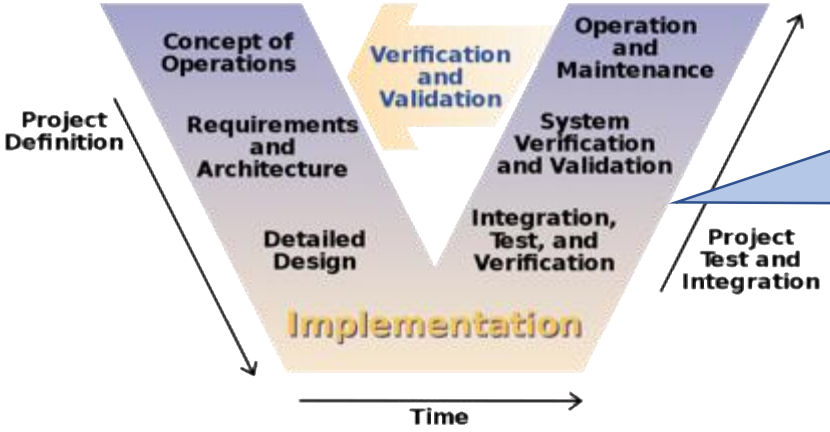
- Use Approved Tools → GCC, LLVM, Clang
- Deprecate Unsafe Functions → Coverify, Fortify, Lint,
- Perform State Analysis → Findbugs, DHS SWAMP

- Dynamic Analysis → AppScan, Purify, Valgrind
- Fuzz Testing → Spike, American Fuzzy Lop, …
- Attack Surface Review → ???

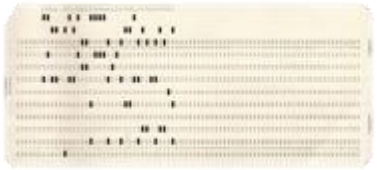
Need metrics and tools to analyze attacks surface

[1] https://www.microsoft.com/en-us/sdl/

Challenges (Developers)



External:



Legacy Code

SW:

Risk???

Trends:

- Lines of Code
- Third Party Dependencies
- OS Security Features

Operating System:

Security Mechanisms

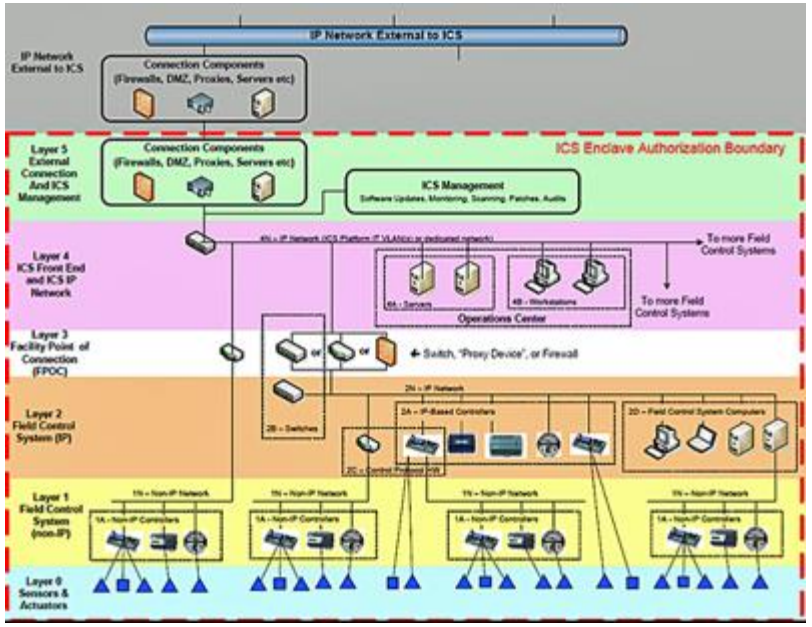
- **ASLR/DEP/CFI**
- **Privileges/Code Sign**
- **Containers/Virtualization**

https://en.wikipedia.org/wiki/V-Model_(software_development)

Challenges (ICS Operator)

Unknown SW

???



Critical ICS

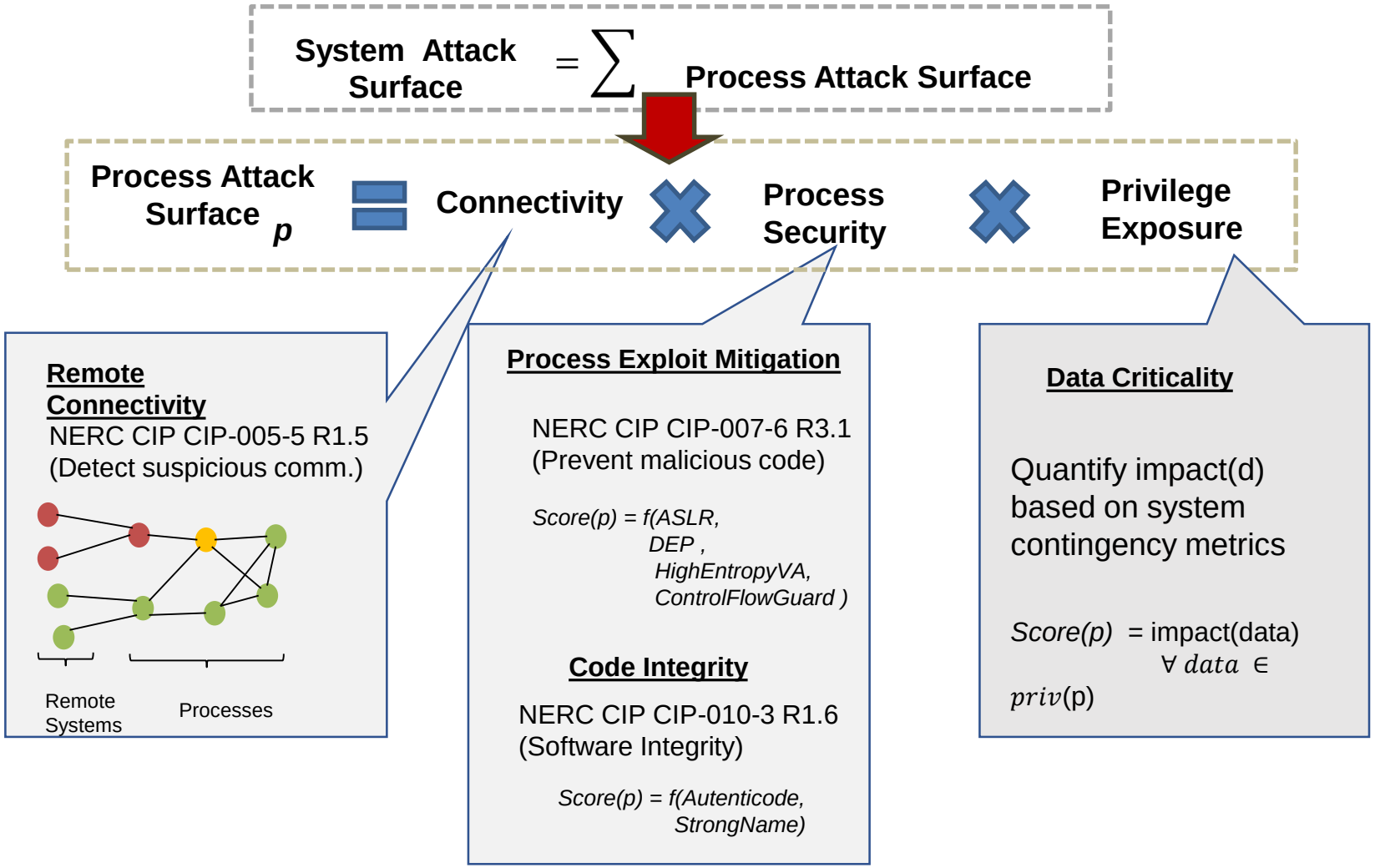
- Does it implement principle of least privilege?
- Does it implement modern exploit mitigations?
- How secure are remotely exposed processes?
- Is all the code properly signed?

How to ensure critical assets are adequately protected???

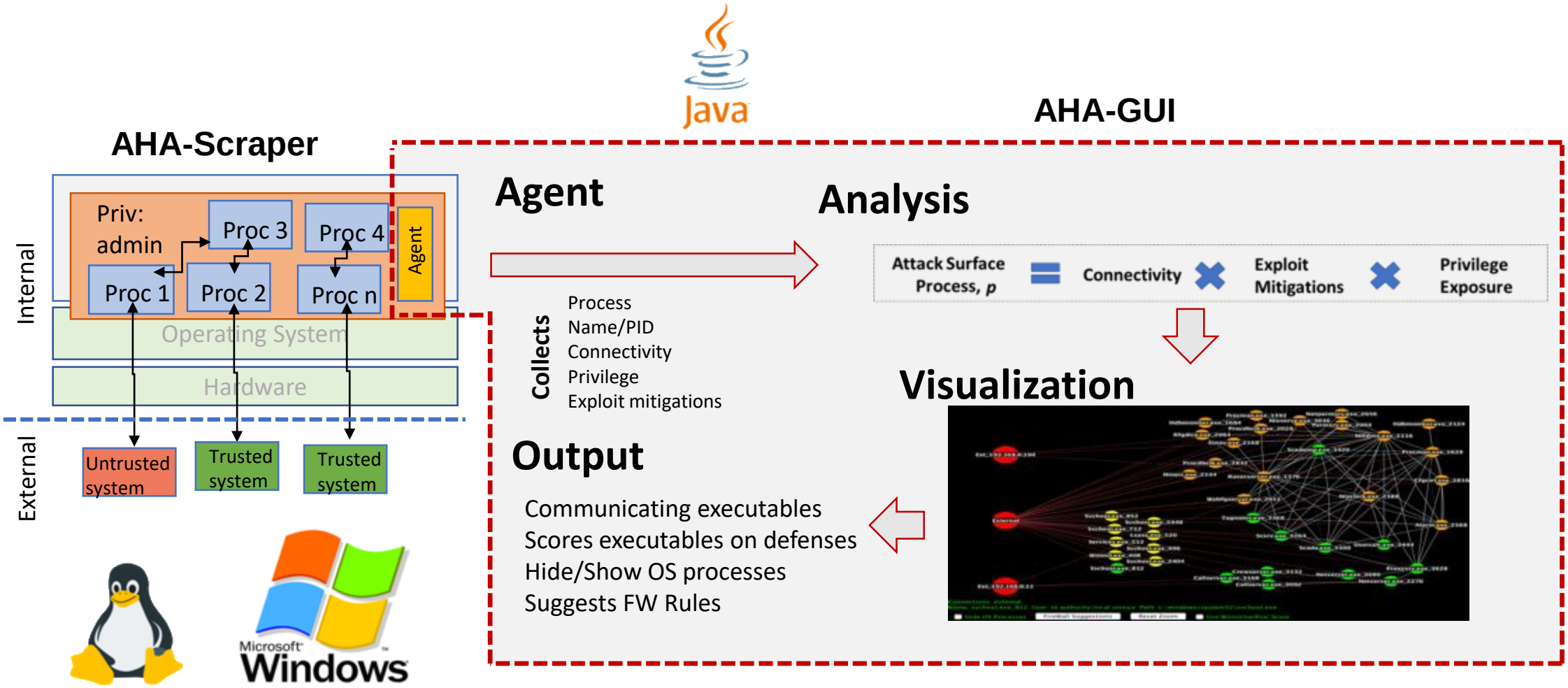
AHA (Attack Surface Host Analyzer)

- Analyze attack surface of critical ICS software platforms
- Provides graphical display of vulnerable processes and connections

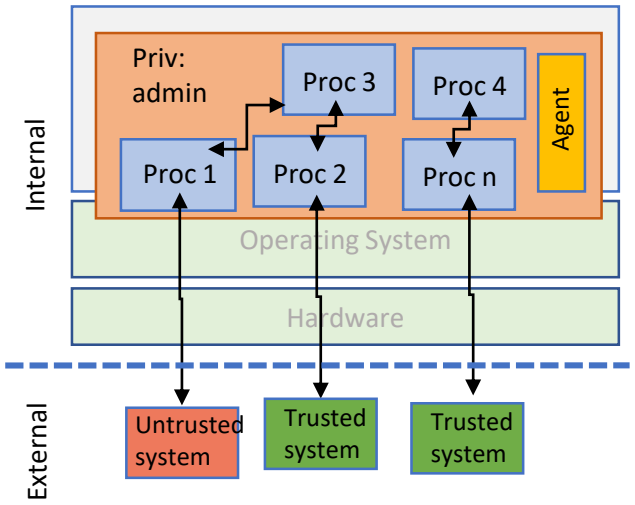
Attack Surface Metric



AHA (**A**ttack Surface **H**ost **A**nalyzer)



Agent and Analysis



Component	Windows Agent	Linux Agent
Platform	PowerShell, Currports, Get-PESecurity	Bash
Proc Name/Id	Currports	PS/Netstat
Connection	Currports	Netstat
ASLR	PE Header	Kernel/Proc
CFG	PE Header	LLVM CFI
Authenticode	PE Header	NA
SafeSEH	PE Header	NA
RELRO	NA	ELF

Collects

Process
Name/PID
Connectivity
Privilege
Exploit mitigations

Scoring Factors

Mitigation	Score
Address Space Layout Randomization (ASLR)	10
Data Execution Prevention (DEP)	10
Code Signing (authenticode)	10
Strongnaming	10
SafeSEH	10
Arch	10
ControlFlowGuard	30
HighentropyVA	10

Privileges

Process privileges (e.g., Local Service)

Mitigation	Score
Local service	10
System	-50

Final Score

(Low)
|
(High)

System-Level Metric

- The system score expands on the process-level score to incorporate the connectivity of the process to both external and internal processes

(1) Normalize the Score

(2) Reversed The Score (1- Normalized Score) => $large\ score \equiv less\ secure$

(3) Definition: {
Parents Process: external processes that are connected to an internal process are its parents
Siblings Process: internal processes that are connected to an internal process are its siblings
Parent score: The System-Level Metric of the parents (using for calculating the System-Level Metric of the process)
Sibling Score: A score that a process provides for its siblings (using for calculating the System-Level Metric of siblings process)

(4) Calculate System-Level Metric of process based on its Parent score and siblings scores of its siblings

Process$_i$		**Scores**		
$Parents(i)$	$Siblings(i)$	Parent Score ($P_{Score}(i)$)	Sibling Score ($S_{Score}(i)$)	**System Score** ($sys_{Score}(i)$)
Null	Null	-	-	$N_{Score}(i)$
Null	not Null	-	$N_{Score}(i)$	$N_{Score}(i) * \sum_{j \in Siblings(i)} S_{Score}(j)$
not Null	Null	$N_{Score}(i) * \sum_{j \in Parents(i)} sys_{Score}(j)$	$P_{Score}(i)$	$P_{Score}(i)$
not Null	not Null	$N_{Score}(i) * \sum_{j \in Parents(i)} sys_{Score}(j)$	$P_{Score}(i)$	$P_{Score}(i) + N_{Score}(i) * \sum_{j \in Siblings(i)} S_{Score}(j)$

Harmonic mean

- provides a stronger emphasis on the lowest valued process in the system and therefore will provide a lower value if any processes provide very low score
- the external harmonic mean demonstrates the processes immediate exposure to remote compromises
- internal harmonic mean represents the overall level of protection within the platform, but which may not be directly vulnerable to remote compromise.

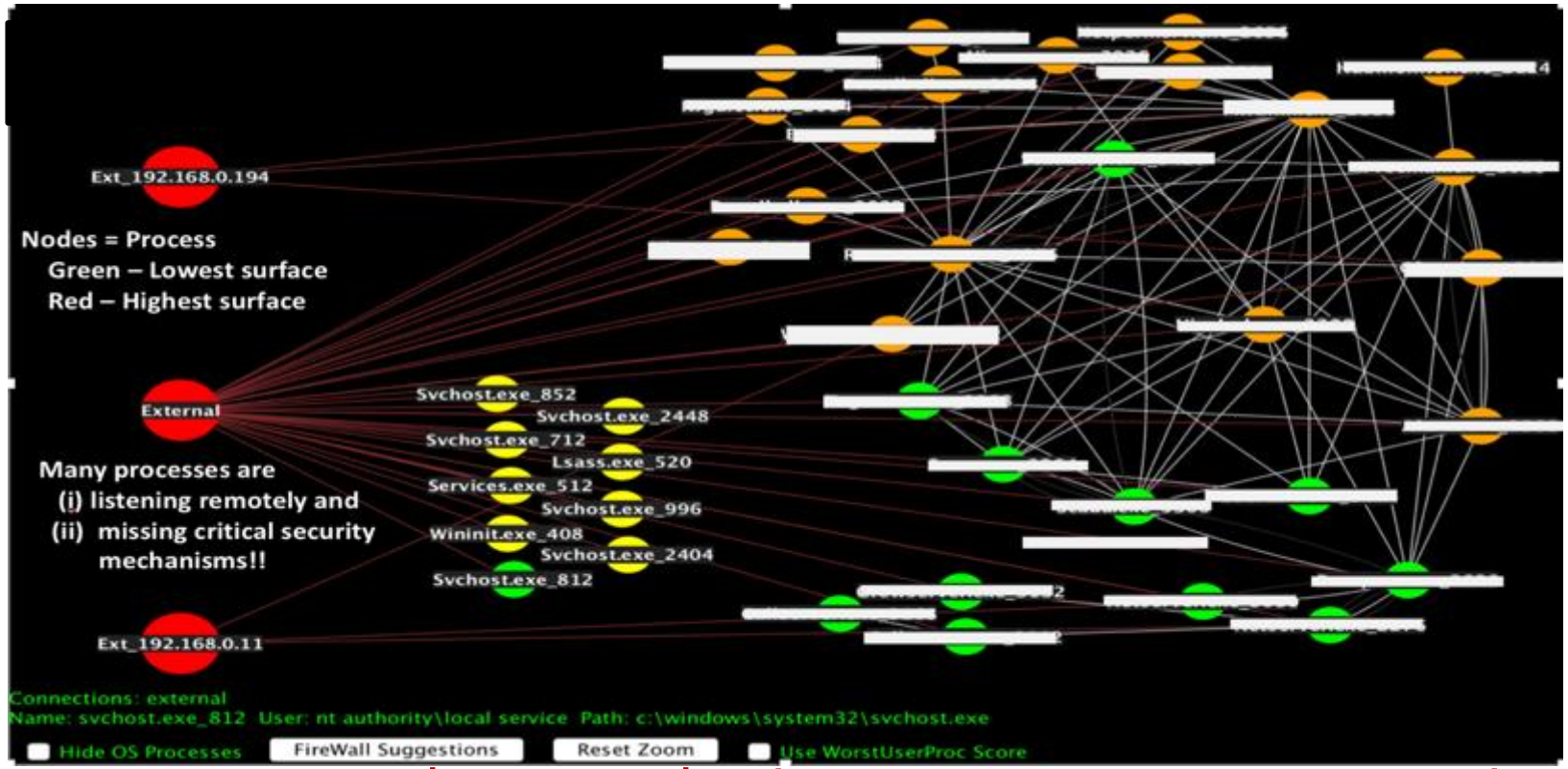
AHA Visualization

Processes/ External Systems

Low Risk

High Risk

Connection

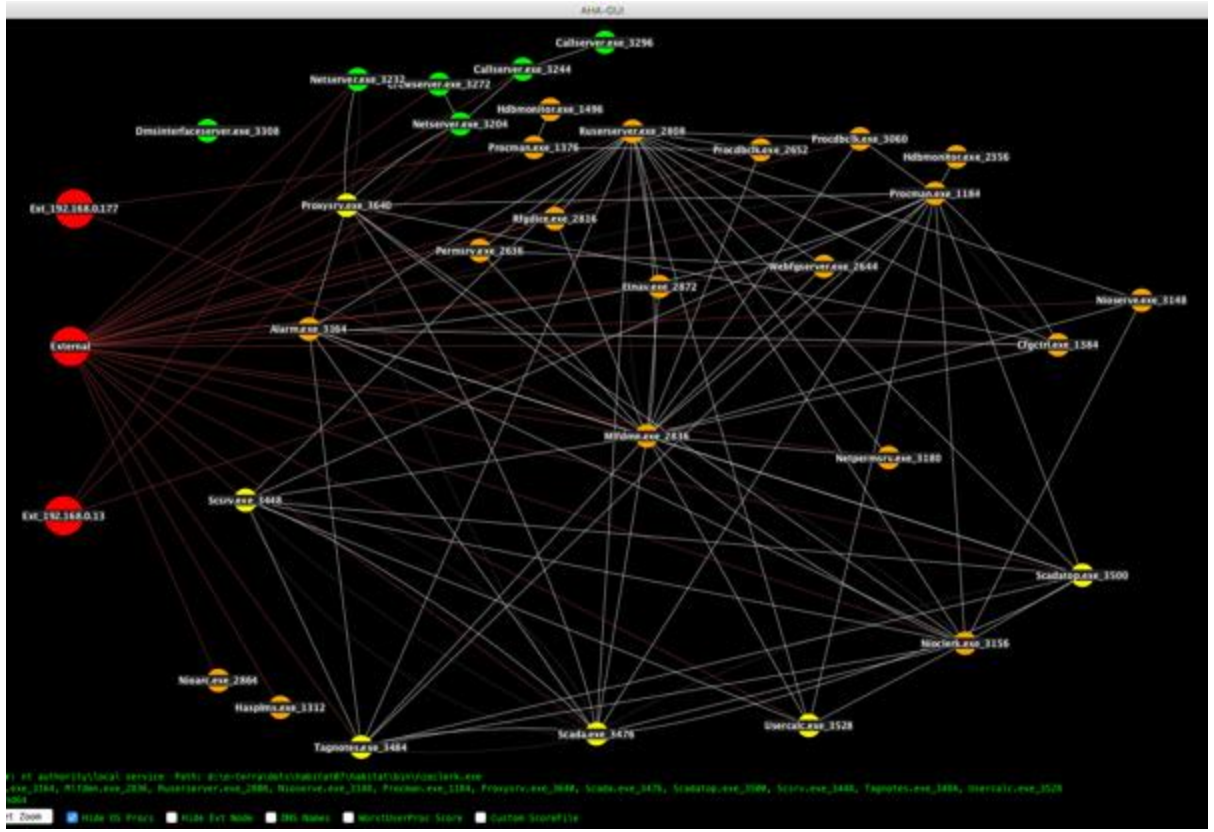


External Systems

OS Processes

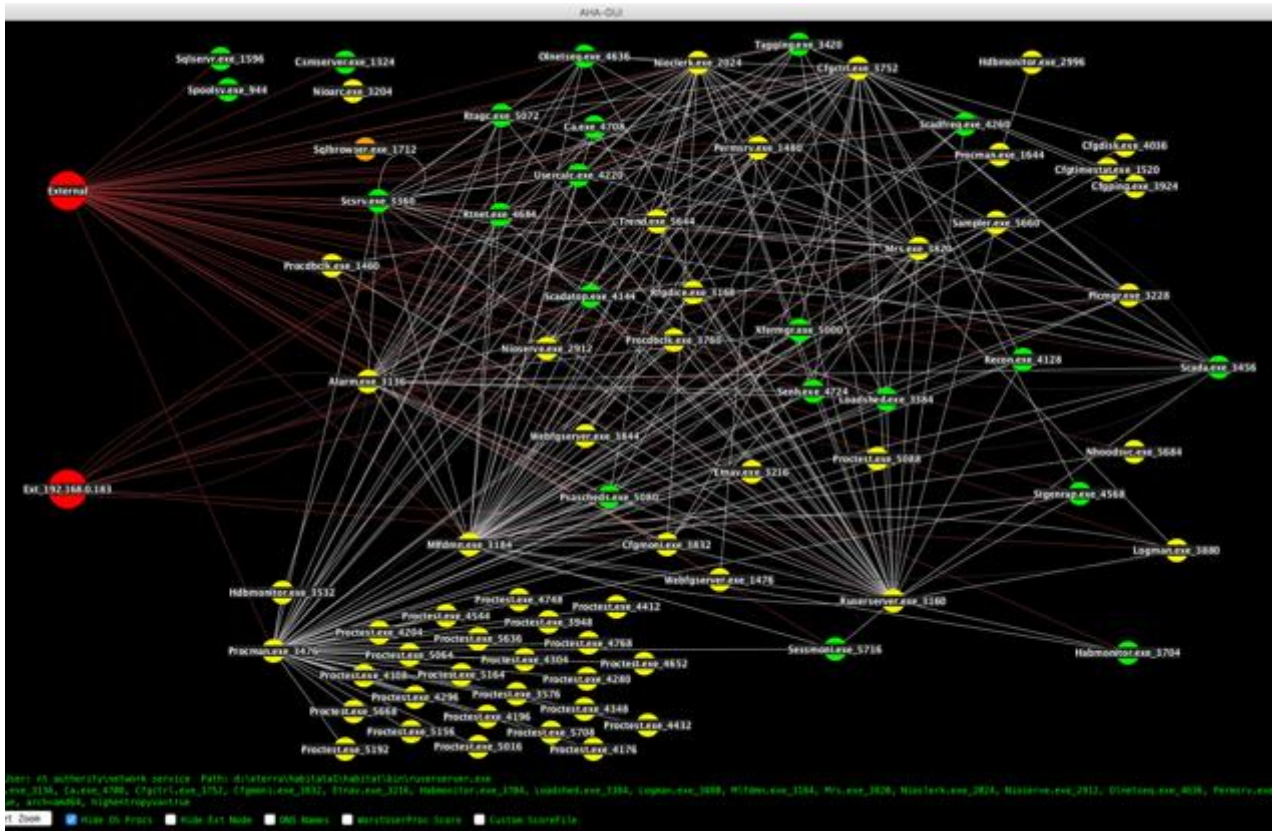
Application Processes

Case Studies (1)



Platform: Control Center Server 1*
OS: Windows 2008, R2

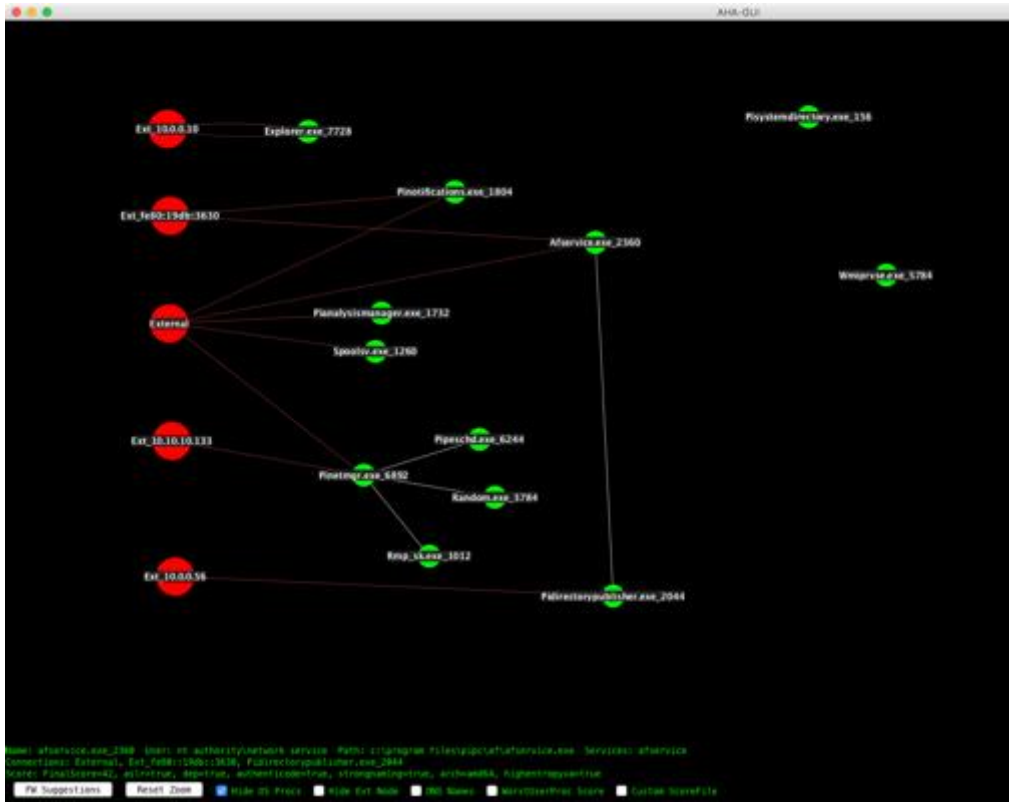
vs



Platform: Control Center Server 2*
OS: Windows 2012, R2

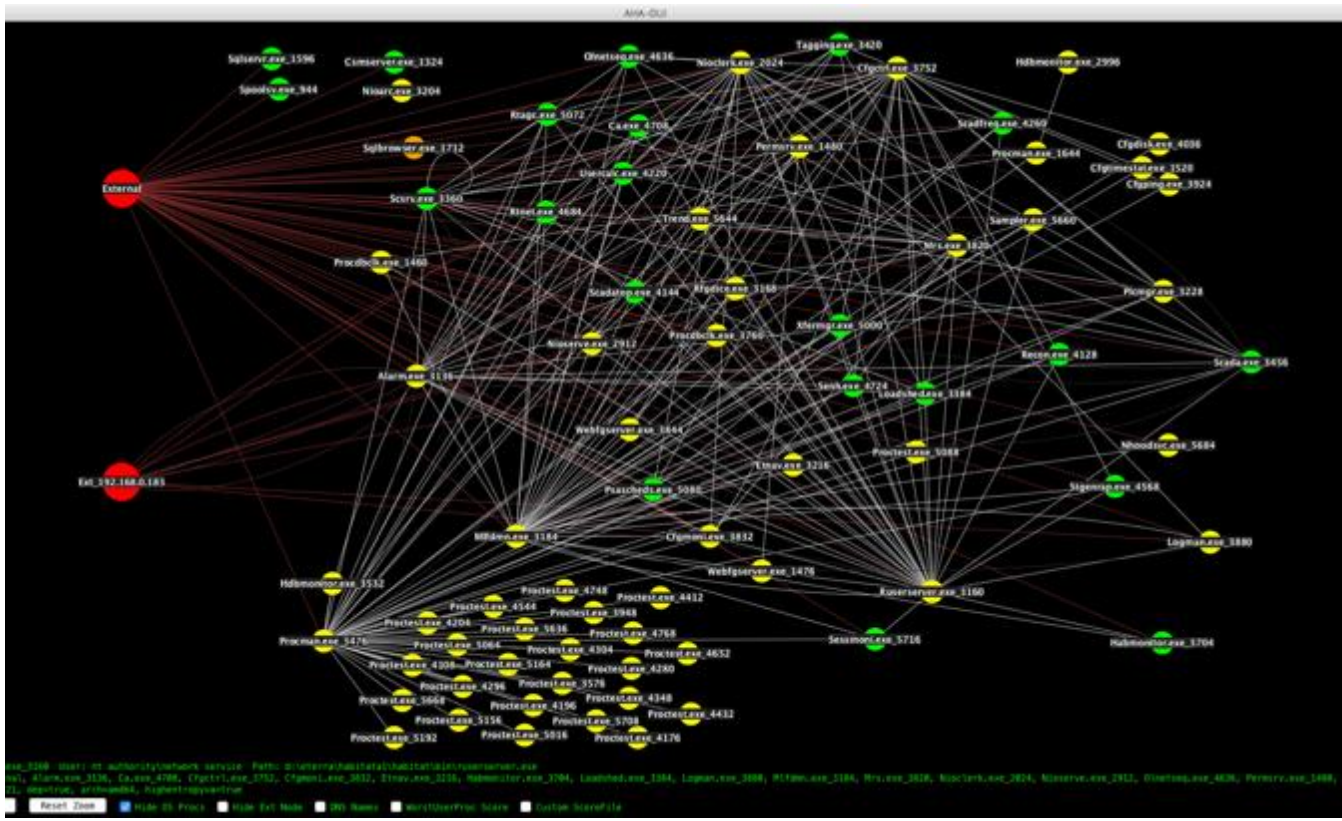
**Actual sw product anonymized…*

Case Studies (2)



Platform: OSIsoft PI Historian
OS: Windows 2012, R2 (**Core**)

vs



Platform: Control Center Server 2*
OS: Windows 2012, R2

**Actual sw product anonymized…*

Case Studies (3)

Tool evaluated on 10+ different industry software platforms across multiple vendors

- **Locations:** WSU/PNNL/ISU/CFU/OSIsoft
- **Platforms**: EMS/DMS, FEPs, Historians, Substation Gateways,
- **Vendors**: GE, ABB, OSIsoft, Siemens

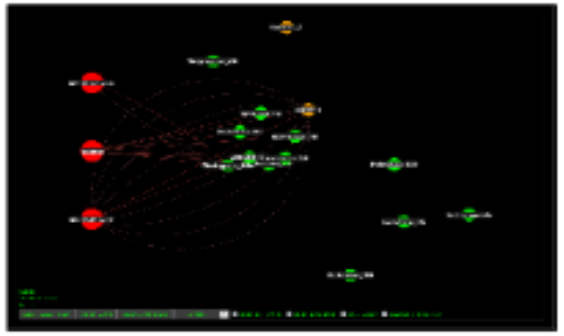


Figure 4: Historian Platform A

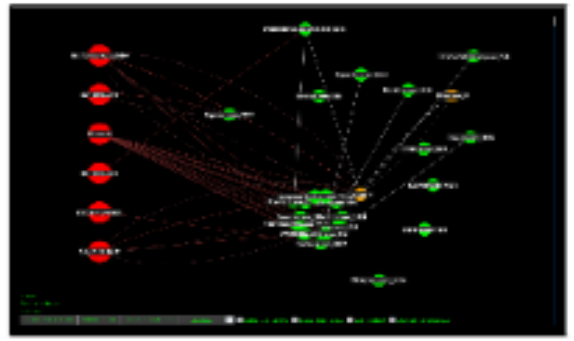


Figure 5: Historian Platform B

Figure 7: Control Center Platform B

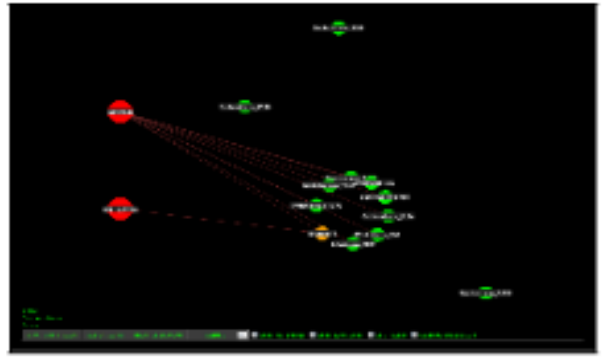


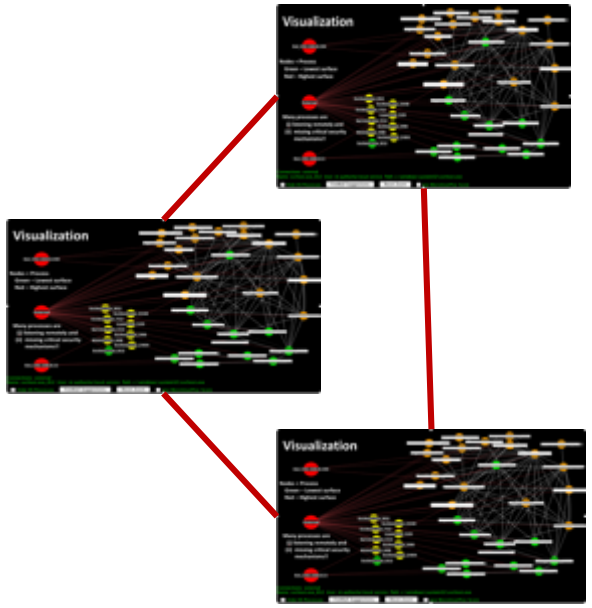
Figure 6: Control Center Platform C

Platform	# Processes	Harmonic Mean of scores		Min R_{score}	Max R_{score}
		Externally accessible	Internally accessible		
Control Center Platform A (Windows Server 2016)	12	38.53	74.78	0.068	1.859
Control Center Platform B (Windows server 2008R2)	43	9.53	8.22	0.177	6.690
Control Center Platform C(Windows Server 2016)	38	29.44	55.55	0.034	3.630
Historian Platform A	14	80	80	0.034	1.859
Historian Platform B	25	70.94	62.22	0.017	2.988

Figure 8: Control Center Platform A

Future Work

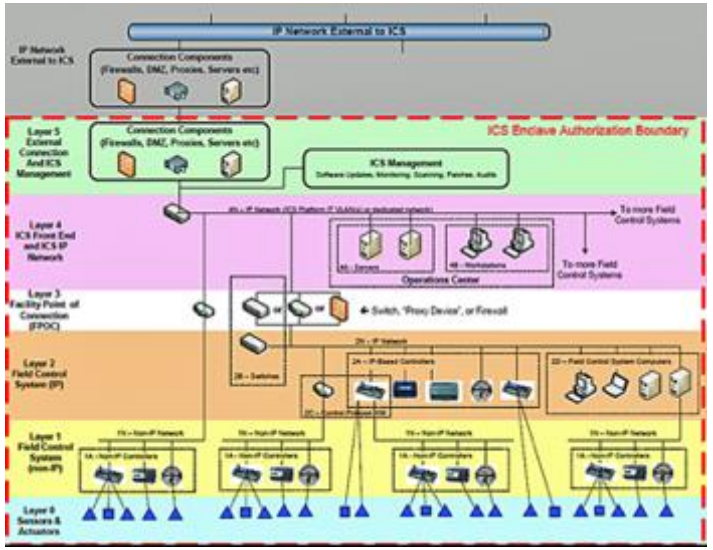
Composability of multiple system



Expanded metrics and analysis

```
for (Node node:graph)
{
    String nodeClass= node.getAttribute("ui.class");
    sScore = 0;
    procScore = 0;

    System.out.println("Node: " + node );
    if(!nodeClass.equalsIgnoreCase("external")){
        sScore = Integer.parseInt(node.getAttribute("score"));
        if(sScore ==0.0){ sScore = .1; }

        Dijkstra dijkstra = new Dijkstra(Dijkstra.Element.NODE, null, null);
        dijkstra.init(graph);
        dijkstra.setSource( node );
        dijkstra.compute();

        for (Node next: graph){
            nodeClass= next.getAttribute("ui.class");
            if(nodeClass.equalsIgnoreCase("external")){
                //System.out.println("  Nodes: " + node + "  " + next);
                try{
                    pathLen = dijkstra.getPathLength(next);
                } catch (Exception e) { pathLen=1; }

                if(Double.isInfinite(pathLen)){ pathLen = 10000; }

                for (Node p : dijkstra.getPathNodes(next)){
                    nodeClass= p.getAttribute("ui.class");
                    if(!nodeClass.equalsIgnoreCase("external") && p!= node){
                        pScore = Integer.parseInt(p.getAttribute("score"));
                        sScore = sScore + pScore;
                    }
                }
                double tmpLog = -1 * Math.log(pathLen/sScore);
                if(tmpLog < 0){tmpLog = 0.1;}
                procScore = procScore + tmpLog;
            }
        }
        dijkstra.clear();
    }
    System.out.println("    procScore: " + procScore);
    totalScore = totalScore + procScore;
```

Case studies and usability



Review

- Growing SW complexity and attack surface
- New tools necessary to evaluate attack surface
- Novel attack surface metrics and evaluation techniques
- Looking for industry collaboration and feedback

thank you.

contact: ali.tamimi@wsu.edu

code: http://aha-project.github.io

This material is based upon work supported by the Department of Energy under Award Number DE-OE0000830. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.