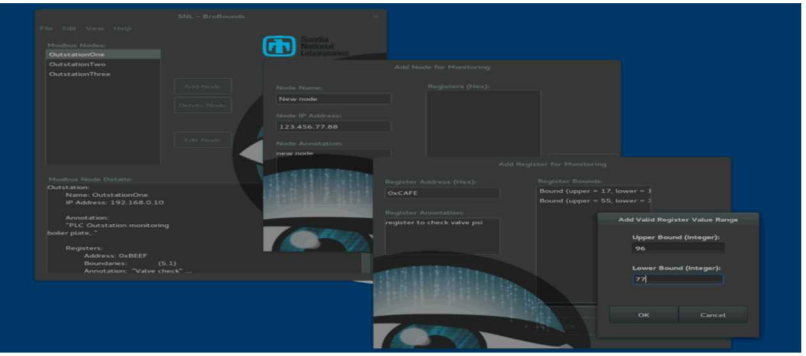
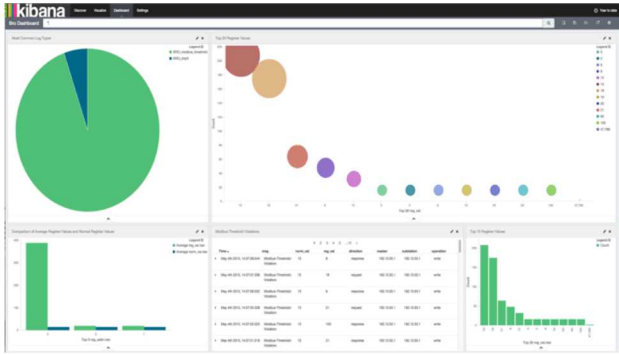
SAND2016-7541PE





ICS-CERT Modbus Anomaly Detection

Mike King, Susan Wade

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Overview

- Original Tasking
  - Why it is hard
  - Why it is possible
- Implementation
- Demonstration
- New Approach
  - Why?
  - How is it better?
  - Short term benefits

Original Tasking

- “Detect anomalies in ICS traffic”
  - Define what malicious ICS traffic means
  - Create a tool detect it

Why Is This Hard?

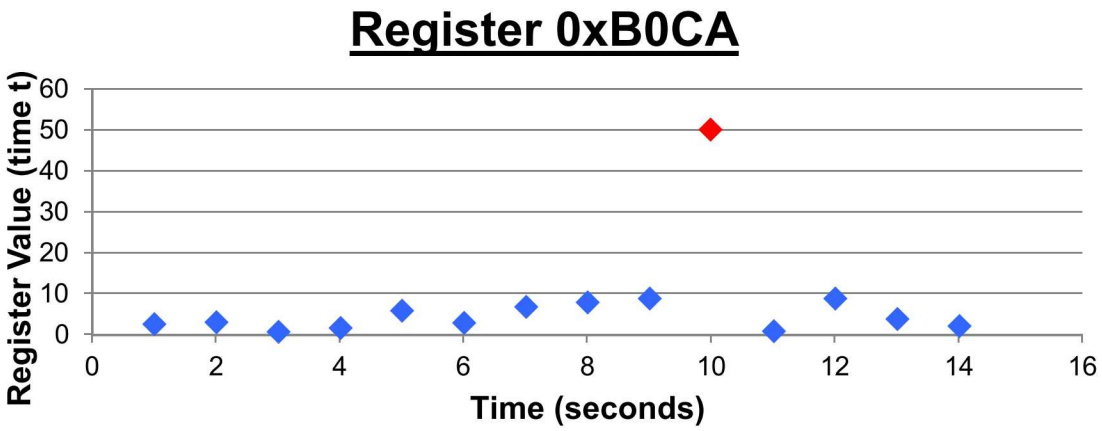
- "Anomaly" is a vague concept
- Anomalies are well-understood in an enterprise network, but the ICS network is different
  - In ICS: Anomaly != Malformed packet
  - In ICS, Anomaly != SQL injection
    - We're not looking for attacks on a browser
- Requires knowledge of ICS Protocols
- Requires some baseline, "non-anomalous" traffic
  - Unique to particular facilities

Why Is This Possible?

- ICS traffic is very regular (deterministic)
- ICS network has a very specific goal (process control)
  - The enterprise network has to be “everything to everyone”
- The Bro network tool has a Modbus protocol analyzer
  - We don’t have to invent one

Implementation – BroBounds

- We chose bad process input values as our anomalies
- How do we do it?
  - Collect every register read and write from a capture of ICS network traffic
  - From that collection of values, create a "default" Bro script (in memory) that alerts on a register value exceeding the bounds from that capture
  - Highlight statistical outliers to the user for modifying the "default" bounds
  - Provide a GUI for modifying that default script (instead of coding in BroScript)
  - Visualize these alerts in a meaningful way

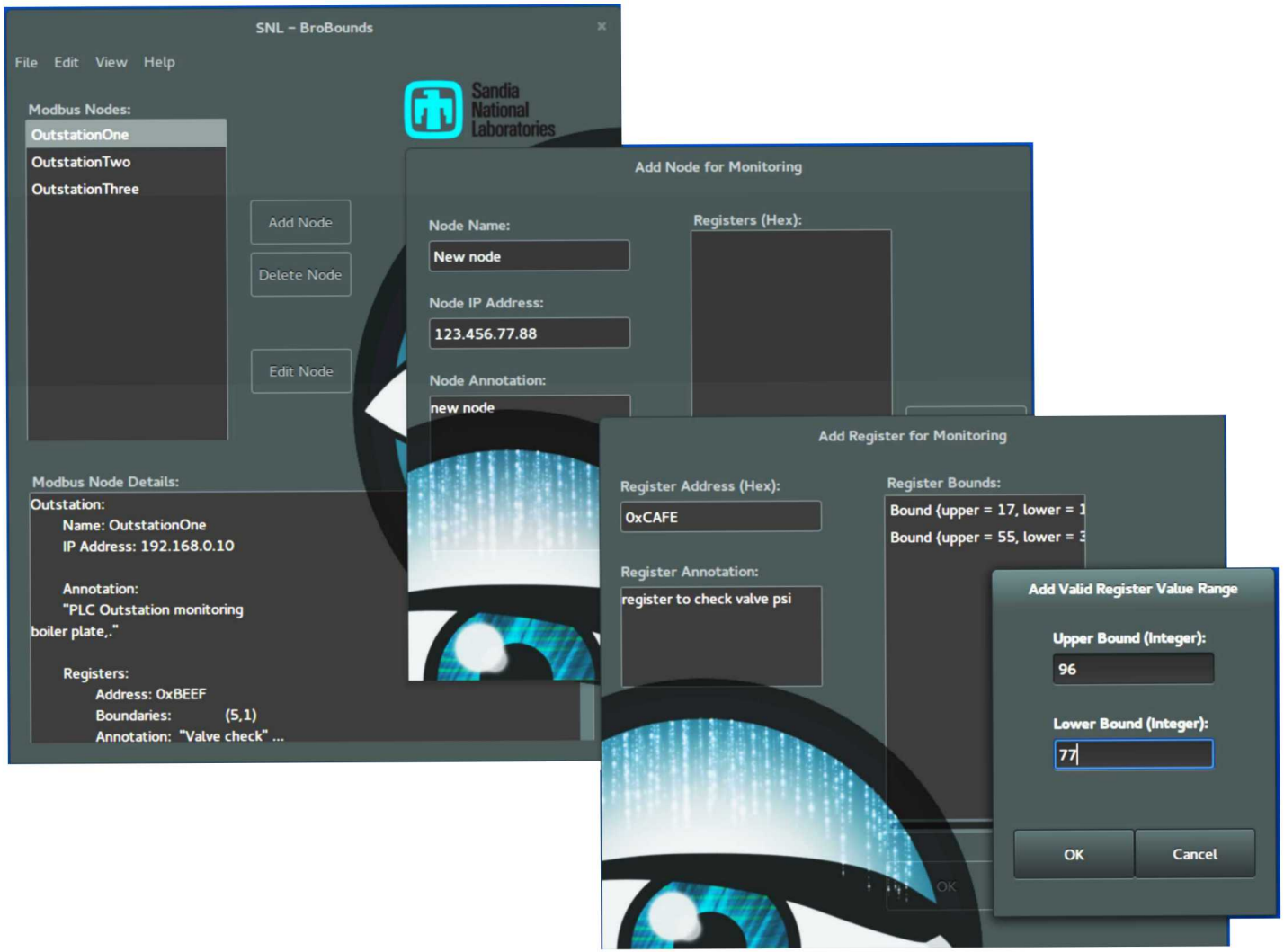


Bro Scripting

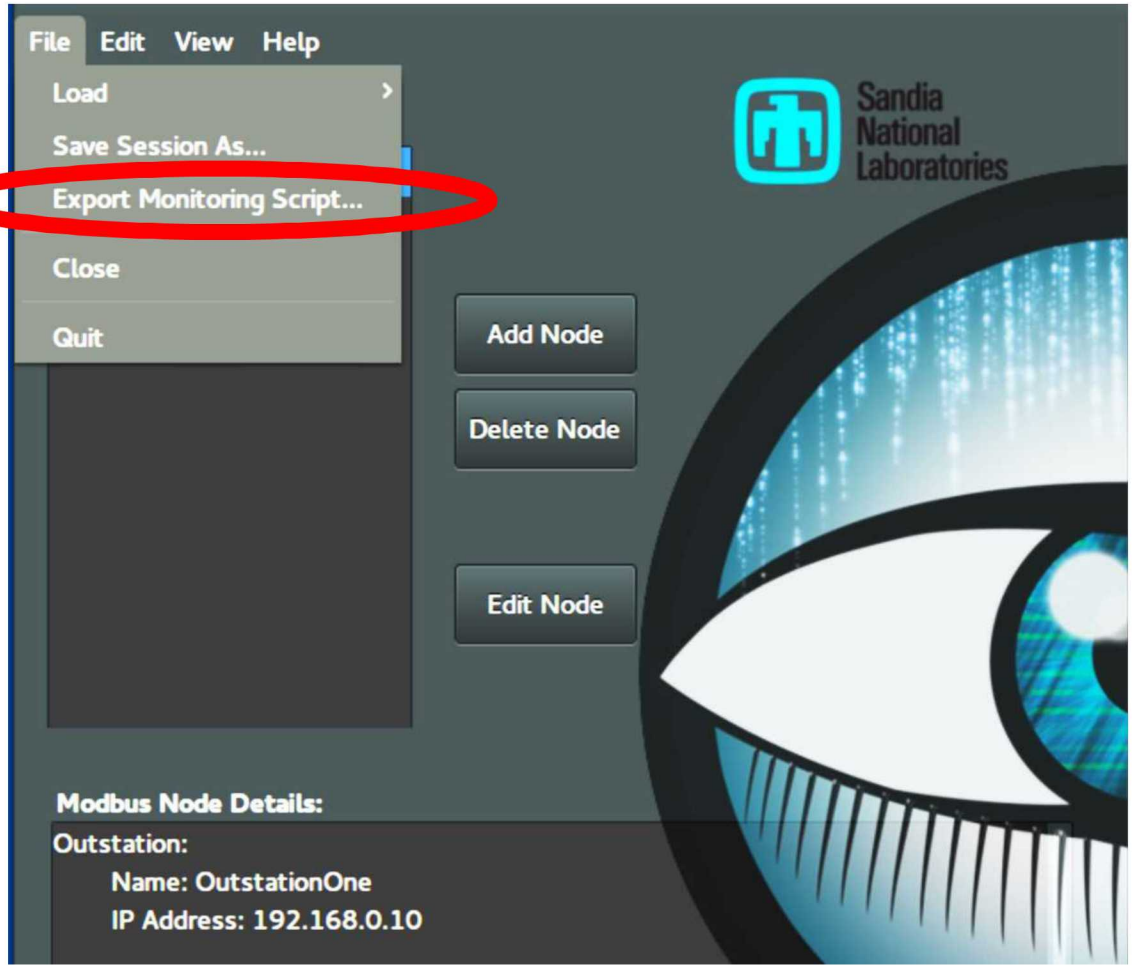
- Turing-complete, domain-specific language
- ICS security should not depend on ICS engineers learning this language
- BroBounds abstracts the details away

```
event modbus_write_single_register_request
        (c: connection, headers: ModbusHeaders, address: count, value: count) {
        local master = c$id$orig_h;
        local outstation = c$id$resp_h;
        if ( master in DiscoveredNodes ) {
            local node = DiscoveredNodes[master];
            if ( address !in DiscoveredNodes[master]$registers ) {
                DiscoveredNodes[master]$registers[address] = vector();
                local new_reg = DiscoveredNodes[master]$registers[address];

                #new_reg = vector();
                new_reg[|new_reg|] = value;
            } else {
                local reg = DiscoveredNodes[master]$registers[address];
                reg[|reg|] = value;
            }
        } else {
          local new_modbus_node: ModbusNode = [$ip_address=master, $registers=table([
          address]=vector(value))];
          DiscoveredNodes[master] = new_modbus_node;
        }
}

event modbus_write_single_register_response
        (c: connection, headers: ModbusHeaders, address: count, value: count) {
        local master = c$id$orig_h;
        local outstation = c$id$resp_h;
        if ( outstation in DiscoveredNodes ) {
            local node = DiscoveredNodes[outstation];
            if ( address !in DiscoveredNodes[outstation]$registers ) {
                local new_reg = DiscoveredNodes[outstation]$registers[address];
                #new_reg = vector();
                new_reg[|new_reg|] = value;
            } else {
                local reg = DiscoveredNodes[outstation]$registers[address];
                reg[|reg|] = value;
            }
        } else {
            local new_modbus_node: ModbusNode = [$ip_address=outstation,
            $registers=table([address]=vector(value))];
            DiscoveredNodes[outstation] = new_modbus_node;
        }
```

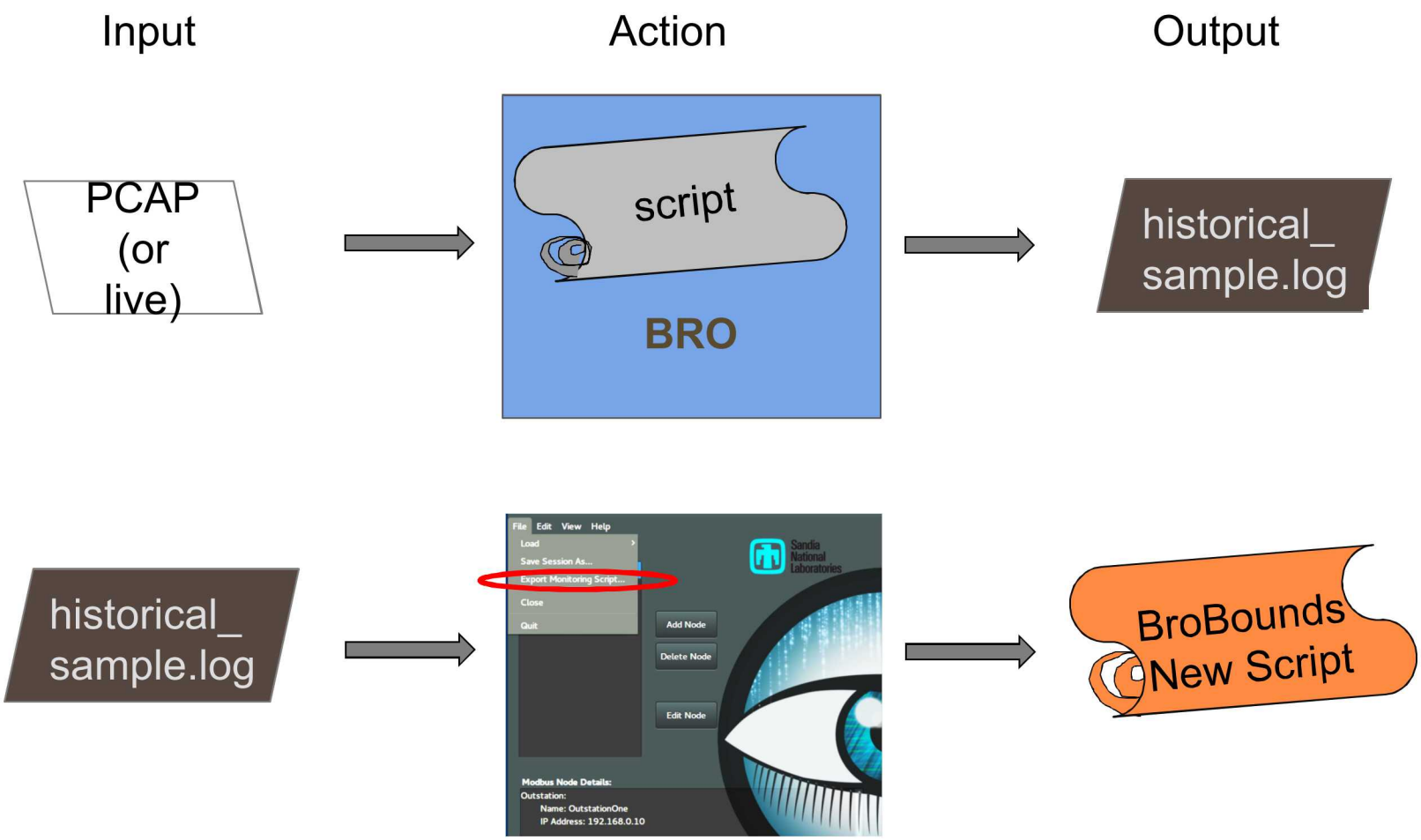
BroBounds Application



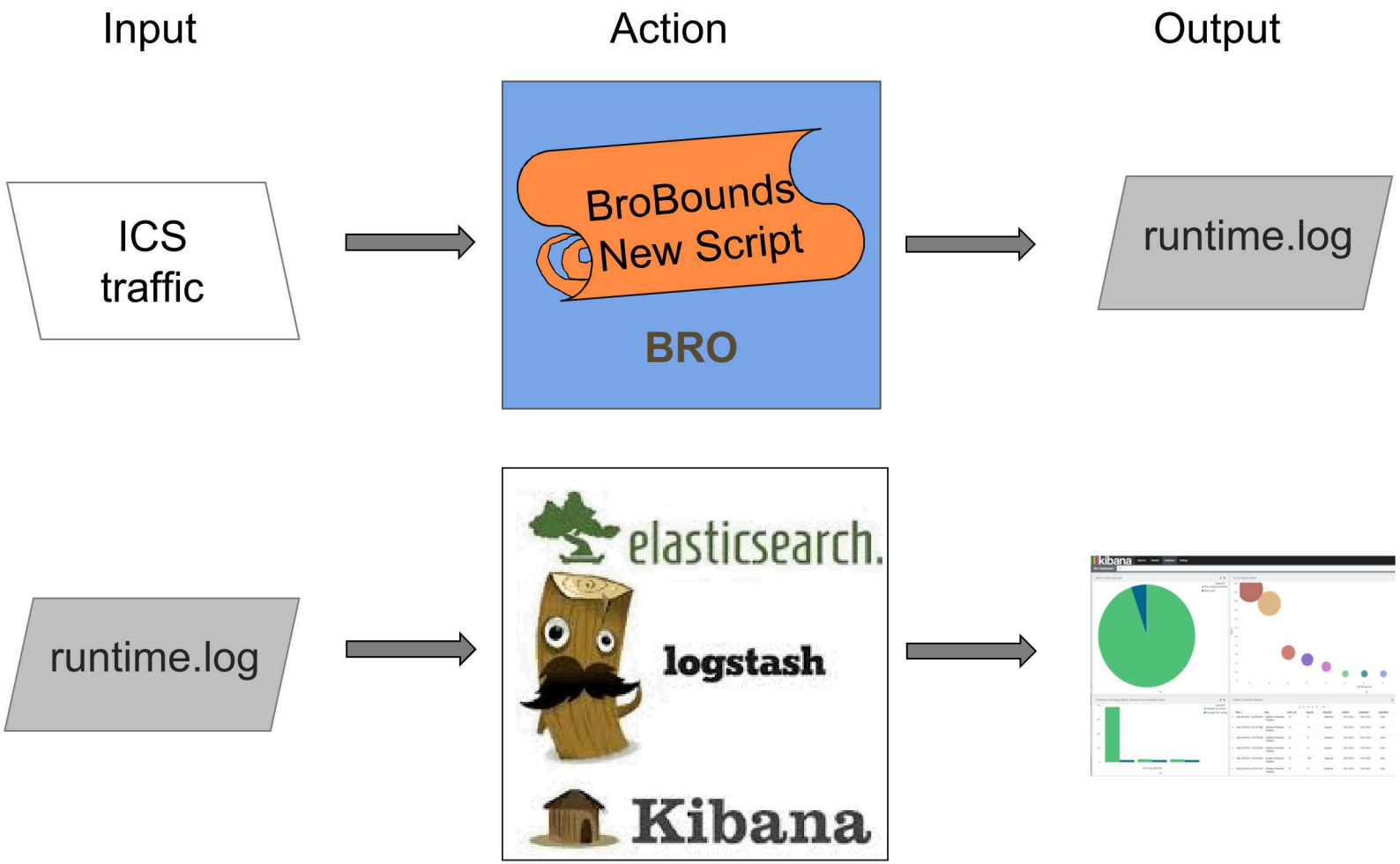
Generate Bro Script – Last Step



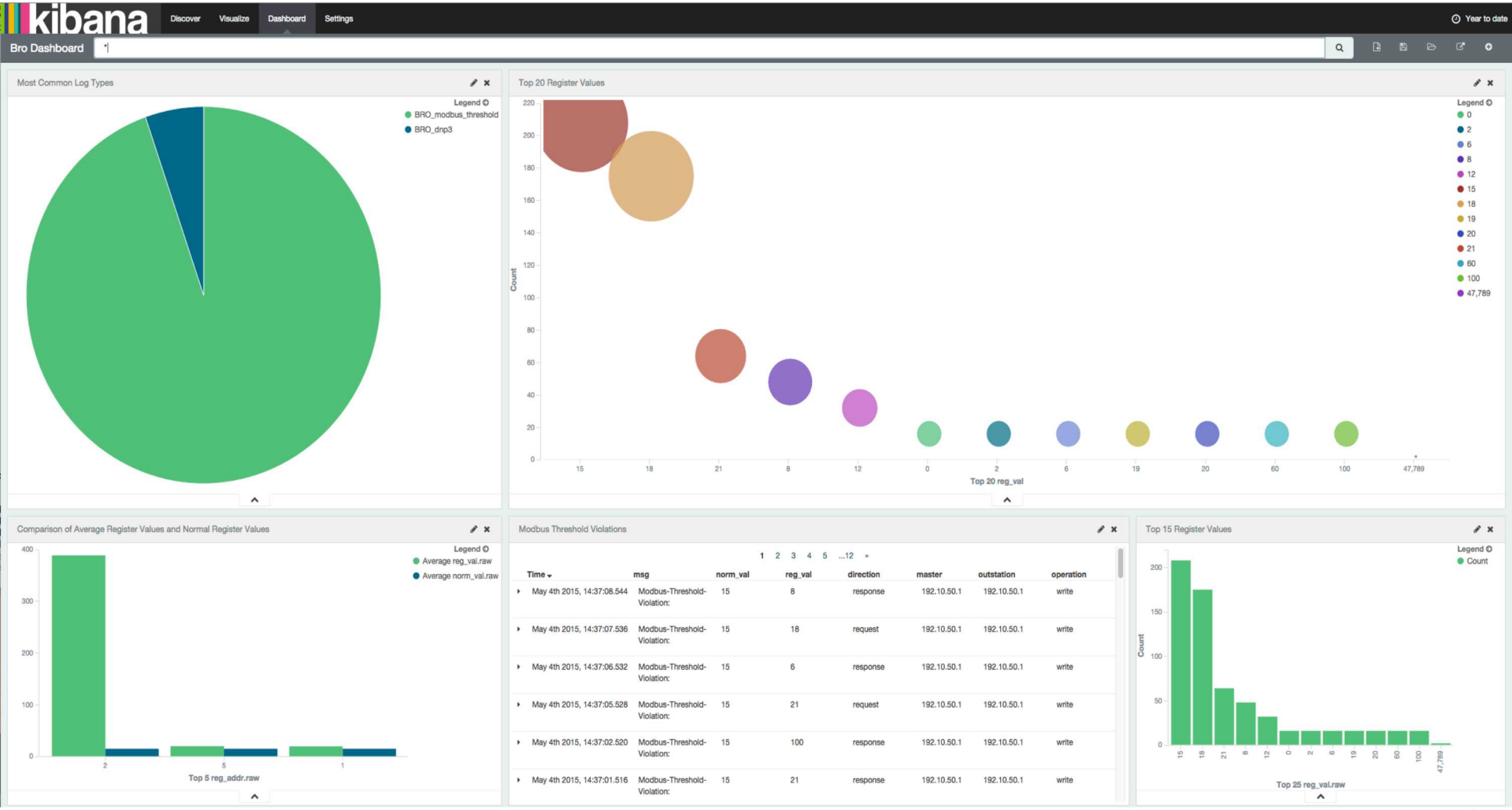
BroBounds Workflow – Create Script



BroBounds Workflow – Run Bro



BroBounds Dashboard



New Approach - Archimedes

Sandia National Laboratories

- Expand on BroBounds analysis model
- More statistical rigor to find anomalies
- In contrast, BroBounds was designed with very tight definition of “anomaly”
  - Simple integer comparison
  - To facilitate rapid development
- Integrates with existing network tools
  - BroBounds is stand-alone
  - New tool designed to integrate with ELK-stack workflow

What is Archimedes?

Sandia National Laboratories

- A platform for running one or more analysis modules on ICS protocol data

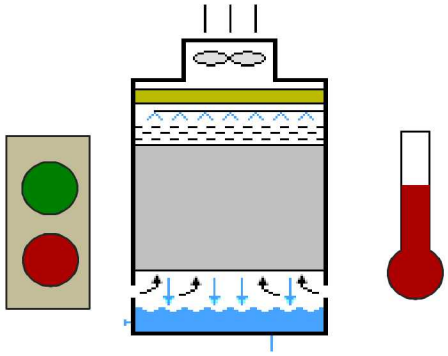
- A set of modules for finding anomalous behaviors from packets/messages which are:
  - Valid
  - Well-formed
  - "Harmless" – (when out of context)

- A platform for storing/sorting/displaying with a json database backend

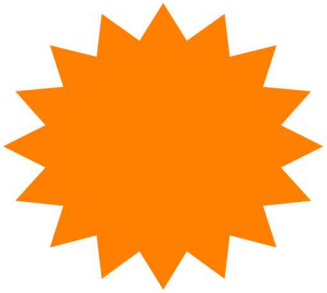
What is an Archimedes Analysis Module?

- A small program that runs an analysis algorithm on a set of ‘register-operations’
- A program that may call one or more analysis programs to find relationships including:
  - Statistical analysis
  - Machine learning
  - Any algorithm a computer scientist can devise
- A program that can be written in:
  - Python
  - Haskell
  - C/C++
  - Anything really…

Archimedes Workflow



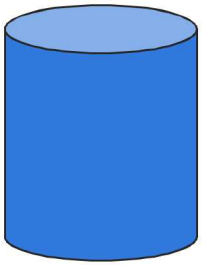
Process



Archimedes

Kibana

Bro



json database (Elasticsearch)

Why A New Approach?

- Multiple statistical approaches to find more kinds of anomalies
- Instead of stand-alone, distributed
- Backend/framework in Haskell
  - Analytics in multiple languages
  - Front-end in JavaScript
- Why Haskell?
  - Designed for mathematical reasoning
  - Designed for parallel processing
    - We are not yet using this, but plan to
- BroBounds is a one-instance-per facility tool
  - But Archimedes is designed to be distributed
  - Multiple analysts can use multiple instances at one time

Why Is This Work Important To ICS-CERT

- Tool will display results automatically to Kibana
- Design goal - CERT analysts will be able to deploy the tool quickly
- Design goal – easy to write and run new, complex algorithms for anomaly detection

How Does This Work Fit in the ICS-CERT Workflow

Sandia National Laboratories

- At present, BroBounds and ELK-stack is a proof-of-concept in a Virtual Machine
- New BroBounds is a prototype
  - Haskell framework in place
  - Python environment in development
  - “Toy” algorithms so far

Next Steps

- Continue Archimedes development
  - Develop more analytics
  - Test Python interface
- Add new ICS protocol analyzers to Bro
- Improvements to BroBounds as required

Questions