NRECA/DOE DE-OE0000684

Task 2 - High Level Design

July 7, 2015

Table of Contents

[Table of Contents 2](#_Toc424045620)

[1 Introduction 1](#_Toc424045621)

[2 Evaluation vs. Success Criteria 3](#_Toc424045622)

[3 High Level System Architecture 6](#_Toc424045623)

[3.1.1 Five Layer Abstract Model 6](#_Toc424045624)

[3.1.2 Overall System Architecture 7](#_Toc424045625)

[3.1.3 Mapping System Architecture to Abstract Model 8](#_Toc424045626)

[4 High Level Component Design 9](#_Toc424045627)

[4.1 Network Mapping 9](#_Toc424045628)

[4.1.1 Capabilities Overview 9](#_Toc424045629)

[4.1.2 High Level Diagram 10](#_Toc424045630)

[4.1.3 Passive Monitoring 10](#_Toc424045631)

[4.1.4 Active Probing 10](#_Toc424045632)

[4.1.5 Comparison Engine 11](#_Toc424045633)

[4.1.6 Key Data Elements 11](#_Toc424045634)

[4.2 Network Security Policy Definition 11](#_Toc424045635)

[4.2.1 Capabilities Overview 11](#_Toc424045636)

[4.2.2 High Level Diagram 12](#_Toc424045637)

[4.2.3 Security Policy Definition Console 12](#_Toc424045638)

[4.2.4 Security Policy Manager 13](#_Toc424045639)

[4.2.5 Network Topology Retriever 13](#_Toc424045640)

[4.2.6 Enforcement Data Retriever 13](#_Toc424045641)

[4.2.7 Traffic Analyzer Adapter 13](#_Toc424045642)

[4.2.8 Protocol Interpreter 13](#_Toc424045643)

[4.2.9 Key Data Elements 14](#_Toc424045644)

[4.3 Traffic Analysis 14](#_Toc424045645)

[4.3.1 Capabilities Overview 14](#_Toc424045646)

[4.3.2 High Level Diagram 15](#_Toc424045647)

[4.3.3 Network Traffic Capture 15](#_Toc424045648)

[4.3.4 Syntactic and Semantic Analysis 16](#_Toc424045649)

[4.3.5 Rule-Based Analysis 16](#_Toc424045650)

[4.3.6 Machine Learning 16](#_Toc424045651)

[4.3.6.1 Reporting Anomalies and Soliciting Feedback 16](#_Toc424045652)

[4.3.7 Key Data Elements 17](#_Toc424045653)

[4.4 Security Policy Enforcement 17](#_Toc424045654)

[4.4.1 Capabilities Overview 17](#_Toc424045655)

[4.4.2 High Level Diagram 19](#_Toc424045656)

[4.4.3 SDN and SCADA Network Controller Interface 19](#_Toc424045657)

[4.4.4 Enforcement Console 20](#_Toc424045658)

[4.4.5 Enforcement Engine 20](#_Toc424045659)

[4.4.6 Anomaly Parser 20](#_Toc424045660)

[4.4.7 User Action Feedback 21](#_Toc424045661)

[4.4.8 Key Data Entities 21](#_Toc424045662)

[4.5 Security Reporting 21](#_Toc424045663)

[4.5.1 Capabilities Overview 21](#_Toc424045664)

[4.5.2 High Level Diagram 22](#_Toc424045665)

[4.5.3 Security Reporting Console 22](#_Toc424045666)

[4.5.4 Policy Enforcer Retriever 23](#_Toc424045667)

[4.6 Network Topology Retriever 23](#_Toc424045668)

[4.6.1 Detector Retriever 23](#_Toc424045669)

[4.6.2 Policy Definition Retriever 23](#_Toc424045670)

[4.6.3 Key Data Elements 23](#_Toc424045671)

[5 Closing Remarks 24](#_Toc424045672)

[6 Contacts 25](#_Toc424045673)

**Document Change Control Log**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 12/27/2013 | 0.1 | Initial Draft | Dev. Team |
| 1/9/2014 | 0.9 | Draft submitted for project team comment | All Teams |
| 1/15/2014 | 1.0 | Final Version, submitted to DOE |  |
| 6/24/2015 | 1.1 | Revised Draft | Dev. Team |
| 6/29/2015 | 1.2 | Revised Draft (Update) | Dev. Team |
| 7/2/2015 | 1.3 | Revised Draft (Update) | Dev. Team |
| 7/7/2015 | 1.4 | Final, submitted to DOE |  |

# Introduction

This document contains a revised high-level design of the system for network management and monitoring as proposed to the Department of Energy. This revised high level design is a deliverable for cycle 2 (milestone M8), as outlined by the Project Management Plan (PMP) submitted to DOE on December 19, 2013, and it covers the full scope of the solution which is designed to meet the goals as outlined in PMP. As detailed in the PMP, these goals include provisioning of the following capabilities to electric utilities:

• A capability to bridge the gap between an electric utility’s operational network security policy and what is actually implemented at the network layer. Doing so creates a tight linkage between the specified policy and implemented network controls, helping network operators ascertain that the security policy is in fact properly implemented (i.e., what is implemented is the same as what is intended to be implemented).

• A capability to detect and prevent potentially malicious traffic flows within an electric utility’s operational network, by creating protocol-specific (e.g., initially specific to MultiSpeak®), semantically rich, and context-aware filtering rules to identify disallowed or anomalous traffic patterns.

• A capability to enable an electric utility to define and enforce its operational network security policies with fewer IT staff members and less reliance on significant internal security expertise.

• A capability to align an electric utility’s operational network security management with the broader trends of software-defined networking, virtualization, and the ongoing migration of utility IT and operational systems into the cloud environment (where they can be provided as a managed service). This alignment is particularly important for smaller electric utilities that have limited IT staff and capabilities.

• A capability to simplify functions such as security reporting and compliance as they relate to an electric utility’s operational network assets and traffic flows.

• A capability to ensure that network configuration changes meet an electric utility’s operational network security policy and that the policy is kept updated to reflect changing business needs and threats.

These capabilities are supported via five feature sets, specifically: network mapping, network security policy definition, traffic analysis, security policy enforcement, and security reporting. This revised high level design document covers the design for each of these five subsystems (corresponding to the five feature sets), including an integrated design view showing how these subsystems fit together.

The purpose of this revised high-level design document is twofold. First, it serves as a key design artifact for the project forming the basis for subsequent design and development. Second, it serves as a formal deliverable to the Department of Energy in accordance with milestone (M8), as specified in both the PMP and the SOPO, to be used as a basis for making a go/no-go decision.

This revised high level design document is based on the successful completion of the development and field testing of the cycle one system, and incorporating the lessons learned during cycle one. The primary focus is to (1) provide a unified operator UI to control over the network that the system is managing, (2) provide a uniform interface for multiple analysis applications to report alerts, especially anomalies from the machine learning subsystem, and (3) provide better operator understanding and situation awareness of the network through interactive network map and annotation.

# Evaluation vs. Success Criteria

* Are the project’s technical objectives being met?

The high-level design presented in this document will perform all of the functions identified in the initial concept paper. In addition, it addresses additional functionality that the team believes are valuable and achievable within the project scope. This table below lists key functionality. The items in bold extend the originally conceived functionality.

1. Allow specification of the network configuration in a manner which allows for machine review and validation
2. Collect network traffic
   1. Actively, in-line devices with mirrored or SPAN parts
   2. **Passively through custom developed device**
   3. **Passively through a virtual appliance**
   4. **Archive network traffic for prolonged periods of time**
3. Organize the information into a high speed, secure, **redundant** database for analysis
4. **Facilitate conventional validation of network traffic**
5. Provide sufficient data to characterize “normal” network traffic. This can include conventional data traffic (source and destination, time, etc.) but can also include deep packet content to allow the use of (1) validation of electrical message protocols and (2) electrical values (e.g. voltage) in detecting compromised systems.
6. Validate network flow
   1. Map the network
   2. Compare “as-is” network to “as-understood”
   3. **Implement a system to create checks that can be determined in advance**
      1. **Message structure**
      2. **Ranges of values**
      3. **Timing**
      4. **Rate of receipt**
      5. **Other, tbd**
   4. Detect deviations from normal by means of artificial intelligence
   5. Alert on discrepancies in (6b, 6c, or 6d)
7. Update model of normal operation using machine learning methods
8. On alert (6e), determine remediation actions based on risk analysis
9. Affect changes through:
   1. Changes to the network using software defined network capabilities
   2. **Communications to SCADA to affect grid operations**
   3. **Messages to human operators to make manual changes to network and/or grid**

The design meets all technical objectives.

* Do we realistically expect the product to meet its technical and cost goals?

At this point, the team’s research into commercial and available laboratory software provide a reasonable level of confidence that the designed system will meet its technical and cost goals. The team is conducting mini-research projects (“spikes”) to address high-risk areas, but to date have not found any insurmountable barriers to achievement of the goals.

* Are the technical objectives still relevant or have changes in requirements or have any of the following obviated the need for the project and its products:
  1. New or improved competing products or solutions
  2. Changing security requirements (e.g., the nature of threats and attacks)
  3. Changing grid cyber security regulations

As part of its design work, the team has been investigating available commercial security projects. To date we have not found any which provide the non-prescriptive approach at the heart of the system under development.

* Is spending consistent with the plan and are there sufficient resources available from all sources to carry the project through to a successful conclusion?

Spending is on track. Since the start of the project, we have sought material assistance from major organizations in the space. So far, Intel and McAfee have committed engineering and hardware for the project. We expect commitments from other organizations shortly. We are confident that this broader team can complete development.

* Have tests to date been successful or, if not, have we identified issues and either corrected the problem or gotten on the path to doing so?

We have not yet reached the testing phase for the cycle 2 stage.

* Is the team working as designed and are all partners still committed? Are changes needed?

The team is all under contract and working well. No changes are needed at this time.

* Is there still commercial interest in the product?

There is strong and growing commercial interest in the project. In addition to Cigital which has been a are core members of the team, there have been expressions of interest from IBM, Cisco, Intel, and McAfee as well as two smaller firms. At the present time, the realization of a commercial product appears to be likely with Intel. Key components of this design would be integrated into an Intel product.

* Summary/Recommendation

The team recommends that no changes to the PMP are necessary at this time and that the project should continue as scheduled.

# High Level System Architecture

### Five Layer Abstract Model

Prior to this project, CRN has done a significant amount of foundational work that culminates in a five layer abstract model for the system, as shown in the diagram below. The central idea of this abstract model is that – with each successive layer our understanding of the system matures. For example, we start with data in the network packets and successively mature our understanding of this data to identify “network events” in real time. Ultimately this understanding of “network events” enables us to make the informed decisions and take appropriate corrective actions.



Figure 1 – Five Layer Abstract Model

A brief description of each abstract layer follows:

* Layer 0: Security Policy Specification - Specifying security policies to guide all five layers of functionalities.
* Layer 1: Data Collection - Collecting network traffic data.
* Layer 2: Data Management - Organizing the network data into structured data stores and enhance the data with machine learning patterns and contextual data.
* Layer 3: Data Analysis - Analyzing the network traffic data to detect deviations from network configuration, specified rules, or normal traffic patterns.
* Layer 4: Decision - Determining how to respond to the detected anomalies and violations, and interacting with the user to take into account user feedback for the suggested decisions.
* Layer 5: Action - Enforcing the decisions made at Layer4 to mitigate the security risks on the network.

The high level design presented in this document adheres to this five layer abstract model.

### Overall System Architecture

The overall system architecture is depicted in the diagram below:

Figure 2 - Overall System Architecture Diagram

The system will be comprised of five subsystems. Each of these five subsystems will perform a distinct function and will provide interfaces that can be used by other subsystems. A brief description of each subsystem follows: ­

1. Network Mapping: The responsibility of this subsystem is to discover a utility network, usually comprised of a field network and back office network, and enable operational users to interact with the network using an intuitive graphical representation. This subsystem will also enable users to import network definitions. Finally, this subsystem will allow users to modify the graphical representation of the network.
2. Security Policy Definition: This subsystem is responsible for defining security policies. It will provide a user interface that will enable users to create policies using the network mapping component or upload policies from other sources.
3. Traffic Analysis: This subsystem is responsible for analyzing traffic utility networks to find anomalies in the traffic associated with utility-specific protocols (such as Multispeak™ or DNP3). The subsystem will use rule-based checks and machine learning methods in order to detect pattern anomalies or violations in real time.
4. Policy Enforcement: This subsystem is responsible for the enforcement of defined network security policies. The subsystem will overlay anomalies from the traffic analysis subsystem via the network mapping subsystem thereby allowing users to make network control decisions. It will include a console where changes can be viewed and modified by users.
5. Reporting: This subsystem is responsible for generating reports that will provide detailed information to users about the detected anomalies/violations, any action(s), and other related information, in a unified operation dashboard.

### Mapping System Architecture to Abstract Model

|  |  |  |
| --- | --- | --- |
| Abstract Layer | Software Module /  Subsystem | Component |
| Layer 0: Security Policy  Specification | Security Definition | Security Policy Manager  Security Policy Datastore |
| Layer 1: Data Collection | Traffic Analysis | Network Traffic Capturer |
| Layer 2: Data Management | Traffic Analysis | Detector Datastore |
| Layer 3: Data Analysis | Traffic Analysis  Network Mapping | Protocol Analyzer  Machine Learning Based Security Detector  Network Mapper  Network Topology Datastore |
| Layer 4: Decision | Security Policy Enforcement  Reporting | Enforcement Console  Enforcement Datastore  Enforcement Engine  Reporting Portal  Reporting Datastore |
| Layer 5: Action | Security Policy Enforcement | Network Controller |

# High Level Component Design

## Network Mapping

### Capabilities Overview

The network mapping component will offer the following capabilities:

1. Passive and active network scanning.
2. Network host detection.
3. Logical connection mapping.
4. Importation of network configuration files.
5. Annotation of network components
6. Provide visual depictions of deviations from the desired network state.

### High Level Diagram

Figure 3 - Network Mapping Subsystem

### Passive Monitoring

The passive monitoring component listens to routing protocol traffic to map networks. The protocol configuration data will allow the passive monitoring component to parse and interpret data for common routing protocols (including OSPF, BGP, and ISIS). The passive monitoring component will calculate primary and secondary routes, and detect routing failures which will impact the network topology. The information gathered by the passive monitoring component will be normalized to remove redundancies, and combined with active probing data to add to the Network Topology Datastore.

### Active Probing

The active probing component sends data to back office and utility network devices in an attempt to discover and further identify devices that may not be found via passive monitoring. Simple Network Management Protocol (SNMP) is included in this component, and will enable communication with network devices, including switches and routers. SNMP queries network hardware (including switches and routers) to get information such as routing tables, IP addresses, switch ports and hostnames.

This component will also interface with the “north-bound” interface of an SDN controller, which will allow the component to query the controller for information about the network architecture. The information collected by the active probing component will be combined with information from the passive monitoring component to form a more complete network map.

The Network Topology Datastore will be accessed by the Enforcement Engine, the Reporting Console, and the Security Policy Datastore. Sharing the Topology Datastore across components will enable interactive graphical user interfaces (GUIs), where enforcement decisions and security policies can be overlaid on a map of the network.

### Comparison Engine

The network mapping console utilizes a comparison engine that determines if the actual network state has deviated from the desired network state. Once this is done, the console will provide the user with the ability to see these deviations graphically. This is done to provide the user with a manageable way to perform the following functions:

* Annotate deviations for further triaging
* Remediate deviations from the desired network state
* Reuse past experiences to handle similar problems

### Key Data Elements

The primary key data element for this subsystem is the network topology map. This map will store the physical connectivity between automatically discovered network elements. The map will enable the presentation of a graphical view of the network which will be overlaid with additional information from other subsystems (e.g. network anomalies). The map will be dynamically updated with network architecture changes, although the frequency of updates is expected to be low due to the static configuration of most utility networks.

## Network Security Policy Definition

### Capabilities Overview

The Network Security Policy Definition subsystem will offer the following capabilities:

1. The creation of context-specific network security policies using granular rules governing the flow of traffic.
2. A view that superimposes the network security policies on the network map.
3. A GUI interface to facilitate the translation of network security policies into network access policies.
4. Applying enforcement knowledge to improve the security policies.
5. Managing rules for SCADA network traffic

### High Level Diagram



Figure : Security Policy Definition Subsystem

The Network Security Policy Definition subsystem has six main components, as depicted in Figure 4. The main functionalities of each component are described in subsequent sections.

### Security Policy Definition Console

The Security Policy Definition Console provides a user interface for network administrators to define security policies and rules. The rules are based on attributes such as protocol, user, service, source host, destination host, source port and destination port. The policies and rules can also be defined using smart-grid specific protocol syntactic and semantic attributes.

The console can also provide support for importing rules or policies from files. It can also use the Protocol Interpreter to support context driven policies or rules based on smart-grid specific protocol syntax.

This console also provides a policy view superimposed on the network topology map. It visualizes the policies at the network nodes and allows the user to assign or adjust security policies or rules applied to those nodes.

The console component depends on the business logic in the Security Policy Manager to provide policy definition services. It can optionally retrieve security policies directly from the Security Policy Datastore.

### Security Policy Manager

The Security Policy Manager is the core of the Network Security Policy Definition subsystem. It encapsulates the business logic behind the security policy definition and exposes a set of Application Programming Interfaces (API) to the Security Policy Definition Console. It stores all the data in the Security Policy Datastore.

The Security Policy Manager (SPM) depends on the Network Topology Retriever to provide information about the network topology. The SPM also depends on the Machine Learning Enhancer to provide the necessary traffic pattern knowledge to enhance security policies or rule definitions.

The SPM supports context driven policy and rule definition, by incorporating the syntactic understanding of various smart-grid specific protocols. At the context level it maintains a semantic understanding of the allowed and expected actions, thereby enabling a deeper understanding of the protocol.

### Network Topology Retriever

The Network Topology Retriever is an adapter to the Network Mapper subsystem. It retrieves network topology data on demand or at regular intervals to support the Security Policy Manager. It can store the network topology in the Security Policy Datastore.

### Enforcement Data Retriever

The Enforcement Data Retriever is an adapter to the Security Policy Enforcement subsystem. It retrieves data anomalies and user decisions on them. If certain criteria are met, it will make recommendation to the user to change existing rules or policies or to add new rules or policies.

### Traffic Analyzer Adapter

The Traffic Analyzer Adapter provides rules and policies to the Traffic Analysis subsystem to support rule-based traffic analysis and/or machine learning. Rules for assigning trust score to a violation or anomaly or alert are also supported.

### Protocol Interpreter

The Protocol Interpreter encapsulates the syntactic and semantic knowledge of smart-grid specific protocols and provides a set of APIs to extend its knowledge to the Security Policy Manager and/or the Security Policy Definition Console. It can store smart-grid specific syntactic and semantic data in the Security Policy Datastore to support policy definition.

### Key Data Elements

Following are the key data elements of the Network Security Policy Definition subsystem. More detailed definitions of these entities will become available when the business requirements and use cases are specified.

* Security rule - a granular enforceable condition and the associated allowed or expected network actions. A rule conditions can be defined based on network topology, protocol context, or security patterns
* Security policy - a named set of Security Rules and its enforcement context, including the network topology, the protocol context, or security patterns. It can also specify allowed or expected network actions based on this enforcement context.
* Security rule type – A value to indicate the purpose of the rule, for traffic analysis, trust scoring, or flow control.

## Traffic Analysis

### Capabilities Overview

The Traffic Analysis subsystem will offer the following capabilities:

1. Capture network traffic of interest from the back office and field networks, including those from SCADA devices.
2. Perform syntactic and semantic protocol analysis on common electric utility protocols, including SCADA protocols.
3. Utilize a machine learning algorithm to determine a “baseline” view of network traffic.
4. Detect anomalous traffic patterns using the machine learning algorithm, and alert the Policy Enforcement subsystem.
5. Detect security policy violations and send them to the Policy Enforcement subsystem.

### High Level Diagram



Figure 5 - Traffic Analysis Subsystem

### Network Traffic Capture

The Network Traffic Capture (NTC) component will passively collect data flowing through the back office and field networks. Depending on the network architecture of the utility, multiple NTC components may be deployed to act as “agents,” which will allow the system to collect data from multiple locations.

The NTC component will also store captured frames in order to buffer data, and to enable historical data processing. Historical data will allow the machine learning algorithm to compare current conditions to other temporally related data (e.g. comparing weekend conditions to a previous weekend). This will allow the Machine Learning Algorithm to achieve greater accuracy by understanding how traffic conditions change over time. Data captured by the NTC component will be sent to the syntactic and semantic protocol analyzers.

### Syntactic and Semantic Analysis

The syntactic protocol analysis will determine the validity of the each message captured by the NTC component. Syntactic analysis determines if the arrangement of data inside the message is valid according to the rules of the message’s protocol configuration. The protocol configurations will initially be programmed with common utility network protocols, initially MultiSpeak™ but eventually extended to DNP3, MODBUS, and SCADA protocols.

The semantic protocol analysis will examine messages for their meaning, structure, and significance at the time they were sent. The semantic analyzer will also utilize the protocol configuration data to inform the analysis.

The syntactic and semantic analysis components are designed to filter out erroneous data from entering the machine learning algorithm, and to enable the algorithm to build a model of “normal” network traffic. The semantic analysis will understand the normal ranges of values of the electrical system traffic and detect deviations to facilitate electrical system risk assessment by the Enforcement Engine.

### Rule-Based Analysis

The rule-based analysis component analyzes captured traffic against rules stored in the security policy datastore. The rule-based analysis component sends data to the policy enforcement subsystem when a rule violation is detected. The component will also assign a trust score to each violation.

### Machine Learning

The machine learning component of the Traffic Analysis subsystem consists of all the components inside the dashed line in Figure 5. The first step in Machine Learning (ML) is to normalize the data. Data Preprocessing ensures that redundancies and dependencies in the input data are reduced. Redundancies may occur if identical data is captured in multiple places by the network traffic capture component (if multiple component “agents” are deployed). This is especially important for the system since the volume of data being processed is likely to be very high. Data vectorization occurs after normalization. Vectorization is the arrangement of all relevant data into one common format for the ML to process.

The data processing element is the core of the ML. Data processing extracts features from the data and builds a model of normal network activity, utilizing Bayesian analysis (in parallel with other techniques) to categorize data points as “normal” or as anomalous. The anomaly detection component informs the Enforcement Engine of potential anomalies and assigns trust scores to them.

### Reporting Anomalies and Soliciting Feedback

The machine learning component will report anomalies to the dashboard where operators can make decisions to carry out policy enforcement. This component will provide a REST service API for other applications to provide feedback to indicate when a reported anomaly is deemed normal by operator and adjust the detection accordingly to not report the same anomaly in the future. The machine learning component identifies an anomaly with a unique anomaly identifier. The anomaly identifier provides a consistent and scalable way for other applications to provide feedback for anomalies that were detected. For example, in the event that an alert should no longer be considered an anomaly, an application can pass the anomaly identifier to the machine learning component and say that the event should no longer be considered an anomaly. In this sense, this behavior that was previously deemed to be anomalous from normal behavior would now become part of the desired/accepted network state.

### Key Data Elements

The key data elements for this subsystem are the protocol configurations and data models. These data elements will enable the processing of captured traffic. The traffic analysis subsystem will generate a list of anomalies with associated metadata (including trust scores and anomaly type) which will be used as the basis for the Enforcement Engine to make decisions. The Policy Definition subsystem will also use this list to enable new policies to be created based on anomalies that have been discovered. Essence will currently focus on four kinds of anomalies:

* A connection between a source and one destination
* A connection source and a set of destinations
* A source receiving/sending an MultiSpeak™ message name
* A source receiving/sending a set of Multispeak™ message name

We’ve decided to call these kinds of anomalies “patterns.” For example, if the pattern is “A connection between a source and a set of destinations”, the user will be able to know that this is the kind of anomaly that took place. Source 1 sending an abnormal amount of messages to destination 1 can indicate one type of attack pattern, while source 1 sending an abnormal amount of messages to a set of destinations can indicate another. The threshold of the first one can be quite different from that of the second.

## Security Policy Enforcement

### Capabilities Overview

The Policy Enforcement subsystem will:

1. Receive anomalies and violations with their associated trust scores from the traffic analysis subsystem and raise alerts if necessary.
2. Use the trust score associated with an anomaly or violation and the contextual information to come up with the suggested enforcement action.
3. Assess electrical system risks based on network anomalies, violations, or semantic analysis on the traffic content.
4. Provide users a method to view the anomalies and violations with suggested enforcement actions.
5. Enable users to take enforcement actions.
6. Interface with the SDN controller to initiate corrective actions on impacted network elements, if any.
7. Interface with the SCADA controller to initiate corrective actions on impacted SCADA devices, if any.
8. Provide feedback to the Network Security Policy Definition subsystem regarding user’s actions.
9. Provide users an ability to dynamically configure the system to automatically take corrective action.

### High Level Diagram



Figure - Policy Enforcement Subsystem

### SDN and SCADA Network Controller Interface

Software Defined Networks (SDN) Network Controller Interface component will provide the rest of the enforcement subsystem an SDN Controller Protocol API (SCP-API) to interface with the actual SDN controller. Since SDN technology is rapidly evolving, it is beneficial to abstracts out a “north-bound interface” provided by a typical SDN controller. In doing so, the rest of the enforcement subsystem will use the SCP-API to communicate with the SDN controller.

The SCADA Controller interface will provide the policy enforcement subsystem a means of accessing SCADA equipment. The SCADA controller will be able to send commands to the SCADA network to perform policy enforcement.

### Enforcement Console

The Enforcement Console represents the user interface of the Policy Enforcement subsystem. It enables users to view anomalies and violations received from the Traffic Analysis subsystem and take informed enforcement actions, if any. Each anomaly or violation will be associated with a suggested set of enforcement actions. The users can:

1. Agree with the suggested enforcement actions,
2. Disagree with the suggested enforcement actions and instead take actions that are different from the ones suggested, or
3. Disagree with the enforcement actions and take no further action.

The console will capture the user’s response to an anomaly or violation and sent it to the Network Security Policy Definition subsystem for necessary adjustments.

The enforcement console will allow the system to be dynamically configurable over a period of time. For example during the first deployment, a user may want to validate/check every anomaly, violation, and suggested enforcement action. However, as the system matures, the users may find that system is accurate in suggesting enforcement actions for a subset of anomalies. In such cases, the user may configure the system to automate the enforcement action.

### Enforcement Engine

The Enforcement Engine is responsible for receiving anomalies and their associated trust scores from the Traffic Analysis subsystem and combining this information with contextual information to suggest and execute enforcement actions. Some examples of the contextual information considered are:

* The network topology, including:
  + Network elements impacted,
  + The current security sensitivity of such network elements,
  + And the current state of such elements (e.g. operations may be in the process of conducting maintenance on the component).
* Type of traffic that triggered the anomaly
* Any other contextual data (e.g. time of the day when the anomaly was received)

The enforcement engine will also perform a risk-based analysis for suggested actions to determine their impact on the electric system. The enforcement engine will use security policy and network state to inform the risk-based analysis.

### Anomaly Parser

The Anomaly Parser component is responsible for parsing the incoming anomalies (from the traffic analysis subsystem) and storing relevant data in the Enforcement Datastore. This component will work with a set of predefined anomaly types and their associated data format. All valid anomalies will be parsed and stored in the Enforcement Datastore.

### User Action Feedback

This User Action Feedback component is responsible for providing user action feedback to the Network Security Policy Definition subsystem. As described in the section 4.4.4 (Enforcement subsystem), the user may choose to ignore the suggested corrective action, take a different action, or ignore the anomaly altogether. It is expected that the Network Security Policy Definition subsystem will use this feedback to adjust its policies and rules.

### Key Data Entities

The policy enforcement subsystem will manage the following data entities -

1. Anomaly Types – same as the ones managed by the Traffic Analysis subsystem
2. Anomalies – same as the ones managed by the Traffic Analysis subsystem along with their trust score
3. Violations – same as the ones managed by the Traffic Analysis subsystem along with their trust score
4. Enforcement Actions – suggested actions based on the context and the trust score
5. User Actions
   1. User’s enforcement actions (these could be different from those suggested by the system)
   2. Dynamic configuration changes by the user
6. Supported SDN Controllers and their profiles
7. Supported SCADA Controllers and their profiles

## Security Reporting

### Capabilities Overview

The Security Reporting subsystem will offer the following capabilities:

1. Providing security operators with detailed data on security policy violations.
2. Providing security operators with detailed data on security alarms and anomalies/violations.
3. Presenting security alarms and anomalies superimposed on the virtual network topology.
4. Providing security operators with history and trending on security alarms and anomalies.

### High Level Diagram



Figure - Security Reporting Subsystem

The Security Reporting subsystem has five main components, as depicted in the diagram above. The main functionalities of each component are described in subsequent sections.

### Security Reporting Console

The Security Reporting Console provides a user interface for network operators to track policy violations, traffic anomalies, and alerts in a visual manner on top of the relevant network topology map. It can also display the trending of policy violations, traffic anomalies, or alerts using the historical data that are stored in the Reporting Datastore.

The network operator can drill down into details on those violations, anomalies, or alerts for example related the detailed traffic data, policy information, etc.

A key feature of this console is that it will have the ability to integrate with any Essence application that needs to provide the network operator with information about the network. This console will also provide the user with the capability to monitor Essence components and make changes to them.

Another key feature for this console is that it will allow different Essence applications to communicate with one another in order to present the user with a view of the desired network state opposed to the actual network state.

### Policy Enforcer Retriever

The Policy Enforcer Retriever interacts with the Security Policy Enforcement subsystem to provide the following support to Security Reporting:

1. Retrieving enforcement actions and user actions from the enforcement process and storing it in the Reporting Datastore;
2. Retrieving enforcement actions and user actions in a consistent manner to provide to the Security Reporting Console component for display and drill-down;

## Network Topology Retriever

The Network Topology Retriever is an adapter to the Network Mapping subsystem. It retrieves relevant network topology data on demand or in batch and stores them in the Reporting Datastore. The retriever attempts to associate their network topology with relevant policy violations, traffic anomalies, and alerts. It retrieves network topology information in a consistent manner to provide to the Security Reporting Console component for display.

### Detector Retriever

The Detector Retriever provides an interface to the Machine Learning Traffic Analysis subsystem so that it can retrieve information about traffic anomalies, violations, or alerts; this information will subsequently be stored inside of the Reporting Datastore. This interface will be implemented with a REST service API; the API will allow future detection/analysis engines to provide it with anomalies or alerts. The Detector Retriever fetches traffic anomalies, violations, and alerts from the Reporting Datastore in a consistent manner to support the Security Reporting Console.

### Policy Definition Retriever

The Policy Definition Retriever is an adapter to the Security Policy Definition subsystem. It retrieves relevant policy information from the Security Policy Datastore and stores them in the Reporting Datastore to provide policy details for security reporting.

### Key Data Elements

The Security Reporting subsystem utilizes data from other subsystem to construct the reports in the Security Reporting Console for network operators. The main data elements are:

1. Security policy violation
2. Network traffic anomaly
3. Network traffic alert

# Closing Remarks

The Essence team has confidence in the high level design articulated in this document and has developed a proof of concept that successfully captures live data in an operational network cooperative, successfully parses MultiSpeak™ in real time, identifies custom rules violations, generates an annotated network graph in real time, and continuously identifies anomalous MultiSpeak™ traffic in real time. This proof of concept was tested at 2 different cooperatives with a custom device and at 1 cooperative over the course of 6 weeks with a virtual appliance. It is important to note that the ability of Essence to perform the functions articulated in this design document via a custom device and virtual appliance are enhancements that were not posed in the original PMP submitted to DOE; these enhancements will substantially increase the number of electrical utilities that have network infrastructures compatible with Essence. The Essence team has established close relationships with the Shenandoah EC cooperative in Virginia, and the Central, Wake and South River cooperatives in North Carolina. These relationships can be utilized by the team to ensure that each milestone in the PMP can be verified with a solution that can work in the field.

# Contacts

| Contact | Title | Organization | Phone # | Email Address |
| --- | --- | --- | --- | --- |
| Robert Larmouth | Project Manager | NRECA | 603-930-9199 | robert.larmouth-contractor@nreca.coop |
| Craig Miller | Principal Investigator | NRECA | 703-626-9683 | Craig.Miller@nreca.coop |
| Evgeny Lebanidze | Managing Consultant, Development Team | Cigital, Inc. | 703-404-9293 x1237 | elebanidze@cigital.com |
| Chandu Ketkar | Technical Manager, Development Team | Cigital, Inc. | 703-404-9293 x1256 | cketkar@cigital.com |
| Andre Joseph | Senior Security Consultant, Development Team | Cigital, Inc. | 703-404-9293x1364 | ajoseph@cigital.com |
| Ping Ning | Senior Security Consultant, Development Team | Cigital, Inc. | 703-404-9293 x1359 | pning@cigital.com |
| Bob Wintemberg | Security Consultant, Development Team | Cigital, Inc. | 703-404-9293 x4224 | bwintemberg@cigital.com |
| Terrence Wong | Development Team | CMU | 412-268-4733 | twong@andrew.cmu.edu |
| Zico Kolter | Development Team | CMU | 412-268-4733 | zkolter@cs.cmu.edu |
| Phil Craig | Senior Cyber Security Engineer | PNNL | 509-375-4464 | pc@blackbytecyber.com |
| William Hutton | Senior Cyber Security Scientist | PNNL | 509-371-6695 | william.hutton@pnnl.gov |