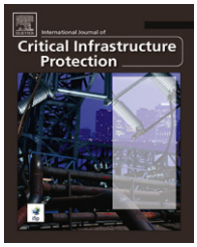
available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/ijcip



Budget constrained optimal security hardening of control networks for critical cyber-infrastructures☆

*Zahid Anwar**, *Mirko Montanari*, *Alejandro Gutierrez*, *Roy H. Campbell*

Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, United States

ARTICLE INFO

Article history:
Received 27 October 2008
Received in revised form
5 February 2009
Accepted 5 February 2009

Keywords:
Security Hardening
Power grid
IEEE 118 bus test
Budget constraints

ABSTRACT

Competing schemes for security-hardening the power grid differ in their installation costs and the amount of coverage they provide against cyber attacks. Manually mapping schemes to vulnerable assets, where each asset has a unique degree of criticality in an arbitrary power network configuration, is a cumbersome process. Moreover finding an optimal scheme combination so as to maximize overall network security under a fixed budget constraint is an NP hard problem. In this paper we describe a dynamic programming solution to this problem and implement it along with logic-based models of the power grid, its control elements and best security practices as a tool-chain. The tool-chain takes, as input, a power network configuration, and the budget constraints and security schemes described in logic, determines the critical assets and automatically selects an optimal scheme combination to apply to maximize security. We demonstrate the feasibility of the tool chain implementation by security hardening the IEEE power system 118-bus test case from a pool of five different best-practice schemes.

1. Introduction

The power grid has been designed with the N-1 principal in mind, meaning that it is built to survive at least one failure. The redundant design is such that if a power asset such as a transmission line or generator were to fail the power would be routed from elsewhere without causing major blackouts. However, infrastructure that resists single points of random failure, may not survive malicious, intelligent attacks by disgruntled employees, terrorist networks, etc, especially if this redundancy is in the power network alone, without isolation in the control. Consider two redundant power lines, designed to handle the extra load if one or the other goes down, but essentially controlled by a common vulnerable relay.

Relays are a popular choice for protection and control in power utilities. They communicate to constantly monitor the status of equipment, and participate in real-time pilot protection schemes [1] that involve detecting and agreeing on presence of faults, de-energizing equipment to protect against short circuits and reclosing circuits automatically in an attempt to clear faults. Relays are programmed to send alarms to operational personnel in case faults cannot be cleared automatically. Despite their important role, relay configurations are generally set up with convenience and efficiency in mind rather than security. Their open accessibility from the enterprise and office LANs and sometimes even the Internet gives the adversary easy opportunities for attacks.

Various government and advisory agencies have published a substantial amount of literature [2–4] on best practices

☆ This work was funded by the UIUC TCIP Project NSF CNS 05-24695.

* *Corresponding author.*
E-mail addresses: anwar@illinois.edu (Z. Anwar), mmontan2@illinois.edu (M. Montanari), agutier3@illinois.edu (A. Gutierrez), rhc@illinois.edu (R.H. Campbell).

1874-5482/$ - see front matter
doi:10.1016/j.ijcip.2009.02.001

for increasing the protection (security hardening) of the power grid control networks against malicious cyber attacks. Unfortunately incidents keep reoccurring [5] due to nonconformance to these schemes, resulting in loss of power, revenue and harm to consumers. These best practices can be implemented using security schemes that differ vastly in the kind of protection they provide against attacks. For instance, firewalls may hinder DoS against control devices but will not prevent eavesdropping attacks. A link encrypter, on the other hand, may solve the latter attack but might prove ineffective against DoS. Additionally cost and effort to implement the two schemes will differ; consider just upgrading the firmware on a router to provide firewall services as opposed to buying special hardware to provide 'bump in the wire' encryption for the real-time traffic demands of control devices. Similarly, cyber assets for controlling power system resources differ vastly in terms of criticality. For instance it would be be unwise to invest in an expensive security lock down of a substation that contributes less than 500 MW to the grid at the expense of cost saving on the security planning for a 50,000 MW power plant.

Perfect security is ideal but in reality security administrators are usually faced with budget constraints and end up trying to balance cost and security. This kind of manual cyber-security planning for a network the size of the power grid can easily become intractable and is actually an NP hard problem. The proof lies in a straightforward reduction from the Multiple-Choice 0-1 Knapsack problem (MCKS) and is detailed in Section 3. The contribution of this paper is that it recognizes the importance of spending more capital (more powerful security controls) on securing assets that are more critical. Security schemes are presented that use best practice guidelines from NIST [3] and other advisory agencies e.g. firewall, VLAN segregation, link encryption to isolate redundant power network assets in the control networks. Metrics have been proposed to evaluate the protection provided by security schemes, the cost to implement them, and determine the criticality of equipment in terms of revenue loss incurred in the event of their compromise. A pseudo-polynomial time automated solution is proposed that uses these metrics together to determine the optimal scheme selection to maximize security given a fixed budget allocated for power grid security hardening.

In our previous work [6,7] it was shown that logic-based models of the power grid and its control elements can be used for automatic conformance checking for adherence to best security practice schemes. Similar to [7] this paper uses first order predicate logic to model power networks which consist of a set of devices such as buses, lines, loads, generators and the corresponding control network consisting of relays and their network connections. By extending, the logical model to include additional attributes such as *overloading violations*, *power flows* and *costs*, we can automatically evaluate for any arbitrary power grid control network, which combination of security schemes would best protect against a knowledgeable adversary who attempts to maximize his damage dealt by attacking relays that control the most critical equipment.

The remainder of this paper is organized as follows: Section 2 provides related work on techniques for securing networks under certain constraints. Section 3 outlines the design of our security model and Section 4 shows our implementation via a tool chain based on Prolog. Section 5 describes the details of an evaluation case study of our model on the 118-bus test case to demonstrate the tool-chain functionality. We conclude the paper with a short discussion and future work in Section 6.

2. Related work

Our research benefits from related work on formal analysis and model checking of the security of large scale safety critical systems, as well as surveys of control systems.

A survey of a control system and the security controls in place [8] reveals a lack of authentication mechanisms, little or no patch updates, and numerous uncontrolled interconnects to the public Internet. The authors were able to break into oil and power production systems by using simple exploits such as SQL Injection. The paper argues that even with knowledge of individual vulnerabilities in the nodes of the system, there are no adequate tools for reasoning about the overall security of the system.

The SINTEF CORAS project [9] supports methodologies for risk analysis of security-critical systems by modelling threats to a system as unwanted features of the system in question. This allows users to model a system and its associated threats as Unified Modelling Language (UML) diagrams. With the UML diagrams the users can perform security risk assessment. It also provides an XML schema for exchanging the risk assessment data and a vulnerability assessment report format. This allows system designers and users to communicate in a more formal and standardized language.

Dewri et al. [10] use attack trees to model networks and employ evolutionary algorithms to solve the optimization problem of what subset of security measures to use so that the cost of implementing these measures and the cost of residual damage is minimized.

Oman et al. [11] use a graph model with multi-dimensional edge properties to characterize device connectivity in an electric power system. By summing the weights of the edges required to traverse and compromise a target device, the authors are able to determine the most vulnerable access paths within their model. The authors however do not elaborate on modeling mitigation strategies or on the impact of a remote attack on power system devices.

Salmeron et al. [12] use bilevel mathematical models to determine the most critical components in a power grid network, i.e., those that, if taken down, will cause the most disruption in the network. Their model however does not include the use of any security schemes to protect against attacks.

In [13] the authors describe a set of security tools suitable for the stringent communication demands of power networks as well as which meet the CIP security standards set by NERC and other government agencies. Merits of devices such as crypto-modems, secure communication processors and firewall solutions are discussed along with their installation costs. While it is clear from this work that multiple competing tools and schemes exist for protecting power networks, how

to map them to individual configurations is unclear. Certainly applying these controls in the entire network would be too cost prohibitive to be feasible.

3. Design

3.1. Power system model

A power system is an electric network consisting of a set of power devices $D = \{B \cup E \cup F \cup G \cup L\}$ where

- B buses;
- E branches where $E \subseteq B \times B$;
- T transmission lines where $T \subset E$;
- F transformers where $(F \subset E) \wedge (E \setminus \{T \cup F\} = \emptyset) \wedge (\{T \cap F\} = \emptyset)$;
- G generators;
- L loads;

and a set of control devices called relays R, and substations S where

$$S = \{S_i | S_i \subseteq B\} \wedge \bigcup_{\forall i} S_i = B \wedge \forall_{\substack{i,j \\ i \neq j}} S_i \cap S_j = \emptyset;$$

the relations:

ConnectedTo zCb	where $z \in E \cup G \cup L \wedge b \in B$;
Controls rNd	where $r \in R \wedge d \in D \setminus B$;

and a set of functions:

$$linesin(b_i) : b_i \rightarrow \mathbb{P}\{T\} \text{ bus to lines mapping where } b_i \in B; \quad (1)$$

$$transin(b_i) : b_i \rightarrow \mathbb{P}\{F\} \text{ bus to transformers mapping where } b_i \in B; \quad (2)$$

$$gensin(b_i) : b_i \rightarrow \mathbb{P}\{G\} \text{ bus to generators mapping where } b_i \in B \text{ and } \forall_{\substack{b_1,b_2 \\ b_1 \neq b_2}} gensin(b_1) \cap gensin(b_2) = \emptyset; \quad (3)$$

$$ldsin(b_i) : b_i \rightarrow \mathbb{P}\{L\} \text{ bus to loads mapping where } b_i \in B \text{ and } \forall_{\substack{b_1,b_2 \\ b_1 \neq b_2}} ldsin(b_1) \cap ldsin(b_2) = \emptyset; \quad (4)$$

$$controls(r_i) : r_i \rightarrow (d_i, b_i) \text{ relay to device and bus mapping where } r_i \in R, d_i \in D \setminus B \text{ and } b_i \in B; \quad (5)$$

$$power(d_i) : d_i \rightarrow P \text{ device to power mapping where } P \in \mathbb{R}_{\geq 0} \text{ and } d_i \in D; \quad (6)$$

A power system is typically depicted as a graph in one-line diagrams where nodes are buses and edges are branches. Branches can be either of type transmission lines or transformers, through which electrical energy is transmitted to supply customers. Devices in a power network conduct power (lines, buses), generate (generators) or consume it (loads) as depicted in function (6). Power flow (energized or deenergized) in a device is controlled by breaker/relay combinations, henceforth called just relays, at the point the device connects to a bus, and can be queried by function (5).

A set of buses are functionally grouped together to form substations. We distinguish between three types of substations, as shown by the predicates (7)–(10). (1) A **power plant** is characterized by one or more generators connected to at least one of the buses. (2) A **distribution substation** has no generators and is characterized by one or more loads connected to one of the buses. (3) A **transmission substation** has no generators and loads, connects two or more transmission lines and may have transformers to convert between two transmission voltages.

$$powerplant(s_i) = s_i \in S \wedge \exists b_i [b_i \in s_i \wedge gensin(b_i) \neq \emptyset]; \quad (7)$$

$$dist\_substation(s_i) = s_i \in S \wedge \forall b_i [b_i \in s_i \wedge gensin(b_i) = \emptyset] \wedge \exists b_j [b_j \in s_i \wedge ldsin(b_j) \neq \emptyset]; \quad (8)$$

$$trans\_substation(s_i) = s_i \in S \wedge \forall b_i [b_i \in s_i \wedge gensin(b_i) \cup ldsin(b_i) = \emptyset \wedge \exists t_i, \exists t_j [t_i, t_j \in linesin(b_i) \wedge t_i \neq t_j]]; \quad (9)$$

$$substation(s_i) = s_i \in S \wedge (trans\_substation(s_i) \vee dist\_substation(s_i)). \quad (10)$$

For instance the left part of Fig. 1 shows an example of a simple power system consisting of 4 buses representing power plants (Bus 1 and 2), transmission (Bus 4 and Bus 5), and distribution substations (Bus 3).

Relays belonging to the same substation (pred (17)) communicate in real-time pilot protection schemes [1] via multicast protocols (for example 61850 GOOSE [14] messages) using the publish–subscribe paradigm over a broadcast medium such as Ethernet (pred (11)). Relays across different substations can communicate if there is a wide area network (WAN) connection via modem lines between the two substations (pred (12)). A WAN network access usually exists between an unmanned substation and a control center for purposes of remote engineering access, monitoring and alarms. Unless otherwise indicated on the power network schematic, we assume that a substation's control center is the power plant with the largest generation connected to it via transmission lines (pred (15)). For instance in Fig. 1 although there exists a transmission line between substations C and D there no WAN connection between the two in the corresponding control network, as none of them serves as a control center. The logical predicates, below, dictate when network access exists between two relays. Note that network access for instance for TCP/IP communication should not be confused with electrical power connections.

$$ethernetlink(r_i, r_j) = r_i, r_j \in R \wedge \exists s \in S [s \in belongsto(r_i) \wedge s \in belongsto(r_j)]; \quad (11)$$

$$modemlink(r_i, r_j) = r_i, r_j \in R \wedge (modem(r_i, r_j) \vee modem(r_j, r_i)) \quad (12)$$

$$netaccess(r_i, r_j) = r_i, r_j \in R \wedge (ethernetlink(r_i, r_j) \vee modemlink(r_i, r_j)); \quad (13)$$

where we have the helper functions:

$$modem(r_i, r_j) = r_i, r_j \in R \wedge \exists s_i \in S [s_i \in belongsto(r_i) \wedge substation(s_i) \wedge \exists s_j \in S [s_j \in controlctr(s_i) \wedge s_j \in belongsto(r_j)]] \quad (14)$$

$$controlctr(s_i) = \{s_j \in S | s_j \in adjacentpplants(s_i) \wedge \forall_{s_i \in S} s_k \in S [s_k \in adjacentpplants(s_i) \wedge (power(s_j) \geq power(s_k))]\}; \quad (15)$$

$$adjacentpplants(s_i) = \{s_j \in S | powerplant(s_j) \wedge \exists b_i \in s_i, \exists b_j \in s_j [linesin(b_i) \cap linesin(b_j) \neq \emptyset]\}; \quad (16)$$

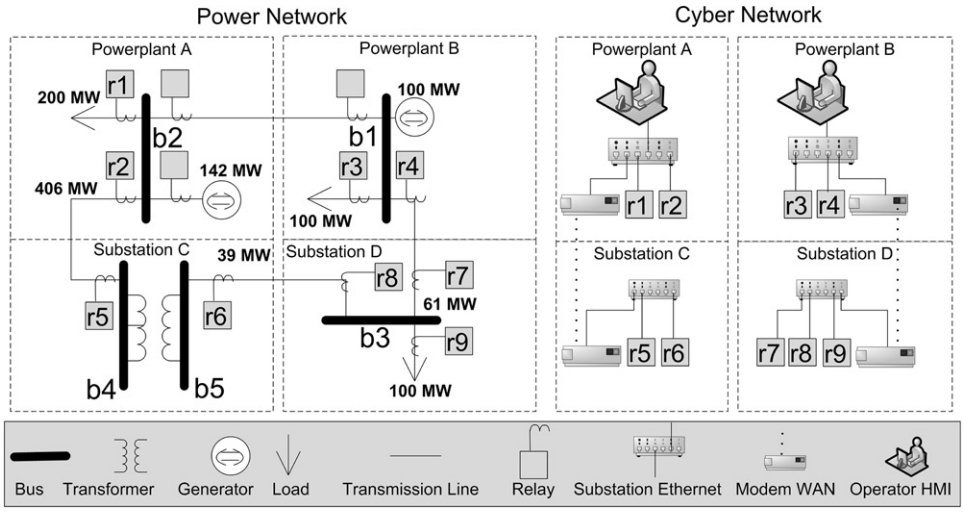


Fig. 1 – A simple One-line Diagram representing a Power Network and its corresponding Control Network.

$$\underset{r_i \in R}{belongsto}(r_i) = \{s_i \in S | \underset{d_i \in D, b_i \in B}{\exists d_i \exists b_i} [(d_i, b_i) \in controls(r_i) \wedge b_i \in s_i]\}. \tag{17}$$

Definition 1. A contingency is a condition where a set of devices D_i are taken out of service during power system operation by misconfiguration of their controlling relays R_i that causes a violation in a set of other devices D_j where $D_i \cup D_j = \emptyset$ i.e. the set of devices D_j exceed their maximum operating limits.

In any electric network, current and voltage are governed by Kirchhoff's current and voltage law and the current flow through the branches is governed by a generalization of Ohm's resistive law. Therefore there is a current limit on each line. If one line is not operational due to some contingency, the current flow will take a different path in the network. This may cause a current in a branch to increase to a dangerous level and might cause a heating and melting of the wires. Similarly if a transformer or a generator is forced to operate beyond its intended capacity, violations may occur causing malfunctioning, for instance burnt insulation and wiring. Since relays can be operated remotely, the ability to cause contingencies offers the malicious adversary an excellent avenue of attack. Function (18) returns the device violations which occur when a set of devices have a contingency.

$$conting(R_i) : R_i \rightarrow D_j \text{ where } R_i \subset R, D_j \subset D. \tag{18}$$

Definition 2. Loss is a metric to estimate the extent of the damage caused by violations and is directly proportional to the product of the cost (per hour) of unmet demand $cost_{umd}$, the load shed (per hour) and the time (per hour) to repair the violations of all n devices under violation:

$$\underset{D' \subset D}{loss}(D') = \sum_{d_i \in D'} power(d_i).cost_{umd}.time_{repair}(d_i); \tag{19}$$

Note that in our definition of *Loss* we do not distinguish between the type of device. The load shed could be due to a bus, line or a transformer. This allows us to apply our model to both transmission as well as distribution networks.

3.2. *Threat model*

We assume a non-global, partial adversary who does not monitor all links. He is limited to one tap which can be put on any substation ethernet and between modem-to-modem links, but not power plants. He is familiar with the power network schematics and knows the exact contingencies that will cause the maximum loss, which is also his objective. He is however limited by a cost that he has to pay every time he compromises a relay needed for a contingency and this cost is deducted from the number of resources he has available. He must have network access to all relays needed to cause contingencies. Finally we assume that he has two kinds of attacks at his disposal- DoS and Masquerade attacks. The former can be protected against by firewalls and traffic segregation and latter via encryption. We assume the attacker has 2 resources and each relay compromise has a cost of 1. Under these assumptions, predicate (21) returns all the pairs of relays possible to attack for an unprotected substation, while predicate (20) determines the contingency relay pair with the maximum loss.

$$\begin{aligned} \underset{s \in S}{maxattack}(s) = \{&(r_k, r_l) \in attack(s) | \forall (r_i, r_j) \in attack(s) \\ &[\exists D_n \in conting(r_k, r_l)[\forall D_m \in conting(r_i, r_j) \\ &[loss(D_n) \geq loss(D_m)]]]\}; \end{aligned} \tag{20}$$

where we have the helper predicate

$$\begin{aligned} \underset{s \in S}{attack}(s) = \{&(r_1, r_2) \in R \times R | netaccess(r_1, r_2) \\ &\wedge \exists (d_1, b_1) \in controls(r_1)[\exists (d_2, b_2) \in controls(r_2) \\ &[((b_1 \in s \wedge b_2 \in s) \\ &\vee (b_1 \in s \wedge b_2 \in controlctr(s)) \\ &\vee (b_1 \in controlctr(s) \wedge b_2 \in s))]]\}; \end{aligned} \tag{21}$$

We will now present a short discussion on why such an adversary model is realistic. Substations are unmanned, with little or no enclosure because of their large size. Modems relay substation messages across large distances mostly

using telephone cables or the public internet infrastructure. Tapping into either one is trivial for a determined attacker. Power plants on the other hand are harder to infiltrate because they are usually manned with physical security in place to protect generators and the fuel. Pilot protection schemes used by relays have stringent demands such as low-latency communication and high susceptibility to replay and error propagation (small blocks of data transmitted in real-time) so traffic manipulation attacks can severely impact system reliability. Once the adversary taps into a substation, he only employs cyber-attacks as opposed to physical because (a) cyber attacks have the potential to cause considerably more damage (as will be evident later in the paper) and (b) they are more subtle. Consider the attention drawn by attempting to damage a transformer using a shotgun as opposed to causing it overload by a simple command sent to a relay.

3.3. *Cyber defense model*

Given the adversary model, we describe the options available to the security engineer to mitigate the damage caused.

Definition 3. A security scheme l_i can be applied to a substation s_j to limit the adversary's network access. Each scheme, once applied to a particular substation, has an associated implementation cost c_{ij} and attack coverage a_{ij}. a_{ij} provides an estimate of the average revenue loss if the substation gets attacked despite the security scheme in place.

The next couple of sections elaborate on the terms used in this definition by describing some schemes formalized from NIST's security best practices and how their costs and attack coverages are determined.

3.3.1. *Intrasubstation traffic segregation via virtual LANs*

Ethernet on its own provides little security from malicious intruders, and segregating it into multiple IP subnets is one approach to narrowing an electronic security parameter. The NIST [3] guide on SCADA security states that:

"VLANs allow switches to enforce security policies and segregate traffic at the ethernet layer, mitigating the risks of broadcast storms that may result from port scanning or worm activity."

The VLAN predicate (22) maps a substation ethernet into n multiple broadcast LAN segments separated by VLAN switches, such that no groups of relays whose contingencies will together cause a violation belong in the same segment.

$$\underset{s\in S}{vlan(s)} \geq \min\{n|\, X_1,\ldots,X_n, \underset{X_i\in\mathbb{P}\{R\}}{\bigcup_{\forall i} X_i = relaysin(s)}, \underset{i\neq j}{X_i\cap X_j=\emptyset} \underset{l\neq k}{\forall r_l \forall r_k \in X_i}\, [conting(r_l,r_k)=\emptyset]\}; \quad (22)$$

where we define the helper function:

$$relaysin(s) = \{r\in R|s\in belongsto(r)\}; \quad (23)$$

Fig. 2 illustrates how a VLAN supporting switch and router combination can be used (right) to replace a simple hub-spoke ethernet configuration (left) to segment the network into multiple broadcast domains such that dependent relay combinations needed to cause violations are isolated.

$$\begin{aligned}\underset{s\in S}{attack_{vlan}}(s) = \{(r_1,r_2)\in attack(s)|\\ \exists(d_1,b_1)\in controls(r_1)[\exists(d_2,b_2)\in controls(r_2)[\\ (ethernetlink(r_1,r_2)\wedge\exists X_i\in vlan(s)[r_1,r_2\in X_i\wedge b_1,b_2\in s])\vee\\ (modemlink(r_1,r_2)\wedge((b_1\in s\wedge b_2\in ctrlcenter(s))\vee\\ (b_2\in s\wedge b_1\in ctrlcenter(s))))]]\}. \end{aligned} \quad (24)$$

As shown in pred (24) the attacker is restricted in the network access he has, and can only compromise devices if they are in the same VLAN as his initial tap or accessible via a modem link.

3.3.2. *Intersubstation traffic segregation via firewalls*

While the VLAN scheme limits the adversary from attacking multiple targets within a substation, it provides little or no protection against attacks traversing multiple connected substations. According to NIST's CIP best security practices rules [4,2] firewalls should be used to segregate traffic between process control networks (PCN), and engineering and monitoring access. Predicate (25) shows that in a firewall protected modem link, the only avenue of attack is the substation ethernet.

$$\begin{aligned}\underset{s\in S}{attack_{firewall}}(s) = \{(r_1,r_2)\in attack(s)|ethernetlink(r_1,r_2)\wedge\\ \exists(d_1,b_1)\in controls(r_1)[\exists(d_2,b_2)\in controls(r_2)[\\ (b_1,b_2\in s)]]\}. \end{aligned} \quad (25)$$

Fig. 3 illustrates how a three-port firewall segregates control traffic into multiple domains; HMI/local generation control, transmission and remote monitoring preventing the attacker from using a compromised modem connection to exploit a cross substation contingency i.e. between relays 1 and 4.

3.3.3. *Intersubstation traffic encryption via link encryption*

Security best-practice guides recommend encrypting traffic when sending control messages over a WAN connection. Since most relays do not support encryption schemes, special devices may be installed next to modems to encrypt all outgoing traffic. A serial encrypting transceiver [15] is an example of such a device which acts as a 'bump in the wire' standalone cryptographic module designed to protect latency-sensitive devices. A logic function to determine where to place an encrypting transceiver would be similar to the one above shown for firewalls except that there would be one placed on either side of a WAN link to perform both encryption and decryption functions. Note that an encryption transceiver does not duplicate the functionality of a firewall — in fact both devices are complementary. This fact becomes clearer in the evaluation section where we show how a composite firewall plus link encryption scheme provides more security then any one scheme used in isolation.

3.4. *Implementation costs and attack coverage of security schemes*

For the purpose of the security analysis, the result of the implementation of the security schemes in a substation

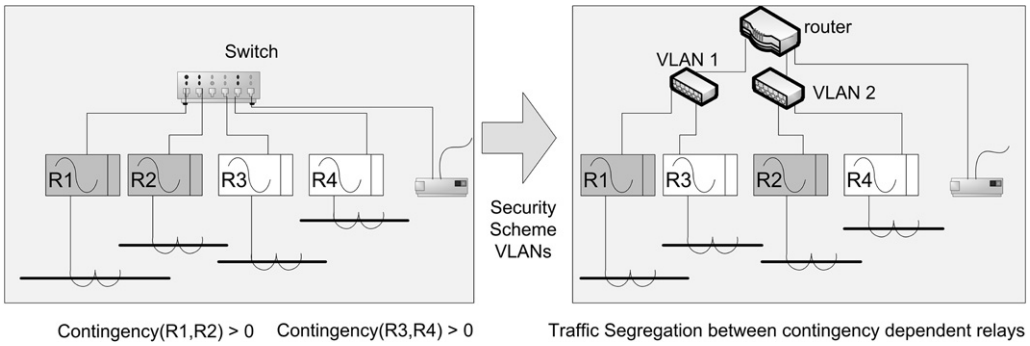


Fig. 2 – Intrasubstation Traffic segregation using VLANs.

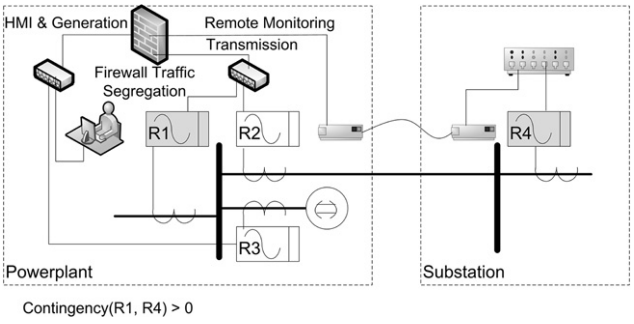


Fig. 3 – Intersubstation Traffic segregation using Firewalls.

is the determination of attack coverage reduction and implementation cost. We determine implementation cost of a scheme to be the sum of the cost of security control devices used in implementing that scheme. For instance the cost of implementing VLANs in a substation s_j would use the predicate (22) to determine the number of switches needed.

For a scheme l_i implemented on substation s_j, predicate (26) gives a set of relay pairs V that are vulnerable to attack because they cause contingencies and $V' = attack_{l_i}(s_j) \cap vuldevpairs(s_j)$ gives the pairs that are not protected by the security scheme.

$$vuldevpairs(s_j) = \{(r_x, r_y) \in attack(s_j) | conting(r_x, r_y) \neq \emptyset\}. \quad (26)$$

Since the adversary will always try to exploit the contingencies in order of greatest to least damage, we can sort the relay pairs $R_1, R_2, \ldots, R_n$ where $R_1, R_2, R_n \subset R$ returned by predicate (26) according to loss damage (predicate (19)) $loss(D_1), loss(D_2), \ldots, loss(D_n)$ where $D_1 = conting(R_1)$, $D_2 = conting(R_2)$, and $D_n = conting(R_n)$. We can associate to every contingency D_i a parameter $\alpha_i \in \mathbb{R}^+_{\leq 1}$ that represents the scheme's inability of preventing the exploiting of the contingency. Then the attack coverage a_{ij} is given by Eq. (27).

$$\begin{aligned} a_{i,j} &= \alpha_1 \cdot loss(D_1) + (1-\alpha_1) \cdot \alpha_2 \cdot loss(D_2) \\ &\quad + (1-\alpha_1) \cdot (1-\alpha_2) \cdot \alpha_3 \cdot loss(D_3) \\ &\quad + \cdots + (1-\alpha_1) \cdots (1-\alpha_{n-1}) \cdot \alpha_n \cdot loss(D_n). \end{aligned} \quad (27)$$

Generally the set of relay pairs V' would have an $\alpha = 1$ indicating that an attack can exploit this contingency successfully every time, while the set $V \setminus V'$, depicting the set of relay pairs that pose an attractive target for the attacker but are protected by a security scheme would have a lower α value. A value of 0 indicates that the scheme completely protects against attacks. The computation of a_{ij} can be considered as a probability that the attacker is able to exploit the given contingency during the attack. The attacker tries to exploit the contingency associated with the maximum coverage. The success of this action is determined by the probability α_1. In case of failure (due to network access denied by a security scheme), the attacker would try to exploit the next contingency with the maximum loss and so on, until one exploitable contingency is found.

3.5. *Optimal security hardening algorithm*

Given a set of substations and a set of independent strategies, each with its unique implementation cost and coverage against malicious attacks, the budget problem is to search for the optimal combination of strategies to apply at each individual substation so as to maximize the overall network security while remaining within a fixed budget.

Assuming that only one strategy can be applied at each substation, it is easy to see that enumerating all possibilities is an NP-hard problem. The proof lies in a straightforward reduction from the Multiple-Choice 0-1 Knapsack problem (MCKS). The goal of a general 0-1 Knapsack problem is to select a set of objects, each with an associated weight and a revenue, such that the sum of the weight is below a predefined bound and the total revenue is maximized. In the Multiple - Choice variation of this problem, the objects are partitioned into n groups, and only a single object can be chosen from each group. This problem can be reduced to our formulation by considering each object to be a different security scheme and each substation to be one of the groups in which the objects are partitioned. The weight of each object becomes the cost of the implementation of the security scheme and the revenue is the opposite of the attack coverage (i.e., such that the maximization of the revenue can be expressed as a minimization of the attack coverage). By selecting the security schemes that minimize the attack coverage under a specified budget, we are solving the general MCKS problem.

The formulation of the optimal security hardening problem is defined as follows. Given the schemes $l_1, \ldots, l_m$ and given a set of substations $s_1, \ldots, s_n$, we define a variable

$x_{ij} \in \{0, 1\}$ to be equal to 1 if the security scheme l_i is applied to the substation s_j, 0 otherwise. Each security scheme l_i, when applied to a substation s_j, has an associated implementation cost c_{ij} and an associated attack coverage a_{ij}. The budget allocated for security hardening is expressed as *budget*. The problem can be expressed as in Eq. (28).

$$\begin{aligned} \min \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij} \cdot x_{ij} \\ & \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} \cdot x_{ij} \leq budget \\ \text{For } x_{ij} \in \{0,1\} \text{ where } \forall j \quad & \sum_{i=1}^{m} x_{ij} = 1. \end{aligned} \tag{28}$$

However, by assuming that the *principal of optimality* holds (i.e. the optimal security strategy decided at a particular substation only depends upon the budget spent so far and is independent of all previous substations), a recursive dynamic programming solution can be formulated.

We divide the recursive solution into multiple states x_j where j denotes the substation number currently under consideration in that state and x is the remaining budget. Similarly $a(l_j)$ denotes the attack coverage for strategy l_j, and $c(l_j)$ the corresponding cost. If $f_j(x_j)$ is final attack coverage for the state x_j then we have the dynamic programming solution given by the recurrence relations in Eq. (29).

$$\begin{aligned} f_1(x) &= \min_{l_1 : c(l_1) \leq x_1} \{a(l_1)\} \\ f_j(x_j) &= \min_{l_j : c(l_j) \leq x_j} \{a(l_j) + f_{j-1}(x_j - c(l_j))\} \quad \text{for } j > 1. \end{aligned} \tag{29}$$

The first equation in (29) is the base case for one substation which returns the security scheme with the minimum attack coverage whose cost is below the budget. The second equation depicts the forward recursion returning the minimum $a(l_j)$ of the current state plus the minimum of the last state. It is easy to see that $f_j(x_j)$ can be stored in a table (or *memorized* in logical programming, as will become apparent in our Prolog implementation later) preventing a state explosion and allowing a polynomial time evaluation.

4. The tool-chain architecture and its implementation

Fig. 4 shows a high level architectural diagram of the tool-chain detailing how the various components sit with respect to each other. We give a detailed description of each of the various modules.

4.1. *Parsing specification files*

The logical model of the power network is auto-generated from specifications written in the standard descriptive language based on Common Information Models (CIM) [16] with the help of a parser tool and stored in a Prolog database. CIM is an object-oriented cyber infrastructure modeling language proposed by the Electric Power Research Institute (EPRI) and represents all major objects normally used within

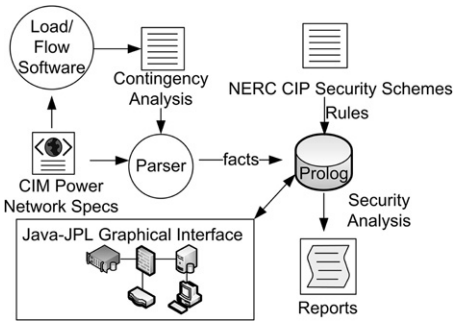


Fig. 4 – High-level architectural diagram of the security assessment tool-kit.

Table 1 – Power network models described as prolog facts.

```
1
2  % device(ID,TYPE,SUBSTATION,POWER,OPERATE_LIMIT,COORDX,COORDY)
3  device(bus2,bus,[a],142,_,10,20).
4  device(bus4,bus,[c],142,_,10,50).
5
6  device(load2,load,[a],200,_,30,40).
7  device(line3,line,[a,c],406,500,_,_).
8
9  % bydirectionlink(SRC,DEST)
10 connected(load2,bus2).
11 connected(line3,bus2).
12 connected(line3,bus4).
```

an electric utility enterprise. The objects are represented as classes having attributes and relations to other classes. Included objects bundled in packages cover equipment, topology, load data, generation profiles, measurement and scheduling. The CIM RDF schema is documented as the IEC standard 61970-501 and is self-describing because it is based on XML. We create a mapping of the classes in the RDF model to entities in our security model. The parser identifies the main entities such as devices and connectivity and then proceeds to populate the attributes of the entities such as power and voltage limits. The attributes can be easily populated by looking at the properties and associations for each object in the CIM model.

4.2. *Implementation in predicate calculus*

We implemented our predicate calculus security model as a form of Horn Clause logic in Prolog using SWI-Prolog version 5.6. The various devices, connectivity and their properties identified by the parser were asserted as 'facts' and their 'attributes' in the Prolog knowledge base. Table 1 shows how Prolog facts describe buses, loads and lines and their interconnections. Prolog facts can be thought of as relational tables for example device IDs serve as foreign keys in the connected entities and primary keys in the device entities. The connected predicate shows a bidirectional link between two devices.

4.3. *Contingency analysis*

Various power-flow simulation software both commercial e.g. Powerworld [17] and open source e.g. InterPSS [18]

exist that allow contingency analysis of power networks. A contingency analysis via a power-flow simulation software will take out of service each device, one-by-one, resolve the power flow, and then check that no *violations* have occurred e.g. no lines have exceeded their rated capacity. Industry planning and operating criteria often refer to the $n-1$ rule, which holds that a system must operate in a stable and secure manner following any single transmission or generation outage. We scripted the Powerworld software to do an $n-1$ and an $n-2$ contingency analysis on each substation and its associated control center in a power network schematic essentially returning the tuples $(contingentdevices, violations)$ to be stored in the Prolog knowledge base. Predicate (18) essentially searches through this table for its solution.

4.4. *Security analysis implementation as logical rules*

This section describes how the various logic predicates are implemented as Prolog rules using two representative examples that of VLAN security scheme selection (Eq. (22)) and the max-security fixed budget optimization problem (Eq. (29)).

Algorithm 1 Greedy algorithm for assigning VLANs to devices

```
/* At the start, all nodes do not have a label associated with
them */
l(n) = -1, ∀n
/* Start the algorithm by assigning label = 1 */
c = 1
Order nodes in non-increasing degree order
/* until all nodes have labels */
while ∃n : l(n) == −1 do
  for each node n do
    if l(n) == −1 and ∀ neighbor nodes j, l(j) ≠ c then
      Assign c to n
    end if
  end for
  c = c + 1
end while
```

4.4.1. *Device to VLAN assignment*

The cost of implementation of this scheme varies depending on the number of VLANs used to secure the substation: high number of VLANs requires more equipment and higher setup costs. For this reason, it is necessary to provide an estimate of the number of VLANs needed. In order to minimize the number of VLANs, the scheme proposes to segregate only devices that, if exploited together, can create a violation. This problem can be mapped into a graph coloring problem. In this model, each device is a node in the graph and edges are created according to the contingency analysis: if two devices can be used together to create a violation, then an edge exists between the them. The goal of this problem is to find the minimum number of labels (i.e., VLANs) that we need to assign to each node such that two connected nodes do not share the same label. For determining the solution to this problem, we used a greedy algorithm for graph coloring, shown in Algorithm 1.

4.4.2. *Optimal selection of security schemes subject to budget constraint*

Table 2 describes part of the implementation of the optimal security scheme selection algorithm described previously. Note that some of the predicates and attributes have been taken out for brevity. We describe the listing bottom up. Line 29 is the base case and line 34 the recursive case of Eq. (29). They take as arguments the state J and the total budget constaint X_j and return the overall Loss in *Result* and a list representing the schemes applied at each substation. The *mklist* rule is called (line 37) in the recursive case to merge the list of current attack coverages with that of the most optimal results in the last state subject to the budget constraint. Lines 8 and 14 detail the implementation of the *mklist* rule, the former dealing with the boundary condition of the budget running out while the latter deals with the regular case. Once the attack coverages list has been completely processed the *mklist* predicate on line 8 unifies the under process lists *Acc* and *Acc3* with the result *MergeAj* and *SLst*. Note the memorization function used in line 16 and declared on line 2 that allows computed solutions to be stored in a look-up table enabling the dynamic programming optimization.

4.5. *Graphical control user interface*

In order to allow the tool to be used by security analysts and have easy and fast use we added a Java based User Interface front-end to the Prolog engine. JPL is a set of Java classes and C functions providing an interface between Java and Prolog by the embedding of a Prolog engine within the Java VM. By annotating each device in our CIM specification with x and y coordinates we can easily import and display the SCADA network in a Java grid panel. The advantage of this approach over static network visualization tools is that it allows more user interaction. For instance a security engineer can hover a mouse over a device icon to see a detailed listing of its security schemes or point and click on devices of interest and formulate a query.

5. Evaluation and results

The IEEE 118 Bus test case was used to test and analyze the optimal security hardening problem formulation. Data for the IEEE 118 bus test case representing a portion of the American Electric Power System in the Midwestern US, was downloaded from the University of Washington Power System [19]. The system consists of 118 buses, 186 transmission elements, 19 committed generators with a total capacity of 5,859 MW, and 99 load buses with a total load of 4,519 MW. The complete power flow simulation along with the line limits that we used can be found at the Powerworld website [20]. Since our threat model assumes the adversary has two resources, we did a N-1 and an N-2 contingency analysis. While our tests were run on the entire network whose results are presented at the end, we initially walk the reader through just a portion of the analysis (south-west part of the network) shown in Fig. 5 for easy understanding.

Table 2 – Optimal scheme selection: Prolog implementation.

Fig. 5 – Contingency analysis of a portion of the 118-bus test case.

5.1. *Security schemes employed*

Suitable candidate devices were picked for each of the security strategies identified. We use the label *VLAN* to indicate the intra-substation traffic segregation via Virtual LANs scheme, the label *FWALL* to indicate the intersubstation traffic segregation via firewall scheme, and *FLINK* for the composite scheme of firewall and link encryption combined. Table 3 gives a descriptive summary of each of security schemes applied along with its coverage and an estimate of individual device cost required to implement each one. In the case of composite strategies, the devices shared among the two security solutions are counted only once. For example, in the case of the *VLAN* + *FWALL* strategy, the CISCO 1760 router can be used, at the same time, for implementing both the *VLAN* and the *FWALL* strategies. Hence, its cost is counted only once.

Once the security strategies have been defined, the first step of the analysis is determining the attack coverages and overall loss for each substation. This process is split into two parts. In the first part of the analysis each contingency is analyzed to determine the degree of protection that a given security scheme provides. During this process, each security scheme associates a value of α to each contingency. In the second phase of the analysis, the substation attack coverage is computed using Eq. (27).

5.2. *Case study results*

Table 4 (a), (b), (c) and (d) show the results of the analysis. In each table, the devices that are part of each contingency are shown on the first column, *Contingency*. The second column, *Loss*, reports an estimation of the loss caused by an exploitation of the associated contingency, as defined in Section 3. In this subset of the power grid, all violations affect lines. In the estimation of the loss we assume an unmet demand cost of $1000 per h and a time to repair of 10 h for all violations. Different choices of coefficients