The Pennsylvania State University

The Graduate School

Department of Computer Science and Engineering

VULNERABILITIES IN ADVANCED METERING INFRASTRUCTURE

A Thesis in

Computer Science and Engineering

by

Dmitry Podkuiko

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2012

The thesis of Dmitry Podkuiko was read and approved* by the following:

Patrick McDaniel
Associate Professor of Computer Science and Engineering
Thesis Adviser

Trent Jaeger
Associate Professor of Computer Science and Engineering

Lee Coraor
Associate Professor of Computer Science and Engineering
Director of Academic Affairs

[0]*Signatures on file in the Graduate School.

Abstract

Smart grid has become a reality in the United States. Billions of dollars are being poured into deploying a major component, - the Advanced Metering Infrastructure, which involves replacing old electromechanical electricity meters with more powerful `smart` meters. The smart meters are frequently enabled with powerful features, such as remote disconnect for non-paying customers. Millions are already deployed while serious and preventable security issues are present in these systems. Smart meter vulnerabilities enable new ways to commit energy fraud, perform large scale attacks to cripple power supply to consumers, and are hard to address across many versions of AMI solutions. Manufacturers appear to be failing to heed past lessons of security learned in the computer industry and require coherent effort to validate multiple AMI solutions for security. Developing attack trees to guide penetration testing efforts achieves a comprehensive view of vulnerabilities in smart meters, understanding of the causes, and assists in implementing countermeasures. In this work, attack tree methodology is used to obtain a global understanding of security vulnerabilities through evaluation of two currently deployed systems using reverse engineering and penetration testing to create a re-usable body of knowledge. Finally, countermeasures and recommendations for deployment of similar systems are suggested.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Smart grid installations are growing rapidly in the United States and other countries powered by demand for a more efficient way to produce and distribute energy. United States alone allocated over 4 billion dollars from the recent economic stimulus package [22] to development and deployment of smart grid technologies. A large componet of smart grid rollout is replacing existing electromechanical meters with `smart meters`. Smart meters can perform functions such as remote energy usage reading, reporting fraudulent activity, and even disconnecting non-paying customers from the grid. However, these powerful features bring new security concerns with them. Smart meters are good at detecting previously know methods of fraud, but security is lacking at addressing new challenges [39, 23, 40, 38]. Proofing smart meters against attacks requires significant penetration testing effort, which is complicated by the fact that multiple vendors are offering smart metering solutions in the current market.

Many aspects of smart meter vendor systems are similar from one solution to another, but each system has to be verified and vetted through penetration testing exersizes due to architectural differences. The process of penetration testing must be repeated for every iteration of the system because upgrades may introduce additional security issues into the system besides ensuring that old issues are solved. Penetration

testing and reverse engineering are two areas that are generally difficult to organize and generalize. This is especially true for efforts that span multiple vendor systems.

This work will show that developing attack trees and using them to guide penetration testing effort creates a re-usable body of knowledge, applicable across many vendor systems, and provides a global understanding of the vulnerabilities and application of countermeasure. This process is broken down into the following steps:

1. Capture architectural description of the system
2. Design and instantiate attack trees
3. Discover vulnerabilities using designed attack trees as a guide
4. Implement countermeasures
5. Repeat for every iteration of the system or a different vendor

Two currently deployed systems are explored using this process for issues such as protocol implementation flaws, configuration errors, new ways to commit energy fraud enabled by powerful functionality of smart meter communication interfaces. Finally, discovered issues are discussed, including their causes and origins in past historical examples, and countermeasure application.

1.1 Background

1.1.1 What is smart grid?

Electricity generation and distribution has not experienced a significant overhaul in more than 100 years, while energy resources are becoming more expensive. U.S.

electrical grid is still very centralized and inefficient in many ways. Smart grid is an ambitious new effort to enable distributed electricity generation, more dynamic pricing, and efficiency.

One point of attention in electricity generation has been the peak electricity demand times when additional generation capacity, such as coal burning plants and gas generators, has to be brought online. This is said to cost utilities a significant proportion of operating costs and any reduction of a few percent may provide multibillion dollar savings [21]. There are several opportune ways to address this inefficiency. Many household appliances, such as dishwashers and heaters, abundant in our homes use the most electricity. If these appliances could be powered on during off-peak demand time, it would flatten demand and keep the amount of generation capacity constant.

Economics of energy pricing are also a significant part of smart grid effort. Utilities need to recover energy resource costs and this is even more true as resources such as oil become more expensive and scarce to avoid exposure in energy crisis periods [29, 6]. Changing consumer energy prices dynamically would allow utilities to react to energy market swings. Feedback to consumers about these energy price changes would also help motivate consumers to behave differently and make necessary adjustments to contain their expenses. This scheme is called TOU (Time-of-Use) pricing [30]. For example, consumers could run their most demanding appliances during the off-peak demand times when prices are lowest and make strides toward purchasing appliances with lower energy consumption to reduce their expenses. Another pricing scheme focuses around a maximum electricity load value recorded within the monthly intervals. Such load value would act as a multiplier on monthly total usage resulting in additional cost to the consumer

on their final total bill, motivating them to smooth their electricity demand. This type of response to changing prices is called DSR (Demand Side Response). Low cost of solar panel technology and wind-based generation are making it possible for households to generate enough electricity for own consumption and to feed surplus back into the grid. Current electro-mechanical meters are poorly equipped to account for this type of activity. For example, giving consumers sufficient credit for electricity returned into the grid is an important factor of distributed generation.

At the forefront of smart grid effort lies the challenge of replacing millions of installed electricity meters. Majority of current electricity meters are electro-mechanical and employ moving parts activated by flowing current to keep track of electricity use and other metrics. Electro-mechanical meters are also prone to various forms of fraud. One prevalent example is meter inversion to rewind energy usage recorded. Utilities and consumers need an advanced set of features from electricity meters to enable smart grid features. The industry produced smart metering solutions that cover everything from the meter itself, communication to utility, and other management infrastructure.

1.1.2 What is AMI and AMR?

Automatic Meter Reading (AMR) is a technology for automatic reading of information from electricity meters. Such information may include meter readings and power line status information. This technology has been developed in 1970's [13]. Over time, the features provided by AMR and surrounding technologies have grown into Advanced Metering Infrastructure (AMI) that governs more than just communication to electricity meters for information. AMI incorporates everything from the sensor network to

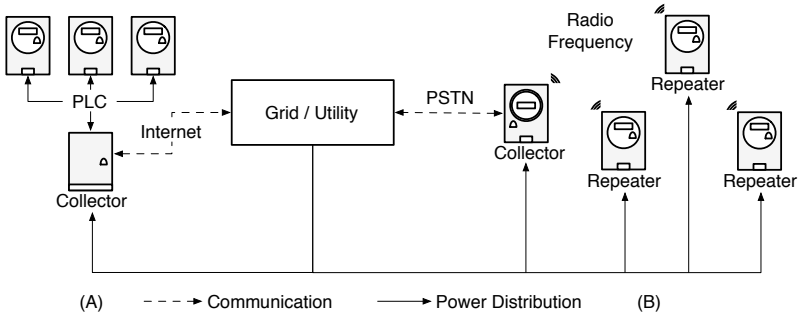

Fig. 1.1 Two example AMI network configurations. In (A) a wired local network connects meters and a dedicated collector node. The collector communicates with the utility over the Internet. In (B) meters act as radio frequency repeaters to a collector node which itself functions as a meter. A backhaul link to the public switched telephone network connects the collector and utility [39].

software that manages collected meter data and its analysis and dissemination back to consumers.

AMI network configurations use multiple network topologies as shown in figure 1.1. Wireless mesh network or star network based topologies are used to connect meters and data collectors/concentrators. Communication from utility to smart meters may occur via ISM band radio, mobile networking units, or telephone network. *Signalling* is any action that can be performed by a utility to smart meter. This includes reading energy usage, remotely disconnecting electricity supply, servicing, troubleshooting, and upgrading firmware.

1.1.3 What is a smart meter?

A solid state smart meter has no moving parts and uses a current transformer to record energy use, power outage information, and voltage sags. All of the information collected can be communicated back to the utility allowing it to judge current state of the system, perform billing functions and tamper activity detection, record and read

peak demand information, and respond to power outages quickly. Generally, a smart meter has a microcontroller (MCU), flash storage, analog-to-digital converters for current transformers, and one or more optional boards such as a wireless card, and most importantly - remote disconnect module. The number of personnel involved in meter reading and maintenance can be significantly reduced, eliminating contetious labor relations, such as with meter reader unions. Remote disconnect promises simpler customer past due account management for utilities and tremendous savings on labor costs used on sending out personnel to disconnect a meter.

Remote disconnect module is similar to the remote car engine kill switches, which are used as antitheft technology, and is essentially a relay that reacts upon commands issued by the utility. Combining powerful functionality of remote disconnect and wireless communication presents a number of consumer and utility security and safety issues.

Remote disconnect is controversial from a safety perspective due to a possibility that a consumer may not be prepared for sudden power restoration. Some manufacturers have developed a special button on the front face of smart meters that must be pressed by a consumer once the power has been restored. Security aspects of remote disconnect technology are conflicting with utilities need for easier customer account management and reduction of labor and service costs. Currently, utilities have to send service personnel to disconnect or service an account, but with remote disconnect technology and remote management this overhead is eliminated. Smart meters are also enabled for prepaid account use and have a digital cash equivalent [13, 12]. This varies on a country by country basis depending on fraud rates, historical, and cultural reasons.

1.1.4 Smart meter security

Smart meters are a hot commodity right now. Many competing solutions are fighting for market share. Large scale deployments are announced by multiple utilities fueled by federal stimulus funds [22]. Security appears to be lacking on many of the solutions due to various reasons. Some solutions have been found to make security mistakes in product design that are well known in the computer industry for many years. Manufacturers did a good job focusing on past attacks on electricity meters, such as meter inversion, meter bypass, and magnetic interference, but when it comes to protocol implementation and embedded systems design for security there are lapses. Smart meters have little in terms of physical security and have ineffective tamper protections [39], such as seals, which are directed at stopping adversarial activity psychologically. A stolen or auction site purchased smart meter can be analyzed by an adversary at their leisure.

Bringing computer-like functionality to an electricity meter without incorporating security by design is a dangerous proposition. Electricity meters are installed in the numbers that can reach millions and all of these meters are identical to one another in hardware and software to some degree on per utility basis. Such an installation is highly homogenous. A single vulnerability exposes entire smart meter installation and helps the attacks scale to cover large areas like cities and regions. This has already been demonstrated with smart meter worm propagation [23], which shows that powerful feature exploitation on a massive scale is possible. Another troublesome prospect in the future is that utilities won't keep up with upgrading smart meter firmware and systems against security vulnerabilities. This is possible due to potential fragmentation in device

homogeniety as new devices enter the market or become unavailable due to product obsolesence.

Remote disconnect feature delivers significant value and at the same time the most risk in a smart meter. Properly timed attack on the remote disconnect feature, such as one during extreme temperatures of summer and winter, has the potential to cause real human casualties. In 2003, a heatwave in France caused over 14,000 deaths [9]. An attack during similar period would surely cut off access to air conditioning for most vulnerable groups of population. Refrigeration for food supply chain will also be impacted if the attack effects last past supplemental generator or other disaster recovery methods' capacity. Critical installations of important civilian and military use simply cannot be denied power supply as they serve as a backbone of the country. Another example is a utility being locked out of its own smart meter network costing precious time and resources to recover and further exacerbating damaging effects when coupled with such an attack as mass remote disconnect.

1.1.5 Efforts to define security for smart grid

There are efforts to define security at the governmental and industry levels. For example, The Guidelines for Smart Grid Cyber Security report was produced by National Institute of Standards and Technology in 2010 [27]. This is an important effort, but it will require manufacturer and utilities to fully subscribe to its goals. Validating available and forthcoming AMI systems for security represents a significant challenge

due to the number of vendor solutions, software and hardware upgrades to these solutions, and market dynamics. Efforts to validate security through reverse engineering and penetration testing can significantly benefit from an organized approach to these tasks.

1.2 Related work

Smart meter security has many related research areas that provide a foundation for this work, our results, and proposed solutions. These areas are SCADA (Supervisory Control and Data Acquisition), embedded systems, wireless sensor networks, and others. Deployment of AMI is so vast that the system scale encompasses all areas of computer security.

1.2.1 Smart meter as an embedded system

Smart meters are essentially embedded devices. They have limited, but yet feature-rich capabilities in part due to lower cost of components such as MCUs. Similarly, since prices of components continue to decrease, embedded devices increasingly integrate communication functionality, such as Ethernet and wireless technologies. Embedded devices are no longer isolated from the Internet, which provides plenty of exposure to attacks. Embedded devices are generally designed for a specific application forcing engineers to design solution under significant constraints of power consumption requirements, bill-of-materials limitations, reliability and standards compliance, and time-to-market deadlines.

All of the constraints mentioned have an impact on security of the overall solution and history has many examples of some security failures that result from this. Embedded

system engineers have to strip any extraneous software functionality or have to reinvent or re-implement software functionality for a specific resource-constrained application targeted by the system. Pacemakers and defibrillators are implantable medical embedded devices designed to significantly improve quality of life for patients with heart problems. Kohno et. al. [32] have shown that pacemakers could be reprogrammed to cause harm. Other examples of embedded system security failures include voting machines [15, 49].

Testing and debugging functionality of smart meters has been a major source of security vulnerabilities for smart meters, and many other types of embedded devices. Engineers frequently leave this functionality intact on production circuit boards for the reasons of needing to perform debugging on a system in the field or simplicity of having one design. This functionality is sometimes trivial to exploit as was demonstrated by this work and many other areas [2]. It is customary for embedded systems to undergo various forms testing for ISO or governmental certifications. Obtaining certification and performing testing to obtain certification for the embedded device is a time consuming, expensive process and it serves as another major reason why testing and debugging functionality remains on PCBs (Printed Circuit Boards). Some MCUs provide debugging mode lock functionality through a lock bit, but even when enabled it can be bypassed by complete overwrite of the MCU flash [7].

Particularly problematic areas for embedded devices are communication protocol design and security. Past examples of protocol security problems are well demonstrated by TCP/IP protocol [14]. Protocol design is difficult for the same reasons described in [42] and this also applies to smart meter system security in general.

1.2.2 SCADA security

SCADA systems is another area closely related to smart grid. Multiple points within smart grid perform SCADA-like functionality. [31] have demonstrated possibility of cascading power failures if a single substation is lost due to demand. Attacker controlling large swaths of smart meters may be able to forge demand readings to the utility facilitating such attack [37]. The most recent example of SCADA systems being targeted is the so called `stuxnet` worm, which reportedly had severe consequences for Iranian uranium enrichment facilities. `stuxnet` took an indirect approach to attacking multiple systems and spreading much like a botnet through multiple vectors. Smart meter systems must be protected from a similar attack on utility control servers. Security of these servers may be defeated in a indirect way, similar to an insider attack. Many SCADA systems themselves have been shown to have multiple vulnerabilities and represent opportune targets as was most recently demonstrated by releases of vulnerability exploiting source code in multiple software packages [50].

Smart meters and related control systems, like other embedded devices, can also be discovered through brute force scanning of the Internet. Most devices can be recognized from the specifics of software communication stacks, hardware, and operating system artifacts as simple as an availability of certain ports on the system. Nmap tool [5] covers a pretty extensive range of these artifacts and can automatically identify a particular version of a system, thereby simplifying targeting process. [18] demonstrated the ability to discover poorly configured devices online, which can be used to search for AMI related systems, which are left exposed through misconfiguration and other reasons.

Another important area of concern are attacks on AMI installations by insiders. This work provides vulnerabilties that can be used in such scenarios to a tremendous effect. Disgruntled employees often have access and expert knowledge to bypass security mechanisms and alarm systems and there are past examples of this happening in industrial settings. An insider unleashed an attack on systems controlling sewage release, while bypassing alarm systems [10, 46, 45]. Security of smart meter installation must be comprehensive to cover such attacks and previously mentioned indirect vectors. Separation of duty principle has been used in practice to prevent such incidents, although not without failures [11]. Practical examples include requiring multiple operators for launch of strategic nuclear ordnances and splitting of credentials between bank ATM maintenance personnel. Similar separation of duties for such tasks as key managment are a necessity for the system.

1.2.3 Smart meter as a wireless sensor node

Wireless sensor networks is a fundamental background technology for smart meter networks. Smart meters utilize protocols designed for fault tolerance and automatic network initialization, which sometimes is described as a mesh network. One area particularly relevant to sensor networks is key management. [20] and [48] describe protocols for automatic secure deployment of sensor nodes and their initialization. Some of the discovered issues during this work are akin to widely publicized password database compromises of popular websites [44, 33] and most infamous UNIX `passwd` issues [24], which help the attack to scale by virtue of password knowledge and applicability to a neighborhood of devices.

At this time, smart meters do not offer robust mechanisms to attest for secure communication and other functionality. Simple checksums are used at various points in the system to confirm firmware validity and proper transmission of packets. Attested meter functionality is demonstrated in [35], which aims to improve this situation. Vast deployments of smart meters can generate large amount of logged event data. This event data may include tampering, meter resets, cover removed, reverse energy flow and magnetic tampering. Responding to such high volume of events is difficult. Remote attestation is a better choice since logging data could be forged or erased as was demonstrated by vulnerabilities uncovered in this work [39, 40].

1.2.4 Privacy and smart meters

Smart meters are opportune targets for invasion of privacy [38]. Electricity usage data contains a wealth of information and could be mined using statistical models to determine whether a target residence is occupied and what appliances are used inside [36, 34]. One example of such mining of electricity usage data was in France, where it was used to uncover energy consumer's with inefficient heating appliances [28] opening possibilites for penalties and other actions by utilities. Consumers in general can be at risk of such activity by a utility with good intentions or not, but when armed with a remote disconnect the utility can exercise it to its advantage [13, 12]. Similarly, other wireless sensors inside our homes could be used to gain such information [47].

Chapter 2

System Description and Setup

2.1 AMI Systems

In this work, two AMI systems were used during penetration testing. Both systems are currently used in the field and were setup in a similar way to real deployments. The two systems differ in aspects such as networking topology.

2.2 Testbed 1

To simulate a real AMI installation with communication from home office at a utility down to smart meters in the field via backhaul network the following setup was built 2.1. Asterisk PBX (Private Branch Exchange) [41] was used to set up an environment with utility communicating to smart meter data concentrators and vice versa via PSTN (public switched telephone network). A data collector is a special type of a smart meter that contains additional functionality. It manages a group of smart meters via wireless mesh network and communicates directly with the utility via modem option board. The system is enabled with intrusion detection on modem line and logging functionality. When a device goes off the hook, it receives a dial tone and voltage via an onboard component called the Foreign Exchange Office (FXO). All endpoint devices in a telephone network use an FXO. The dial tone and voltage are supplied from the

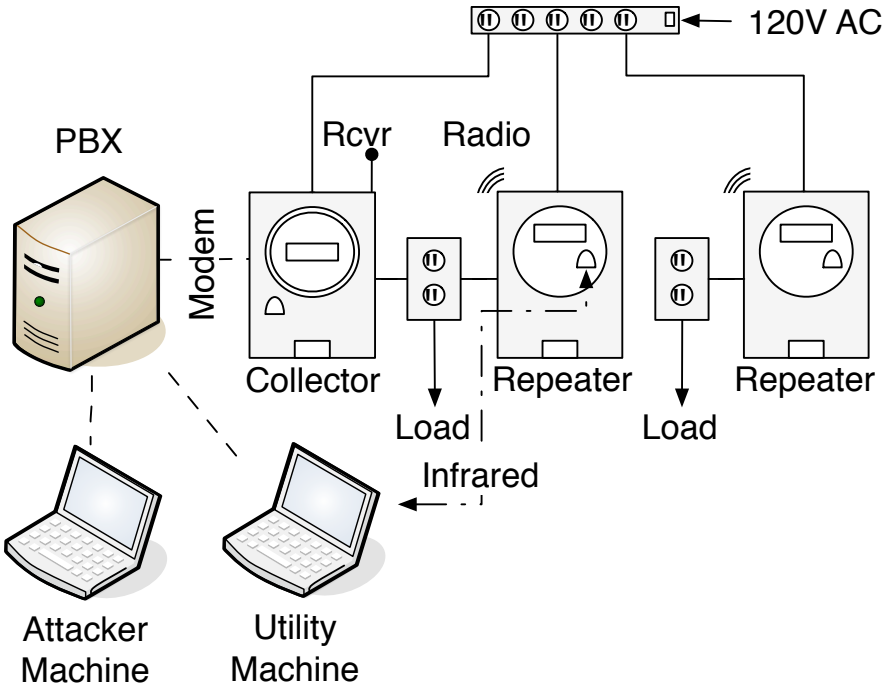

Fig. 2.1 Experimental testbed 1. This system utilizes wireless networking and a PSTN backhaul [40].

other end of the line by the Foreign Exchange Service (FXS), usually implemented by the phone company [39, 40].

A laptop with utility management software was used to configure and establish communication with the data collector via modem. Communication sessions were recorded using monitoring software to recover protocol specifics and packet structure. This capture of communication sessions coupled with ANSI C12.21 and C12.19 standards allowed to obtain operational details and understanding of the system. Several smart meter nodes and collectors were eventually dissambled to allow extraction of ROM firmware.

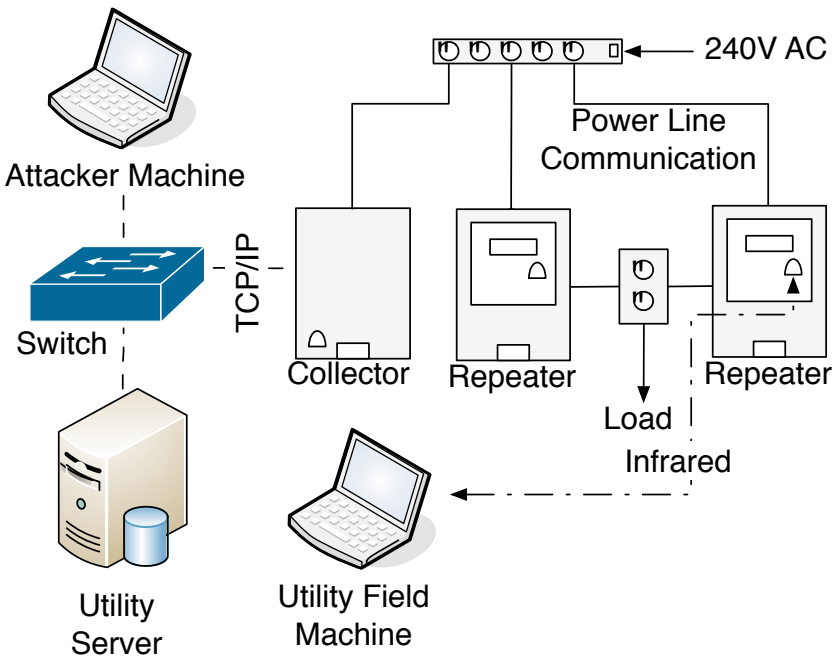

Fig. 2.2 Experimental testbed 2. This testbed utilizes wired network for communication. [40].

2.3 Testbed 2

The second system is different in networking topology 2.2. It uses wired local network in a star configuration. Smart meter nodes are connected to a gateway node, which manages a large number of them as well as serves as a backup for other gateways. The gateway itself is generally installed at the neighborhood transformer level. Collector connects to the utility via backhaul network. In this case, data concentrator came with a standard Ethernet networking option board, which allowed to configure a local network. The utility management software is installed on a server running Windows system. Utility provides a web service API (Application Programmable Interface) via SOAP (Simple Object Access Protocol) to communicate with the data concentrator.

Source code of the management software was provided for analysis of the system, which allowed to reconstruct signalling calls to the data concentrator by placing applications in debug mode and stepping through execution. Having access to source code significantly speeds up the process of penetration testing. Black box penetration testing is significantly harder because underlying functionality is unknown.

Tools such as Wireshark [8] were used to intercept and log communication sessions. Python was used to process communication logs and create scripts to communicate with the meter and utility software. `nmap` tool was used to fingerprint the underlying OS.

Chapter 3

Methodology

The section below introduces attack trees, which are used to capture the architectural description and key knowledge about the systems, and the tools and techniques used by this work during destructive research process to find system vunerabilities.

3.1 Penetration testing and reverse engineering

Penetration testing and reverse engineering are two areas that are synonymous with destructive research. Destructive research is an important methodology and forensic technique that aims to obtain understanding of why a system could be broken by design or discover problem areas missed during design. One example is decoding reasons behind air plane crashes by Federal Aviation Administration (FAA) to track down machine and human factors involved and prevent future disasters.

Destructive research is time intensive and has to be repeated for multiple product iterations due to updates to hardware and software. Any change to a system requires validation that the system wasn't left vulnerable. Peforming such destructive research on multiple systems designed for a particular application also involves similar techniques and knowledge that can be applied to all systems to a large extent. These reasons point to the need for a process that creates re-usable knowledge and communicates value to those who are performing these tests. Figure 3.1 captures the steps of the process. The key

component of the process is construction of attack trees, but the overall outcome of their use is a continous re-use of knowledge captured from previous iterations of the process to set up ongoing penetration testing efforts and countermeasure solution application to discovered vulnerabilities.

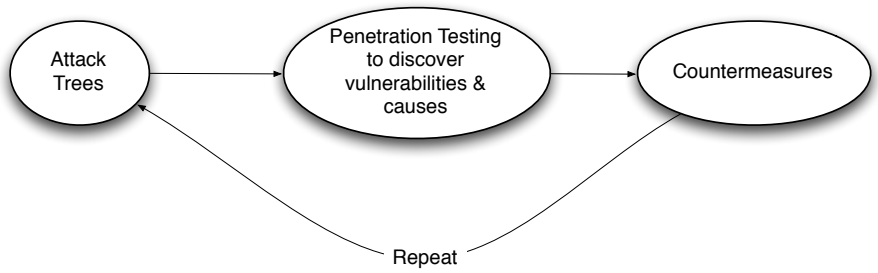

Fig. 3.1 Attack trees help capture the knowledge and facilitate any regression of penetration testing efforts in the future

3.2 Attack trees

Penetration testing of systems is a task that is difficult to organize. Many useful aspects of a system under test are being looked at, but it is difficult to construct and communicate overall importance of discovered vulnerabilities as a global understanding of system's security. Attack trees have been proposed by Bruce Shneier [43] to aid in developing such global understanding. An attack tree is a structure for enumerating the kinds of attacks that achieve a particular adversarial goal. It does this by recursively breaking down a goal into finer- and finer-grained subgoals and finally to a set of attacks that achieve the original goal. An example attack tree that formed the genesis of this work [39] is shown in 4.1. The root specifies the end goal, committing energy fraud by

forging the energy usage information reported to the utility. The internal nodes (those with parents and children) describe the different combinations of conditions that must be met to commit fraud. Finally, the leaves of the tree are the attacks necessary for energy fraud. The final attribute of the tree is the conjunctions (AND/OR) between each layer of child nodes. These specify whether all or just one of the child branches must be followed to reach the goal in the parent node. Attack trees are split into the following categories - archtypal trees and concrete trees. The definitions are repeated from prior work [40] to facilitate understanding.

3.2.1 Archtypal trees

Archetypal trees that describe attacks in a broad sense that is applicable to any system within the architecture. An archetypal tree is an attack tree that is general enough to be applicable to all systems of a given architecture. As with a regular attack tree, the root of an archetypal tree is a single adversarial goal. This goal is repeatedly broken down into subgoals that describe the individual conditions that must exist to reach the root goal. Unlike a regular attack tree, the leaf nodes of the archetypal tree are not targeted at a specific system. Instead, the leaves constitute the points to which *concrete trees* are grafted. It is thus critical that they be selected to clearly define the boundary between broad architectural goals and vendor-specific goals.

3.2.2 Concrete trees

Concrete attack trees function as a guide for performing penetration testing on a specific system. As with the archetypal trees, we use basic guidelines to determine when

a concrete tree is specific enough. Any details not elaborated in the concrete tree must either already be known about the system, or must be discovered during pen-testing. In constructing the concrete trees for fraud and targeted disconnect, the following two rules are used:

1. *A goal should be a leaf if it is achievable completely by known means in the system.* This is the simplest case as no additional pen-testing is required.
2. *A goal should be a leaf if no vulnerability is yet known that would allow it to be executed.* At this point, determining the existence of a vulnerability enabling the goal becomes the job of penetration testing.

Concrete trees are instantiated for the two adversarial goals. The root of each concrete tree shares a reference number with a leaf in one or more archetypal attack trees to which it may be grafted. Fraud and targeted disconnect are instantiated for Testbed 1.

3.3 Reverse Engineering Tools and Other Sources

To perform actual reverse engineering tasks necessary to satisfy the details of a concrete tree a variety of tools and information sources was used.

3.3.1 Development kits

Development kits are the most important tools in reverse engineering process. Development kit generally comes with a software IDE, tools for programming and debugging, and allows access to the compatible device functionality. Sometimes development kits cannot be obtained due to products being discontinued or security-by-obscurity purposes. It may make reverse engineering task harder, but not impossible for an interested party. A development kit may even make access to actual target embedded system unnecessary since it is made of similar components.

3.3.2 Programmers and logic analyzers

Programmers, debugging emulators and logic analyzers are essential to embedded systems development. Embedded systems require debugging of new firmware images and at the same time they lack rich feedback facilities, such as monitors and verbose output for developers.

Many embedded systems have test functionality that has been hidden or left open by developers. Debugging systems in the field is frequently the reason for leaving test functionality open in circuit boards as well as avoiding re-certification of boards after this functionality is removed. This functionality can be accessed by soldering interfaces onto open circuit board contacts to obtain access to firmware programming and reading functionality. Such functionality can be manifested in serial, I2C bus, and JTAG ports. Exposed circuitry may also be used to sniff active traffic or recover loading firmware [25].

3.4 Firmware recovery

A smart meter may have its wireless functionality provided as part of main mainboard or a separate wireless card. It is possible in some cases to recover the firmware from embedded MCU ROMs or external ROM (Read Only Memory) chips. Firmware recovery may be accomplished using techniques ranging from using an off-the-shelf programmers to creating custom programmers.

ROM chips come in a variety of packaging from SOP38 and may or may not require desoldering to be read with a programmer. Pomona clips that fit the packaging style eliminate the need to desolder the chips or solder contacts directly on the ROM pins. In this work, it was possible to desolder and recover firmware from an on-board ROM chip by using an off-the-shelf programmers to read the ROM chip firmware and re-flash the chips with iterative changes to the firmware.

3.4.1 FCC test reports

In the United States, every manufacturer that plans to sell or deploy a device enabled with wireless communication must submit to compliance testing to show that the device follows federal standards. The Federal Communications Comission has historically made information, such as wireless emission test reports produced during compliance testing, public. Useful information about the system components, software, setup, and operation can be recovered to aid in the reverse engineering process. In this work, it was possible to recover information to identify various components of the system that proved useful during reverse engineering process just from these test reports. FCC is

making steps toward removing this information, which is useful for attackers, as it was noted from review of emission test reports submitted to FCC at later dates.

3.4.2 Standards

Standards always serve an important role in reverse engineering. For example, TCP/IP protocol standards have undergone extensive examination in order to find implementation errors [14]. Smart meters have a set of related standards - ANSI C12.18 Protocol Specification for ANSI Type 2 Optical Port, ANSI C12.19 American National Standard for Utility Industry End Device Data Tables, and ANSI C12.21 Protocol Specification For Telephone Modem. In this work, communication protocol details were recovered from ANSI 12.21 and uncovered details of packet structure and command execution. Useful utilities such as packet checksum calculator were also recovered from the standard.

3.4.3 Documentation

Manufacturer provided documentation provides useful informaition that can be used to gain overall understanding of the system and its functionality and to get a glimpse at the design assumptions made by system creators and maintainers. During this work, an extensive review of manufacturer documentation was undertaken to obtain information for use in reverse engineering and penetration testing. One example of useful information contained in documentation is error code and troubleshooting information provided to users of the systems. Error codes were matched to exact *crib* values in communication session dumps to assist in recovery of protocol state machine. *Crib*

values are known before hand and are generally set to facilitate identifying responsible system functionality for a key function such as authentication and password storage.

Chapter 4

Attack Trees, Issues, and Countermeasures

This chapter shows how the entire process of creating attack trees to performing the actual penetration testing using those attack trees works. The two chosen attack goals are targeted disconnect and fraud. Remote disconnect functionality and ability to perform fraud represent probably the most desired targets. First, two archtypal trees are created with each goal as a root and then concrete trees are instantiated to take care of detail related to a particular system and graft onto leaf nodes of archtypal trees. Countermeasures to address discovered vulnerabilities and discussion of repeatability of the actions conclude the chapter.

4.1 Archtypal Attack Trees

4.1.1 Energy Fraud

For initial pen-testing efforts [39], an archetypal tree for energy fraud was constructed (shown in Figure 4.1). It is described here so that it may be instantiated later. Energy fraud can be defined as any tampering with the metering infrastructure that leads to a customer not being billed for some energy consumed. In AMI, fraud may be committed in the field by modifying the recorded energy usage before it is read by the utility. Known methods for fraud in electromechanical meters include interfering with

the meter's sensors using magnets and rewinding usage gauges by inverting the meter in the socket (thereby reversing current flow through the meter).

Smart meters, present new opportunities for tampering with usage data. As shown in the first level of subgoals in the example tree, this can be done in three places (a) in the meter's low-level components, (b) the meter's long-term storage, and (c) in transmission to the utility. The archetypal attacks in this tree, as in the others, are labeled as $TX.Y$, where T is a letter specific to the tree, X is the index of the subtree below the root to which the attack belongs, and Y is the index of the attack within that subtree. Starting with the physical attacks in subtree a, there are two means to interrupt a smart meter's physical measurement of usage. A1.1 simply requires that the meter is removed from the path of current flow, and A1.2 that it be reversed in its socket. Virtually all smart meters will log and report both of these events (power cycle and reverse energy flow respectively). Thus, in the archetypal level, it is already recognized that the log messages will need to be cleared of these events.

Modifying logs and usage in meter storage is the goal of subtree b. This can be achieved in one of two ways. Either the meter's administrator password can be obtained and used to clear the log files: A2.1 AND A2.2, or the physical storage device may be tampered without interfering with the meter. As this is an archetypal tree, the implementation of the storage is left unmentioned.

The strategies for forging usage data on the wire are shown in subtree c. The interception of network communications is assumed to be necessary both for the purposes of understanding the meter's protocol stack, assuming it is non-standard, and for interposing one's self in the communication path with the utility. In the archetypal tree,

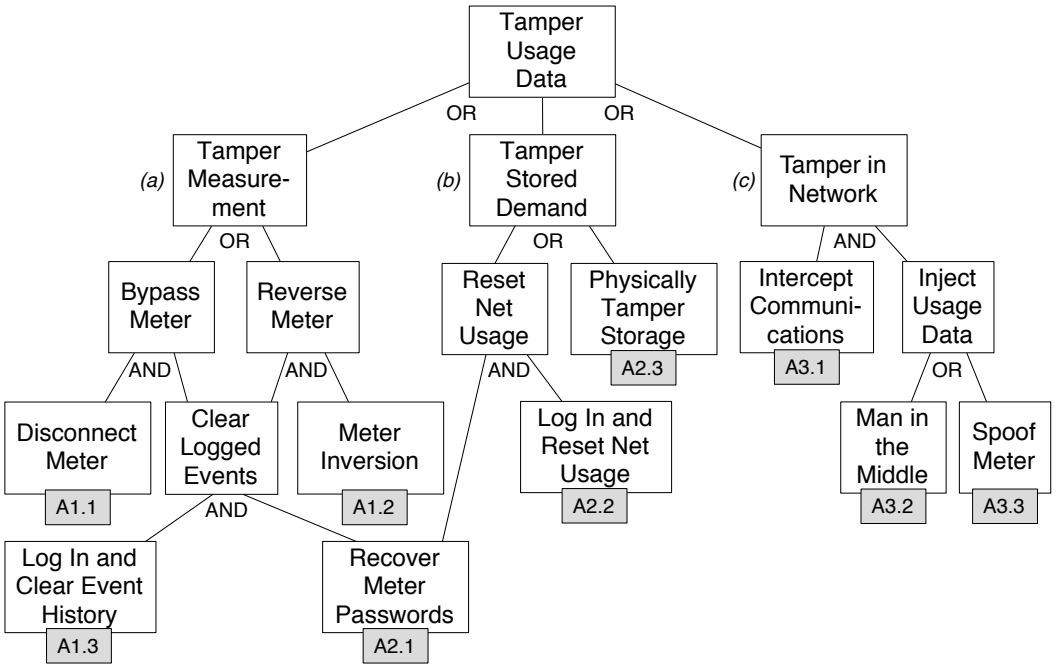

Fig. 4.1 An archtypal attack tree for fraud attack.

we ignore over which network (meter LAN or backhaul) the interception occurs, as well as any potential protection mechanisms. Along with A3.1, the adversary must either hijack a session between the meter and utility (A3.2) or impersonate a meter for the entire session (A3.3).

4.1.2 Targeted Disconnect

Archtypal attack tree construction for a targeted disconnect attack is shown in figure 4.2. The archtypal targeted disconnect tree remains general enough to be applied to most AMI vendor systems. It shows a scenario where the goal is to perform a remote disconnect command on a number of smart meter systems, that is broken down into subgoals that show necessary conditions to be satisfied suchs as recover meter passwords and include any additional activity that could be performed by an attacker to cover up

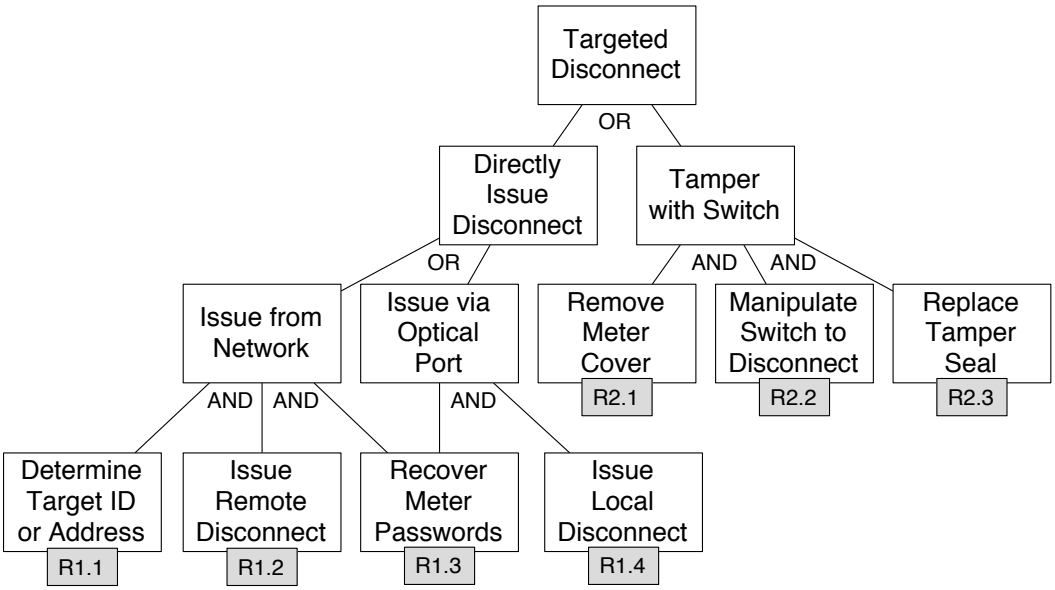

Fig. 4.2 An archtypal attack tree for targeted remote disconnect and utility lockout.

the tracks, such as replace tamper seals. Note the use of *AND* and *OR* by child nodes specifying mandatory or optional nature of individual conditions to satisfy a particular subgoal.

Most meter vendors include remote disconnect functionality in their meters. The ability to disconnect a target's power can cause at best, inconvenience and in worse scenarios, financial or physical harm depending on the setting. As described earlier, remote disconnect systems consist of a physical switch that breaks the current flowing to the house, and a set of remote commands to operate this switch.

The ideal case for an adversary would be to issue the disconnect command remotely. Doing this requires at least that the ID be known for the target device (R1.1), and that its administrator password has been recovered (R1.3). Notice that this is the second archetypal tree with a leaf node requiring meter passwords to be recovered. This

illustrates a secondary usefulness of attack trees: they act as a reference for quickly mapping security flaws to the adversarial goals they enable.

We reason that the disconnect functionality will be accessible through the optical ports on most systems because optical port functionality needs to contain at least the network functionality to allow the meter to function in the event that it is not network accessible, e.g. the meter's network card is malfunctioning. This is the basis of archetypal attacks R1.3 and R1.4.

Finally, physical access to a meter may also be useful for manipulating the disconnect switch, be it by mechanical or electrical means (R2.2). From experience, it was found that virtually all smart meters use the same tamper seal [1]. When contacted, the manufacturer of these seals and confirmed that there are no limitations on the text which we could have embossed on the flag.

4.2 Concrete Attack Trees

4.2.1 Energy Fraud in Testbed 1

The archetypal attack tree for energy fraud 4.1 presented three broad strategies: tampering with the measurement process, tampering with the recorded usage in meter storage, and tampering with the usage data in transmission. For the first attempt to implement a fraud attack in Testbed 1, the third strategy was chosen because of its relatively low invasiveness and our understanding of the backhaul network operation. This strategy terminated in three archetypal attacks: a mandatory requirement of being interposed on the backhaul link (A3.1), and the option of either performing a man in

the middle attack (A3.2) or meter spoofing (A3.3). After evaluating the ANSI C12.21 specification via a trace of Testbed 1's telephony-based diagnostic protocol, it was determined that meter spoofing was more straightforward. Thus, to complete the goal of fraud in Testbed 1, we must instantiate and execute concrete trees for archetypal attacks A1.1 and A1.3. Both concrete trees are shown in Figure 4.3.

Archetypal attack A3.1 requires that the adversary be interposed somewhere on the path between the meter's networking interface card (NIC) and the utility. In one extreme end, this may be achieved by directly tampering with the communications bus on which the NIC resides (a1.1). Two more likely places are the mesh network (a3.1), and the telephone backhaul (a2.1). For the latter, the additional prerequisite of bypassing the "intrusion detection" mechanism is necessary (a2.2).

The second archetypal attack for energy fraud requires meter spoofing. This calls for three steps to successfully deliver forged usage data as part of Testbed 1's diagnostic protocol. First, the spoofing device must initiate a new diagnostic session with the utility. This will require first identifying itself as the expected meter (a4.1), and second, completing the authentication round (a4.2). Once the session is established, the spoofing device must answer all diagnostic queries up to the forged demand (a5.1), and finally, insert the forged demand value (a6.1). The remainder of work to realize these attacks is achieved by pen-testing as described in section 4.3.

4.2.2 Targeted Disconnect in Testbed 1

The final concrete attack tree analyzed here is for the disruption of electrical service. As an adversary would ideally want to execute this attack remotely, archetypal

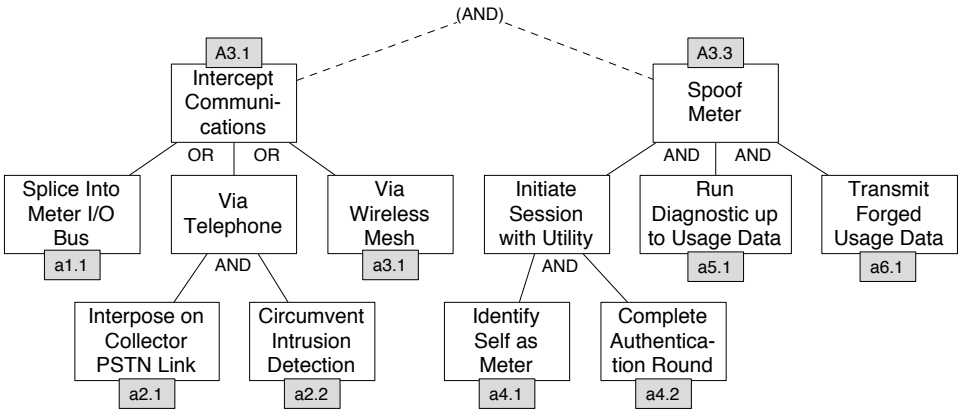

Fig. 4.3 Concrete attack tree demonstrating dependency chain to fraud attack.

attacks R1.1 - R1.3 are chosen for instantiation. In Testbed 1, the meter ID is printed on the front of each meter, making R1.1 achievable by visual inspection. The concrete trees for R1.2 and R1.3 are shown in Figure 4.6 and in 4.8 showing achieved node coverage.

Two strategies are feasible for meter password recovery in Testbed 1 (R1.3). If the optical port can be physically monitored, then the password can be obtained upon the next visit by the utility (r1.1). Alternatively, if the contents of meter storage can be extracted, the password may be recoverable, though potentially only in a hashed format (r1.3). As both of these are physical attacks, they may only be used to recover a password from a single meter. This would normally be a limiting factor in the impact of an attack against Testbed 1, but we observe that its architecture encourages utilities to use the same password for a large number of meters. In the administrative utility-end software, a single password set (consisting of a read-only and administrative user) is chosen for a template program that is pushed to the meters at configuration time. This makes it very tedious to create a different program template for each meter. A brute force guessing attack is not considered, as the maximum length of a password in

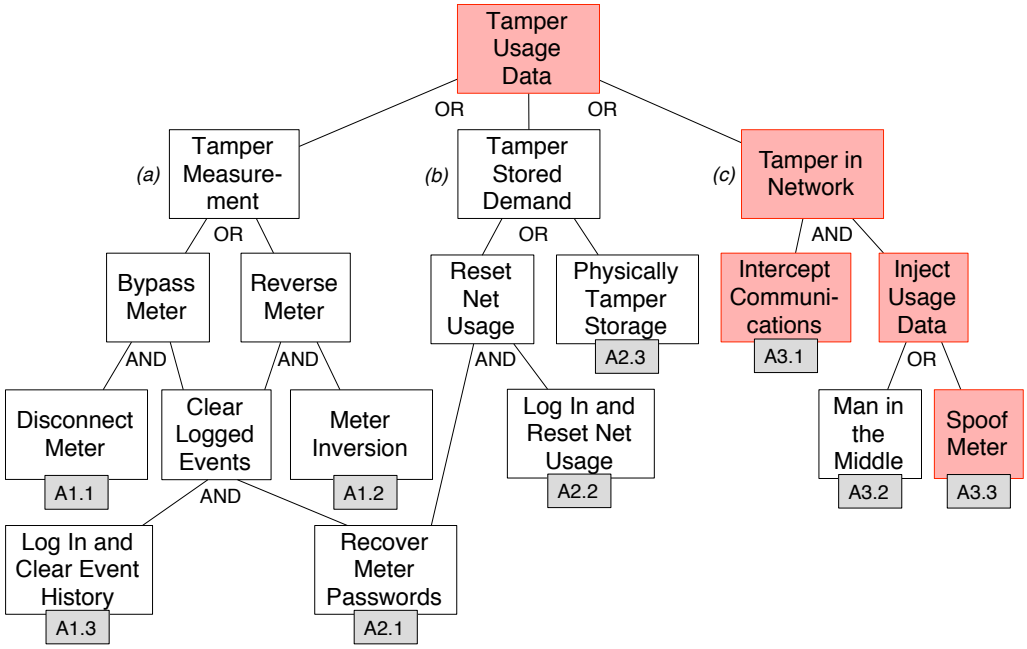

Fig. 4.4 An archtypal attack tree for targeted remote disconnect and utility lockout with red nodes showing coverage.

Testbed 1 is well over ten bytes. The final archetypal attack needed is the issuance of the command to the target meter. This requires that the known password be used in the mutual authentication round (the same as that used in Testbed 1's diagnostic protocol) (r2.1). Once authenticated, the command can be issued (r2.2).

4.3 Issues

The issues presented below were discovered through penetration testing and reverse engineering of the two testbed systems. The implications of these issues are indicative of presence of attack prerequisites in smart meter systems that can help launch a large scale attack at a neighborhood level or greater. Besides the above, privacy and fraud implications are running in parallel due to the similarity of prerequisites.

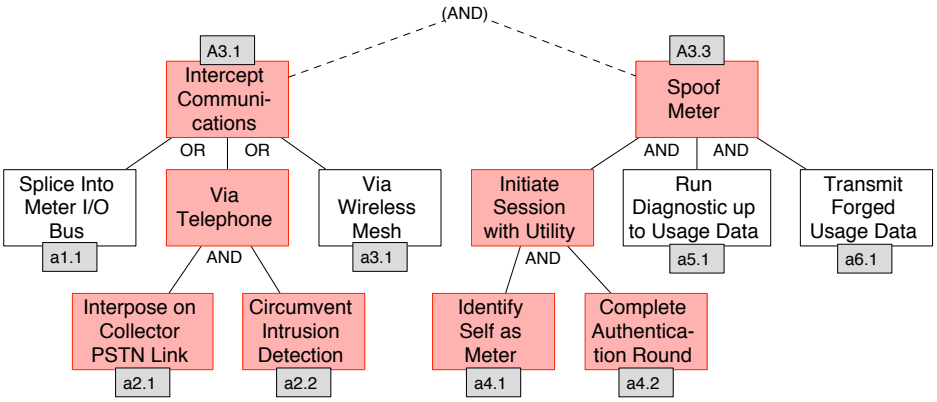

Fig. 4.5 Concrete attack tree demonstrating dependency chain to remote disconnect and utility lockout with red nodes demonstrating achieved coverage.

- **Password keys and derived credentials are stored in plaintext in wireless card ROM firmware.**

  This failure is trivial in nature as the system does not attempt to protect the password via hashing and salting a known plaintext string. Hashing by itself provides no security and due to advent of rainbow tables [3], which make brute force attacks - trying every permutation of string - significantly faster [24, 4]. Salting is a technique of ensuring uniqueness of the resulting hashed string, which can remain stored as plain text. Salting still doesn't guarantee that adversaries won't succeed on their own as they are free to perform bruteforce computation at their leasure or force a system to use a known salt value, but it is a step in the right direction. This vulnerability is an implementation and design failure. The vulnerability was confirmed by setting password sets to plaintext values known as *cribs* and searching through firmware images recovered from firmware ROM image. Following some

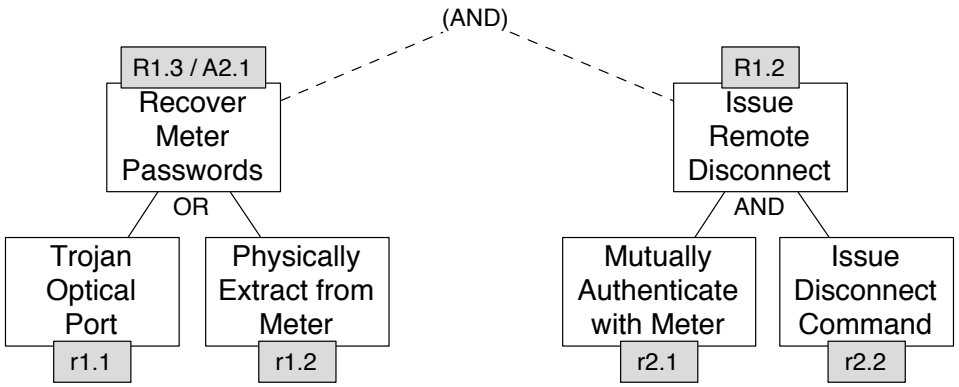

Fig. 4.6 Concrete attack tree demonstrating dependency chain to remote disconnect and utility lockout.

form of a well defined and accepted implementation standards, such as PKCS, would be a countermeasure to such vulnerability.

- **Passwords are duplicated inside the system both on the MCU flash storage and on the wireless card storage.**

  This issue has been verified via booting a target smart meter system with and without wireless card after discovering plaintext password stored in the cards on-board ROM chip. The motivation behind such duplication may be offloading processing from the smart meter onto intelligent wireless card, which has its own MCU. By itself, this is not a vulnerability if password management is implemented correctly. A discarded or recycled wireless card may contain enough information for an attacker to recover passwords and other useful information.

- **Passwords are identical for all interfaces - optical port and wireless authentication.**

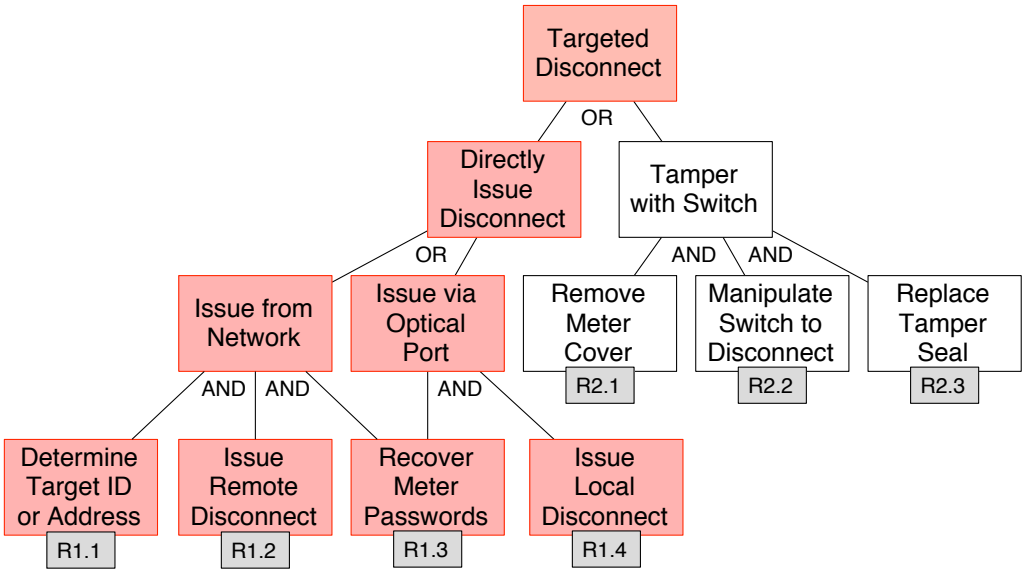

Fig. 4.7 An archtypal attack tree for targeted remote disconnect and utility lockout with red nodes showing coverage.

Having separate credentials provides some degree of separation between interfaces. Having a single credential for all interfaces makes it easy to obtain access via multiple methods.

- **Passwords are transmitted in plaintext via optical port connections.**

  An attacker may intercept the password at an optical interface serial line or host ports. When combined with the previously described issue of credential duplication, an adversary can re-use it to launch an attack on a wireless interface.

- **Replay and spoofing attacks are possible due to implementation errors in authentication.**

  This attack enables fraud through spoofing and MITM (Man-In-The-Middle) attacks [17]. This attack is also discussed in detail in [39]. Previously recorded

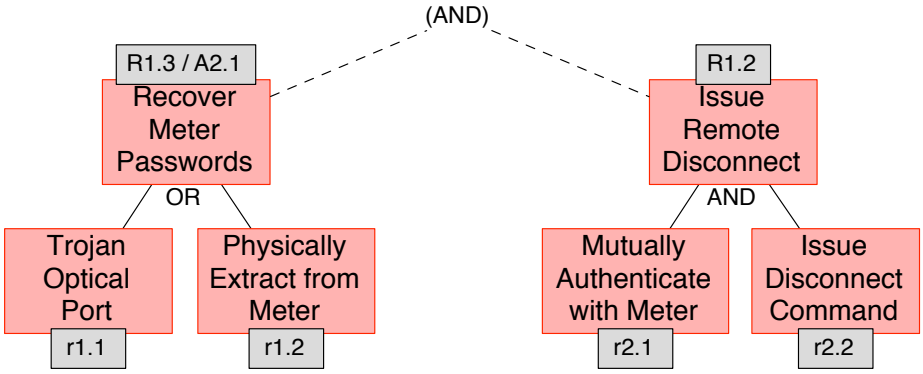

Fig. 4.8 Concrete attack tree demonstrating dependency chain to remote disconnect and utility lockout with red nodes showing coverage.

sessions of serial port communication between the utility machine and the smart meter collector node were used to generate scripted replay. The replay attack was performed in both directions to and from the utility. The importance of this attack is that a customer may spoof a smart meter that has been circumvented to deliver falsified usage and demand data hence enabling theft of service. Figure 4.9 explains the failure to check so "nonce" value to ensure freshness of communication.

- **Communication protocols use simple byte summing and checksums to verify integrity.**

  Byte summing and checksum routines are widely available on the internet and in starndards documentation so it was fairly straightforward to recover them and begin constructing valid communication packets. Using a communication protocol that is standard and allows for better notion of packet integrity and random starting sequences will help prevent this.

- **Tamper seals are ineffective in preventing tampering with smart meters.**

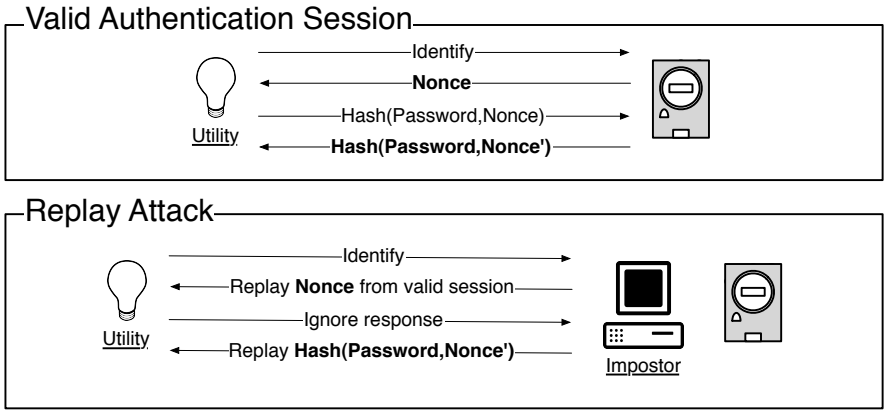

Fig. 4.9 The replay attack discovered in the studied system. Because the two messages in the mutual authentication round are dictated by the nonce, replaying a previously recorded nonce will allow the impostor to authenticate without knowing the password used to key the hash [39].

It was discovered that tamper seals could be ordered in bulk from the manufacturer with any text on them. Such tamper seals are offering only psychological deterrent value in real life.

- **Modem intrusion detection is defeatable**

  Whenever a meter detects that another device on the same line has gone off the hook, it will hang up. This works correctly in the case when a device such as a telephone or modem (technically any FXO) picks up on the line. This feature does not work in the case of devices placed on the path between the meter and first link to the phone company. The PBX in our experimental testbed is one example of such a device (technically an FXS). Using the open source software running on the PBX, we were able to monitor modem communication.

Table 4.1 Summary of concrete attacks and discovered vulnerabilities for each adversarial goal.

Ref.	Description	Enabling Feature or Vulnerability
	Energy Fraud in Testbed 1	
a2.1	Interpose between utility and collector	Telephone line may be accessible.
a2.2	Defeat modem intrusion detection	The mechanism cannot detect an FXS.
a4.1	Identify self as meter	A meter's ID is printed on its faceplate.
a4.2	Complete authentication round	Lack of nonce-tracking allows replayed authentication.
a5.1	Run diagnostic up to usage data	Protocol is standardized.
a6.1	Transmit forged usage data	Usage data is not integrity protected.
	Targeted Disconnect in Testbed 1	
r1.2	Physically extract passwords	Passwords are stored in the clear in ROM storage.
r2.1	Mutually authenticate with meter	The encryption key is derived from passwords.
r2.2	Issue disconnect command	Administrative software is commercially available.

4.4 Countermeasures

Applying countermeasures for discovered vulnerabilities involves finding the best possible solution, which addresses the attack vector. Attack trees provide another tool in finding this solution. Attack trees help visualize the path of an attack from the leaf to the root goal. The goal is to place the countermeasure as close as possible to the root in order to render as many child nodes, which represent attack prerequisites, ineffective. An example of a countermeasure that would be of a high value and close to the root would be a firewall, which prevents known attacks on computer system in a hypothetical Internet worm attack tree. Placing countermeasures at the leaf nodes to ensure that they cannot be instantiated may provide a partial, but a practical solution motivated by the requirements of the system.

4.4.1 Energy fraud countermeasures

Energy fraud in Testbed 1 has been enabled via replay attack. Testbed 1 allowed scripted replay in both directions to the utility and to the smart meter allowing an adversarial customer to spoof the meter. Additional nodes that were needed for successful instantiation are defeating intrusion detection and completing authentication rounds to the meter. The main countermeasure here is to stop the attack from happening by addressing poor implementation of nonce tracking in a4.2. The are multiple implementations of authentication protocols such as CHAP,MS-CHAP, and MS-CHAP version 2, which can be used to alleviate this problem.

Weak notion of packet sequencing and integrity checking using byte summing and checksums can be prevented through the use of a standard protocol supporting those features or introducing this functionality. Employing random starting sequences, similar to TCP protocol packet sequencing, for packets on network would also help protect session from hijacking. Use of standardized authentication and communication protocol is necessary to take advantage of years in security research of those protocol.

Unencrypted communication is another pitfall that can be prevented using an existing protocol, such as SSL/TLS.

4.4.2 Targeted disconnect countermeasures

Targeted disconnect attack attack has been successful because it was possible to instantiate concrete trees specific to Testbed 1. Issuing remote disconnect (R1.2) involved physical extraction of passwords and obtaining passwords via snooping on optical

port communication(r1.2,r1.1), both of which relate to the subject of proper key management. Plaintext and derived keys were easily extracted because the system didn't use hashing and salting of the keys with a known plaintext string. Plaintext comminication and transmission of passwords also allowed to recover keys. Employing encryption of communication sessions on optical and other interfaces will also help prevent recovery of keys(r1.1). Following a proper implementation standard and will address physical extraction vulnerability at the leaf.

Another discovered issue was the ability to use the same keys for all interfaces. Any notion of key sets applicable to even neighborhood scale installations runs counter to the observation that it is advantageous for security to make the attacker work equally hard for every node. Smart meter interfaces, such as optical and PSTN, must not share common passwords and should be separated as is currently done on some of the smart meter systems to prevent insider attacks. Some of the existing systems are already enabled with separate credentials for each of the interfaces limiting access to critical portions of the system. Production-time generation of keys and placing those keys into firmware has been long employed to speed up deployment process and may help prevent setting neighborhood level passwords. Utility software must also be enabled with the same capability to regenerate keys for maintenance purposes, cyclically, or, in the event of a staffing change. For example, in Testbed 2 remote use of recovered keys was precluded by having to know regularly refreshed session keys for communication between utility and home collector.

Many smart meters have open test functionality still present on system boards. In this case presense of such test functionality allowed us to recover the firmware and the

keys contained in it. Removing such functionality may help slow the attacker, but runs counter to engineer's need to debug system in the field. Additionally, taking advantage of security bit functionality, which prevents firmware dumping, may make the process slower, but not impossible [16], [7]. This is another countermeasure applied at the leaf. Again, use of this feature may run counter to the need of engineers to debug and reprogram the system. Removing or changing functionality of the circuit boards requires expensive validation tests, which are often mandatory for compliance and certification by certain standards bodies such as ISO. Each iteration of software must be verified placing signficant time and effort burden on manufacturers resources. Manufacturers may not be motivated to make this investment frequently.

Duplication of keys on wireless card firmware with identical key management implementation issues described earlier was also discovered in Testbed 1. Addressing this implementation error is another countermeasure applied at the leaf. In this case use of proprietary wireless stacks places a smart meter system at a disadvantage. Open stacks, such as Zigbee, while not bulletproof [26, 25], receives significant attention from multiple parties and may offer the best option for smart meter designers by providing examples of good key management. A single manufacturer will be unable to match such communal effort of standard body and other contributing parties. Some smart meter systems have chosen to employ existing 802.11X technologies, which allows them to benefit from years of prior security work on the general purpose computers.

Another countermeasure that lies in the realm of policy making is creation of guidelines for remote disconnect technology deployment. For example, critical infrasture installations may not receive remote disconnect enabled smart meters and regions with

extreme weather conditions should not receive them either. A set of such guidelines will most likely evolve naturally over time.

4.5 Application of attack trees to other systems and regression

The two remaining points to validate the process that has been shown with Testbed 1 are application to other similar vendor systems - our Testbed 2, and regression testing. Attack tree methodology was successfully applied to Testbed 2 to discover a DoS(Denial-of-Service) attack, which prevented utility's ability to communicated with the meter. Information gathered during the targeted disconnect attack on Testbed 1 was also applied to Testbed 2 to check for the existence of same vulnerability. Targeted disconnect attack success in Testbed 2 was precluded by key management implementation requiring knowledge of temporal session keys. For details of the DoS attack and related attack trees you may refer to [40, 19].

Chapter 5

Conclusion

This work has shown that it is possible to perform penetration testing using a repeatable process similar to regression testing. Knowledge captured in the attack trees is re-usable across iterations of a system or similar systems. Attack trees are a valuable tool in organizing penetration testing and reverse engineering efforts for vulnerability discovery.

This work also demonstrates a number of preventable and well known issues found through penetration testing of currently deployed systems. Smart grid deployment will go on for many years to come. This first generation of smart meters and AMI/AMR products may help utilities and manufacturers learn and address some of the security issues in the future iterations. Failure to heed the lessons of secure embedded systems design and other known facts and past examples in computer security may result in deployment of a vulnerable grid. Continous and systematic penetration testing effort must be an integral part of AMI/AMR products.

Bibliography

[1] B.T. Aluminum Tamper Seal. http://www.brooksutility.com/catalog/product-detail.asp?ID=302.

[2] Extracting keys from second generation zigbee chips. http://www.blackhat.com/presentations/bh-usa-09/GOODSPEED/BHUSA09-Goodspeed-ZigbeeChips-PAPER.pdf.

[3] Free rainbow tables. http://www.freerainbowtables.com/.

[4] John the ripper password cracking tool. http://www.openwall.com/john/.

[5] Nmap tool. http://www.nmap.org.

[6] Rising electricity costs: A challenge for consumers, regulators, and utilities. http://www.entergy.com/global/documents/utility/industry/EEi_rising_electricity_costs.pdf.

[7] Security protection in motorola microcontrollers. http://www.etlweb.com/?ref=articles_secprotect.

[8] Wireshark network capture tool. http://www.wireshark.org/.

[9] France heat wave death toll set at 14802. http://www.usatoday.com/weather/news/2003-09-25-france-heat_x.htm, 2003.

[10] M Abrams and J Weiss. Malicious control system cybersecurity attack case study - maroochy water services. http://www.mitre.org/work/tech_papers/tech_papers_08/08_1145/08_1145.pdf.

[11] Ross J. Anderson. Why cryptosystems fail. In *ACM Conference on Computer and Communications Security*, pages 215–227, 1993.

[12] Ross J. Anderson and Shailendra Fuloria. On the security economics of electricity metering. In *Cambridge University Computer Laboratory*, 2010.

[13] Ross J. Anderson and Shailendra Fuloria. Who controls the off switch? In *Workshop on the Economics of Information Security*, 2010.

[14] Steven M. Bellovin. A look back at "security problems in the tcp/ip protocol suite". In *ACSAC*, pages 229–249. IEEE Computer Society, 2004.

[15] Kevin R. B. Butler, William Enck, Harri Hursti, Stephen E. McLaughlin, Patrick Traynor, and Patrick McDaniel. Systemic issues in the hart intercivic and premier voting systems: Reflections on project everest. In David L. Dill and Tadayoshi Kohno, editors, *EVT*. USENIX Association, 2008.

[16] Matthew Carpenter, Travis Goodspeed, Bradley Singletary, Ed Skoudis, and Joshua Wrigh. Ami attack methodology. http://inguardians.com/pubs/AMI_Attack_Methodology.pdf.

[17] Radia Perlman Charlie Kaufman and Mike Speciner. Network security:private communication in a public world (2nd edition). Prentice Hall, 2002.

[18] Ang Cui and Salvatore J. Stolfo. A quantitative analysis of the insecurity of embedded network devices: results of a wide-area scan. In Carrie Gates, Michael Franz, and John P. McDermott, editors, *ACSAC*, pages 97–106. ACM, 2010.

[19] Adam Johaness Delozier. Characterization of vulnerabilities and countermeasures in advanced metering infrastructure collectors. Master's thesis, Penn State University Computer Science and Engineering, 2011.

[20] Laurent Eschenauer and Virgil D. Gligor. A key-management scheme for distributed sensor networks. In Vijayalakshmi Atluri, editor, *ACM Conference on Computer and Communications Security*, pages 41–47. ACM, 2002.

[21] Ahmad Faruqui, Ryan Hledik, Sam Newel, and Johannes Pfeifenberge. The power of five percent: How dynamic pricing can save $35 billion in electricity costs. http://sites.energetics.com/MADRI/pdfs/ArticleReport2441.pdf.

[22] Katie Fehrenbacher. Smart grid stiumulus funding revealed! gigaom.com/cleantech/smart-grid-stimulus-funding-revealed.

[23] Katie Fehrenbacher. Smart Meter Worm Could Spread Like A Virus. http://earth2tech.com/2009/07/31/smart-meter-worm-could-spread-like-a-virus/.

[24] Vivek Gite. Linux password cracking: Explain unshadow and john commands (john the ripper tool). http://www.cyberciti.biz/faq/unix-linux-password-cracking-john-the-ripper/.

[25] Travis Goodspeed. Breaking 802.15.4 aes128 by syringe. http://travisgoodspeed.blogspot.com/2009/03/breaking-802154-aes128-by-syringe.html.

[26] Travis Goodspeed. Prng vulnerability of z-stack zigbee sep ecc. http://travisgoodspeed.blogspot.com/2009/12/prng-vulnerability-of-z-stack-zigbee.html.

[27] The Smart Grid Interoperability Panel Cyber Security Working Group. Smart Grid Cyber Security Strategy and Requirements DRAFT NISTIR 7628, February 2010.

[28] Mabrouka El Guedri, Guy D'Urso, Chrstian Lajaunie, and Gilles Fleury. Time-Frequency Characterisation for Electric Load Monitoring. In *Proceedings of the 17th European Signal Processing Conference (EUSIPCO)*, 2009.

[29] James D. Hamilton. Causes and consequences of the oil shock of 2007-08. 2009.

[30] Chris S King. The economics of real-time and time-of-use pricing for residential consumers., 2001.

[31] R. Kinney, P. Crucitti, R. Albert, and V. Latora. Modeling cascading failures in the north american power grid. *The European Physical Journal B - Condensed Matter and Complex Systems*, 46(1):101–107, July 2005.

[32] Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, Tadayoshi Kohno, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, and Stefan Savage. Experimental security analysis of a modern automobile. In *IEEE Symposium on Security and Privacy*, pages 447–462. IEEE Computer Society, 2010.

[33] Brian Krebs. Plentyoffish.com hacked, blames messenger. `http://krebsonsecurity.com/2011/01/plentyoffish-com-hacked-blames-messenger/`.

[34] C. Laughman, K. Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong. Power signature analysis. *IEEE Power and Energy Magazine*, 1:56–63, 2010.

[35] Michael LeMay and Carl A. Gunter. Cumulative attestation kernels for embedded systems. In Michael Backes and Peng Ning, editors, *ESORICS*, volume 5789 of *Lecture Notes in Computer Science*, pages 655–670. Springer, 2009.

[36] Mikhail A. Lisovich, Deirdre K. Mulligan, and Stephen B. Wicker. Inferring personal information from demand-response systems. *IEEE Security and Privacy*, 8:11–20, 2010.

[37] Yao Liu, Peng Ning, and Michael K. Reiter. False data injection attacks against state estimation in electric power grids .

[38] Patrick McDaniel and Stephen McLaughlin. Security and Privacy Challenges in the Smart Grid. *IEEE Security & Privacy Magazine*, May/June 2009.

[39] Stephen E. McLaughlin, Dmitry Podkuiko, and Patrick McDaniel. Energy theft in the advanced metering infrastructure. In Erich Rome and Robin E. Bloomfield, editors, *CRITIS*, volume 6027 of *Lecture Notes in Computer Science*, pages 176–187. Springer, 2009.

[40] Stephen E. McLaughlin, Dmitry Podkuiko, Sergei Miadzvezhanka, Adam Delozier, and Patrick Drew McDaniel. Multi-vendor penetration testing in the advanced metering infrastructure. In Carrie Gates, Michael Franz, and John P. McDermott, editors, *ACSAC*, pages 107–116. ACM, 2010.

[41] The Asterisk Project. Asterisk open source pbx. `http://www.asterisk.org`.

[42] J H Saltzer, D P Reed, and D D Clark. End-to-end arguments in system design. `http://web.mit.edu/Saltzer/www/publications/endtoend/endtoend.pdf`.

[43] Bruce Schneier. Attack trees. *Dr Dobb's Journal*, 24(12), December 1999.

[44] Zachary M. Seward and Albert Sun. The top 50 gawker media passwords. http://blogs.wsj.com/digits/2010/12/13/the-top-50-gawker-media-passwords/.

[45] J Slay and M Miller. Lessons learned from the maroochy water breach, ifip series. Boston, MA:Springer, 2007, vol. 253, ch 6, pp 73-82.

[46] T Smith. Hacker jailed for revenge sewage attacks. http://www.theregister.co.uk.

[47] Vijay Srinivasan, John Stankovic, and Kamin Whitehouse. Protecting your daily in-home activity information from a wireless snooping attack. In *UbiComp '08: Proceedings of the 10th international conference on Ubiquitous computing*, pages 202–211, New York, NY, USA, 2008. ACM.

[48] Patrick Traynor, Raju Kumar, Heesook Choi, Guohong Cao, Sencun Zhu, and Thomas F. La Porta. Efficient hybrid security mechanisms for heterogeneous sensor networks. *IEEE Trans. Mob. Comput.*, 6(6):663–677, 2007.

[49] Scott Wolchok, Eric Wustrow, J. Alex Halderman, Hari K. Prasad, Arun Kankipati, Sai Krishna Sakhamuri, Vasavya Yagati, , and Rop Gonggrijp. Security analysis of india's electronic voting machines. *To appear in Proc. 17th ACM Conference on Computer and Communications Security (CCS '10)*, 2010.

[50] Kim Zetter. Attack code for scada vulnerabilities released online. http://www.wired.com/threatlevel/2011/03/scada-vulnerabilities/.