APTA STANDARDS DEVELOPMENT PROGRAM
WHITE PAPER
American Public Transportation Association
1666 K Street, NW, Washington, DC, 20006-1215

APTA SS-CC-03-15
SSPP Approved April 8, 2015
Control and Communications
Security Working Group

Securing Control and Communications Systems in Rail Transit Environments

Part IIIa: Attack Modeling Security Analysis White Paper

Abstract: This document contains recommendations for attack modeling analysis that may be specified in transit agency procurement documents to examine security aspects for new rail systems or major upgrades.

Keywords: attack modeling, attack trees, communications based-train control (CBTC), control and communications security, cybersecurity, positive train control (PTC), radio, rail transit vehicle, signaling, supervisory control and data acquisition (SCADA), train control

Summary: This *White Paper*, part of a series of related documents (see Introduction), covers the APTA attack modeling procedure for transit agencies and their systems integrators and vendors, which may be specified by transit agencies in their procurement documents. This document should be used in conjunction with Part I and Part II of this series.

Scope and purpose: For large and/or complex new projects or major upgrades, especially those projects using new technology, transit agencies may want additional analysis and assurance of the security of control and communication systems. These agencies may specify that the supply chain (system integrators and vendors) perform an attack modeling analysis as part of the deliverables. Since this analysis is detailed and time-consuming, it is suggested that attack modeling only be performed on certain key sections of the whole design, and that transit agency representatives be an integral part of the attack modeling team, with oversight responsibilities. In the event that security/safety or other standards exist for any of the above systems, this *White Paper* will supplement, provide additional guidance for, or provide guidance on how control systems may securely interface with these systems. These documents are not to be construed as legally binding requirements of, or official implementing guidance for, any current or future regulations of the Department of Homeland Security.

This *White Paper* represents a common viewpoint of those parties concerned with its provisions, namely, transit operating/planning agencies, manufacturers, consultants, engineers and general interest groups. The application of any standards, practices or guidelines contained herein is voluntary. In some cases, federal and/or state regulations govern portions of a transit system’s operations. In those cases, the government regulations take precedence over this standard. NATSA recognizes that for certain applications, the standards or practices, as implemented by individual transit agencies, may be either more or less restrictive than those given in this document.

Participants

The North American Transportation Services Association greatly appreciates the contributions of the following individuals, who provided the primary effort in the drafting of this *White Paper*.

At the time this document was completed, the Control and Communications Security Working Group included the following members:

Joy Thompson, *Chair*
Thomas Joseph, *Vice Chair*
David Teumim, CISSP, *Facilitator*
John Moore, *Secretary*
David Trimble, *Editor*

Lee Allen
Mark Fabro
Kevin Garben
Mark Hartong
Dan Hillman
Stanley Kwa
Theo Lawrence
Neal Mondschein
Mark O'Shea
Mridu Oravakandy
Gene Overly
Fred Polsky
Kosit Tesuchata
Tom VanNorman
Leigh Weber
John Weikel
Chuck Weissman

APTA Standards Project Team
Dave Hahn, *APTA Senior Program Specialist*

Special thanks to Terry Ingoldsby of Amenaza Technologies Limited for his contribution.

Contents

Introduction iii

1. About this series 1
1.1 Intent of the series 1
1.2 Parts of the series 2

2. APTA's attack modeling procedure 2
2.1 What is attack modeling? 2
2.2 Why is attack modeling used? 2
2.3 Who should do attack modeling? 2
2.4 How does attack modeling fit into the APTA document series? . 3
2.5 When should attack modeling be used? 3

3. Defining attack tree analysis scope 3
3.1 What to model 3
3.2 How big an area to model 4
3.3 Forming an attack tree modeling team 4
3.4 Necessary tools 4
3.5 Expected time and effort 4
3.6 Expected deliverables 4
3.7 Attack modeling software 5
3.8 Attack modeling process 5

4. Illustrating the attack modeling process, a case study 8
4.1 Background on case study: The security question to be answered 8
4.2 Assumptions 8
4.3 Scenario development and physical layout 8
4.4 Attack modeling case study 12
4.5 Explanation of Figure 6 16

5. Attack scenario analysis (short method) 19
5.1 Describe in detail the selected higher-risk scenarios 20

6. Select and propose countermeasures (security controls) 23

7. Summary of attack modeling method 24

References 25

Definitions 25

Abbreviations and acronyms 26

Introduction

This White Paper is Part IIIa in a series of documents:

- **Part I:** Released in July 2010, Part I addresses the importance of control and communications security to a transit agency, provides a survey of the various systems that constitute typical transit control and communication systems, identifies the steps that an agency would follow to set up a successful program, and establishes the stages in conducting a risk assessment and managing risk.
- **Part II:** Part II, published in July 2013, presents Defense-In-Depth as a recommended approach for securing rail communications and control systems, defines recommended security zone classifications, and defines a minimum set of recommended security controls for the most critical classification: safety-critical.
- **Part III:** Part III consists of two subparts, a and b, with a possibility of adding a subpart c later.
    - Subpart IIIa (this *White Paper*) covers the APTA attack modeling procedure for transit agencies and their systems integrators and vendors, which may be specified by transit agencies in their procurement documents.
    - Subpart IIIb (a companion document) will cover the Operationally Critical Security Zone (OCSZ), in the same manner as how Part 2 covered the SCSZ and FLSZ zones.
    - Subpart IIIc (a future document) will cover application of three security zones, the OCSZ, FLSZ, and SCSZ to rail transit vehicles.

APTA recommends the use of this *White Paper* by:

- individuals or organizations that operate rail transit systems;
- individuals or organizations that contract with others for the operation of rail transit systems; and
- individuals or organizations that influence how rail transit systems are operated (including but not limited to consultants, designers and contractors).

Attack Modeling Security Analysis

1. About this series

This *White Paper* is Part IIIa in a series of documents. Due to the comprehensive amount of information to be conveyed, this series is divided into multiple parts (see **Table 1**).

TABLE 1
List of Series Documents

Part I	Published July 2010	Elements, Organization and Risk Assessment/Management
Part II	Published July 2013	Defining a Security Zone Architecture for Rail Transit and Protecting Critical Zones
Part IIIa	This document	Attack Modeling for Rail Transit
Part IIIb	Publication date TBD	The Operationally Critical Security Zone
Part IIIc	Future document	Security Zones Onboard the Train Set

This division of material parallels the progression of recommended steps a transit agency would follow to develop and implement a control and communications security program.

Part I addresses the importance of control and communications security to a transit agency, provides a survey of the various systems that constitute typical transit control and communication systems, identifies the steps that an agency would follow to set up a successful program, and establishes the stages in conducting a risk assessment and in managing risk. Part II presents Defense-in-Depth as a recommended approach for securing rail communications and control systems, defines security zone classifications, and defines a minimum set of security controls for the most critical zones, the Safety-Critical Security Zone (SCSZ) and the Fire/Life-Safety Security Zone (FLSZ). Part III will cover recommended practices for other zones and the rail vehicles and provide security analysis guidance for a transit agency.

1.1 Intent of the series

The intent of this document series is to provide guidance to transit agencies on securing control and communications systems for their rail environments. This series spearheads an effort within APTA to extend cybersecurity best practices to the transit industry.

It represents the contribution of leading-edge information from transit agencies that already have a control security program, as well as recommendations from the U.S. Department of Homeland Security (DHS), the Transportation Security Administration (TSA), the National Institute of Standards and Technology (NIST) and vendors who serve the transportation and IT communities. APTA intends for this series to serve as a guide for transit agencies to develop a successful and comprehensive cybersecurity program.

This document series is not intended to supplant existing safety or security standards and regulations. Instead, it provides an overview of the need for control and communications protection, and it fills in potential gaps in current standards and regulations.

1.2 Parts of the series

1.2.1 Part I, Elements, Organization and Risk Assessment/Management

Part I addresses the importance of control and communications security to a transit agency, provides a survey of the various systems that constitute typical transit control and communication systems, identifies the steps that an agency would follow to set up a successful program, and establishes the stages in conducting risk assessment and managing risk.

1.2.2 Part II, Defining a Security Zone Architecture for Rail Transit and Protecting Critical Zones

Part II assumes that the agency has completed the risk assessment and risk management steps of Part I. This document covers how to define security architecture for control and communications systems based on the Defense-in-Depth model. It also defines a minimum set of controls for the SCSZ and the FLSZ, which are the most critical zones. The primary application is intended to be for new rail projects or major upgrades rather than for retrofitting legacy systems.

1.2.3 Part IIIa, Attack Modeling Security Analysis

Part IIIa (this *White Paper*) covers the APTA attack modeling for transit agencies and their systems integrators and vendors, which may be specified by transit agencies in their procurement documents. Please note that this document should be used in conjunction with Parts I and II of this series.

2. APTA’s attack modeling procedure

2.1 What is attack modeling?

Attack modeling is a method for analyzing security in critical sections of rail transit security zones. It uses the procedure shown in Section 3.8, properly scaled to model the critical sections and to answer project team questions.

Attack modeling uses “attack trees” as part of the procedure. An attack tree is a graphical representation of how a system under study may be compromised, either accidently or by those with harmful intent. It is closely related to a fault tree, which is used in system safety studies but does not include actions with harmful intent. As with a fault tree, the harmful event, the one the transit agency is trying to prevent, such as an explosion or a derailment, goes at the top of the tree. For an introductory view on attack trees and attack tree software, see the Amenaza Technologies Limited publication “The SecurITree® BurgleHouse Tutorial” (see References).

2.2 Why is attack modeling used?

- To allay security concerns of transit agencies on new control and communications system designs.
- To explore the security aspects of a new technology during the development stages.
- To evaluate security implications of alternate design approaches.
- To ensure that all parties in a business transaction (transit agency, engineers, supply chain) are viewing security considerations of a proposed design in the same way.
- To have a method to examine the risk associated with different threats.

2.3 Who should do attack modeling?

It is anticipated that the systems integrator will do the attack modeling with help from its vendors, while transit agency control engineers will do the oversight. It should be emphasized that attack modeling is a sophisticated security analysis technique. Personnel from the systems integrator must have experience and training in the techniques and logic behind the analysis and in the software package used; otherwise it must hire a suitable qualified contractor to do the analysis.

2.4 How does attack modeling fit into the APTA document series?

Attack modeling may be used to model and evaluate any section of the SCSZ, FLSZ or OCSZ in a transit control and communication system design. These terms, defined below, have been used extensively in Part II of this series and are also defined in Definitions at the end of this document.

- **SCSZ:** Safety Critical Security Zone
- **FLSZ:** Fire/Life-Safety Security Zone
- **OCSZ:** Operationally Critical Security Zone

2.5 When should attack modeling be used?

Attack modeling could be performed in the preliminary design stage, to evaluate alternative architectures and design solutions, and in the final system design stages, when the vendor's products have been chosen.

The transit agency should establish a trigger level defining when to request that attack modeling be used by the supply chain. A transit agency would define its trigger levels for attack modeling using criteria such as the following:

- **Size and scope of project.** Projects such as a complete new rail line or a complete re-signaling upgrade of an existing line might be considered.
- **The engineering analysis capability of the supply chain,** especially the systems integrator, to perform the analysis. In general, a large systems integrator who has done safety studies using fault trees, and who also has an active control system security analysis and design capability, should be able to acquire the capability to do attack modeling. (If the systems integrator has neither, then it is not realistic to assume that it could acquire the skills for one project.)
- **The magnitude of the cybersecurity question to be answered** or issue to be explored. Generally an analysis might be requested on a first-use technology or on a more automated or advanced technology. It may be a design or technology about which a transit agency has many security questions, requiring a more thorough security analysis. For example:
  - first use of a new type of communications-based train control (CBTC) within an agency that has been using fixed-block signaling only;
  - first use of a safety bus from a signal bungalow or train control room to final signaling elements, like track switches, signal lights (versus point-to-point hard wired connections); or
  - first use of commercial off-the-shelf (COTS) technology or protocols to replace proprietary technology or protocols in a system intended for SCSZ or FLSZ applications.
- **The availability of funds for the analysis,** which is labor-intensive, and the willingness of the transit agency/vendor supply team to put in the extra effort to gain additional assurance and perform the extra analysis that they believe is necessary.
- **The necessity of having all parties to the analysis** (transit agency, engineering consultants, systems integrators and vendors) subscribe to the same cybersecurity philosophy. All parties to the analysis should commit to interacting in a highly cooperative mode for the common goal of increased cybersecurity.

3. Defining attack tree analysis scope

3.1 What to model

Only those sections of a new design or concept needing a thorough analysis to answer the security question posed should be modeled. It is important to note that attack modeling scope is "narrow and deep" — i.e., it focuses on answering a carefully phrased security question relating to a well-defined harmful event at the top of the attack tree. Usually, but not always, this is a worst-case event, such as a collision, derailment or fire. Some sample security questions are given below.

The following is a sample security issue that might be explored:

A transit agency will pioneer the use of a safety bus from a signal bungalow or train control room to final signaling elements, like track switches and signal lights (vs. point-to-point hard-wired connections on a new rail line). What are the additional cyber-risks that might be added over and above the existing hard-wired system? What is a worst-case scenario? What are the countermeasures to prevent this worst case scenario?

3.2 How big an area to model

The attack tree model should be contained to the smallest area of the design that will completely contain the security issue. It is important not to include duplicate networks. If one network will completely illustrate the problem, then create a single model. As an example, do not show more than one signal bungalow if one will suffice.

3.3 Forming an attack tree modeling team

The attack tree modeling team should consist of the following people:

- **Team leader/facilitator:** Should be expert in attack modeling and the use of attack modeling tools. (Ideally the team leader/facilitator would be an employee of the systems integration company or its consultant.)
- **Transit agency cybersecurity representative:** The customer (usually a controls, communications or signal engineer) should be familiar with attack modeling concepts and this *White Paper*.
- **Contributing team members:** Other team members (systems integrator, consultants, vendor technical rep, etc.) should be familiar with attack modeling and attack tree concepts.
- **Subject matter experts (SMEs):** As required.
- **Scribe:** If appropriate.

3.4 Necessary tools

- **Nondisclosure agreement (NDA):** This is necessary because a detailed attack tree can potentially be used as an attack plan if an adversary were able to acquire it.
- **Attack modeling software:** See Section 3.7.
- **Spreadsheets**
- **Network drawing software:** Such as MS Visio® or CAD software.

3.5 Expected time and effort

The time required for the project will vary depending on the scope of the project and the nature of the security question to be answered. Attack modeling in general involves a substantial amount of effort.

3.6 Expected deliverables

Deliverables are to be defined by the transit agency. In general, deliverables should answer the security questions being asked, with attack tree analysis to back up the results to the satisfaction of the transit agency. Deliverables may include, but not be limited to, a management summary, committee minutes and notes, high-level important attack tree scenarios, and all attack tree files.

The technical results should be given at a high enough level that management will relate them directly to operational activities and executive decision making.

3.7 Attack modeling software

Attack modeling software has been designed and built expressly for the creation, modification and analysis of attack trees. Commercial products such as SecurITree® by Amenaza Technologies Limited and Isograph+, or open-source applications such as ADTool and Poseidon Community Edition, are available.

Amenaza Technologies Limited graciously provided training for the APTA subgroup and allowed APTA's working group free use of the SecurITree® software in the development of this *White Paper*. The rail attack trees compatible with the Amenaza software are available at http://www.amenaza.com/APTA-WhitePaper.php. These templates will be available on Amenaza's website for a minimum of three years following publication of this *White Paper*.

3.8 Attack modeling process

Once the decision has been made to answer the security issue/question by using attack modeling, and a team has been formed and trained, the process outlined in this section should be followed.

3.8.1 Characterize the system

For this step, the following materials should be prepared:

- **Asset list:** Identify all equipment of interest, including hardware, software, physical enclosures, and wiring networks, including temporary data storage items and computing equipment like laptops that remain on premises or are introduced temporarily to the area being modeled.
- **Network diagram:** Draw or obtain network diagrams to completely describe how the assets in the area being modeled are hooked up electronically.
- **Trust boundaries:** A trust boundary is an imaginary logical and/or physical perimeter that is drawn around the area being modeled to define a zone (as previously defined in Part II of this document series) and separate it from other security zones. For example, the SCSZ zone would have a trust boundary as its perimeter.
- **Entry and exit points:** These are wired or wireless connections that cross a trust boundary, where data flows in and out of the zone.
- **Data format and data exchange chart:** Create a list of all data that flows between electrical and computing assets of interest, along with the electrical and networking protocols that are being used. For instance, an Ethernet line using a serial-based signaling protocol tunneled over TCP/IP would have five layers of protocol:
  - physical layer (e.g., CAT 5 Ethernet cable)
  - link layer
  - IP layer
  - TCP layer
  - application layer (perhaps containing signaling information serial protocol tunneled over TCP/IP)
- **Assumptions:** Determine any assumptions that will qualify or place limitations on the results of the analysis. For instance, one may assume that all "insiders" — e.g., company employees — are completely trustworthy, because all employees who work in the area being modeled have been with the transit agency more than 15 years and were originally security screened. Please note that these assumptions, when compared with the real world, may not turn out to be valid. In fact, as part of the attack modeling process, the original (and perhaps simplifying) assumptions should be challenged. Another assumption might be the maximum skill level of an attacker. The modeling may assume only a medium skilled hacker as a potential attacker, never an expert. Another assumption that may have to be made is that areas similar to the modeled area will be wired and configured correctly — i.e., the analysis will not take into account a wiring mistake in units that are supposed to be built and wired

identically. During attack modeling, it is a good practice to challenge the original assumptions to see if they are realistic in light of what the attack tree reveals.

3.8.2 Describe normal sequence of operations, along with data flows

- **Startup/shutdown:** To do a thorough analysis, one must go through the normal lifecycle of the area of the system being modeled, not just normal operations when everything is working correctly. The startup and shutdown sequence is particularly important. The startup/shutdown sequences may open up vulnerabilities in the system that might be closed during normal operation. Likewise, when the system is shut down, connections must be examined. Will the system be isolated from other systems, or will it be more vulnerable than when in normal operation?
- **Normal operation:** The system will presumably spend most of its time here, if it is a production and not a backup, testing or contingency system.
- **Foreseeable failure modes:** The analysis should include failure modes of operation, such as when the primary server is down and the backup is the master.
- **Maintenance/troubleshooting/component hardware and software updates:** Just as with startup and shutdown, it is necessary to look at the state of the system during troubleshooting and maintenance, particularly if hardware and software updates are being implemented.

3.8.3 Decompose operations into sequence diagrams

A sequence diagram may be an actual diagram, such as a flowchart, with successive operations in boxes or blocks. If it is a simple sequence without branches, it may be done with text as a numbered list. In all cases, it should show the sequence of steps in a process in enough detail to proceed with the security analysis but skip unneeded detail.

3.8.4 Identify threats to system during operating sequences

- **From an authorized user:** It is assumed the authorized user may make accidental mistakes, such as deleting data, issuing the wrong command or making a programming error. If this happens, it should be assumed that there is no deliberate intent on the part of the user to harm the system. If there is such intent, then this authorized user becomes a “malicious insider” on the system.

  NOTE: There may be a gray area here, where an authorized user circumvents system security features in attempt to do his or her job faster or more conveniently but does not intend to deliberately harm the system. A thorough attack modeling process may need to account for this type of behavior.

- **From a malicious insider:** An “insider” is any person with privileged knowledge and/or access to systems and locations not open to the public, such as computer rooms, signal bungalows, control rooms or remote-access connections to transit systems. An insider threat may come from a full-time or part-time employee, a cleaning or maintenance person with access to areas closed to the public, or a contractor who comes onsite. It may be an employee who has just been discharged or laid off, or who anticipates such actions and harbors ill will. Ill will toward the transit agency or its employees, may lead a malicious insider to take harmful actions, such as equipment sabotage, hacking, data loss or creating a safety hazard.
- **From a malicious outsider:** An “outsider” is any person who, unlike an insider, does not have any privileged knowledge or access to a transit agency’s property or systems. This would include a trespasser, a computer hacker trying to get access to a transit agency’s computers or networks, or anyone who would sabotage a transit agency’s property normally accessible to the public, such as trackwork or switch machines. Harmful intent and actions turn an outsider into a malicious outsider.

3.8.5 Build attack trees

Attack trees should be built for each of the scenarios listed below. Any other combination of causes, both accidental and malicious, may be included:

- **Accidental misuse by an authorized user:** Working downward from the top (worst-case) event, add nodes and branches for the accidental misuse/damage to the system under study.
- **Malicious insider and outsider:** Continue the tree by adding nodes and branches for malicious activity by malicious insiders and outsiders.
- **Collusion between malicious insider and outsider:** Add possibilities for the malicious insider and outsider in league with each other, if such cases are not covered by considering these adversaries separately.
- **Physical and cyber possibilities:** Even though this analysis and document concentrates on the cyber side of control and communications security, do not limit actions only to cyberspace. For instance, consider a forced entry into a computer room or signal bungalow, followed by malicious cyber activity, for instance at a keyboard or PLC.

3.8.5.1 Assumptions for building attack trees

- While building the tree and reviewing the interim products, it is always good for the attack modeling team to question the original assumptions made in light of what the tree is revealing. For instance if an assumption was made that "all our employees are perfectly loyal," it is worth going through a "what-if" scenario and asking, "What if this assumption is wrong? What would happen to my attack tree?" and examining the consequences. It may be that the team decides to change an original assumption, and the tree, after consideration of the "what-ifs."
- Assume probability is 1 for all nodes and paths. In other words, consider all events (leaf nodes and branches) to be equally likely and add them to the tree if they are possible. Wait until the tree is completely drawn to consider the probabilities of these events happening and possibly eliminating them as improbable (i.e., "pruning the tree").

3.8.6 Decision point: evaluation type

Once the tree is drawn, the attack modeling team should make a decision on the type of evaluation to be implemented: to work with attack scenarios qualitatively (without assigning relative probabilities) with the short method, or to proceed with full (and more lengthy) analysis using risk/probability features of attack tree software.

- To proceed without assigning relative probabilities (short method), proceed with the next section in this document (3.8.7).
- If the decision is reached to work with full risk analysis of the attack tree software (long method), then continue the evaluation using the method available by request from Amenaza's SecurITree website: https://www.amenaza.com/request_methodology.php.

3.8.7 Complete the analysis using the short method

- Review each attack scenario and choose an agreed-upon percentage of the most-likely scenarios using engineering judgment, any available historical data and the consensus of the team.
- Make note of other scenarios for the project record.
- Brainstorm on applying countermeasures for the remaining scenarios. A project report will include a list of the most likely scenarios, a list of suitable countermeasures and, for the record, the list of scenarios the team judged less likely.

4. Illustrating the attack modeling process, a case study

4.1 Background on case study: The security question to be answered

A transit agency in the Northeast United States with a conventional fixed-block signaling system has had an incident in which the wrong binary file was loaded on a vital PLC in a signal bungalow by accident. Luckily the substitution was caught at the last minute during final testing at the interlocking — with trains on the track. If the error had not been caught, a train derailment might have occurred under a specific combination of train routings.

The manager of Signals and Communications Engineering knew he needed to add some precautions and in-process checks to ensure that the wrong file could not be loaded by accident again, but he had an additional thought — what if a malicious insider or outsider wanted to cause a train derailment (in the worst case)? Could that be done, and, if so, how difficult would it be?

Since the signal system was due for a routine five-year refurbishment and incremental upgrade of legacy components in a few months, the manager thought this would be a good time to do the following:

- Ask the question and answer it using APTA's attack modeling procedure; and
- Consider adding security controls if the analysis indicated that the cyber-sabotage would be "too easy" and therefore a tempting target for the wrong people.

Although the security question was primarily cyber in nature (substituting a malicious file), certain factors were considered that would involve physical events. Referring to Section 4.3.1 ("Physical layout"), such events would include unauthorized activities of a malicious insider within the signal engineer's office, or unauthorized entry into the signal bungalow by an outsider.

> **NOTE:** For some perspective on how other factors, both cyber and physical, may cause a derailment at the interlocking, refer to the attack tree in **Figure 6**, which looks at other malicious events that may cause a derailment (obstructing the rail, physically tampering with the switch, etc.). This figure gives a "40,000-foot view" on what could cause a derailment.

4.2 Assumptions

Using the case study above, involving a generic transit agency, three physical areas of concern arise:

- Interlocking with track, signals, track circuits (**Figure 1**).
- Signal bungalow, one of 20 identical bungalows on a surface line (**Figure 2** and **Figure 3**).
- Signal engineering office area, where signal programs for the vital PLCs are created, tested and checked (**Figure 4**).

Based on this initial information, a scenario applying the attack modeling procedure described in Section 3 can be developed.

4.3 Scenario development and physical layout

4.3.1 Physical layout

This agency in a medium-sized city has commuter rail lines extending out into the city suburbs. There is a single interlocking with one switch and a signal bungalow servicing that switch. A drawing of the interlocking is in **Figure 1**. The interlocking is serviced by a signal bungalow (**Figure 2**), and the network diagram for the signal bungalow is shown in **Figure 3**.

Figure 4 shows the relative locations of the signal bungalow, the Operations Control Center (OCC), and train station, while **Figure 5** shows the signal engineering office LAN, located in the OCC.

FIGURE 1
Interlocking

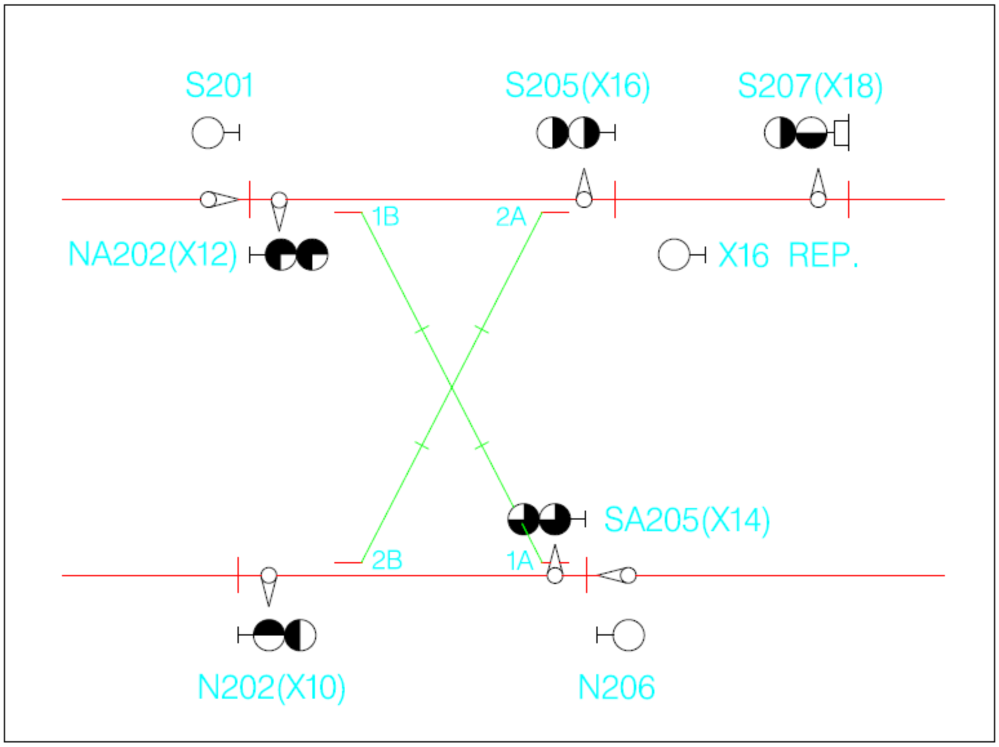


FIGURE 2
Signal Bungalow

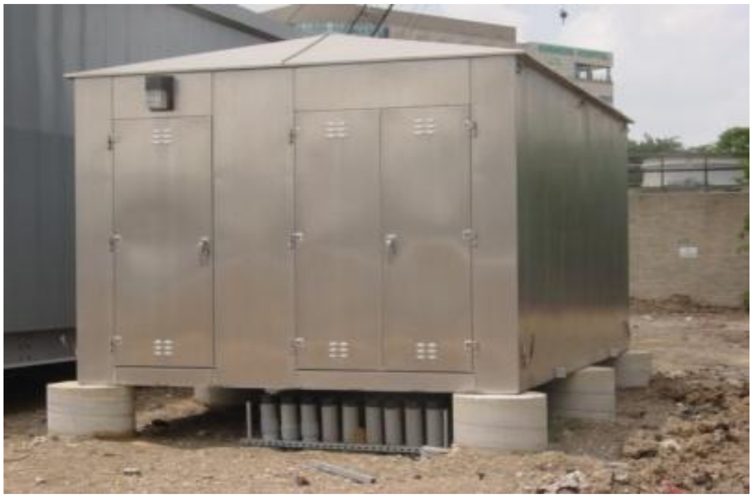


FIGURE 3
Signal Bungalow Network

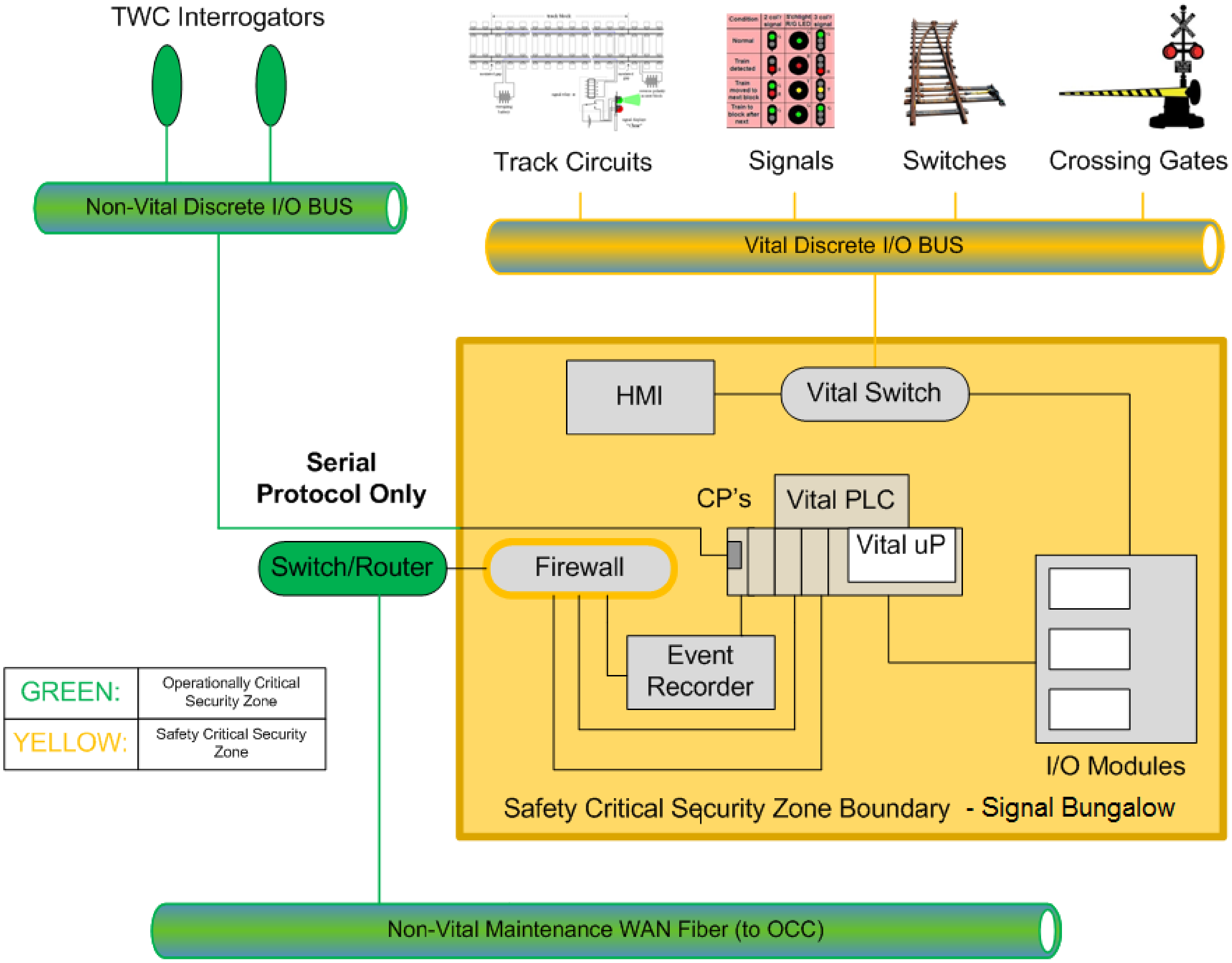

TWC Interrogators
Non-Vital Discrete I/O BUS
Track Circuits
Signals
Switches
Crossing Gates
Vital Discrete I/O BUS
HMI
Vital Switch
Serial
Protocol Only
CP's
Vital PLC
Vital uP
Switch/Router
Firewall
Event
Recorder
I/O Modules
Safety Critical Security Zone Boundary - Signal Bungalow
GREEN: Operationally Critical Security Zone
YELLOW: Safety Critical Security Zone
Non-Vital Maintenance WAN Fiber (to OCC)

FIGURE 4
Security Zones

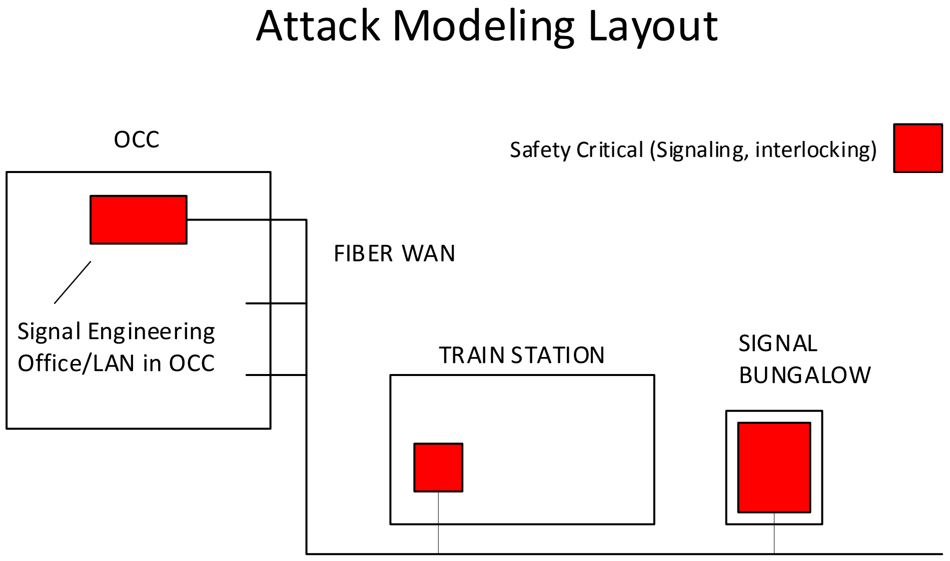
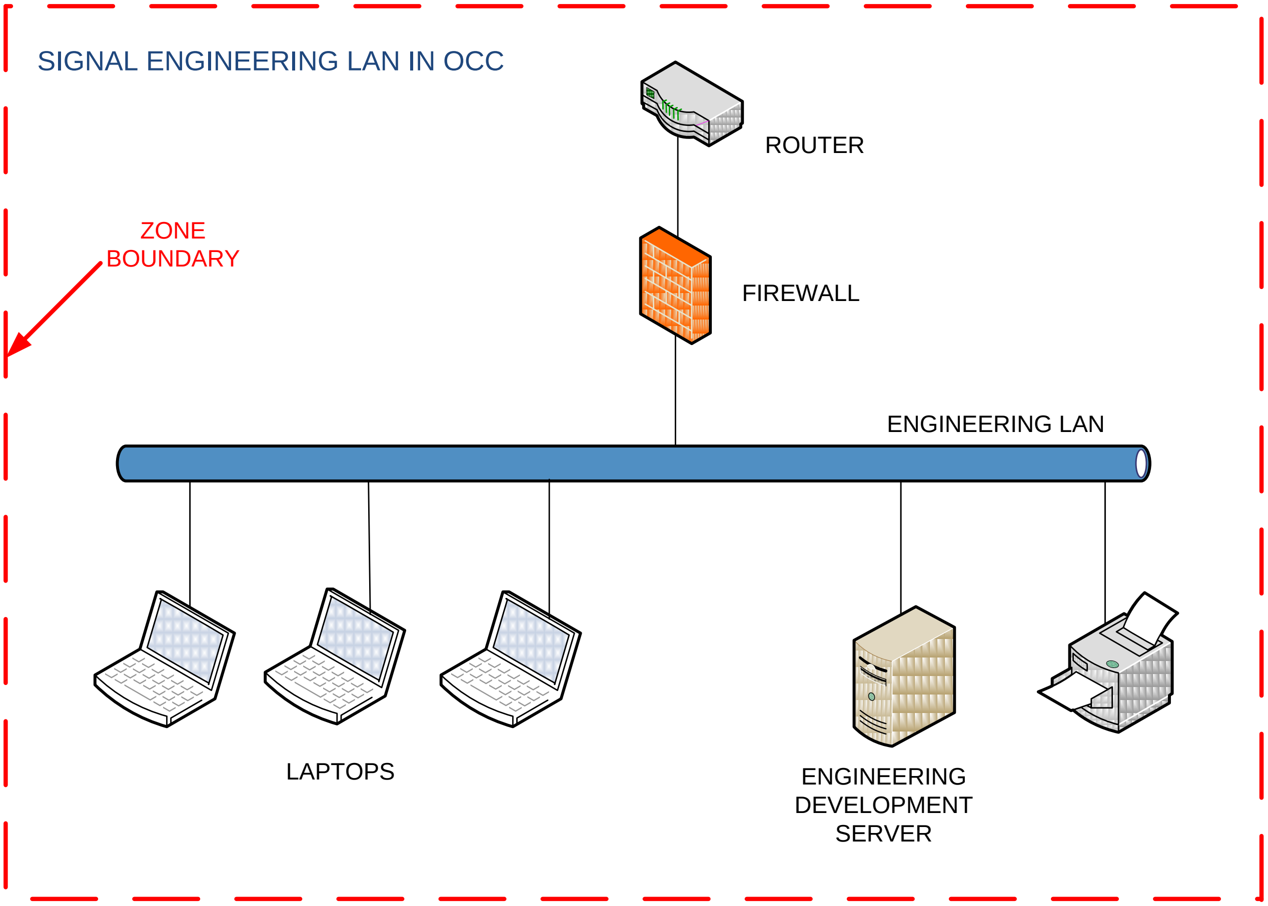

Attack Modeling Layout
OCC
Safety Critical (Signaling, interlocking) 🟥
FIBER WAN
Signal Engineering Office/LAN in OCC
TRAIN STATION
SIGNAL BUNGALOW

FIGURE 5
Signal Engineering Office LAN



4.3.2 Signal engineer staffing

There are two permanent, full-time signal engineers, who have been with the railway for more than 15 years, Signal Engineers A and B. In addition, the signal contractor sends the railway a “floater,” a vendor employee (Signal Contractor 1) who comes in during peak workload times and substitutes for Signal Engineers A and B when they are absent. Signal Contractor 1 works at several different railways during the year and has been filling in at the present railway for a period of two years.

4.3.3 Signal engineering office

The local area network in the signal engineering office is shown in **Figure 5**. Signal Engineers A and B each have a docking station for their laptops (Laptops 1 and 2), while the remaining laptop, Laptop 3, has its Docking Station 3 and is used by Signal Contractor 1, or is used as a spare backup by either signal engineer when Signal Contractor 1 is not there. All laptops are locked up in Desks 1, 2 and 3 overnight and on weekends. When a signal engineer or signal contractor goes on the road to visit signal bungalows and train stations, the laptops are locked up in the trunk of his car.

4.3.4 Signal engineering office LAN

Figure 5 also shows the overall network configuration of the signal engineering office local area network (LAN). The LAN is protected from outside networks by a router/firewall/switch combination, which allows email and Web access.

4.4 Attack modeling case study

This section outlines the APTA attack modeling method for the case study just outlined.

4.4.1 Characterize system

4.4.1.1 Network diagrams

- signal bungalow network (**Figure 3**)
- overall layout (**Figure 4**)
- signal engineering office (**Figure 5**)

4.4.1.2 Trust boundaries and entrance/exit points

Signal bungalow (Figure 3)

- From non-vital discrete I/O bus into signal bungalow SCSZ (entrance).
- OCSZ (green) connection from non-vital maintenance wide area network (WAN) switch/router to SCSZ firewall (entrance and exit).
- From SCSZ vital switch (entrance/exit).
- Signal engineer physical entry to signal bungalow with laptop to update vital PLC firmware (entrance).

Signal engineering office LAN (Figure 5)

- From enterprise network into signal engineering office LAN (entrance/exit).
- Laptop physical removal/return to docking stations (entrance/exit).
- Removable media (CDs, flash drives, etc.) to and from signal engineering laptop/LAN (entry/exit).

4.4.1.3 Data format and data exchange chart

Signal bungalow (Figure 3)

- From non-vital discrete I/O bus into signal bungalow SCSZ (entrance) to serial protocol from TWC interrogators.

- OCSZ (green) connection from non-vital maintenance WAN switch/router to SCSZ firewall to Ethernet connection layers 1 and 2, TCP/IP protocol layers 3 and 4, Genisys/TCP protocol layers 5-7.
- From SCSZ vital switch to safety bus protocol, such as Profisafe.
- Signal engineer physical entry to signal bungalow with laptop to update vital PLC firmware with application binary.

Signal engineering office LAN (Figure 5)

- From OCC router to signal engineering firewall/switch to Ethernet connection layers 1 and 2, TCP/IP protocol layers 3 and 4, application layers 5-7.
- Manual movement of signal engineer's laptops in and out of docking stations.
- Manual movement of CDs, memory sticks, other electronic storage devices in and out of signal engineering office, including file transfer to/from laptops.

4.4.1.4 Assets

Signal bungalow (Figure 3)

- firewall
- event recorder
- vital PLC
- I/O modules
- vital switch
- HMI

Signal engineering office LAN (Figure 5)

- firewall/switch
- Docking Stations 1, 2 and 3
- Laptops 1, 2 and 3
- engineering development server
- network printer

4.4.1.5 Assumptions (physical security, etc.)

Assume physical security to prevent unauthorized entry into the signal bungalow is active. Physical security for the signal bungalow includes two-factor authentication, such as a key to enter the door plus a keypad inside. Physical security for the signal engineering office is a card swipe or key to the office door.

4.4.2 Describe normal sequence of operations

4.4.2.1 Startup/shutdown

- Security concerns with startup/shutdown occur at the time when vital PLC files could be changed. These are identified in the "Maintenance/updates" section below.

4.4.2.2 Normal operation

- Dispatch signals come from the OCC ATS/Dispatch system over the (green) non-vital maintenance LAN using the Genisys/TCP protocol. To enter the signal bungalow, they pass through the firewall to communications input modules on the vital PLC. (At the same time, signals go from the PLC communications output module the reverse path through the firewall back to the OCC ATS system with TWC information to identify the trains, etc.). Any input signal prompting action runs through the logical interlocks of the vital processor. Then verified outputs go from the vital microprocessor, through the vital switch to the vital I/O bus for actions by switches and signals.

- As indicated above, signals come in from the TWC interrogators via serial protocol to the signal bungalow and the non-vital communications module on the vital PLC. These signals get recorded by the event recorder and go back out through the firewall and Interface 2 to the maintenance WAN back to the control room ATS system for monitoring and display.

4.4.2.3 Maintenance/updates

At some point, the two firmware files on the vital PLC (executive logic and application binary) will need a change. Sometimes new executive logic files are issued as an update by the signal vendor, and periodically the signal engineers or signal contractor will need to make a change or update in the application binary.

All coding and compiling for vital PLC binaries takes place in the signal engineering office. The new or changed files are then transported on the signal engineer's/contractor's laptop to the bungalows, where they are downloaded onto the PLC.

The following sections identify the steps that a signal engineer or contractor will go through to create, install and run new files on the vital PLC in the signal bungalows.

4.4.2.4 New executive logic update from signal vendor

New vital PLC executive logic firmware update is delivered from signal vendor:

1. Receive certified executive logic update from signal vendor on storage medium (disk, flash drive, etc.).
2. Load on signal engineer's laptop.
3. Bring laptop from signal engineering office to bungalow.
4. Take vital PLC off-line, switch it to "update executive logic" mode.
5. Download the update.
6. Check for proper functioning of vital PLC with new executive logic.
7. Any other additional checks before return to normal operation.

4.4.2.5 New application logic update from signal engineers

These steps take place in the signal engineering office and during transportation:

1. Signal engineer identifies nature and reason for change.
2. Load "old" source file on signal engineer's laptop.
3. Modify source file.
4. Compile and test new code on laptop.
5. Test new code on signal system simulator.
6. Crosscheck and peer review of new code.
7. Finalize changes, file new source and binary code on laptop, and duplicate copy on signal engineering server.
8. Remove laptop from docking station; bring to signal engineer's car.
9. Transport laptop in signal engineer's car to the signal bungalow.

4.4.2.6 New application logic update from signal engineers

These steps take place in the signal bungalow:

1. Signal engineer arrives at signal bungalow with new application binary loaded on laptop.
2. Engineer takes vital PLC off-line, ready to receive new application logic binary.
3. Engineer downloads new file.

4. Engineer does integrity check of new file.
5. Engineer and maintenance technician run internal test of vital PLC with the new logic program installed, using jumpers, test settings, etc. for other equipment in the signal bungalow.
6. Testing with actual trains.
7. Put interlocking/new logic program into production; monitor first few days of operation.

4.4.3 Decompose operations

It is sufficient to document the above sequences in words, since the process would be serial in nature (there are no branching operations). A process flow diagram would be necessary for a process with branching operations.

4.4.4 Identify threats

- **Accidental threat by authorized user:** A trusted signal engineer makes an error in creating new vital PLC application instructions, creating an accident-prone source and/or binary file.
- **Malicious insider:** A malicious signal contractor creates or substitutes malicious source code and/or binary, which the contractor intends to install/substitute on the vital PLC in the signal bungalow.
- **Malicious outsider:** A malicious outsider creates malicious source code and/or binary and installs/substitutes it on vital PLC in the bungalow by some means, including physical entry to bungalow.

4.4.5 Build attack trees

- For educational purposes, a “master event” tree with top node as “derailment of train at interlocking” was built to provide an overview and context for the case study tree, displaying several different ways a train could be derailed, including mechanical tampering, breaking into the bungalow and installing jumpers, etc.
- Then the “Unsafe file on vital PLC” subsection of the tree for the actual attack modeling analysis is built by extending downward from the top node, giving all sequences or attack paths that could lead to the top node occurring. This is how the attack tree is used to answer the security question and is the only tree that will be used for further analysis.

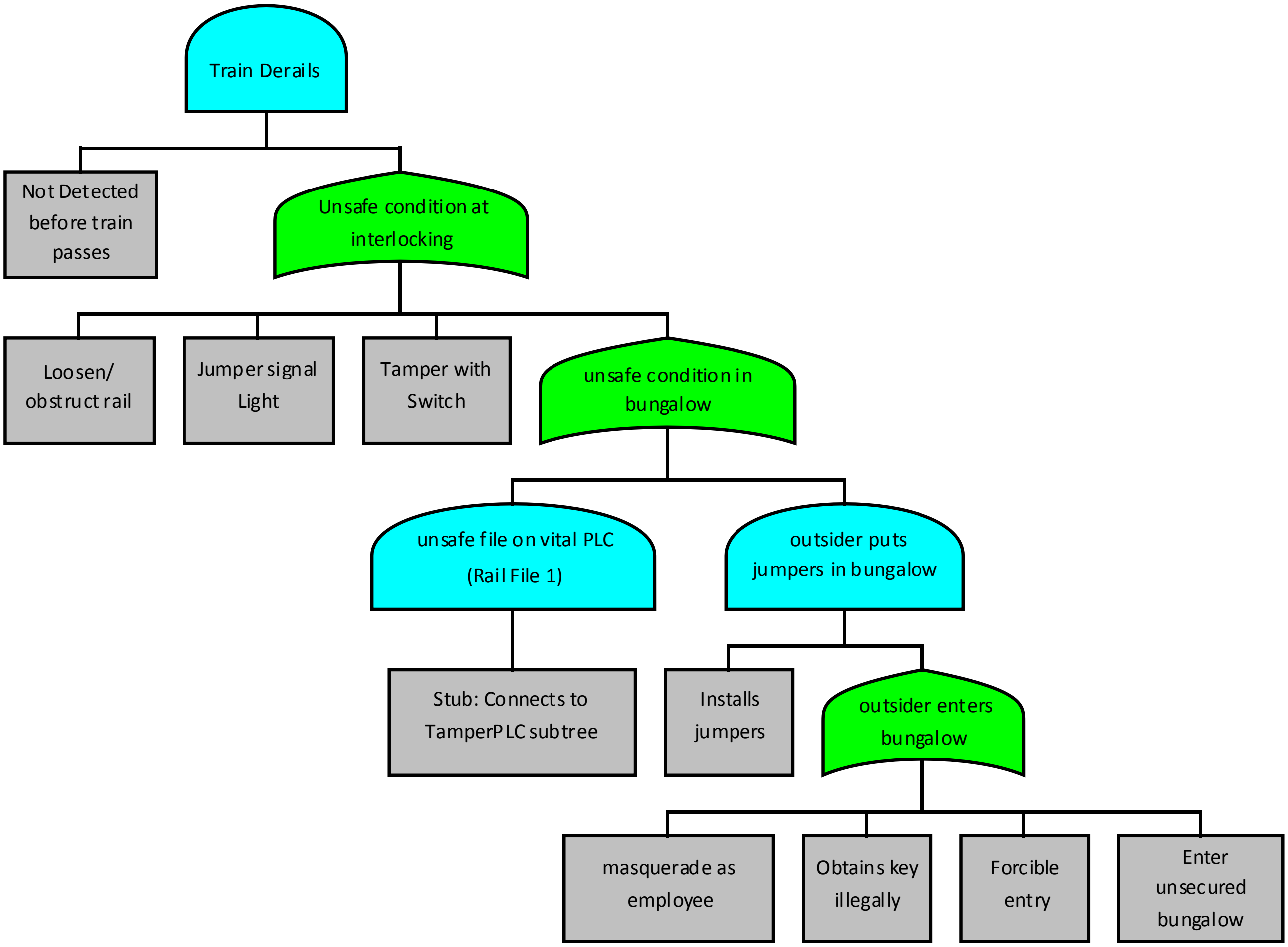
4.4.6 Derailment attack tree

For educational purposes, and to show context for case study, the entire attack scenario of many derailment possibilities at an interlocking, including physical and cyber interventions is given below.

Note the color scheme of the Boolean symbols used in the attack tree in **Figure 6**:

- **AND nodes,** where both inputs (if they exist) have to be present to have an output, are blue.
- **OR nodes,** where either input must be present to have an output, are green.
- **Leaf nodes,** which contain an action or an event, are gray.

FIGURE 6
Overall Derailment Attack Tree


Train Derails
Not Detected before train passes
Unsafe condition at interlocking
Loosen/ obstruct rail
Jumper signal Light
Tamper with Switch
unsafe condition in bungalow
unsafe file on vital PLC (Rail File 1)
outsider puts jumpers in bungalow
Stub: Connects to TamperPLC subtree
Installs jumpers
outsider enters bungalow
masquerade as employee
Obtains key illegally
Forcible entry
Enter unsecured bungalow

4.5 Explanation of Figure 6

Examining the attack tree in **Figure 6**, a derailment at the interlocking may result from the following actions:

- Tamper with rail itself (loosen section, cut rail, remove section, etc.).
- Tamper the signal light to always read green.
- Tamper with switch.
- Break into signal bungalow, install jumpers.
- Install an unsafe application file on the vital PLC (master scenario under study).

For the case study, “detach” the “Unsafe file on vital PLC” in the signal bungalow from **Figure 6** and show the entire attack tree underneath (**Figure 7** and **Figure 8**). This becomes the master tree for our analysis. Once again, this tree is built downward from the top event, through the detrimental actions of the three actors described. Please notice that the malicious file that winds up on the vital PLC has to be created and then transferred to the signal bungalow to cause the top event.

FIGURE 7
Attack Tree (Left Side)

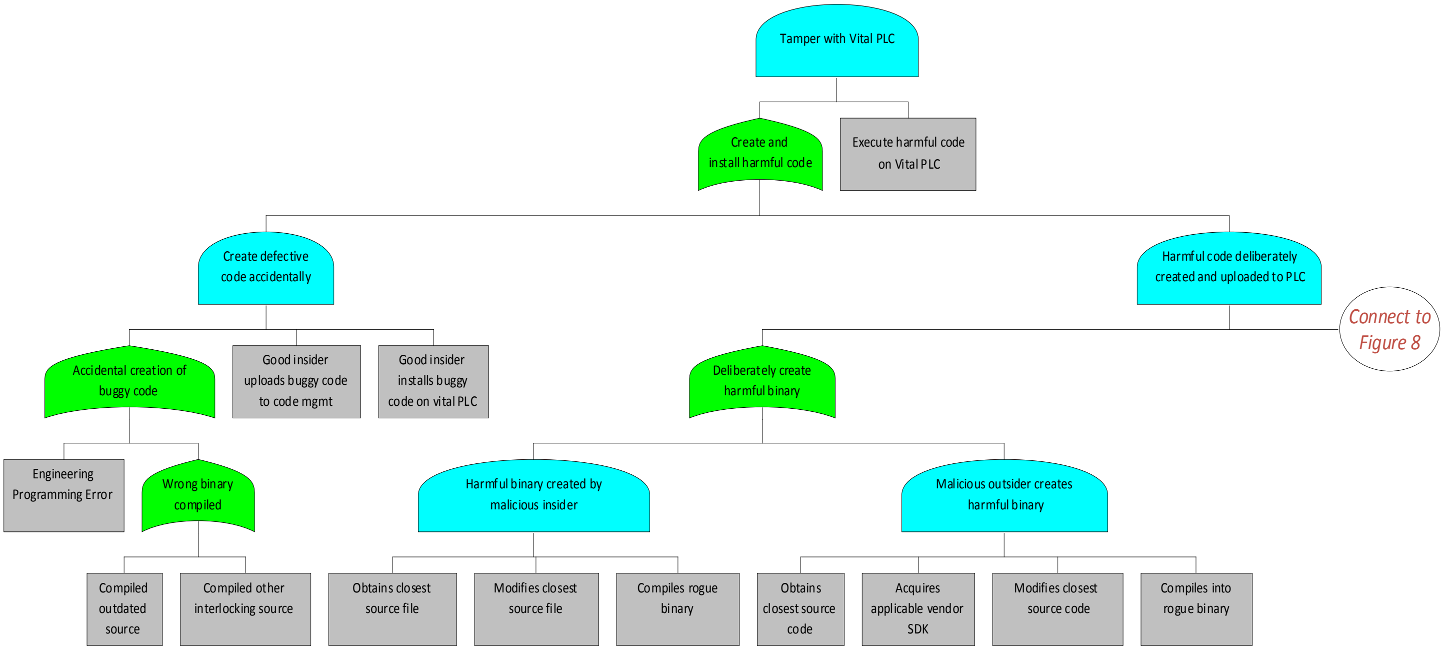
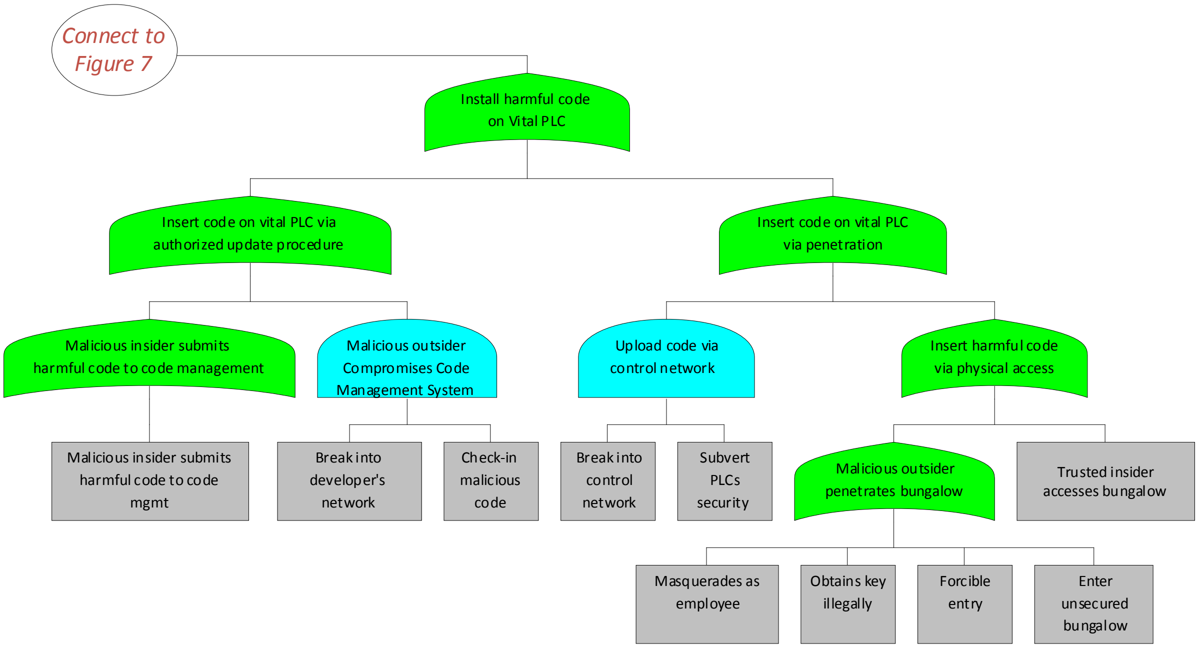

Tamper with Vital PLC
Create and install harmful code
Execute harmful code on Vital PLC
Create defective code accidentally
Harmful code deliberately created and uploaded to PLC
Connect to Figure 8
Accidental creation of buggy code
Good insider uploads buggy code to code mgmt
Good insider installs buggy code on vital PLC
Deliberately create harmful binary
Engineering Programming Error
Wrong binary compiled
Harmful binary created by malicious insider
Malicious outsider creates harmful binary
Compiled outdated source
Compiled other interlocking source
Obtains closest source file
Modifies closest source file
Compiles rogue binary
Obtains closest source code
Acquires applicable vendor SDK
Modifies closest source code
Compiles into rogue binary

FIGURE 8
Attack Tree (Right Side)


Connect to Figure 7
Install harmful code on Vital PLC
Insert code on vital PLC via authorized update procedure
Insert code on vital PLC via penetration
Malicious insider submits harmful code to code management
Malicious outsider Compromises Code Management System
Upload code via control network
Insert harmful code via physical access
Malicious insider submits harmful code to code mgmt
Break into developer's network
Check-in malicious code
Break into control network
Subvert PLCs security
Malicious outsider penetrates bungalow
Trusted insider accesses bungalow
Masquerades as employee
Obtains key illegally
Forcible entry
Enter unsecured bungalow

4.5.1 Create minimum set of attack paths or attack scenarios using routines in the attack tree software

Figure 9 shows the listing provided by the SecurITree software given the attack tree as shown in **Figure 7** and **Figure 8**.

FIGURE 9
Attack Scenarios

SCENARIO	ATTACK PATH							
1	Engineering Programming Error	Good insider uploads buggy code to code management		Good insider installs buggy code on vital PLC				Execute harmful code on Vital PLC
2	Compiled outdated source	Good insider uploads buggy code to code management		Good insider installs buggy code on vital PLC				Execute harmful code on Vital PLC
3	Compiled other interlocking source	Good insider uploads buggy code to code management		Good insider installs buggy code on vital PLC				Execute harmful code on Vital PLC
4	Obtains closest source file	Modifies closest source file	Compiles rogue binary			Malicious insider submits harmful code to code management		Execute harmful code on Vital PLC
5	Obtains closest source file	Modifies closest source file	Compiles rogue binary		Break into developer's network	Check-in Malicious code		Execute harmful code on Vital PLC
6	Obtains closest source file	Modifies closest source file	Compiles rogue binary		Break into control network	Subvert PLC's security controls and install malware		Execute harmful code on Vital PLC
7	Obtains closest source file	Modifies closest source file	Compiles rogue binary			Masquerades as employee		Execute harmful code on Vital PLC
8	Obtains closest source file	Modifies closest source file	Compiles rogue binary			Obtains key illegally		Execute harmful code on Vital PLC
9	Obtains closest source file	Modifies closest source file	Compiles rogue binary			Forcible entry		Execute harmful code on Vital PLC
10	Obtains closest source file	Modifies closest source file	Compiles rogue binary			Enter unsecured bungalow		Execute harmful code on Vital PLC
11	Obtains closest source file	Modifies closest source file	Compiles rogue binary			Trusted insider accesses bungalow		Execute harmful code on Vital PLC
12	Obtains closest source code	Acquires applicable vendor SDK	Modifies closest source code		Compiles into rogue binary	Malicious insider submits harmful code to code management		Execute harmful code on Vital PLC
13	Obtains closest source code	Acquires applicable vendor SDK	Modifies closest source code		Compiles into rogue binary	Break into developer's network	Check-in Malicious code	Execute harmful code on Vital PLC
14	Obtains closest source code	Acquires applicable vendor SDK	Modifies closest source code		Compiles into rogue binary	Break into control network	Subvert PLC's security controls and install malware	Execute harmful code on Vital PLC
15	Obtains closest source code	Acquires applicable vendor SDK	Modifies closest source code		Compiles into rogue binary	Masquerades as employee		Execute harmful code on Vital PLC
16	Obtains closest source code	Acquires applicable vendor SDK	Modifies closest source code		Compiles into rogue binary	Obtains key illegally		Execute harmful code on Vital PLC
17	Obtains closest source code	Acquires applicable vendor SDK	Modifies closest source code		Compiles into rogue binary	Forcible entry		Execute harmful code on Vital PLC
18	Obtains closest source code	Acquires applicable vendor SDK	Modifies closest source code		Compiles into rogue binary	Enter unsecured bungalow		Execute harmful code on Vital PLC
19	Obtains closest source code	Acquires applicable vendor SDK	Modifies closest source code		Compiles into rogue binary	Trusted insider accesses bungalow		Execute harmful code on Vital PLC

4.5.2 Decision point: Evaluation type

- A team decision is needed on whether to work with attack scenarios qualitatively (without assigning relative probabilities) or to proceed with full analysis using risk/probability features of attack tree software.
- To proceed qualitatively, review each attack scenario, choose only an agreed-upon percentage of the more likely scenarios using a consensus of the modeling team, record the others, and decide on applying countermeasures for the scenarios that were considered the most likely. This document assumes the team decided to use the short method.
- If a decision is reached to work with full risk analysis capability of software, continue using the method available by request from Amenaza's SecurITree website: https://www.amenaza.com/request_methodology.php. For purposes of this *White Paper*, this is known as the long method.

5. Attack scenario analysis (short method)

Using **Figure 7** and **Figure 8** and the SecurITree software, produce a list of attack paths or attack scenarios (shown as the chart in **Figure 9**). This represents all the different unique paths the system can be attacked by, listing all the valid input combinations for that path.

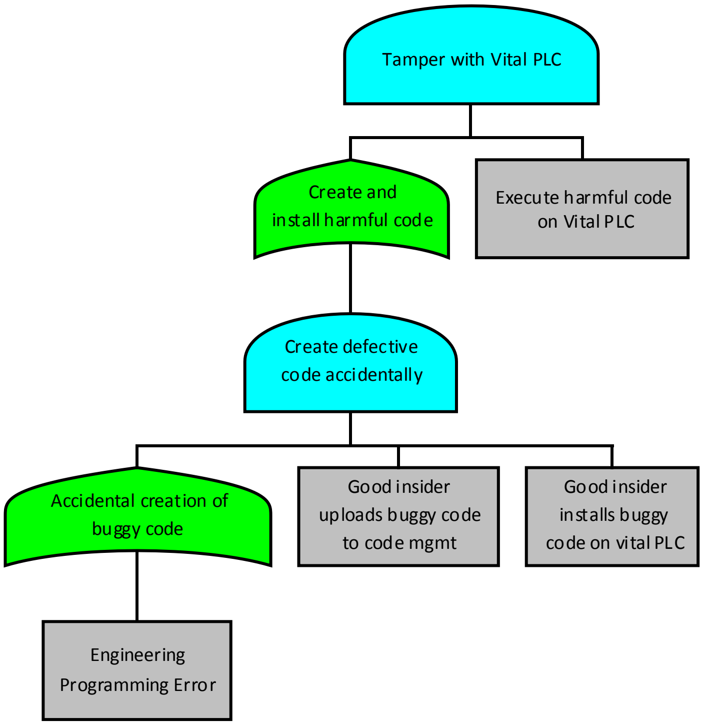
Using this list of attack scenarios (**Figure 9**), examine the individual attack paths and choose four or five of the most likely scenarios for countermeasure analysis.

- Using the team's engineering judgment, scenarios 1, 5, 8 and 16 were chosen as "most likely."
- The four attack trees associated with the above scenarios are shown in the following section.

5.1 Describe in detail the selected higher-risk scenarios

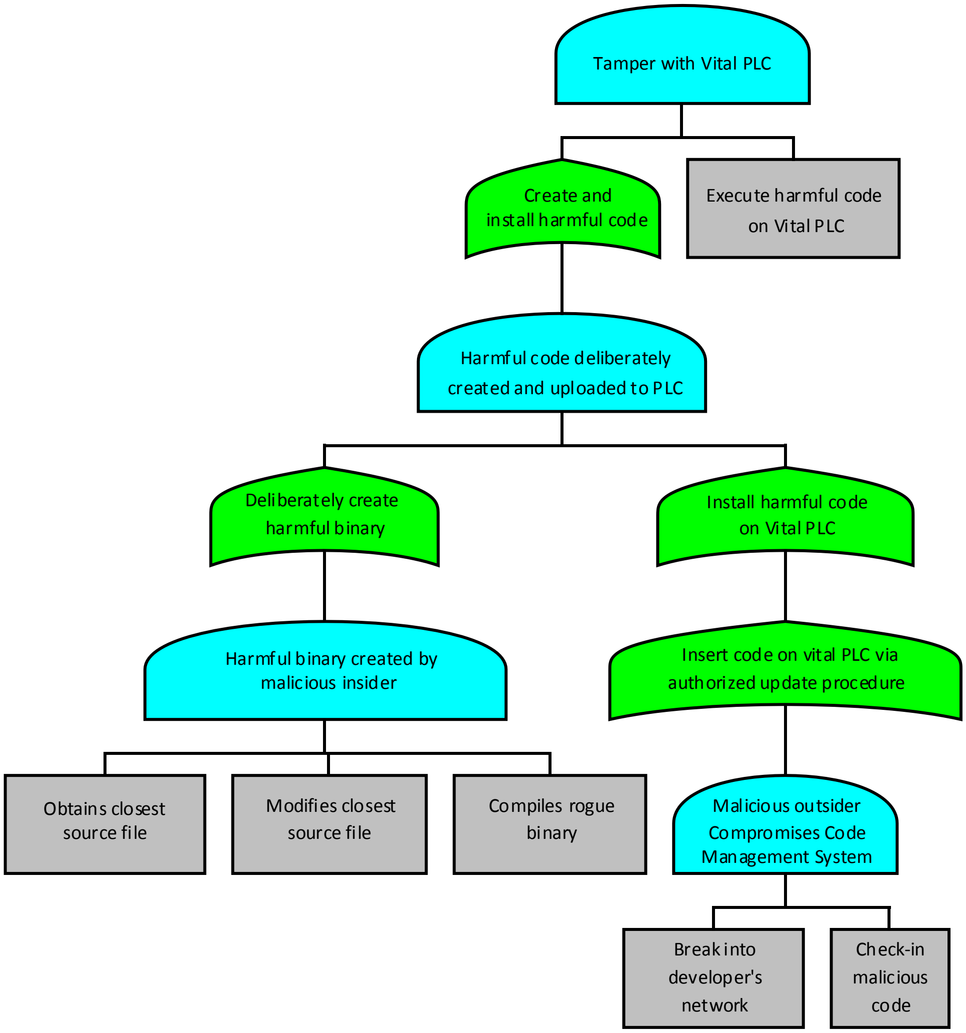
- **Scenario 1:** Loyal employee accidently creates buggy code in application source, compiles into binary, doesn't catch the error in simulation and testing in office and on-site, error is not caught, derails train. See **Figure 10**.

FIGURE 10
Attack Scenario 1


Tamper with Vital PLC
Create and install harmful code
Execute harmful code on Vital PLC
Create defective code accidentally
Accidental creation of buggy code
Good insider uploads buggy code to code mgmt
Good insider installs buggy code on vital PLC
Engineering Programming Error

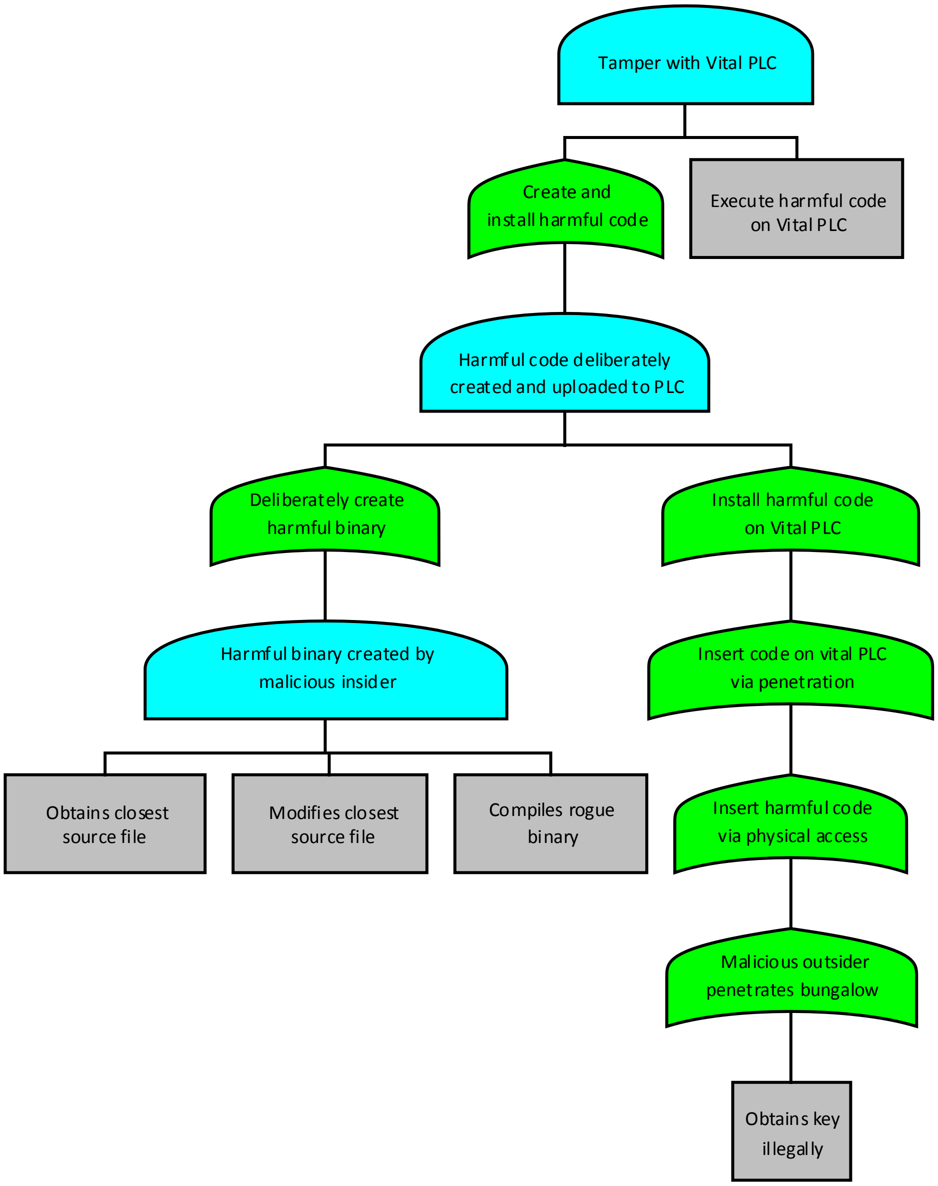
- **Scenario 5:** Malicious insider (Signal Contractor 1) in collusion with a malicious outsider. The malicious insider takes existing (correct) source files and modifies them to cause a derailment. An outside partner then hacks into the signal developer's network to substitute this for the correct file. This way the malicious insider does not have log file trace back to himself; instead it looks like the work of an intruder acting alone. See **Figure 11**.

FIGURE 11
Attack Scenario 5


Tamper with Vital PLC
Create and install harmful code
Execute harmful code on Vital PLC
Harmful code deliberately created and uploaded to PLC
Deliberately create harmful binary
Install harmful code on Vital PLC
Harmful binary created by malicious insider
Insert code on vital PLC via authorized update procedure
Obtains closest source file
Modifies closest source file
Compiles rogue binary
Malicious outsider Compromises Code Management System
Break into developer's network
Check-in malicious code

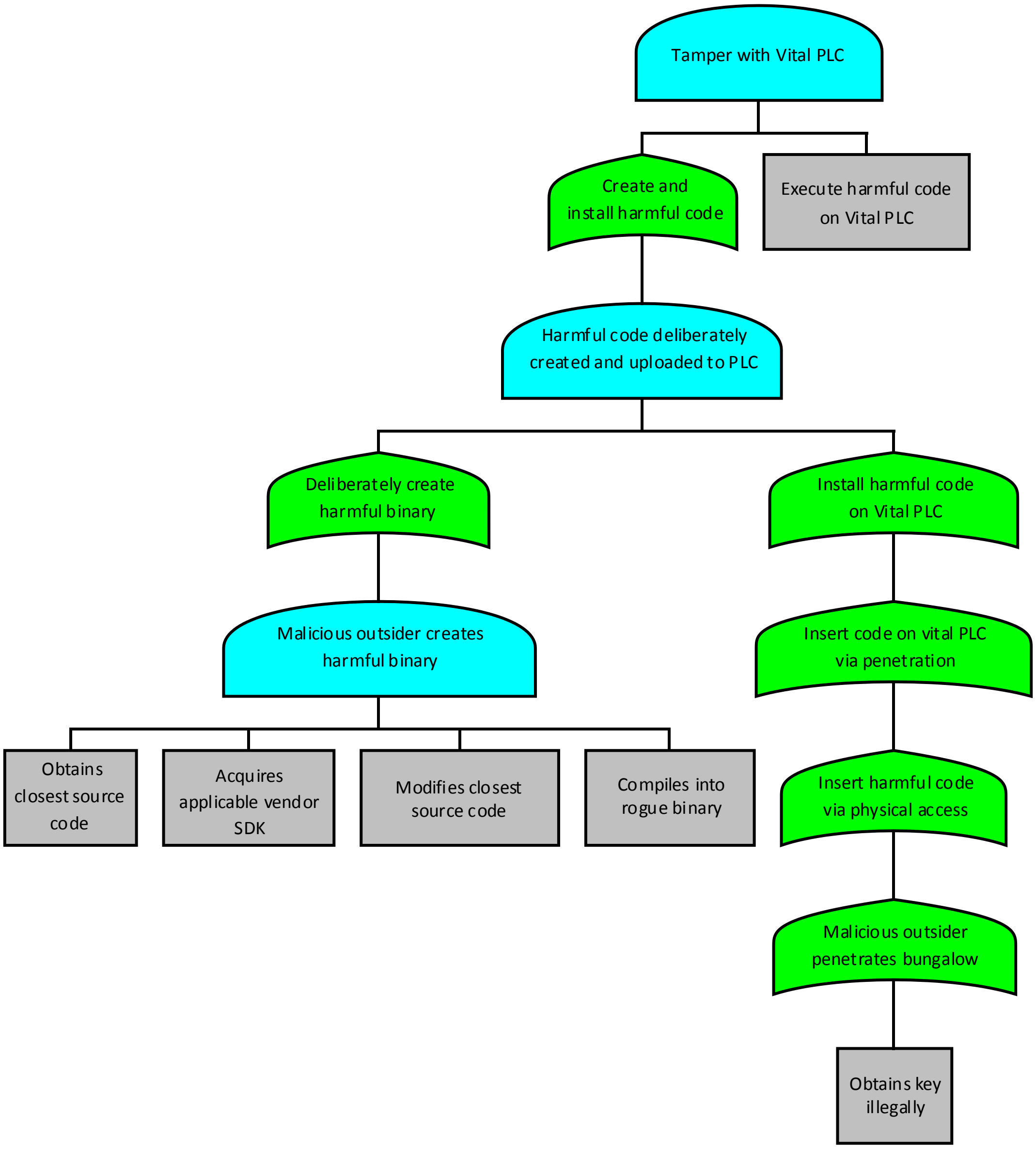
- **Scenario 8:** Collusion between malicious insider and outside partner. Starts the same as the above Scenario 5, but instead the malicious outsider physically plants the rogue binary in the signal bungalow on the vital PLC (using instructions given to him by the malicious insider, who has been in the bungalow many times). The outsider obtains the key illegally, and once again there is no direct trace back to the malicious insider, who retains his job with the transit agency after the derail. See **Figure 12**.

FIGURE 12
Attack Scenario 8


Tamper with Vital PLC
Create and install harmful code
Execute harmful code on Vital PLC
Harmful code deliberately created and uploaded to PLC
Deliberately create harmful binary
Install harmful code on Vital PLC
Harmful binary created by malicious insider
Insert code on vital PLC via penetration
Obtains closest source file
Modifies closest source file
Compiles rogue binary
Insert harmful code via physical access
Malicious outsider penetrates bungalow
Obtains key illegally

- **Scenario 16:** The malicious outsider is the only actor. (Perhaps the malicious outsider has been fired by the transit agency and wants revenge.) In this case, the outsider may have a copy of the closest source code, or can hack in to get it, and then enters the signal bungalow using a key obtained illegally and plants the malicious file on the vital PLC. See **Figure 13**.

FIGURE 13
Attack Scenario 16


Tamper with Vital PLC
Create and install harmful code
Execute harmful code on Vital PLC
Harmful code deliberately created and uploaded to PLC
Deliberately create harmful binary
Install harmful code on Vital PLC
Malicious outsider creates harmful binary
Insert code on vital PLC via penetration
Obtains closest source code
Acquires applicable vendor SDK
Modifies closest source code
Compiles into rogue binary
Insert harmful code via physical access
Malicious outsider penetrates bungalow
Obtains key illegally

6. Select and propose countermeasures (security controls)

For the attack paths judged higher risk by the analysis team (Scenarios 1, 5, 8 and 16 above), brainstorm some countermeasures (security controls) that might be applied to lessen the risk.

These are listed below by scenario:

- **Scenario 1:** Institute a cross-check of all application source and binary code by the other (full time) signal engineer. This can be "white box" — i.e., code examination or simulation runs. Adopt more stringent simulation, office and field testing by the signal engineers.
- **Scenario 5:** Do a more thorough background check of the signal contractor (malicious insider). Since this scenario involves the malicious outsider breaking into the developer's network (in the signal engineering office), strengthen perimeter defenses to the enterprise and the signal engineering office. Add cryptographic checksum to each application binary as an improved system for identifying rogue files.
- **Scenario 8:** This scenario involves two parts: The malicious insider obtaining and modifying a legitimate source file and giving it to the malicious outsider and then the malicious outsider illegally entering the signal bungalow and mounting the rogue binary on the vital PLC. It would be very difficult to prevent the malicious insider from obtaining and modifying a legitimate source file. Instead, concentrate on preventing the malicious outsider from having access to the signal bungalow and successfully mounting the rogue binary on the vital PLC without detection. This might involve a more secure key/lock system, improving entry detection alarms and using a periodic cryptographic checksum on the vital PLC.
- **Scenario 16:** Make it more difficult for the malicious outsider, operating alone, to obtain a legitimate PLC source file by strengthening defenses on signal engineering offices, requiring escorts for any visitors to those offices. Better protect the signal engineer's laptop while in transit to bungalows. Use the same security controls as for above Scenario 8 to make it more difficult for the malicious outsider to enter the signal bungalow illegally to plant a rogue file.

The analysis team would then write up the entire attack modeling case, including attack trees, attack paths, scenarios judged higher risk, etc., and present the case and recommended security controls to management.

7. Summary of attack modeling method

Attack modeling is a powerful security analysis technique that could be used in the preliminary or final phase of a new design. As described in Section 2.5, a transit project must be sufficiently large, and the security question(s) under consideration must be important enough, to warrant the expenditure of funds. A team can then be formed and trained to go through attack modeling analysis.

As described in Section 3.3, a team would include a team leader, transit agency member for oversight, team subject matter experts, and a scribe. Training would involve selecting and learning how to use attack tree software, and an understanding of the analysis method contained in this document. Use of an attack tree allows a team to form a common understanding of the threats and vulnerabilities associated with a new design. In turn, appropriate countermeasures can be put in place to provide sufficient security and operational integrity.

As is indicated in Section 3.8.6, attack modeling analysis can be implemented in a shorter period of time by using engineering judgment and the consensus of the team to apply the short method. A more extensive long method, which is detailed on the Amenaza website, adds detailed probability and risk assessment.

Once the analysis is completed, it is important to convey findings (along with recommendations for countermeasures and/or design changes) to management in a clear, understandable way. Management need not be involved in the intricacies of the attack modeling method itself, which is highly technical. The recommendations may include design change by the systems integrator or equipment vendors, or a change of equipment or operations procedures by the transit agency.

References

Amenaza Technologies Limited, APTA-related Web page. http://www.amenaza.com/APTA-WhitePaper.php

Amenaza Technologies Limited, "The SecurITree® BurgleHouse Tutorial," Terry Ingoldsby. http://www.amenaza.com/demos/case_study_burgle_house.html

American Public Transportation Association, *Recommended Practices*:

"Securing Control and Communications Systems in Transit Environments, Part I," APTA RP-CCS-1-RT-001-10, July 2010. http://www.apta.com/resources/standards/Documents/APTA-SS-CCS-RP-001-10.pdf

"Securing Control and Communications Systems in Transit Environments, Part II," APTA-SS-CCS-RP-002-13, June 2013. http://www.apta.com/resources/standards/Documents/APTA-SS-CCS-RP-002-13.pdf

U.S. Department of Homeland Security National Cyber Security Division:

"Recommended Practice: Improving Industrial Control Systems Cybersecurity with Defense-In-Depth Strategies," October 2009. https://ics-cert.us-cert.gov/sites/default/files/recommended_practices/Defense_in_Depth_Oct09.pdf

"Recommended Practice for Patch Management of Control Systems," December 2008. https://ics-cert.us-cert.gov/sites/default/files/recommended_practices/PatchManagementRecommendedPractice_Final.pdf

Definitions

attack modeling: A method of detailed security analysis of a control and communications system considering a range of threats and in what ways a system may be attacked. By studying the pathways through which an attack may be carried out, a relative ranking of the risks of system compromise from these threats may be compiled and countermeasures planned to prevent these attacks.

attack tree: A graphical representation of how a system under study may be compromised, either accidently or by those with harmful intent. It is closely related to a fault tree.

AND node: A logical gate of an attack tree in which all inputs must be present to obtain an output.

attack path: The detailed sequence of events by which an attacker would penetrate and compromise a system.

cryptographic checksum: A class of algorithm that produces one, and only one, digital characterization of a digital file. It is extremely difficult to reproduce the original file from the output.

floater: An employee or contractor who fills in a position when the need arises, such as when regular employees are on vacation.

human-machine interface (HMI): A software application that presents information to an operator or user about the state of a process.

leaf node: An element of an attack tree that represents an action.

OR node: A logical gate of an attack tree in which any input or combination of inputs will provide an output.

pruning: The elimination of very improbable attack paths in a given attack tree in order to concentrate on the more probable attack paths.

sequence diagram: A flowchart or other graphical representation illustrating a step-by-step process.

vital PLC: A programmable logic controller operating in a safety-critical environment (e.g., railway signaling).

white-box testing: A method of quality or security analysis of computer code in which the actual code is available for a detailed inspection.

Abbreviations and acronyms

APTA	American Public Transportation Association
ATS	automatic train supervision
CAD	computer-aided design
CBTC	computer/communications-based train control
CCSWG	Control and Communications Security Working Group
COTS	commercial off-the-shelf
DHS	Department of Homeland Security
FLSZ	Fire/Life-Safety Security Zone
HMI	human-machine interface
IT	information technology
LAN	local area network
NATSA	North American Transportation Services Association
NDA	nondisclosure agreement
NIST	National Institute of Standards and Technology
OCC	Operations Control Center
OCSZ	Operationally Critical Security Zone
PLC	programmable logic controller
PTC	positive train control
SCADA	supervisory control and data acquisition
SCSZ	Safety Critical Security Zone
TCP/IP	transmission control protocol/Internet protocol
TSA	Transportation Security Administration
TWC	train-to-wayside communications
WAN	wide area network