Cybersecurity Assessment of Communication-Based Train Control systems

Sergey Gordeychik, serg.gordey@gmail.com
Dmitry Kuznetsov, malotavr@gmail.com

Recently published information on the cybersecurity assessment of railway computer and communication-based control systems (CBCS) identified several weaknesses and vulnerabilities, which allow threat agents to not only degrade system reliability and bypass safety mechanisms, but to carry out attacks which directly affect the rail traffic safety [1]. Despite these findings, remarkably these systems meet all relevant IT security and functional safety requirements and have the required international, national and industrial certificates. To reduce the risks associated with cyberattacks against CBCS and their components, we recommend that system certification procedures be designed to include elements of security assessment and penetration testing.

Cybersecurity threats targeting transportation facilities differ from others in that attackers are typically unable to achieve their goals in one go or by exploiting a single vulnerability. This means that they usually carry out a series of attacks; each exploiting a different vulnerability to help expand their capabilities or create further conditions to give them the desired result. As such, analysis of cybersecurity threats in this sector should involve examining all vulnerabilities as well as attacks that can be carried out by exploiting these vulnerabilities.

[1] http://scadastrangelove.blogspot.com/2015/12/32c3-slides.html

As a result, we suggest following a specific procedure to analyze vulnerabilities and cybersecurity threats, wich was adopted by Russian Ralway as cybersecurity standart STO 02.049-2014[2]:

- build a threat model which involve transport security breaches, deterioration of economic effectiveness and functional safety;
- identify software and hardware weaknesses in CBCS components;
- assess the weaknesses found, identify related vulnerabilities and possible attacks exploiting these vulnerabilities;
- analyze possible attack scenarios and identify threats that can be realized as a result of the above.

In an ideal scenario, this checklist of vulnerabilities and threats can then be used to develop a qualitative assessment of the risks associated with possible breaches of cybersecurity, functional safety and traffic safety.

Work Planning

When planning work and selecting the appropriate methods of vulnerability analysis, one should bear in mind that it must be carried out for all possible CBCS operating modes, including:

- normal operating mode;
- all available emergency operational modes, according to CBCS design documentation;
- CBCS or individual system component maintenance modes.

[2] http://jd-doc.ru/2014/dekabr-2014/14238-rasporyazhenie-oao-rzhd-ot-30-12-2014-n-3192r

CBCS vulnerability analysis can be carried out in the form of lab or field studies. In a lab study, vulnerability analysis is performed on a test bench that reproduces CBCS operations in conditions similar to real-life operating conditions. In laboratory research, any CBCS malfunctions arising from intrusive research methods such as fault injection being applied to it will not lead to adverse consequences. This enables researchers to employ a full range of vulnerability detection methods, providing maximum coverage. That said, the combination of CBCS software and hardware components represented on the test bench or their configurations may not exactly reproduce the CBCS in actual operation. As a consequence, analysis results may not be fully applicable to the systems in operation.

Field research can be performed on a CBCS in actual operating conditions. As the CBCS being analyzed needs to operate without interruption, some vulnerability detection methods may have to be forgone, substantially reducing the comprehensiveness of the results. The main advantage of this form of research is that it makes it possible to demonstrate in practice what an attacker can do.

An optimum approach is to combine laboratory research with subsequent verification in the field. Operators can select the actual analisys methods at preparation stage, after evaluating the benefits and shortcomings of both approaches.

It is a common mistake to think that if attackers lack information about a system, this prevents them from finding its vulnerabilities. Unlike in a test lab environment, attackers have virtually unlimited time, and the type of information obtained during a preliminary survey can be gleaned by attackers through trial-and-error. Therefore, to ensure a thorough approach is taken, we recommend that security analysis be carried out based on a white-box approach, with the auditors having full access to design and operating documentation, as well as to the source code of each system. The preliminary survey stage enables the researcher to carry out a comprehensive vulnerability analysis and reduce the time required to perform it.

Below we discuss the main stages of implementing the above approach, using the computer-based interlocking (CBI) system as an example.

Threat Model

In this scenario, we use a three-level mission-centric classification[3] of possible threats affecting CBI cybersecurity, which is based on the requirements of railway technical operation rules and other fundamental documents:

1. breaches of train movement safety;
2. reduced efficiency;
3. other breaches of device functional safety and reliability.

Threats resulting in railway safety breaches are usually the most difficult to put into practice and require the greatest effort by the perpetrators. This is primarily due to the fact that the attackers need to bypass the CBI's functional safety mechanisms. If object controllers cannot be manipulated directly, for example by exploiting vulnerabilities in the radio channel, such attacks require modification of the operating logic of the main CBI modules to change the rules of switch and signal interlocking, which is a complicated task. However, if this is possible, an attacker can perform such actions as:

1.1 Setting a clear entry signal light on a route leading to an occupied track (false clear);

[3] http://www.railjournal.com/index.php/signalling/signalling-cyber-security-the-need-for-a-mission-centric-approach.html

1.2 setting a signal to a less-restrictive aspect such as a green entry signal for a section with track divergence on a switch;

1.3 operating a switch with a train passing over it;

1.4 guiding trains over split points;

1.5 setting conflicting routes.

Threats aimed at disrupting freight traffic do not usually require the attackers to be highly professional and can be put into practice using standard malware. This increases the chance of this type of threat being deployed, since it does not involve the development of dedicated tools to carry out an attack. Examples of such threats include:

2.1 putting the CBI system out of operation;

2.2 blocking control for an extended period of time;

2.3 displaying incorrect train positions on the yardmaster’s workstation;

2.4 false occupancy.

Putting non-redundant components such as the CP/CPU out of operation, will lead to the CBI system becoming non-operational, forcing a switch to manual operation, which would reduce the efficiency of freight traffic management. Spoofing or blocking network interaction between the yardmaster's workstation and CP/CPU or continually rebooting these components can result in blocking the system’s ability to send commands for an extended period of time. This would require a switch to manual operation, reducing freight traffic management efficiency. Spoofed interactions between the CP/CPU and the yardmaster’s workstation can be used to indicate false occupancy of a track circuit or display incorrect train positions on the yardmaster’s workstation, requiring additional control from the yardmaster.

The following threats reduce the system’s overall reliability.

3.1 putting CBI out of operation temporarily;

3.2 putting auxiliary equipment out of operation;

3.3 displaying false diagnostic results on the electrical mechanic's workstation.

Temporarily putting the CBI out of operation by rebooting the CP/CPU or the yardmaster's workstation reduces the mean time between failures, which is defined for software products as the time until completely restarting a program or rebooting an operating system. This can be achieved through a variety of attacks, including attacks designed to exhaust network or computational resources (Denial of Service or DoS), attacks on networking equipment designed to change configuration, TCP/IP or Ethernet parameters, or to remove/replace the firmware of networking devices.

In order to carry out attacks, perpetrators take advantage of vulnerabilities and weaknesses in CBI components. As a rule, several attacks need to be carried out to put a threat into practice.

Identifying Weaknesses and Vulnerabilities

To identify CBCS weaknesses and vulnerabilities, lab and/or field research should be carried out using a methodology based on the threat model, in order to find as many vulnerabilities and defects as possible. A variety of methods are used, such as:

- Analyzing the physical security of the facility, the CBCS and its components;
- Detecting known vulnerabilities using vulnerability scanners;
- Performing manual and automated analysis of component configurations (networking equipment, OS, DBMS) to determine whether they are in line with the vendors' recommendations and best-practice configuration standards;

- Analyzing authentication and access control mechanisms, analyzing the password policy, identifying storage of standard and fixed passwords and encryption keys, key distribution process;
- Surveying the work of operators to identify any violation of security requirements in their established practices (bypassing the limitations of the graphical interface, connecting external devices, etc.). It is recommended that this stage be carried out at the actual workplaces;
- Identifying vulnerabilities using source code analysis, fuzzing and other methods;
- Analyzing network communication, including those carried out over wireless connections, such as Wi-Fi and GSM-R;
- Analyzing system maintenance procedures and tools, including those which use remote management tools;
- Identifying security mechanisms and testing their effectiveness;
- Verifying technical security compliance.

The source code and network communication analysis stages are described in more detail below.

Source code analysis should be carried out in accordance with industrial best practice, suh as OWASP Code Review Guide[4] methodology and combines two areas of analysis:

– searching for typical programming errors;

– searching for typical errors in the implementation of specific security features.

Searching for typical programming errors involves searching code for fragments that contain programming errors which give rise to vulnerabilities. The

[4] https://www.owasp.org/images/2/2e/OWASP_Code_Review_Guide-V1_1.pdf

search is performed purposefully using criteria designed to identify certain classes of programming errors. Generally, the following errors are searched for:

– buffer overflow errors;

– errors in using language constructs (operating system commands, SQL operators, programing language operators, etc.);

– parallel computing synchronization errors (race conditions, TOCTOU[5]) wich are critical for logic of interlocking process;

– runtime error and exception handling errors;

– typical web application errors (cross-site scripting, session identification errors, etc.).

Searching for typical errors is performed using automated source code analysis tools and manual analysis. Searching for typical errors in the implementation of security functions is based on a combination of automated and expert analysis methods. This includes:

– identifying the source code fragments that implement the main security functions;

– performing static analysis of the algorithms implementing security functions;

– performing dynamic analysis of algorithm execution by emulating object code execution or debugging using test data, including the use of standard methods for exploiting vulnerabilities.

When defining the scope of work and evaluating what needs to be performed in the process of analyzing object code, keep in mind that the applicability of this method of identifying vulnerabilities has the following limitations:

[5] https://en.wikipedia.org/wiki/Time_of_check_to_time_of_use

- analysis can only be performed for a certain set of programming languages that is determined by the tools used and the existing analysis methods;
- this type of analysis is designed to identify certain classes of vulnerabilities and specific standard security function implementation errors;
- analysis is performed locally, at program function and module level and may not cover architectural vulnerabilities or security function implementation vulnerabilities and errors at the information technology level.

Experts will need to make independent decisions as to which security features are to be analyzed in the process of assessing the implementation of security features. These decisions are made based on the programming language used, the architecture of the computing devices on which the program code is executed, the availability of suitable analysis methods and tools, and other factors. Typically, the implementation of the following security features is assessed:

- identification and authentication;
- access control;
- session management;
- network communication and critical system components integrity control;
- input data validation;
- handling runtime errors and exceptions
- generation and storage of audit reports;
- cryptographic functions and management of encryption keys.

Analysis of network communication is used as an auxiliary method of passive vulnerability analysis in the process of solving the following problems:

- creating an inventory of CBCS components using passive methods, including the nomenclature and versions of operating systems and other software components;
- identifying information flows for the purposes of analyzing CBCS topology;

– detecting prohibited types of interaction, as well as information flows which may be indicative of the presence of malware;

– detecting events in which confidential or sensitive information, including login credentials, is communicated insecurely.

In addition, research should include the analysis of communications hardware such as network modems, (U)SIM cards, GSM-R and SDR radio stations. Such devices are sophisticated computer systems in which vulnerabilities have been previously detected. Such vulnerabilities can be exploited in attacks against the entire CBCS infrastructure[6].

Attack scenarios

These processes will result in a list of CBCS weaknesses, some of which could be potential vulnerabilities. To confirm, detected weaknesses need to be assessed and identified as vulnerabilities. This can be done in one of several ways:

- practical demonstration of how the vulnerability can be used to pose a real threat to cyber security;
- description of the theoretical possibility of the vulnerability being used to pose a threat to cyber security that raises no objections from CBCS specialists;
- for known vulnerabilities in the code – presence of the vulnerability in the database of one or more resources used for identifying vulnerabilities or a security bulletin from the software developer confirming the release of a security update that eliminates the vulnerability and availability of exploits;
- for unknown or unpublished vulnerabilities in the code – a message in the software developer notes confirming that a defect is a vulnerability;

[6] http://securityaffairs.co/wordpress/31663/hacking/hacking-4g-usb-modems.html

- in the event of obsolete and unsupported software being used – a press release or other statements confirming the termination of software support;
- configuration errors – a publication by the software developer, or other authoritative sources, recognizing the negative impact of this configuration on the overall security of CBCS or on an individual component of the system.

The severity of confirmed vulnerabilities can then be assessed and recommendations formulated to address them.

If a previously unknown vulnerability is revealed in the course of the work, the testing laboratory informs the CBCS developer, notifying them about the vulnerabilities in conventional form, in line with the policy of 'responsible disclosure'[7].

Threat Analisys

The collected data is then used to analyze cyber security threats. This involves the construction of a sequence of attacks (attack graphs) that meet the following conditions:

- for each attack on the CBCS, the vulnerability that enables that attack is identified;
- by virtue of the initial conditions and/or as a result of previous attacks, at the time a specific attack is carried out, the intruder has acquired the capabilities required to perform the attack;
- carrying out the final attack results in an objective being fulfilled.

To create a directed graph of attack, the initial vertex needs to be the ultimate goal of the attacker. At the first stage of analysis, the vulnerable CBCS components

[7] https://en.wikipedia.org/wiki/Responsible_disclosure

are determined. An attack on these components then leads to one or more of these objectives being attained. At this point, the capabilities needed by an attacker to carry out the identified attacks can be determined. Vulnerable CBCS components are then defined for every capability the attacker possesses. These are the components that, if successfully attacked, will give the intruder the required access level. The process is then repeated until such time that the CBCS vulnerabilities are exhausted or until the set of CBCS component vulnerabilities required to carry out all of the analyzed attacks are determined, taking into account the subsequent acquisition by the intruder of new capabilities as a result of each attack.

Conclusions

The cyberassesment and threat modelling approach outlined in this paper can help to identify the most likely attack vectors, security mechanisms that counteract them and the weaknesses in the system's cybersecurity. This data can then be used to develop a qualitative risk analysis associated with possible breaches of cybersecurity, functional and traffic safety.