Transportation Software Assurance

SOFTWARE ASSURANCE GUIDELINES DOCUMENT

Joshua Powell | CSS321 |Software Assurance | 08Aug2022

Table of Contents

Project outline and requirements

BRIEF DESCRIPTION OF THE ORGANIZATION

This organization is one of the many driving forces to keep our nation moving and staying on top of the threats that face our nation and country. In light of recent events with the increase in security threats and vulnerabilities, it has become more prevalent the need for security analysis of the software in our transportation vehicles.

COMPANY SIZE, LOCATION(S), AND OTHER PERTINENT INFORMATION

This company has vehicles all around the country and world that are used to move and transport products of all types. Along with sharing the road with other motorists.

LIST OF THE SOFTWARE APPLICATIONS PROVIDED BY THE COMPANY FOR THE GOVERNMENT

- They provided freightliner branded single and tandem axle trucks with there own ecus and with the military's version of diagnostics back in early to mid-1980's they have not made any recently.

A SUMMARY OF THE SOFTWARE DEVELOPMENT ORGANIZATION WITHIN THE COMPANY, EMPLOYEES AND REPORTING STRUCTURE, SYSTEMS AND TECHNOLOGIES USED FOR SOFTWARE DEVELOPMENT, TESTING, SOURCE CONTROL, AND DOCUMENT STORAGE

- For several years now, Daimler trucks north America was not formally a tech company, but a company that made tractors that moved semi-trailers across America. They lacked a good software development cycle. So, they went with IBM UrbanCode Deploy Software, where they could deploy a DevOps approach to keep up with technology demands in their vehicles (1).
- So with IBM UrbanCode Deploy Daimler basically teamed up so they can provide better functionality to their customers (2).
- Their process for their trucks they have certain Ecus within the truck from the cab ECU, engine ECU, transmission ECU, chassis ECU, and the central gateway. These all play a role and communicate on an SAE J1939 which is a standard worldwide for communication in vehicles.
- They have a new line of vehicles that may be in the making, they spend hours upon hours going back and forth creating the firmware

needed for these trucks to stay up and running. So they have to go through a sort of DevOps approach of there own making sure that they make a quality product and make sure it is secure.

Security in the development life cycle

- In the security development life cycle, there are 12 practices to a successful security development within the lifecycle.
- Provide Training: This is where you want to make sure everyone understands security best practices.
- Define Security Requirements: Here you want to make sure you update and inform changes in the threat landscape to make sure that threats are updated regularly
- Define Metrics and compliance reporting: This is where you define the minimal requirements for the engineering teams to hold them accountable.
- Perform Threat Modeling: With threat modeling this is where you want to make sure you find threats and vulnerabilities to make mitigations to prevent them.
- Establish design requirements: Define the security features that the engineers should use.
- Define and use cryptography standards: This is where defining the security requirements is necessary to make sure you use the appropriate cryptography.
- Manage the Security Risk of using Third-Party Components: This is where you want to make sure you keep tabs on third party vulnerabilities to make sure that you evaluate them appropriately.
- Use Approved Tools: Make sure that you publish a defined list of the tools you approve and their security checks.
- Perform static analysis security testing(sast): Here you want to make sure that you analyze the source code so you can secure the coding policies.
- Perform dynamic analysis security testing (dast): this is where you test the fully compiled software by doing a run-time verification.
- Perform Penetration Testing: In this practice you want to make sure that you find any vulnerabilities in the coding so you can fix any issues that may exist.
- Establish a standard incident response process: This is where you want to make sure that you establish a response plan to counteract any issues that may arise (3).

Software Assurance Techniques

- ANALYSIS:
  - Software applications that are produced by the organization:

    1. The first one is diagnostic link which is what they use to read the software on there ecus inside the truck along with access to Daimler trucks database.
    2. Fleet monitoring through wireless connectivity and database access (4)
    3. Tire pressure and temperature monitoring application

  - Identify at least 2 areas of each application that are at security risk and describe the possible threats and their implications to the organization and to the client (in this case, the government).
    1. The tire pressure system operates wirelessly that means there is a vulnerability for anyone trying to access the unit that can penetrate the system through this application (5).
    2. Fleet monitoring because this gives a wireless vulnerability within a truck as it puts it into the IoT of things. Connecting it to the internet along with access to its database.
  - For each security risk, identify at least 1 software assurance technique that can be applied to reduce the security threat.
    1. The best way to secure a Tire pressure monitoring system is by incorporating a lightweight randomization scheme (6) and making sure you update and patch accordingly.
    2. With fleet monitoring and data logging one software assurance is implementing “hardening” within the database it is accessing to store the data logs (7).
- GUIDELINES:
  - Based on the analysis that was performed in the previous step, prepare a set of software assurance guidelines that the organization can use for all the applications that it creates.
    1. Make sure that the hardware connecting to the trucks themselves is free of malware that could infect the truck and have their own malware monitoring
    2. For the Wireless Fleet Monitoring you need to make sure to implement hardening within the database along with following the fleet monitoring software and hardware for any security risks that may exist.
    3. The tire pressure monitoring system needs to be taken seriously as hackers can access the ID used for TPMS sensors and hack into the network of the system in the truck so

implementing a randomization id generator will make it secure from hackers and deter them away.

Security in Nontraditional Development Models

NON-TRADITIONAL PROGRAMMING

- A non-traditional software development model that could be used for the better of my company is the Extreme Programming model.
- Extreme programming summary

    a. One step is code review which allows for an efficient review of the code to detect and correct issues.
    b. Another step of extreme programming is the testing phase where the reliability of the code is increased
    c. A third process is incremental development where after every iteration the code is tested and processed, and customer feedback is given to improve the code
    d. Fourth process is the integrated testing where the code is tested daily to improve the quality and to catch any bugs that may have slipped through (8).

- The company needs to make sure that each process has a person to see over each process and a Project manager checking in to make sure that each person is doing their job appropriately. The company needs to make sure that they implement security protocols for the extreme programming model. The area between the client and vendors is to make sure that security protocols are implemented to guarantee the security of data transmissions between them.

Security Static Analysis

- Complete the Security Static Analysis section:
    - Prepare a design for an application your organization might produce.

      One such design they might prepare would be one that receives and input and delivers an output such as the current fuel level. A driver must be aware of the current fuel level so that they can properly route their course and

know when to stop. One such example could be a receive integer and output function

- #inlude <stdio.h>
  #include <string.h>

  int main() {
  char str1[20] = “C programming”;
  char str2[20];

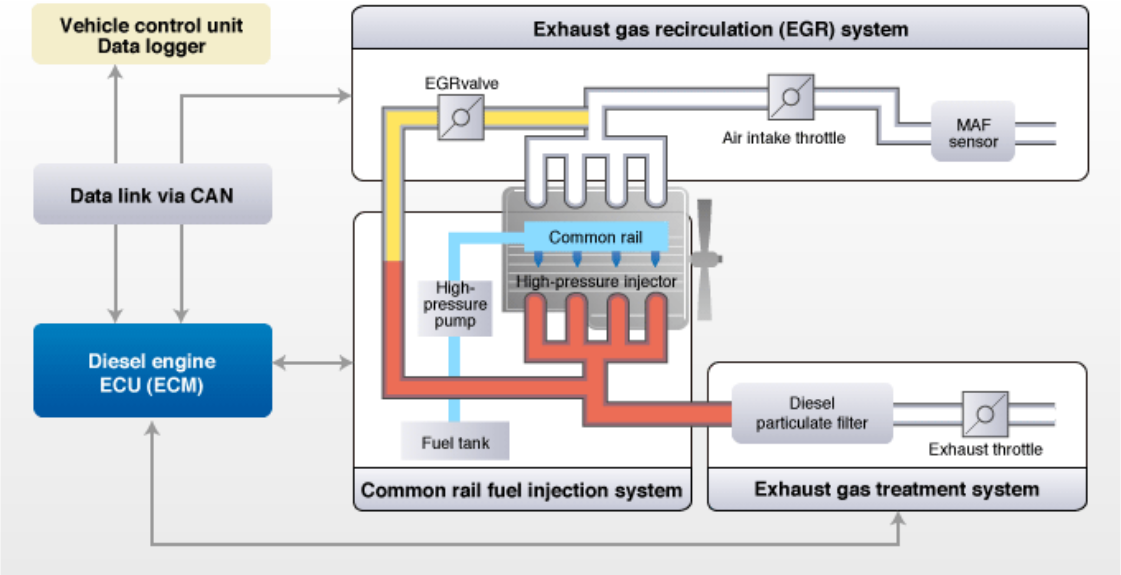
  // copying str1 to str2
  Strcpy(str2, str1);

  puts(str2); // C programming

  return 0;
  }
  (9)

- Include appropriate diagrams to identify the major components of the application.

  (10)



- Describe the major components and potential security issues where appropriate and as related to the security development model.

  One of the main issues with this design is that without a central gateway which is the firewall for the truck when someone plugs up to the diagnostic

port. The unit will stand the chance of having malware injected into the software of the units ECU's. As you can see from the image you have a vcu data logger along with the data link via can which allows for control of network traffic from the diagnostic can port.

- Create code samples in C, C++, or Java to illustrate the tenets of the security development model.
  - One such way to increase the security of the unit using C language is to use a function that limits the input sting unlike its predesesor and that is sprintf() as seen below.
  - Sprintf(buffer, "usage: %.124s argument\n", argv[0]);

- Identify at least 3 security static-analysis tools, and prepare guidelines for how they would be used in the sample code and throughout the software development in the company
  - **Klocwork:** This static-analysis tool is one of the industry leaders as it considers both false positives and false negatives. There is also the feature that Klocwork uses differential analysis so you can get faster results. This would mean that with the sample above not only would It be shorter analysis time, but it would also pick up on the fact that sprintf would be the better option than strcpy as it puts a limit on the input string.
  - **PARASOFT:** This tool is particularly geared towards this side of things as it tests embedded systems and keeps things in check when putting ecu's together to communicate over the j1939 network.

  - **PVS STUDIO:** When it comes to a commercial tool it cannot be complete without having a tool that can reach into software and find those embedded copy and paste mistakes. A typo or two that may exist and so on. (11)

Software Assurance Policies and Processes

- Prepare a plan for the training of the software developers in the organization on the new software assurance guidelines.
  1. First is the entry criteria: Management support, Training policy and objectives, Resources, and then organizational context.
  2. Then you need the Tasks that are needed to be done : first is to conduct a training analysis, then create a training plan, design the curriculum, create training products, Pilot and deliver the training, and Evaluate the training

3. Next is validations: Training plan approved, course development and delivery processes followed, quality standards met, training results analyzed and reported
4. Lastly is exit criteria: Needed Training delivered and Training Objectives Met(12)

- Define the metrics that will be collected to track the effectiveness of software assurance in the company.
    - Include a description of how each of the metrics will be obtained and used.
        1. **Defect Density:** this metric is to be obtained after each phase is completed by gathering all the defects within the software that need to be corrected. Defect density is Defect Density = Defect Count/Size of the Release/Module
        2. **Defect Category:** This Metric offers insight to the qualities of the software: functionality, stability, robustness, etc. This metric allows for the defect to be more centralized and allows for a better understanding of where a defect may exist.
        3. **Test Case Productivity:** This is an ideal metric to always keep in play as it makes sure that the team running the test metrics are investing the appropriate time and assets to conduct good quality testing. (13)
- Identify the roles and responsibilities of the members of the security team with respect to software assurance in the organization.

    1. The software Assurance security team should take part in the early phase of the software development life cycle. This is because they are the ones who make sure that the appropriate policies are put into place along with guidelines. One can say that the software security team are the backbone to the software lifecycle. The reason being is they play a vital role in making sure that the software is robust and sound. That it is secure, but also meets local and federal regulations. That there are no hidden bugs that could potentially harm the customers or company in a non-ethical matter.

References

1. *Daimler Trucks North America*. (2019). IBM. https://www.ibm.com/case-studies/daimler-trucks-north-america
2. I.B.M. (2019). *UrbanCode Deploy - Overview*. IBM. Retrieved 2022, from https://www.ibm.com/cloud/urbancode/deploy
3. Microsoft. (2020). *Microsoft Security Development Lifecycle Practices*. https://www.microsoft.com/en-us/securityengineering/sdl/practices
4. *Daimler Truck North America*. (2020). Daimler. https://northamerica.daimlertruck.com/
5. Daimler. (2012, September 24). *Daimler Truck North America*. Daimler. https://northamerica.daimlertruck.com/company/press-releases/pressdetail/daimler-trucks-north-america-adds-tire-2012-09-24

6. *Securing Tire Pressure Monitoring System for Vehicular Privacy*. (2021, January 9). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/9369576
7. Murray, R., Nakar, O., Nakar, O., Lynch, B., Naim, E., Lynch, B., Lynch, B., & Lynch, B. (2021, November 8). *What is Database Security | Threats & Best Practices | Imperva*. Learning Center. https://www.imperva.com/learn/data-security/database-security/
8. GeeksforGeeks. (2022, June 13). *Software Engineering | Extreme Programming (XP)*. https://www.geeksforgeeks.org/software-engineering-extreme-programming-xp/?ref=gcse
9. *C strcpy() - C Standard Library*. (n.d.). Retrieved August 7, 2022, from https://www.programiz.com/c-programming/library-function/string.h/strcpy#:%7E:text=Example%3A%20C%20strcpy%28%29%20%23include%20%3Cstdio.h%3E%20%23include%20%3Cstring.h%3E%20int,C%20programming%20return%200%3B%20%7D%20Output.%20C%20programming
10. *Electronic Control Units: Diesel Engine ECU - Transtron*. (n.d.). Copyright (C) Transtron Inc. Retrieved September 7, 2022, from https://www.transtron.com/en/products/control/dieselengine.html
11. Exterman, D. (2022, July 31). *Top 9 C++ Static Code Analysis Tools*. Incredibuild. Retrieved September 7, 2022, from https://www.incredibuild.com/blog/top-9-c-static-code-analysis-tools

12. *Creating a Custom Training Plan for Your Organization*. (n.d.). SimplifyTraining. Retrieved September 12, 2022, from https://simplifytraining.com/article/creating-a-custom-training-plan/
13. “Top 34 Software Testing Metrics & Kpis.” ThinkSys Inc, 5 May 2022, https://www.thinksys.com/qa-testing/software-testing-metrics-kpis/.