The University of Melbourne

SWEN90010: High Integrity Software Engineering

Assignment 2

Due Date: 11:59pm, Tuesday 30 April, 2019

Introduction

This handout is the Assignment 2 sheet. The assignment is worth 20% of your total mark. You will carry out the assignment in the same pairs as for Assignment 1.

This is a two-part assignment, each part of which is worth 10% of your total mark (i.e. the two parts are equally weighted).

The first part has you apply some safety engineering knowledge and principles in the context of the recent Boeing 737 MAX 8 incidents. In the second part you will use the Alloy formal modelling and analysis tool to analyse the security of a general network logging scenario, inspired by modern aeroplane and automobile design.

Part I: Safety Analysis — Boeing 737 MAX 8

Boeing's 737 MAX 8 aircraft has suffered two fatal crashes in the past 5 months [8]. Both incidents are still under investigation, and so definitive information on their precise causes is not yet available. However, there has been substantial reporting about both incidents [6, 3, 4, 2]. At this stage, it appears as if a software system called the Manoeuvring Characteristics Augmentation System (MCAS) [5, 9] played a major role in one if not both incidents. The MCAS is a new software system added to the MAX aircraft, designed to automatically adjust the pitch of the aircraft to prevent the engines from stalling in certain situations.

Boeing has since announced how it will fix the MCAS issues, including by updating the flight control software [2].

In the first part of this assignment, you will study the publicly available information regarding the 737 MAX 8 incidents to understand how MCAS is believed to have contributed to them. You will then perform a Fault Tree Analysis to understand how the announced fixes to the MCAS address the suspected causes of the incidents and to determine for yourself (or, more precisely, your pair) how effective those fixes are.

Note: there has been substantial reporting around the sufficiency of the certification process for the 737 MAX, including with regards to the MCAS system [3]. The issues raised in that reporting are important but are not directly relevant to this assignment.

Your Tasks

1. Prepare a short (up to one page) description of the MCAS's role in the first of the two 737 MAX 8 incidents.

   The first incident involved Lion Air Flight 610, and occurred on October 29 2018. At present, it remains better understood than the second, more recent, incident, and so is a

better target for analysis.

Note: as to be expected, there is some contradiction and confusion between various reporting around the October 29 2018 incident. Therefore, in your one-page description, please cite the sources you are relying on for you understanding of the incident and its causes. References do not count towards the one-page limit and should appear at the end of your assignment submission.

2. Based on your description, draw a Fault Tree that outlines the causes and their interactions of the October 29 2018 incident.

   You should include up to half a page of text explaining the choices you made when constructing your fault tree.

   Hint: it is believed that a contributing factor was that the MCAS activated multiple times (i.e. repeatedly). You will have to think carefully about the best way to represent this repetition and its consequences in your fault tree. The accompanying half-page explanatory text provides you with the opportunity to explain how you chose to represent this repetition. You are free to depart from the traditional tree structure (i.e. for your diagram to not have a tree structure) if you wish.

3. Repeat the fault tree exercise but now focusing on the system *after* Boeing's fix has been applied. That is, draw a fault tree analysing this same hazard but assuming that the fix has been applied, based on your understanding of the fix.

4. In one page of text or less, explain how you believe Boeing's fix addresses the cause of the October 29 2018 incident. As above, cite your sources. Your explanation should also state and justify whether you believe the fix is sufficient, e.g. are there additional factors identified in your fault tree analysis that the fix does not (satisfactorily) address?

Part II: Formal Security Analysis with Alloy

The second part of the assignment requires you to carry out a formal security analysis of a generic network logging scenario. The scenario is depicted in Figure 1. Here there is a *Sender*, which sends *log messages* onto the network. Those messages are received by a *Logging Service*, which records the messages it receives into a *log*.

Figure 1: Network Logging Scenario

This scenario is extremely common in high integrity systems. For example, the Boeing 737 MAX (and prior 737) series aircraft include a network logging service, exposed via the Network File System (NFS) protocol, which receives log messages from the Electronic Engine Control (EEC) system about the performance of the engines [7]. That information is used during subsequent

maintenance (e.g. to carry out Engine Trim Balancing). Thus the integrity of the log data is critical for correct engine maintenance.

At the same time, recent aircraft design has tended towards interconnecting the various aircraft networks, including passenger networks (e.g. in-flight wireless or in-flight entertainment) and those for aircraft systems (e.g. to allow access to shared services like satellite communications) [10]. By compromising (i.e. exploiting security vulnerabilities) in the in-flight entertainment system, it has been alleged that attackers can then interfere with aircraft systems (e.g. the Thrust Management Computer) [11]. Similar scenarios have also been demonstrated in the context of modern cars, in which by hacking the entertainment system one can send commands to interfere with the car's critical systems [1].

Thus your task for this part of the assignment is to consider a potential attacker who has gained access to the network, in between the Sender and Logging Service, and to use Alloy to understand how the attacker can interfere with the log's integrity.

You are provided with a partial Alloy model `logger.als` of the logging scenario, which you will extend. To keep the assignment tractable, the model of the network is simplified by assuming that the network can hold at most one message at a time.

The Alloy model includes predicates that model the sending and receipt of log messages by the Sender and Logging Service respectively. It also contains placeholder predicates for various *attacker actions*, which you will implement, that model actions that the attacker might perform on the network.

Traces of system actions are captured using the `util/ordering` module described in lectures. *Hint: you may find it helpful to make use of the functions defined in that module when writing assertions and predicates for this assignment. A copy of the Alloy source for the module is available on the LMS with the assignment.*

The log itself is modelled as an Alloy *sequence*, of type `seq LogMessage`. The Alloy file includes an example predicate `log_only_grows` to give some ideas about how to specify properties of the sequence type. *Hint: you should read up on the sequence type in Alloy's online documentation, including here:* `http://alloy.lcs.mit.edu/alloy/documentation/quickguide/seq.html`.

Your Tasks

1. The attacker action predicates are:

    - `attacker_action_drop`: this models the action in which the attacker intercepts a log message and prevents it from reaching the Logging Service, by removing it from the network.
    - `attacker_action_fabricate`: this models the action in which the attacker invents a *new* log message and injects it into the network, to be received by the Logging Service. This action can only be performed when the network does not already contain a message.
    - `attacker_action_replay`: this models the action in which the attacker injects an old message onto the network, i.e. a message that was already present on the network in some prior state of the model. This action can only be performed when the network does not already contain a message.

Fill in the implementations of these three predicates to model each of the actions described above. Add comments as you feel are appropriate to explain your model.

2. The model contains a placeholder predicate `log_correct`, which takes a single system state s as its argument. The intent of this predicate is that it should assert the correctness of the log with respect to the execution trace leading to state s (i.e. with respect to the previous states before s). In particular: **if `log_correct` holds for *all* states s, then it should be the case that the log only ever contains messages that were sent by the Sender and that the messages in the log should not be out of order.**

   Implement this predicate. As above, add comments as appropriate to explain your model.

   Hint: there are multiple ways to implement it. The main thing to keep in mind is the bold phrase above.

   Hint: note that the description of correct logging above does not rule out the possibility that log messages sent by the sender might be missing *from the log. This is intended and would still be judged as correct logging behaviour. In particular, the network between the Sender and Logging Service (like all networks) is assumed to be unreliable sometimes, causing messages to get lost. Therefore your `log_correct` predicate should not rule out this behaviour.*

   Hint: there might also be additional behaviours that you might consider to be undesirable that would nonetheless be permitted by the above description of correct logging behaviour. If you identify such behaviours, you should include those in your answer to question 4, on additional attacks, below.

3. The Alloy file contains a number of assertions that check whether the `log_correct` predicate holds under different *attacker models*, i.e. under different sets of assumptions about which actions the attacker might perform.

   For example, `log_correct_when_attacker_only_drops` asserts the correctness of the log when the only action the attacker performs is to drop messages. This assertion should hold, since correct logging does not preclude messages being lost by the network (see the hint above).

   Each of the other assertions, however, assert correct logging when the attacker is more powerful. For each of these you should use Alloy to discover an attack that the attacker can perform that violates correct logging.

   Add comments to the Alloy file describing each of the attacks. You should find one attack for each of the `log_correct_*` assertions (except `log_correct_when_attacker_only_drops` of course).

4. Are there further attacks that an attacker could carry out that are not captured, whether by your `log_correct` predicate or by the attacker actions?

   Add comments to your Alloy file to describe any additional attacks that you believe would be possible but that the model does not capture. For each, describe how the model would need to be improved to capture them.

Criteria

Part I

Criterion	Description	Marks
Incident Analysis	The description and understanding of the incident is thorough, and accurate with respect to the sources cited	3 marks
Incident Fault Tree	All major causes and their interactions are accurately captured and depicted.	3 marks
Fix Fault Tree	The updated Fault Tree correctly captures the effects of the fix, in terms of its effects on the causes of the incident	2 marks
Fix Analysis	The analysis of the fix, including its adequacy, is thorough, and accurate with respect to the sources cited.	2 marks
Total		10 marks

Part II

Criterion	Description	Marks
Attacker Actions	The attacker actions are correctly modelled and adequately described by their comments.	2 marks
Log Correctness Spec	The specification is correct, clear, succinct, and adequately explained by comments.	4 marks
Attack Scenarios	The attacks discovered are accurately explained and correctly reflect the attacker's ability in each case.	2 marks
Additional Attacks	Any additional attacks are identified and adequately explained, including how the model would need to be changed to capture them.	2 marks
Total		10 marks

Academic Misconduct

The University misconduct policy applies to this assignment. Students are encouraged to discuss the assignment topic, but all submitted work must represent the pair's understanding of the topic.

The subject staff take plagiarism very seriously. In the past, we have successfully prosecuted several students that have breached the university policy. Often this results in receiving 0 marks for the assessment, and in some cases, has resulted in failure of the subject.

Submission

Submit each part of the assignment using the Turnitin links on the subject LMS. Go to the SWEN90010 LMS page, select *Assignments* from the subject menu, and then select *View/Complete* from the appropriate *Assignment 2 Part X submission* item, for each part. For Part I, upload a **PDF** file containing your solution. For Part II, upload the Alloy file containing your solution.

Only *one* student from the pair should submit the solutions, and each submission should clearly identify **both** authors.

Late submissions Late submissions will attract a penalty of 10% (1 mark for each part that is late) for every day that they are late. If you have a reason that you require an extension, email Toby *well before the due date* to discuss this.

Please note that having assignments due around the same date for other subjects is not sufficient grounds to grant an extension. It is the responsibility of individual students to ensure that, if they have a cluster of assignments due at the same time, they start some of them early to avoid a bottleneck around the due date.

References

[1] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno. Comprehensive experimental analyses of automotive attack surfaces. In *USENIX Security Symposium*, 2011.

[2] Dominic Gates. Boeing details its fix for the 737 max, but defends the original design. *The Seattle Times*, March 2019. `https://www.seattletimes.com/business/boeing-aerospace/boeing-details-its-fix-for-the-737-max-but-defends-the-original-design/`.

[3] Dominic Gates. Flawed analysis, failed oversight: How boeing, faa certified the suspect 737 max flight control system. *The Seattle Times*, March 2019. `https://www.seattletimes.com/business/boeing-aerospace/failed-certification-faa-missed-safety-issues-in-the-737-max-system-implicated-in-the-lio`

[4] Dominic Gates. Lack of redundancies on boeing 737 max system baffles some involved in developing the jet. *The Seattle Times*, March 2019. `https://www.seattletimes.com/business/boeing-aerospace/a-lack-of-redundancies-on-737-max-system-has-baffled-even-those-who-worked-on-the-jet/`.

[5] Jon Ostrower. What is the boeing 737 max maneuvering characteristics augmentation system? *The Air Current*, November 2018. `https://theaircurrent.com/aviation-safety/what-is-the-boeing-737-max-maneuvering-characteristics-augmentation-system-mcas-jt610/`.

[6] Nicolas Rivero. Everything we know about the boeing 737 max 8 crisis. *Quartz*, March 2019. `https://qz.com/1578227/everything-we-know-about-the-boeing-737-max-8-crashes/`.

[7] FEAM Technical Training. B737 max general familiarization engine course: 737 7/8/9 training manual. http://feamtechnicaltraining.com/files/B737%20MAX%20GEN%20FAM%20POWERPLANT%20BOOK.pdf.

[8] Wikipedia. Boeing 737 max: Accidents and incidents. https://en.wikipedia.org/wiki/Boeing_737_MAX#Accidents_and_incidents.

[9] Wikipedia. Boeing 737 max: Maneuvering characteristics augmentation system (mcas). https://en.wikipedia.org/wiki/Boeing_737_MAX#Maneuvering_Characteristics_Augmentation_System_(MCAS).

[10] Rick Wilber. A system of systems approach to e-enabling the commercial airline applications from an airframers perspective. http://home.iitk.ac.in/~lbehera/indous2/Talks_files/Day%202/Rick%20Wilber.pdf.

[11] Kim Zetter. Is it possible for passengers to hack commercial aircraft? *Wired*, May 2015. https://www.wired.com/2015/05/possible-passengers-hack-commercial-aircraft/.