The University of Melbourne
SWEN90010: High Integrity Software Engineering

Assignment 3

Due Date: 11:59pm, Sunday 14 May, 2017

1 Introduction

The assignment is worth 20% of your total mark and is done in pairs (the same pairs as assignment 2).

The aim of this assignment is to specify a formal model in Alloy of (part of) the EBS system, and to carry out some formal security analysis. The assignment evaluates your ability to apply formal specification techniques to help engineer a security- and safety-critical system.

2 Your tasks

The file `ebs.als` (available on the LMS) is an Alloy model of (part of) the EBS system from Assignments 1 and 2, focussing on the system's *external* interface from the point of view of the CAN Bus, and describing aspects of its behaviour in response to the receipt of CAN Bus messages.

The CAN Bus, which interconnects the various electronic control units (ECUs) on modern cars, has been a common vector by which various "car hacking" attacks and demonstrations have been carried out in recent years. Since the CAN Bus connects various ECUs (e.g. the brake controller ECU and the ECU that runs the entertainment system), researchers have demonstrated that by compromising the car's entertainment system (e.g. by playing a malicious CD) they can send attacker-crafted CAN Bus messages that, in turn, could take control of the car's critical systems [1].

The provided Alloy model focuses on the interactions between the brake pedal and the brake controller, via the CAN Bus, as well as those between the engine and the brake controller, again via the CAN Bus. To keep the assignment tractable, it does *not* model the anti-lock braking functionality of the EBS, and it models only a small part of the system's external interface via the CAN Bus.

It has declarations for predicates (some of whose bodies you will have to fill in as part of the assignment) to model a subset of the system's behaviour, such as sending and receiving the *EngineOn* message, sending and receiving *BrakePressureUpdate* messages, and the car's driver changing the amount of foot pressure applied to the brake pedal. It also includes a predicate to model the actions of a potential *attacker*, which we assume has access to the CAN Bus. Part of the assignment involves using Alloy to help reason about the EBS system in the presence of such an intruder which, if recent history is any guide, is not an unreasonable assumption.

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.*

Your tasks are:

1. **Modelling Actions in Alloy (6 marks).** Read the comments in the provided Alloy file and, in the *Actions* section of the file, fill in both the missing comments describing the pre- and postconditions of each action as well as the missing action predicate bodies.

2. **Specifying and Checking Simple Properties (3 marks).** Besides an initial example assertion, there are two simple assertions in the *Properties* section of the provided file (and a third one that we come to later). The first one, `recv_brake_pressure_update_safe`, states that some property `recv_brake_pressure_update_safety` always holds whenever the `recv_brake_pressure_update` action occurs. By referring to the Assignment 1 requirements, choose a suitable condition and fill in the body of the `recv_brake_pressure_update_safety` predicate accordingly. Add a comment to the file to justify your choice.

   The second assertion states that the predicate `inv`, described in the comment above its declaration, always holds in all states. `inv`'s body is missing. Fill in its body by referring to the comment.

   Use Alloy to check whether these two assertions hold, increasing the scope of the checks as necessary to increase your confidence in the results. Add comments below each of the checks describing whether each is true or not and why.

3. **Updating the Attacker Model (4 marks).** The model of the attacker, captured by the `attacker_action` predicate, is very powerful. It models an attacker that has the ability not only to alter the contents of messages on the CAN Bus, but also to inject new messages onto the bus, amongst other things.

   A straightforward way to help guard against this kind of attacker would be to design the system such that all legitimate CAN Bus messages carry a *Message Authentication Code (MAC)* generated with a secret key that is infeasible for the attacker to obtain. All legitimate components know the secret key, which they use to compute MACs for their messages and to verify the authenticity of the message/MAC pairs they receive. Under this design, the attacker is unable to forge new messages on the CAN Bus[1].

   Update the `attacker_action` predicate to faithfully model the attacker's reduced abilities under this new design, remembering to update the comments accordingly. Any checks that might have failed before will most likely now hold; however, if any don't, add a comment to explain why.

4. **Specifying and Checking Non-Trivial Properties (5 marks).** For this part of the assignment, think carefully about how *BrakePressureUpdate* messages are sent from the brake pedal to the brake controller.

   The assertion `brake_at_correct_pressure` in the *Properties* section is missing its body. This assertion is intended to describe a fairly precise specification of the correct value of the brake pressure (i.e. the value of the `brake_pressure` field of the state), given the prior receipt of *BrakePressureUpdate* messages on the CAN Bus.

   Your task is to provide a precise enough body for this assertion to allow Alloy to detect attacks that can arise under the updated attacker model as counterexamples to your

[1] Although they can still capture legitimate MAC'd messages and then later *replay* them, by injecting them onto the bus.

assertion. Add comments to explain your finished assertion.[2]

Add comments to the file to describe the kinds of attacks that are still possible under the updated attacker model.

Note: when thinking about the intended value of the brake pressure, given the prior receipt of BrakePressureUpdate *CAN Bus messages, you should naturally refer to the Assignment 1 requirements. However, you should ignore aspects of those requirements that deal with anti-lock braking functionality, which we ignore for the purposes of this assignment.*

Note: to get full marks here, you will have to write a logical assertion that is precise enough for Alloy to find counter-examples to it, under the updated attacker model, and then describe the associated attacks represented by these counter-examples in the comments, as well as any other kinds of attack that are possible under the updated attacker model that would violate the intended behaviour.

5. **Relationship to HAZOP Study (2 marks).** Consider *your* HAZOP study in Assignment 2. Add comments describing:

   (a) Which of these attacks are covered by hazards that you identified.

   (b) Any new hazards suggested by these attacks, including the design item that each applies to and an appropriate HAZOP guideword for each.

3 Criteria

Criterion	**Description**	**Marks**
Model	Action predicates bodies and comments are correct and complete. Updated attacker model and comments are correct and complete. The solution is clear and succinct.	10 marks
Assertions	Appropriate choice for `recv_brake_pressure_update_safety`. Precise enough definition of `brake_at_correct_pressure`. Formal properties accurately capture their descriptions. The attacks under the updated attacker model are identified. The solution is clear and succinct.	8 marks
Relationship to HAZOP	Correct identification of hazards and guidewords.	2 marks
Total		20 marks

4 Submission

Submit the assignment using the submission link on the subject LMS. Go to the SWEN90010 LMS page, select *Assignments* from the subject menu, and then select *View/Complete* from the *Assignment 3 submission* item. Upload your commented `ebs.als` file, ensuring that it has this file name and that the comments in the file clearly identify *both* authors in your pair.

[2] If you are unable to work out how to phrase your desired assertion formally, at least describe it in the comments.

Late submissions Late submissions will attract a penalty of 2 marks 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. The content required for this assignment was presented before the assignment was released, so an early start is possible (and encouraged).

5 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 individual'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.

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.