Exam 3

Bharath Karumudi

September 15, 2019

Abstract

This project is to demonstrate the capabilities of implementing constructing and deconstructing HOL Terms using the tools and techniques - LaTeX, AcuTeX, emacs and ML.

Each chapter documents the given problems with a structure of:

1. Problem Statement
2. Relevant Code
3. Execution Transcripts
4. Explanation of results

Acknowledgments: Professor Marvine Hamner and Professor Shiu-Kai Chin who taught the Certified Security By Design.

Contents

Chapter 1

Executive Summary

All requirements for this project are statisfied specifically, and by using HOL solved the below questions:

1. Question 1
2. Question 2
3. Question 3

Chapter 2

Question 1

2.1 Problem Statement

Create Access control matrix by considering the following relationships:

File 1 comprises the engineering files, including the engineering drawings Alice is working on for a new, custom, design which are considered proprietary. File 2 are the accounting departments records including for Mr. Coyotes account with Acme, past billing, payments, project data (hours) for custom projects, etc., as well as various company data regarding assets, liabilities, revenue, etc. File 3 is a relational database that stores all Acmes basic account information for its customers, e.g. Customer name, address, type of business, contacts with the customer (data, Acme rep contacting customer, product(s) discussed, type of contact e.g. phone call, letter, etc.), and past purchases including date of purchase, product(s) purchased, etc. This relational database is managed by Acmes Customer Relationship Department. File 4 contains Acmes CEO, Mr. Knowsitall, personal files. For example, this include the correspondence with Acmes customers as is typically generated by accessing Acmes relational database to merge product data, accounting data used to create proposed product pricing, etc.

2.2 Relevant Code

	File1	File2	File3	File4
Alice	r,w,x	r,w		
Bob	r	r,w,x	r	
Eve	r	r,w	r	
Knowsitall	r,x	r,x	r,w,x	r,w,x

2.3 Explanation of Results

The above access matrix shows who has access to what files from the given information.

1. Alice will have complete access to File1, which is engineering files to perform her job, read and write on File2, to log the timekeeping and also to read her timesheets. The access on File2 also depends on how system is designed - to allow the reads on timesheets.

2. Bob will have complete access on File2, which is accounts. read-only access on File1 and File3 to audit the information.

3. Eve will have read and write on File2 to log her timekeeping and also to read her timesheets. read-only access on File1 and File3 to read designs and her customer(s) data.

4. Knowsitall has complete access on File3 and File4. read and execute on File1 and File2 to get the information to create product pricing. To maintain data integrity, though he is a CEO it is not required to have write access on File1 (Engineering Files) and File2 (Account department).

Chapter 3

Question 2

3.1 Problem Statement

Consider the access control matrix you generated and a concept of operations Using Higher Order Logic in which Eve, as Mr. Coyotes customer rep, wants to create a new letter for Mr. Coyote. Eve wants this letter to describe the wonderful things Acmes new, custom design will do for him and how little it will cost him to get it. Derive the inference rule that controls this situation and prove it

3.2 Relevant Code

```
1. Eve says <create letter, File2^File3>          Eve Request
2. Bob controls (Eve controls <read, File2>)      Access Policy
3. CRM controls (Eve controls <read, File3>)      Access Policy
4. ACL2 => Bob                                    Trust Assumption
5. ACL3 => CRM                                    Trust Assumption
6. ACL says Eve controls <read, File2> ^
            Eve controls <read, File3>            Access List
7. ACL says (Eve controls <read, File2 ^ File3)   6 simplify says
8. <create letter, File2^File3>                   1,2,3,4,5,7 ticket rule
```

```
(*****************************************************************************)
(*  Author: Bharath Karumudi                                                 *)
(*  Date: Sep 14, 2019                                                       *)
(*****************************************************************************)

structure question2Script = struct

open HolKernel boolLib Parse bossLib
open acl_infRules aclrulesTheory aclDrulesTheory

(*****************************************************************************)
(* Create New Theory                                                         *)
(*****************************************************************************)
val _ = new_theory "question2"

(*****************************************************************)
(* Defining instructions - grant, deny *)
(*****************************************************************)
val _ =
Hol_datatype
`commands = grant | deny`
```

```
(*************************************************)
(* Defining Principals  - Eve and Bob            *)
(*************************************************)
val _ =
Hol_datatype
`people = Eve | Bob`

(*************************************************************************)
(* Define roles                                                          *)
(*************************************************************************)
val _ =
Hol_datatype
`roles = owner | requester`

(*************************************************************************)
(* Define principals that will have keys                                 *)
(*************************************************************************)
val _ =
Hol_datatype
`keyPrinc = Staff of people | Role of roles | Ap of num`

(*************************************************************************)
(* Define principals as keyPrinc and keys                                *)
(*************************************************************************)
val _ =
Hol_datatype
`principals =  PR of keyPrinc | Key of keyPrinc`

(*************************************************************************)
(* Proof for question2Thm                                                *)
(*************************************************************************)

val question2Thm =
let
 val th1 = ACL_ASSUM``((Name (PR(Role owner))) controls (prop grant)):(commands,principals
 val th2 = ACL_ASSUM``(reps(Name(PR (Staff Eve)))(Name(PR(Role owner))) (prop grant)):(comm
 val th3 = ACL_ASSUM``((Name(Key(Staff Eve))) quoting (Name((PR(Role owner)))) says (prop g
 val th4 = ACL_ASSUM``((prop grant) impf (prop deny)):(commands,principals,'d,'e)Form``
 val th5 = ACL_ASSUM``((Name(Key(Role requester))) speaks_for (Name(PR(Role requester)))):(
 val th6 = ACL_ASSUM``((Name(Key(Role requester))) says ((Name(Key(Staff Eve)) speaks_for
 val th7 = ACL_ASSUM``((Name(PR(Role requester))) controls ((Name(Key(Staff Eve)) speaks_fo
 val th8 = SPEAKS_FOR th5 th6
 val th9 = CONTROLS th7 th8
 val th10 = IDEMP_SPEAKS_FOR``(Name((PR(Role owner))))``
 val th11 = INST_TYPE[``:'a`` |-> ``:commands``] th10
 val th12 = MONO_SPEAKS_FOR th9 th11
 val th13 = SPEAKS_FOR th12 th3
 val th14 = REPS th2 th13 th1
 val th15 = ACL_MP th14 th4
 val th16 = SAYS ``((Name(Key(Staff Bob))) quoting (Name(PR(Role Operator)))):principals P
 val th17 = DISCH(hd(hyp th7)) th16
 val th18 = DISCH(hd(hyp th6)) th17
```

```
 val th19 = DISCH(hd(hyp th5)) th18
 val th20 = DISCH(hd(hyp th4)) th19
 val th21 = DISCH(hd(hyp th3)) th20
 val th22 = DISCH(hd(hyp th2)) th21
in
 DISCH(hd(hyp th1)) th22
end;
val _ = save_thm("question2Thm",question2Thm)


(*****************************************************************************)
(* Exporting Theory                                                          *)
(*****************************************************************************)

val _ = print_theory "-";

val _ = export_theory();

end (* structure *)
```

3.3 Execution Transcripts

1

```
------------------------------------------------------------------
       HOL-4 [Kananaskis 11 (stdknl, built Sat Aug 19 09:30:06 2017)]

       For introductory HOL help, type: help "hol";
       To exit type <Control>-D
------------------------------------------------------------------
[extending loadPath with Holmakefile INCLUDES variable]
> > > > *** Globals.show_assums now true ***
> *** Globals.show_assums now false ***
> # # <<HOL message: Defined type: "commands">>
> # # <<HOL message: Defined type: "people">>
> # # <<HOL message: Defined type: "roles">>
> # # <<HOL message: Defined type: "keyPrinc">>
> # # <<HOL message: Defined type: "principals">>
> val question2Thm =
   |- (M,Oi,Os) sat Name (PR (Role owner)) controls prop grant
   (M,Oi,Os) sat
   reps (Name (PR (Staff Eve))) (Name (PR (Role owner))) (prop grant)
   (M,Oi,Os) sat
   Name (Key (Staff Eve)) quoting Name (PR (Role owner)) says
   prop grant
   (M,Oi,Os) sat prop grant impf prop deny
   (M,Oi,Os) sat
   Name (Key (Role requester)) speaks_for Name (PR (Role requester))
   (M,Oi,Os) sat
   Name (Key (Role requester)) says
   Name (Key (Staff Eve)) speaks_for Name (PR (Staff Eve))
   (M,Oi,Os) sat
   Name (PR (Role requester)) controls
   Name (Key (Staff Eve)) speaks_for Name (PR (Staff Eve))
   (M,Oi,Os) sat
   Name (Key (Staff Bob)) quoting Name (PR (Role Operator)) says
   prop deny:
   thm
val it = (): unit
>
*** Emacs/HOL command completed ***

>
```

3.3.1 Explanation of Results

The above results shows that the requirements are satisfied.

Chapter 4

Question 3

4.1 Problem Statement

Discuss if the derived inference rule has been proven and if so is adequate to answer Mr, Coyotes concerns about security. If it is adequate to answer these concerns, you need to support that assertion. That is, you need to specifically say why it is adequate. In addition, briefly discuss if this access control system could result in Eve providing information to Acmes competitors, which would enable them to work for or against Acmes customers.

4.2 Summary

The derived inference rule was proved but it is not fully adequate to answer the Mr. Coyote's concerns on the security of complete system. Because this could answer on access to the files. But even with the read access how the system is restricting the data sharing to outside needs to be addressed.

Also this access control system **could result** in Eve providing information to Acmes competitors, which would enable them to work for or against Acmes customers. Because Eve has access to the Engineering files, where she can get the new custom designs that are being worked by Engineering teams and also the customers information. So if Eve is a bad person, she can read this information and share it to the competitors. This can be refered as **internal threat**.

4.2.1 Conclusion

More detailed information is needed to answer the Coyote's question and this will also helps how the internal threats can be controlled.

Chapter 5

Appendix A: Question 2

The following code is from the file question2Script.sml

```
(*****************************************************************************)
(*   Author: Bharath Karumudi                                                *)
(*   Date: Sep 14, 2019                                                      *)
(*****************************************************************************)

structure question2Script = struct

open HolKernel boolLib Parse bossLib
open acl_infRules aclrulesTheory aclDrulesTheory

(*****************************************************************************)
(* Create New Theory                                                         *)
(*****************************************************************************)
val _ = new_theory "question2"

(*****************************************************************)
(* Defining instructions - grant, deny *)
(*****************************************************************)
val _ =
Hol_datatype
`commands = grant | deny`


(*****************************************************)
(* Defining Principals  - Eve and Bob                *)
(*****************************************************)
val _ =
Hol_datatype
`people = Eve | Bob`

(*****************************************************************************)
(* Define roles                                                              *)
(*****************************************************************************)
val _ =
Hol_datatype
`roles = owner | requester`

(*****************************************************************************)
(* Define principals that will have keys                                     *)
(*****************************************************************************)
val _ =
Hol_datatype
```

```
‘keyPrinc = Staff of people | Role of roles | Ap of num‘

(*************************************************************************)
(* Define principals as keyPrinc and keys                                *)
(*************************************************************************)
val _ =
Hol_datatype
‘principals = PR of keyPrinc | Key of keyPrinc‘

(*************************************************************************)
(* Proof for question2Thm                                                *)
(*************************************************************************)

val question2Thm =
let
 val th1 = ACL_ASSUM‘‘((Name (PR(Role owner))) controls (prop grant)):(commands,principals
 val th2 = ACL_ASSUM‘‘(reps(Name(PR (Staff Eve)))(Name(PR(Role owner))) (prop grant)):(comm
 val th3 = ACL_ASSUM‘‘((Name(Key(Staff Eve))) quoting (Name((PR(Role owner)))) says (prop g
 val th4 = ACL_ASSUM‘‘((prop grant) impf (prop deny)):(commands,principals,'d,'e)Form‘‘
 val th5 = ACL_ASSUM‘‘((Name(Key(Role requester))) speaks_for (Name(PR(Role requester)))):(
 val th6 = ACL_ASSUM‘‘((Name(Key(Role requester))) says ((Name(Key(Staff Eve)) speaks_for
 val th7 = ACL_ASSUM‘‘((Name(PR(Role requester))) controls ((Name(Key(Staff Eve)) speaks_fo
 val th8 = SPEAKS_FOR th5 th6
 val th9 = CONTROLS th7 th8
 val th10 = IDEMP_SPEAKS_FOR‘‘(Name((PR(Role owner))))‘‘
 val th11 = INST_TYPE[‘‘:'a‘‘ |-> ‘‘:commands‘‘] th10
 val th12 = MONO_SPEAKS_FOR th9 th11
 val th13 = SPEAKS_FOR th12 th3
 val th14 = REPS th2 th13 th1
 val th15 = ACL_MP th14 th4
 val th16 = SAYS ‘‘((Name(Key(Staff Bob))) quoting (Name(PR(Role Operator)))):principals P
 val th17 = DISCH(hd(hyp th7)) th16
 val th18 = DISCH(hd(hyp th6)) th17
 val th19 = DISCH(hd(hyp th5)) th18
 val th20 = DISCH(hd(hyp th4)) th19
 val th21 = DISCH(hd(hyp th3)) th20
 val th22 = DISCH(hd(hyp th2)) th21
in
 DISCH(hd(hyp th1)) th22
end;
val _ = save_thm("question2Thm",question2Thm)


(*************************************************************************)
(* Exporting Theory                                                      *)
(*************************************************************************)

val _ = print_theory "-";

val _ = export_theory();

end (* structure *)
```