Dictionary-Based Attack Design

Eunsaem Lee

December 8th, 2022

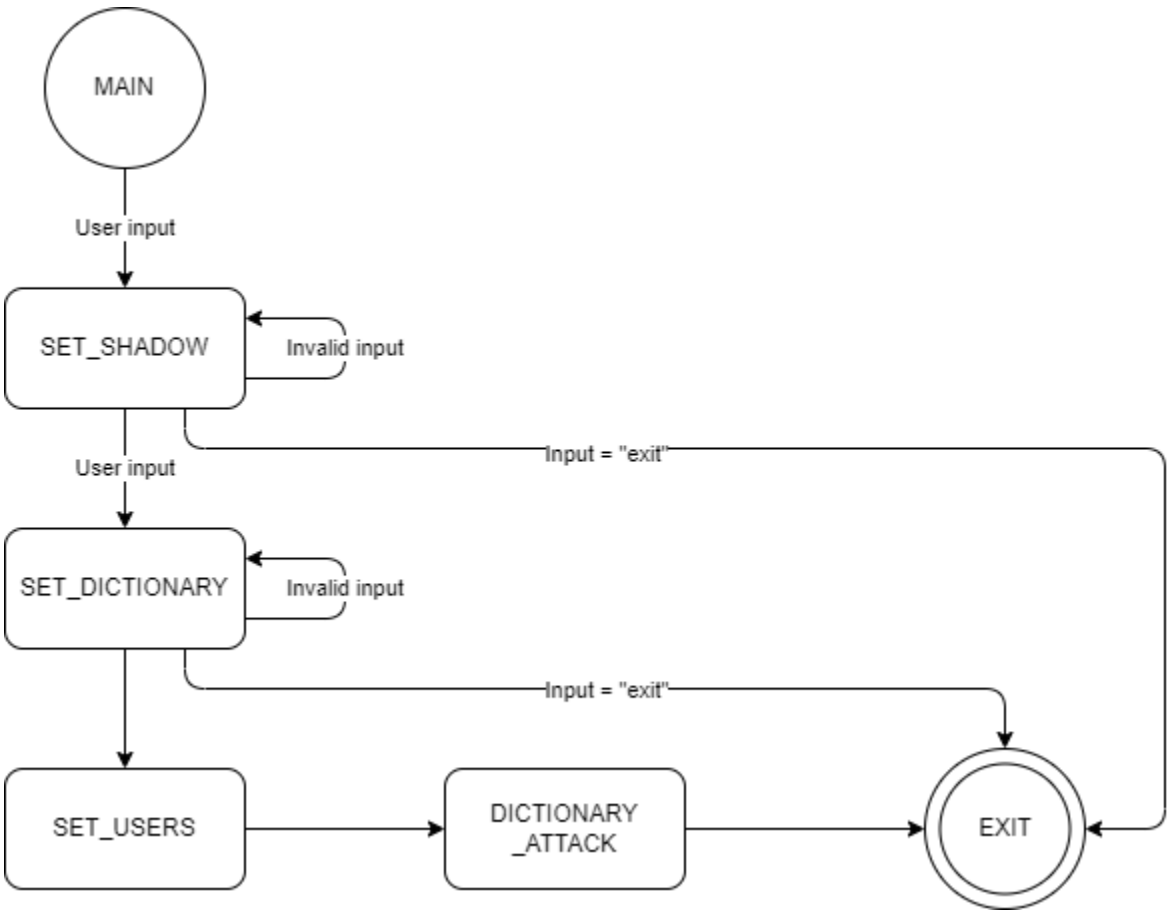
Table of Contents

Finite State Machine

State Table

From State	To State	Action
MAIN	SET_SHADOW	init_state
SET_SHADOW	SET_DICTIONARY	read_commands
SET_DICTIONARY	SET_USERS	handler_error
SET_USERS	DICTIONARY_ATTACK	reset_state
DICTIONARY_ATTACK	EXIT	separate_commands

State Transition Diagram



Functions

main

Purpose

Initialize the state object.

Parameters

The state object to initialize.

Return

Type	Next State
Success	`SET_SHADOW`
Failure	`ERROR`

Pseudocode

```
If any errors occur
     return ERROR
```

set_shadow

Purpose

Sets file path of shadow file. If input is "exit", exit the program. If invalid input, re-ask for shadow file path.

Parameters

None.

Return

Type	Next State
Success	SET_DICTIONARY
Failure	ERROR

Pseudocode

```
If any errors occur
     return ERROR

Prompts path input for shadow file from the user.

If the path input is "exit",
     exit the program.

Checks if path exists.
     If it exists, checks if the path is a file.

     If not, prints out an ERROR message and re-asks for
shadow file path.

Checks if the path is a file.
     If it is a path, sets the path to shadow file.

     If not, prints out an ERROR message and re-asks for
shadow file path.
```

set_dictionary

Purpose

Sets file path of dictionary file. If input is "exit", exit the program. If invalid input, re-ask for dictionary file path.

Parameters

None.

Return

Type	Next State
Success	SET_USERS
Failure	ERROR

Pseudocode

```
If any errors occur
     return ERROR

Prompts path input for dictionary file from the user.

If the path input is "exit",
     exit the program.

Checks if path exists.
     If it exists, checks if the path is a file.

     If not, prints out an ERROR message and re-asks for
dictionary file path.

Checks if the path is a file.
     If it is a path, sets the path to dictionary file.

     If not, prints out an ERROR message and re-asks for
dictionary file path.
```

set_users

Purpose

Reads user information from shadow file and appends them to "users" list.

Users: a list of users in the shadow file
[user1, user2, …]
User: a list of user information
[username, salt value, hashed password]

Parameters

None.

Return

Type	Next State
Success	DICTIONARY_ATTACK
Failure	ERROR

Pseudocode

```
If any errors occur
     return ERROR

Reads user information from shadow file and creates a user
list with the following information:
```

- username
- salt value
- hashed password

```
If the user information does not contain a password,
     skips the user.

Appends the user to a "users" list.
```

dictionary_attack

Purpose

Runs the dictionary-based attack and determines whether a dictionary word matches the password from shadow file.

Parameters

None.

Return

Type	Next State
Success	`EXIT`
Failure	`MENU`

Pseudocode

```
If any errors occur
     return ERROR

Reads a hashed password from the "users" list.

Iterates through the words in the dictionary until there is a match.
     If there is a match, prints the username and password on the command line and writes them into a log file.

     If there is no match, moves onto the next hashed password in the "users" list.
```