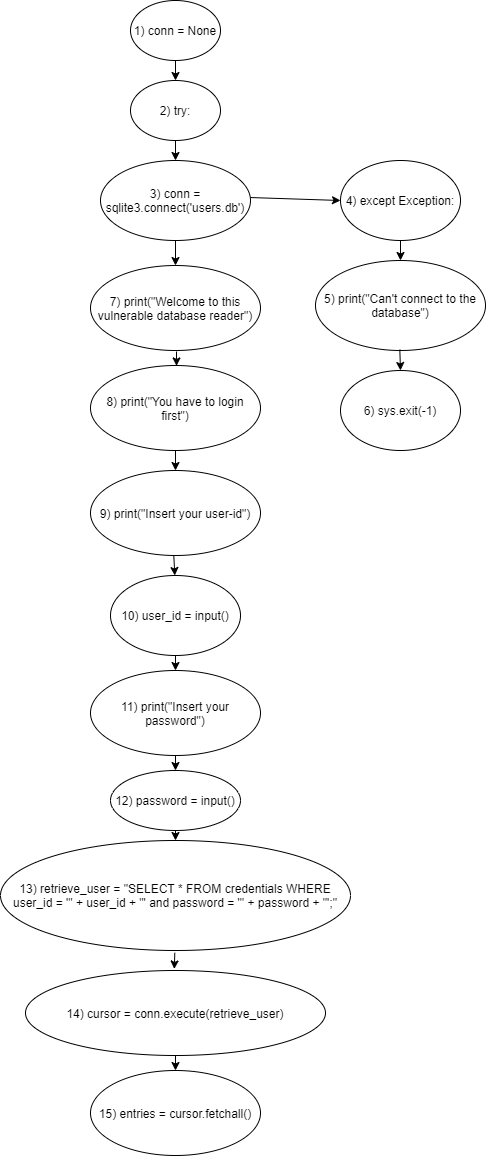
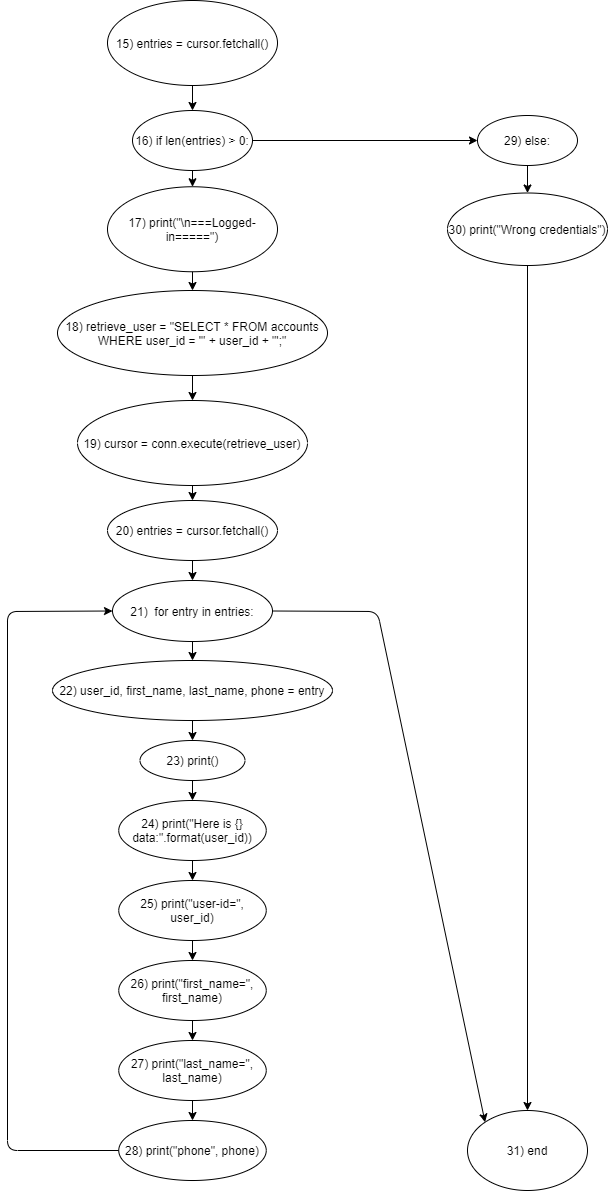
VINCI NICOLO’ 220229

TAINT ANALYSIS

EXERCISE 1: sql-injection

First of all, I do not consider the first three lines for buildin the CFG, where there are only some imports.

STEP 0: draw CFG.



STEP 1 and STEP 2: compute GEN and KILL, initialize IN and OUT.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | GEN | KILL | IN | OUT |
| 1 | {} | {} | {} | {} |
| 2 | {} | {} | {} | {} |
| 3 | {} | {} | {} | {} |
| 4 | {} | {} | {} | {} |
| 5 | {} | {} | {} | {} |
| 6 | {} | {} | {} | {} |
| 7 | {} | {} | {} | {} |
| 8 | {} | {} | {} | {} |
| 9 | {} | {} | {} | {} |
| 10 | {user\_id} | {} | {} | {user\_id} |
| 11 | {} | {} | {} | {} |
| 12 | {password} | {} | {} | {password} |
| 13 | {[user\_id->T |  password->T]retreive\_user} | {[user\_id->F &  password->F]retreive\_user} | {} | {[user\_id->T |  password->T]retreive\_user} |
| 14 | {[retrieve\_user->T]cursor} | {[retrieve\_user->F]cursor} | {} | {[retrieve\_user->T]cursor} |
| 15 | {[cursor->T]entries} | {[cursor->F]entries} | {} | {[cursor->T]entries} |
| 16 | {} | {} | {} | {} |
| 17 | {} | {} | {} | {} |
| 18 | {[user\_id->T]retrieve\_user} | {[user\_id->F]retrieve\_user} | {} | {[user\_id->T]retrieve\_user} |
| 19 | {[retrieve\_user->T]cursor} | {[retrieve\_user->F]cursor} | {} | {[retrieve\_user->T]cursor} |
| 20 | {[cursor->T]entries} | {[cursor->F]entries} | {} | {[cursor->T]entries} |
| 21 | {[entries->T]entry} | {[entries->F]entry} | {} | {[entries->T]entry} |
| 22 | {[entry->T]user\_id, first\_name, last\_name, phone} | {[entry->F]user\_id, first\_name, last\_name, phone} | {} | {[entry->T]user\_id, first\_name, last\_name, phone} |
| 23 | {} | {} | {} | {} |
| 24 | {} | {} | {} | {} |
| 25 | {} | {} | {} | {} |
| 26 | {} | {} | {} | {} |
| 27 | {} | {} | {} | {} |
| 28 | {} | {} | {} | {} |
| 29 | {} | {} | {} | {} |
| 30 | {} | {} | {} | {} |
| 31 | {} | {} | {} | {} |

STEP 3: compute IN and OUT with transfer function.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | GEN | KILL | IN | OUT |
| 1 | {} | {} | {} | {} |
| 2 | {} | {} | {} | {} |
| 3 | {} | {} | {} | {} |
| 4 | {} | {} | {} | {} |
| 5 | {} | {} | {} | {} |
| 6 | {} | {} | {} | {} |
| 7 | {} | {} | {} | {} |
| 8 | {} | {} | {} | {} |
| 9 | {} | {} | {} | {} |
| 10 | {user\_id} | {} | {} | {user\_id} |
| 11 | {} | {} | {user\_id} | {user\_id} |
| 12 | {password} | {} | {user\_id} | {user\_id, password} |
| 13 | {[user\_id->T |  password->T]retreive\_user} | {[user\_id->F &  password->F]retreive\_user} | {user\_id, password} | {user\_id, password, retrieve\_user} |
| 14 | {[retrieve\_user->T]cursor} | {[retrieve\_user->F]cursor} | {user\_id, password, retrieve\_user} | {user\_id, password, retrieve\_user, cursor} |
| 15 | {[cursor->T]entries} | {[cursor->F]entries} | {user\_id, password, retrieve\_user, cursor} | {user\_id, password, retrieve\_user, cursor, entries} |
| 16 | {} | {} | {user\_id, password, retrieve\_user, cursor, entries} | {user\_id, password, retrieve\_user, cursor, entries} |
| 17 | {} | {} | {user\_id, password, retrieve\_user, cursor, entries} | {user\_id, password, retrieve\_user, cursor, entries} |
| 18 | {[user\_id->T]retrieve\_user} | {[user\_id->F]retrieve\_user} | {user\_id, password, retrieve\_user, cursor, entries} | {user\_id, password, retrieve\_user, cursor, entries} |
| 19 | {[retrieve\_user->T]cursor} | {[retrieve\_user->F]cursor} | {user\_id, password, retrieve\_user, cursor, entries} | {user\_id, password, retrieve\_user, cursor, entries} |
| 20 | {[cursor->T]entries} | {[cursor->F]entries} | {user\_id, password, retrieve\_user, cursor, entries} | {user\_id, password, retrieve\_user, cursor, entries} |
| 21 | {[entries->T]entry} | {[entries->F]entry} | {user\_id, password, retrieve\_user, cursor, entries} | {user\_id, password, retrieve\_user, cursor, entries, entry} |
| 22 | {[entry->T]user\_id, first\_name, last\_name, phone} | {[entry->F]user\_id, first\_name, last\_name, phone} | {user\_id, password, retrieve\_user, cursor, entries, entry} | {  user\_id, password, retrieve\_user, cursor, entries, entry, first\_name, last\_name, phone } |
| 23 | {} | {} | {  user\_id, password, retrieve\_user, cursor, entries, entry, first\_name, last\_name, phone } | {  user\_id, password, retrieve\_user, cursor, entries, entry, first\_name, last\_name, phone } |
| 24 | {} | {} | {  user\_id, password, retrieve\_user, cursor, entries, entry, first\_name, last\_name, phone } | {  user\_id, password, retrieve\_user, cursor, entries, entry, first\_name, last\_name, phone } |
| 25 | {} | {} | {  user\_id, password, retrieve\_user, cursor, entries, entry, first\_name, last\_name, phone } | {  user\_id, password, retrieve\_user, cursor, entries, entry, first\_name, last\_name, phone } |
| 26 | {} | {} | {  user\_id, password, retrieve\_user, cursor, entries, entry, first\_name, last\_name, phone } | {  user\_id, password, retrieve\_user, cursor, entries, entry, first\_name, last\_name, phone } |
| 27 | {} | {} | {  user\_id, password, retrieve\_user, cursor, entries, entry, first\_name, last\_name, phone } | {  user\_id, password, retrieve\_user, cursor, entries, entry, first\_name, last\_name, phone } |
| 28 | {} | {} | {  user\_id, password, retrieve\_user, cursor, entries, entry, first\_name, last\_name, phone } | {  user\_id, password, retrieve\_user, cursor, entries, entry, first\_name, last\_name, phone } |
| 29 | {} | {} | {user\_id, password, retrieve\_user, cursor, entries} | {user\_id, password, retrieve\_user, cursor, entries} |
| 30 | {} | {} | {user\_id, password, retrieve\_user, cursor, entries} | {user\_id, password, retrieve\_user, cursor, entries} |
| 31 | {} | {} | {user\_id, password, retrieve\_user, cursor, entries, entry} | {user\_id, password, retrieve\_user, cursor, entries, entry} |

In line 14 , the program makes a tainted query due to the tainted variable “retrieve\_user”, that it is tainted by user input variable “user\_id” and “password”.

In line 19, it makes another tainted query due to user input variable “user\_id” that is tainted.

In lines 24, 25, 26, 27, 28, it prints some tainted variables due to tainted variable “entry”.

STEP 4: iterates at least once to check the tainted variable.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | GEN | KILL | IN | OUT |
| 1 | {} | {} | {} | {} |
| 2 | {} | {} | {} | {} |
| 3 | {} | {} | {} | {} |
| 4 | {} | {} | {} | {} |
| 5 | {} | {} | {} | {} |
| 6 | {} | {} | {} | {} |
| 7 | {} | {} | {} | {} |
| 8 | {} | {} | {} | {} |
| 9 | {} | {} | {} | {} |
| 10 | {user\_id} | {} | {} | {user\_id} |
| 11 | {} | {} | {user\_id} | {user\_id} |
| 12 | {password} | {} | {user\_id} | {user\_id, password} |
| 13 | {[user\_id->T |  password->T]retreive\_user} | {[user\_id->F &  password->F]retreive\_user} | {user\_id, password} | {user\_id, password, retrieve\_user} |
| 14 | {[retrieve\_user->T]cursor} | {[retrieve\_user->F]cursor} | {user\_id, password, retrieve\_user} | {user\_id, password, retrieve\_user, cursor} |
| 15 | {[cursor->T]entries} | {[cursor->F]entries} | {user\_id, password, retrieve\_user, cursor} | {user\_id, password, retrieve\_user, cursor, entries} |
| 16 | {} | {} | {user\_id, password, retrieve\_user, cursor, entries} | {user\_id, password, retrieve\_user, cursor, entries} |
| 17 | {} | {} | {user\_id, password, retrieve\_user, cursor, entries} | {user\_id, password, retrieve\_user, cursor, entries} |
| 18 | {[user\_id->T]retrieve\_user} | {[user\_id->F]retrieve\_user} | {user\_id, password, retrieve\_user, cursor, entries} | {user\_id, password, retrieve\_user, cursor, entries} |
| 19 | {[retrieve\_user->T]cursor} | {[retrieve\_user->F]cursor} | {user\_id, password, retrieve\_user, cursor, entries} | {user\_id, password, retrieve\_user, cursor, entries} |
| 20 | {[cursor->T]entries} | {[cursor->F]entries} | {user\_id, password, retrieve\_user, cursor, entries} | {user\_id, password, retrieve\_user, cursor, entries} |
| 21 | {[entries->T]entry} | {[entries->F]entry} | {  user\_id, password, retrieve\_user, cursor,  entries, entry, first\_name, last\_name, phone  } | {  user\_id, password, retrieve\_user, cursor,  entries, entry, first\_name, last\_name, phone  } |
| 22 | {[entry->T]user\_id, first\_name, last\_name, phone} | {[entry->F]user\_id, first\_name, last\_name, phone} | {  user\_id, password, retrieve\_user, cursor,  entries, entry, first\_name, last\_name, phone  } | {  user\_id, password, retrieve\_user, cursor,  entries, entry, first\_name, last\_name, phone  } |
| 23 | {} | {} | {  user\_id, password, retrieve\_user, cursor,  entries, entry, first\_name, last\_name, phone  } | {  user\_id, password, retrieve\_user, cursor,  entries, entry, first\_name, last\_name, phone  } |
| 24 | {} | {} | {  user\_id, password, retrieve\_user, cursor,  entries, entry, first\_name, last\_name, phone  } | {  user\_id, password, retrieve\_user, cursor,  entries, entry, first\_name, last\_name, phone  } |
| 25 | {} | {} | {  user\_id, password, retrieve\_user, cursor,  entries, entry, first\_name, last\_name, phone  } | {  user\_id, password, retrieve\_user, cursor,  entries, entry, first\_name, last\_name, phone  } |
| 26 | {} | {} | {  user\_id, password, retrieve\_user, cursor,  entries, entry, first\_name, last\_name, phone  } | {  user\_id, password, retrieve\_user, cursor,  entries, entry, first\_name, last\_name, phone  } |
| 27 | {} | {} | {  user\_id, password, retrieve\_user, cursor,  entries, entry, first\_name, last\_name, phone  } | {  user\_id, password, retrieve\_user, cursor,  entries, entry, first\_name, last\_name, phone  } |
| 28 | {} | {} | {  user\_id, password, retrieve\_user, cursor,  entries, entry, first\_name, last\_name, phone  } | {  user\_id, password, retrieve\_user, cursor,  entries, entry, first\_name, last\_name, phone  } |
| 29 | {} | {} | {user\_id, password, retrieve\_user, cursor, entries} | {user\_id, password, retrieve\_user, cursor, entries} |
| 30 | {} | {} | {user\_id, password, retrieve\_user, cursor, entries} | {user\_id, password, retrieve\_user, cursor, entries} |
| 31 | {} | {} | {  user\_id, password, retrieve\_user, cursor,  entries, entry, first\_name, last\_name, phone  } | {  user\_id, password, retrieve\_user, cursor,  entries, entry, first\_name, last\_name, phone  } |

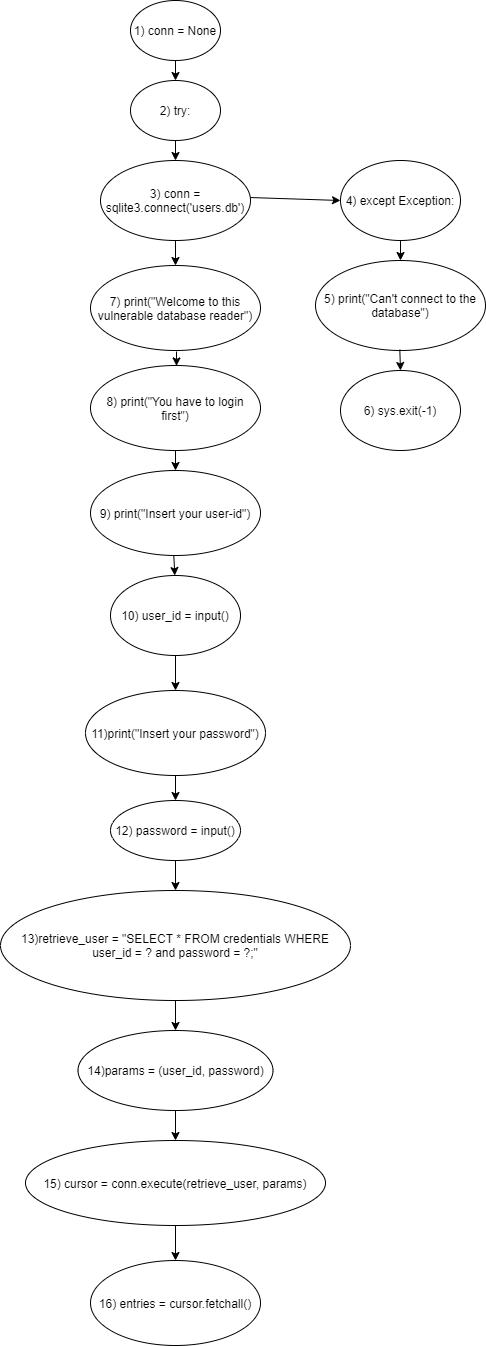
The node 21 is the only changed. It gets in input the output of node 20 and the output of node 28, the latter has some tainted variables in output now. Instead, in step 3 the output of node 28 comes to step 2 (initialization) that was empty.

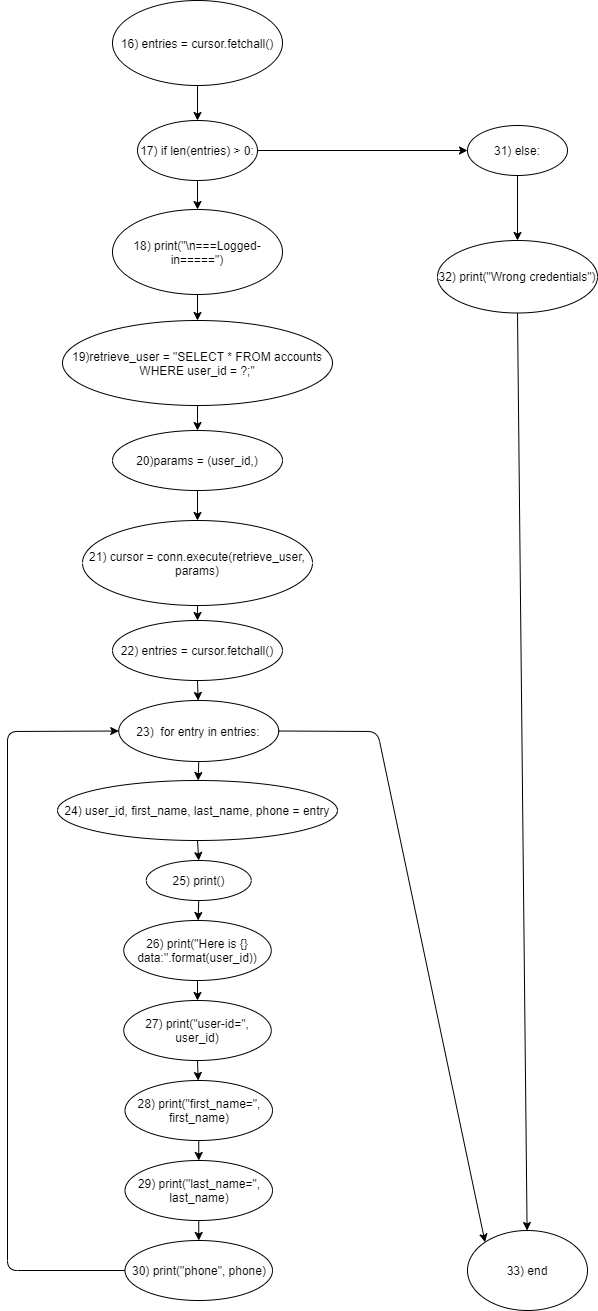
Anyway, the program makes two tainted queries to database and fetches the results, without sanitization of the two input variables.

FIXED sql-injection:

I have to check the two input variables to avoid a SQL injection attack. One solution may be to do parameterized query. So, I introduce the variable “params” to intepret the user input as a string so that the user can not concatenate his input with the query, because the query is built at run time binding the original query with parameters.

STEP 0: draw CFG.





STEP 1 and STEP 2: compute GEN and KILL, initialize IN and OUT.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | GEN | KILL | IN | OUT |
| 1 | {} | {} | {} | {} |
| 2 | {} | {} | {} | {} |
| 3 | {} | {} | {} | {} |
| 4 | {} | {} | {} | {} |
| 5 | {} | {} | {} | {} |
| 6 | {} | {} | {} | {} |
| 7 | {} | {} | {} | {} |
| 8 | {} | {} | {} | {} |
| 9 | {} | {} | {} | {} |
| 10 | {user\_id} | {} | {} | {user\_id} |
| 11 | {} | {} | {} | {} |
| 12 | {password} | {} | {} | {password} |
| 13 | {} | {} | {} | {} |
| 14 | {[user\_id->T |  password->T]params} | {[user\_id->F &  password->F]params} | {} | {[user\_id->T |  password->T]params} |
| 15 | {} | {cursor, params} | {} | {} |
| 16 | {[cursor->T]entries} | {[cursor->F]entries} | {} | {[cursor->T]entries} |
| 17 | {} | {} | {} | {} |
| 18 | {} | {} | {} | {} |
| 19 | {} | {} | {} | {} |
| 20 | {[user\_id->T]params} | {[user\_id->F]params} | {} | {[user\_id->T]params} |
| 21 | {} | {cursor, params} | {} | {} |
| 22 | {[cursor->T]entries} | {[cursor->F]entries} | {} | {[cursor->T]entries} |
| 23 | {[entries->T]entry} | {[entries->F]entry} | {} | {[entries->T]entry} |
| 24 | {[entry->T]user\_id,  first\_name,last\_name,phone} | {[entry->F]user\_id,  first\_name,last\_name,phone} | {} | {[entry->T]user\_id,  first\_name,last\_name,phone} |
| 25 | {} | {} | {} | {} |
| 26 | {} | {} | {} | {} |
| 27 | {} | {} | {} | {} |
| 28 | {} | {} | {} | {} |
| 29 | {} | {} | {} | {} |
| 30 | {} | {} | {} | {} |
| 31 | {} | {} | {} | {} |
| 32 | {} | {} | {} | {} |
| 33 | {} | {} | {} | {} |

STEP 3: compute IN and OUT with transfer function.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | GEN | KILL | IN | OUT |
| 1 | {} | {} | {} | {} |
| 2 | {} | {} | {} | {} |
| 3 | {} | {} | {} | {} |
| 4 | {} | {} | {} | {} |
| 5 | {} | {} | {} | {} |
| 6 | {} | {} | {} | {} |
| 7 | {} | {} | {} | {} |
| 8 | {} | {} | {} | {} |
| 9 | {} | {} | {} | {} |
| 10 | {user\_id} | {} | {} | {user\_id} |
| 11 | {} | {} | {user\_id} | {user\_id} |
| 12 | {password} | {} | {user\_id} | {user\_id, password} |
| 13 | {} | {} | {user\_id, password} | {user\_id, password} |
| 14 | {[user\_id->T |  password->T]params} | {[user\_id->F &  password->F]params} | {user\_id, password} | {user\_id, password, params} |
| 15 | {} | {cursor, params} | {user\_id, password, params} | {user\_id, password} |
| 16 | {[cursor->T]entries} | {[cursor->F]entries} | {user\_id, password} | {user\_id, password} |
| 17 | {} | {} | {user\_id, password} | {user\_id, password} |
| 18 | {} | {} | {user\_id, password} | {user\_id, password} |
| 19 | {} | {} | {user\_id, password} | {user\_id, password} |
| 20 | {[user\_id->T]params} | {[user\_id->F]params} | {user\_id, password} | {user\_id, password, params} |
| 21 | {} | {cursor, params} | {user\_id, password, params} | {user\_id, password} |
| 22 | {[cursor->T]entries} | {[cursor->F]entries} | {user\_id, password} | {user\_id, password} |
| 23 | {[entries->T]entry} | {[entries->F]entry} | {user\_id, password} | {user\_id, password} |
| 24 | {[entry->T]user\_id,  first\_name,last\_name,phone} | {[entry->F]user\_id,  first\_name,last\_name,phone} | {user\_id, password} | {user\_id, password} |
| 25 | {} | {} | {user\_id, password} | {user\_id, password} |
| 26 | {} | {} | {user\_id, password} | {user\_id, password} |
| 27 | {} | {} | {user\_id, password} | {user\_id, password} |
| 28 | {} | {} | {user\_id, password} | {user\_id, password} |
| 29 | {} | {} | {user\_id, password} | {user\_id, password} |
| 30 | {} | {} | {user\_id, password} | {user\_id, password} |
| 31 | {} | {} | {user\_id, password} | {user\_id, password} |
| 32 | {} | {} | {user\_id, password} | {user\_id, password} |
| 33 | {} | {} | {user\_id, password} | {user\_id, password} |

Now, the program makes safe queries, because the two queries are built binding the input variables as parameters with the original query (Line 13, 21).

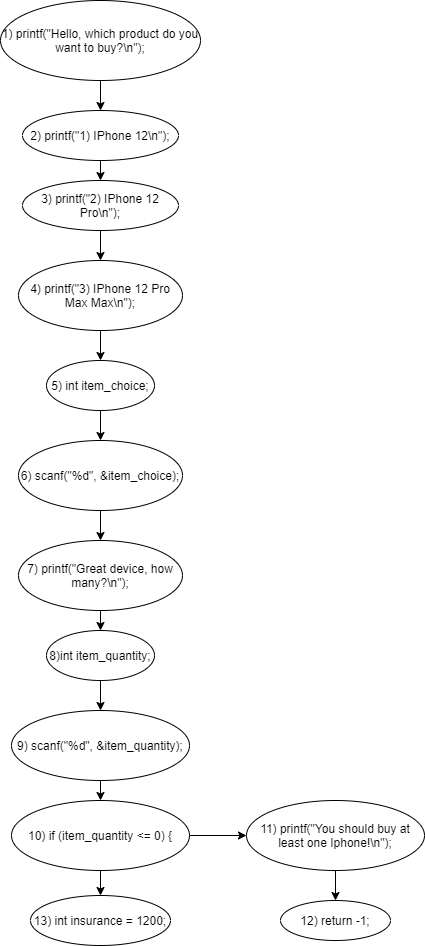
STEP 4: iterates at least once to check the tainted variable.

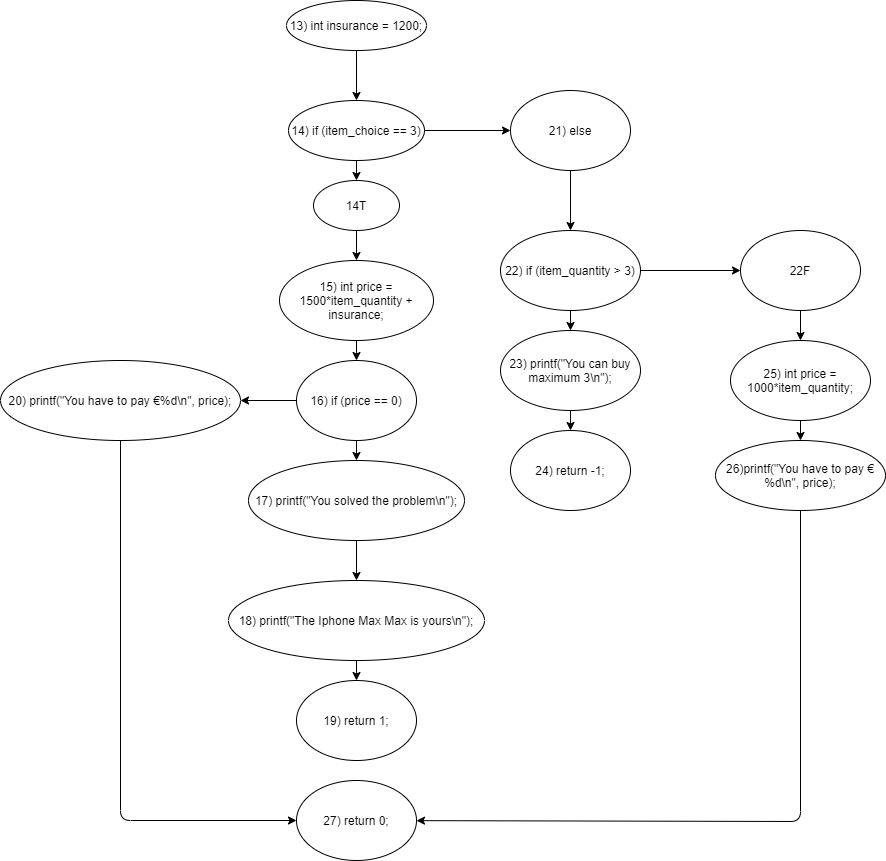
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | GEN | KILL | IN | OUT |
| 1 | {} | {} | {} | {} |
| 2 | {} | {} | {} | {} |
| 3 | {} | {} | {} | {} |
| 4 | {} | {} | {} | {} |
| 5 | {} | {} | {} | {} |
| 6 | {} | {} | {} | {} |
| 7 | {} | {} | {} | {} |
| 8 | {} | {} | {} | {} |
| 9 | {} | {} | {} | {} |
| 10 | {user\_id} | {} | {} | {user\_id} |
| 11 | {} | {} | {user\_id} | {user\_id} |
| 12 | {password} | {} | {user\_id} | {user\_id, password} |
| 13 | {} | {} | {user\_id, password} | {user\_id, password} |
| 14 | {[user\_id->T |  password->T]params} | {[user\_id->F &  password->F]params} | {user\_id, password} | {user\_id, password, params} |
| 15 | {} | {cursor, params} | {user\_id, password, params} | {user\_id, password} |
| 16 | {[cursor->T]entries} | {[cursor->F]entries} | {user\_id, password} | {user\_id, password} |
| 17 | {} | {} | {user\_id, password} | {user\_id, password} |
| 18 | {} | {} | {user\_id, password} | {user\_id, password} |
| 19 | {} | {} | {user\_id, password} | {user\_id, password} |
| 20 | {[user\_id->T]params} | {[user\_id->F]params} | {user\_id, password} | {user\_id, password, params} |
| 21 | {} | {cursor, params} | {user\_id, password, params} | {user\_id, password} |
| 22 | {[cursor->T]entries} | {[cursor->F]entries} | {user\_id, password} | {user\_id, password} |
| 23 | {[entries->T]entry} | {[entries->F]entry} | {user\_id, password} | {user\_id, password} |
| 24 | {[entry->T]user\_id,  first\_name,last\_name,phone} | {[entry->F]user\_id,  first\_name,last\_name,phone} | {user\_id, password} | {user\_id, password} |
| 25 | {} | {} | {user\_id, password} | {user\_id, password} |
| 26 | {} | {} | {user\_id, password} | {user\_id, password} |
| 27 | {} | {} | {user\_id, password} | {user\_id, password} |
| 28 | {} | {} | {user\_id, password} | {user\_id, password} |
| 29 | {} | {} | {user\_id, password} | {user\_id, password} |
| 30 | {} | {} | {user\_id, password} | {user\_id, password} |
| 31 | {} | {} | {user\_id, password} | {user\_id, password} |
| 32 | {} | {} | {user\_id, password} | {user\_id, password} |
| 33 | {} | {} | {user\_id, password} | {user\_id, password} |

The table remains the same as the step 3.

EXERCISE 2: integer\_overflow

STEP 0: draw CFG.





STEP 1 and STEP 2: compute GEN and KILL, initialize IN and OUT.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | GEN | KILL | IN | OUT |
| 1 | {} | {} | {} | {} |
| 2 | {} | {} | {} | {} |
| 3 | {} | {} | {} | {} |
| 4 | {} | {} | {} | {} |
| 5 | {} | {} | {} | {} |
| 6 | {item\_choice} | {} | {} | {item\_choice} |
| 7 | {} | {} | {} | {} |
| 8 | {} | {} | {} | {} |
| 9 | {item\_quantity} | {} | {} | {item\_quantity} |
| 10 | {} | {} | {} | {} |
| 11 | {} | {} | {} | {} |
| 12 | {} | {} | {} | {} |
| 13 | {} | {} | {} | {} |
| 14 | {} | {} | {} | {} |
| 14T | {} | {item\_choice} | {} | {} |
| 15 | {[item\_quantity->T]price} | {[item\_quantity->F]price} | {} | {[item\_quantity->T]price} |
| 16 | {} | {} | {} | {} |
| 17 | {} | {} | {} | {} |
| 18 | {} | {} | {} | {} |
| 19 | {} | {} | {} | {} |
| 20 | {} | {} | {} | {} |
| 21 | {} | {} | {} | {} |
| 22 | {} | {} | {} | {} |
| 22F | {} | {item\_quantity} | {} | {} |
| 23 | {} | {} | {} | {} |
| 24 | {} | {} | {} | {} |
| 25 | {[item\_quantity->T]price} | {[item\_quantity->F]price} | {} | {[item\_quantity->T]price} |
| 26 | {} | {} | {} | {} |
| 27 | {} | {} | {} | {} |

STEP 3: compute IN and OUT with transfer function.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | GEN | KILL | IN | OUT |
| 1 | {} | {} | {} | {} |
| 2 | {} | {} | {} | {} |
| 3 | {} | {} | {} | {} |
| 4 | {} | {} | {} | {} |
| 5 | {} | {} | {} | {} |
| 6 | {item\_choice} | {} | {} | {item\_choice} |
| 7 | {} | {} | { item\_choice } | { item\_choice } |
| 8 | {} | {} | { item\_choice } | { item\_choice } |
| 9 | {item\_quantity} | {} | { item\_choice } | { item\_choice, item\_quantity} |
| 10 | {} | {} | { item\_choice, item\_quantity} | { item\_choice, item\_quantity} |
| 11 | {} | {} | { item\_choice, item\_quantity} | { item\_choice, item\_quantity} |
| 12 | {} | {} | { item\_choice, item\_quantity} | { item\_choice, item\_quantity} |
| 13 | {} | {} | { item\_choice, item\_quantity} | { item\_choice, item\_quantity} |
| 14 | {} | {} | { item\_choice, item\_quantity} | { item\_choice, item\_quantity} |
| 14T | {} | {item\_choice} | { item\_choice, item\_quantity} | { item\_quantity} |
| 15 | {[item\_quantity->T]price} | {[item\_quantity->F]price} | {item\_quantity} | { item\_quantity, price} |
| 16 | {} | {} | {item\_quantity, price} | {item\_quantity, price} |
| 17 | {} | {} | {item\_quantity, price} | {item\_quantity, price} |
| 18 | {} | {} | {item\_quantity, price} | {item\_quantity, price} |
| 19 | {} | {} | {item\_quantity, price} | {item\_quantity, price} |
| 20 | {} | {} | {item\_quantity, price} | {item\_quantity, price} |
| 21 | {} | {} | { item\_choice, item\_quantity} | { item\_choice, item\_quantity} |
| 22 | {} | {} | { item\_choice, item\_quantity} | { item\_choice, item\_quantity} |
| 22F | {} | {item\_quantity} | { item\_choice, item\_quantity} | { item\_choice} |
| 23 | {} | {} | { item\_choice, item\_quantity} | { item\_choice, item\_quantity} |
| 24 | {} | {} | { item\_choice, item\_quantity} | { item\_choice, item\_quantity} |
| 25 | {[item\_quantity->T]price} | {[item\_quantity->F]price} | { item\_choice} | { item\_choice} |
| 26 | {} | {} | { item\_choice} | { item\_choice} |
| 27 | {} | {} | { item\_choice, item\_quantity, price} | { item\_choice, item\_quantity, price} |

In this first iteration, I can observe from line 15 that the program makes an operation with a tainted input variable “item\_quantity” and it could overflow. Moreover, in line 20 the program prints the tainted variable “price” that it is tainted due to the input variable “item\_quantity”.

Instead, in the line 25 the program makes an operation with the input variable “item\_quantity” that it is untainted due to the check at line 22 and the initial check at line 10.

STEP 4: iterates at least once to check the tainted variable.

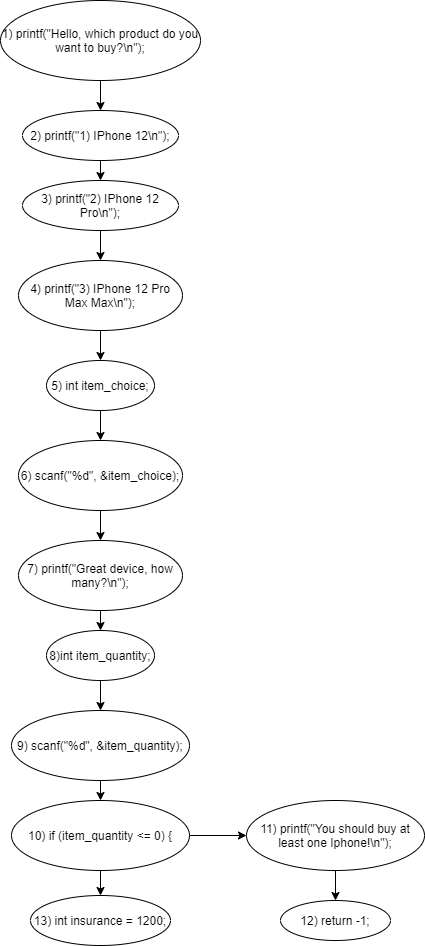
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | GEN | KILL | IN | OUT |
| 1 | {} | {} | {} | {} |
| 2 | {} | {} | {} | {} |
| 3 | {} | {} | {} | {} |
| 4 | {} | {} | {} | {} |
| 5 | {} | {} | {} | {} |
| 6 | {item\_choice} | {} | {} | {item\_choice} |
| 7 | {} | {} | { item\_choice } | { item\_choice } |
| 8 | {} | {} | { item\_choice } | { item\_choice } |
| 9 | {item\_quantity} | {} | { item\_choice } | { item\_choice, item\_quantity} |
| 10 | {} | {} | { item\_choice, item\_quantity} | { item\_choice, item\_quantity} |
| 11 | {} | {} | { item\_choice, item\_quantity} | { item\_choice, item\_quantity} |
| 12 | {} | {} | { item\_choice, item\_quantity} | { item\_choice, item\_quantity} |
| 13 | {} | {} | { item\_choice, item\_quantity} | { item\_choice, item\_quantity} |
| 14 | {} | {} | { item\_choice, item\_quantity} | { item\_choice, item\_quantity} |
| 14T | {} | {item\_choice} | { item\_choice, item\_quantity} | {item\_quantity} |
| 15 | {[item\_quantity->T]price} | {[item\_quantity->F]price} | {item\_quantity} | {item\_quantity, price} |
| 16 | {} | {} | {item\_quantity, price} | {item\_quantity, price} |
| 17 | {} | {} | {item\_quantity, price} | {item\_quantity, price} |
| 18 | {} | {} | {item\_quantity, price} | {item\_quantity, price} |
| 19 | {} | {} | {item\_quantity, price} | {item\_quantity, price} |
| 20 | {} | {} | {item\_quantity, price} | {item\_quantity, price} |
| 21 | {} | {} | { item\_choice, item\_quantity} | { item\_choice, item\_quantity} |
| 22 | {} | {} | { item\_choice, item\_quantity} | { item\_choice, item\_quantity} |
| 22F | {} | {item\_quantity} | { item\_choice, item\_quantity} | { item\_choice} |
| 23 | {} | {} | { item\_choice, item\_quantity} | { item\_choice, item\_quantity} |
| 24 | {} | {} | { item\_choice, item\_quantity} | { item\_choice, item\_quantity} |
| 25 | {[item\_quantity->T]price} | {[item\_quantity->F]price} | { item\_choice} | { item\_choice} |
| 26 | {} | {} | { item\_choice} | { item\_choice} |
| 27 | {} | {} | { item\_choice, item\_quantity, price} | { item\_choice, item\_quantity, price} |

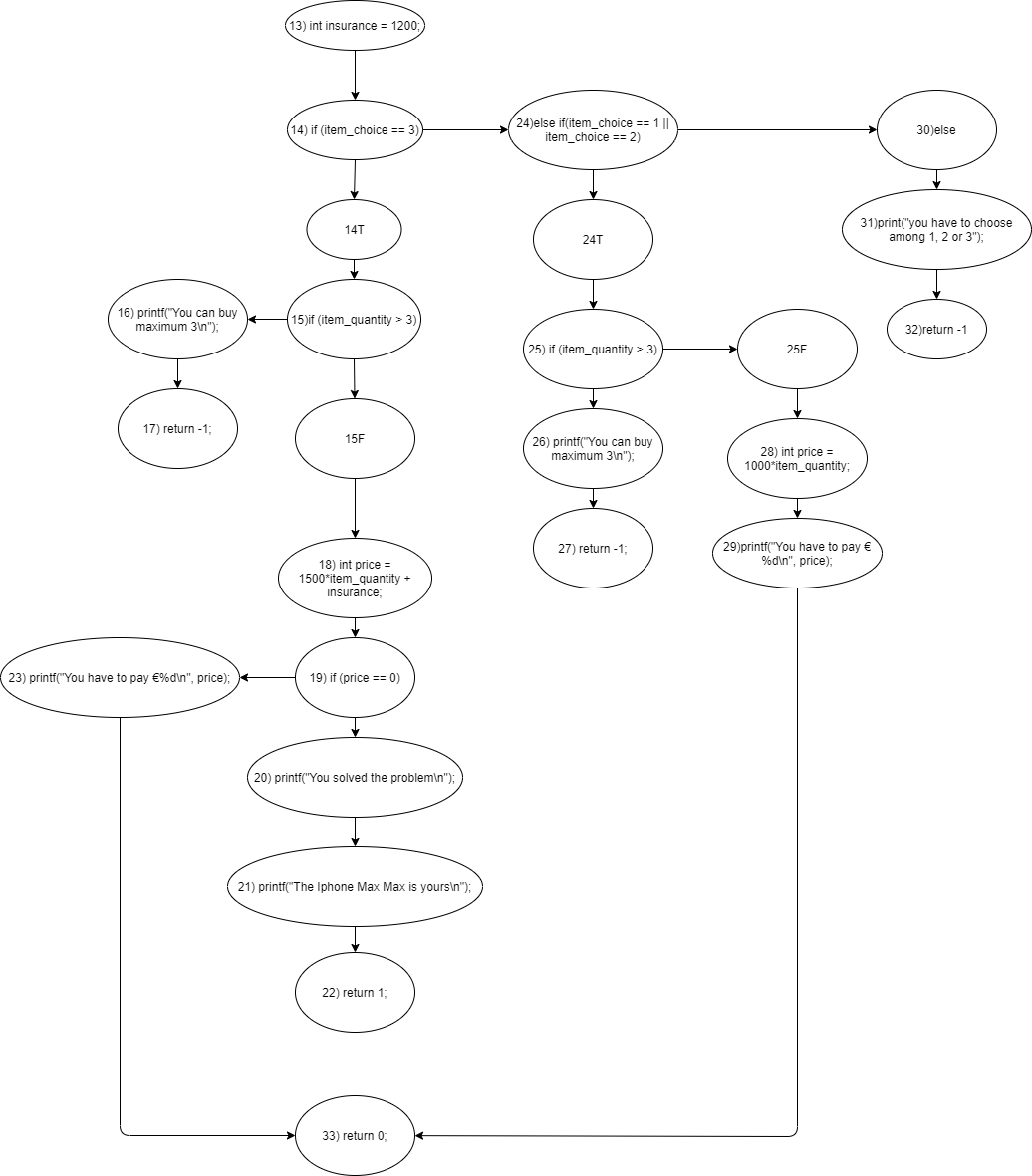
In the step 4 the table remains the same as step 3.

FIXED integer\_overflow:

To fix the program, I should check the input variable “item\_quantity” before the program makes the operation at line 15 for avoiding the integer overflow. So, I choose that an user can buy at most 3 IPhone 12 Pro Max Max. To sanitize all the script, I should check the input variable “item\_choice” so that the user can insert only 1, 2 or 3.

STEP 0: draw CFG.





STEP 1 and STEP 2: compute GEN and KILL, initialize IN and OUT.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | GEN | KILL | IN | OUT |
| 1 | {} | {} | {} | {} |
| 2 | {} | {} | {} | {} |
| 3 | {} | {} | {} | {} |
| 4 | {} | {} | {} | {} |
| 5 | {} | {} | {} | {} |
| 6 | {item\_choice} | {} | {} | {item\_choice} |
| 7 | {} | {} | {} | {} |
| 8 | {} | {} | {} | {} |
| 9 | {item\_quantity} | {} | {} | {item\_quantity} |
| 10 | {} | {} | {} | {} |
| 11 | {} | {} | {} | {} |
| 12 | {} | {} | {} | {} |
| 13 | {} | {} | {} | {} |
| 14 | {} | {} | {} | {} |
| 14T | {} | {item\_choice} | {} | {} |
| 15 | {} | {} | {} | {} |
| 15F | {} | {item\_quantity} | {} | {} |
| 16 | {} | {} | {} | {} |
| 17 | {} | {} | {} | {} |
| 18 | {[item\_quantity->T]price} | {[item\_quantity->F]price} | {} | {[item\_quantity->T]price} |
| 19 | {} | {} | {} | {} |
| 20 | {} | {} | {} | {} |
| 21 | {} | {} | {} | {} |
| 22 | {} | {} | {} | {} |
| 23 | {} | {} | {} | {} |
| 24 | {} | {} | {} | {} |
| 24T | {} | {item\_choice} | {} | {} |
| 25 | {} | {} | {} | {} |
| 25F | {} | {item\_quantity} | {} | {} |
| 26 | {} | {} | {} | {} |
| 27 | {} | {} | {} | {} |
| 28 | {[item\_quantity->T]price} | {[item\_quantity->F]price} | {} | {[item\_quantity->T]price} |
| 29 | {} | {} | {} | {} |
| 30 | {} | {} | {} | {} |
| 31 | {} | {} | {} | {} |
| 32 | {} | {} | {} | {} |
| 33 | {} | {} | {} | {} |

STEP 3: compute IN and OUT with transfer function.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | GEN | KILL | IN | OUT |
| 1 | {} | {} | {} | {} |
| 2 | {} | {} | {} | {} |
| 3 | {} | {} | {} | {} |
| 4 | {} | {} | {} | {} |
| 5 | {} | {} | {} | {} |
| 6 | {item\_choice} | {} | {} | {item\_choice} |
| 7 | {} | {} | {item\_choice} | {item\_choice} |
| 8 | {} | {} | {item\_choice} | {item\_choice} |
| 9 | {item\_quantity} | {} | {item\_choice} | {item\_choice, item\_quantity} |
| 10 | {} | {} | {item\_choice, item\_quantity} | {item\_choice, item\_quantity} |
| 11 | {} | {} | {item\_choice, item\_quantity} | {item\_choice, item\_quantity} |
| 12 | {} | {} | {item\_choice, item\_quantity} | {item\_choice, item\_quantity} |
| 13 | {} | {} | {item\_choice, item\_quantity} | {item\_choice, item\_quantity} |
| 14 | {} | {} | {item\_choice, item\_quantity} | {item\_choice, item\_quantity} |
| 14T | {} | {item\_choice} | {item\_choice, item\_quantity} | {item\_quantity} |
| 15 | {} | {} | {item\_quantity} | {item\_quantity} |
| 15F | {} | {item\_quantity} | {item\_quantity} | {} |
| 16 | {} | {} | {item\_quantity} | {item\_quantity} |
| 17 | {} | {} | {item\_quantity} | {item\_quantity} |
| 18 | {[item\_quantity->T]price} | {[item\_quantity->F]price} | {} | {} |
| 19 | {} | {} | {} | {} |
| 20 | {} | {} | {} | {} |
| 21 | {} | {} | {} | {} |
| 22 | {} | {} | {} | {} |
| 23 | {} | {} | {} | {} |
| 24 | {} | {} | {item\_choice, item\_quantity} | {item\_choice, item\_quantity} |
| 24T | {} | {item\_choice} | {item\_choice, item\_quantity} | {item\_quantity} |
| 25 | {} | {} | {item\_quantity} | {item\_quantity} |
| 25F | {} | {item\_quantity} | {{item\_quantity} | {} |
| 26 | {} | {} | {item\_quantity} | {item\_quantity} |
| 27 | {} | {} | {item\_quantity} | {item\_quantity} |
| 28 | {[item\_quantity->T]price} | {[item\_quantity->F]price} | {} | {} |
| 29 | {} | {} | {} | {} |
| 30 | {} | {} | {item\_choice, item\_quantity} | {item\_choice, item\_quantity} |
| 31 | {} | {} | {item\_choice, item\_quantity} | {item\_choice, item\_quantity} |
| 32 | {} | {} | {item\_choice, item\_quantity} | {item\_choice, item\_quantity} |
| 33 | {} | {} | {} | {} |

Now, as you can see, in line 18 the program makes a mathematical operation with untainted user input “item\_quantity”, thanks to the new check at line 15. Moreover, at line 23 the program does not print a tainted variable like before. So, now there is no more risk of integer overflow. Besides, I also check “item\_choice”.

STEP 4: iterates at least once to check the tainted variable.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | GEN | KILL | IN | OUT |
| 1 | {} | {} | {} | {} |
| 2 | {} | {} | {} | {} |
| 3 | {} | {} | {} | {} |
| 4 | {} | {} | {} | {} |
| 5 | {} | {} | {} | {} |
| 6 | {item\_choice} | {} | {} | {item\_choice} |
| 7 | {} | {} | {item\_choice} | {item\_choice} |
| 8 | {} | {} | {item\_choice} | {item\_choice} |
| 9 | {item\_quantity} | {} | {item\_choice} | {item\_choice, item\_quantity} |
| 10 | {} | {} | {item\_choice, item\_quantity} | {item\_choice, item\_quantity} |
| 11 | {} | {} | {item\_choice, item\_quantity} | {item\_choice, item\_quantity} |
| 12 | {} | {} | {item\_choice, item\_quantity} | {item\_choice, item\_quantity} |
| 13 | {} | {} | {item\_choice, item\_quantity} | {item\_choice, item\_quantity} |
| 14 | {} | {} | {item\_choice, item\_quantity} | {item\_choice, item\_quantity} |
| 14T | {} | {item\_choice} | {item\_choice, item\_quantity} | {item\_quantity} |
| 15 | {} | {} | {item\_quantity} | {item\_quantity} |
| 15F | {} | {item\_quantity} | {item\_quantity} | {} |
| 16 | {} | {} | {item\_quantity} | {item\_quantity} |
| 17 | {} | {} | {item\_quantity} | {item\_quantity} |
| 18 | {[item\_quantity->T]price} | {[item\_quantity->F]price} | {} | {} |
| 19 | {} | {} | {} | {} |
| 20 | {} | {} | {} | {} |
| 21 | {} | {} | {} | {} |
| 22 | {} | {} | {} | {} |
| 23 | {} | {} | {} | {} |
| 24 | {} | {} | {item\_choice, item\_quantity} | {item\_choice, item\_quantity} |
| 24T | {} | {item\_choice} | {item\_choice, item\_quantity} | {item\_quantity} |
| 25 | {} | {} | {item\_quantity} | {item\_quantity} |
| 25F | {} | {item\_quantity} | {{item\_quantity} | {} |
| 26 | {} | {} | {item\_quantity} | {item\_quantity} |
| 27 | {} | {} | {item\_quantity} | {item\_quantity} |
| 28 | {[item\_quantity->T]price} | {[item\_quantity->F]price} | {} | {} |
| 29 | {} | {} | {} | {} |
| 30 | {} | {} | {item\_choice, item\_quantity} | {item\_choice, item\_quantity} |
| 31 | {} | {} | {item\_choice, item\_quantity} | {item\_choice, item\_quantity} |
| 32 | {} | {} | {item\_choice, item\_quantity} | {item\_choice, item\_quantity} |
| 33 | {} | {} | {} | {} |

In the step 4 the table remains the same as step 3.