

PL/SQL Secure Coding



Important Terms

Exploit: Take advantage of a flaw or feature

SQL Injection: Change a sql statement so it executes code that was not intended. Think change the code execution path.

Hack: Anything can be hacked. Do something it was not intended to do or something you did not think it could do.

Spillage: Sensitive data has “spilled” outside it’s protected environment. It may not be compromised.

Leak: Sensitive data has spilled outside of it’s protected environment. It has been compromised.



Brain Hacking Demo

Anything can be hacked. Hacking is getting something to do what it was not intended to do or something you did not think it could do.

“Young man, success comes in can,
failure comes in can’t.”
Adm Grace Hopper to a
young Robert Lockard 1978.



PL/SQL Secure Coding



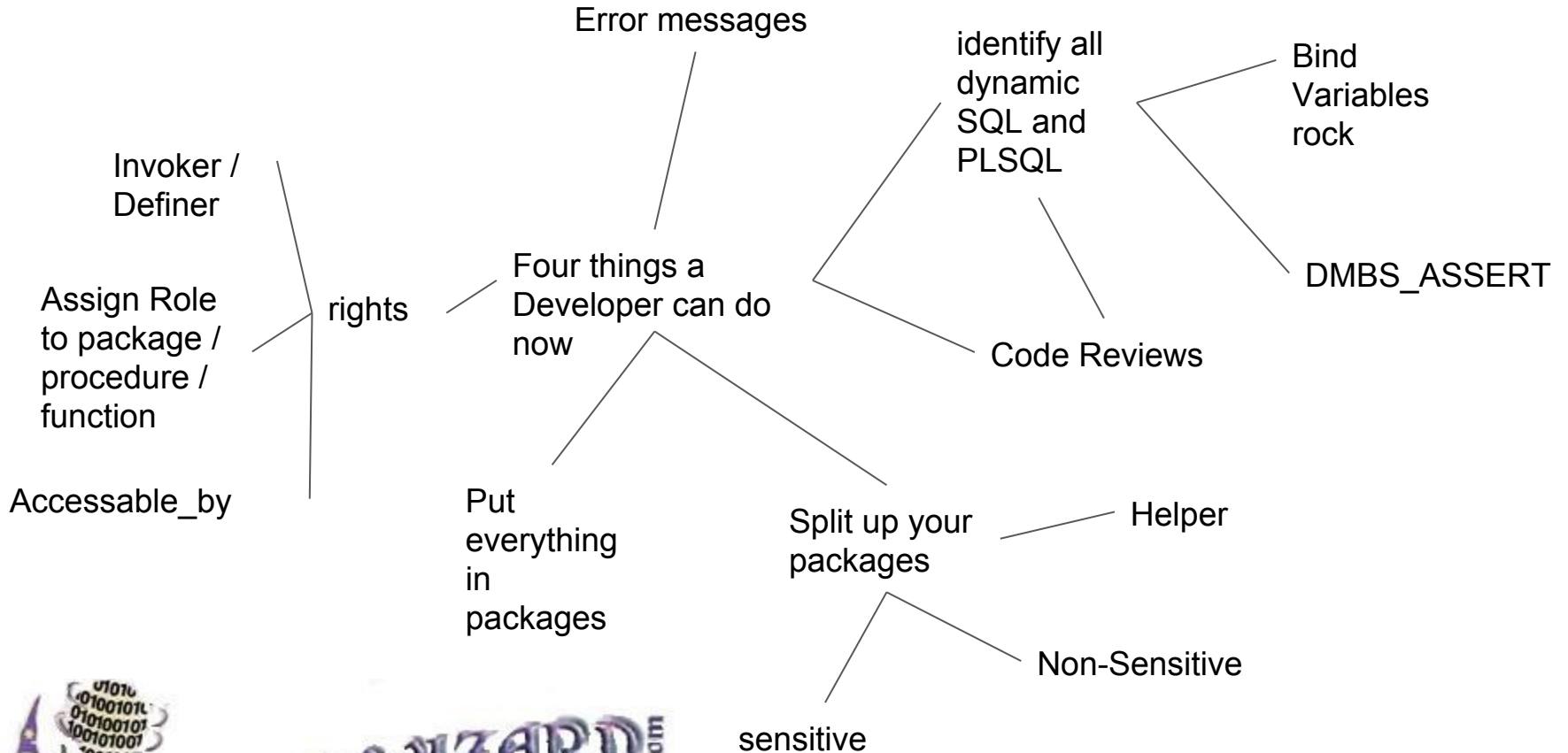
ORACLE
ACE

PL/SQL Secure Coding



4 things developers can do now to improve security... in process

Oooopsy, I lied, there are more than 4 things. :-)



Error Messages

```
oracle@localhost:~/demo/sql_injection

we are going to create a generic help desk utility
this will require a few objects and held in two
schemas.
1) the schema helpdesk will hold the errors table
2) the schema helpdesk_api will hold an api that inserts
and select data from the api table
3)

*/

drop user helpdesk cascade;
drop user helpdesk_api cascade;

create user helpdesk identified by My#92SecretPassword;
create user helpdesk_api identified by My#92SecretPassword;
create sequence helpdesk_api.errors_seq;

ALTER USER HELPDESK QUOTA UNLIMITED ON USERS;

create table helpdesk.errors (id          number primary key,
                             unitname    varchar2(65),
                             username    varchar2(65),
                             linenumber   number,
                             errorcode    varchar2(11),
                             errormsg     varchar2(225),
                             createdate   timestamp);

grant select, insert on helpdesk.errors to helpdesk_api;
grant select on helpdesk.errors_seq to helpdesk_api;
```



Error Messages

```
oracle@localhost:~/demo/sql_injection
create or replace package helpdesk_api.tab_api
AUTHID DEFINER AS

    type t_errors is table of helpdesk.errors%rowtype
        index by pls_integer;

    function ins_error(pUnitName      IN VARCHAR2,
                      pUserName      IN VARCHAR2,
                      pLineNumber    IN NUMBER,
                      pErrorCode     IN VARCHAR2,
                      pErrorMsg      IN VARCHAR2,
                      PCreateDate    IN TIMESTAMP DEFAULT CURRENT_TIMESTAMP)
RETURN NUMBER;

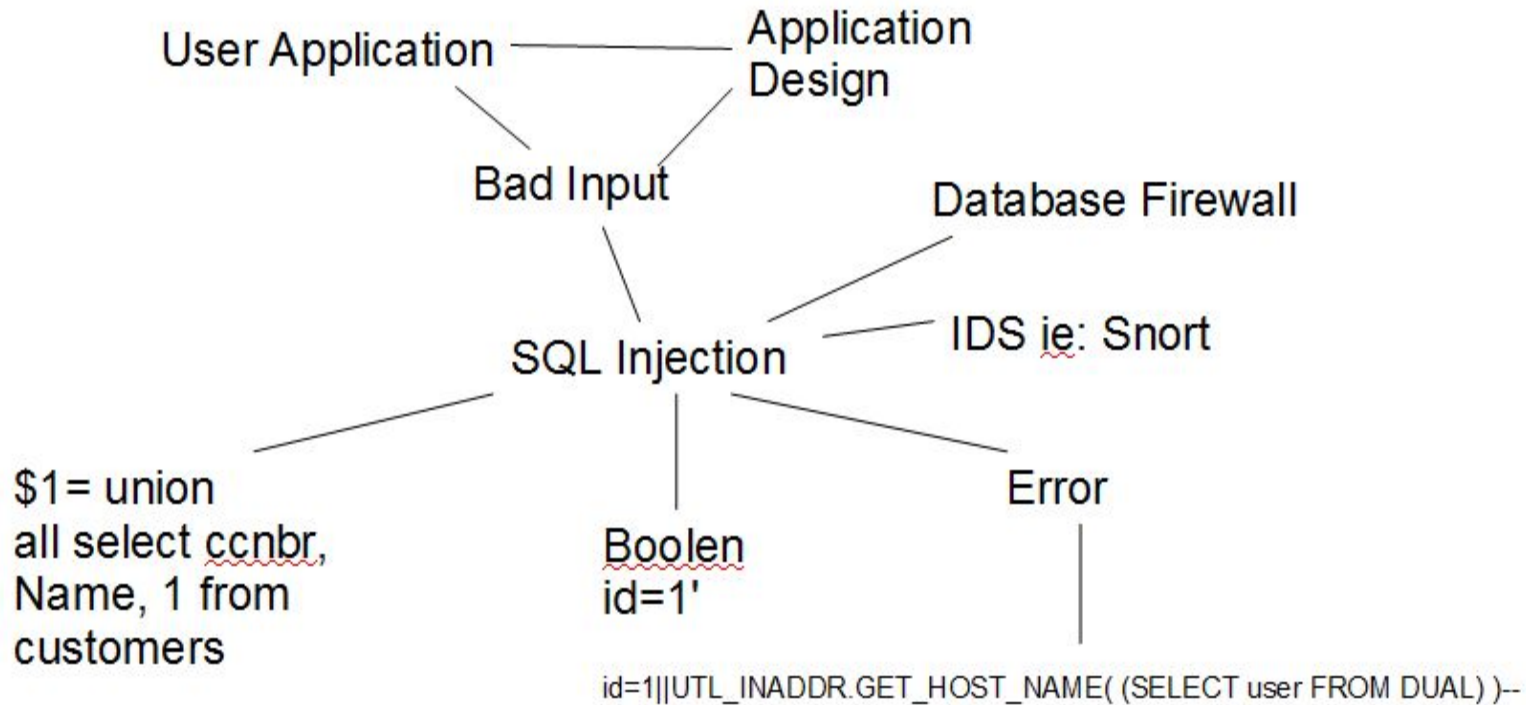
    function sel_error(pId           IN NUMBER) RETURN t_errors;

end;
/
```

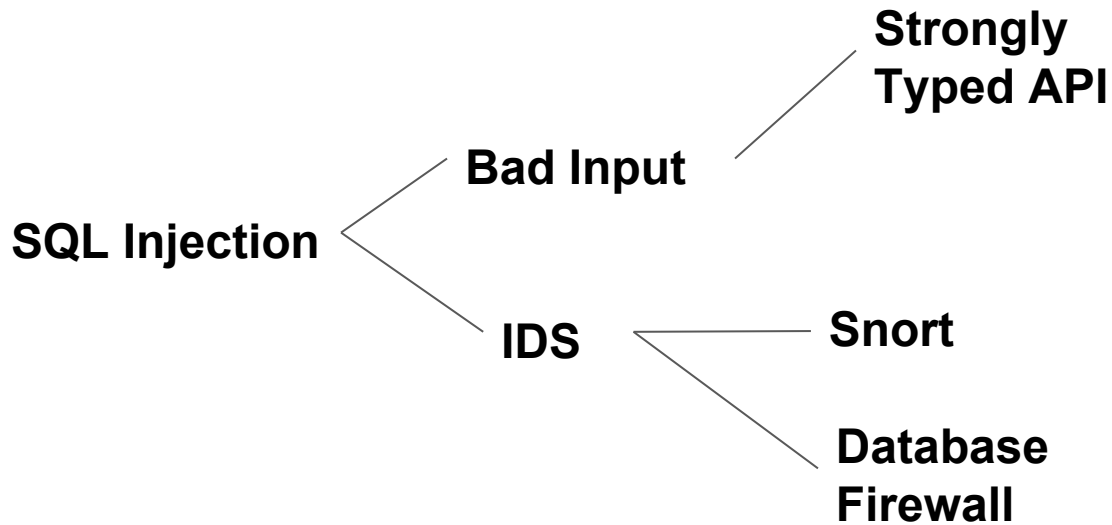
31, 0-1 32%



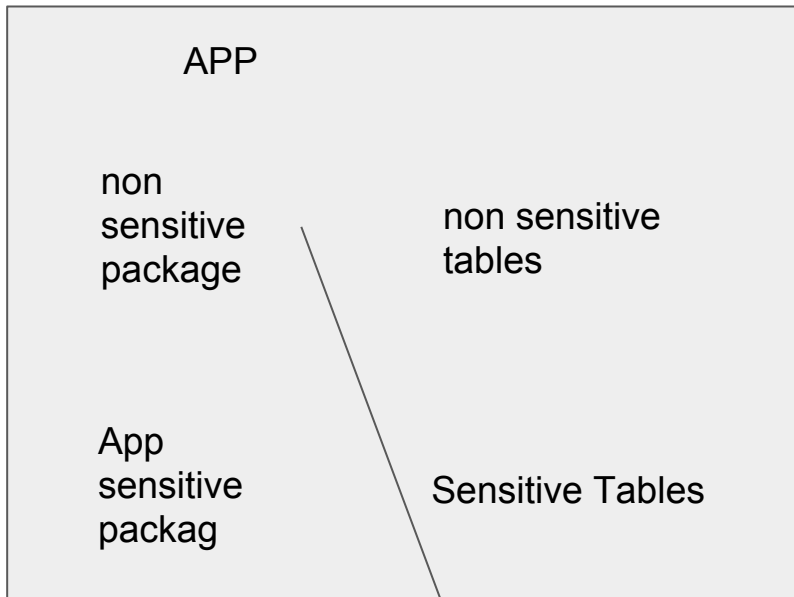
SQL Injection



SQL Injection



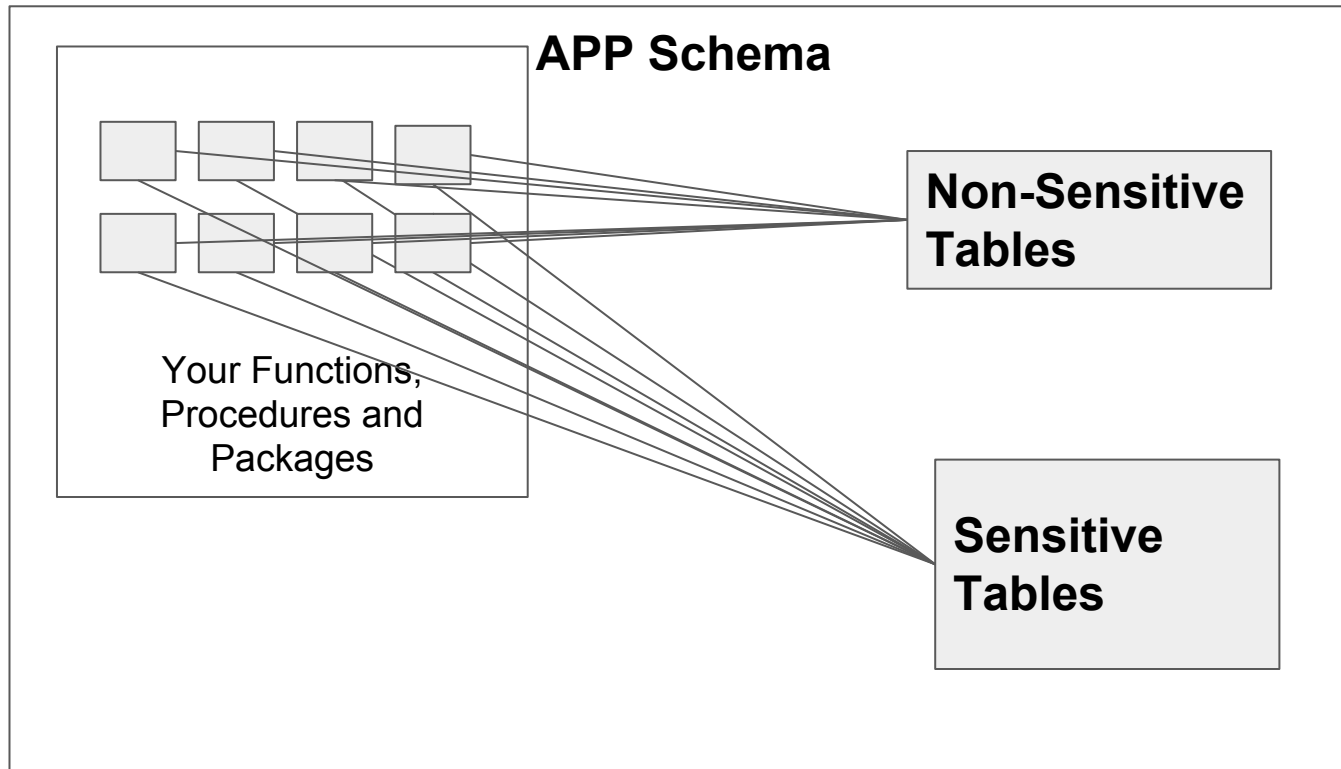
Separate your data from your code



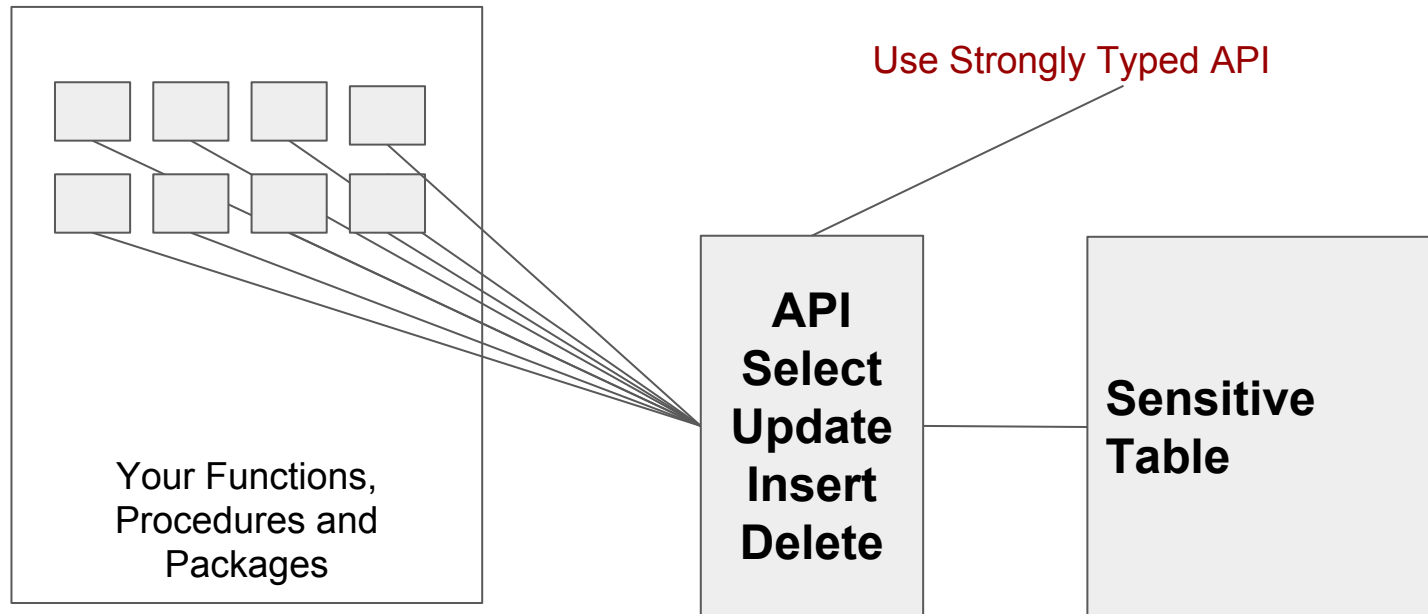
SQL INJECTION
BUG



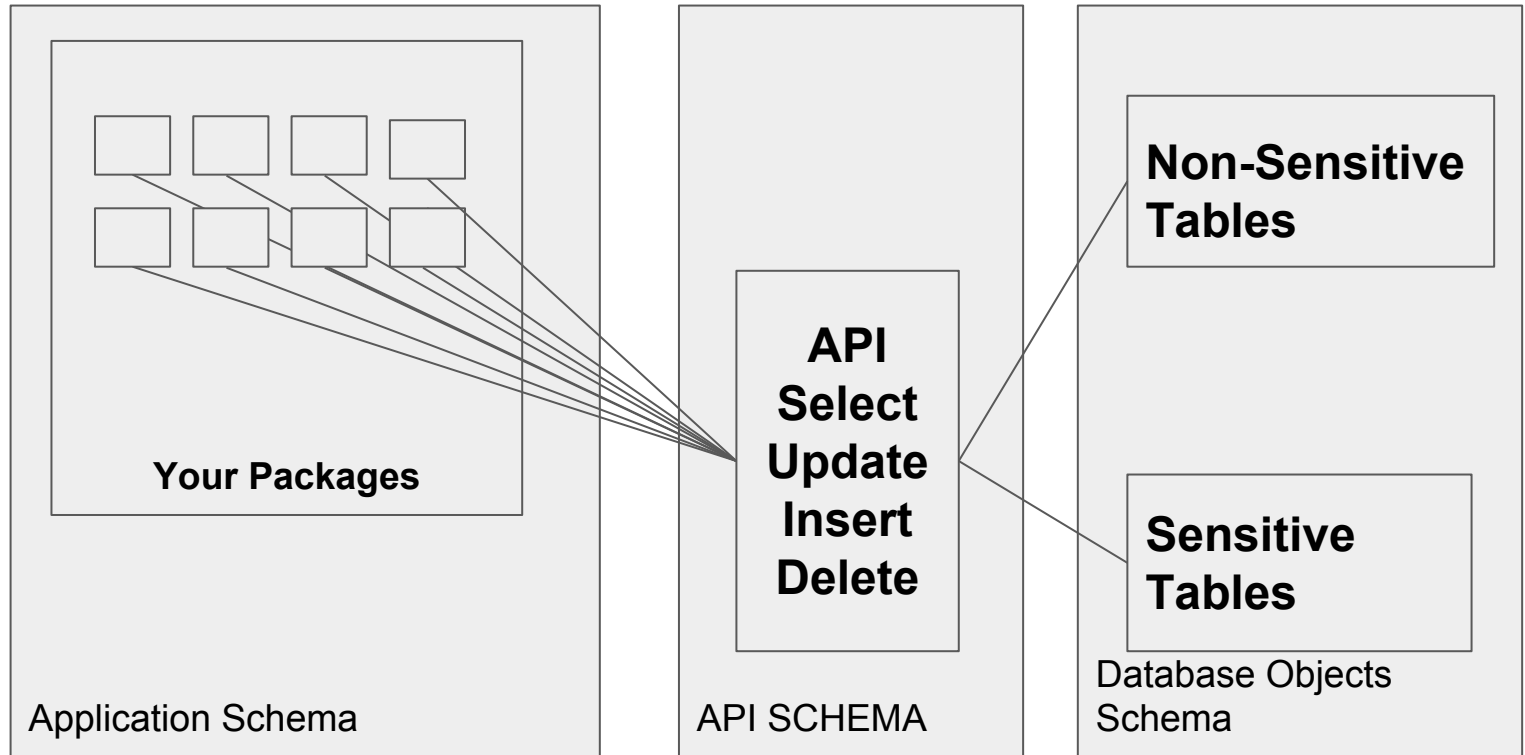
Limit the number of ways to get to your sensitive data. Trusted Path



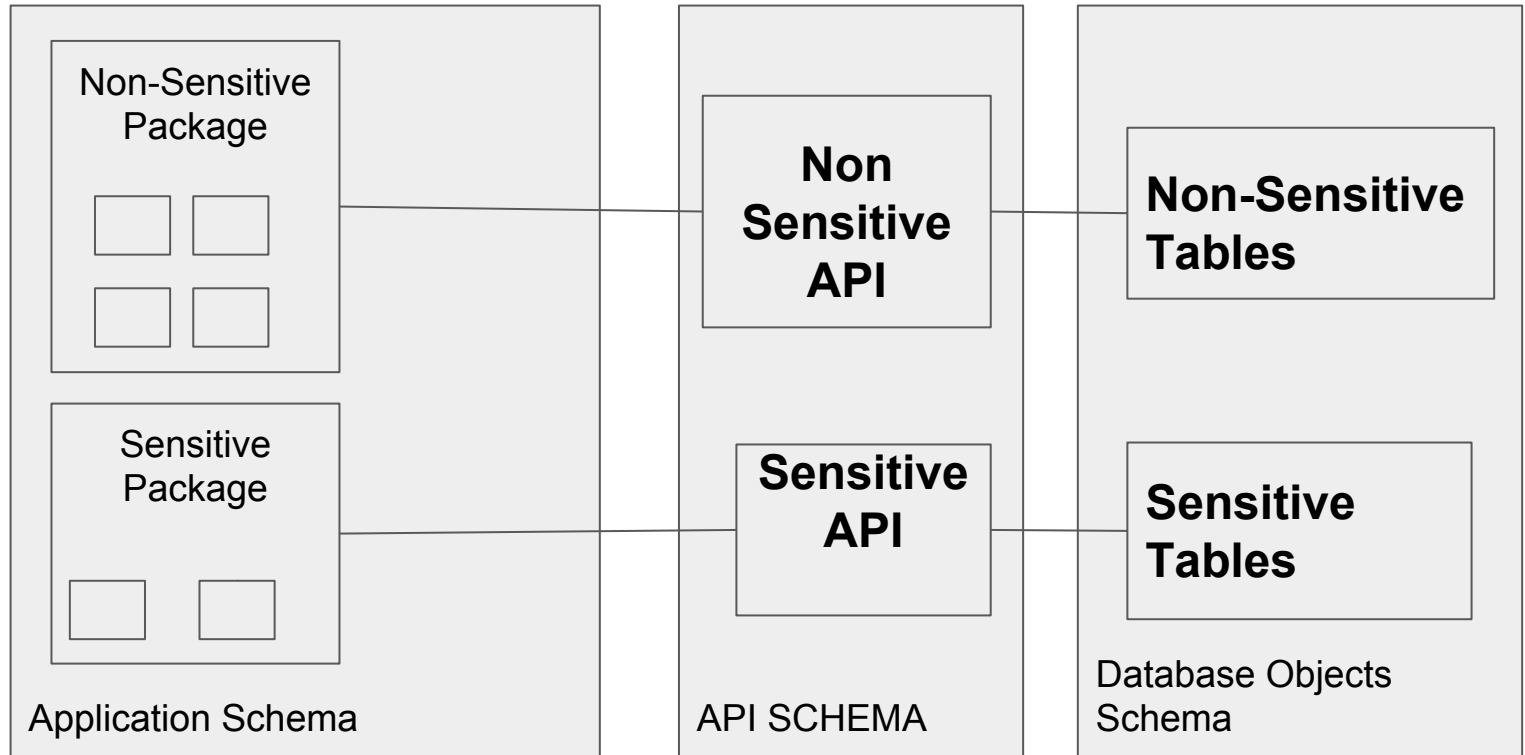
Limit the number of ways to get to your sensitive data. Trusted Path



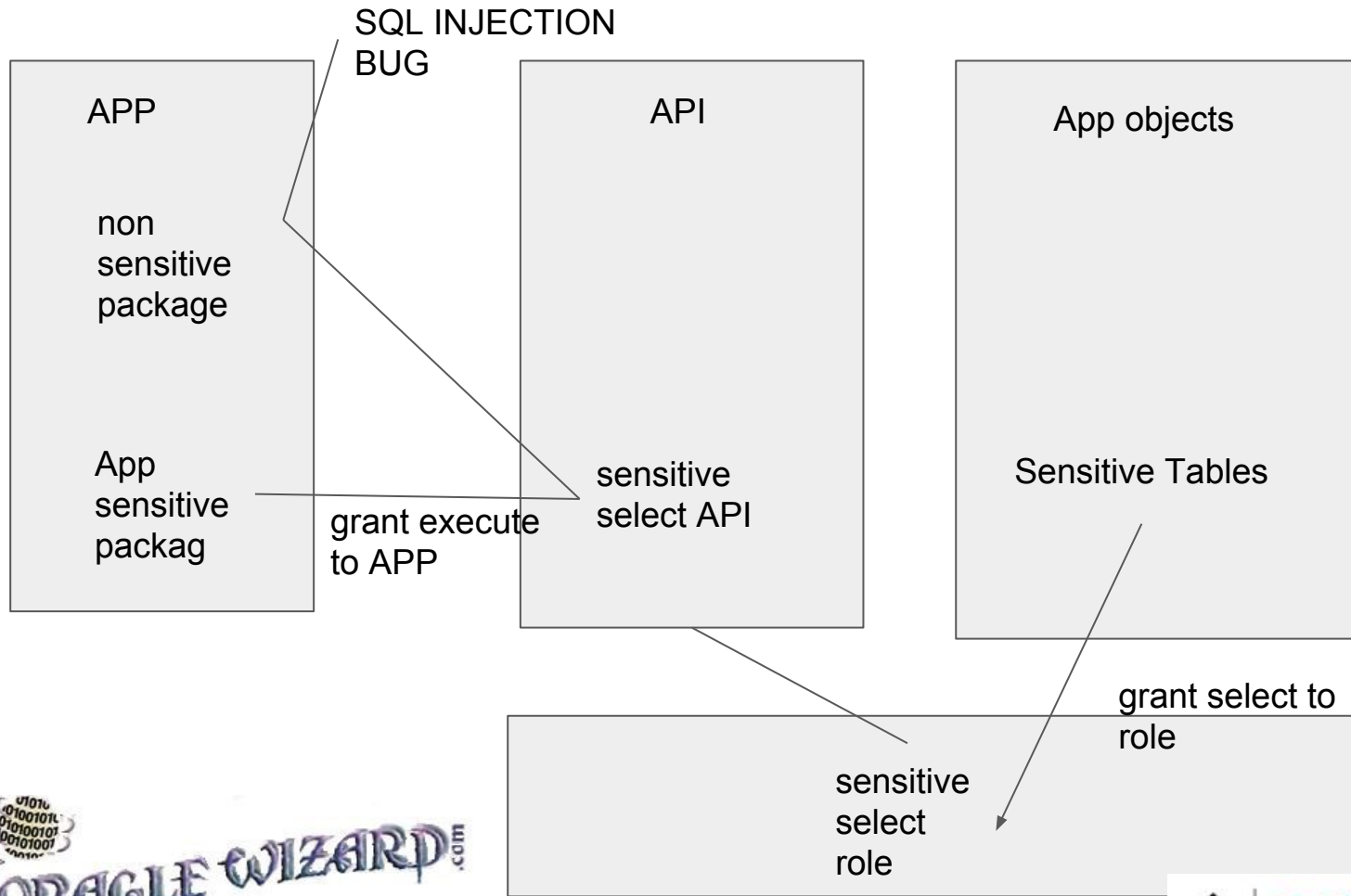
Limit the number of ways to get to your sensitive data. Trusted Path



Limit the number of ways to get to your sensitive data. Trusted Path

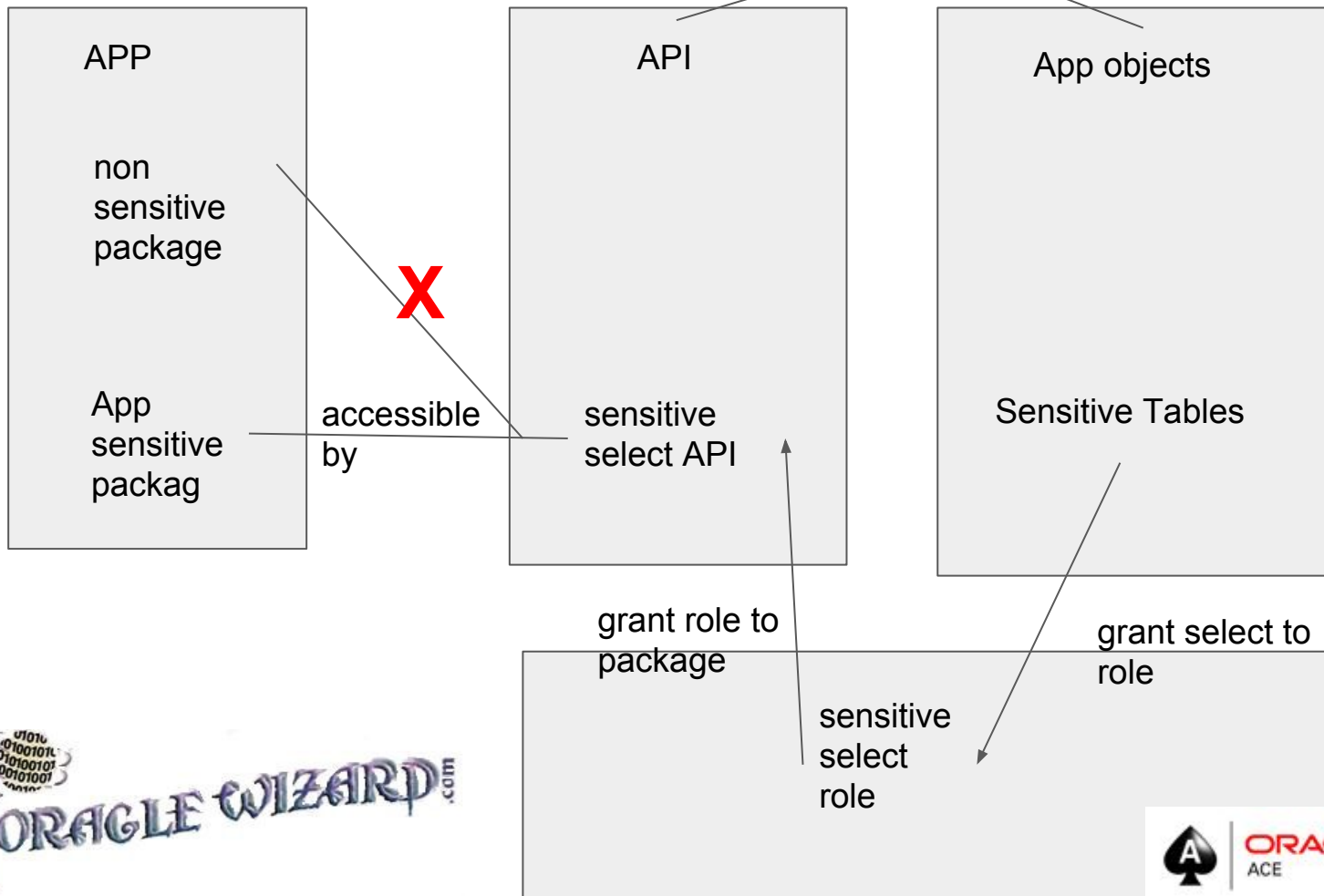


Separate your data from your code



Separate your data from your code

Does not have
connect privs



Resources

<http://csrc.nist.gov/publications/nistpubs/800-122/sp800-122.pdf>

https://docs.google.com/spreadsheets/d/1Dvl_CbX2b0NGFzE2gVLQb1-Nc6litfpGtoTB9iytWfM/edit?usp=sharing

Youtube search for “May 2016 CodeTalk: Securing PL/SQL Code From Attacks

Google Search “sql injection proof pl/sql”

“



Contact Information

Robert P. Lockard
Oraclewizard, Inc.
Hubzone Certified
Small Veteran Owned Business
Glen Burnie, MD
USA

email: security@oraclewizard.com
twitter: @YourNavionPilot
blog: www.oraclewizard.com
youtube: www.youtube.com/user/n4281k

