Securing PostgreSQL
From External Attack

BRUCE MOMJIAN

EnterpriseDB®
The Enterprise PostgreSQL Company

January, 2012

Database systems are rich with attack vectors to exploit. This presentation explores the many potential PostgreSQL external vulnerabilities and shows how they can be secured.

Creative Commons Attribution License   *http://momjian.us/presentations*

External Attack Vectors

- 'Trust' security
- Passwords / authentication theft
- Network snooping
- Network pass-through spoofing
- Server / backup theft
- Administrator access

Internal Attack Vectors
(Not Covered)

- Database object permissions
- SQL injection attacks
- Application vulnerability
- Operating system compromise

http://www.my-time-machines.net/mosler_34.htm

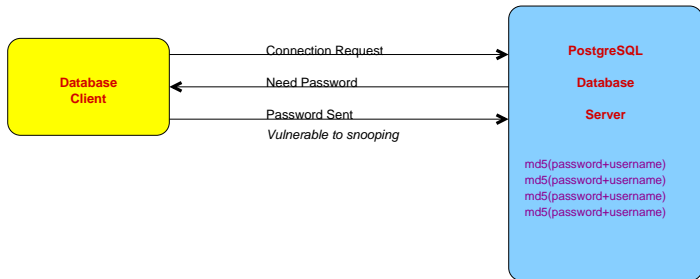# Avoid 'Trust' Security

```
# TYPE  DATABASE      USER          CIDR-ADDRESS          METHOD
# "local" is for Unix domain socket connections only
local   all           all                                 trust
# IPv4 local connections:
host    all           all           127.0.0.1/32          trust
# IPv6 local connections:
host    all           all           ::1/128               trust
```

Solution: Use the initdb *-A* flag, i.e., you don't want to see this:

WARNING: enabling "trust" authentication for local connections
You can change this by editing pg_hba.conf or using the -A option the
next time you run initdb.

Password Snooping

Database Client

Connection Request

Need Password

Password Sent
Vulnerable to snooping

PostgreSQL

Database

Server

md5(password+username)
md5(password+username)
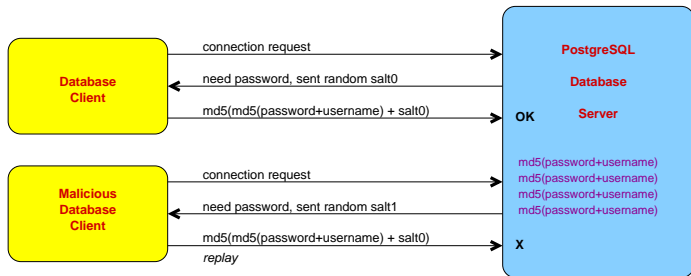md5(password+username)
md5(password+username)

Using 'username' in the MD5
string prevents the same password
used by different users from
appearing the same. It also adds
some randomness to the md5
checksums.

MD5 Authentication
Prevents Password Snooping
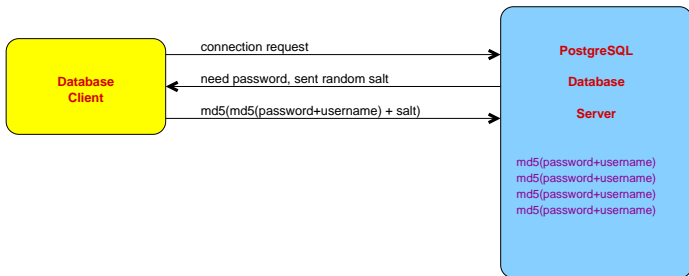
MD5 Authentication
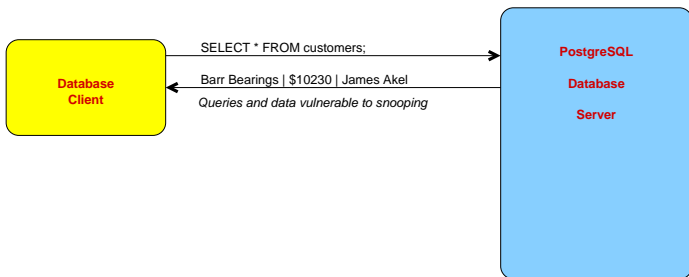# Prevents Password Replay

salt is a random four-byte integer so millions of connection attempts might allow the reuse of an old authentication reply.

Password Attacks

- Weak passwords
- Reuse of old passwords
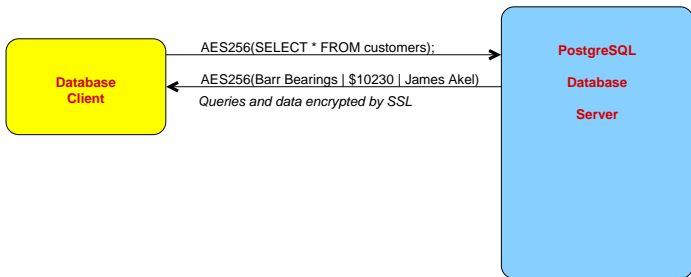- Brute-Force password attacks

None of these vulnerabilities is prevented by Postgres directly, but external authentication methods, like LDAP, PAM, and SSPI, can prevent them.

Queries and Data Still
Vulnerable to Network Snooping

Database
Client

SELECT * FROM customers;

Barr Bearings | $10230 | James Akel

Queries and data vulnerable to snooping

PostgreSQL

Database

Server

Password changes are also vulnerable to snooping.

SSL Prevents Snooping
By Encrypting Queries and Data

Database
Client

AES256(SELECT * FROM customers);

AES256(Barr Bearings | $10230 | James Akel)

Queries and data encrypted by SSL

PostgreSQL

Database

Server

Preventing Spoofing

http://redwing.hutman.net/~mreed/warriorshtm/impostor.htm

Localhost Spoofing
While the Database Server Is Down

Fake PostgreSQL

Database

Server

x

Records passwords for later
use with the real server

Connection Request

Need Plain Password

Password Sent

Database
Client
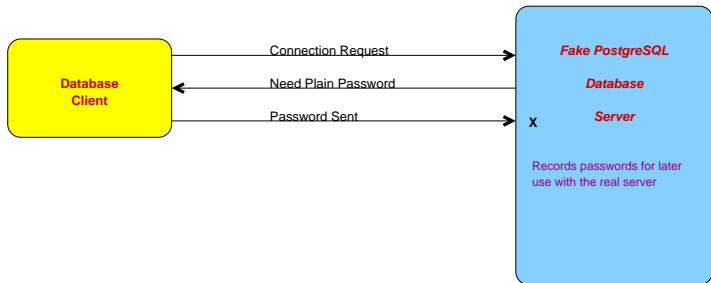
Uses a fake socket or binds to
port 5432 while the real server
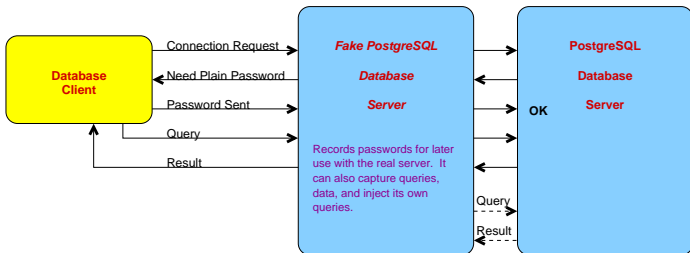is down. (/tmp is world-writable
and 5432 is not a root-only port.)

The server controls the choice of 'password' instead of 'md5'.

Network Spoofing
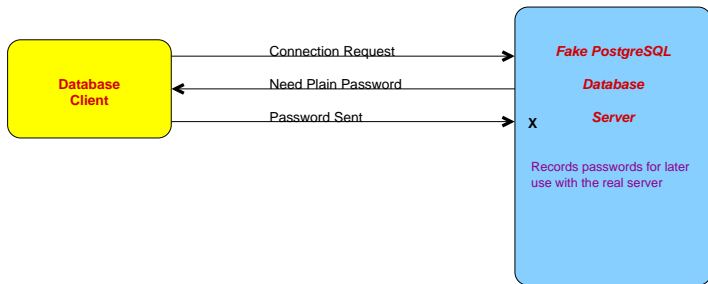
Database
Client

Connection Request

Need Plain Password

Password Sent

Fake PostgreSQL

Database

Server
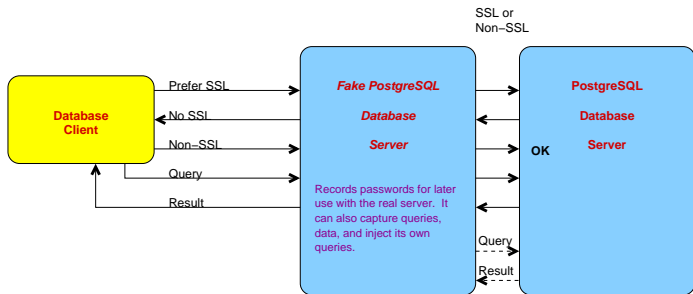
Records passwords for later
use with the real server

Without SSL 'root' certificates
there is no way to know if the
server you are connecting
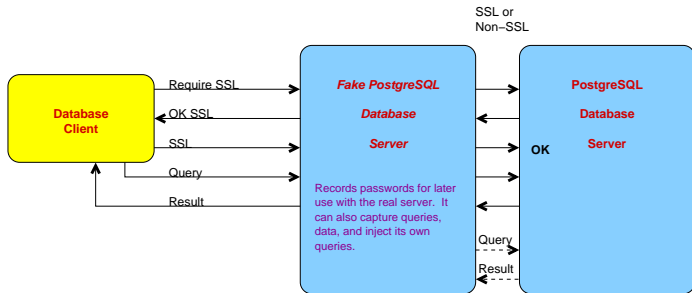to is a legitimate server.

Network Spoofing Pass-Through

Database Client

Connection Request
Need Plain Password
Password Sent
Query
Result

Fake PostgreSQL

Database

Server

Records passwords for later use with the real server. It can also capture queries, data, and inject its own queries.

Query
Result

OK

PostgreSQL

Database

Server

Without SSL 'root' certificates there is no way to know if the server you are connecting to is a legitimate server.

SSL 'Prefer' Is Not Secure

Database Client

Prefer SSL
No SSL
Non−SSL
Query
Result

SSL or
Non−SSL

Fake PostgreSQL

Database

Server

Records passwords for later
use with the real server. It
can also capture queries,
data, and inject its own
queries.

Query
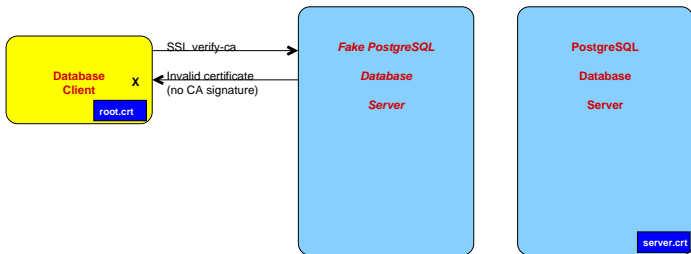Result

OK

PostgreSQL

Database

Server

Without SSL 'root' certificates
there is no way to know if the
server you are connecting
to is a legitimate server.

SSL 'Require' Is Not Secure From Spoofing

SSL or
Non–SSL

Database Client

Require SSL →
← OK SSL
SSL →
Query →
Result

Fake PostgreSQL
Database
Server

Records passwords for later use with the real server. It can also capture queries, data, and inject its own queries.

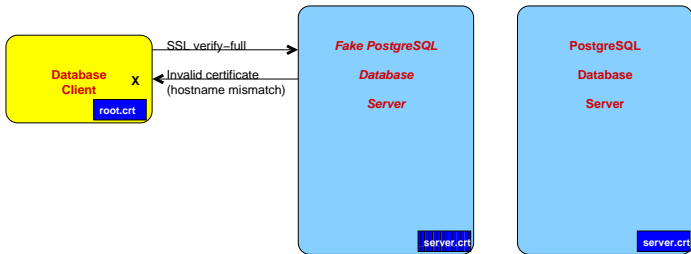**PostgreSQL**
Database
Server

OK

Query
Result

Without SSL 'root' certificates there is no way to know if the server you are connecting to is a legitimate server.

SSL 'Verify-CA' Is Secure From Spoofing

Database Client
x
root.crt

SSL verify-ca →
← Invalid certificate (no CA signature)

Fake PostgreSQL Database Server

PostgreSQL Database Server
server.crt

SSL 'Verify-full' Is Secure
Even From Some Certificate Thefts

Database Client

root.crt

SSL verify–full

Invalid certificate (hostname mismatch)

Fake PostgreSQL
Database
Server

server.crt

PostgreSQL
Database
Server

server.crt

Certificate stolen from a CA–trusted computer, but not the database server.

Data Encryption
To Avoid Data Theft

http://jproc.ca/crypto/enigma.html
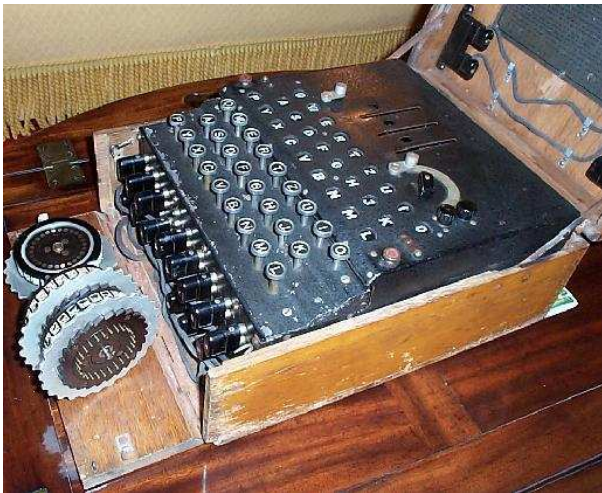
Disk Volume Encryption

http://www.pclaunches.com/
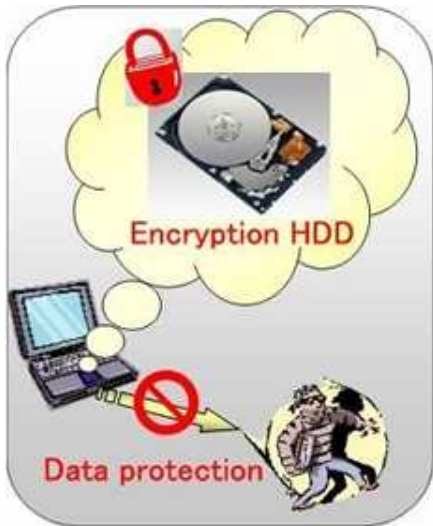
Column Encryption

```
   id   |        name        |         credit_card_number
--------+--------------------+-----------------------------
 428914 | Piller Plaster Co. | \xc30d04070302254dc045353f28
                             ; 456cd241013e2d421e198f3320e8
                             ; 41a7e4f751ebd9e2938cb6932390
                             ; 5c339c02b5a8580663d6249eb24f
                             ; 192e226c1647dc02536eb6a79a65
                             ; 3f3ed455ffc5726ca2b67430d5
```
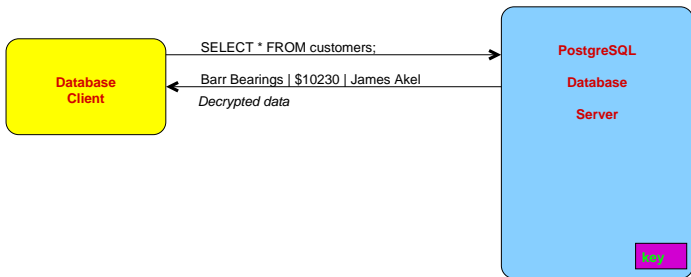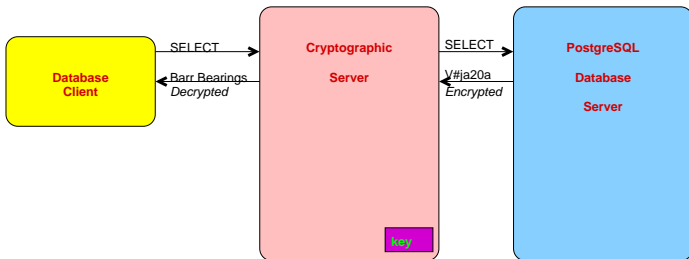
Encryption methods are decryptable (e.g. AES), while hashes are one-way (e.g. MD5). A one-way hash is best for data like passwords that only need to be checked for a match, rather than decrypted.
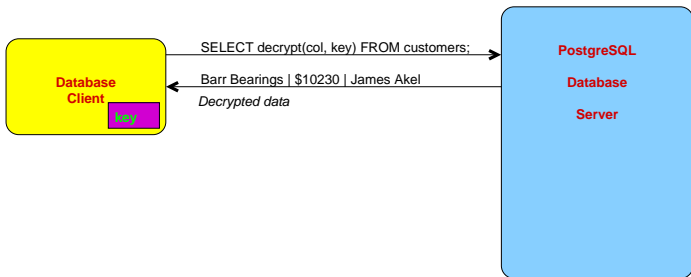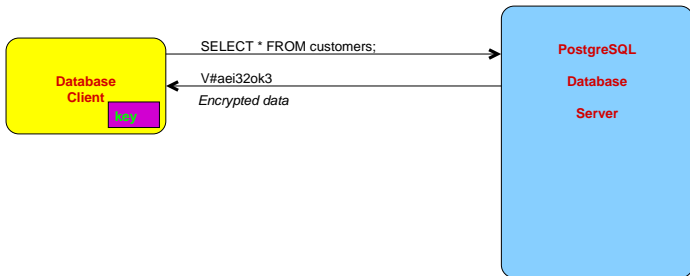
Where to Store the Key?
On the Server

SELECT * FROM customers;

Database Client

Barr Bearings | $10230 | James Akel
Decrypted data

PostgreSQL

Database

Server

key

Store the Key on an Intermediate Server

Store the Key on the Client and Encrypt/Decrypt on the Server

SELECT decrypt(col, key) FROM customers;

Barr Bearings | $10230 | James Akel
Decrypted data

Database Client

key

PostgreSQL Database Server

Encrypt/Decrypt on the Client

SELECT * FROM customers;

Database Client
key

V#aei32ok3
Encrypted data

PostgreSQL
Database
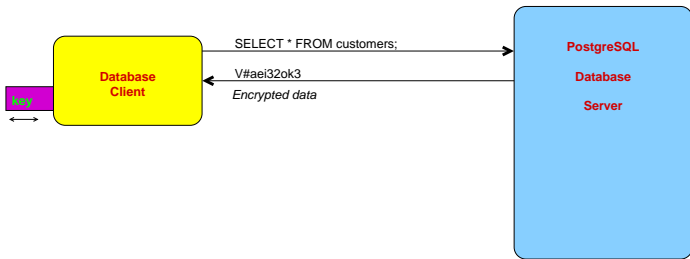**Server**

This prevents server administrators from viewing sensitive data.

Store the Key on a
Client Hardware Token

This prevents problems caused by client hardware theft.

Conclusion

Todd Ehlers, Flickr

http://momjian.us/presentations