Redpill Linpro

Replication & Database Security

Char(10)
Oxford, United Kingdom

Magnus Hagander
Redpill Linpro AB

Redpill Linpro

Uh. Security?

- Isn't this about:
  - **C**lustering
  - **H**igh **A**vailability
  - **R**eplication

Redpill Linpro

Uh, Security?

- Even more important in a distributed world
- Many different users, hosts and networks involved
- What do you *relly* know about «your» cloud?

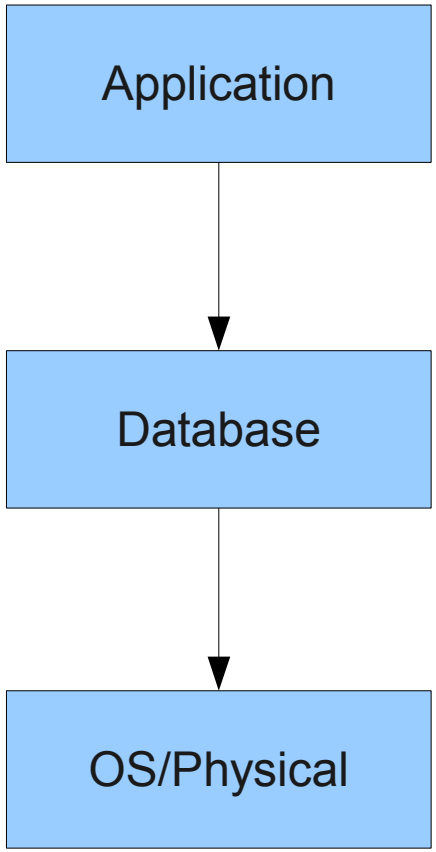
Redpill Linpro

Security with PostgreSQL replication

- Applies differently to different systems
- «Slony-style»
  - Includes Londiste, Bucardo etc
- «PITR-style»
  - Warm/Hot standby, streaming repl, etc

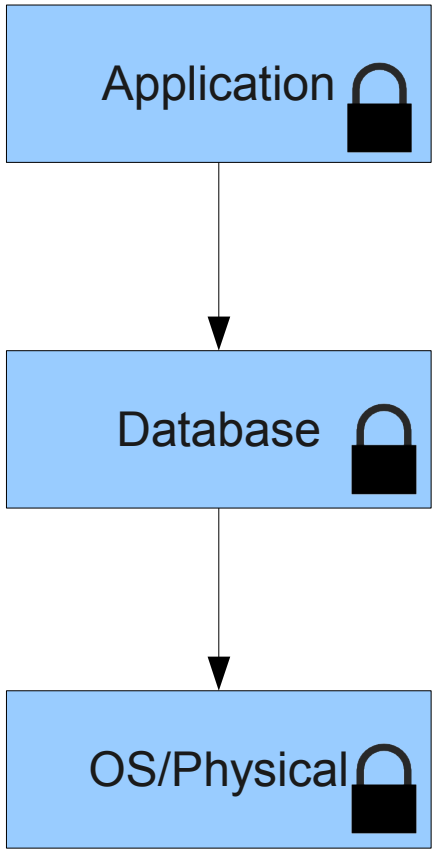
Redpill Linpro

Steps to security



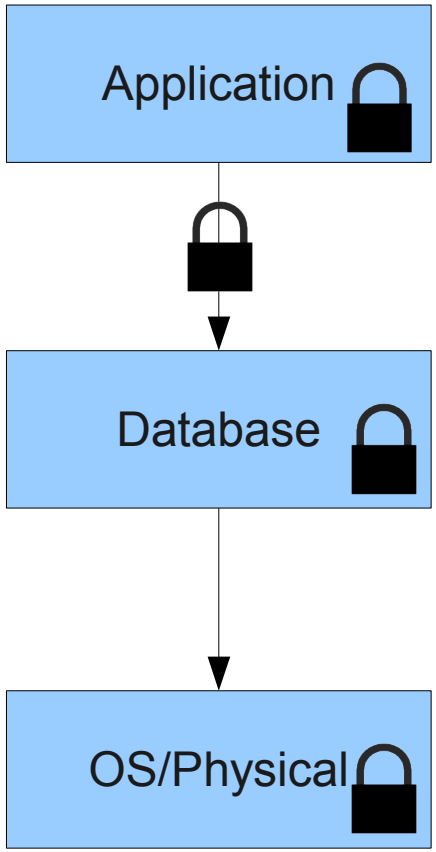
Redpill Linpro

Steps to security



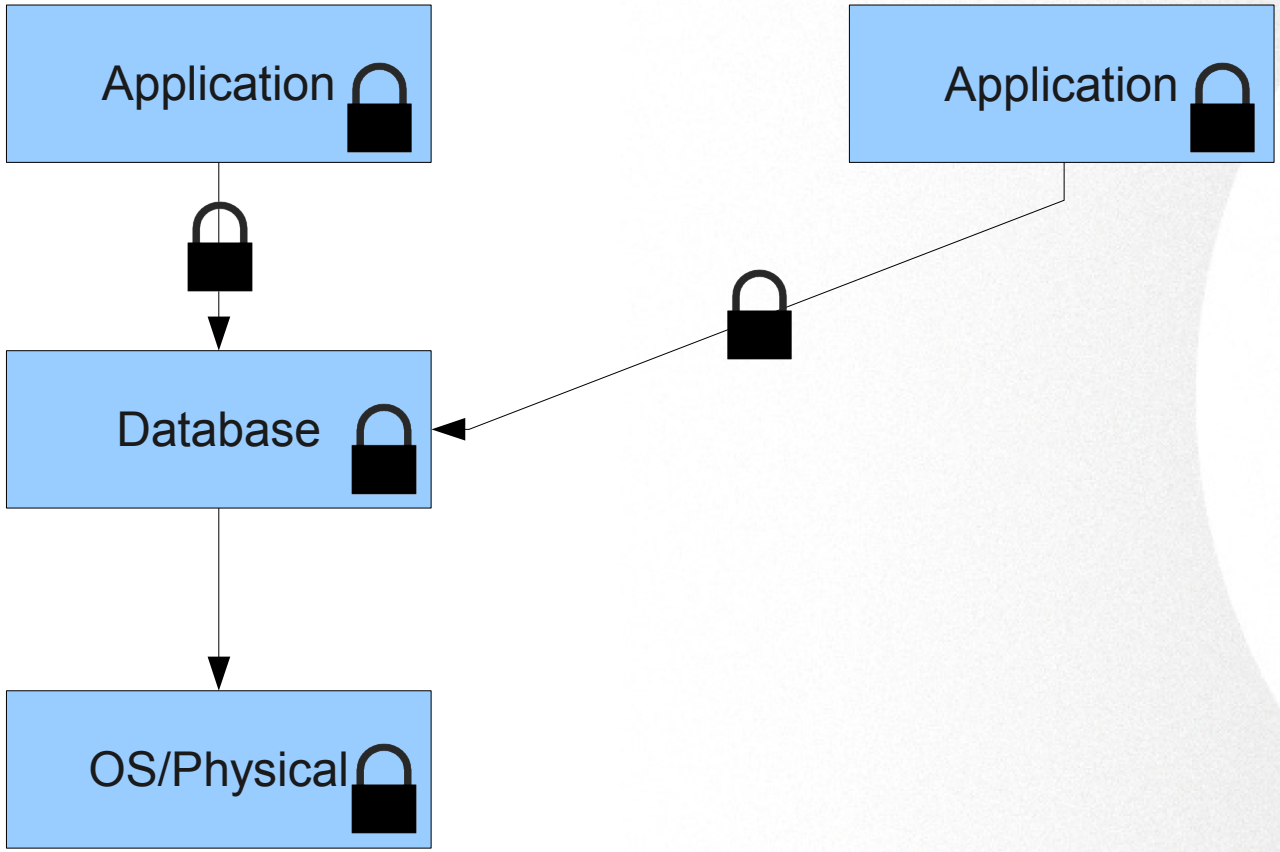
Redpill Linpro

Steps to security



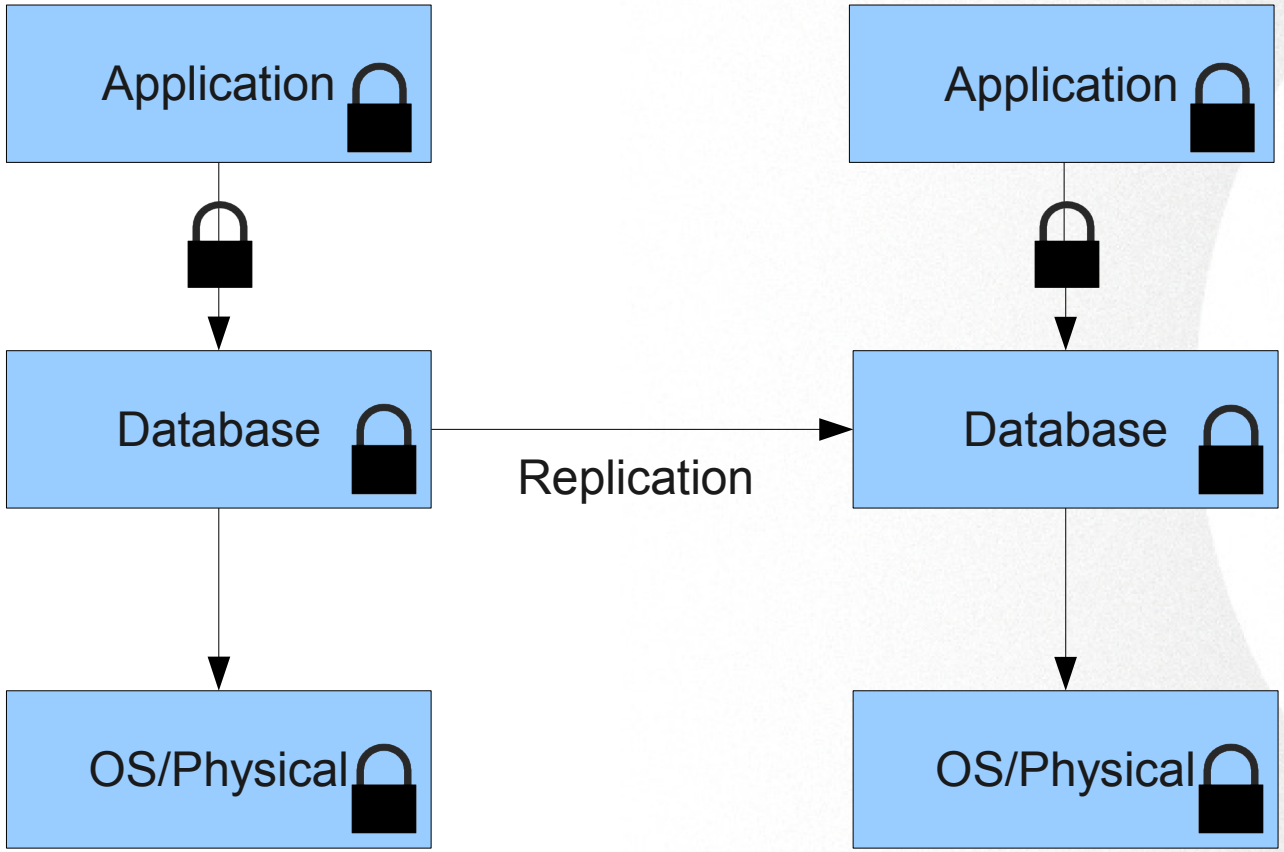
Redpill Linpro

Then we add distribution



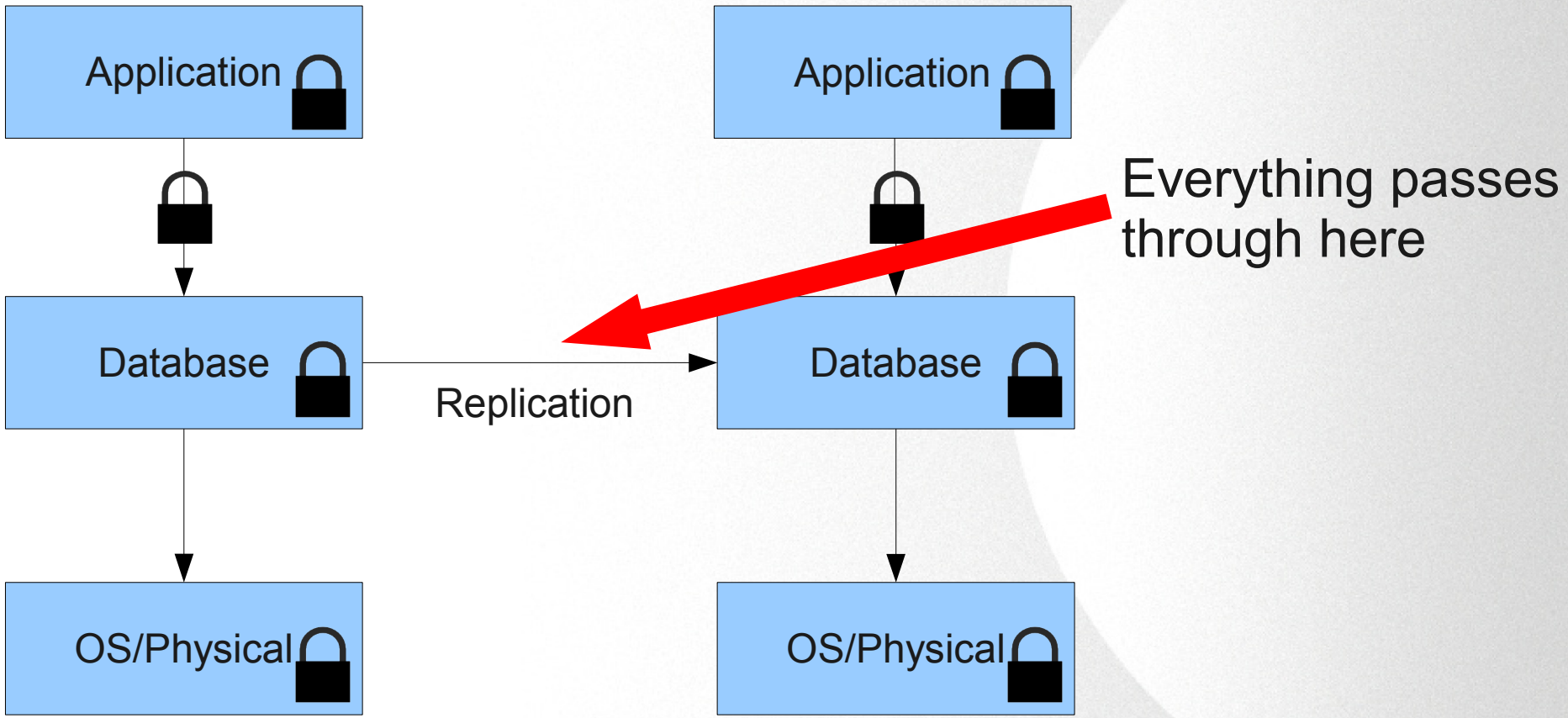
Redpill Linpro

Then we add distribution



Redpill Linpro

Keys to the kingdom



Redpill Linpro

Over the replication channel

- Data is unencrypted (usually)
- Data is unfiltered
- Data is *assumed valid*
- And yet...
  - Reading the data may not be the biggest problem

Redpill Linpro

Slony-style systems

- Requires high-priv account
  - Or if restricted, still full data access
  - Read and write
- In both directions
- *Not* restricted to replication

Redpill Linpro

Slony-style systems

- Probably most common:
  - User: slony
  - Password: slony
- Second most common?
  - User: slony
  - Password: <none>

Redpill Linpro

So what can we do?

- Use a *strong* password for replication user
  - There are no manual logins after all
  - Don't even consider «trust»
  - Use certificates!
- Use *different* replication users for different databases/parts
  - Restrict damage

Redpill Linpro

Utilize pg_hba.conf

- Allow replication user *only* from replication partners
- Allow *only replication user* from replication partners

Redpill Linpro

But....

- My replication connection is on a secure network!

Redpill Linpro

But....

- My replication connection is on a secure network!
- No, it's not. Really.
  - Far too valuable to be «assumed secure»

Redpill Linpro

So secure the connection

- Enable SSL
- *Require* SSL
  - (on both client and server)
- Overhead actually not so bad
  - Slony uses persistent connections
- That's the easy part...

Redpill Linpro

Configure (limited) PKI

- Set up PKI
  - Just for replication if not used for app
- Hand out both client and server certificates
- *Validate* certificates
  - libpq: sslmode=validate-[ca|full]
  - pg_hba.conf: clientcert=1

Redpill Linpro

Warm/Hot Standby

- Based on copying files
- Files are *not* signed/verified
- Transport security is handled outside

Redpill Linpro

Transfer user

- Always use a separate user
- *Don't* use the PostgreSQL service user

Redpill Linpro

Secure transfer

- rsync+ssh
  - Security through SSH
  - Integrity/atomicity through rsync
- Use SSHv2 (but you knew that)
- Transfer/verify server-side SSH key!
- Use StrictHostKeyChecking=yes

Redpill Linpro

Secure transfer

- Use ~/.ssh/authorized_keys
- *Restrict command access*
  - *Only* allow rsync access

```
command="rsync --server --sender -vlogDtpre.i . wal/",
no-port-forwarding,no-agent-forwarding,no-X11-
forwarding,no-pty ssh-rsa (...) replication@local
```

Redpill Linpro

Streaming replication in 9.0

- Uses libpq for connections
- User must be *superuser*
- «Fake» database *replication*
- So secure the same way as Slony

Redpill Linpro

Monitor!

- IDS
- «tcpdump |grep SELECT»
- Log connections/disconnections

Redpill Linpro

Audit!

- If you're not auditing pg_hba.conf today...
- Now is a good time to start!

Redpill Linpro

Replication & Database Security

Questions?

magnus@hagander.net
Twitter: @magnushagander
http://blog.hagander.net