

# When Space Elephants Attack

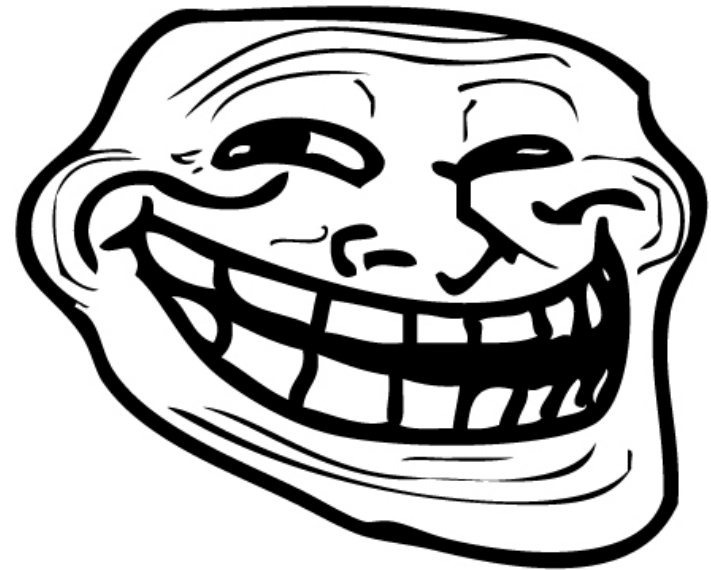
DEFCON 19

Presented by Abstrct



# Who Am I?

- Expert in designing videogame UI
- Specialized in Human-Computer Interaction
- Graphic Designer
- Technical Writer



Who am I really?

Database and Information Security Geek

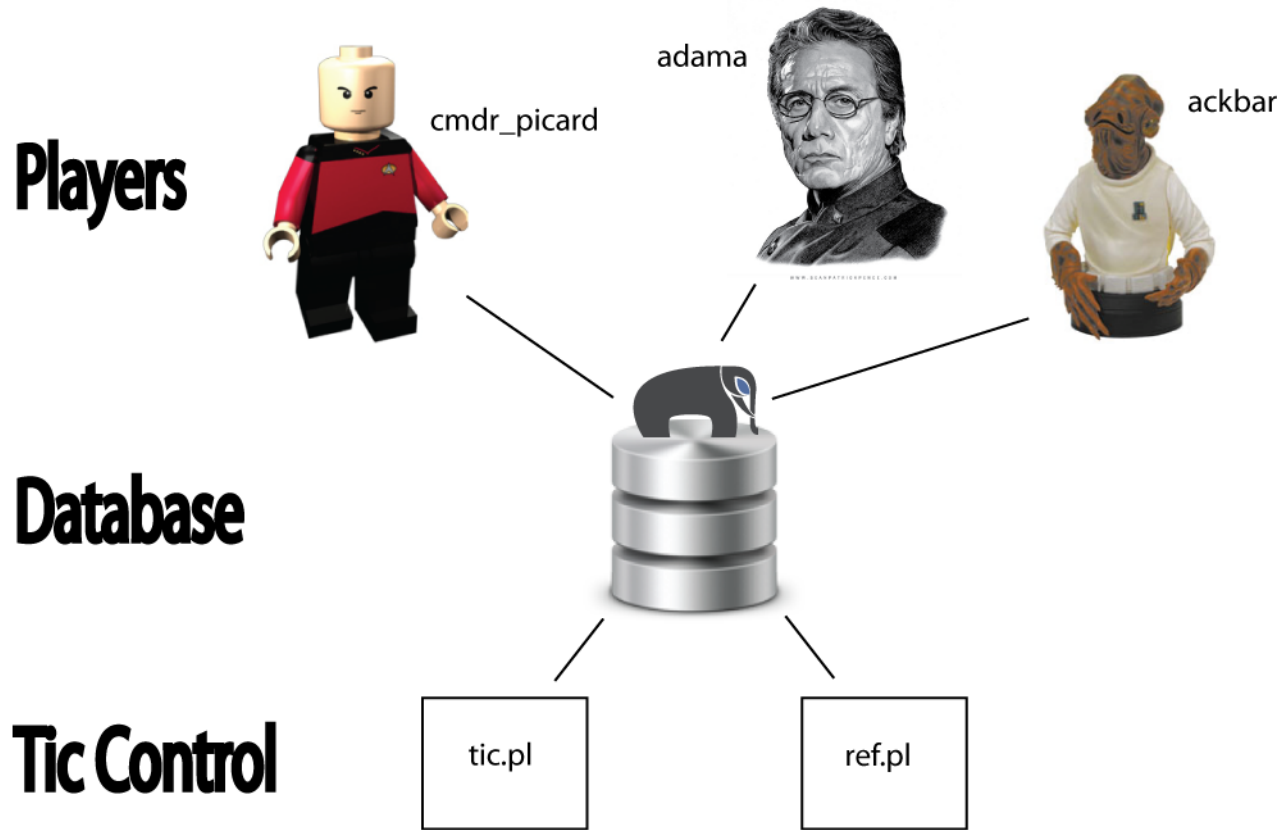
# What is The Schemaverse?

- Space game built entirely in PostgreSQL
- No UI other than the output of SQL queries
- Database is completely open to the Internet
- Security implemented within the database itself
- Unexpectedly addictive, fun and challenging

# Why?

- Be Educated and Educate on Database Design and Security
- Push Database Systems
- It's also pretty funny

# Architecture





**SECURITY**

# Securing PostgreSQL.. Insecurely

- Query Limits
  - Authentication
  - SQL Injections
- 
- some database communities were less than helpful



# SQL Injection Example

```
-- Fleet Script Creation
EXECUTE '
CREATE OR REPLACE FUNCTION FLEET_SCRIPT_' || NEW.id || '(
    RETURNS boolean as
$fleet_script$
DECLARE
    this_fleet_id integer;
    ' || NEW.script_declarations || '
BEGIN
    this_fleet_id := ' || NEW.id || ';
    ' || NEW.script || '
    RETURN 1;
END
$fleet_script$ LANGUAGE plpgsql;'::TEXT;
```

# SQL Injection Example

```
-- Fleet Script Creation
EXECUTE '
CREATE OR REPLACE FUNCTION FLEET_SCRIPT_' || NEW.id || '()' RETURNS boolean as
$fleet_script$
DECLARE
    this_fleet_id integer;
    ' || NEW.script_declarations || '
BEGIN
    this_fleet_id := ' || NEW.id || '
    ' || '
    UPDATE player SET balance=0;
    RETURN 1;
END
$fleet_script$ LANGUAGE plpgsql SECURITY DEFINER;
CREATE OR REPLACE FUNCTION MOVE() RETURNS boolean as
$fleet_script$
BEGIN
    --add the move code but altered slightly to send players backwards
    ' || '
    RETURN 1;
END
$fleet_script$ LANGUAGE plpgsql;'::TEXT;
```

# SQL Injection Fix

```
-- Fleet Script Creation
secret := 'fleet_script_' || (RANDOM()*1000000)::integer;
EXECUTE '
CREATE OR REPLACE FUNCTION FLEET_SCRIPT_' || NEW.id || '('
    RETURNS boolean as
    $' || secret || '$
DECLARE
    this_fleet_id integer;
    ' || NEW.script_declarations || '
BEGIN
    this_fleet_id := ' || NEW.id || ';
    ' || NEW.script || '
    RETURN 1;
END
$' || secret || '$ LANGUAGE plpgsql;'::TEXT;
```

# SchemaSecurity

- Roles
- Triggers
- Rules
- Views
- Functions



# **UNDERSTANDING THE GAME**

# Getting Started

- Using a client to connect
  - pgAdmin Demo

# Sequence of a Tic

- Every Ship Moves
- All Enabled Fleets Execute
- The Perform Mining Function Runs
- Some planets randomly have their fuel increased
- Damage/Repair is committed to the ship table
- Ships damaged for a set time are destroyed
- `tic_seq.nextval`

# Understanding the Relationships

- Ships
- Planets
- Events
- Player
- Fleets



# Creating Ships

- *INSERT INTO my\_ships(name) VALUES('Shipington');*
- *INSERT INTO my\_ships(name, attack, defense, engineering, prospecting) VALUES('My First Attacker',15,5,0,0);*
  - Total skill on insert must not be greater than 20
- *INSERT INTO my\_ships(name, location\_x, location\_y) VALUES("My Strategically Placed Ship", 100, 100);*
  - You can only create a ship where one of the following is true:
    - location\_x and location\_y is between -3000 and 3000
    - location\_x and location\_y are the same coordinates as a planet you are the current conqueror of

# Functions - Actions

- Attack(AttackerShip, EnemyShip)

*SELECT Attack(ship\_in\_range\_of, id), name FROM ships\_in\_range;*

*Uses my\_ships.attack and your enemy ships defense*

- Repair(RepairShip, DamagedShip)

*SELECT Repair(10, id) FROM my\_ships ORDER BY current\_health ASC;*

*Uses my\_ships.engineering to determine repair amount*

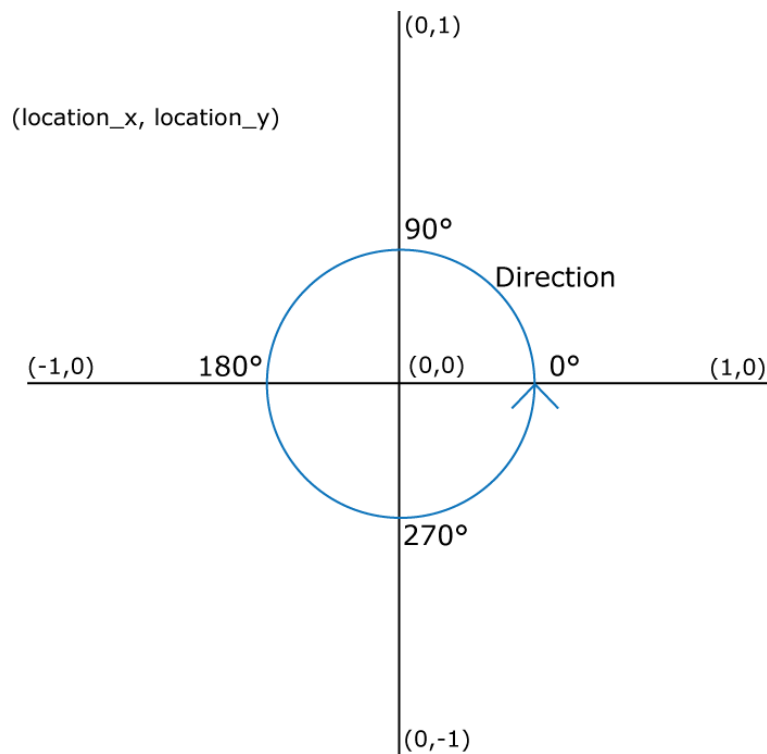
- Mine(MinerShip, Planet)

*SELECT mine(my\_ships.id, planets.id) FROM my\_ships, planets  
WHERE my\_ships.location\_x=planets.location\_x AND  
my\_ships.location\_y=planets.location\_y;*

*Uses luck and my\_ships.prospecting when mining is performed*

# Functions - MOVE

- *Move(Ship ID, Speed, Direction, Destination X, Destination Y)*



```
SELECT  
MOVE(id,100, NULL, destination_x, destination_y),  
id, name, location_x, location_y  
FROM my_ships;
```

```
UPDATE my_ships SET  
direction=90,  
speed=10  
WHERE name='Shipington'
```

# Functions - Others

- UPGRADE(Ship ID, Code, Quantity)
- REFUEL\_SHIP(Ship ID)
- CONVERT\_RESOURCE(Resource, Quantity)
- READ\_EVENT(Event ID)
- GET\_CHAR\_VARIABLE(Variable Name)
- GET\_NUMERIC\_VARIABLE(Variable Name)
- And some more

# How to Win

- Conquered Planets
- Most Ships
- Best Ships
- Most amount of destruction



**FLEET SCRIPTS**

# Fleets

- Why
- What
- How
  
- Demo of creating a script
- Error handling



- First ever Schemaverse Tournament at DEFCON!
- Starts: Thursday afternoon(?)
- Ends: Sunday at Noon
- Registration is in the Contest Area
- Prizes Include...





- The Schemaverse DEFCON Tournament
  - <http://defcon.schemaverse.com>
- Project Home: <http://Schemaverse.com>
- Github: <http://github.com/Absrtct/Schemaverse>
- Wiki: <http://github.com/Absrtct/Schemaverse/Wiki>

# Thank You

## DEFCON 19 Organizers!



Tigereye

appl

rick