EAR

(E)xtensible (A)pi for (R)econnaissance

What/Why/How

- Automatable Reconnaissance
- Similar Maltego
- Very early stage
- Rails 2.3.8

Pentesters care about

- Organizations
- Users
- Domains
- Devices (Hosts) / Services / Apps
- Locations
- etc...

USAGE

```
o = Organization.create :name => “masshackers”
o.tasks
o.run_task(“dns_tld_brute”)
```

USAGE

```
o = Organization.find_by_name("masshackers")
o.children
```

USAGE

o = Organization.find_by_name(“masshackers”)
o.domains
o.devices
o.users

USAGE

```
o = Organization.find_by_name("masshackers")
o.run_task("dns_tld_brute")
o.domains.each do |d|
d.run_task("dns_sub_brute")
end
```

Background Concepts

- Database Schema / Objects
- Active Record (Rails ORM)
- Task Manager
- Object Manager

Background Concepts

- ORM makes it easy to interact w/ a view of the world.

- Keep track of things you care about, but for free

Objects

- Rails makes it simple to declare objects
- Migration

```
create_table "organizations" do |t|
  t.string   "name"
  t.text     "description"
  t.string   "address"
  t.string   "email_mask"
end
```

- Class Definition

```
def Organization
end
```

Object Manager

- Maintains relationships between objects
- Who created who? (Parent / Child)

Tasks

- Methods to make sure they can operate on an object
- Setup/Run/Cleanup
- Create new objects!

Task Manager

- Maintains a list of known tasks
- Lets us check to see if we can operate on an object
- Runs task methods in the right order (setup / run / cleanup)
- Records task runs

Objects->Tasks

- Each Task has a:
  - allowed_types
  - update_types
  - create_types
- TaskManager checks these at task run time

Tasks -> Objects

- Task has a create_object method
  - creates the object
  - uses the object manager to maintain parent child relationships

Interacting

- Rails isn't just a web framework
- script/console is super-powerful
- interact directly with models

demo!