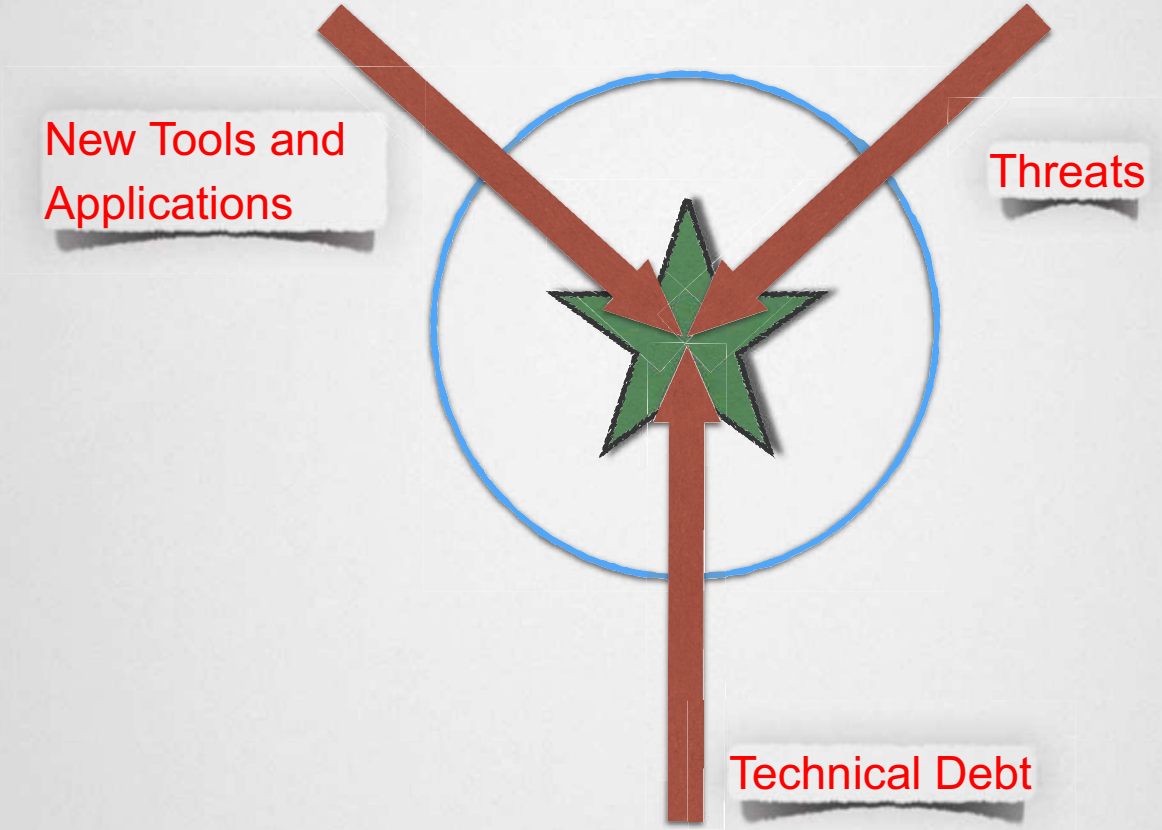
Securosis

Presents

Building Your SecDevOps Toolkit

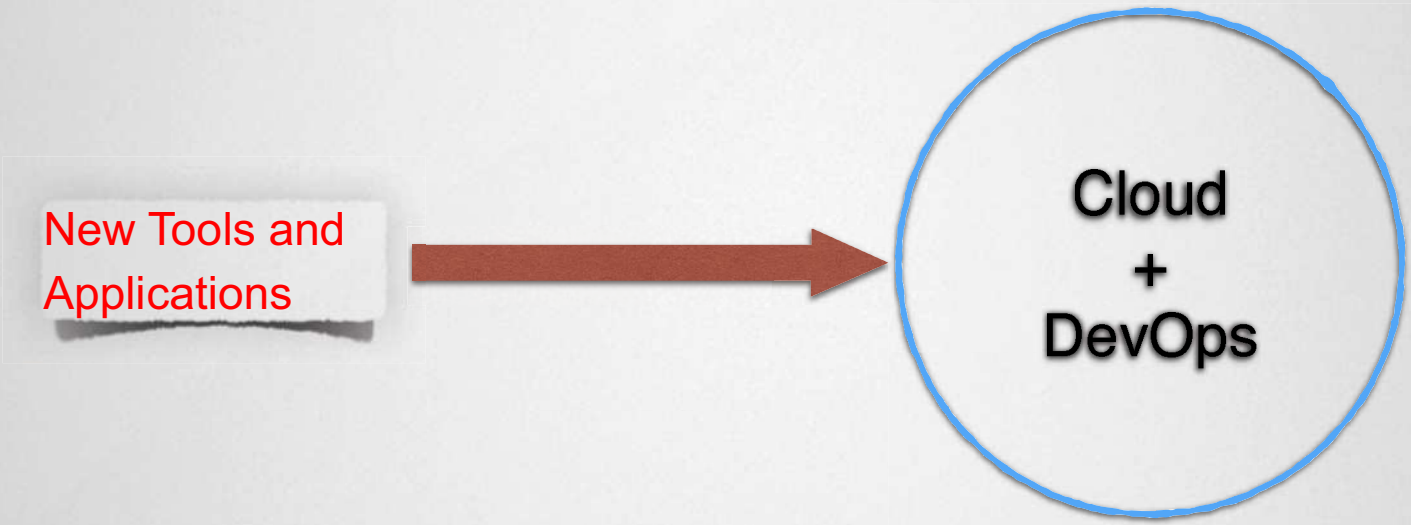
Rich Mogull
Analyst and CEO, Securosis LLC
@rmogull

The Problem



Securosis

The Problem



Cloud and DevOps

- Cloud is a new operational model.
- It requires a re-thinking of fundamental architectures.
- DevOps is a new operational framework, highly attuned to cloud.
- Both shatter existing security approaches.

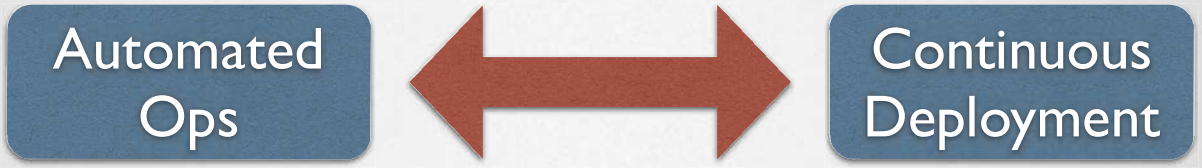
Securosis

Natives vs. Tourists

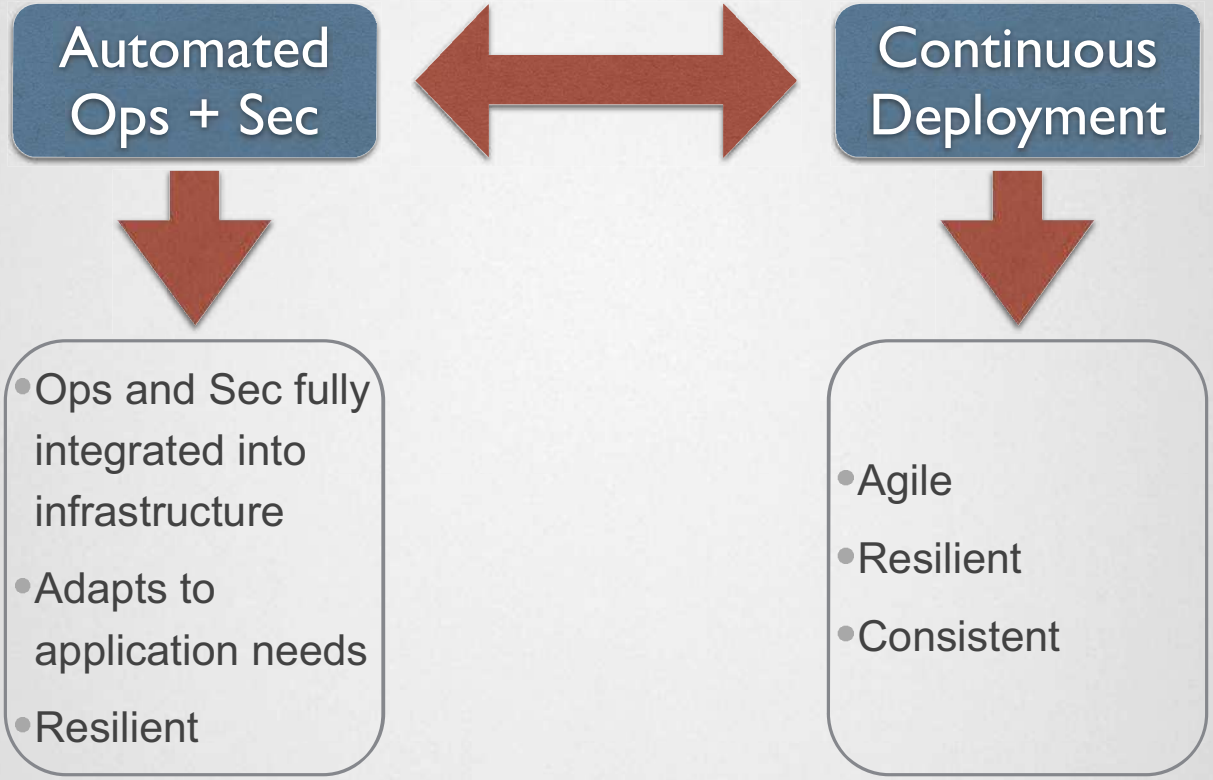
- Cloud tourists deploy their existing operational models and frameworks onto a cloud service, losing most of the benefits of cloud.
- Typically due to lack of knowledge, institutional momentum, and arbitrary economic models.

Securosis

Native Advantages



Native Advantages



The Security Profession Problem

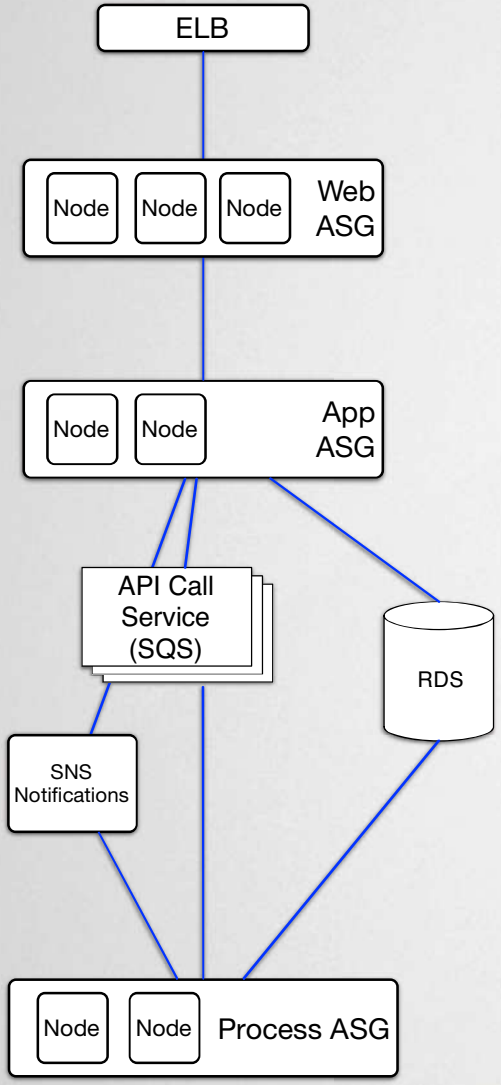
- The discipline that is most resistant to change and least likely to adapt is “Security”
- This resistance is usually excused due to a lack of trust and a reliance on people because we don’t trust security automation.
- “Security” continues to rely on a manual supply chain operated by the “Meat Cloud”
- Trustable automation and an operational model to support it is needed

Securosis

The Technical Security Challenge

- The vast majority of information security is really infrastructure-centric security.
- Infrastructure-centric security relies on fixed locations of relatively static resources.
  - Even many of our application security models rely on fixed infrastructure.
- It is context-unaware. DevOps and cloud are all about context.

Securosis



- No fixed servers.
- No fixed connections.
- Workloads scattered across IaaS and PaaS.
- All components elastic.
- Assessment and monitoring break.

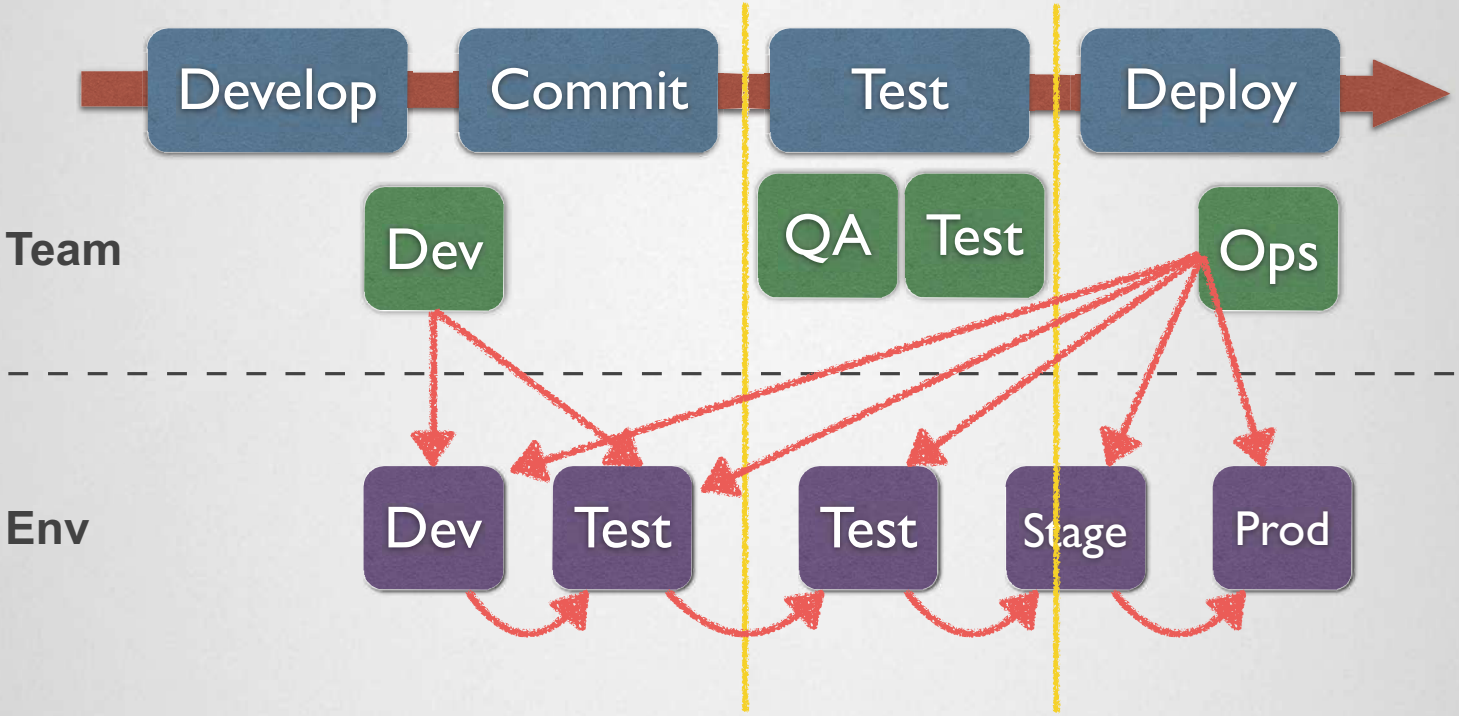
If you try to apply the traditional security operational framework to the new operational model!

Securosis

Why DevOps?

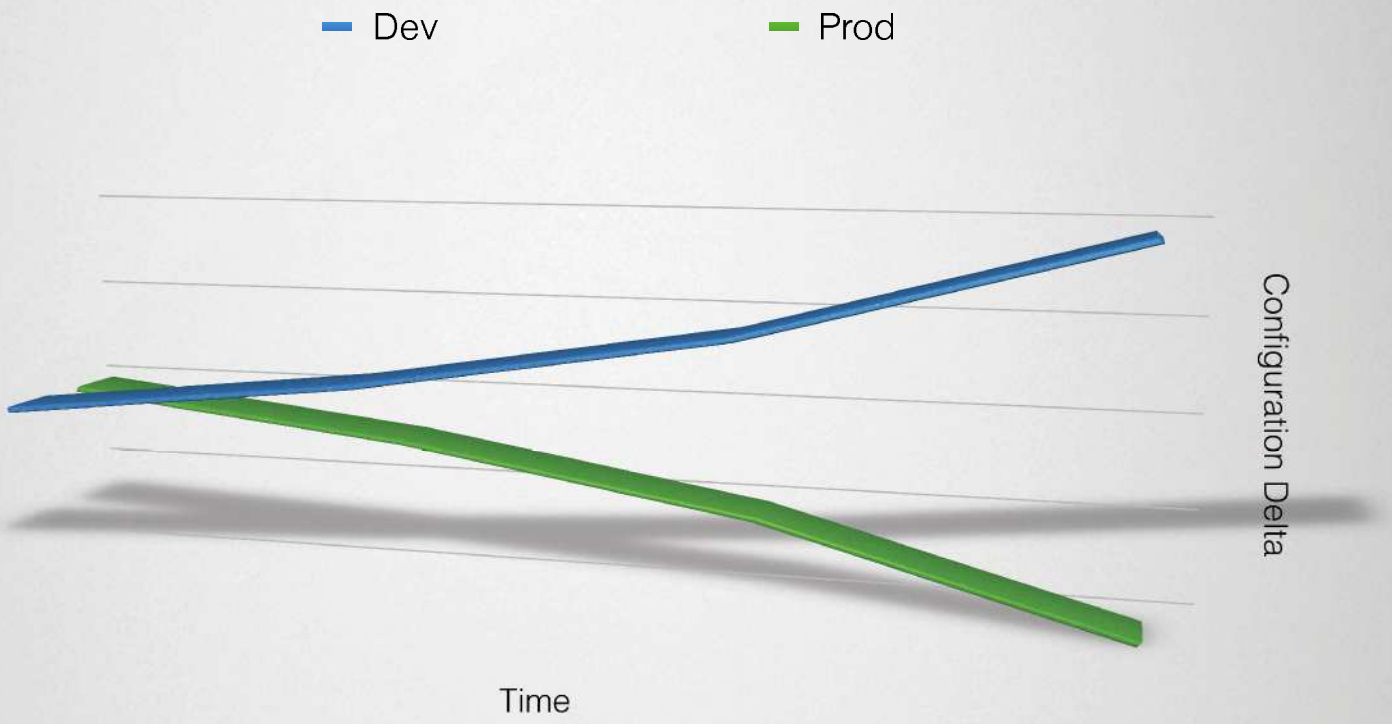
Application Deployment

Complexity Breeds Error



Securosis

Environments, Requirements, and Configurations Drift



Securosis

Cycle Times Matter

Agile? Waterfall?

All the same…

Securosis

The Dev and Ops Problem

- Configuratio
- The less aut
- Manual inter on.
- Environments become de-synced.
- Different teams work in different environments.

No standards

=

More complexity

Securosis

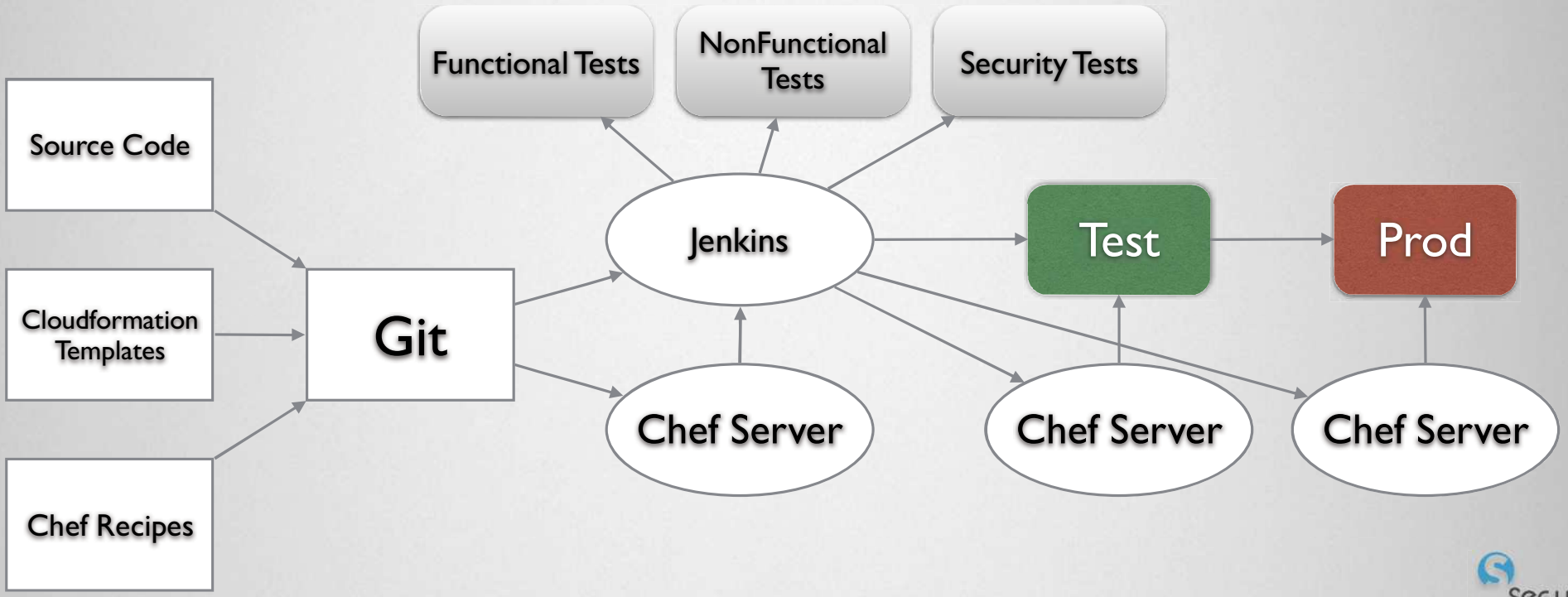
Enter DevOps

- DevOps is an *operational framework* that increases standardization, agility, and reliability.
- It relies heavily on virtualization, cloud, and automation.
- The principles come direct from Denning and Lean Manufacturing.

https://workingsmartercafe.files.wordpress.com/2012/09/kool-aid-man.jpeg

Securosis

Development Pipelines and Continuous Deployment



Why DevOps Works

- All environments, including supporting third-party applications, are consistent.
- The deployment pipeline is automated, for code, configurations, and toolsets.
- There is no drift.There are no human-induced errors.
- Faster deployment cycles reduce error and improve business agility.
- Version control and consistency support instant rollback.

[Sec]DevOps

Securosis

DevOps  Dream!

Securosis

Trustable Security Automation!

Securosis

DevOps + Security = ♡

- DevOps provides a **consistency** and **control** impossible with manual application deployments.
- Security can easily **embed** and **automate**.
- Security can steal DevOps techniques to apply to diverse workloads and infrastructure requirements.

Building Your Toolkit

- Integrate security tests into the deployment pipeline
- Build resilient server configurations
- Embrace Stateless Security
- Automate with Code and APIs

Yes, you need to learn some new skills, but you don't need to become a programmer.

Tooling the CI Server

GAUNTLT

BE MEAN TO YOUR CODE AND LIKE IT

README.txt

```
Mittn
=====

"For that warm and fluffy feeling"
```

Introduction to BDD-Security

(Need cooler logos)

Securosis

Build a Secure Config Library

```
if node[:mod_security][:crs][:bundled]
  # Bundled install
  remote_directory "owasp-modsecurity-crs-#{node[:mod_security][:crs][:version]}" do
    path  node[:mod_security][:crs][:rules_root_dir]
    owner "root"
    group "root"
    mode  "0755"
    action :create
    notifies :restart, 'service[apache2]', :delayed
  end
```

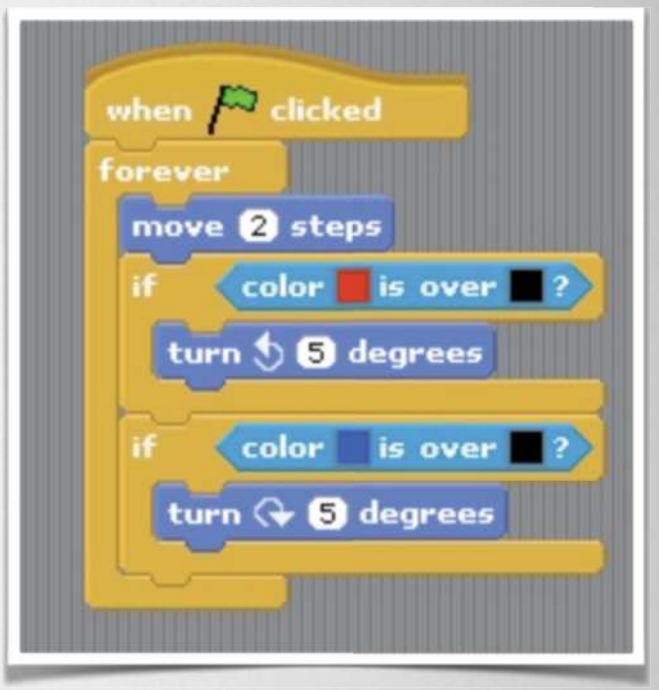
```
class Puppet::Provider::Firewall < Puppet::Provider

  # Prefetch our rule list. This is ran once every time before any other
  # action (besides initialization of each object).
  def self.prefetch(resources)
    debug("[prefetch(resources)]")
    instances.each do |prov|
      if resource = resources[prov.name] || resources[prov.name.downcase]
        resource.provider = prov
      end
    end
  end
```

Embrace Stateless Security

- AWS CloudTrail + CloudWatch
- AWS Lambda
- Automate with CloudFormation or scripts
  - Allows you to replicate across multiple accounts without having to create it from scratch.

Code without Coding

- Work with your devs to build a library of building blocks
- Learn just enough to glue it together
- Build some core scrips
- Mix and match the blocks
- Pull in the dev when you have new requirements



Securosis

Examples

- Snapshot all volumes attached to an instance
- Quarantine on the network
- Collect metadata
- Assess and change security group rules

```
487   def store_metadata
488     # Method collects the instance metadata and stores as a JSON variable
489     # TODO send data to Dynamo, with incident ID
490     # TODO update the incident_id to reflect the workflow id
491
492     data = @@ec2.describe_instances(instance_ids: ["#{@instance_id}"])
493     timestamp = Time.new
494     incident_id = {:timestamp => timestamp, :incident_id => "placeholder"}
495     # metadata = data.reservations.first.instances.to_h
496     metadata = data.to_json
497     puts "Instance metadata recorded"
498   end
```

Securosis

Demo Time

Software Defined Security

- Meet SecuritySquirrel, the first warrior in the Rodent Army (apologies to Netflix).
- The following tools are written by an analyst with a Ruby-for-Dummies book.
- Automated security workflows spanning products and services.

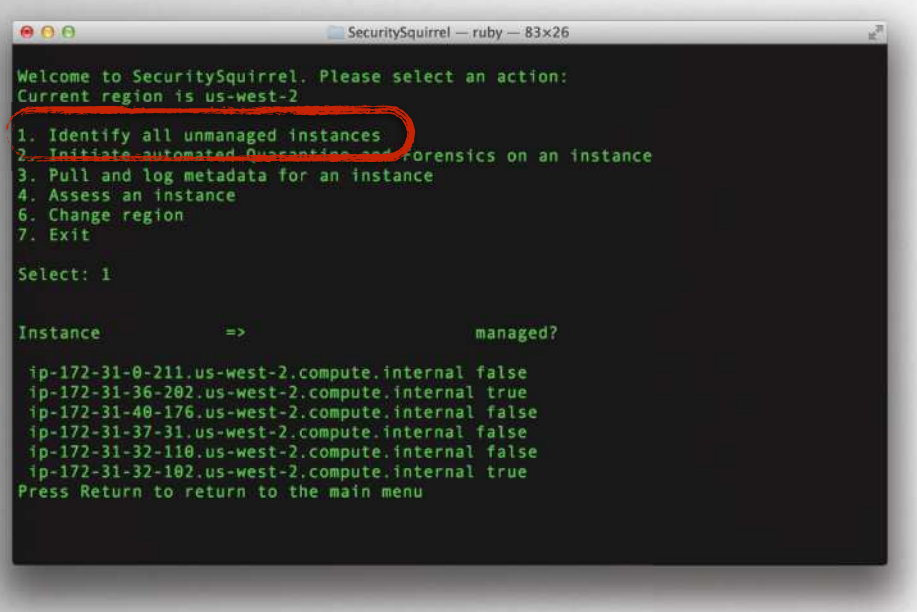
Securosis

Problem: Identify Unmanaged Servers

1. Scan the network
2. Scan again and again for all the parts you missed
3. Identify all the servers as best you can
4. Pull a config mgmt report
5. Manually compare results

Securosis

DEMO



1. Get list of all servers from cloud controller (can filter on tags/OS/etc).
   - Single API call
2. Get list of all servers from Chef
   - Single API call
3. Compare in code

Securosis

Problem: Incident Response

1. Detect Compromise
2. Pull server information (If you have it)
3. Quarantine
4. Image 
5. Analyze = Hours!
6. Recover

Each step is manual, and uses a different set of disconnected tools

Securosis

1. Pull metadata
2. Quarantine
3. Swap control to security team
4. Identify and image all storage
5. Launch and configure analysis server
6. Can re-launch

Securosis

DEMO

How do you do this without automation?

Securosis

Automagic WAF

Winning SecDevOps

- Add security to deployment pipelines and automate testing.
  - New tools are here, but we need more…
- Build a library of server hardening scrips and inject into config management.
- Integrate directly into your cloud service for alerting/ monitoring.
- Create a library of automation code you can mix and match for different security needs.

Securosis

Securosis

Presents

Building Your SecDevOps Toolkit

Rich Mogull
rmogull@securosis.com
@rmogull
http://securosis.com

Securosis

Presents

Building Your SecDevOps Toolkit

Rich Mogull
Adrian Lane
Mike Rothman

http://securosis.com

@rmogull
@adrianlane
@securityincite