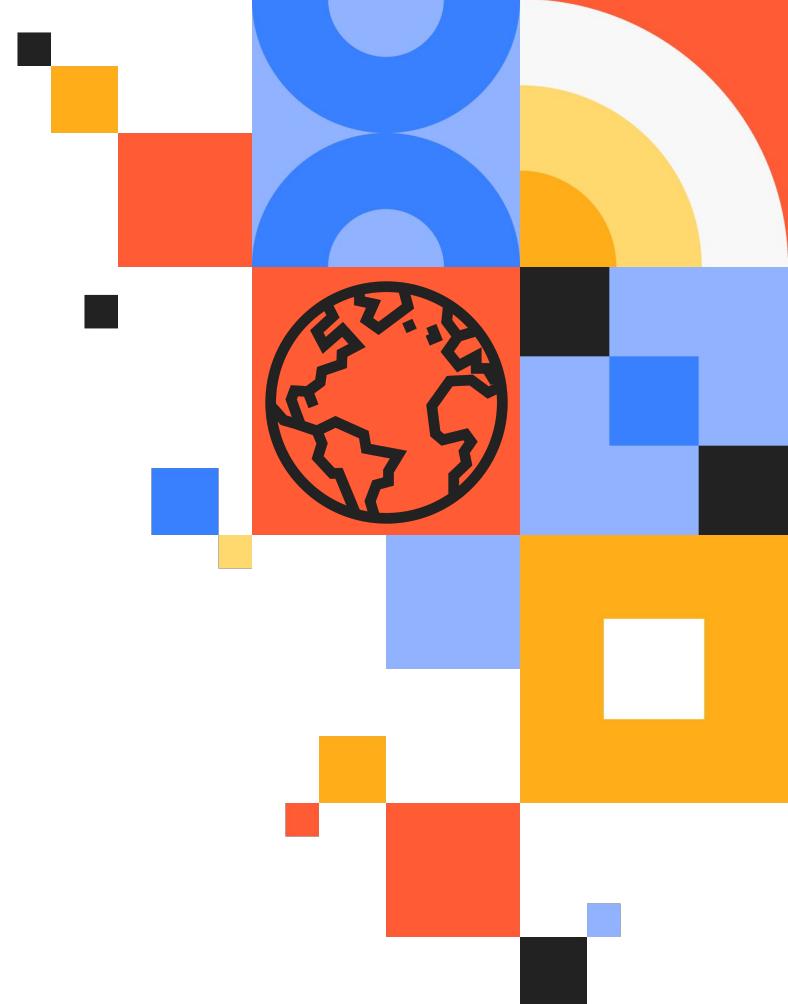




Puppet Enterprise with Windows

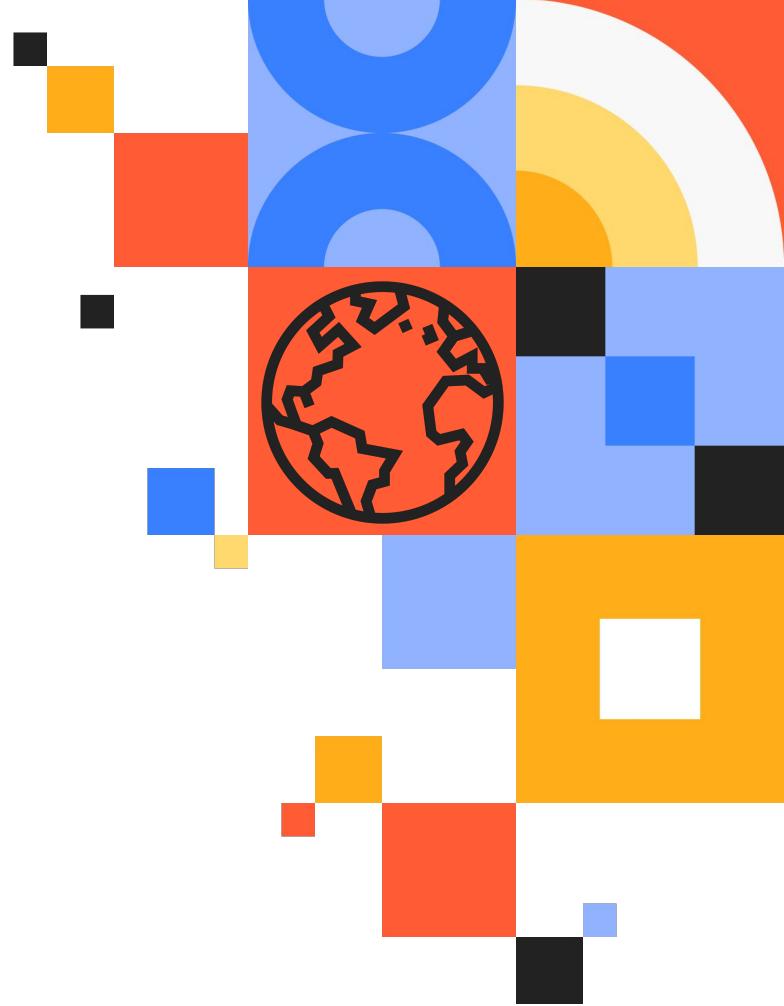
John O'Connor
Claire Cadman
Dave Johnston

23 September 2019





WIFI Sid: XENIA
Username: ccentre
Password: 99cityroad



We do DevOps better
because we need to
Deliver better software faster

Accelerate infrastructure delivery

- Remediate security vulnerabilities
- Capabilities for every stage of your automation journey
- Continually monitor & enforce compliance
- Integrated CI/CD workflows for your infrastructure code

The screenshot shows the Puppet Enterprise web interface. The top navigation bar includes the Puppet logo and the word "Enterprise". On the left, a sidebar menu is open under the "INSPECT" heading, with "Events" highlighted in yellow. Other menu items include "Overview", "Nodes", "Packages", "Reports", "Jobs", and "Events". Below the sidebar, there are three main sections: "Classes with events", "Nodes with events", and "Resources with events", each displaying a list of categories and their counts. A search bar at the top right allows filtering by class name. A note at the top says "Get insight into the failures and changes across your Puppet runs." and a dropdown menu says "Show: Events from the last run".

Events

Get insight into the failures and changes across your Puppet runs.

Show: Events from the last run

[Summary](#) Classes with corrective changes

Classes with events

Category	Count
All	25
Failures	5
Corrective changes	18
Intentional changes	2
Corrective no-ops	0
Intentional no-ops	0
Skips	0

Nodes with events

Category	Count
All	13
Failures	7
Corrective changes	3
Intentional changes	4
Corrective no-ops	0
Intentional no-ops	0
Skips	0

Resources with events

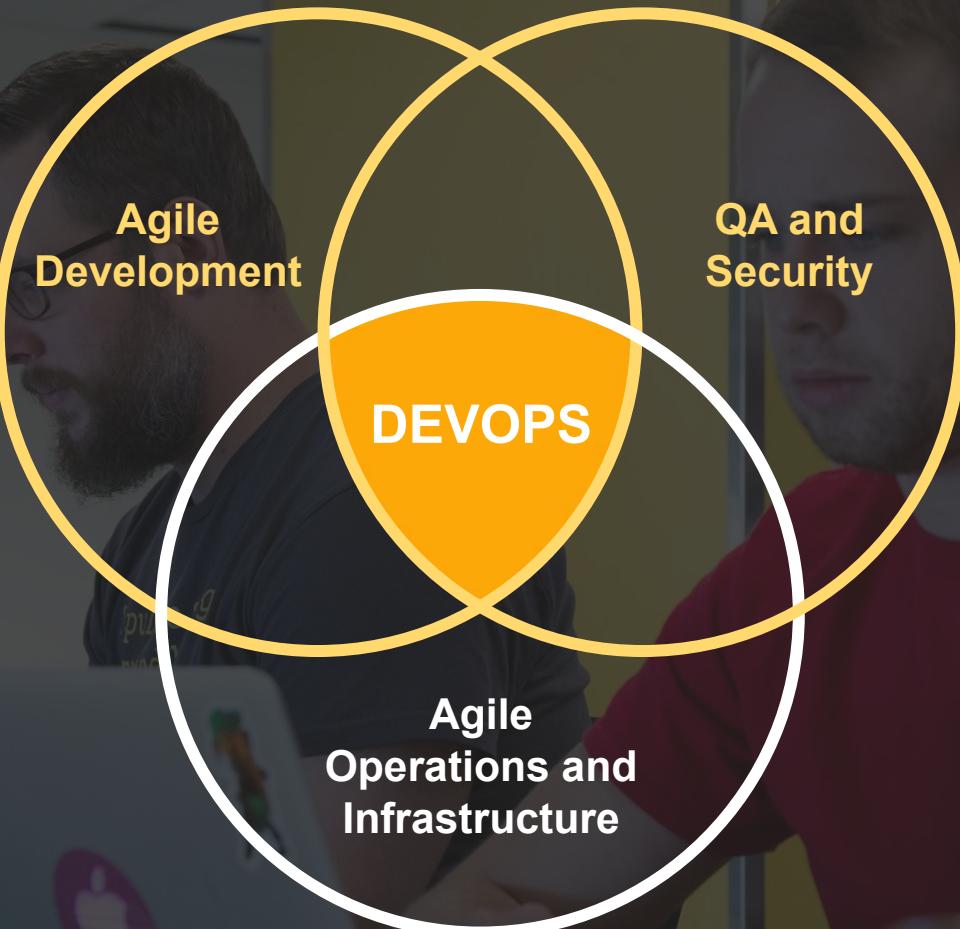
Category	Count
All	6
Failures	1
Corrective changes	2
Intentional changes	3
Corrective no-ops	0
Intentional no-ops	0
Skips	0

Filter by class name:

Classes with corrective changes: 18

Table shows nodes affected per class.

Class	Failure
apache	
apache::vhost	
app	
base::linux	
base::unix	
chocolatey	
chocolatey::install.pp	
chocolatey::config.pp	
docker	
java	
java::config	
java::params	
postgresql::globals	
stdlib	
stdlib::stages	



Agile
Development

QA and
Security

DEVOPS

Agile
Operations and
Infrastructure

Fast Feedback

Collaboration

Iteration

Visibility

Infrastructure as Code

Version
Control

Configuration
Management

Peer Review

Continuous
Delivery

Automated
Testing &
Deployments



Learning is not compulsory, but neither is survival.

Edward W. Deming, <http://bit.ly/deming14pts>

Pizza Anyone ????



Ordering a pizza - MENU

- Cheese & Tomato
- Mozzarella cheese (tomato, cheese)
- Vegetarian (tomato, onions, pepper, black olives)
- Pepperoni (tomato, cheese and spanish sausage)

Ordering a pizza - MENU & INSTRUCTIONS

Cheese & Tomato

Tomato sauce

Where is the tomato sauce?

How do I open the can?

How do I get the sauce out of the can?

How much tomato sauce should I put
on?

EXECUTION

Put a thin layer of sauce on the
dough.

Sprinkle some mozzarella cheese

Add a handful of onions

Cheese

How do I get from the tomato sauce to
the cheese?

Where is the cheese?

How do I slice it?

Add a handful of peppers

Add a few black olives

Procedural scripts for a pizza order

```
***** Begin Procedure Script *****

BEGIN

GOTO cupboard in the stockroom.
GOTO third shelf from the bottom on left hand side facing the back.

can = Take the can of tomato sauce

open $can

sauce = pour  $\frac{1}{2}$  $can onto the pizza dough

END

***** End Procedure Script *****
```

Order a pizza with Puppet

Easy for a person to write and understandable for a computer

```
pizza { "John's order":  
    sauce => tomato,  
    cheese => mozzarella,  
    vegetables => [ onions, peppers, black olives ],  
    meat => pepperoni  
}
```

Procedural (imperative) vs Declarative

Or Imperative .vs. XXXX (if you prefer)

Ordering from a computer

I need to tell the computer how to do what I want, step by step.



Ordering from a person

I just need to know what I want. The person knows how to do it.



A Business Scenario

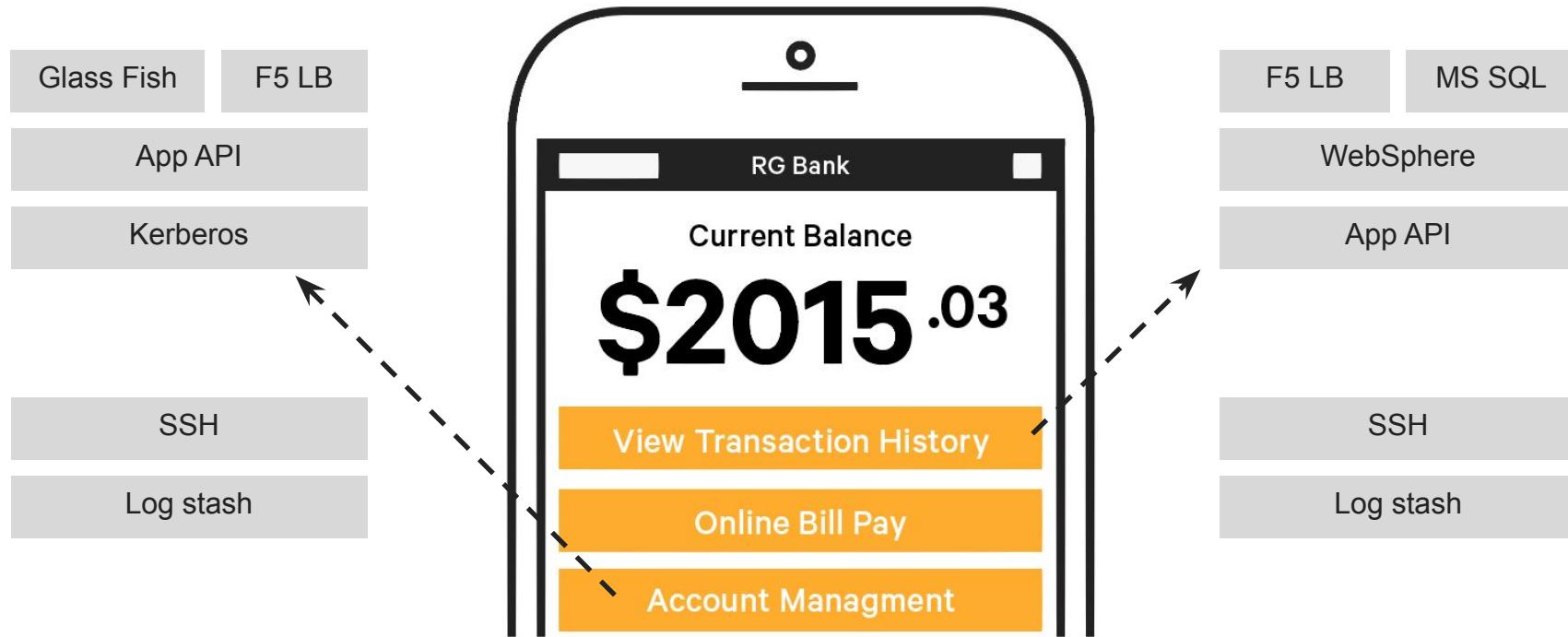
Why are we doing all this?

Why does Puppet matter in the modern world?



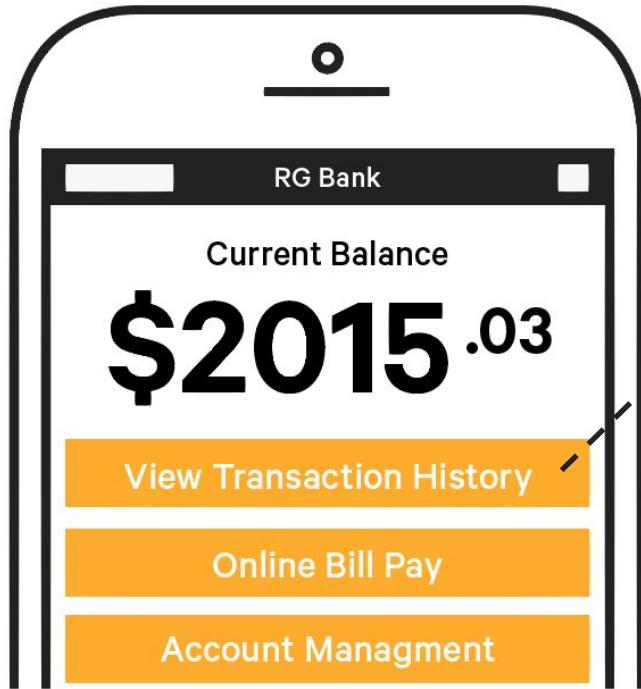
The APP example

App relies on multiple services that span infrastructure

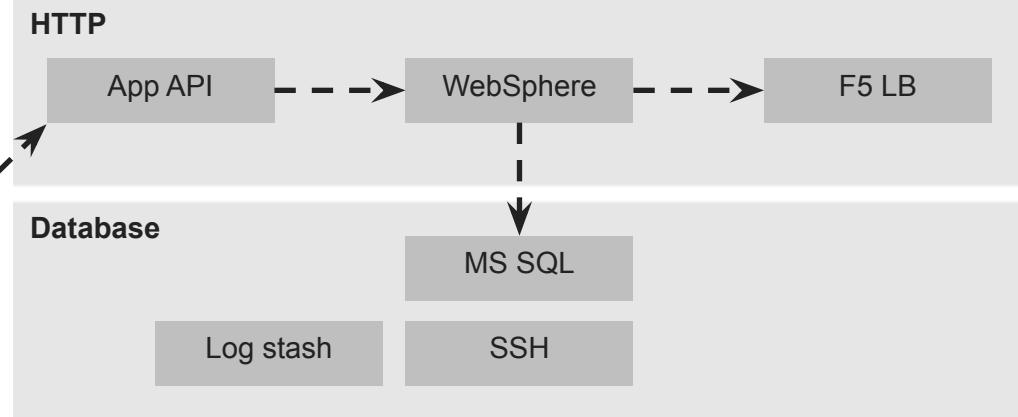


The APP example

Use Puppet language to model your application



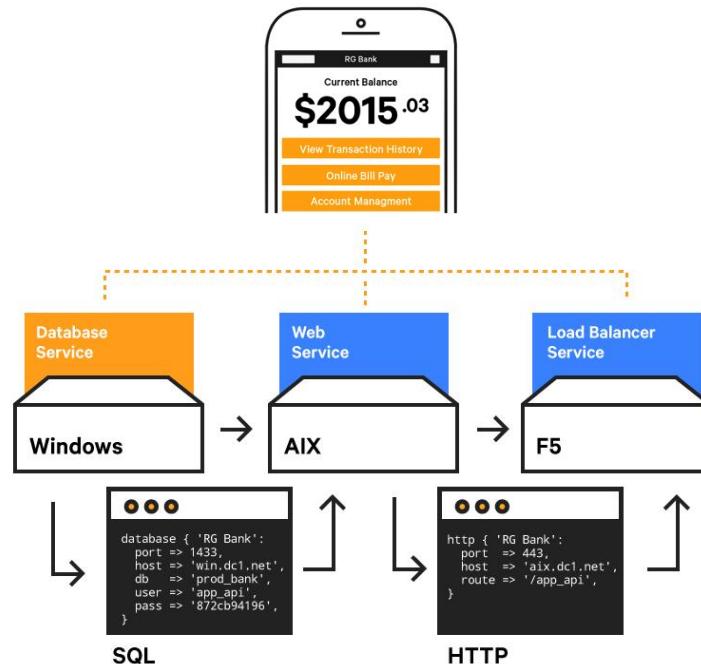
Model Services, Dependencies, Relationships and Instances



Across all environments: Dev, Staging, Production

Example application

Puppet Enterprise determines order and deploys your application.



Benefits of using Puppet over scripts

Scripts only provision, they don't manage “Day 2” and beyond.

- Puppet manages the infrastructure that it provisions. Because it knows the desired state of the infrastructure it provisions, it knows automatically if something has changed AND has the ability to enforce that configuration.

Scripts are specific to an operating system or network device.

- Puppet's modules (over 5000 available to all Puppet users at forge.puppet.com) deploy the desired state to any server with any operating system.

Scripts are typically dependent on one person.

- Puppet employs a common language that any developer or administrator can easily adopt. The problem of individuals and siloed teams managing OS dependent scripts is eliminated.

Scripts aren't managed as software.

- Puppet is software. It employs the disciplines of software development like check-in and check out using any standard code repository so it can be leveraged by any team member.

Summary of Benefits

- Automates provisioning, configuration and ongoing management
- Eliminates time spent configuring, deploying, troubleshooting and managing infrastructure
- Allows teams to quickly and confidently use repeatable processes to deploy applications
- Make infrastructure-wide changes
- Enforces consistency for all systems, devices and applications
- Automatically remediates unauthorized changes
- Minimizes and often eliminates both configuration drift and downtime

A photograph showing a close-up of a person's hands using metal tongs to serve food from a large, shallow white platter. The food appears to be a salad or a mix of shredded vegetables. In the foreground, the edge of another platter filled with similar food is visible. The background is slightly blurred.

Puppet provides IT organisations with a declarative language for infrastructure as code that all teams can use and that automates the management of IT infrastructure.

Puppet basics learning

Puppet Language, Roles and Profiles,
Modules and the Forge



The basics of Puppet language - Resources

Resource Type listing

```
[root@training ~]# puppet describe --list
These are the types known to puppet:
anchor          - A simple resource type intended to be used a ...
augeas          - Apply a change or an array of changes to the ...
computer        - Computer object management using DirectorySer ...
cron            - Installs and manages cron jobs
exec             - Executes external commands
file             - Manages files, including their content, owner ...
file_line        - Ensures that a given line is contained within ...
firewall         - This type provides the capability to manage ...
firewallchain   - This type provides the capability to manage ...
group            - Manage groups
host             - Installs and manages host entries
ini_setting      - ... no documentation ...
ini_subsetting   - ... no documentation ...
interface        - This represents a router or switch interface
java_ks          - Manages entries in a java keystore
k5login          - Manage the '.k5login' file for a user
macauthorization - Manage the Mac OS X authorization database
mount            - Manages mounted filesystems, including puttin ...
...
...
```

Common resource types

```
package { 'ssh':
  ensure => present,
  name   => 'openssh-clients',
}

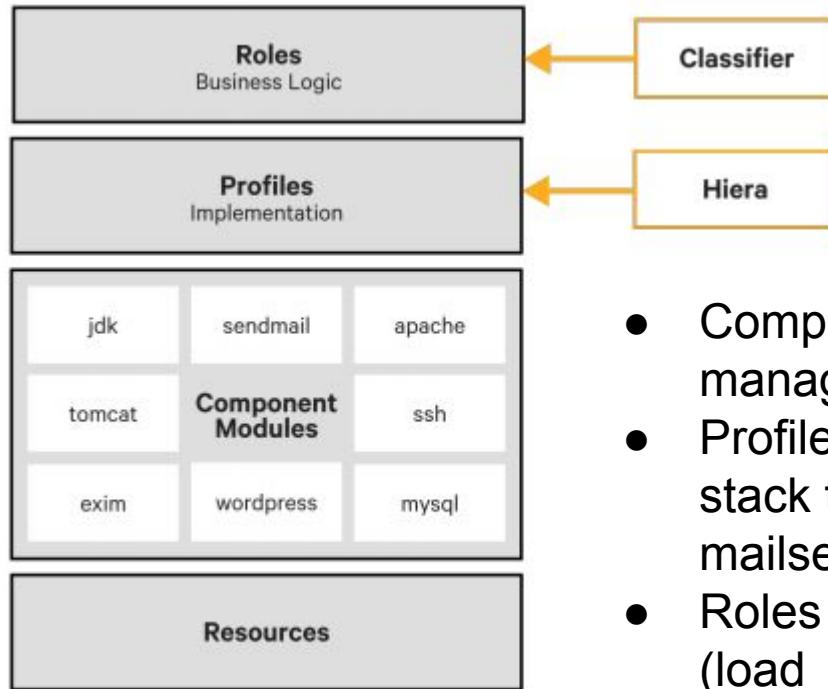
file { 'sudoers':
  ensure => file,
  path   => '/etc/sudoers',
  source => 'puppet:///modules/sudo/sudoers',
}
```

The basics of Puppet language - Classes

```
# A class with no parameters
class base::linux {
    file { '/etc/passwd':
        owner => 'root',
        group => 'root',
        mode  => '0644',
    }
    file { '/etc/shadow':
        owner => 'root',
        group => 'root',
        mode  => '0440',
    }
}

# A class with parameters
class apache (String $version = 'latest') {
    package {'httpd':
        ensure => $version, # Using the class parameter from above
        before => File['/etc/httpd.conf'],
    }
    file {'/etc/httpd.conf':
        ensure  => file,
        owner   => 'httpd',
        content => template('apache/httpd.conf.erb'), # Template from a module
    }
    service {'httpd':
        ensure    => running,
        enable    => true,
        subscribe => File['/etc/httpd.conf'],
    }
}
```

Roles and profiles



- Components should be named after what they manage (apache, ssh, mysql);
- Profiles should be named after the technology stack they implement (database , bastion , mailserver)
- Roles should be named by business roles (load_balancer , web_cluster , application , archive)

Modules and the Forge

Modules are self-contained bundles of code and data. You can write your own modules or you can download pre-built modules from [the Puppet Forge](#). - <https://forge.puppetlabs.com/>

The demo's today focuses on modules for windows including:

- puppet-windows_env
- puppetlabs-dsc_lite
- puppetlabs-iis
- puppetlabs-powershell
- puppetlabs-registry
- puppetlabs-stdlib
- puppetlabs-wsus_client

Puppet VSCode Extension

Get started writing Puppet manifests

Claire Cadman

VSCode extension website:

<https://puppet-vscode.github.io/>

Office hours:

<https://puppet.com/community/office-hours>



Hands on Puppet

Use Puppet to configure and manage a group of pre-provisioned Windows servers on Azure

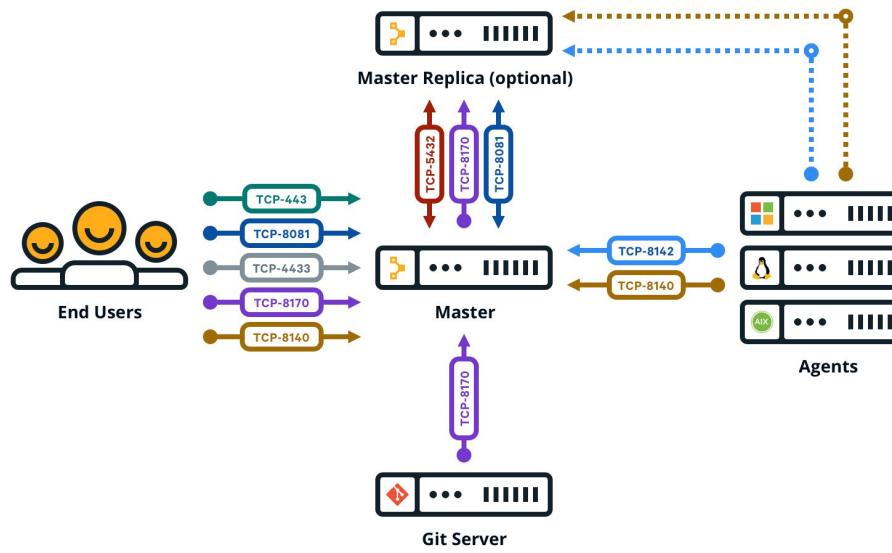
John O'Connor



WinOps “Demo” Configuration

- Add some registry entries using puppet-registry
- Create two Demo Users
- Install various packages
- Change Power Setting to Balanced
- DSC Examples:
 - Add some DSC Registry Entries
 - Use DSC to add a file to the desktop
 - Use DSC to add a user

Puppet Master/Agent Architecture



Key	HTTPS TCP-443	PUPPETDB TCP-8081	NODE CLASSIFIER TCP-4433	PUPPET AGENT/STATUS ENDPOINT TCP-8140
	ORCHESTRATION TCP-8142	POSTGRESQL TCP-5432	CODE MANAGER TCP-8170	

Demonstration Nodes

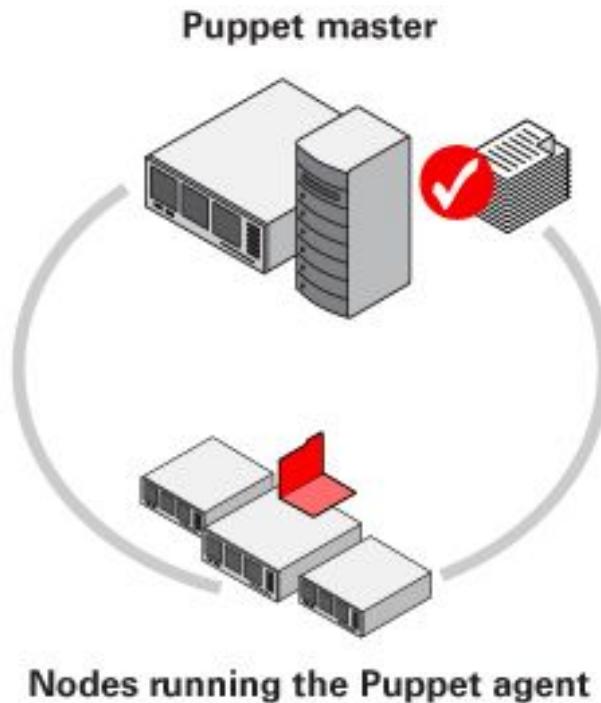
- Windows Server 2019 on Azure setup with Bolt/Azure_ARM
- Have pre-installed Puppet Agent and:
 - Visual Studio Code
 - Google Chrome
 - Git
- Will use Puppet Agent to further Configure the node and demonstration Puppet Corrective actions
- Package Inspection
- Enabling WSUS

Puppet Agent

- Is installed and runs on all client nodes (and on the master).
- Normally runs every 30 minutes but can be configured to:
 - Run at different intervals
 - Run Manually
 - Run in NoOP mode
- Can also be run on Demand from the Puppet Enterprise Console or through a Remote Management Process

Puppet Master/Agent Interaction

1. Node that is running the Puppet agent collects data about itself using facts
2. Agent sends facts to Puppet master
3. Master compiles a catalog based on data for how the node should be configured
4. Master sends catalog back to agent
5. Agent configures itself and reports back to master



“Puppet Agent -t” First Run

```
PS C:\Users\puppet> puppet agent -t

Info: Downloaded certificate for ca from
puppet.305dypwoztkuhgv5no5fu5a5xa.zx.internal.cloudapp.net
Info: Downloaded certificate revocation list for ca from
puppet.305dypwoztkuhgv5no5fu5a5xa.zx.internal.cloudapp.net
Info: Creating a new SSL key for winlarge-04
Info: csr_attributes file loading from C:/ProgramData/PuppetLabs/puppet/etc/csr_attributes.yaml
Info: Creating a new SSL certificate request for winlarge-04
Info: Certificate Request fingerprint (SHA256):
4B:F6:70:09:4A:F1:AE:F7:EE:CA:7E:EB:5B:0F:11:43:76:6C:D1:A4:DA:7F:85:DF:A6:9F:DC:3C:79:CE:9B:22
Info: Downloaded certificate for winlarge-04 from
puppet.305dypwoztkuhgv5no5fu5a5xa.zx.internal.cloudapp.net
Info: Using configured environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Notice: /File[C:/ProgramData/PuppetLabs/puppet/cache/lib/facter]/ensure: created
Notice: /File[C:/ProgramData/PuppetLabs/puppet/cache/lib/facter/aio_agent_build.rb]/ensure:
```

“Puppet Agent -t” - Run - No Changes

```
PS C:\Users\puppet> puppet agent -t
Info: Using configured environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Retrieving locales
Info: Loading facts
Info: Caching catalog for winlarge-04
Info: Applying configuration version '1568974493'
Notice: Applied catalog in 29.26 seconds
PS C:\Users\puppet>
```

“Puppet Agent -t” First Run

```
PS C:\Users\puppet> puppet agent -t
Info: Using configured environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Retrieving locales
Info: Loading facts
Info: Caching catalog for winlarge-04
Info: Applying configuration version '1568974493'
Notice: /Stage[main]/Profile::Users::Demousers/User[winopsdemo_1]/ensure: created
Notice: /Stage[main]/Profile::Users::Demousers/User[winopsdemo_2]/ensure: created
Notice: /Stage[main]/Profile::Power::Power/Exec[set power scheme]/returns: executed successfully
Notice: /Stage[main]/Profile::Util::Util/Package[noteepadplusplus]/ensure: created
Notice: /Stage[main]/Profile::Util::Util/Package[treesizefree]/ensure: created
Notice: Applied catalog in 29.26 seconds
PS C:\Users\puppet>
```

Puppet Remediate

Find and Fix Vulnerabilities

Dave Johnston



What are we doing today?

- What is Puppet Remediate (vulnerability management)
- Where can you get it
- Demo: Remediating a Windows CVE

Vulnerability management - What is it?

Vulnerability management is the practice of identifying, classifying, prioritizing, remediating, and mitigating, software **vulnerabilities**.



Is vulnerability management and patching the same thing?

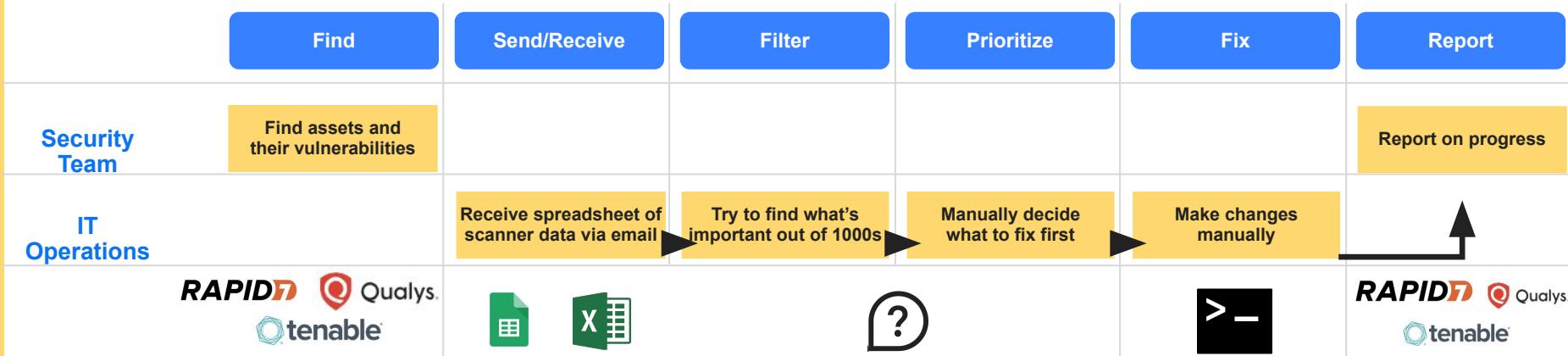
They are related but they are not the same thing.

Patching helps preventing vulnerabilities but is not enough!

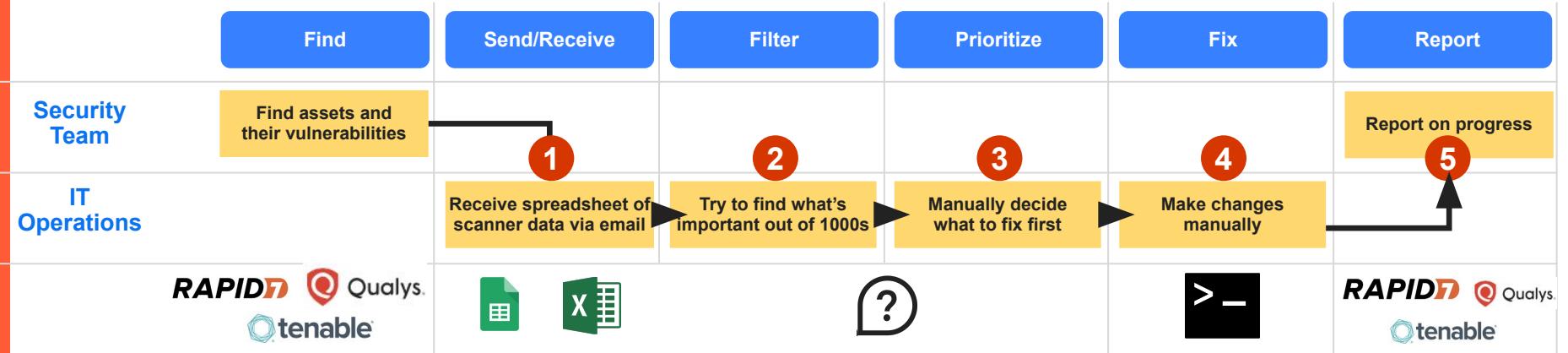
How do I identify vulnerabilities left in the system?

I can use vulnerability scanners, like:
Qualys, Rapid7 and Tenable.

Vulnerability management process



Typical problems with managing vulnerabilities

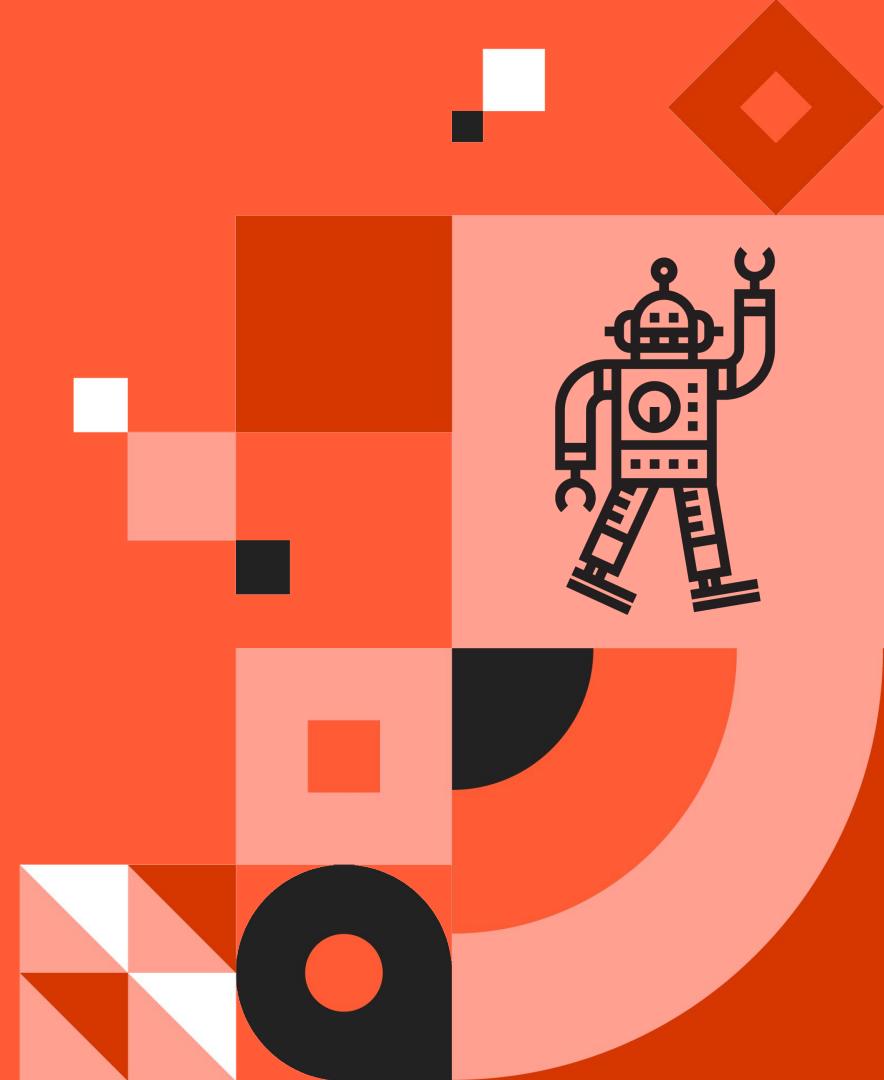


- 1 IT Ops wastes time confirming whether data is current.
- 2 Don't know what's important to fix out of thousands.
- 3 Takes days/weeks to prioritize critical vulnerabilities.
- 4 Takes too long. No standard approach to fixing.
- 5 Completely manual. No reliable way to show what's fixed.

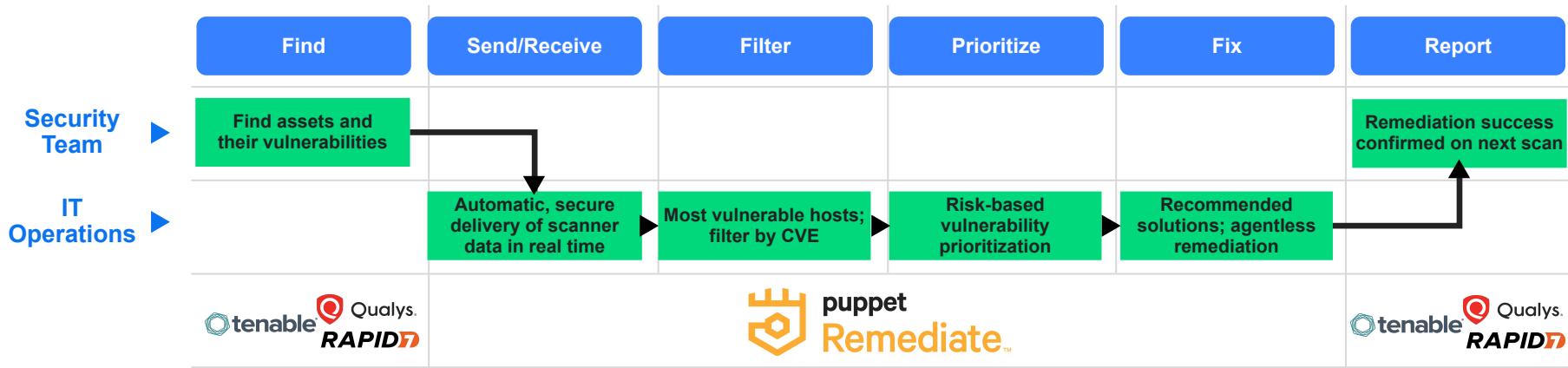
Implications

Data breaches/loss of share & brand value
Losing people due to lack of DevSecOps
Inability to fix vulnerabilities fast enough

Puppet Remediate helps organizations mitigate their security risks.



Puppet Remediate + scanner integrations accelerate the vulnerability remediation workflow



Customers such as a global retailer and a U.S. Federal agency expect to:

- Reduce the time to fix vulnerabilities from weeks or days down to less than a day.
- Reduce the number vulnerabilities in their systems.
- Get through audits faster and with less effort.



Remediate Windows Hosts



Installation

Register for free trial license here:

<https://puppet.com/products/puppet-remediate>

Download the compose file:

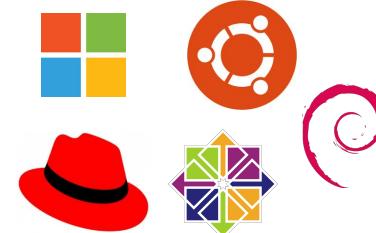
<https://storage.googleapis.com/remediate/stable/latest/docker-compose.yml>

```
choco install docker-desktop  
docker swarm init  
docker-compose run remediate start --license-file  
.license.json
```

Login to Remediate:

<https://<hostname/ip>:8443>

Supported Platforms

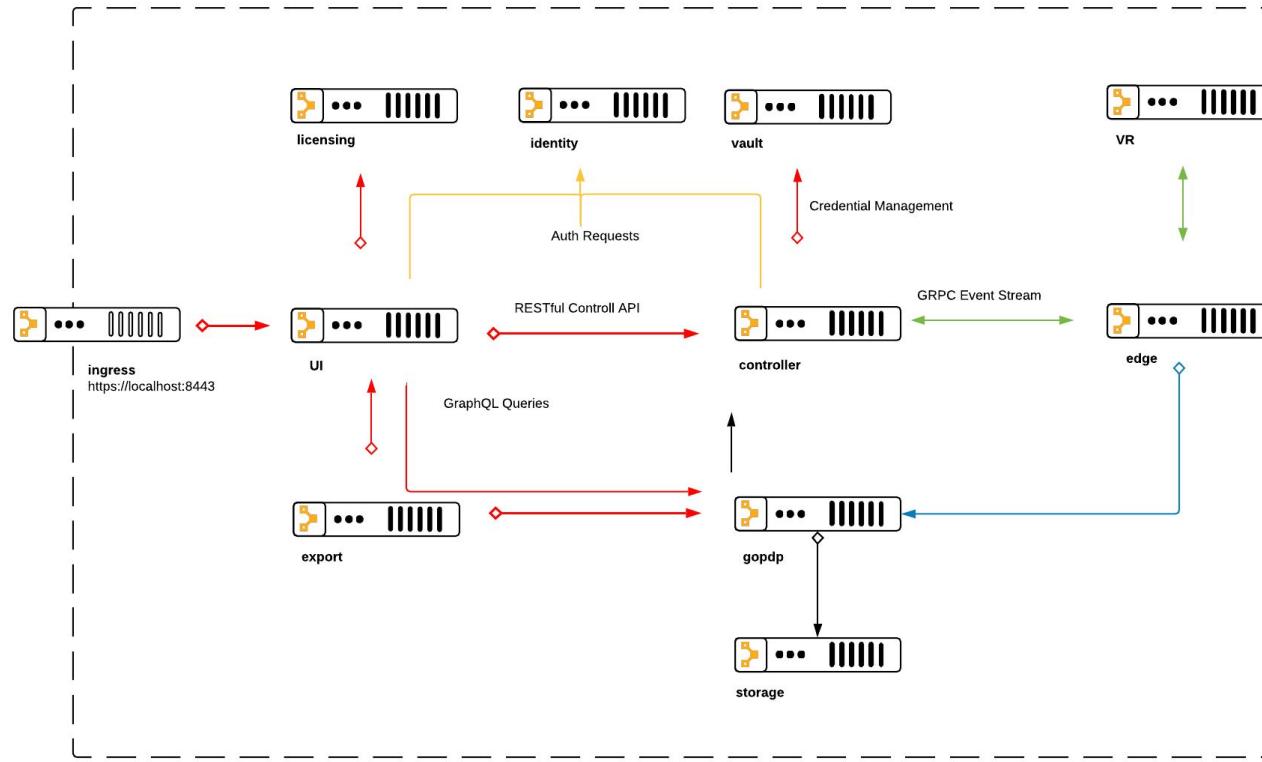


Prerequisites

Docker



Architecture



RAPID7 Qualys

Tenable

aws

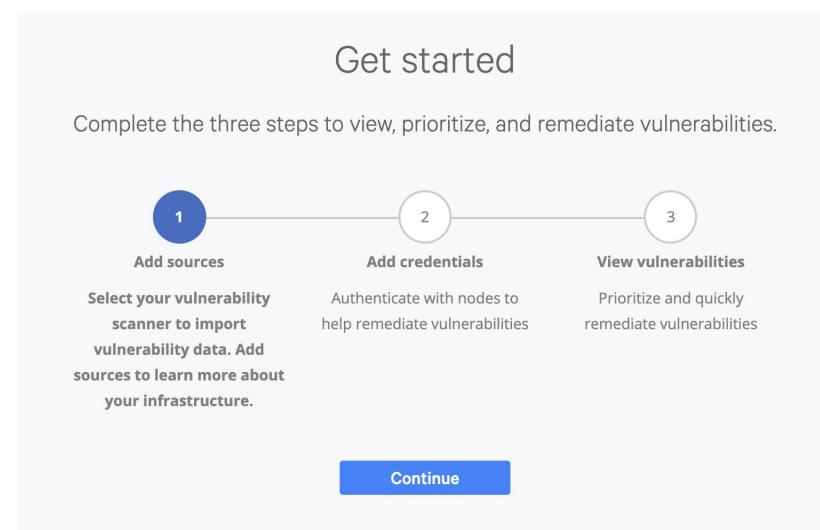
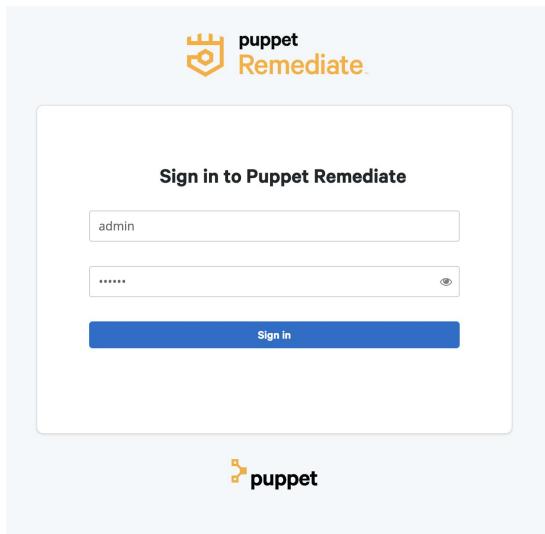


Login and first time flow

<https://remediate.winops2019.automationdemos.com:8443>

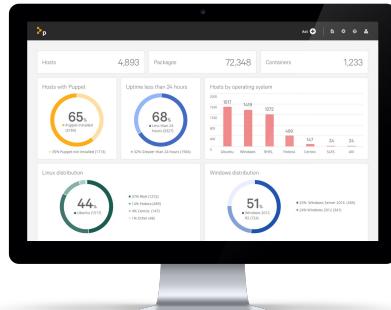
Login: user01

Password: remediate

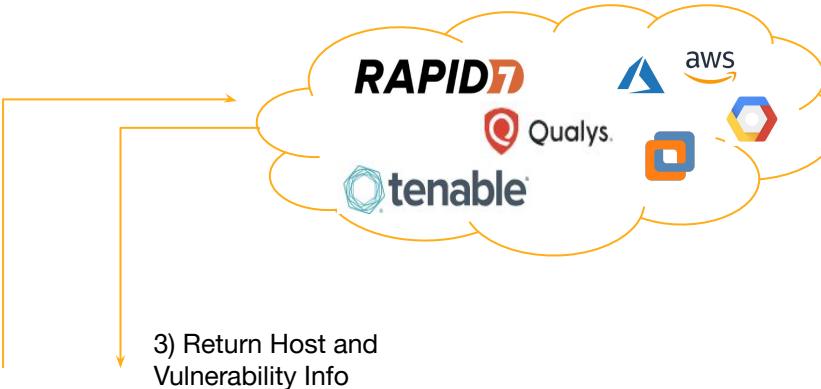


Discovery

- 1) Add Account Details WinRM/SSH Credentials



- 2) Query Providers for Host and Vulnerability data



- 3) Return Host and Vulnerability Info

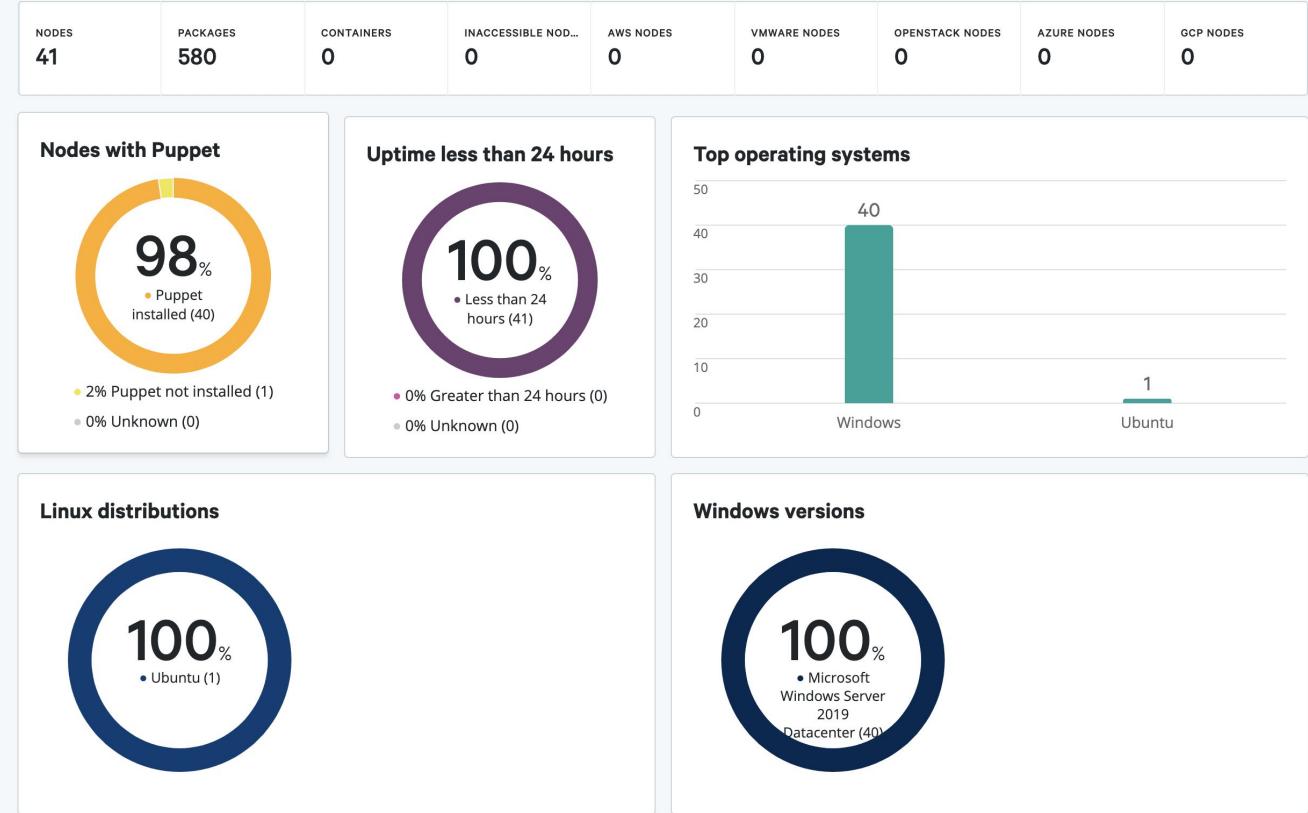
- 4) Query Hosts for Package, User, Service Info



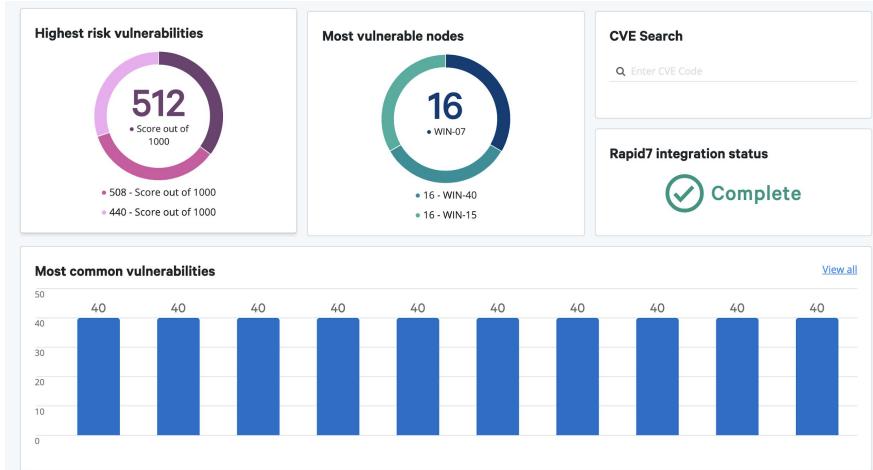
- 5) Return package, user and service info



Exploring the Dashboard



Finding Vulnerabilities



Top vulnerabilities			
1	TLS/SSL Server is enabling the BEAST attack There is no server-side mitigation available against the BEAST attack. The only option is to disable the affected protocols (SSLv3 and TLS 1.0). The only fully safe configuration is to use Authenticated Encryption with Associated Data (AEAD) capable ciphers.	Risk score	512
		Nodes affected	40
2	TLS/SSL Birthday attacks on 64-bit block ciphers (SWEET32) Configure the server to disable support for 3DES suite. For Microsoft IIS web servers, see Microsoft Knowledgebase article 245030 (http://support.microsoft.com/kb/245030/) for instructions on disabling 3DES cipher suites.	Risk score	508
		Nodes affected	40
3	TLS Server Supports TLS version 1.0 Configure the server to require clients to use TLS version 1.2 using Authenticated Encryption with Associated Data (AEAD) capable ciphers.	Risk score	440
		Nodes affected	40
4	TLS Server Supports TLS version 1.1 Configure the server to require clients to use TLS version 1.2 using Authenticated Encryption with Associated Data (AEAD) capable ciphers.	Risk score	403
		Nodes affected	40
5	TLS/SSL Server Supports The Use of Static Key Ciphers Configure the server to disable support for static key cipher suites. For Microsoft IIS web servers, see Microsoft Knowledgebase article 245030 (http://support.microsoft.com/kb/245030/) for instructions on disabling...	Risk score	391
		Nodes affected	40

Working with Tasks

The screenshot shows the Puppet Forge website interface. At the top, there's a search bar labeled "What do you want to automate?" and dropdown filters for "Supported/Approved", "Operating System", and "With Tasks?". Below the search bar, a banner for "bolt" is displayed with the tagline "Automating your infrastructure just got easier." and a "Learn more" link.

The main content area shows a module card for "timidri/meltdown" by Dimitri Tischenko. It includes a profile picture of Dimitri, the module name, and a brief description: "Detect and remediate Meltdown / Spectre vulnerability". There are buttons for "TASKS" and "PDK". Below this, a "Tasks:" section lists "meltdown::windows_update | meltdown::linux_update | meltdown::force_offer_hotfix". A note indicates the latest version is compatible with various Puppet Enterprise versions and specific Linux distributions like RedHat, windows, Ubuntu, Debian, Scientific, CentOS, OracleLinux.

At the bottom, there's a section titled "Start using this module:" with dropdown menus for "new! Bolt", "r10k or Code Manager", and "Manual installation".

The screenshot shows the "Remediate" upload interface. It consists of two main sections: "1 Upload file" and "2 Confirm details".

The first section, "Choose an upload option", contains two radio button options:

- Upload a module**: Adds frequently used tasks to Remediate by uploading a module with Bolt Tasks as a .tar.gz file. It includes a file input field and a "Browse" button.
- Upload a script**: Converts uploaded scripts into tasks. Scripts can be combined into one task to run on both operating systems. It includes fields for "Add Linux, Windows, or both." and separate "Add Linux script" and "Add Windows script" buttons.

Fixing the Vulnerability

Risk score	512
Nodes affected	40
Infrastructure affected	98 %

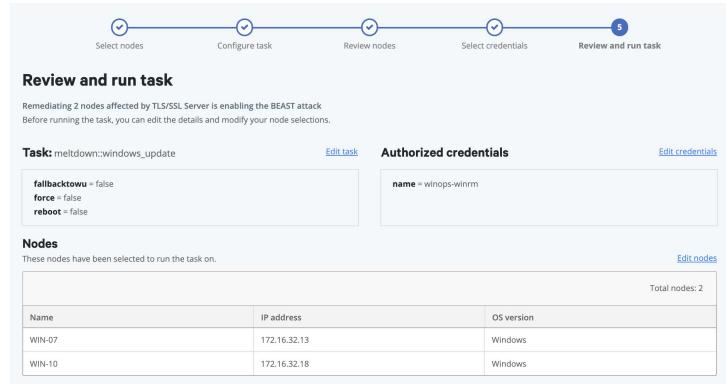
Analysis

The SSL protocol, as used in certain configurations of Microsoft Windows, and browsers such as Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Opera and other products negotiating SSL connections) encrypts data by using CBC mode with chained initialization vectors. This potentially allows man-in-the-middle attackers to obtain plaintext HTTP headers via a blockwise chosen-boundary attack (BCBA) on an HTTPS session, in conjunction with JavaScript code that uses (1) the HTML5 WebSocket API, (2) the Java URLConnection API or (3) the Silverlight WebClient API, aka a "BEAST" attack. By supporting the affected protocols and ciphers, the server is enabling the clients in to being exploited.

Remediation

There is no server-side mitigation available against the BEAST attack. The only option is to disable the affected protocols (SSLv3 and TLS 1.0). The only fully safe configuration is to use Authenticated Encryption with Associated Data (AEAD), e.g. AES-GCM, AES-CCM in TLS 1.2.

Nodes affected	Run task on nodes below	Run Task
Create filter		
Total nodes affected: 40		Export
Resource name	IP address	
WIN-01	172.16.32.6	
WIN-02	172.16.32.7	



Recent events

TOTAL EVENTS	DISCOVERY RUNS	TASKS RUN
5	3	2

All events	All states	Total events: 5
Event	Status	Submission time
Task: Run a shell command	● Succeeded	Sep 21, 2019 7:46 PM
Discover data	● Succeeded	Sep 21, 2019 7:37 PM

Roadmap

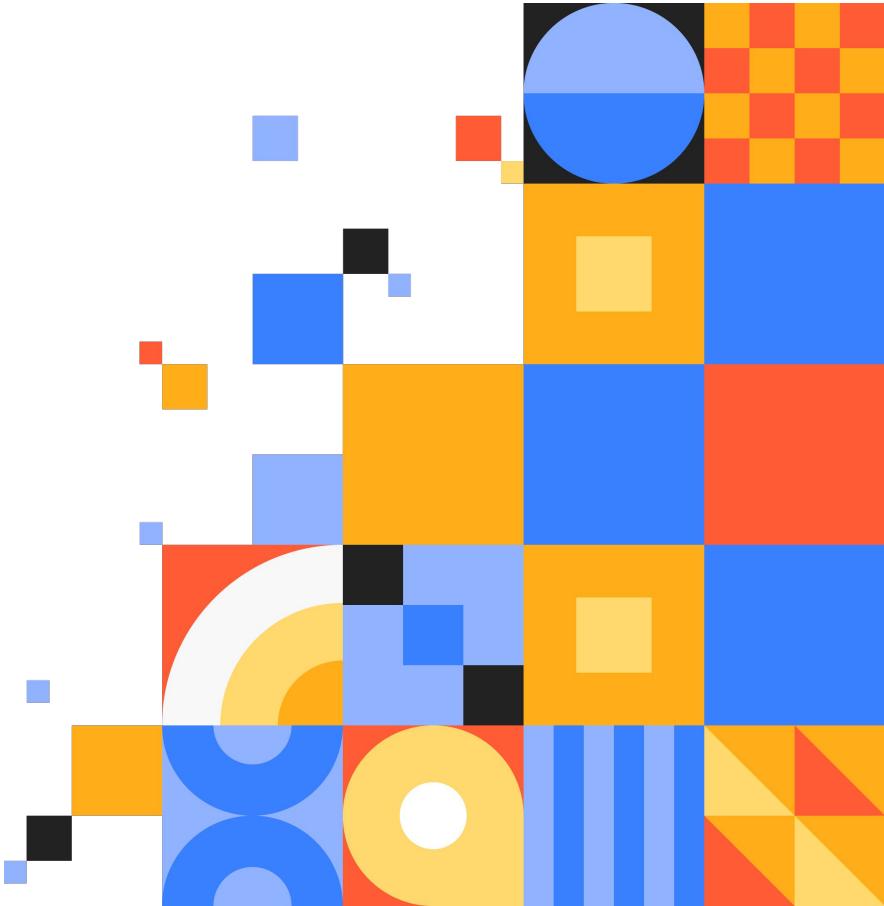
- RBAC - fine grained controls
- Full Audit Logging
- Improved dashboards
- Tracking Fixes

Find out more about Remediate

puppetcommunity.slack.com @DaveJ



Q&A



Stay Connected

github.com/jcoConnor/winops2019



John O'Connor
Senior Software Engineer
john.oconnor@puppet.com



Claire Cadman
Technical Writer
claire.cadman@puppet.com



Dave Johnston,
Senior Software Engineer
dave.johnston@puppet.com