APP100

Configuration Management and Application Weaknesses

LEARNING OBJECTIVES

At the completion of this lecture, students should be able to:

- LO1: Define and utilize configuration and deployment management
- LO2: Discover weaknesses in authorization and authentication

Configuration Management

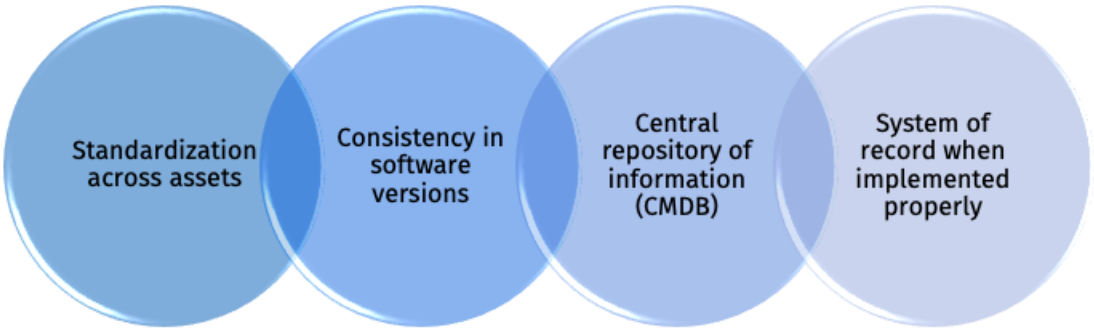
BASIC SECURITY PRINCIPALS

From a security perspective think of the #1 critical security control	Also CIS Control 2 (CIS Critical Security Controls, formerly SANS Top 20):
• Inventory and Control of Hardware Assets	• Inventory and Control of Software Assets

If we don't know what's on the network, we can't defend it

CONFIGURATION MANAGEMENT

Configuration Management (CM) helps us in several ways



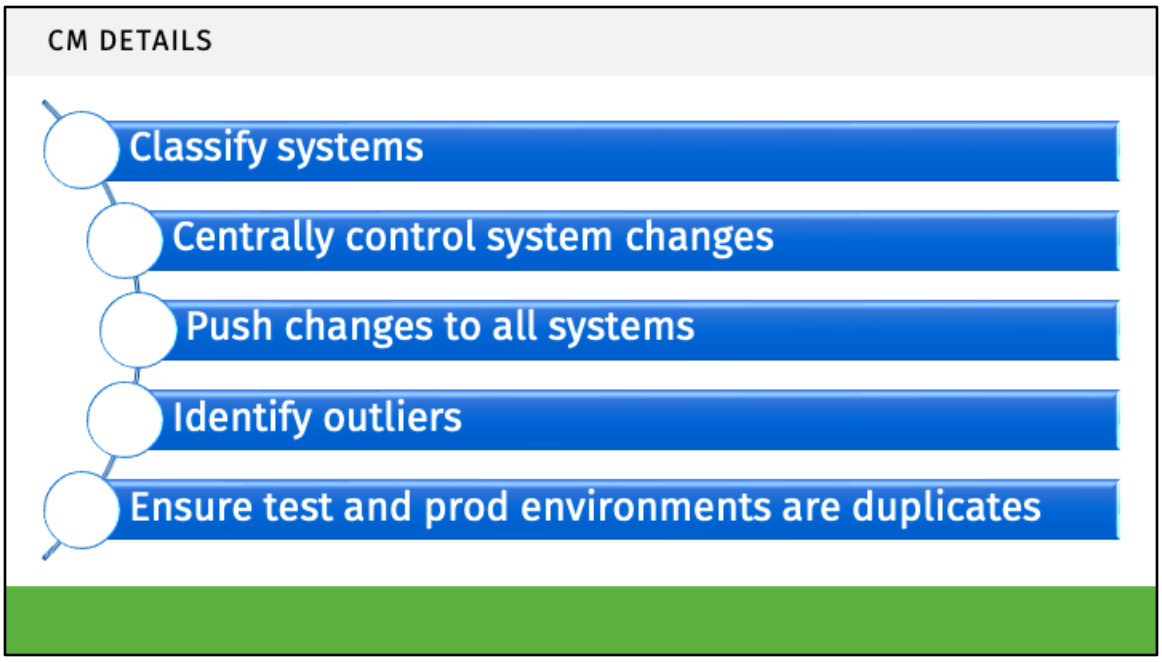
CM AND DEPLOYMENT DETAILS

Why was CM developed?

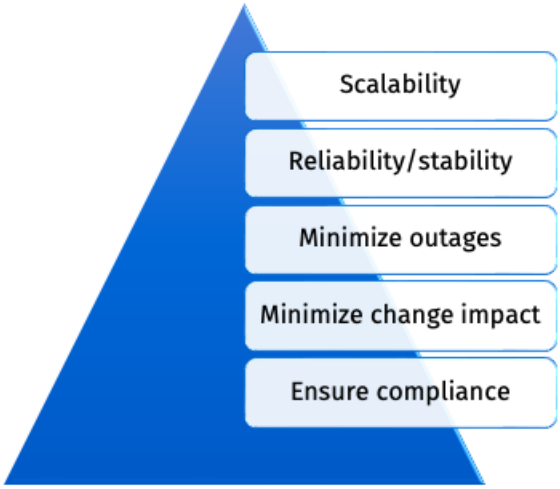
- Needed a way to ensure stability, consistency, and documentation
- Think of a large quantity of systems

Consider the constant change to the environment:

- Patches
- New software releases
- Incidents/Outages
- Employee turnover
- New technology solutions being implemented


CM DETAILS
Classify systems
Centrally control system changes
Push changes to all systems
Identify outliers
Ensure test and prod environments are duplicates

CM BENEFITS



How would this help with security?

CM AND MATURITY

Not all companies embrace CM

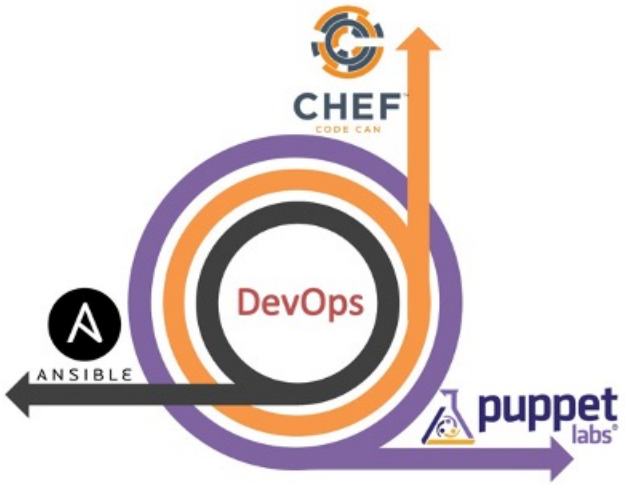
CM can be applied to any organization, regardless of size

Implementation does take a level of expertise and buy-in

CM TOOLS

Here are some of the applicable CM tools:

- Ansible
- Chef
- Puppet
- CFEngine
- SaltStack



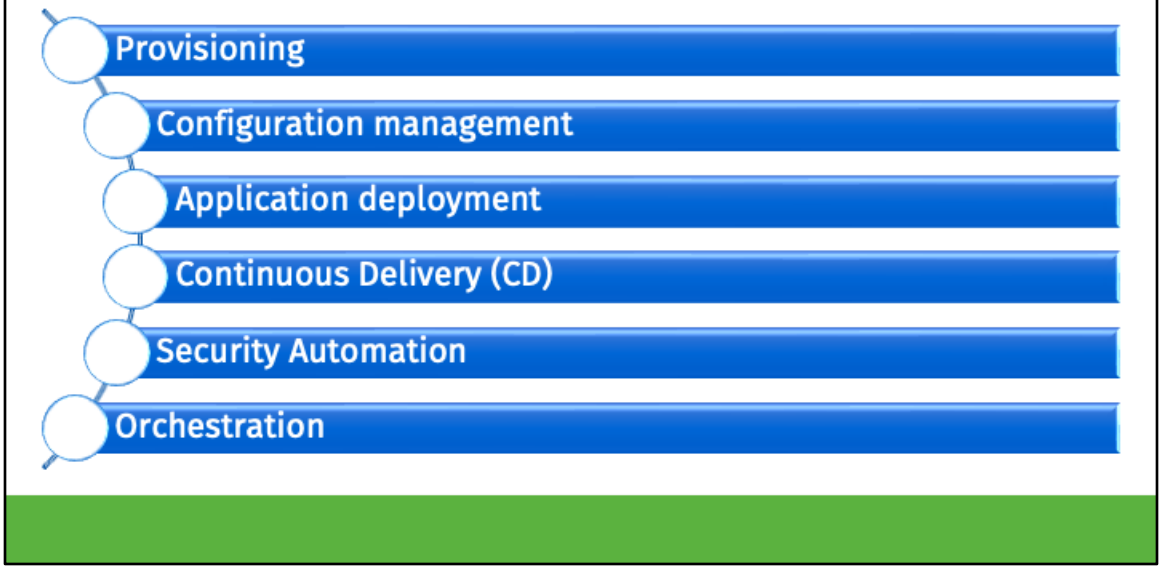
FOCUS ON ANSIBLE

It makes sense to delve into an IT solution in more detail

Take Ansible:

- Red Hat – open-source
- CM and application-deployment tool (as well as orchestration)
- Works on Unix/Linux and similar systems
- Can be used to configure Windows machines as well
- Regularly updated
- Used by many large enterprises

ANSIBLE – WHAT DOES IT DO?



ANSIBLE – HOW DOES IT WORK?

YAML – Playbooks

Uses SSH to connect to nodes/endpoints/machines

- WinRM for Windows connection and APIs for other systems

Pushes out Ansible modules (small programs)

- Complete with tasks we want to accomplish

No database, agents, or servers

Basically, we tell it what machine(s) to talk to, and what we want it to do on those machines

ANSIBLE – SAMPLE PLAYBOOK

```
---
- name: install and start apache
  hosts: web
  remote_user: justin
  become_method: sudo
  become_user: root
  vars:
   http_port: 80
   max_clients: 200

  tasks:
  - name: install httpd
    yum: name=httpd state=latest
  - name: write apache config file
    template: src=srv/httpd.j2 dest=/etc/httpd.conf
    notify:
    - restart apache
  - name: start httpd
    service: name=httpd state=running

  handlers:
  - name: restart apache
```

Application Bypass

AUTHORIZATION AND AUTHENTICATION

How do we gain access to a system?

From an application perspective:

- Authentication: Prove we are who we say we are
- Authorization: privilege – what we are allowed to do

AUTHENTICATION

What do we know?
- Username
- Password
- Q/A (city where you were born)

What do we have?
- Mobile device (Authenticator token, soft token, key, SMS)

What are we (Biometrics)?
- Retina, iris, face
- Voice, DNA
- Fingerprint

AUTHORIZATION

What are we allowed to do?

- What is our user role/level of permission?

Occurs after authentication

We log in, then we can access certain functions of the app

But how does this work?

AUTHORIZATION AND AUTHENTICATION IN ACTION

Typically, the framework handles this

PHP, Java, or .NET

Also, typically done with a cookie

Could be a different session token

https://docs.google.com/spreadsheets/d/1R7BBkoD4fuHJIhVtpB5Z3mPLKDIESAMAT8XtN/edit?pli=1#gid=173

JSESSIONID	"ajax:671380242415...	.www.linkedi...	/	2021-09-28T08:57:40...

SCENARIO FOR TODAY: BYPASS AUTHENTICATION AND AUTHORIZATION

If we can gain access to a valid session token:

- We now share the identity of the valid user
- We are logged in until one of us logs out (or the session expires)
- This requires no username and no password
- We just need to find a flaw with the session token and exploit it

Once we can gain access to the application, the goal is then to escalate our privilege:

- Privilege escalation attacks are powerful attacks
- Change our permissions from regular user -> up to admin rights

So how might we go about doing this?

BYPASS AUTHENTICATION

Identification and Authentication Failures was the #2 security risk in the OWASP top ten in 2017. In 2021, it fell to #7, due to the increased availability of standard frameworks, but is still far too common.

- However, not all broken authentication issues result in bypasses

Need to consider how the application works

Pay close attention to session tokens

Flags on cookies

- Missing Secure Flag
- Missing HTTPOnly Flag

Business logic flaws

ESCALATE PRIVILEGE

To successfully escalate privilege requires a keen eye:

- Pay attention to profile/user editing capabilities
- Always request various test accounts during penetration testing
- Check what pages an admin can access then try using regular user rights
- Check what functions and admin can submit and try to reproduce
- Trick admin into submitting a request on our behalf (CSRF)
- Social Engineering