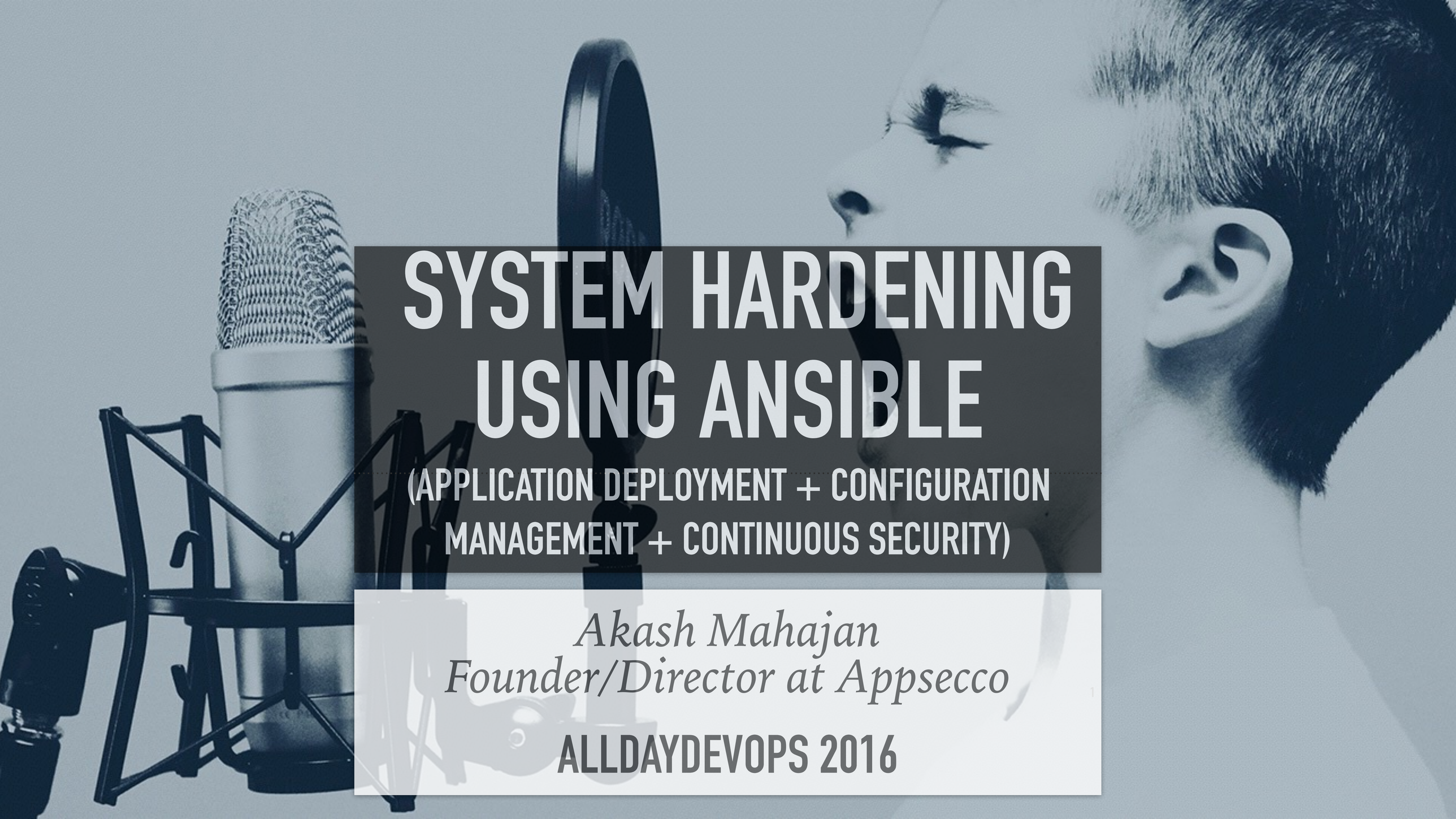


SYSTEM HARDENING USING ANSIBLE

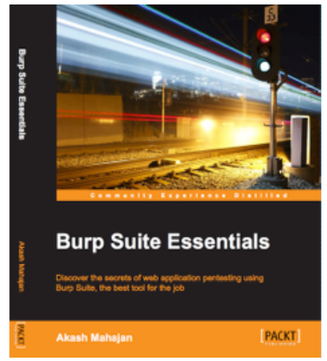
(APPLICATION DEPLOYMENT + CONFIGURATION MANAGEMENT + CONTINUOUS SECURITY)

Akash Mahajan
Founder/Director at Appsecco

ALLDAYDEVOPS 2016

THAT WEB APPLICATION SECURITY GUY

OWASP
Bangalore Chapter



APPSECCO
THE APPLICATION SECURITY COMPANY

XTREME
WEB HACKING

n|u

@makash | https://linkd.in/webappsecguy | akashm.com

“

Start with Why?

- Simon Sinek


THIS IS A STORY ABOUT APPSEC

OWASP TOP 10 – A5 SECURITY MISCONFIGURATION

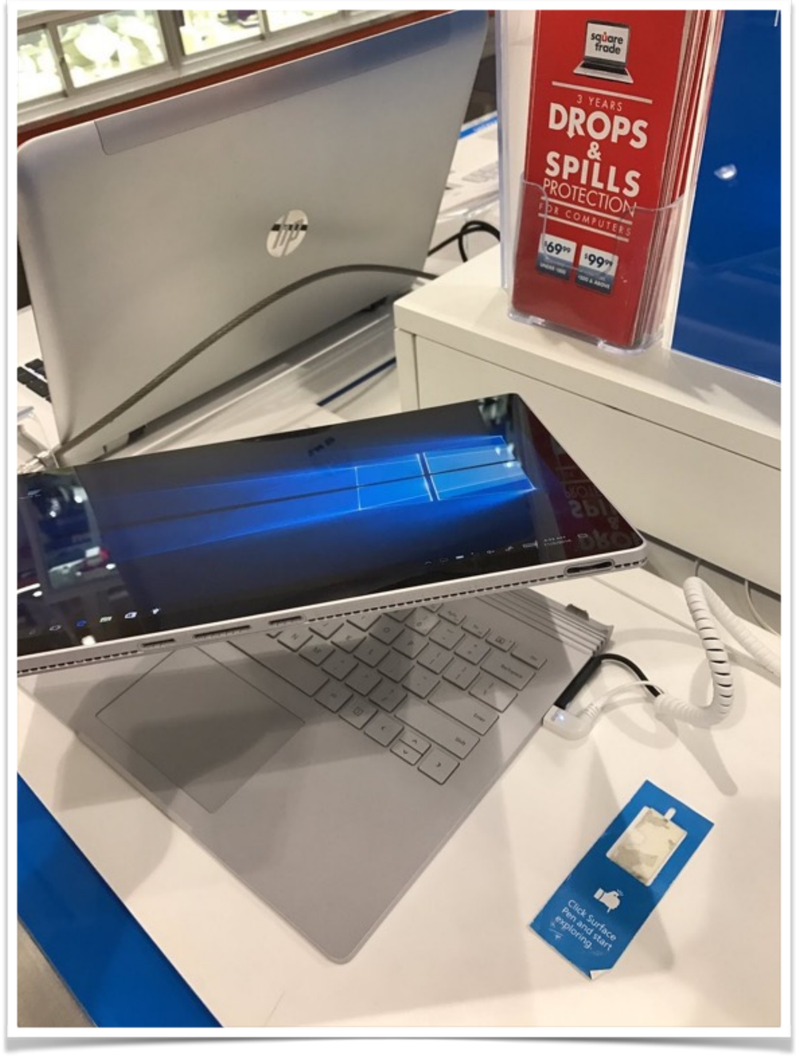
http://cheezburger.com/4834698752

AM I VULNERABLE TO 'SECURITY MISCONFIGURATION'?

- ☐ Is any of your software out of date?
- ☐ Are there any un-necessary features enabled/installed?
  - ☐ Ports, Services, Accounts, Pages, Privileges
- ☐ Are default accounts and their passwords enabled/ unchanged?
- ☐ Are security settings and libraries not set to secure values?

https://www.owasp.org/index.php/Top_10_2013-A5-Security_Misconfiguration

OWASP TOP 10 – A5 SECURITY MISCONFIGURATION



https://twitter.com/SwiftOnSecurity/status/793976943276265472

EXAMPLE ATTACK SCENARIOS

- Attacker discovers the standard admin pages are on your server, logs in with default passwords, and takes over.
- Attacker finds due to directory listing and downloads all your compiled Java classes, which she decompiles and reverse engineers to get all your custom code. She then finds a serious access control flaw in your application.
- App server configuration allows stack traces to be returned to users, potentially exposing underlying flaws. Attackers love the extra information error messages provide.

https://www.owasp.org/index.php/Top_10_2013-A5-Security_Misconfiguration

OWASP TOP 10 – A5 SECURITY MISCONFIGURATION

http://www.failking.com/41473-security-fail.html

PREVENTING SECURITY MISCONFIGURATION

- A *repeatable hardening process* that makes it fast and easy to deploy a properly locked down environment
  - Dev/QA/Prod should be *configured identically* but with different passwords used
  - This process *should be automated* to minimise the effort required to setup a new secure environment.
- A process for deploying all new *software updates and patches* in a timely manner to each deployed environment
- Consider *running scans and doing audits periodically* to help detect future misconfigurations or missing patches.

https://www.owasp.org/index.php/Top_10_2013-A5-Security_Misconfiguration

OUR SECURITY REQUIREMENTS DERIVED (0/5)

- ☐ A repeatable hardening process
- ☐ Dev/QA/Prod should be configured identically but with different passwords used
- ☐ This process should be automated to minimise the effort required to setup a new secure environment.
- ☐ A process for deploying all new software updates and patches in a timely manner to each deployed environment
- ☐ Consider running scans and doing audits periodically to help detect future misconfigurations or missing patches

Basically taken from https://www.owasp.org/index.php/Top_10_2013-A5-Security_Misconfiguration

Deploying software once may not be rocket science,
but doing that repeatedly eliminating human error is

Satellite deploying solar panels - From Wikipedia

HOW DO WE DEPLOY SOFTWARE, APPS & CODE?

CUSTOM BASH SCRIPTS

```
NAME
       rsync - faster, flexible replacement for rcp

SYNOPSIS
       rsync [OPTION]... SRC [SRC]... DEST

       rsync [OPTION]... SRC [SRC]... [USER@]HOST:DEST

       rsync [OPTION]... SRC [SRC]... [USER@]HOST::DEST

       rsync [OPTION]... SRC [SRC]... rsync://[USER@]HOST[:PORT]/DEST
```

- rsync
- ssh/scp
- FTP
- curl/wget

CUSTOM PROGRAMS

```
# Create a directory
sudo("mkdir /var/www")

# Create a directory as another user
sudo("mkdir /var/www/web-app-one", user="web-admin")

# Return the output
result = sudo("ls -l /var/www")
```

```
// Use gulp-rsync to sync the files
return gulp.src(rsyncPaths)
.pipe(gulpif(
    argv.production,
    prompt.confirm({
      message: 'Heads Up! Are you SURE you want to push to PRODUCTION?',
      default: false
    })
))
.pipe(rsync(rsyncConf));

});
```

PROVISIONING TOOLS

Many others as well

PROS AND CONS OF THE APPROACHES

CUSTOM BASH SCRIPTS

- GUI tools discourage automation
- For folks like me custom scripts are inherently difficult to
  - maintain,
  - track
  - reuse

CUSTOM PROGRAMS

- Great for programmers and devs
  - As custom as it can get
- Non-programmers find it difficult
- Overhead of a programming language and syntax

PROVISIONING TOOLS

- Meant for provisioning and deploying code, software & applications
- Automation is a primary objective
- Allows for repeatability in deployment
- Reduces human errors

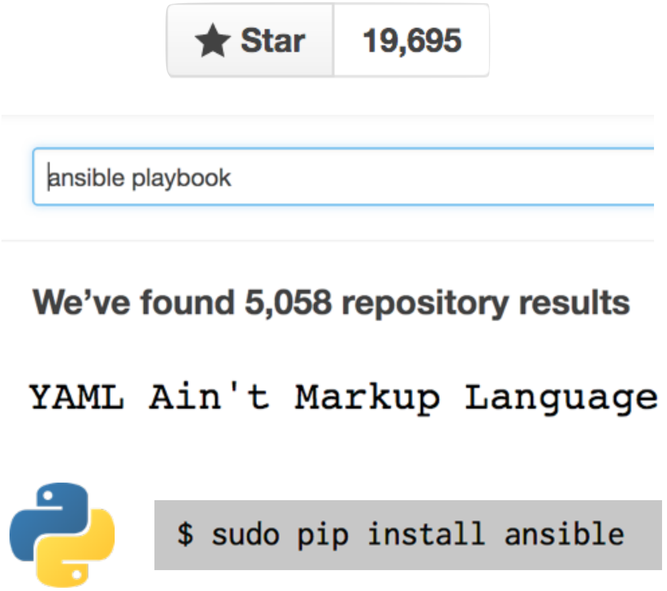
WHAT IS SECURITY HARDENING?

Security hardening is the process where we identify insecure default configuration present on a system and apply changes that will change the configuration to secure values.

The process can be applied to all the layers

- ☐ Network - Enable firewall/security groups with restrictive rule sets
- ☐ Transport - Enable TCP wrappers for a service/subnet matching
- ☐ Application - Enable web server to allow specific IPs to admin panel
- ☐ Kernel Networking parameters - Enable defences for the networking stack

WHY USE ANSIBLE FOR SECURITY HARDENING?



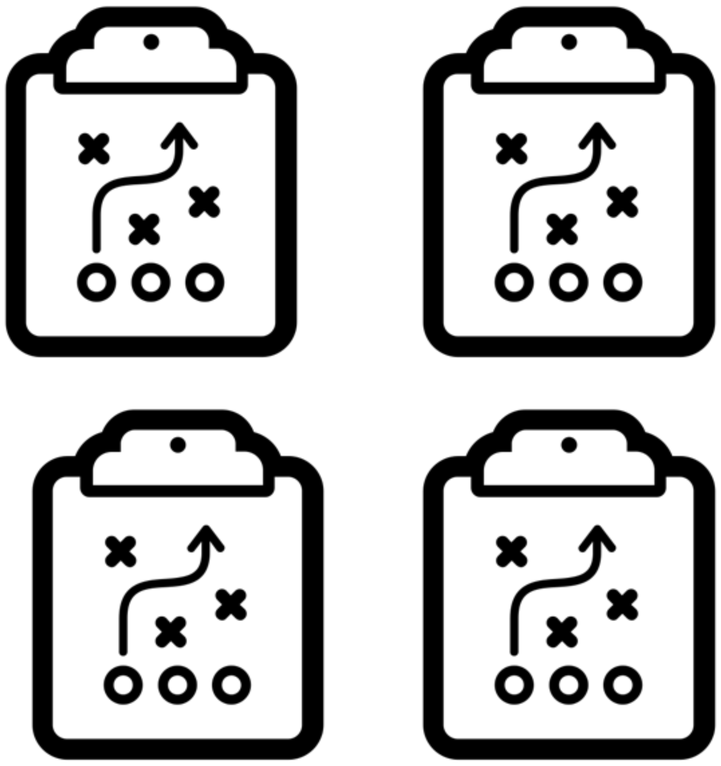
ANSIBLE IS MADE FOR SECURITY AUTOMATION

Attribute	Benefit
☑ YAML language	Provides a structured way to define applications, systems
☑ Modular	Makes it deployment friendly
☑ Enables Automation	Makes it easy to script, program
☑ Uses SSH for access	Secure by default with encrypted transmission and host authentication
☑ Python FOSS	Easy to integrate and get started
☑ Community Driven	Lots of helpful samples and documentation available

- *playbook by Nick Bluth from the Noun Project*
- *github stargazers, ansible search results*

ANSIBLE PLAYBOOK + IDEMPOTENT == WIN

ANSIBLE PLAYBOOK CAN BE A CODIFIED SECURITY DOCUMENT



- [ ] Ansible uses playbooks to execute a series of commands/modules on the target
- [ ] An Ansible playbook is written in YAML which makes it machine readable and provides structure
- [ ] Ansible follows the concept of idempotent, which translates into describing the state that we would like the system to be in
- [ ] All we need to do is express our security assertions in the YAML format in a playbook and we get a codified security document

ANSIBLE PLAYBOOK SNIPPET – MYSQL HARDENING

```
---
- name: Deletes anonymous MySQL server user for localhost
1 mysql_user: user="" state=absent login_password={{ mysql_root_password }} login_user=root

- name: Secures the MySQL root user
2 mysql_user: user="root" password="{{ mysql_root_password }}" host="{{ item }}"
    login_password="{{mysql_root_password}}" login_user=root
  with_items:
    - 127.0.0.1
    - localhost
    - ::1
    - "{{ ansible_fqdn }}"

- name: Removes the MySQL test database
3 mysql_db: db=test state=absent login_password="{{ mysql_root_password }}" login_user=root
```

1. Delete anonymous MySQL user
2. Change MySQL root user password
3. Remove test database

THE CONCEPT OF IDEMPOTENCY

OUR JOB IS NOW TO ENSURE THAT WE NEED TO DEFINE WHAT CONSTITUTES A SECURE AND HARDENED SYSTEM

```
db=test state=absent
```

```
- name: installing nginx server
  apt: name={{ item }} state=present

  with_items:
    - nginx
    - nginx-common
    - nginx-full
```

The concept that change commands should only be applied when they need to be applied, and that it is better to describe the desired state of a system than the process of how to get to that state

http://docs.ansible.com/ansible/glossary.html#term-idempotency

STRUCTURED MANUALS (PLAYBOOKS) + GIT == WIN

VARIABLES ALLOW FOR CREATING GENERIC INSTRUCTION MANUALS

help by Viktor Vorobyev from the Noun Project

repository by Nick Bluth from the Noun Project

secure document by Creative Stall from the Noun Project

- ☑ All playbooks are written in YAML providing us with structure that we can learn and train on
- ☑ Since playbooks are text files, we can use Git to do version control on them
- ☑ By using Git or another version control software, managing the playbooks is just like managing any software project.
- ☑ Therefore *infrastructure as code* but for security

OUR SECURITY REQUIREMENTS DERIVED (2/5)

- ☐ A repeatable hardening process
- ☑ Dev/QA/Prod should be configured identically but with different passwords used
- ☑ This process should be automated to minimise the effort required to setup a new secure environment.
- ☐ A process for deploying all new software updates and patches in a timely manner to each deployed environment
- ☐ Consider running scans and doing audits periodically to help detect future misconfigurations or missing patches

Basically taken from https://www.owasp.org/index.php/Top_10_2013-A5-Security_Misconfiguration

HOW DO WE CREATE SECURITY BEST PRACTICES?

YOU DON'T NEED TO, BEST PRACTICES HAVE ALREADY BEEN CREATED

Dahi Handi by Ramnath Bhat under CC2.0 license

https://www.flickr.com/photos/ramnath1971/7943196628

Various organisations publish best practices

- ☑ CIS Benchmarks
- ☑ DISA-STIG
- ☑ NIST Guidelines
- ☑ Linux Distribution specific guidelines
- ☑ Application security specific guidelines

ANSIBLE PLAYBOOK IS MADE UP OF ROLES

ROLES CAN EASILY BE ADDED TO A PLAYBOOK FOR MAXIMUM FLEXIBILITY

```
roles/
├── database
│   ├── defaults
│   │   └── main.yml
│   ├── handlers
│   │   └── main.yml
│   └── tasks
│       ├── dbimport.yml
│       ├── main.yml
│       └── mysql_security.yml
├── php
│   ├── handlers
│   │   └── main.yml
│   └── tasks
│       └── main.yml
├── webserver
│   ├── handlers
│   │   └── main.yml
│   ├── tasks
│   │   ├── main.yml
│   │   └── nginxconfig.yml
│   └── templates
│       ├── nginx-appsecco.com.conf.j2
│       ├── nginx-default.conf.j2
│       ├── nginx-sta-apsc-co-conf.j2
│       └── wordpress.conf.j2
```

Ansible Roles are the moving parts of a playbook

☑ Roles are how we should be organising a playbook

☑ Grouping content by roles allows easy sharing of roles with other users

☑ By using roles_path configuration variable, roles can be downloaded from git, Ansible Galaxy and stored in one location, to use with multiple playbooks

WHERE DO WE FIND REFERENCE ANSIBLE PLAYBOOKS

GREAT NEWS IS THAT THERE ARE MANY HARDENING PROJECTS ALREADY

Notable projects to get started with, right now

- ☑ Hardening Framework - Server Hardening Framework
- ☑ Ansible role for DISA STIG
- ☑ OpenStack-Ansible - Host Security Hardening
- ☑ CIS Ansible Role against CentOS/RHEL
- ☑ Linux Security Hardening with OpenSCAP and Ansible
- ☑ First Five Minutes on a Server with Ansible

ANSIBLE GALAXY IS LIKE GITHUB BUT FOR ROLES

GALAXY IS NOW OSS, SO THAT YOU CAN SETUP PRIVATE GALAXY SERVERS

```
$ ansible-galaxy \
search hardening
```

```
$ ansible-galaxy  \
install
username.rolename
```

Galaxy is an online tool to manage Ansible roles

☑ Using the CLI client, roles can be searched for and installed with just one command

☑ Galaxy is like the central repository information for roles

☑ Galaxy offers automated testing of roles as well

OUR SECURITY REQUIREMENTS DERIVED (3/5)

☑ A repeatable hardening process

☑ Dev/QA/Prod should be configured identically but with different passwords used

☑ This process should be automated to minimise the effort required to setup a new secure environment.

☐ A process for deploying all new software updates and patches in a timely manner to each deployed environment

☐ Consider running scans and doing audits periodically to help detect future misconfigurations or missing patches

Basically taken from https://www.owasp.org/index.php/Top_10_2013-A5-Security_Misconfiguration

CONTINUOUS MONITORING FOR SECURITY

ANSIBLE CAN BECOME PART OF YOUR CI/CD WORKFLOW

go

ANSIBLE TOWER by Red Hat

RUNDECK

- ☑ Integrate with your favourite CI/CD tool
- ☑ Schedule regular runs against the targets as specified
- ☑ Get information on when your run (build) failed and why
- ☑ Get granular control to secure credentials and secrets and get Role Based Access Control (RBAC) as well

Jenkins logo from https://jenkins.io/ Go.cd logo from https://go.cd

Ansible Tower logo from https://ansible.com Rundeck logo from https://xebialabs.com

OUR SECURITY REQUIREMENTS DERIVED (5/5)

☑ A repeatable hardening process

☑ Dev/QA/Prod should be configured identically but with different passwords used

☑ This process should be automated to minimise the effort required to setup a new secure environment.

☑ A process for deploying all new software updates and patches in a timely manner to each deployed environment

☑ Consider running scans and doing audits periodically to help detect future misconfigurations or missing patches

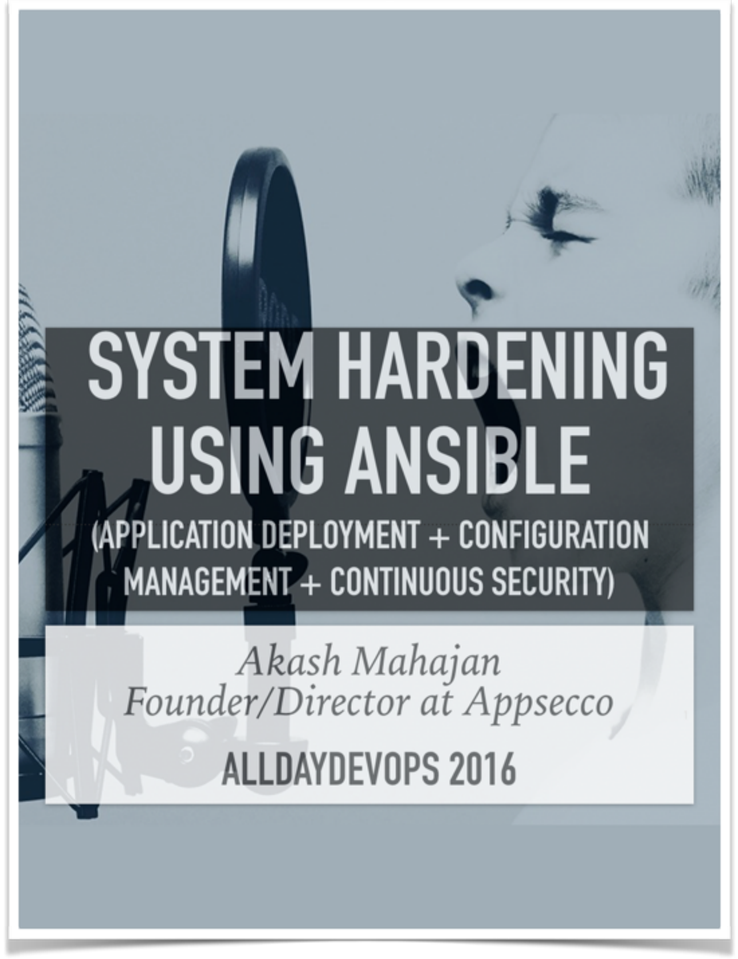
Basically taken from https://www.owasp.org/index.php/Top_10_2013-A5-Security_Misconfiguration

TAKEAWAYS AND CONCLUSION

1. Using Ansible (and others) we can build a security automation workflow
2. Since the security part is codified in documents, we can do version control
3. A lot of work has already been done in finding out the best practices
4. For Ansible, using the above mentioned best practices, there are already multitude of playbooks and roles available on github and Ansible Galaxy
5. Using CI/CD tools like Jenkins/Go.cd or specialised software like Ansible Tower/ Rundeck we can repeatedly schedule Ansible playbooks and monitor their outcome

BONUS TAKEAWAY – FREE EBOOK

https://github.com/appsecco/alldaydevops-shua



- ☑ Ebook in PDF/Mobi/Epub format
- ☑ Will keep it updated and add more integrations
- ☑ Available with the presentation and other materials at the above mentioned github repo

COMPANIES WHO MADE THIS EVENT POSSIBLE

Sponsors for All Day DevOps 2016

Sonatype

SAUCELABS

Signal Sciences

cloudbees

XebiaLabs

SUPPORTERS OF ALL DAY DEVOPS

QUESTIONS

@makash | https://linkd.in/webappsecguy | akash@appsecco.com

APPSECCO

THE APPLICATION SECURITY COMPANY