# Change & Configuration Management

Project Name: Operations Security & Project

Report Written by Victor Poh Hong Rong

Supervised by Mr Sim Xiangyuan

DSF-1802, Team 2

180272P

Team Members:

| | | |
|---|---|---|
| • Lye Jia Jun (Group Leader) | 180245D | Continuous Monitoring |
| • Tan Song Ze Nigel | 181391W | BCP & Backup |
| • Muhammad Noor Bin Saidee | 185014T | Network Security |
| • Victor Poh Hong Rong | 180272P | Change & Config Management |

Reported last amended: 15/8/2020

# Contents

Change and Configuration Management ........ 3

Security Architecture & Measures ........ 4

Industry Best Practices / Guidelines ........ 5

Systems Integration ........ 7

Advanced features displaying innovative technologies ........ 8

Test Scenarios ........ 12

# Change and Configuration Management

The purpose of IT Change Management is to prevent unintended consequences and ensures that changes or alterations to systems are implemented according to an approved framework or model.

Configuration Management focuses on establishing and maintaining consistency of a product’s performance, and its functional and physical attributes with its requirements, design, and operational information throughout its life.

Real-life problems faced by an organization in the change and configuration management sector includes:

- Not having an appropriate Change management process (i.e. new ideas are not properly communicated to the stakeholders, owners, and employees of the organization)
- Developers setting up software without proper documentation/traceability
- Employees cannot cope with the implementation of an Agile process
- No consistency in configurating changes (i.e. one person has to redo and refollow setup procedures for tens or hundreds or even thousands of machines)

These problems delay and hinder the change and configuration management process in an organization. Therefore, for my project and as a change and configuration manager, I would like to tackle these problems by setting up an appropriate change and configuration management process, with the help of my knowledge, online resources, as well as software solutions in the market that balance costs, risks, resources and benefits.

# Security Architecture & Measures

The main and supporting features that I will be implementing for my project are as follows:

- Ensuring that all Configuration changes are **properly documented and traced**
- Enabling Configuration changes to be **simultaneously pushed** to multiple machines
- **Automating** the Change and Configuration Management process (to save time and manpower)
- Ensuring that all Changes (which can be any type of changes) to systems in the organization are **tracked**

Software/Systems involved:

1. Ansible (Open-source software provisioning, configuration management, and application-deployment tool enabling Infrastructure as Code)

2. ManageEngine AD Audit Plus (on-premise auditing solution)

3. Change Request Form (self-coded using C# WPF)

4. Ubuntu VM (Change & Configuration Server, a.k.a. Ansible Control Machine)

5. The rest of the organization systems involved also include
    - Windows Server 2016 (Active Directory Domain Controller)
    - Windows Server 2016 (Victim Server)
    - Windows Server 2016 (Backup Server)
    - Centos (Migration Server)

## Note:

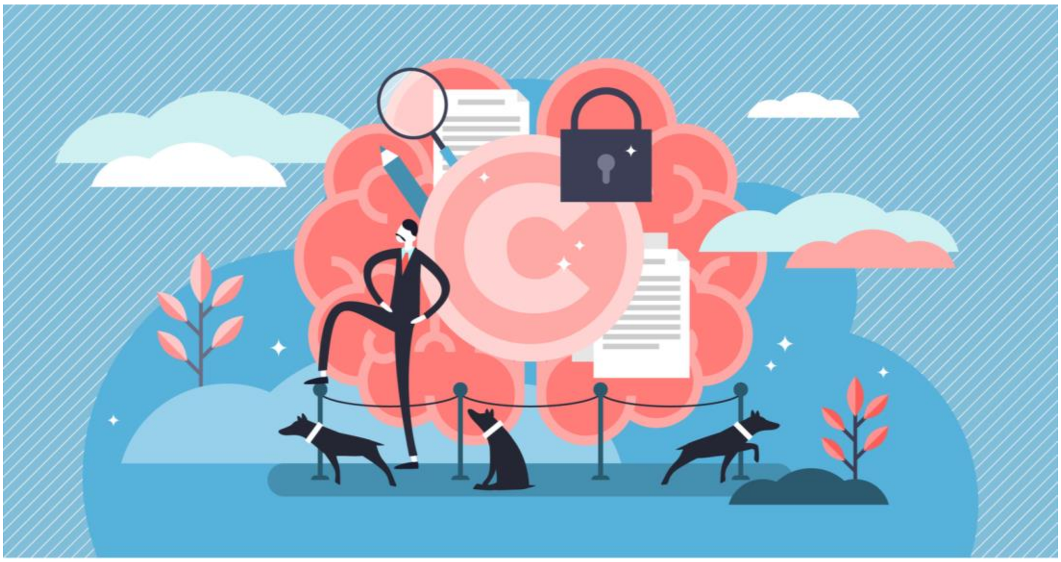
The Change Request Form contains 3 parts:

- General Change Request Form
- Software Request Form
- Policy Request Form

## Industry Best Practices / Guidelines

The “Change Request Form” is aligned to the following standards:

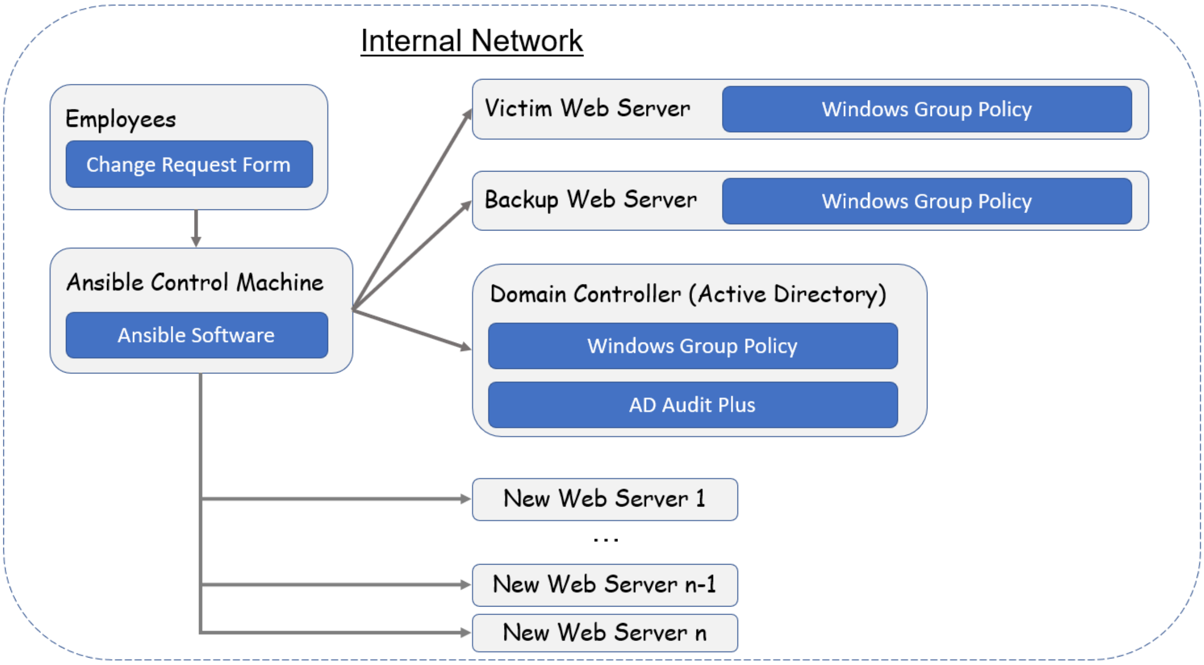
1. ISO/IEC 38500:2008, Corporate governance of IT
2. ISO/IEC 31000:2018, Risk Management



For more best practices, refer to pages 53-61 of “Appendix Test Results.docx”

# Security Architecture & Measures

Before I move on to my documentation, I would like to present the network/system architecture for my part in the project. The following architecture shows the persons/systems involved as well as the installed software/policies in the systems.



Brief Description of the above Architecture as well as the functionalities in my project

1. For the main functionality, I am using the "Change Request Form", "Ansible Control Machine" (with "Ansible Software" installed), as well as the "Victim Web Server" for my demo.

2. For the "Change Request Form", there are three parts to it, with them being the "General Change Form", "Software Request Form", as well as the "Policy Request Form".

3. "General Change Form" allows for an orderly and "neat" change management process.

4. What actually happens in the main functionality is that,

    a. in a scenario where an employee finds it necessary to install a software into the respective machines in the organization, the employee will be using the "Software Request Form" to insert the name(s) of the software he wants to install, as well as the supporting reasons for such a decision. Upon approval by senior management, the "Software Request Form" connects to the "Ansible Control Machine" which is the Change & Configuration (C&C) Server, and instructs the C&C server to install the necessary software into the respective machines in the organization. The connection and installation process is automated (which means no human intervention is required).

b. in a scenario where an employee finds it necessary to make the necessary policy changes in the respective machines in the organization, the employee will be using the “Policy Request Form” to insert the policy name as well as the change to be made. Upon approval by senior management, the “Policy Request Form” connects to the “Ansible Control Machine” which is the Change & Configuration (C&C) Server, and instructs the C&C server to make the necessary policy changes into the respective machines in the organization. The connection and installation process is automated (which means no human intervention is required). That summarizes the main functionality that I will be doing for my project.

5. For the supporting functionality, I am using the “Domain Controller (Active Directory)” (with “AD Audit Plus” installed), to connect to the “Victim Web Server” and “Backup Web Server” for my demo.

6. What actually happens in the supporting functionality is that, in a scenario where a person (administrator/authorized/unauthorized) makes changes to the systems in the organization, such changes will be detected and recorded in ManageEngine AD Audit Plus, so that ultimately, all changes can be tracked.

# Systems Integration

1. Please refer to pages 62-65 of “Appendix Test Results.docx” and pages 40-43 of “Appendix Installation and Configuration.docx” for the documentation on the integration with Splunk. (Integration with Jia Jun)

2. Please refer to pages 66-67 of “Appendix Test Results.docx” and pages 44-45 of “Appendix Installation and Configuration.docx” for the documentation on the integration with pfSense. (Integration with Noor)

3. Please refer to pages 46-47 of “Appendix Installation and Configuration.docx” for the documentation on the integration with Automated Backup
(Integration with Nigel

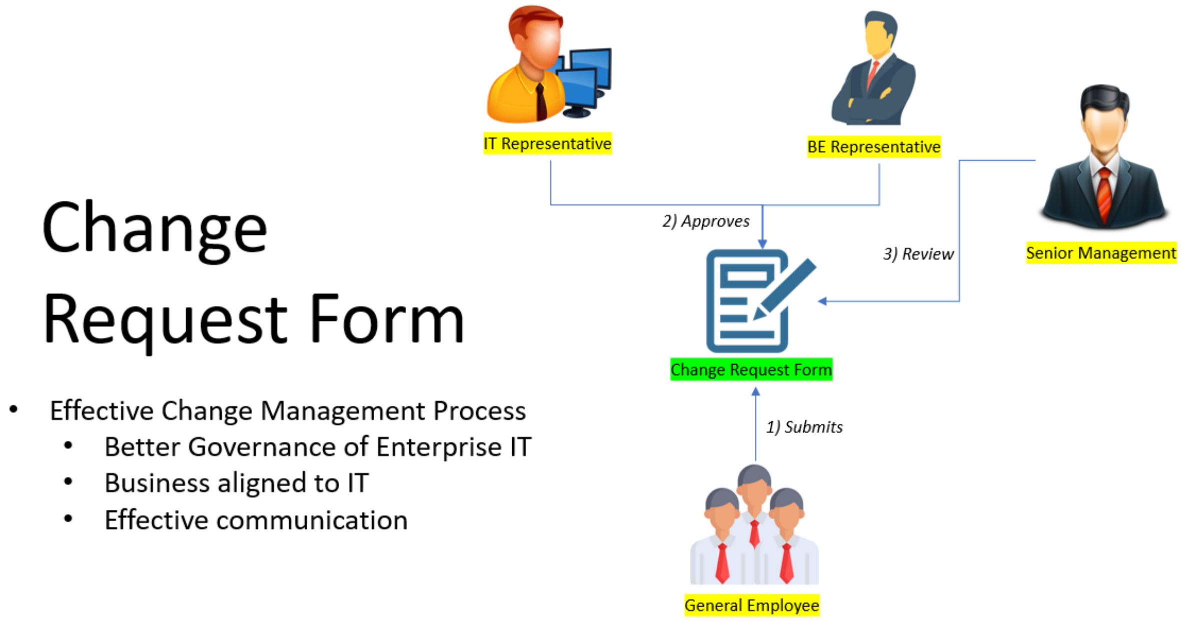
# Advanced features displaying innovative technologies

Before I move on to explain the advance features displaying innovative technologies, I would like to briefly describe the functions of both

- Change Request Form and
- Ansible Software

*The "Change Request Form" is an application that employees working in the organization can use for effective and centralized communication of changes throughout the enterprise.*

*The workflow of the change request form is as follows:*



Change Request Form

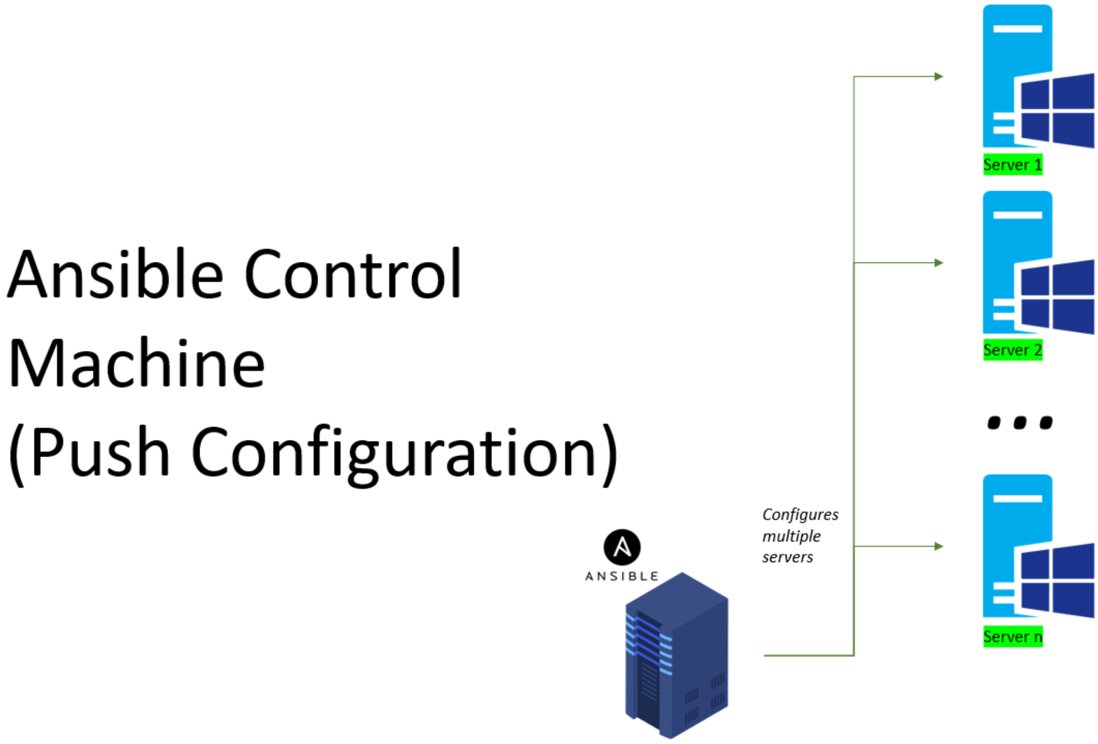
- Effective Change Management Process
  - Better Governance of Enterprise IT
  - Business aligned to IT
  - Effective communication

# Advance features displaying innovative technologies

*Ansible is an open-source software provisioning, configuration management, and application-deployment tool enabling infrastructure as code. It runs on many UNIX-like systems and can configure both UNIX-like systems as well as Microsoft Windows.*

*Ansible is a push-configuration management type of configuration management tool.*

*The workflow of Ansible is as follows:*

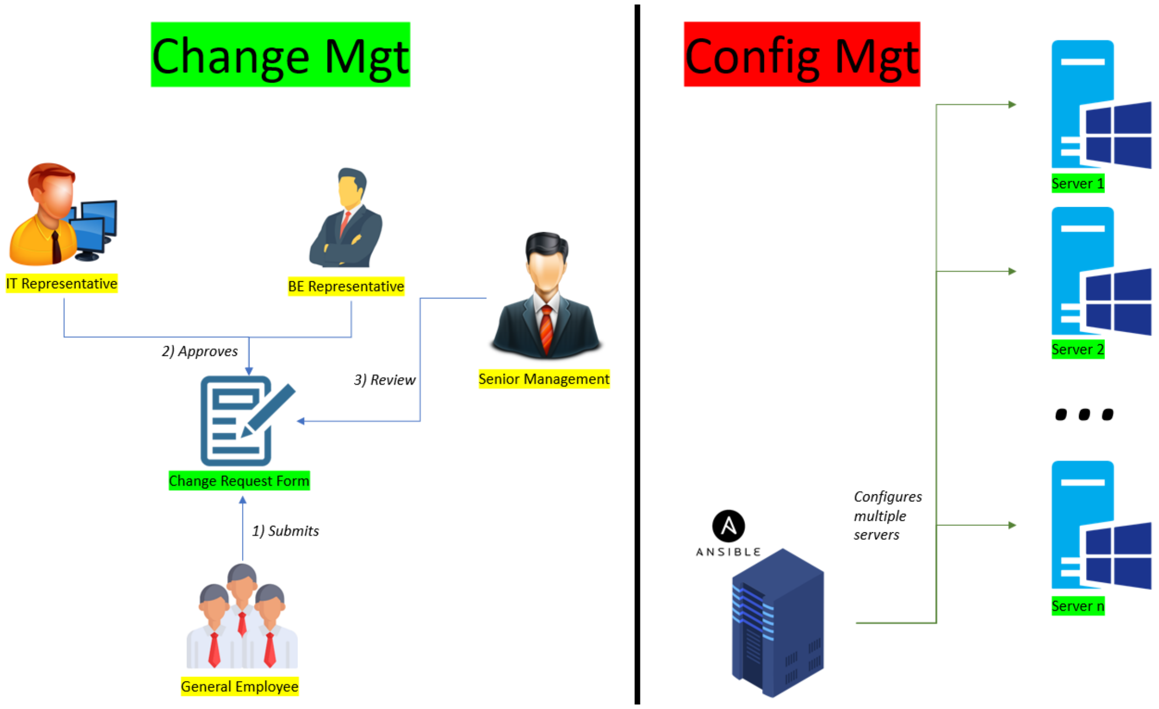


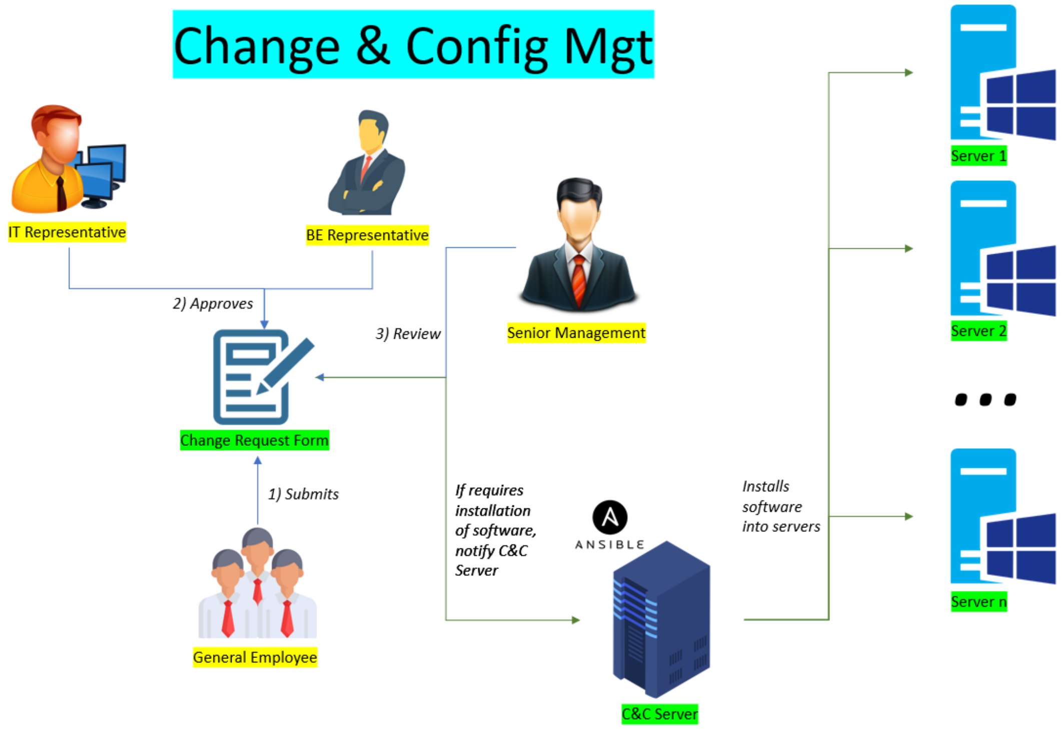
Since the “Change Request Form” supports an effective **change management process**, and “Ansible” supports an effective **configuration management process**, I thought that it was something innovative if I could combine my change request form to work with Ansible to ensure that the **change and configuration management process** can be effective, non-technical, require almost no manpower for the configuration management process, and communicated throughout the enterprise.

# Advance features displaying innovative technologies

## Advanced features displaying innovative technologies – Ensuring Automation & Interoperability

Connecting “Change Request Form” with “Ansible Software”





# Advance features displaying innovative technologies

Why Connect the “Change Request Form” with the “Ansible” Software?

## High-level description of the benefits

- Ensuring that all changes (including configuration changes) follows the following standards
  - ISO/IEC 38500:2008, Corporate governance of IT
  - ISO/IEC 31000:2018, Risk Management

- Allowing the “Change Request Form” to be the **core** of all changes
  - Effective communication among Employees, IT and BE (Business) representatives, as well as Senior Management
  - Change Request Form connects to the C&C server to push configuration into target servers
    - This process is therefore non-technical, automated, simultaneous, parallel, and non-time-consuming

## Detailed (i.e. low-level) description of the benefits

- Effective Change Management Process
  - Better Governance of Enterprise IT
  - Business aligned to IT
  - Effective communication

- Save manpower, cost, time and efforts
  - Automation of change & configuration management process

- Centralized change request logs
  - Changes can be tracked easily
  - New employees can appreciate the current situation in the organization

- Less susceptible to attack
  - Change Request Form application only requires port 1433 to be open (and doesn’t require port 80)
    - Port 80 is well-known for launching web-based-application attacks
    - Less port exposed = Less vulnerability = Lower chance of exploitation

# Test Scenarios

High-level description

1. Detecting configuration/file changes, e.g. when administrator or authorized/unauthorized personnel make changes to system configuration/file, these should be detected.

2. Detecting a need for a change to the organization’s security policy, systems, procedures, installation of additional software etc., through employees’ participation and by using systems automation. In a scenario where for example, the company wants to change 8 Servers from Windows platform to Linux platform, and also have web server/service installed in the 8 new Linux Servers (with the required webpage also installed and run in the 8 Linux Servers). The company wants this process to be **simultaneous, parallel and automated**.

3. Penetration testing on the application that is being used to support the Change & Configuration Management Process.

4. Detecting certain ports or services are being enabled/disabled through the C&C server.

5. Ensuring that authorized connections are allowed between the C&C server and the authorized machines, and rejecting any unauthorized connection.

# Test Scenarios

| Test Scenarios | ID | Test Case Names |
|---|---|---|
| Detecting configuration changes | #1 | A person (administrator/authorized/unauthorized) make changes to system configurations |
| Detecting file changes | #2 | A person (administrator/authorized/unauthorized) make changes to system file/folder |
| Automate the policy request & software installation process | #3 | Employee requests change through General Change Request Form |
| | #4 | Employee requests installation of software through Software Request Form |
| | #5 | (Continued from Test Case #4) Automate software installation |
| | #6 | (Alternate Flow of Test Case #5) Install already installed software into target server(s) |
| | #7 | Employee requests policy changes through Policy Request Form |
| | #8 | (Continued From Test Case #7) Automate policy changes |
| | #9 | Employee requests system updates through Policy Request Form |
| | #10 | (Continued from Test Case #9) Automate policy changes |
| | #11 | Employee requests configuration changes (e.g. disabling domain firewall) through Policy Request Form |
| | #12 | (Continued from Test Case #11) Automate configuration changes |
| | #13 | Simultaneously install web server/service and deploy required web pages into 8 new Linux Servers |
| System hardening | #14 | A person (penetration tester/unauthorized hacker) exploits misconfigurations in target application |
| | #15 | Review of configuration management best practices |
| Trigger Alerts and/or Logs for each script | #16 | A person (administrator/authorized/unauthorized) enables/disables a specific unused service using Ansible Playbook in the target servers |

| | | |
|---|---|---|
| run using Ansible Playbook | #17 | A person (administrator/authorized/unauthorized) enables/disables a specific critical service using Ansible Playbook in the target servers |
| Enabling connection through pfSense | #18 | An administrator enables Ansible WinRM connection to the Webserver through pfSense Firewall |
| Disallow unauthorized access through pfSense | #19 | An unauthorized person makes an unauthorized connection to the webserver |