

Automatização de configuração segura de sistemas

João Tomás Pereira Costa - a44582

Trabalho realizado sob a orientação de

Prof. Tiago Pedrosa

Prof. Rui Alves

Licenciatura em Engenharia Informática

2023-2024

Automatização de configuração segura de sistemas

Relatório da UC de Projeto
Licenciatura em Engenharia Informática
Escola Superior de Tecnologia e Gestão

João Tomás Pereira Costa - a44582

2023-2024

A Escola Superior de Tecnologia e de Gestão não se responsabiliza pelas opiniões expressas neste relatório.

Declaro que o trabalho descrito neste relatório é da minha autoria e é da minha vontade que o mesmo seja submetido a avaliação.

João Tomás Pereira Costa - a44582

Dedicatória

Dedico este trabalho aos meus pais, avós, irmãos e amigos, que me apoiaram em todos os momentos com amor, sabedoria e amizade. Agradeço por estarem sempre ao meu lado e por acreditarem em mim. Sem vocês, nada disto seria possível.

Agradecimentos

Um grande agradecimento ao IPB, como instituição de ensino que me acolheu, à Escola Superior de Tecnologia e Gestão, onde pude integrar este percurso formativo, e aos professores que sempre me ajudaram, se dedicaram e contribuíram para eu chegar a este patamar. Ao longo deste projeto tive a oportunidade de contar com o apoio de diversas pessoas que contribuíram, diretamente ou indiretamente, para a conclusão deste projeto. Um agradecimento especial ao Professor Tiago Pedrosa que sempre se mostrou disponível para ajudar no que fosse preciso e esclarecer qualquer dúvida que tivesse. Para tudo, para todos, o meu bem-haja!

Resumo

Este projeto tem como objetivo automatizar a configuração segura de servidores *linux*, para minimizar assim possíveis erros de configurações, e também documentar as discrepâncias entre a configuração que os sistemas deveriam ter e as configurações reais.

Através das ferramentas de automatização *Ansible* e *Ansible AWX*, o objetivo é aprimorar a gestão eficiente da configuração segura de servidores e de acordo com as melhores práticas estabelecidas pelos *benchmarks CIS (Center for Internet Security)*, este processo de segurança denominado como *hardening*, tem como objetivo eliminar os meios de intrusão disponíveis num sistema através da correção de vulnerabilidades e da desativação de serviços não essenciais. Para analisar as vulnerabilidades dos sistemas foi utilizado o *OpenSCAP* em conjunto com o *Ansible AWX*. Estas duas ferramentas foram utilizadas também para gerar remediações baseadas nos relatórios gerados, corrigindo assim vulnerabilidades que não foram conseguidas através do role utilizado.

O projeto enfatiza a importância da automatização na satisfação das necessidades organizacionais, mostrando a capacidade de simplificar operações e lidar com os desafios decorrentes da constante evolução no cenário de TI.

Palavras-chave: Automatização, *Ansible*, *AWX*, *OpenSCAP*

Abstract

This project aimed to automate the secure configuration of linux servers, to minimize possible configuration errors, and also document discrepancies between the configuration that the systems should have and the actual configurations.

Through the automation tools Ansible and Ansible AWX, the objective was to improve the efficient management of secure server configuration and in accordance with the best practices established by benchmarks CIS (Center for Internet Security), this process of security called hardening, aims to eliminate the means of intrusion available in a system by correcting vulnerabilities and deactivating non-essential services. To analyze system vulnerabilities, OpenSCAP was used in conjunction with Ansible AWX. These two tools were also used to generate remediation based on the reports generated, thus correcting vulnerabilities that were not achieved through the role used.

The project emphasizes the importance of automation in satisfying organizational needs, showing the ability to simplify operations and deal with challenges arising from the constant evolution of the IT landscape.

Keywords: Automation, Ansible, AWX, OpenSCAP.

Conteúdo

Lista de Abreviaturas	xvi
1 Introdução	1
1.1 Enquadramento	1
1.2 Objetivos	2
1.3 Estrutura do Documento	2
2 Contexto e Tecnologias	3
2.1 Ferramentas Utilizadas	5
2.1.1 VirtualBox	5
2.1.2 Ansible	6
2.1.3 Ansible AWX	6
2.1.4 OpenSCAP	7
3 Abordagem	9
4 Implementação	11
4.1 Criação Cenário de Testes	11
4.1.1 Configuração Servidor Principal 20.04	12
4.1.2 Configuração dois servidores <i>Ubuntu Server</i> 20.04	12
4.1.3 Configuração <i>Ubuntu Desktop</i> 20.04	13
4.2 Instalação e configuração do Ansible	14
4.3 Instalação e configuração do <i>Ansible AWX</i>	15

4.3.1	Configurar a automatização segura dos servidores através do Ansible AWX	20
4.3.2	Integração com o git	20
4.3.3	Inventário	23
4.3.4	Credenciais	23
4.3.5	Templates	24
4.3.6	Workflow <i>templates</i>	38
4.4	Instalação e configuração dos serviços nos servidores	39
4.4.1	Servidor 1	40
4.4.2	Servidor 2	42
4.5	Transferência de ficheiros para o local host	44
5	Testes	45
5.0.1	Teste para a configuração base das máquinas	46
5.0.2	Teste para configuração com serviços nos sistemas	58
6	Conclusões	63
	Bibliografia	64
A	Proposta Original do Projeto	A1
B	Repositório GitHub Ansible AWX	B1
C	Demo Hardening Sistemas	C1

Lista de Tabelas

4.1	Informações do cenário de testes	11
5.1	Pontuações de segurança nas diferentes fases	57
5.2	Pontuações de segurança nas diferentes fases, comparação com a configuração base	62

Lista de Figuras

2.1	Ciclo de vida de processos de hardening em sistemas[1].	5
3.1	Diagrama cenário geral	9
3.2	Diagrama de Sequência	10
4.1	Configuração IP estático Master	12
4.2	Configuração IP estático no servidor 1	12
4.3	Configuração IP estático no servidor 2	13
4.4	Configuração IP estático no desktop 1(Ubuntu Desktop)	13
4.5	Ficheiro de configuração dos <i>hosts</i> do <i>Ansible</i>	14
4.6	Testar conexão	15
4.7	Verificar porta	18
4.8	Endereço IP	18
4.9	Interface gráfica Ansible AWX	19
4.10	Criar projeto	20
4.11	Criar inventário	23
4.12	Credenciais	24
4.13	Template hardening para o servidor principal	28
4.14	Template hardening para os hosts	28
4.15	Template para executar teste de segurança no servidor principal	30
4.16	Template para executar teste de segurança em ambiente de servidores	31
4.17	Template para executar teste de segurança em ambiente de desktop	31
4.18	Template para gerar <i>playbooks</i> de remediação nos servidores.	34

4.19	Template para gerar <i>playbooks</i> de remediação nos desktops.	34
4.20	Template para gerar <i>playbooks</i> de remediação no servidor principal.	35
4.21	Template para aplicar <i>playbooks</i> de remediação nos hosts.	35
4.22	Template para aplicar <i>playbooks</i> de remediação no servidor principal	36
4.23	Template para reiniciar os hosts.	37
4.24	Template para encerrar os hosts.	37
4.25	Workflow para robustecer a segurança em servidores	38
4.26	WebServer server1	41
4.27	Arquivo de configuração Samba	42
4.28	Samba	43
5.1	OpenSCAP relatório servidor principal (master)	47
5.2	OpenSCAP relatório hosts (server1)	48
5.3	OpenSCAP relatório hosts (server2)	48
5.4	OpenSCAP relatório hosts (desktop1)	49
5.5	Output hardening do servidor principal (master)	50
5.6	Output hardening dos hosts (servidor1, servidor2, desktop1)	50
5.7	OpenSCAP relatório servidor principal (master)	51
5.8	OpenSCAP relatório hosts (server1)	52
5.9	OpenSCAP relatório hosts (server2)	52
5.10	OpenSCAP relatório hosts (desktop1)	53
5.11	OpenSCAP relatório (master)	55
5.12	OpenSCAP relatório hosts (server1)	56
5.13	OpenSCAP relatório hosts (server2)	56
5.14	OpenSCAP relatório hosts (desktop1)	57
5.15	OpenSCAP relatório após instalação do serviços no server1	58
5.16	OpenSCAP relatório após instalação do serviços no server2	59
5.17	OpenSCAP relatório após instalação do serviços e remediação no server1 .	60
5.18	OpenSCAP relatório após instalação do serviços e remediação no server2 .	61

Lista de Abreviaturas

IPB	Instituto Politécnico de Bragança
ESTiG	Escola Superior de Tecnologia e Gestão
YML	Yet Another Markup Language
TI	Tecnologia da Informação
AWX	Ansible WorX
SSH	Secure Shell
IP	Internet Protocol
CPU	Central Processing Unit
SCAP	Security Content Automation Protocol
NIST	National Institute of Standards and Technology
CIS	Center for Internet Security
SFTP	Secure File Transfer Protocol
GRUB	GRand Unified Bootloader
MOTD	Message Of The Day
httpd	Hypertext Transfer Protocol daemon
nfs	Network File System
SSH	Secure Shell
PHP	Hypertext Preprocessor
WSL	Windows Subsystem for Linux
RBAC	Role-based Access Control

Capítulo 1

Introdução

Este relatório descreve o trabalho desenvolvido pelo autor no âmbito da unidade curricular de Projeto do Curso Licenciatura em Engenharia Informática do Instituto Politécnico de Bragança. O projeto teve como orientador o prof. Tiago Pedrosa e como co-orientador o prof. Rui Alves.

1.1 Enquadramento

A crescente complexidade dos sistemas informáticos e a necessidade de garantir a segurança em ambientes de TI tornam a gestão e a configuração de servidores uma tarefa cada vez mais crítica. No contexto empresarial, é essencial que os sistemas operem de acordo com as melhores práticas de segurança e que as configurações sejam consistentes em todos os servidores da infraestrutura.

O presente projeto insere-se neste contexto, propondo a automatização do processo de configuração segura em servidores Linux, utilizando a ferramenta *Ansible*. A proposta original do projeto, detalhada no apêndice A, identifica a necessidade de uma solução que não apenas automatize a configuração segura, mas que também ofereça um mecanismo para identificar e relatar divergências entre as configurações de base que devem ser respeitadas e as configurações reais.

1.2 Objetivos

O presente projeto pretende atingir os seguintes objetivos genéricos:

- Automatizar a configuração segura nos servidores Linux com recurso ao *Ansible*.
- Desenvolver uma ferramenta de relatório para identificar e documentar as diferenças entre a configuração padrão desejada e a configuração real nos sistemas.

1.3 Estrutura do Documento

O presente trabalho foi dividido por seis capítulos. O primeiro capítulo inclui uma introdução, um enquadramento do projeto, objetivos e uma descrição da organização do relatório. As ferramentas e tecnologias utilizadas no processo de desenvolvimento do projeto são discutidas no capítulo dois. O capítulo três tem vários diagramas que ilustram a estrutura geral e a sequência de interações do sistema. O capítulo quatro aborda a criação instalação e configuração do *Ansible* e do *Ansible AWX* para automatizar a configuração segura dos sistemas, assim como o desenvolvimento da solução para reportar as discrepâncias entre a configuração de base e a configuração real nos sistemas. O quinto capítulo apresenta os testes realizados e uma discussão do que poderia ter sido feito de outra forma. E por fim, o sexto capítulo, contém a conclusão.

Capítulo 2

Contexto e Tecnologias

A base de configurações seguras de um sistema é essencial para minimizar erros de configuração e assegurar uma segurança constante em todos os sistemas. Este capítulo tem como objetivo examinar os problemas mais comuns associados a configurações inseguras ou deficientes. Além disso, aborda também as melhores práticas para configurar sistemas de forma segura, as vantagens da automatização e as ferramentas específicas que podem ser usadas para alcançar esses objetivos, como Ansible, OpenSCAP e Ansible AWX.

Configurações inseguras podem expor os sistemas a vários tipos de ataques, o que pode levar a consequências muito graves. Compreender estas questões é importante para o desenvolvimento de medidas de segurança eficazes.

Existem muitos ataques associados a configurações inseguras, *brute force* é um exemplo. *Brute force* consiste na tentativa de descobrir a password de um determinado sistema através da utilização de todas as combinações possíveis da password. E pode ser facilmente evitado com boas configurações de segurança.

Escala de privilégios, mais conhecido como *Privelege Escalation* é outra ameaça comum a servidores. Resume-se basicamente a explorar os pontos fracos da configuração para obter direitos de acesso mais elevados do que os originalmente concebidos. Os atacantes podem utilizar várias técnicas, como explorar vulnerabilidades de *software* ou aproveitar configuração incorretas nas permissões do utilizador para elevar os seus privilégios. Esta falha pode assim levar ao acesso não autorizado a dados confidenciais, recursos do sistema

e funcionalidades administrativas. Prevenir esta ameaça envolve frequentemente garantir que os utilizadores têm as permissões mínimas necessárias e auditar regularmente os direitos de acesso dos utilizadores do sistema.

Falta de atualização de *software* são outra grande preocupação. Os servidores executam frequentemente um número *software* complexos que requerem atualizações regulares para corrigir falhas de seguranças de versões anteriores. As configurações inseguras podem resultar na não aplicação atempada destes *patches*, deixando o sistema exposto a vulnerabilidades conhecidas. Garantir um processo robusto de gestão de atualizações é essencial para manter a segurança de um sistema.

Uma boa forma de evitar ou minimizam os efeitos destes ataques é a aplicação de boas práticas de segurança nos sistemas, mas dada a complexidade e a quantidade de práticas de segurança que precisam de ser analisadas e configuradas, é necessária a utilização de ferramentas que validem de forma automatizada a conformidade com boas práticas no que diz respeito a configurações seguras de sistemas. Na figura seguinte retirada do livro [1], representa as três fases no ciclo de vida de um processo de *hardening* em sistemas operativos:

- **Fase 1** - Definir uma base de segurança tendo em conta práticas aprovadas por players de cibersegurança conhecidos, como por exemplo as *benchmarks* de segurança *CIS*.
- **Fase 2** - Utilizar ferramentas de análise e *scan* de não conformidades com a *benchmark* de segurança.
- **Fase 3** - A fase final consiste na monitorização de todas estas ferramentas e *benchmarks* de modo a garantir que todos os sistemas contenham configurações atualizadas.

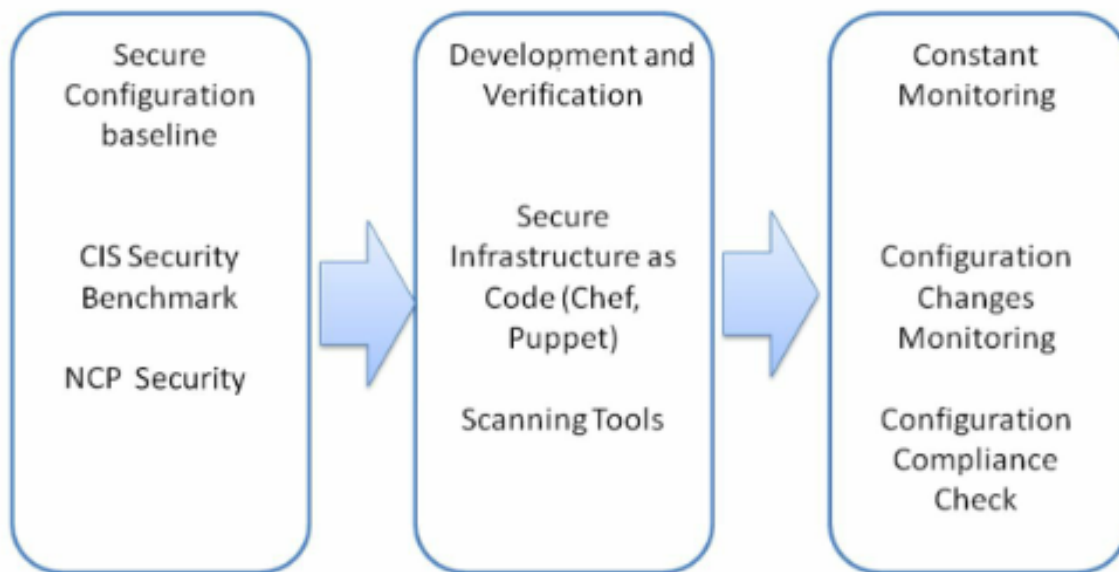


Figura 2.1: Ciclo de vida de processos de hardening em sistemas[1].

2.1 Ferramentas Utilizadas

2.1.1 VirtualBox

VirtualBox é uma ferramenta de virtualização open-source que permite a criação e gestão de máquinas virtuais. Com esta ferramenta, é possível executar múltiplos sistemas operativos em simultâneo num único computador físico, proporcionando uma solução prática e eficiente para desenvolvimento de teste e administração de sistemas. Neste caso foi utilizado para criar todo o cenário de testes. Todas as máquinas virtuais foram instaladas, configuradas e utilizadas neste software.

2.1.2 Ansible

Ansible é uma ferramenta *open-source* de automatização. Utiliza arquivos *YAML*, chamados de *playbooks*. Simplifica a configuração, administração e implementação principalmente em larga escala. Essa abordagem simplifica tarefas repetitivas como atualizações de sistema, configuração de serviços e aplicação de atualizações de segurança, reduzindo comuns associados ao erro humano. Os *playbooks* permitem definir o estado desejado do sistema, facilitando a implementação de configurações complexas e configurações repetitivas em vários sistemas. Além disso, esta ferramenta permite também registrar todas as ações executadas. Ao automatizar as aplicações de *patches* e a implementação de políticas de segurança, ajuda a proteger os servidores de vulnerabilidades, garantindo uma gestão de configuração centralizada e eficiente, possibilitando a administração consistente de infraestruturas. Em resumo, o *Ansible* não simplifica apenas a administração de servidores, mas também fortalece a segurança ao automatizar a aplicação de práticas recomendadas. Essa automatização resulta em maior eficiência operacional, menor tempo de inatividade e uma infraestrutura mais resiliente às ameaças de segurança.

2.1.3 Ansible AWX

O *Ansible AWX* é uma interface *web* e *API REST open source* para o *Ansible*, que permite automatizar, gerir e visualizar tarefas. Facilita a execução de *playbooks Ansible*, agendamento de tarefas e controlo de inventário, tornando a automatização mais acessível em ambientes complexos. Não só alarga as capacidades do *Ansible* em termos de gestão de controlo como também melhora a segurança e a gestão da automatização da infraestrutura em ambientes mais complexos. Aqui estão alguns benefícios de utilizar o *Ansible AWX*:

- **Interface gráfica:** O *Ansible AWX* oferece uma interface gráfica, permitindo que os utilizadores consigam gerir e automatizar a sua infraestrutura de forma fácil e intuitiva.

- **Gestão centralizada:** O *Ansible AWX* proporciona uma interface de gestão centralizada para o *Ansible*, facilitando a gestão e automatização da infraestrutura.
- **RBAC:** O *Ansible AWX* permite o controlo de acesso baseado em funções, permitindo assim que se controle quem tem acesso a quais recursos.

2.1.4 OpenSCAP

OpenSCAP é uma coleção de ferramentas open-source destinadas à implementação de normas de segurança, especialmente no contexto de conformidade e avaliação de vulnerabilidades. Baseado no *Security Content Automation Protocol* (SCAP), esta ferramenta é utilizada para automatizar verificações de conformidade com normas de segurança, é frequentemente utilizado para realizar verificações, avaliando a segurança com base em *benchmarks* como o *CIS* (*Center for Internet Security*).

O OpenSCAP opera principalmente através da análise de perfis de segurança como o *SCAP Security Guide* (SSG), que fornece conteúdos de segurança ajustados para diversas plataformas e serviços. O processo de avaliação de segurança com o OpenSCAP segue várias etapas, começando pela instalação e configuração das ferramentas necessárias, como o *OpenSCAP* e o *SCAP Security Guide*.

Durante a avaliação, o *OpenSCAP* utiliza perfis *SCAP* específicos para gerar relatórios detalhados sobre o estado de conformidade dos sistemas. Caso sejam identificadas vulnerabilidades que não foram tratadas pelo *hardening* inicial, *playbooks* específicos são utilizados para aplicar remediações. Essas remediações são adaptadas ao software e serviços instalados em cada máquina, garantindo que todas as vulnerabilidades sejam abordadas de maneira eficaz.

Além da criação e aplicação de remediações, o *OpenSCAP* permite a execução de testes regulares e a integração desses testes em *workflows* automatizados, facilitando a manutenção contínua da segurança dos sistemas, garantindo que as políticas de segurança sejam seguidas e que as vulnerabilidades sejam tratadas prontamente.

Capítulo 3

Abordagem

Para compreender melhor os requisitos do cenário a ser desenvolvido, foram criados os seguintes diagramas que ilustram a estrutura geral e a sequência de interações do sistema.

O primeira diagrama ajuda a visualizar e a perceber melhor a forma como todo o cenário de testes funciona.

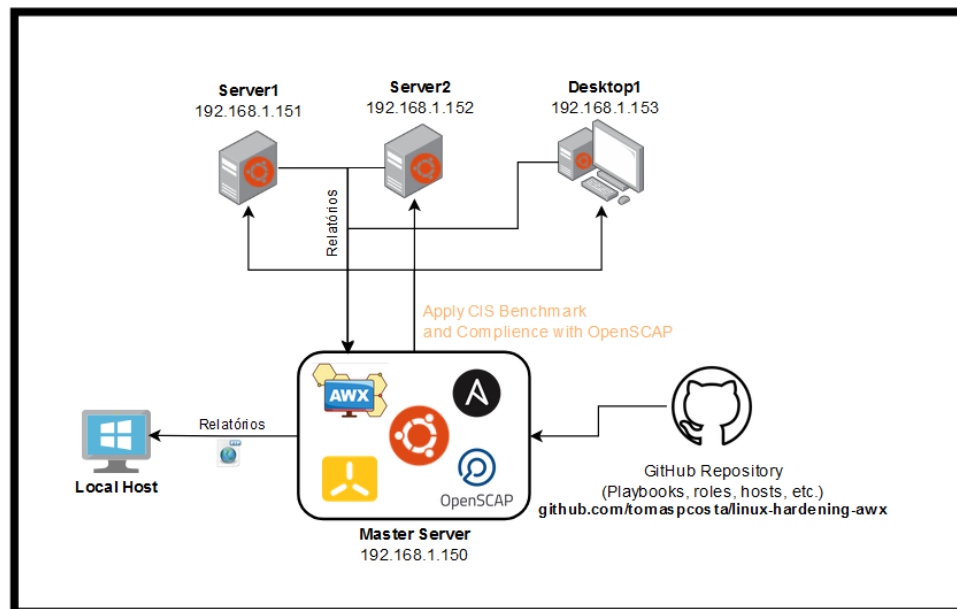


Figura 3.1: Diagrama cenário geral

O diagrama de sequência demonstra todo o processo desde a criação até à execução de uma *template* com o *Ansible AWX*.

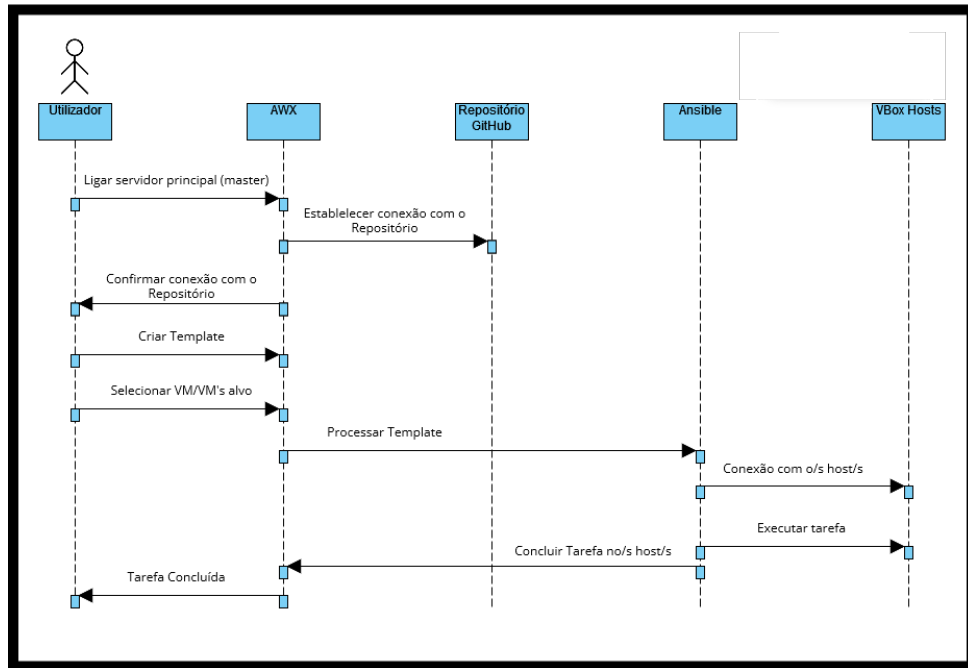


Figura 3.2: Diagrama de Sequência

Capítulo 4

Implementação

4.1 Criação Cenário de Testes

Para a criação do cenário de testes, foi escolhida a distribuição Linux *Ubuntu* na versão 20.04. Foram utilizadas quatro máquinas virtuais, uma para a instalação do *Ansible* e do *Ansible AWX*, outras três para implementar configurações seguras, duas com *Ubuntu Server* 20.04 e outra com *Ubuntu Desktop* 20.04 para testar também as configurações num sistema *Desktop*. A única configuração base necessária em todas as máquinas virtuais, exceto no servidor principal, é atualizar a lista de pacotes de software disponíveis, configurar a conexão *SSH* e atribuir um IP estático, tudo o resto é realizado e configurado no servidor principal.

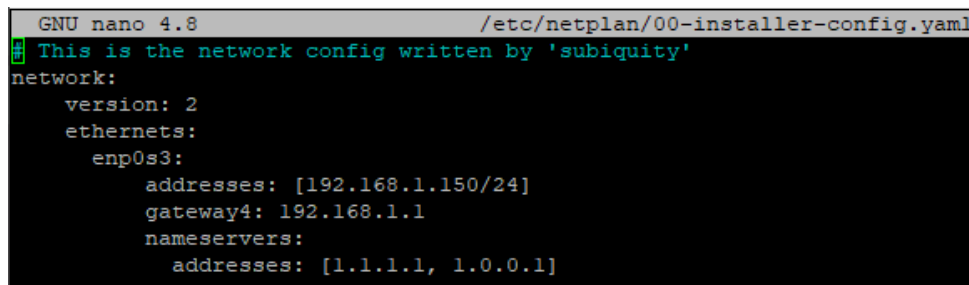
Sistema Operativo	Hostname	Endereço IP
Ubuntu Server 20.04	master	192.168.1.150
Ubuntu Server 20.04	server1	192.168.1.151
Ubuntu Server 20.04	server2	192.168.1.152
Ubuntu Desktop 20.04	desktop1	192.168.1.153

Tabela 4.1: Informações do cenário de testes

4.1.1 Configuração Servidor Principal 20.04

Para a configuração do servidor principal, começou-se por instalar o servidor *SSH* e atribuir um IP estático ao servidor.

```
$ sudo apt-get update && sudo apt-get upgrade  
$ sudo apt install openssh-server
```

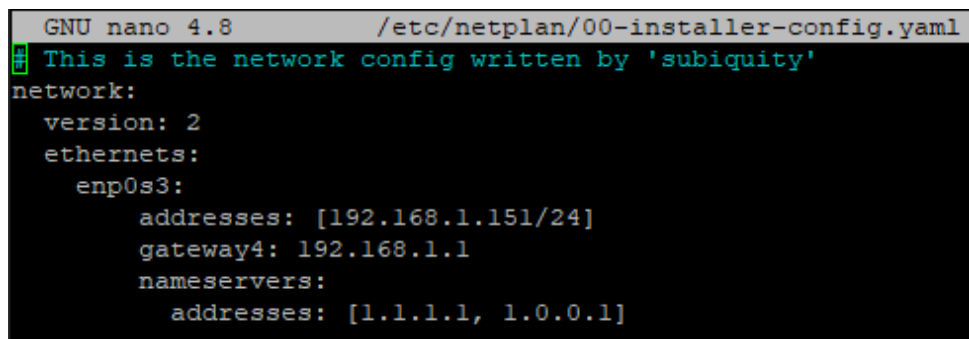


```
GNU nano 4.8 /etc/netplan/00-installer-config.yaml  
# This is the network config written by 'subiquity'  
network:  
  version: 2  
  ethernet:  
    enp0s3:  
      addresses: [192.168.1.150/24]  
      gateway4: 192.168.1.1  
      nameservers:  
        addresses: [1.1.1.1, 1.0.0.1]
```

Figura 4.1: Configuração IP estático Master

4.1.2 Configuração dois servidores *Ubuntu Server* 20.04

Para a configuração dos dois servidores (servidor1, servidor2), começou-se por atualizar os sistemas atribuir e atribuir IP estático editando a configuração do *netplan* em ‘/etc/netplan’.



```
GNU nano 4.8 /etc/netplan/00-installer-config.yaml  
# This is the network config written by 'subiquity'  
network:  
  version: 2  
  ethernet:  
    enp0s3:  
      addresses: [192.168.1.151/24]  
      gateway4: 192.168.1.1  
      nameservers:  
        addresses: [1.1.1.1, 1.0.0.1]
```

Figura 4.2: Configuração IP estático no servidor 1

```

GNU nano 4.8 /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  version: 2
  ethernets:
    enp0s3:
      addresses: [192.168.1.152/24]
      gateway4: 192.168.1.1
      nameservers:
        addresses: [1.1.1.1, 1.0.0.1]

```

Figura 4.3: Configuração IP estático no servidor 2

4.1.3 Configuração *Ubuntu Desktop* 20.04

Para a configuração da máquina virtual com o *Ubuntu Desktop*, após atualizar o sistema foi atribuído o IP estático através do ficheiro de configuração do netplan.

```

GNU nano 4.8 /etc/netplan/01-network-manager-all.yaml
## Let NetworkManager manage all devices on this system
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: no
      addresses:
        - 192.168.1.153/24
      gateway4: 192.168.1.1
      nameservers:
        addresses:
          - 8.8.8.8
          - 1.1.1.1

```

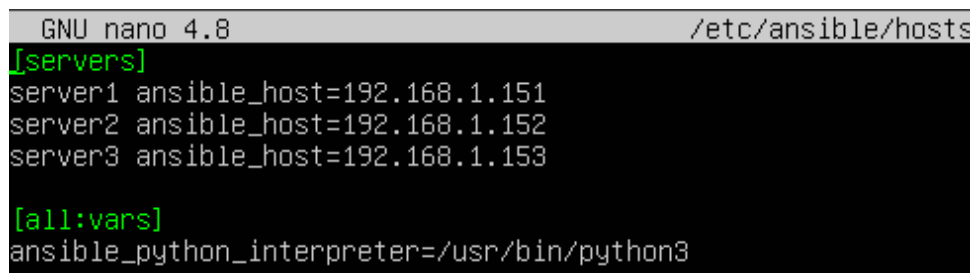
Figura 4.4: Configuração IP estático no desktop 1(Ubuntu Desktop)

4.2 Instalação e configuração do Ansible

Para a instalação do **Ansible** no servidor principal, foi consultado o tutorial fornecido no site da DigitalOcean [2] com o título *"How To Install and Configure Ansible on Ubuntu 20.04"*. Assim, foram utilizados no terminal do sistema os seguintes comandos:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt install ansible
$ sudo nano /etc/ansible/hosts
```

O último comando permite aceder ao ficheiro de configuração de *hosts*, onde foram adicionados os endereços *IP* dos servidores, neste caso servidor 1 com o endereço *IP* 192.168.1.151, servidor 2 com o endereço *IP* 192.168.1.152 e o desktop 1 com o endereço *IP* 192.168.1.153.

A screenshot of a terminal window showing the configuration of the /etc/ansible/hosts file using the GNU nano 4.8 editor. The file content is as follows:

```
GNU nano 4.8 /etc/ansible/hosts
[servers]
server1 ansible_host=192.168.1.151
server2 ansible_host=192.168.1.152
server3 ansible_host=192.168.1.153

[all:vars]
ansible_python_interpreter=/usr/bin/python3
```

Figura 4.5: Ficheiro de configuração dos *hosts* do *Ansible*

Para autenticação sem palavra-passe entre o *master* e os servidores é necessário gerar um par de chaves *SSH* pública e privada com a ajuda do comando **ssh-keygen**. De seguida, é necessário copiar a chave pública *SSH* para os servidores remotos com os seguintes comandos:

```
$ ssh-copy-id root@192.168.1.151
$ ssh-copy-id root@192.168.1.152
$ ssh-copy-id root@192.168.1.153
```

Já com o *Ansible* instalado e com a conexão *SSH* configurada, fez-se o uso de um comando para testar a conectividade com todos os servidores listados no ficheiro do inventário com o uso do módulo *ping*.

```
root@AnsibleMaster20:/etc/ansible# ansible all -m ping -u root
server1 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
server2 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
server3 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

Figura 4.6: Testar conexão

4.3 Instalação e configuração do *Ansible AWX*

Para a instalação e configuração do *Ansible AWX* no servidor principal, foram consultadas várias documentações, a instalação do *Kubernetes K3s* [3] e a do *Ansible AWX* [4] [5]. *Kubernetes* é uma plataforma que automatiza a implementação e gestão de aplicações em *containers*, sendo essencial para executar o *Ansible AWX* de forma eficiente. O K3s, uma versão leve do *Kubernetes*, é ideal para ambientes com recursos limitados, oferece uma instalação simples e rápida, além de exigir menos memória e *CPU*. Utilizando *Kubernetes* ou K3s, o *Ansible AWX* pode ser executado num ambiente escalável e resiliente, facilitando a automação e gestão de configurações em larga escala através de uma interface *web*. O processo de instalação foi realizada sem grandes problemas. Foram utilizados os seguintes comandos para instalar e configurar o *Ansible AWX*:

- **Instalar K3s**

```
$ curl -sfL https://get.k3s.io | sh -
```

- **Clonar o Repositório do AWX Operator**

```
$ git clone https://github.com/ansible/awx-operator.git
```

- **Listar e escolher as tags disponíveis no repositório**

```
$ cd awx-operator/
```

```
$ git tag
```

```
$ export VERSION=2.9.0
```

- **Deploy do AWX Operator no cluster Kubernetes**

```
$ make deploy
```

- **Criar ficheiro "Kustomize"**

```
$ mkdir kustomize
```

```
$ cd kustomize
```

- **Criar ficheiro "kustomization.yaml" e introduzir o seguinte conteúdo**

```
$ nano kustomization.yaml
```

```
apiVersion: kustomize.config.k8s.io/v1beta1
```

```
kind: Kustomization
```

```
resources:
```

```
  - github.com/ansible/awx-operator/config/default?ref=2.9.0
```

```
images:
```

```
  - name: quay.io/ansible/awx-operator
```

```
    newTag: 2.9.0
```

```
namespace: awx
```


- Criar ficheiro "awx.yml" e introduzir o seguinte conteúdo

```
$ nano awx.yml
```

```

—
  apiVersion: awx.ansible.com/v1beta1
  kind: AWX
  metadata:
    name: awx
  spec:
    service_type: nodeport

```

- Criar a instância do AWX no cluster

```
$ kubectl apply -k .
```

- Obter a palavra-passe de administrador do AWX

```
$ kubectl get secret awx-admin-password -o jsonpath="{.data.password}" -n awx | base64 --decode ; echo
```

- Alterar a palavra-passe de administrador do AWX

```

$ kubectl get pods --all-namespaces | grep awx-web
$ kubectl exec -it awx-web-6b89d9b54d-sqblg -n awx -- /bin/
  bash
$ awx-manage changepassword admin

```

- Verificar a porta do serviço "awx-service"

```
$ kubectl get all -n awx
```

```
root@master:/home/master# kubectl get all -n awx
```

NAME	READY	STATUS	RESTARTS
pod/awx-operator-controller-manager-85c949b69-rzc8b	2/2	Running	25 (78s ago)
pod/awx-postgres-13-0	1/1	Running	12 (78s ago)
pod/awx-task-b45bf8fd8-hzkxg	4/4	Running	48 (78s ago)
pod/awx-web-6b89d9b54d-sqblg	3/3	Running	37 (78s ago)

NAME	EXTERNAL-IP	PORT(S)	AGE	TYPE	CLUSTER-IP
service/awx-operator-controller-manager-metrics-service	38 <none>	8443/TCP	15d	ClusterIP	10.43.89.1
service/awx-postgres-13	<none>	5432/TCP	15d	ClusterIP	None
service/awx-service	49 <none>	80:31622/TCP	15d	NodePort	10.43.91.2

Figura 4.7: Verificar porta

- Verificar o endereço IP do servidor e utilizar ambos (IP e porta) para aceder à interface gráfica do Ansible AWX

```
$ ip a show enp0s3
```

```
root@master:/home/master# ip a show enp0s3ip a show enp0s3
Error: either "dev" is duplicate, or "a" is a garbage.
root@master:/home/master# ip a show enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b0:30:e1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.150/24 brd 192.168.1.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 2a01:14:100:5b60:a00:27ff:feb0:30e1/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 445sec preferred_lft 445sec
    inet6 fe80::a00:27ff:feb0:30e1/64 scope link
        valid_lft forever preferred_lft forever
```

Figura 4.8: Endereço IP

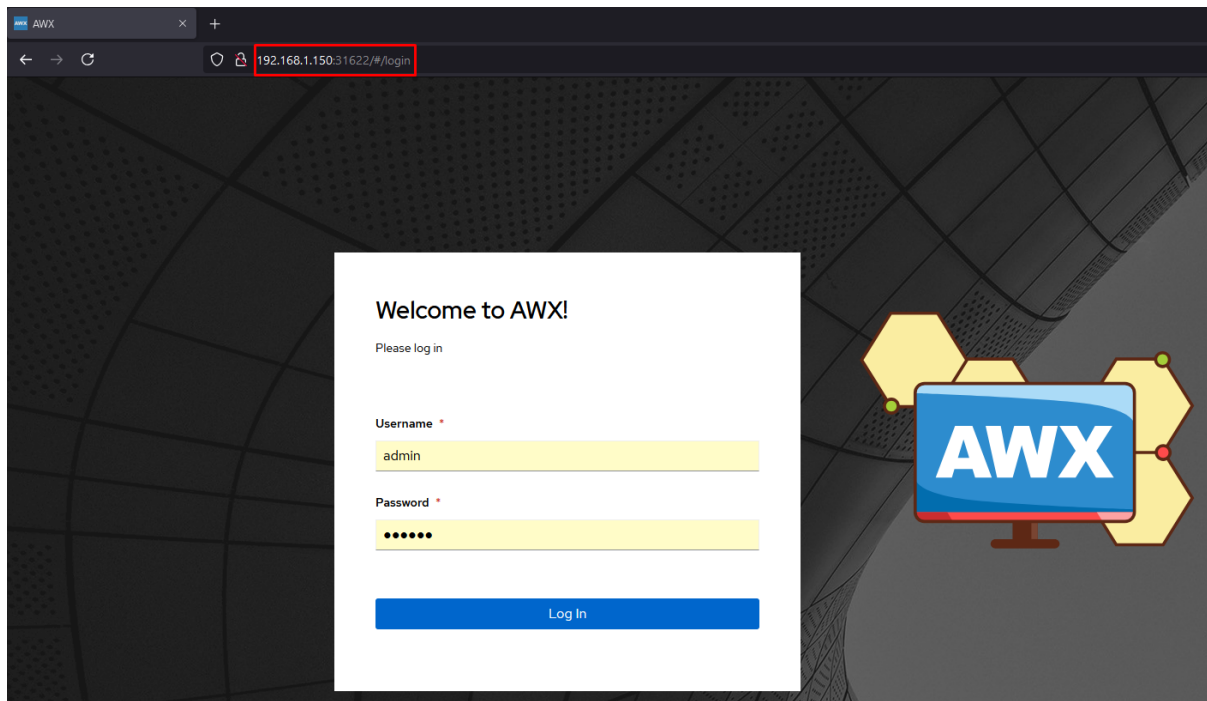


Figura 4.9: Interface gráfica Ansible AWX

4.3.1 Configurar a automatização segura dos servidores através do Ansible AWX

Nesta seção, será abordada a exploração e configuração da interface gráfica do Ansible AWX.

4.3.2 Integração com o git

O AWX detém algumas medidas que facilitam a gestão, como a criação de utilizadores, o controlo de permissões, a gestão da organização e regras de inventário. Além disso, conta com um sistema de gestão de projetos. Os projetos são entidades que representam repositórios de código, geralmente armazenados em sistemas de controlo de versão (*GitHub*). Com o AWX, é possível criar projetos tendo como fonte o *Git*. Na Figura 4.10, pode-se observar o tipo de controlo de origem como *Git*. Após a finalização de criação do projeto, basta clicar no botão de "*Sync*" e aguardar a atualização. Com alterações no repositório, é possível atualizar o projeto em tempo real

The screenshot shows the 'Create Project' form in the Ansible AWX web interface. The form is divided into several sections:

- Top Section:** Contains fields for 'Name' (highlighted with a red box, containing 'cis-hardening'), 'Description', 'Organization' (with a search icon and 'Default' selected), 'Execution Environment' (with a search icon), 'Source Control Type' (a dropdown menu showing 'Git'), and 'Content Signature Validation Credential' (with a search icon).
- Type Details Section:** Contains fields for 'Source Control URL' (highlighted with a red box, containing 'https://github.com/tomaspocosta/awx.git'), 'Source Control Branch/Tag/Commit', and 'Source Control Refspec'. Below these is a 'Source Control Credential' field with a search icon.
- Options Section:** Contains a group of checkboxes: 'Clean', 'Delete', 'Track submodules', 'Update Revision on Launch' (checked and highlighted with a red box), and 'Allow Branch Override'.
- Option Details Section:** Contains a 'Cache Timeout' field with a search icon, set to '7200'.

At the bottom of the form, there are 'Save' and 'Cancel' buttons.

Figura 4.10: Criar projeto

O repositório criado no *github* contém os seguintes ficheiros:

- **collections/requirements.yml**: Utilizado para adicionar as *collections* que são necessárias para que uma determinada role ou *playbook* funcione corretamente.
- **group_vars/cis.yml**: Utilizado para adicionar variáveis gerais.
- **host_vars**: Utilizado para adicionar algumas variáveis específicas para hosts.
- **hosts/hosts.yml**: Adicionar os *hosts* em formato *yaml*, na pasta dos *hosts*, porque o AWX tem problemas com inventários *.yaml* na raiz do projeto.
- **roles/requirements.yml**: Utilizado para adicionar o role *cis-hardening*.
- **ansible.cfg**: Arquivo de configuração para controlar como o Ansible se comporta ao executar *playbooks* e outras tarefas no contexto de um ambiente gerido pelo AWX.
- **hardening.yml**: Cria o *playbook* para executar o role *cis-hardening*.
- **openscap_report_desktop.yml**: *playbook* para executar avaliações de segurança com o *Openscap* e gerar relatórios no ambiente *desktop*.
- **openscap_report_servers.yml**: *playbook* para executar avaliações de segurança com o *Openscap* e gerar relatórios em servidores.
- **poweroff_hosts.yml**: *playbook* para desligar os *hosts*.
- **reboot_hosts.yml**: *playbook* para reiniciar os *hosts*.
- **remediate_desktop.yml**: *playbook* gerado automaticamente com o *Openscap* para remediar aquilo que não foi conseguido através do role, para executar em ambientes *desktop*.
- **remediate_hosts.yml**: *playbook* gerado automaticamente com o *Openscap* para remediar aquilo que não foi conseguido através do role, para executar em servidores.

- **remediate_master.yml:** *playbook* gerado automaticamente com o *Openscap* para remediar aquilo que não foi conseguido através do role, para executar no servidor principal.
- **apply_remediation.yml:** Aplica o arquivo de remediação scap nos hosts alvo.
- **scap_remediation_generator_desktop.yml:** Gera e transfere o arquivo de remediação scap para sistemas desktop.
- **scap_remediation_generator_master.yml:** Gera e transfere o arquivo de remediação scap para o servidor principal.
- **scap_remediation_generator_servers.yml:** Gera e transfere o arquivo de remediação scap para sistemas de servidores.

4.3.3 Inventário

É necessário criar um inventário onde serão adicionados os endereços IP dos *hosts* que se pretende gerir. O ficheiro dos *hosts* encontra-se no repositório, por isso é necessário criar uma fonte (*source*) para obter os dados dos *hosts* diretamente a partir do repositório e basta apenas sincronizar sempre que forem feitas alterações no ficheiro.

The screenshot shows the 'Create inventory' form in Ansible. The form is divided into several sections. The top section contains 'Name' (set to 'hosts-sync'), 'Description', and 'Execution Environment'. Below this is the 'Source' dropdown, which is highlighted with a red box and set to 'Sourced from a Project'. The 'Source details' section follows, containing 'Credential', 'Project' (set to 'cis-hardening', highlighted with a red box), and 'Inventory file' (set to 'hosts/hosts.yml', highlighted with a red box). Below this are 'Verbosity' (set to '1 (Info)'), 'Host Filter', and 'Enabled Variable'. The 'Enabled Value' field is empty. The 'Update options' section has three checkboxes: 'Overwrite', 'Overwrite variables', and 'Update on launch' (checked, highlighted with a red box). The 'Cache timeout (seconds)' field is set to '7200'.

Figura 4.11: Criar inventário

4.3.4 Credenciais

As credenciais são utilizadas para autenticar o acesso a servidores remotos sobre os quais o *Ansible* vai executar comandos. Para facilitar este processo, foi criado um utilizador comum em todos os servidores, com o nome de *"server"*. Caso contrário, seria necessário adicionar as credenciais de cada servidor individualmente, o que inviabilizaria a execução simultânea dos *playbooks* em vários servidores (*"ubuntu-server-user"*). Foram adicionadas as credenciais dos servidores e, por motivos de segurança, foram criadas credenciais específicas para o servidor principiapl (master) (*"ubuntu-master-user"*).

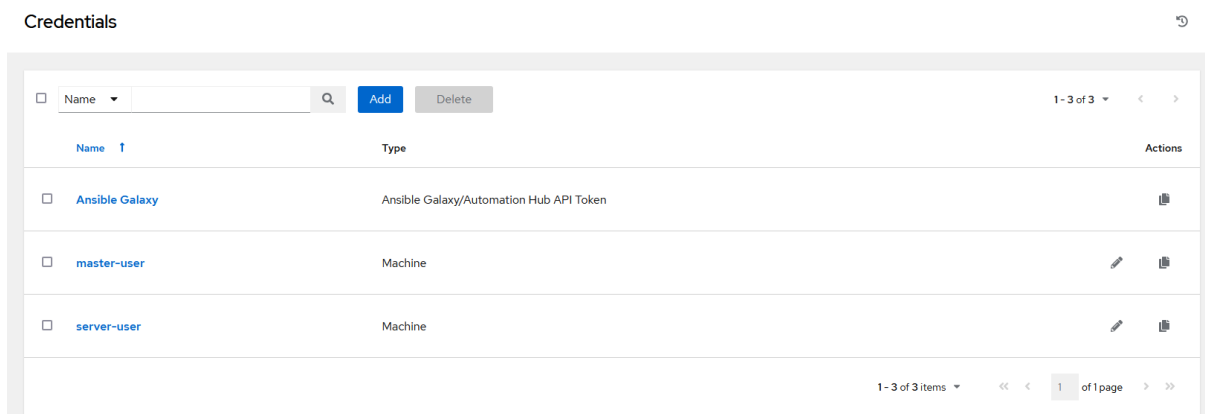


Figura 4.12: Credenciais

4.3.5 Templates

No *Ansible AWX*, *templates* são configurações que definem como *playbooks* devem ser executados. Eles permitem a parametrização de execuções, associando *playbooks* a inventários, credenciais e variáveis específicas, facilitando assim a automatização e a reutilização de tarefas em diferentes cenários.

Templates de Robustecimento dos Sistemas com Ansible Lockdown

Inicialmente para o robustecimento dos sistemas, foram criadas duas *templates* para aplicar as melhores práticas estabelecidas pelos *benchmarks CIS*, uma nos *hosts* e outra para o servidor principal. Primeiro foi utilizado o *playbook "hardening.yml"*, que utiliza o *role* gerado através do repositório *ansible-lockdown*.

O *Ansible Lockdown* é um conjunto de *roles* e *playbooks* desenvolvido para assegurar que sistemas operacionais estejam em conformidade com *benchmarks* de segurança estabelecidos, como o *CIS Benchmark*. No caso do *Ubuntu 20.04*, o *Ansible Lockdown* é utilizado para aplicar automaticamente as configurações recomendadas pelo *CIS Ubuntu Linux 20.04 LTS Benchmark v2.0.1*. Este *role* é uma ferramenta de remediação projetada para ser usada após a realização de uma auditoria de segurança, visando implementar as

melhores práticas e aumentar a segurança do sistema.

O *playbook* está dividido em diferentes secções, cada uma focada em um aspeto específico da segurança do sistema, desde a configuração inicial, passando pela gestão de acessos e sincronização de tempo, até auditorias e permissões de ficheiros. A seguir, é apresentada uma descrição detalhada do que este *role* implementa nos sistemas, onde se pode ver nos anexos [6] [7].

1. **Secção 1 - Configuração Inicial do Sistema:** Esta secção aborda a configuração de elementos críticos na fase inicial de configuração do sistema, incluindo a integridade do sistema, configurações de partição, mensagens de boas-vindas e inicialização segura.

- **Partição /tmp:** Verifica se a partição /tmp está montada corretamente utilizando `fstab` ou `systemd`.
- **AIDE:** Ferramenta de verificação de integridade do sistema configurada para execução diária através de *cron*.
- **GRUB:** Configurações de password no *GRUB* para evitar acessos não autorizados. A password é gerada com o comando `grub-mkpasswd-pbkdf2` e armazenada como *hash*.
- **MOTD (Message of the Day):** O *dynamic MOTD* é desativado, sendo substituído por uma mensagem simples: *"Authorized uses only. All activity may be monitored and reported."*

2. **Secção 2 - Sincronização de Tempo e Configurações de Servidores:** Esta secção trata da sincronização de tempo do sistema e da gestão de serviços de rede, desativando serviços desnecessários e ajustando as configurações de segurança.

- **Sincronização de Tempo:** A ferramenta de sincronização de tempo usada é o `systemd-timesyncd`, embora *chrony* ou *ntp* possam ser alternativas.
- **Servidores NTP:** Configuram-se servidores *NTP* como `time.nist.gov`.

- **Serviços de Rede:** Desativam-se serviços desnecessários, como `cups`, `httpd`, `nfs`, entre outros, para reduzir a superfície de ataque.

3. **Secção 3 - Firewall e Segurança de Rede:** Esta secção foca-se em proteger a rede e configurar a *firewall*, definindo regras de tráfego e reforçando a segurança do serviço SSH.

- **Firewall (UFW):** Configuração para permitir apenas tráfego de saída nas portas 53 (DNS), 80 (HTTP) e 443 (HTTPS), e apenas tráfego de entrada na porta 22 (SSH).
- **SSH:** Assegura a segurança no SSH, limitando as tentativas de autenticação a 4, configurando criptografia forte (`ciphers`) e um tempo de sessão máximo de 300 segundos (`client_alive_interval`). O acesso SSH é restrito aos usuários `vagrant` e `ubuntu`.

4. **Secção 4 - Gestão de Acessos e Passwords:** Aqui são definidas políticas de gestão de utilizadores, *sudo* e *passwords*, garantindo que as credenciais de acesso são geridas de forma segura.

- **Sudo:** O pacote `sudo` é instalado, e o ficheiro de *logs* para atividades *sudo* é definido como `/var/log/sudo.log`. O tempo de sessão *sudo* é limitado a 15 minutos.
- **Segurança de Passwords:** As passwords são protegidas com o algoritmo `sha512`. Estabelece-se uma política de expiração de passwords que obriga à sua alteração a cada 365 dias, com uma inatividade máxima de 30 dias antes de bloquear a conta.
- **Histórico de Passwords:** Impede-se a reutilização das últimas 5 passwords. O ficheiro `/etc/profile` é ajustado para aplicar uma `umask` de 027, assegurando permissões padrão mais seguras.

5. **Secção 5 - Logging e Auditoria:** Esta secção aborda o registo e auditoria das atividades do sistema, garantindo a capacidade de monitorização contínua e retenção segura dos *logs*.

- **Syslog:** O *rsyslog* é configurado como serviço de *logging*. Se o sistema atuar como servidor de *logs*, deve-se especificar o IP do servidor de *logs* remoto.
- **Auditd:** O *auditd* é configurado para auditar atividades administrativas. Em caso de falta de espaço para *logs*, o sistema é configurado para bloquear (ação *halt*).
- **Logrotate:** O *logrotate* é configurado para rodar as *logs* diariamente, assegurando que não crescem indefinidamente.

6. **Secção 6 - Gestão de Permissões de Ficheiros:** Esta secção é responsável por gerir permissões de ficheiros, corrigindo permissões incorretas e assegurando a ausência de ficheiros sem proprietário.

- **Ficheiros sem proprietário:** Ficheiros sem "dono" são atribuídos ao `root`, o mesmo para ficheiros sem grupo.
- **World-writable:** Ficheiros com permissões de escrita para todos (*world-writable*) têm essas permissões removidas automaticamente.

Templates > [1] cis-hardening-master

Details

Back to TemplatesDetailsAccessNotificationsSchedulesJobsSurvey

Name	[1] cis-hardening-master	Job Type	run	Organization	Default
Inventory	awx-hosts	Project	cis-hardening-github-repo	Execution Environment	AWX EE (latest)
Playbook	hardening.yml	Forks	0	Limit	master
Verbosity	0 (Normal)	Timeout	0	Show Changes	Off
Job Slicing	1	Created	11/08/2024, 22:29:06 by admin	Last Modified	02/09/2024, 03:32:54 by admin
Enabled Options	Privilege Escalation Concurrent Jobs Fact Storage				
Credentials	SSH: master-user				
Variables	YAMLJSON				
<div>1 ---</div>					
<div>EditLaunchDelete</div>					

Figura 4.13: Template hardening para o servidor principal

Templates > [12] cis-hardening-servers

Details

Back to TemplatesDetailsAccessNotificationsSchedulesJobsSurvey

Name	[12] cis-hardening-servers	Job Type	run	Organization	Default
Inventory	awx-hosts	Project	cis-hardening-github-repo	Execution Environment	AWX EE (latest)
Playbook	hardening.yml	Forks	0	Limit	server1,server2,desktop1
Verbosity	0 (Normal)	Timeout	0	Show Changes	Off
Job Slicing	1	Created	11/08/2024, 22:29:09 by admin	Last Modified	02/09/2024, 03:32:59 by admin
Enabled Options	Privilege Escalation Concurrent Jobs Fact Storage				
Credentials	SSH: server-user				
Variables	YAMLJSON				
<div>1 ---</div>					
<div>EditLaunchDelete</div>					

Figura 4.14: Template hardening para os hosts

Templates de Teste de Segurança com Openscap

Para garantir a conformidade e a segurança dos sistemas, foram criados três *playbooks*, e consequentemente três *templates*, para realizar testes de segurança com o *Openscap*.

Os *playbooks* foram configurados para três tipos diferentes de sistemas: *desktop*, servidores e servidor principal. Cada *playbook* segue um processo semelhante para instalar e configurar o *Openscap*, executar uma avaliação de segurança e gerar relatórios, alterando apenas o perfil utilizado. A seguir, são apresentados os perfis utilizados nos diferentes *playbooks*:

- **openscap_report_desktop.yml:**

xccdf_org.ssgproject.content_profile_cis_level1_workstation

- **openscap_report_servers.yml:**

xccdf_org.ssgproject.content_profile_cis_level1_server

- **openscap_report_master.yml:**

xccdf_org.ssgproject.content_profile_cis_level2_server

Os *playbooks* seguem as seguintes etapas:

1. Atualizar e fazer upgrade dos pacotes do sistema para garantir que o ambiente esteja atualizado.
2. Instalar o Openscap e o scap Security Guide, se ainda não estiverem instalados.
3. Verificar se o scap Security Guide foi instalado e descompactado.
4. Executar a avaliação de segurança utilizando o perfil scap específico para o tipo de sistema.
5. Gerar um relatório HTML com os resultados da avaliação.
6. Transferir o relatório para o servidor principal para armazenamento e análise posterior.

Esses *playbooks* permitem uma avaliação automatizada e padronizada da segurança dos sistemas, facilitando a identificação de vulnerabilidades e a conformidade com as políticas de segurança estabelecidas.

Templates > [21] openscap-reporting-master Details

◀ Back to Templates Details Access Notifications Schedules Jobs Survey					
Name	[21] openscap-reporting-master	Job Type ⓘ	run	Organization	Default
Inventory ⓘ	awx-hosts	Project ⓘ	cis-hardening-github-repo	Execution Environment ⓘ	AWX EE (latest)
Playbook ⓘ	openscap_reports_playbooks/ openscap_report_master_v12.yml	Forks ⓘ	0	Limit ⓘ	master
Verbosity ⓘ	0 (Normal)	Timeout ⓘ	0	Show Changes ⓘ	Off
Job Slicing ⓘ	1	Created	21/08/2024, 18:41:04 by admin	Last Modified	04/09/2024, 03:15:03 by admin
Enabled Options ⓘ	Privilege Escalation Concurrent Jobs Fact Storage				
Credentials ⓘ	SSH: master-user				
Variables ⓘ	YAML JSON				
1 ---					
Edit Launch Delete					

Figura 4.15: Template para executar teste de segurança no servidor principal

Details

[Back to Templates](#) [Details](#) [Access](#) [Notifications](#) [Schedules](#) [Jobs](#) [Survey](#)

Name	[2.2] openscap-reporting-servers	Job Type	run	Organization	Default
Inventory	awx-hosts	Project	cis-hardening-github-repo	Execution Environment	AWX EE (latest)
Playbook	openscap_reports_playbooks/ openscap_report_hosts.yml	Forks	0	Limit	server1,server2
Verbosity	0 (Normal)	Timeout	0	Show Changes	Off
Job Slicing	1	Created	15/08/2024, 16:38:42 by admin	Last Modified	04/09/2024, 03:15:29 by admin
Enabled Options	Privilege Escalation Concurrent Jobs Fact Storage				
Credentials	SSH: server-user				
Variables	<div>YAML JSON</div> <div><pre>1 --- 2 ansible_ssh_pass: 'xyz123'</pre></div> <div>Edit Launch Delete</div>				

Figura 4.16: Template para executar teste de segurança em ambiente de servidores

Details

[Back to Templates](#) [Details](#) [Access](#) [Notifications](#) [Schedules](#) [Jobs](#) [Survey](#)

Name	[2.3] openscap-reporting-desktop	Job Type	run	Organization	Default
Inventory	awx-hosts	Project	cis-hardening-github-repo	Execution Environment	AWX EE (latest)
Playbook	openscap_reports_playbooks/ openscap_report_desktop.yml	Forks	0	Limit	desktop1
Verbosity	0 (Normal)	Timeout	0	Show Changes	Off
Job Slicing	1	Created	02/09/2024, 03:21:02 by admin	Last Modified	04/09/2024, 03:11:36 by admin
Enabled Options	Privilege Escalation Concurrent Jobs Fact Storage				
Credentials	SSH: server-user				
Variables	<div>YAML JSON</div> <div><pre>1 --- 2 ansible_ssh_pass: 'xyz123'</pre></div> <div>Edit Launch Delete</div>				

Figura 4.17: Template para executar teste de segurança em ambiente de desktop

Templates de Remediação Openscap

Após o processo de *hardening* realizado com o role **ansible-lockdown**, algumas vulnerabilidades podem não ter sido completamente mitigadas. Para tratar essas restantes vulnerabilidades, foram criados *playbooks* para gerar e aplicar remediações *scap*. O processo de criação dos *playbooks* de remediação é baseado na configuração dos softwares e serviços instalados em cada máquina.

Os seguintes *playbooks* são utilizados para gerar arquivos de remediação *scap*, adaptados para diferentes tipos de sistemas:

- **scap_remediation_generator_desktop.yml:** Gera o arquivo de remediação *scap* para sistemas desktop com base no perfil *xccdf_org.ssgproject.content_profile_cis_level1_workstation*.
- **scap_remediation_generator_master.yml:** Gera o arquivo de remediação *scap* para o servidor principal, utilizando o perfil *xccdf_org.ssgproject.content_profile_cis_level2_server*.
- **scap_remediation_generator_servers.yml:** Gera o arquivo de remediação *scap* para sistemas de servidores, com o perfil *xccdf_org.ssgproject.content_profile_cis_level1_server*.

Cada *playbook* realiza as seguintes etapas para gerar o arquivo de remediação:

1. Verifica se o Openscap e o scap Security Guide estão instalados.
2. Gera o arquivo de remediação *scap* utilizando o comando `oscap xccdf generate fix` com o perfil específico para o tipo de sistema. Este arquivo contém as instruções necessárias para remediar as vulnerabilidades identificadas.
3. Transfere o arquivo de remediação gerado para o servidor principal.

O *playbook* **apply__remediation.yml** é responsável por aplicar as remediações geradas. No entanto, devido a erros de formatação encontrados nos arquivos de remediação, não foi possível utilizar o *playbook* logo após os *playbooks* de remediação serem gerados, impossibilitando assim a criação de um *workflow* para automatizar este processo. Foi necessário corrigir esses erros manualmente antes de prosseguir com a aplicação das remediações, garantindo assim que as vulnerabilidades fossem tratadas de acordo com os perfis *scap* estabelecidos.

Para automatizar o processo de gerar e remediar as vulnerabilidades identificadas, foram criados cinco *templates* no *Ansible AWX*. Cada *template* corresponde a uma tarefa específica no ciclo de aplicação de remediações:

- **openscap-remediation-generator-desktop:** Utilizado para gerar o *playbook* de remediação para sistemas desktop. Utiliza o *playbook* *scap_remediation_generator_desktop.yml*.
- **openscap-remediation-generator-servers:** Utilizado para gerar o *playbook* de remediação para servidores. Utiliza o *playbook* *scap_remediation_generator_servers.yml*.
- **openscap-remediation-generator-master:** Utilizado para gerar o *playbook* de remediação para o servidor principal. Utiliza o *playbook* *scap_remediation_generator_master.yml*.
- **apply-remediation-master:** Utilizado para aplicar os *playbooks* de remediação no servidor principal. Utiliza o *playbook* *apply_remediation.yml*.
- **apply-remediation-hosts:** Utilizado para aplicar o *playbook* de remediação em todos os hosts. Utiliza o *playbook* *apply_remediation.yml*.

Estas *templates* facilitam e automatizam a criação e aplicação de remediações, garantindo que as vulnerabilidades sejam abordadas de maneira eficaz em diferentes tipos de sistemas.

Templates > [14] openscap-remediation-generator-servers

Details

Back to Templates

Details

Access

Notifications

Schedules

Jobs

Survey

Name	[14] openscap-remediation-generator-servers	Job Type	run	Organization	Default
Inventory	awx-hosts	Project	cis-hardening-github-repo	Execution Environment	AWX EE (latest)
Playbook	scap_remediation_automate/ scap_remediation_generator_servers.yml	Forks	0	Limit	server1,server2
Verbosity	0 (Normal)	Timeout	0	Show Changes	Off
Job Slicing	1	Created	02/09/2024, 03:22:08 by admin	Last Modified	09/09/2024, 01:32:48 by admin
Enabled Options	Privilege Escalation Concurrent Jobs Fact Storage				
Credentials	SSH: server-user				
Variables	<div>YAMLJSON</div> <div>1 --- 2 ansible_ssh_pass: 'xyz123'</div>				

Edit

Launch

Delete

Figura 4.18: Template para gerar *playbooks* de remediação nos servidores.

Templates > [15] openscap-remediation-generator-desktop

Details

Back to Templates

Details

Access

Notifications

Schedules

Jobs

Survey

Name	[15] openscap-remediation-generator-desktop	Job Type	run	Organization	Default
Inventory	awx-hosts	Project	cis-hardening-github-repo	Execution Environment	AWX EE (latest)
Playbook	scap_remediation_automate/ scap_remediation_generator_desktop.yml	Forks	0	Limit	desktop1
Verbosity	0 (Normal)	Timeout	0	Show Changes	Off
Job Slicing	1	Created	02/09/2024, 03:22:31 by admin	Last Modified	09/09/2024, 01:33:17 by admin
Enabled Options	Privilege Escalation Concurrent Jobs Fact Storage				
Credentials	SSH: server-user				
Variables	<div>YAMLJSON</div> <div>1 --- 2 ansible_ssh_pass: 'xyz123'</div>				

Edit

Launch

Delete

Figura 4.19: Template para gerar *playbooks* de remediação nos desktops.

Templates

>

[13] openscap-generate-remediation-master

Details

Back to Templates

Details

Access

Notifications

Schedules

Jobs

Survey

Name

[13] openscap-generate-remediation-master

Job Type

run

Organization

Default

Inventory

awx-hosts

Project

cis-hardening-github-repo

Execution Environment

AWX EE (latest)

Playbook

scap_remediation_automate/
scap_remediation_generator_servers.yml

Forks

0

Limit

master

Verbosity

0 (Normal)

Timeout

0

Show Changes

Off

Job Slicing

1

Created

02/09/2024, 03:22:56 by admin

Last Modified

09/09/2024, 01:31:09 by admin

Enabled Options

Privilege Escalation
Concurrent Jobs
Fact Storage

Credentials

SSH: master-user

Variables

YAMLJSON

1

2

ansible_ssh_pass: 'xyz123'

Edit

Launch

Delete

Figura 4.20: Template para gerar *playbooks* de remediação no servidor principal.

Templates

>

[51] Apply-Remediation-hosts

Details

Back to Templates

Details

Access

Notifications

Schedules

Jobs

Survey

Name

[51] Apply-Remediation-hosts

Job Type

run

Organization

Default

Inventory

awx-hosts

Project

cis-hardening-github-repo

Execution Environment

AWX EE (latest)

Playbook

scap_remediation_automate/
apply_remediation.yml

Forks

0

Verbosity

0 (Normal)

Timeout

0

Show Changes

Off

Job Slicing

1

Created

09/09/2024, 01:34:32 by admin

Last Modified

09/09/2024, 01:34:32 by admin

Enabled Options

Privilege Escalation
Concurrent Jobs
Fact Storage

Credentials

SSH: server-user

Variables

YAMLJSON

1

Edit

Launch

Delete

Figura 4.21: Template para aplicar *playbooks* de remediação nos hosts.

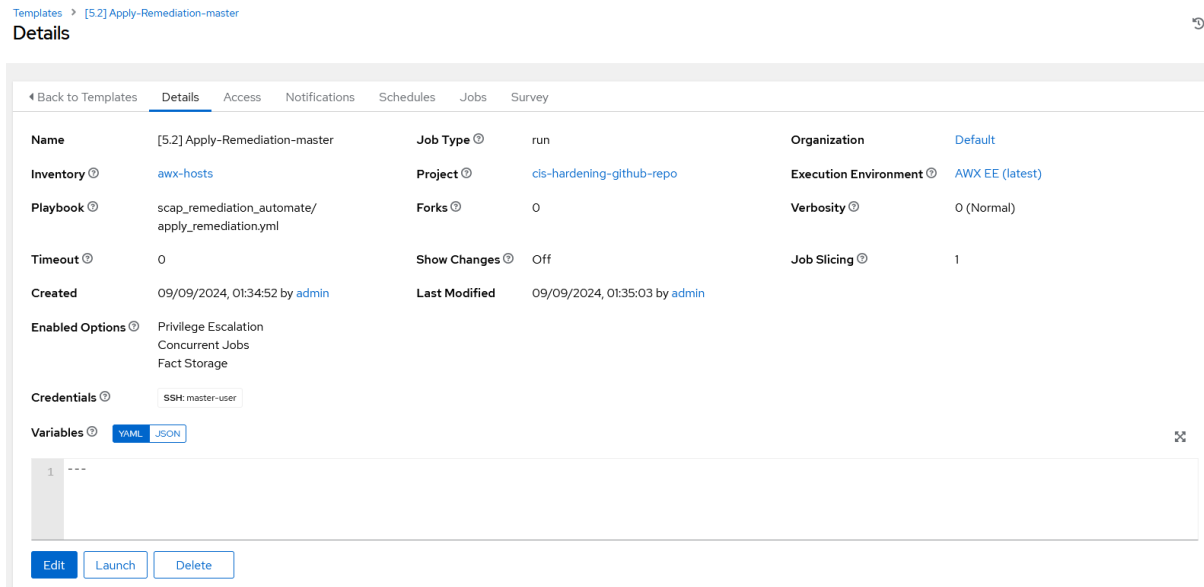


Figura 4.22: Template para aplicar *playbooks* de remediação no servidor principal

Templates Adicionais

Além das *templates* voltadas para remediação, foram criadas duas *templates* adicionais para auxiliar na gestão dos sistemas, permitindo realizar ações de reinício e de encerramento de forma automatizada. Estas *templates* podem ser integradas em *workflows*, facilitando o reinício ou o encerramento automático dos *hosts*, sem necessidade de intervenção manual.

- **REBOOT-hosts:** Esta template foi criada para reiniciar os hosts automaticamente, eliminando a necessidade de o fazer manualmente em cada sistema. Ela utiliza o *playbook* `reboot_hosts.yml` para reiniciar os hosts de forma controlada e eficiente.
- **SHUTDOWN-hosts:** Esta template permite o encerramento automático dos hosts, sendo útil em cenários onde é necessário realizar o encerramento em massa dos sistemas. O *playbook* utilizado é o `poweroff_hosts.yml`, garantindo que o processo de encerramento seja feito de maneira adequada.

Templates > REBOOT-hosts

Details

◀ Back to Templates Details Access Notifications Schedules Jobs Survey

Name	REBOOT-hosts	Job Type ⓘ	run	Organization	Default
Inventory ⓘ	awx-hosts	Project ⓘ	cis-hardening-github-repo	Execution Environment ⓘ	AWX EE (latest)
Playbook ⓘ	reboot_hosts.yml	Forks ⓘ	0	Limit ⓘ	server1,server2,desktop1
Verbosity ⓘ	0 (Normal)	Timeout ⓘ	0	Show Changes ⓘ	Off
Job Slicing ⓘ	1	Created	31/08/2024, 01:41:36 by admin	Last Modified	02/09/2024, 04:29:39 by admin
Enabled Options ⓘ	Privilege Escalation Fact Storage				
Credentials ⓘ	SSH: server-user				
Variables ⓘ	YAML JSON				

1

Edit

Launch

Delete

Figura 4.23: Template para reiniciar os hosts.

Templates > SHUTDOWN-hosts

Details

◀ Back to Templates Details Access Notifications Schedules Jobs Survey

Name	SHUTDOWN-hosts	Job Type ⓘ	run	Organization	Default
Inventory ⓘ	awx-hosts	Project ⓘ	cis-hardening-github-repo	Execution Environment ⓘ	AWX EE (latest)
Playbook ⓘ	poweroff_hosts.yml	Forks ⓘ	0	Limit ⓘ	server1,server2,desktop1
Verbosity ⓘ	0 (Normal)	Timeout ⓘ	0	Show Changes ⓘ	Off
Job Slicing ⓘ	1	Created	31/08/2024, 03:33:46 by admin	Last Modified	02/09/2024, 04:29:47 by admin
Enabled Options ⓘ	Privilege Escalation Fact Storage				
Credentials ⓘ	SSH: server-user				
Variables ⓘ	YAML JSON				

1

Edit

Launch

Delete

Figura 4.24: Template para encerrar os hosts.

4.3.6 Workflow *templates*

Outra grande funcionalidade do *AWX* é a capacidade de conectar as *templates*, de modo que após uma execução, dependendo do valor obtido, seja possível criar fluxos de execução. Na Figura 4.24, essa função é demonstrada por meio de um *workflow* para *hardening* dos servidores com todas as fases. E o mesmo *workflow* (com *templates* diferentes) podia ser utilizado para aplicar *hardening* no *desktop* e no servidor principal, automatizando assim ainda mais a configuração segura dos sistemas.

- **1º Fase** - Sincronizar lista de servidores com o ficheiro de *hosts* no repositório.
- **2º Fase** - Aplicar *hardening* nos servidores com o role '*cis-hardening*'.
- **3º Fase** - Executar análise de vulnerabilidades com a ferramenta *Openscap*.
- **4º Fase** - Aplicar remediação de acordo com o relatório gerado no passo anterior.
- **5º Fase** - Reiniciar servidores
- **6º Fase** - Voltar a executar análise de vulnerabilidades com a ferramenta *Openscap* para comparar com a análise anterior.

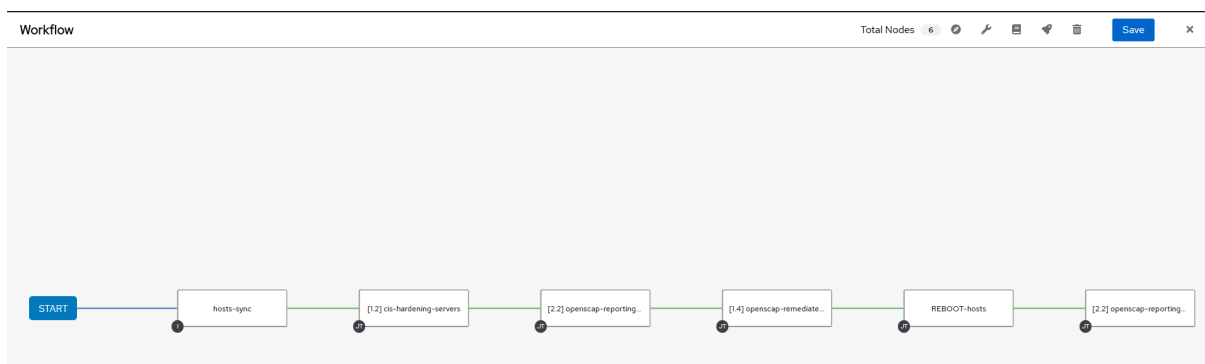


Figura 4.25: Workflow para robustecer a segurança em servidores

4.4 Instalação e configuração dos serviços nos servidores

Antes de detalhar o processo de instalação dos serviços nos servidores, é importante contextualizar a função de cada ambiente. A configuração dos servidores foi realizada com base em cenários amplamente utilizados, visando atender a requisitos específicos de hospedagem web e análise de dados.

O servidor 1 foi configurado para hospedar páginas web dinâmicas, utilizando um conjunto de serviços amplamente utilizados em servidores Linux, composta por *Nginx*, *MySQL* e *PHP*. Este conjunto de ferramentas permite a implementação de um ambiente robusto e escalável, adequado para a gestão de conteúdo web por meio de plataformas como o *WordPress*.

O servidor 2, por sua vez, foi destinado à partilha de ficheiros e à gestão de redes. Para isso, optou-se pela instalação do Samba, uma solução que permite a interoperabilidade entre sistemas Linux e Windows, facilitando a partilha de ficheiros numa rede.

Com essa base definida, a seguir serão descritos os procedimentos de instalação e configuração realizados em cada servidor.

4.4.1 Servidor 1

- **Instalar nginx**

```
$ sudo apt install nginx
```

- **Instalar MySQL**

```
$ sudo apt install mysql-server
```

- **Instalar PHP e módulos necessários**

```
$ sudo apt install php-fpm php-mysql php-cli php-xml php-  
mbstring php-curl
```

- **Instalar WordPress**

```
$ wget https://wordpress.org/latest.tar.gz  
$ tar xzvf latest.tar.gz
```

- **Mover os arquivos do WordPress para o ficheiro raiz do Nginx**

```
$ sudo mv wordpress/* /var/www/html/  
$ sudo chown -R www-data:www-data /var/www/html/
```

- **Criar o base de dados e o utilizador do WordPress no MySQL**

```
$ sudo mysql  
$ CREATE DATABASE wordpress;  
$ CREATE USER 'wordpressuser'@'192.168.1.151' IDENTIFIED BY  
  'xyz123';  
$ GRANT ALL PRIVILEGES ON wordpress.* TO 'wordpressuser'@'  
  192.168.1.151';  
$ FLUSH PRIVILEGES;
```


- Adicionar regras no UFW para permitir tráfego HTTP e HTTPS

```
$ sudo ufw allow 'Nginx_Full'
```

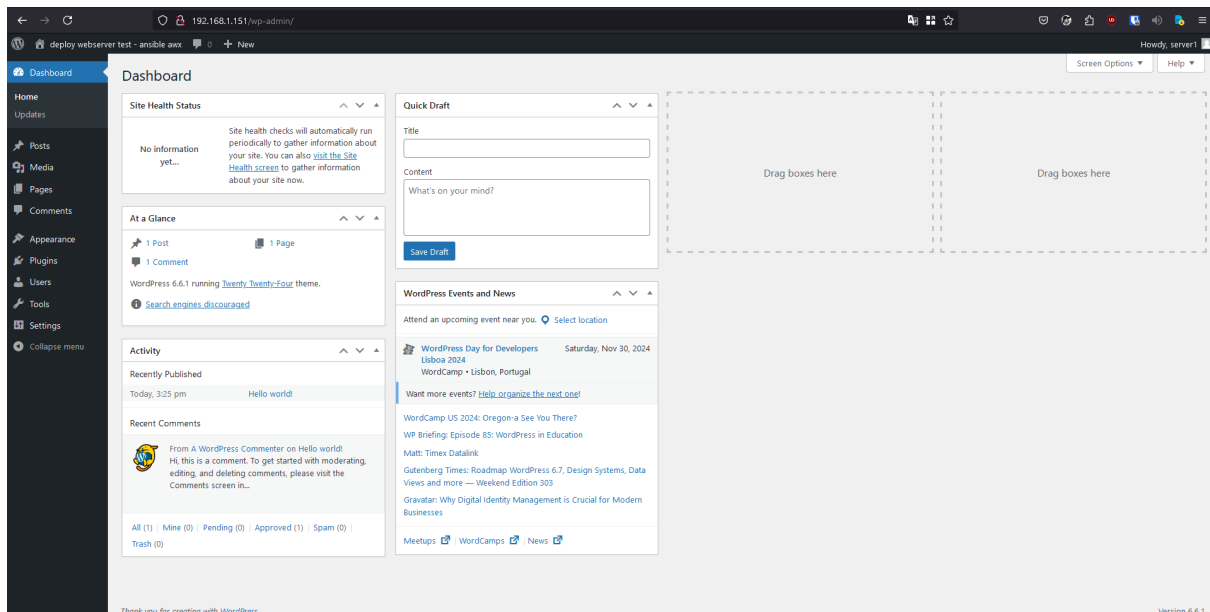


Figura 4.26: WebServer server1

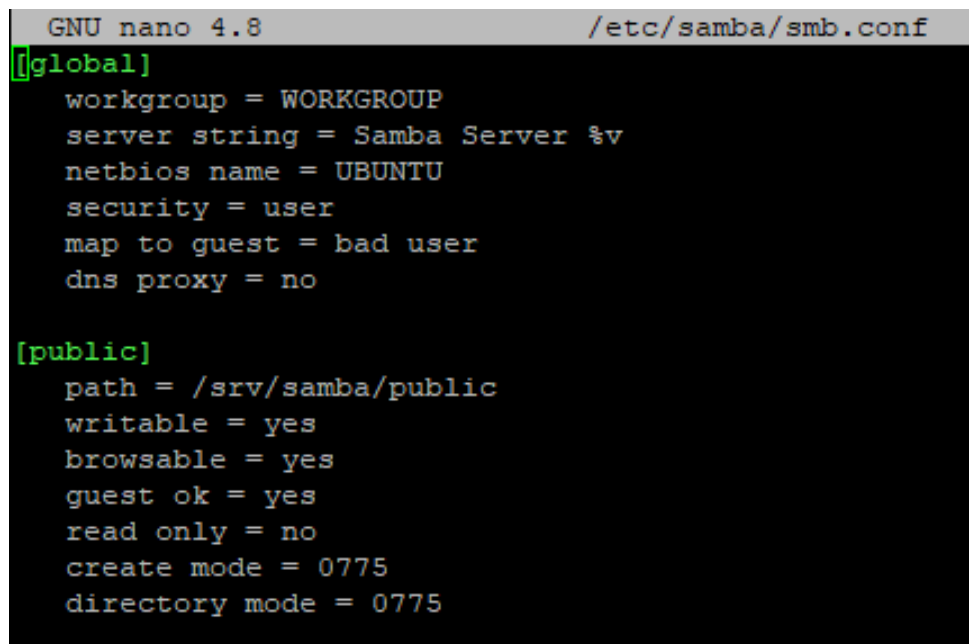
4.4.2 Servidor 2

- Instalar Samba

```
$ sudo apt-get install samba
```

- Editar o arquivo de configuração do Samba

```
$ sudo nano /etc/samba/smb.conf
```

A screenshot of a terminal window showing the contents of the /etc/samba/smb.conf file. The terminal has a black background with green and white text. The title bar at the top reads 'GNU nano 4.8' on the left and '/etc/samba/smb.conf' on the right. The file content is organized into two sections: '[global]' and '[public]'. The '[global]' section contains settings for the workgroup, server string, netbios name, security, map to guest, and dns proxy. The '[public]' section contains settings for the path, writability, browsability, guest access, read-only status, and file/directory permissions.

```
GNU nano 4.8 /etc/samba/smb.conf
[global]
    workgroup = WORKGROUP
    server string = Samba Server %v
    netbios name = UBUNTU
    security = user
    map to guest = bad user
    dns proxy = no

[public]
    path = /srv/samba/public
    writable = yes
    browsable = yes
    guest ok = yes
    read only = no
    create mode = 0775
    directory mode = 0775
```

Figura 4.27: Arquivo de configuração Samba

- Crie o ficheiro para partilha

```
$ sudo mkdir -p /srv/samba/public
```

```
$ sudo chmod 0775 /srv/samba/public
```

```
$ sudo chown nobody:nogroup /srv/samba/public
```

- Reiniciar os serviços do Samba para aplicar as novas configurações

```
$ sudo systemctl restart smbd
```

```
$ sudo systemctl restart nmbd
```

- Aceder o ficheiro partilhado a partir do Windows

```
$ \\192.168.1.152
```

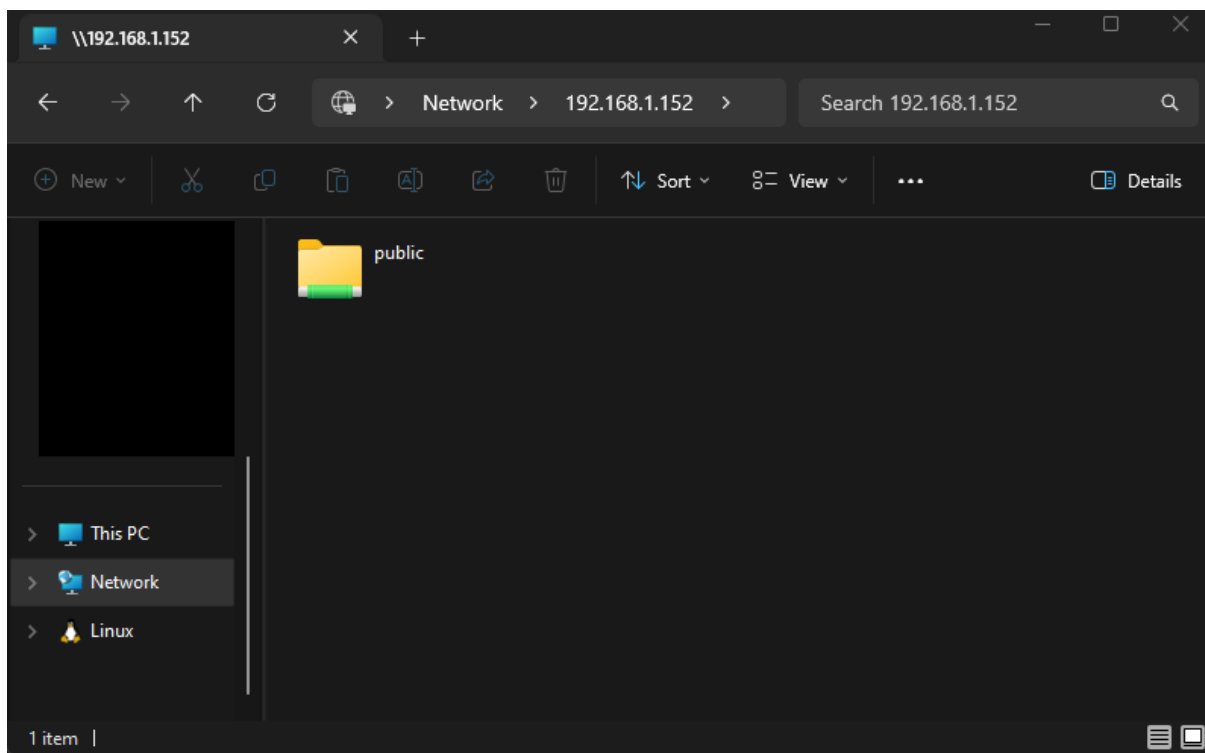


Figura 4.28: Samba

4.5 Transferência de ficheiros para o local host

Como o servidor principal não possui uma interface gráfica, foi necessário transferir os relatórios para outro sistema para facilitar sua análise e visualização. Para isso, foi utilizado o *Windows Subsystem for Linux (WSL)*, que permite a execução de um ambiente Linux diretamente no Windows via terminal. A transferência dos arquivos foi realizada de forma rápida e eficiente utilizando comandos FTP, copiando os relatórios do servidor principal para a máquina com WSL.

Os seguintes comandos permitem copiar todos os arquivos das pastas `/home/master/reports/` e `/home/master/remediation/` do servidor principal, onde tanto os relatórios como os playbooks de remediação gerados pelo *OpenSCAP* estão armazenados, para a máquina WSL:

```
$ sudo scp -r master@192.168.1.150:/home/master/reports/* /  
home/tom/reports  
$ sudo scp -r master@192.168.1.150:/home/master/remediation  
/* /home/tom/remediation
```

Após a transferência, basta inserir o seguinte comando no explorador de arquivos do Windows para aceder aos documentos armazenados na máquina WSL:

```
$ \wsl$\Ubuntu-20.04\home\tom\
```

Capítulo 5

Testes

Durante a execução destes *playbooks*, foram encontrados vários problemas. Um dos primeiros ocorreu ao executar o *playbook* num ambiente *desktop* (*desktop1*), onde o ambiente gráfico foi removido. Para resolver isso, foi necessário adicionar variáveis no repositório, no ficheiro *"host_vars/desktop1/cis.yml"*, para desativar essa configuração. Também foi necessário desativar outras variáveis devido aos mesmos problemas relacionados com o ambiente gráfico.

Algumas variáveis foram desativadas/ativadas para executar o *playbook* no ambiente *desktop*:

- **ubtu20cis__desktop__required: true**
- **ubtu20cis__rule__1__8__2: false**
- **ubtu20cis__rule__1__8__4: false**
- **ubtu20cis__rule__1__8__5: false**
- **ubtu20cis__rule__1__8__6: false**
- **ubtu20cis__rule__1__8__7: false**
- **ubtu20cis__rule__1__8__8: false**

- **ubtu20cis_rule_1_8_9: false**

Em relação ao servidor principal (*master*) também foi necessário desativar algumas variáveis do *playbook*, neste caso relacionadas com configurações da rede e configurações do *firewall*, mais concretamente desativar redes não utilizadas, protocolos de rede invulgares e configuração de *firewall*, que interferiam com o *ansible awx*, foi então desativada toda a secção 3 do *playbook* "*ubuntu-lockdown*" com a seguinte variável:

- **ubtu20cis_section3_patch: false**

5.0.1 Teste para a configuração base das máquinas

Nas seguintes imagens, estão os testes realizados nos *hosts* e no servidor principal, ainda sem a execução de qualquer *playbook* de *hardening*.

Evaluation Characteristics

Evaluation target	master
Benchmark URL	/usr/share/xml/scap/ssg/content/scap-security-guide-0.1.69/ssg-ubuntu2004-ds-1.2.xml
Benchmark ID	xccdf_org.ssgproject.content_benchmark_UBUNTU_20
Profile ID	xccdf_org.ssgproject.content_profile_cis_level2_server
Started at	2024-09-06T02:06:21
Finished at	2024-09-06T02:11:10
Performed by	master

CPE Platforms

• cpe:/o:canonical:ubuntu_linux:20.04:--fls--

Addresses

- IPv4 127.0.0.1
- IPv4 192.168.1.150
- IPv4 10.42.0.0
- IPv4 10.42.0.1
- IPv6 0:0:0:0:0:0:1
- IPv6 2a01:14:100:5b60:a00:27ff:feb0:30e1
- IPv6 fe80:0:0:0:a00:27ff:feb0:30e1
- IPv6 fe80:0:0:0:901e:d4ff:fe93:f050
- IPv6 fe80:0:0:0:8433:7dff:fe9e:60ce
- IPv6 fe80:0:0:0:8c5b:6eff:fe05:3d8c
- IPv6 fe80:0:0:0:143d:f2ff:fe24:3d52
- IPv6 fe80:0:0:0:7c04:f7ff:fed7:287f
- IPv6 fe80:0:0:0:e86b:9ff:fea6:b0d8
- IPv6 fe80:0:0:0:44c0:b7ff:fe6c:a292
- IPv6 fe80:0:0:0:e491:5bff:fe14:c482
- IPv6 fe80:0:0:0:d826:26ff:fe04:81ea
- IPv6 fe80:0:0:0:60b6:56ff:fe9e:1d3d
- IPv6 fe80:0:0:0:886c:a5ff:fee0:1a7f
- IPv6 fe80:0:0:0:58f4:a5ff:fe83:40a5
- MAC 00:00:00:00:00:00
- MAC 08:00:27:B0:30:E1
- MAC 92:1E:D4:93:F0:50
- MAC 86:33:7D:FE:6C:CE
- MAC 6E:5B:6E:05:3D:8C
- MAC 16:3D:F2:24:3D:52
- MAC 7E:04:F7:D7:28:7F
- MAC EA:6B:09:A6:B0:D8
- MAC 46:C0:B7:6C:A2:92
- MAC E8:91:5B:F4:C4:82
- MAC DA:26:26:C4:81:EA
- MAC 62:B6:56:9E:1D:3D
- MAC 8A:6C:AF:E0:1A:7F
- MAC 5A:F4:A5:83:40:A5

Compliance and Scoring

The target system did not satisfy the conditions of 125 rules! Please review rule results and consider applying remediation.

Rule results



Severity of failed rules



Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	63.243164	100.000000	63.24%

Figura 5.1: OpenSCAP relatório servidor principal (master)

Evaluation Characteristics

Evaluation target	server1	CPE Platforms	<ul style="list-style-type: none">cpe:/o:canonical:ubuntu_linux:20.04:-its-	Addresses	<ul style="list-style-type: none">IPv4 127.0.0.1IPv4 192.168.1.151IPv6 0:0:0:0:0:0:1IPv6 2a01:14:100:5b60:a00:27ff:fe27:fbf7IPv6 fe80:0:0:a00:27ff:fe27:fbf7MAC 00:00:00:00:00:00MAC 08:00:27:27:FB:D7
Benchmark URL	/usr/share/xml/scap/ssg/content/scap-security-guide-0.1.69/ssg-ubuntu2004-ds-1.2.xml				
Benchmark ID	xccdf_org.ssgproject.content_benchmark_UBUNTU_20-04				
Profile ID	xccdf_org.ssgproject.content_profile_cis_level1_server				
Started at	2024-09-05T22:52:37				
Finished at	2024-09-05T22:53:31				
Performed by	server				

Compliance and Scoring

The target system did not satisfy the conditions of 107 rules! Please review rule results and consider applying remediation.

Rule results



Severity of failed rules



Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	67.383125	100.000000	67.38%

Figura 5.2: OpenSCAP relatório hosts (server1)

Evaluation Characteristics

Evaluation target	server2	CPE Platforms	<ul style="list-style-type: none">cpe:/o:canonical:ubuntu_linux:20.04:-its-	Addresses	<ul style="list-style-type: none">IPv4 127.0.0.1IPv4 192.168.1.152IPv6 0:0:0:0:0:0:1IPv6 2a01:14:100:5b60:a00:27ff:fe53:8873IPv6 fe80:0:0:a00:27ff:fe53:8873MAC 00:00:00:00:00:00MAC 08:00:27:53:88:73
Benchmark URL	/usr/share/xml/scap/ssg/content/scap-security-guide-0.1.69/ssg-ubuntu2004-ds-1.2.xml				
Benchmark ID	xccdf_org.ssgproject.content_benchmark_UBUNTU_20-04				
Profile ID	xccdf_org.ssgproject.content_profile_cis_level1_server				
Started at	2024-09-05T22:52:37				
Finished at	2024-09-05T22:53:30				
Performed by	server				

Compliance and Scoring

The target system did not satisfy the conditions of 107 rules! Please review rule results and consider applying remediation.

Rule results



Severity of failed rules



Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	67.383125	100.000000	67.38%

Figura 5.3: OpenSCAP relatório hosts (server2)

Evaluation Characteristics

Evaluation target	server3Desktop	CPE Platforms	Addresses
Benchmark URL	/usr/share/xml/scap/ssg/content/scap-security-guide-0.1.65-ssg-ubuntu2004-ds-1.2.xml	• cpe:/o:canonical:ubuntu_finnix:20.04c--fts-	• IPv4 127.0.0.1
Benchmark ID	xccdf_org.ssgproject.content_benchmark_UBUNTU_20-04		• IPv4 192.168.1.153
Profile ID	xccdf_org.ssgproject.content_profile_cis_level1_workstation		• IPv6 0.0.0.0.0.0.1
Started at	2024-09-06T00:11:09		• IPv6 2a01:14:100:5b60:a00:27ff:febc:be29
Finished at	2024-09-06T00:12:45		• IPv6 fe80:0:0:a00:27ff:febc:be29
Performed by	server		• MAC 00:00:00:00:00:00
			• MAC 08:00:27:BC:BE:29

Compliance and Scoring

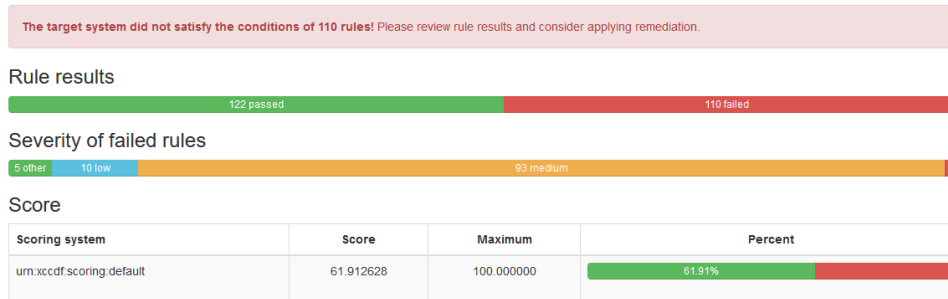


Figura 5.4: OpenSCAP relatório hosts (desktop1)

Para o robustecimento dos sistemas, foi inicialmente utilizado o repositório *ansible-lockdown*, criando o *role cis-hardening* para melhorar a segurança dos sistemas de acordo com os CIS Benchmarks. A seguir, estão descritos os *templates* e *playbooks* utilizados em cada caso:

- Para os hosts (*desktop1*, *servidor1*, *servidor2*), foi criada a template *cis-hardening-servers*, utilizando o *playbook hardening.yml*.
- Para o servidor principal (*master*), foi criada a template *cis-hardening-master*, utilizando o *playbook hardening.yml*.

A única diferença entre estas duas *templates* são as credenciais, como já foi citado anteriormente todos os hosts têm um utilizador em comum (*server*) para facilitar o *deploy* dos *playbooks* em simultâneo em todos os *hosts*, por motivos de segurança o mesmo não se aplica ao servidor principal, que tem umas credenciais diferentes.

Nas seguintes imagens é possível observar o output do processo de *hardening* dos *hosts* e do servidor principal:

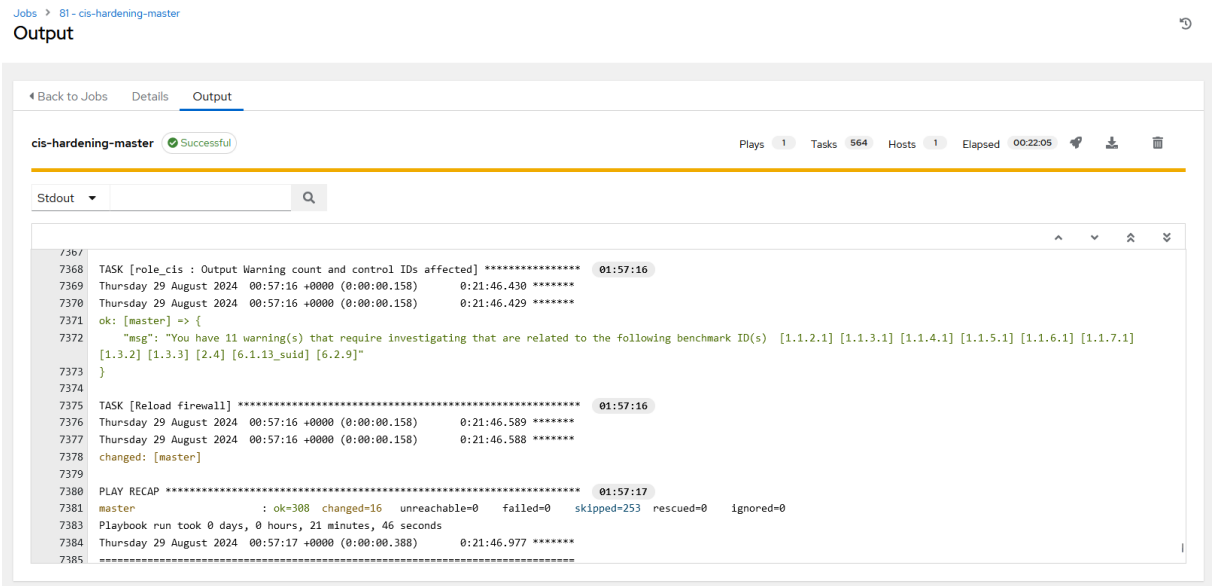


Figura 5.5: Output hardening do servidor principal (master)

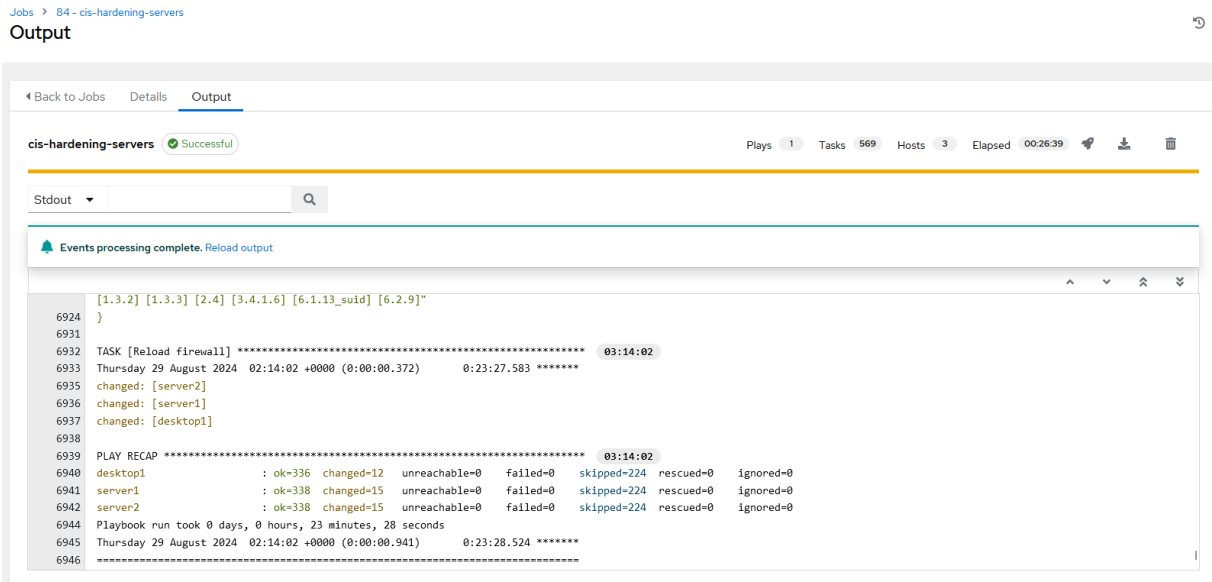


Figura 5.6: Output hardening dos hosts (servidor1, servidor2, desktop1)

Depois de executar os *playbooks* foi realizado outro teste de segurança, como já foi realizado anteriormente, para verificar o impacto da execução do *playbook* de hardening.

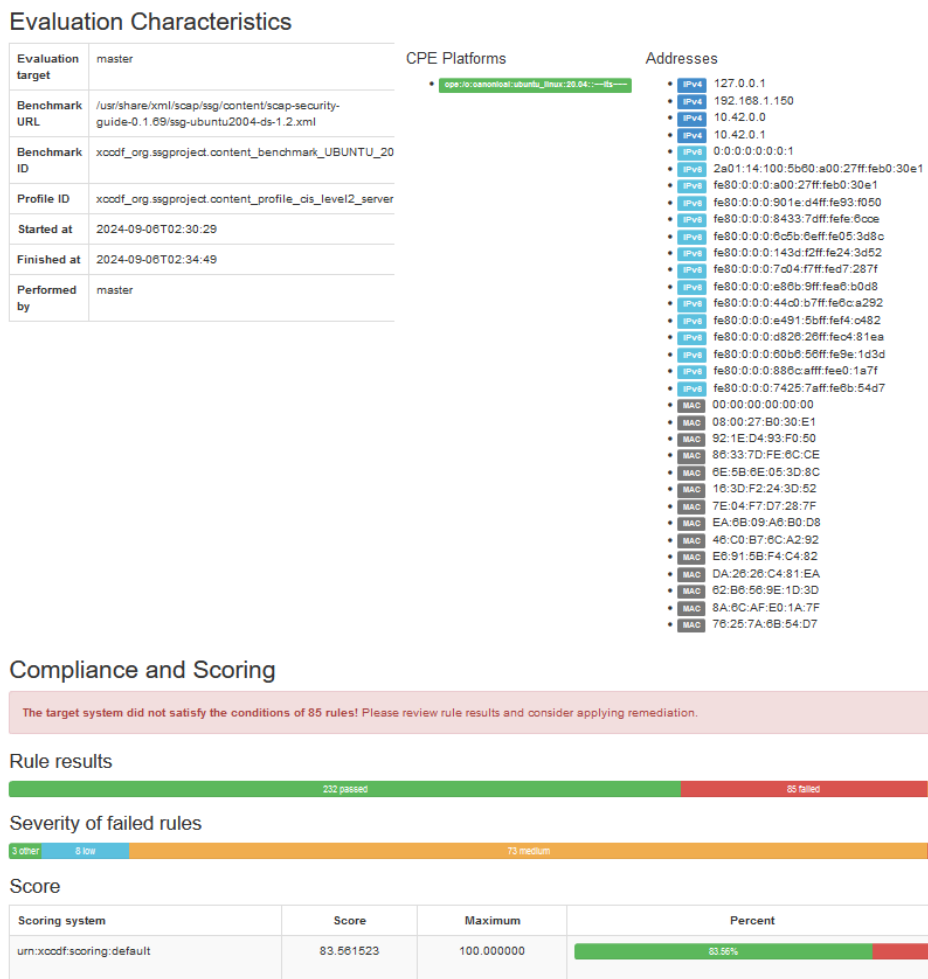


Figura 5.7: OpenSCAP relatório servidor principal (master)

Evaluation Characteristics

Evaluation target	server1	CPE Platforms	Addresses
Benchmark URL	/usr/share/xml/scap/ssg/content/scap-security-guide-0.1.69/ssg-ubuntu2004-ds-1.2.xml	<ul style="list-style-type: none"> cpe:/o:canonical:ubuntu_linux:20.04:-:its- 	<ul style="list-style-type: none"> IPv4 127.0.0.1 IPv4 192.168.1.151 IPv6 0.0.0.0.0.0.1 IPv6 2a01:14:100:5b60:a00:27ff:fe27:1bd7 IPv6 fe80:0:0:a00:27ff:fe27:1bd7 MAC 00:00:00:00:00:00 MAC 08:00:27:27:FB:D7
Benchmark ID	xccdf_org.ssgproject.content_benchmark_UBUNTU_20-04		
Profile ID	xccdf_org.ssgproject.content_profile_cis_level1_server		
Started at	2024-09-05T23:02:32		
Finished at	2024-09-05T23:03:10		
Performed by	server		

Compliance and Scoring

The target system did not satisfy the conditions of 32 rules! Please review rule results and consider applying remediation.

Rule results



Severity of failed rules



Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	85.375961	100.000000	85.38%

Figura 5.8: OpenSCAP relatório hosts (server1)

Evaluation Characteristics

Evaluation target	server2	CPE Platforms	Addresses
Benchmark URL	/usr/share/xml/scap/ssg/content/scap-security-guide-0.1.69/ssg-ubuntu2004-ds-1.2.xml	<ul style="list-style-type: none"> cpe:/o:canonical:ubuntu_linux:20.04:-:its- 	<ul style="list-style-type: none"> IPv4 127.0.0.1 IPv4 192.168.1.152 IPv6 0.0.0.0.0.0.1 IPv6 2a01:14:100:5b60:a00:27ff:fe53:8873 IPv6 fe80:0:0:a00:27ff:fe53:8873 MAC 00:00:00:00:00:00 MAC 08:00:27:53:88:73
Benchmark ID	xccdf_org.ssgproject.content_benchmark_UBUNTU_20-04		
Profile ID	xccdf_org.ssgproject.content_profile_cis_level1_server		
Started at	2024-09-05T23:02:32		
Finished at	2024-09-05T23:03:09		
Performed by	server		

Compliance and Scoring

The target system did not satisfy the conditions of 32 rules! Please review rule results and consider applying remediation.

Rule results



Severity of failed rules



Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	85.375961	100.000000	85.38%

Figura 5.9: OpenSCAP relatório hosts (server2)

Evaluation Characteristics

Evaluation target	server3Desktop	CPE Platforms	Addresses
Benchmark URL	/usr/share/xml/scap/ssg/content/scap-security-guide-0.1.69/ssg-ubuntu2004-ds-1.2.xml	<ul style="list-style-type: none"> cpe:/o:canonical:ubuntu_linux:20:04:~:its--- 	<ul style="list-style-type: none"> IPv4 127.0.0.1 IPv4 192.168.1.153 IPv6 0:0:0:0:0:0:1 IPv6 2a01:14:100:5b60:a00:27ff:febc:be29 IPv6 fe80:0:0:0:a00:27ff:febc:be29 MAC 00:00:00:00:00:00 MAC 08:00:27:BC:BE:29
Benchmark ID	xccdf_org.ssgproject.content_benchmark_UBUNTU_20-0		
Profile ID	xccdf_org.ssgproject.content_profile_cis_level1_workstat		
Started at	2024-09-06T00:30:40		
Finished at	2024-09-06T00:32:00		
Performed by	server		

Compliance and Scoring

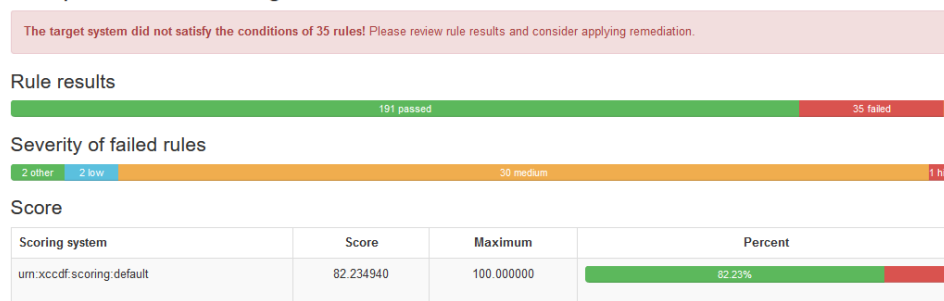


Figura 5.10: OpenSCAP relatório hosts (desktop1)

Após a análise dos relatórios, foi possível observar que os sistemas não estavam completamente seguros, uma vez que a pontuação recomendada para considerar um sistema como seguro é superior a 85. Para melhorar a segurança dos sistemas, foi necessário mitigar as vulnerabilidades que não foram resolvidas pelos *playbooks* de hardening.

Para criar os *playbooks* de mitigação das vulnerabilidades, foi utilizada o *OpenSCAP*. O comando *oscap xccdf generate fix* gera um *playbook* Ansible personalizado com base na configuração atual do sistema e no perfil de segurança especificado (CIS) neste caso. Os seguintes comandos foram utilizados para gerar os *playbooks* para os *hosts* (*servers*), *hosts* (*workstation*) e servidor principal (*master*):

Gerar *playbook* para os hosts (desktop1):

```
$ oscap xccdf generate fix --profile xccdf_org.ssgproject.
content_profile_cis_level1_workstation --template urn:
xccdf:fix:script:ansible --output remediation.yml /usr/
share/xml/scap/ssg/content/scap-security-guide-0.1.69/ssg
-ubuntu2004-ds-1.2.xml
```

Gerar *playbook* para os hosts (server1, server2):

```
$ oscap xccdf generate fix --profile xccdf_org.ssgproject.
content_profile_cis_level1_server --template urn:xccdf:
fix:script:ansible --output remediation.yml /usr/share/
xml/scap/ssg/content/scap-security-guide-0.1.69/ssg-
ubuntu2004-ds-1.2.xml
```

Gerar *playbook* para o servidor principal (master):

```
$ oscap xccdf generate fix --profile xccdf_org.ssgproject.
content_profile_cis_level2_server --template urn:xccdf:
fix:script:ansible --output remediation.yml /usr/share/
xml/scap/ssg/content/scap-security-guide-0.1.69/ssg-
ubuntu2004-ds-1.2.xml
```

Foram criados *playbooks* para automatizar este processo:

- scap_remediation_generator_servers.yml
- scap_remediation_generator_desktop.yml
- scap_remediation_generator_master.yml

Que geram os *playbooks* para remediar as vulnerabilidades do sistema e enviam esses *playbooks* para o servidor principal (master). Um dos objetivos não conseguidos foi este, porque depois deste processo foi criado outro *playbook* "apply_remediation.yml", que tinha

como objetivo executar esses *playbooks* nos hosts desejados, mas os *playbooks* gerados continham erros de formatação e era necessário corrigi-los um a um.

Então como não foi possível utilizar o *playbook* para automatizar este processo de aplicação da remediação, foi necessário adicionar os *playbooks* gerados no repositório github utilizado neste projeto, criar uma template e executá-los a partir do AWX.

Depois de *excutar* os *playbooks* foi executado outro teste de segurança para comparar com os resultados anteriores.

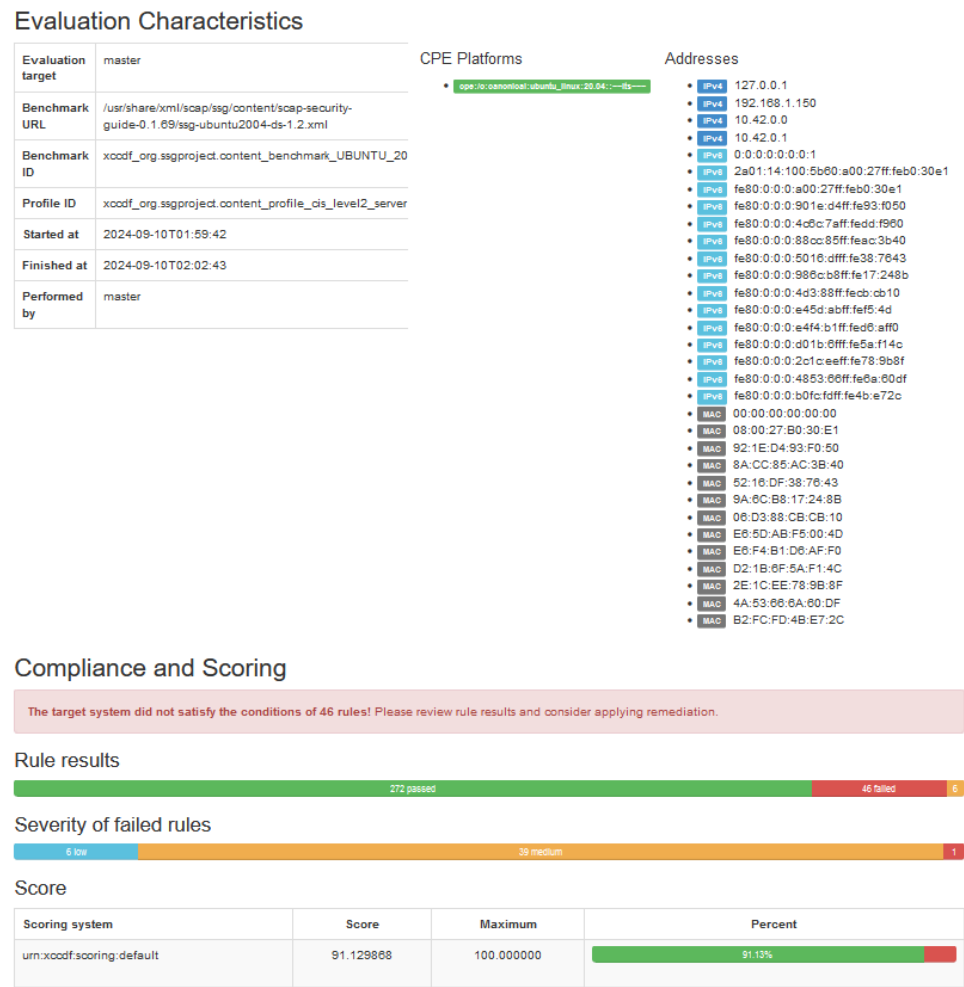


Figura 5.11: OpenSCAP relatório (master)

Evaluation Characteristics

Evaluation target	server1
Benchmark URL	/usr/share/xml/scap/ssg/content/scap-security-guide-0.1.69/ssg-ubuntu2004-ds-1.2.xml
Benchmark ID	xccdf_org.ssgproject.content_benchmark_UBUNTU_20-04
Profile ID	xccdf_org.ssgproject.content_profile_cis_level1_server
Started at	2024-09-06T01:34:54
Finished at	2024-09-06T01:35:32
Performed by	server

CPE Platforms

- cpe:ocanonical:ubuntu_linux:20.04:-its-

Addresses

- IPv4 127.0.0.1
- IPv4 192.168.1.151
- MAC 00:00:00:00:00:00
- MAC 08:00:27:27:FB:D7

Compliance and Scoring

The target system did not satisfy the conditions of 12 rules! Please review rule results and consider applying remediation.

Rule results

216 passed12 failed3

Severity of failed rules

1 low10 medium1 high

Score

Scoring system	Score	Maximum	Percent
um:xccdf.scoring.default	92.329086	100.000000	92.33%

Figura 5.12: OpenSCAP relatório hosts (server1)

Evaluation Characteristics

Evaluation target	server2
Benchmark URL	/usr/share/xml/scap/ssg/content/scap-security-guide-0.1.69/ssg-ubuntu2004-ds-1.2.xml
Benchmark ID	xccdf_org.ssgproject.content_benchmark_UBUNTU_20-04
Profile ID	xccdf_org.ssgproject.content_profile_cis_level1_server
Started at	2024-09-06T01:34:54
Finished at	2024-09-06T01:35:33
Performed by	server

CPE Platforms

- cpe:ocanonical:ubuntu_linux:20.04:-its-

Addresses

- IPv4 127.0.0.1
- IPv4 192.168.1.152
- MAC 00:00:00:00:00:00
- MAC 08:00:27:53:88:73

Compliance and Scoring

The target system did not satisfy the conditions of 12 rules! Please review rule results and consider applying remediation.

Rule results

216 passed12 failed3

Severity of failed rules

1 low10 medium1 high

Score

Scoring system	Score	Maximum	Percent
um:xccdf.scoring.default	92.329086	100.000000	92.33%

Figura 5.13: OpenSCAP relatório hosts (server2)

Evaluation Characteristics

Evaluation target	server3Desktop	CPE Platforms	Addresses
Benchmark URL	/usr/share/xml/scap/ssg/content/scap-security-guide-0.1.69/ssg-ubuntu2004-ds-1.2.xml	<ul style="list-style-type: none"> cpe:/o:canonical:ubuntu_smxs:20.04:--its-- 	<ul style="list-style-type: none"> IPv4 127.0.0.1 IPv4 192.168.1.153 MAC 00:00:00:00:00:00 MAC 08:00:27:BC:BE:29
Benchmark ID	xccdf_org.ssgproject.content_benchmark_UBUNTU_20-04		
Profile ID	xccdf_org.ssgproject.content_profile_cis_level1_workstation		
Started at	2024-09-06T01:30:13		
Finished at	2024-09-06T01:31:44		
Performed by	server		

Compliance and Scoring

The target system did not satisfy the conditions of 16 rules! Please review rule results and consider applying remediation.

Rule results

210 passed 16 failed 3

Severity of failed rules

2 low 13 medium 1 high

Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	90.543671	100.000000	90.54%

Figura 5.14: OpenSCAP relatório hosts (desktop1)

Como é possível observar os resultados de segurança melhoraram significativamente e o objetivo de remediar aquilo que não foi conseguido através do role utilizado foi cumprido, agora com os sistemas com uma pontuação acima dos 90.

Hostname	Fresh Machine	Role Hardening	<i>OpenSCAP</i> Remediation
master	63.243164	83.561523	91.128868
server1	67.383125	85.375961	92.329086
server2	67.383125	85.375961	92.329086
desktop1	61.912628	82.234940	90.543671

Tabela 5.1: Pontuações de segurança nas diferentes fases

5.0.2 Teste para configuração com serviços nos sistemas

Os testes feitos anteriormente foram realizados nos sistemas sem serviços a correr, tendo assim uma superfície de ataque pequena. Nestes testes foram instalados serviços no servidor 1 e no servidor 2. No servidor 1, como uma das configurações mais comuns em servidores Linux para hospedar páginas web dinâmicas, foram instalados o *Nginx*, *MySQL*, *PHP* dando *deploy* do *wordpress*. No Servidor 2, seguindo uma configuração comum para gestão de arquivos e partilha de recursos, foi instalado o Samba, permitindo a criação e a administração de arquivos em rede.

Correndo os mesmo testes feitos anteriormente estes foram os dados gerados:

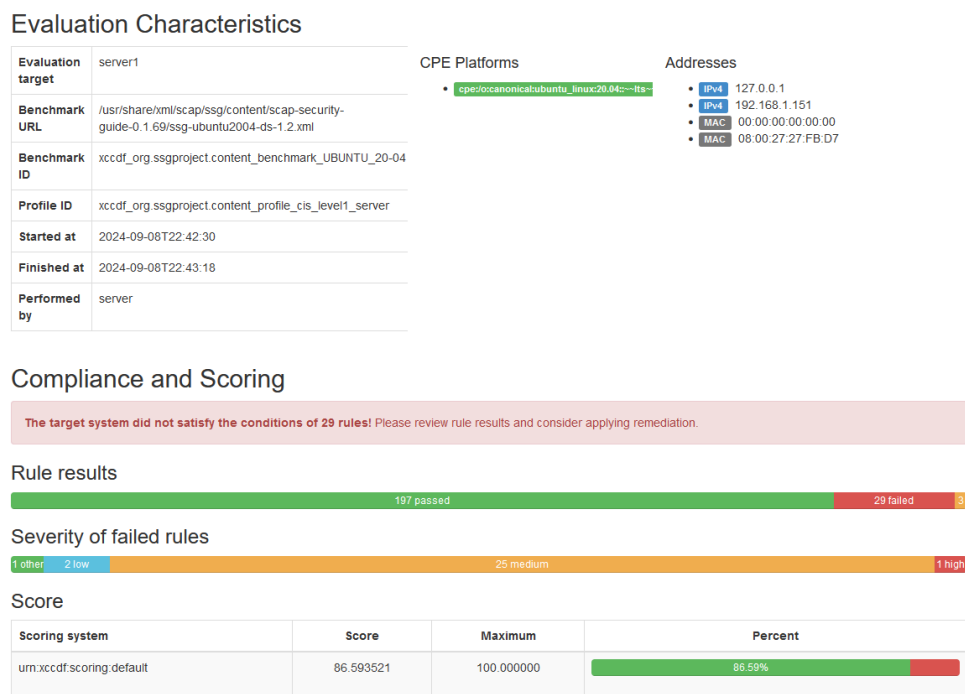


Figura 5.15: OpenSCAP relatório após instalação do serviços no server1

Evaluation Characteristics

Evaluation target	server2
Benchmark URL	/usr/share/xml/scap/ssg/content/scap-security-guide-0.1.69/ssg-ubuntu2004-ds-1.2.xml
Benchmark ID	xccdf_org.ssgproject.content_benchmark_UBUNTU_20-04
Profile ID	xccdf_org.ssgproject.content_profile_cis_level1_server
Started at	2024-09-08T23:29:26
Finished at	2024-09-08T23:30:13
Performed by	server

CPE Platforms

cpe:/o:canonical:ubuntu_linux:20.04

Addresses

IPv4 127.0.0.1
 IPv4 192.168.1.152
 MAC 00:00:00:00:00:00
 MAC 08:00:27:53:88:73

Compliance and Scoring

The target system did not satisfy the conditions of 26 rules! Please review rule results and consider applying remediation.

Rule results

204 passed 26 failed 3

Severity of failed rules

1 other 1 low 23 medium 1 high

Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	89.932266	100.000000	89.93%

Figura 5.16: OpenSCAP relatório após instalação do serviços no server2

Como é possível observar as pontuações de segurança diminuíram nos dois servidores, devido aos serviços que foram instalados, é necessário então gerar outra vez novos *playbooks* de remediação como já foi feito anteriormente para mitigar as novas vulnerabilidades que surgiram depois de instalar estes serviços.

Depois de executar os *playbooks* de remediação, é necessário executar de novo os testes de segurança para comparar os dados com o teste feito depois de instalar os serviços.

Evaluation Characteristics

Evaluation target	server1
Benchmark URL	/usr/share/xml/scap/ssg/content/scap-security-guide-0.1.69/ssg-ubuntu2004-ds-1.2.xml
Benchmark ID	xccdf_org.ssgproject.content_benchmark_UBUNTU_20-04
Profile ID	xccdf_org.ssgproject.content_profile_cis_level1_server
Started at	2024-09-09T10:34:54
Finished at	2024-09-09T10:35:32
Performed by	server

CPE Platforms

• cpe:/o:canonical:ubuntu_linux:20.04 --Itz---

Addresses

• IPv4 127.0.0.1
• IPv4 192.168.1.151
• MAC 00:00:00:00:00:00
• MAC 08:00:27:27:FB:D7

Compliance and Scoring

The target system did not satisfy the conditions of 15 rules! Please review rule results and consider applying remediation.

Rule results

219 passed

15 failed

Severity of failed rules

1 low

13 medium

1 high

Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	92.070986	100.000000	92.07%

Figura 5.17: OpenSCAP relatório após instalação do serviços e remediação no server1

Evaluation Characteristics

Evaluation target	server2	CPE Platforms	Addresses
Benchmark URL	/usr/share/xml/scap/ssg/content/scap-security-guide-0.1.69/ssg-ubuntu2004-ds-1.2.xml	<ul style="list-style-type: none">cpe:/o:canonical:ubuntu_linux:20.04:-its-	<ul style="list-style-type: none">IPv4 127.0.0.1IPv4 192.168.1.152MAC 00:00:00:00:00:00MAC 08:00:27:53:88:73
Benchmark ID	xccdf_org.ssgproject.content_benchmark_UBUNTU_20-04		
Profile ID	xccdf_org.ssgproject.content_profile_cis_level1_server		
Started at	2024-09-09T10:54:26		
Finished at	2024-09-09T10:55:13		
Performed by	server		

Compliance and Scoring

The target system did not satisfy the conditions of 16 rules! Please review rule results and consider applying remediation.

Rule results

217 passed 16 failed 3

Severity of failed rules

1 low 14 medium 1 high

Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	91.432266	100.000000	91.43%

Figura 5.18: OpenSCAP relatório após instalação dos serviços e remediação no server2

Após a execução dos playbooks de remediação, a maioria das vulnerabilidades que surgiram após a instalação dos serviços nos dois servidores foi mitigada, melhorando assim os resultados dos testes de segurança realizados com o *OpenSCAP*. A tabela abaixo apresenta uma comparação entre as três fases deste processo.

O primeiro valor refere-se à configuração inicial, obtido após a aplicação do playbook de hardening com o role *ansible-lockdown* e o playbook de remediação gerado pelo *OpenSCAP*, quando os servidores ainda não tinham serviços instalados, resultando em uma menor superfície de ataque. O segundo valor corresponde ao estado do sistema após a instalação dos serviços, sem a execução de qualquer playbook de hardening, o que resultou em uma queda na pontuação e no aumento do número de vulnerabilidades. Por fim, o último valor mostra os resultados após a geração e aplicação dos playbooks de remediação nos sistemas, evidenciando um aumento na pontuação de segurança e uma diminuição no número de vulnerabilidades detectadas.

Hostname	Configuração Base	Pós Instalação Serviços	Depois da Remediação
server1	92.329086	86.593521	92.070986
server2	92.329086	89.932266	91.432266

Tabela 5.2: Pontuações de segurança nas diferentes fases, comparação com a configuração base

Capítulo 6

Conclusões

Após a execução deste projeto, conclui-se que a automatização do robustecimento de sistemas é uma tarefa crucial para melhorar a segurança de servidores e sistemas em geral. O estudo de metodologias e abordagens para aumentar a resiliência dos sistemas e reduzir a sua superfície de ataque foi fundamental para compreender as opções disponíveis para a implementação de soluções seguras. Este projeto não só foi uma mais valia significativa para a minha preparação para a entrada no mercado de trabalho, como também me proporcionou a oportunidade de trabalhar com tecnologias amplamente utilizadas e requisitadas atualmente. Além disso, a implementação das soluções demonstrou que a automatização permite uma gestão eficiente e escalável dos sistemas, a capacidade de aplicar as mesmas configurações e garantir a conformidade, independentemente do número de servidores, confirma que o processo é igualmente eficiente tanto para pequenos ambientes, como foi o caso durante o projeto, como para grandes ambientes. Tive a oportunidade de aplicar tecnologias de automatização, virtualização, gestão de *containers*, gestão de infraestruturas, conformidade de segurança e administração de sistemas, que aprendi durante o curso de Engenharia Informática e cuja utilização pude integrar no desenvolvimento deste projeto.

Bibliografia

- [1] T. Hsu, *Hands-on security in devops: Ensure continuous security, deployment, and delivery with devsecops*. Packt Publishing, 2018.
- [2] *How To Install and Configure Ansible on Ubuntu 20.04* / DigitalOcean — *digitalocean.com*, <https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-ansible-on-ubuntu-20-04>, [Accessed 26-07-2024].
- [3] *Quick-Start Guide* / K3s — *docs.k3s.io*, <https://docs.k3s.io/quick-start>, [Accessed 25-07-2024].
- [4] *Basic Install - Ansible AWX Operator Documentation* — *ansible.readthedocs.io*, <https://ansible.readthedocs.io/projects/awx-operator/en/latest/installation/basic-install.html>, [Accessed 25-07-2024].
- [5] *Como instalar o Ansible AWX no Ubuntu 22.04/20.04/18.04* — *pt.linux-console.net*, <https://pt.linux-console.net/?p=21347>, [Accessed 25-07-2024].
- [6] *UBUNTU20-CIS/tasks at devel · ansible-lockdown/UBUNTU20-CIS* — *github.com*, <https://github.com/ansible-lockdown/UBUNTU20-CIS/tree/devel/tasks>, [Accessed 09-09-2024].
- [7] *UBUNTU20-CIS/defaults/main.yml at devel · ansible-lockdown/UBUNTU20-CIS* — *github.com*, <https://github.com/ansible-lockdown/UBUNTU20-CIS/blob/devel/defaults/main.yml>, [Accessed 09-09-2024].

Apêndice A

Proposta Original do Projeto



Curso de Licenciatura em Engenharia Informática
Projeto 3º Ano - Ano letivo de 2023/2024

Automatização de configuração segura de sistemas

<input type="checkbox"/> Inteligência artificial	<input type="checkbox"/> Multimédia/Realidade aumentada/...
<input type="checkbox"/> Aplicações móveis	<input type="checkbox"/> Redes e gestão de sistemas
<input type="checkbox"/> Plataformas web	<input checked="" type="checkbox"/> Cibersegurança

* - A especificar pelo proponente

Orientador: Tiago Pedrosa | rftman@ipb.pt

Coorientador: Rui Alves | rui.alves@ipb.pt

1 Objetivo

A configuração segura base de sistemas deve ser consistentemente em todos os sistemas e para minimizar possíveis erros de configurações. É também necessário muitas vezes documentar as discrepâncias entre a configuração que os sistemas deveriam ter e as configurações reais. Este projeto pretende desenvolver uma solução para automatizar estas tarefas em servidores Linux.

2 Detalhes

O principal objetivo é utilizar soluções como o Ansible para automatizar a alteração da configuração dos sistemas com base numa configuração de base que os sistemas devem respeitar, apoiada por boas práticas de segurança. Deve também desenvolver uma solução para reportar as diferenças entre a configuração de base que deve ser respeitada e a configuração real nos sistemas.

3 Metodologia de trabalho

Propõe-se a seguinte metodologia:

1. Análise de boas práticas de configuração de sistemas e serviço em Linux;
2. Análise do estado da arte;
3. Proposta de solução;
4. Implementação;
5. Testes e melhorias;
6. Escrita de documentação técnica e do relatório;

Dimensão da equipa:	2
Recursos necessários:	Computador com capacidade virtualização para desenvolvimento e criação dos cenários

Apêndice B

Repositório GitHub Ansible AWX

No seguinte link está disponível o repositório que fui utilizado para configurar a automação da configuração segura dos sistemas.

Repositório url: <https://github.com/tomaspcosta/linux-hardening-awx>

Apêndice C

Demo Hardening Sistemas

No seguinte link está disponível uma demo do processo completo de *hardening* dos servidores (*hosts*) e do servidor principal.

Demo url: https://drive.google.com/file/d/1fSaLCIpJanbwjs8p3zzf2JRt3B9McSuG/view?usp=drive_link