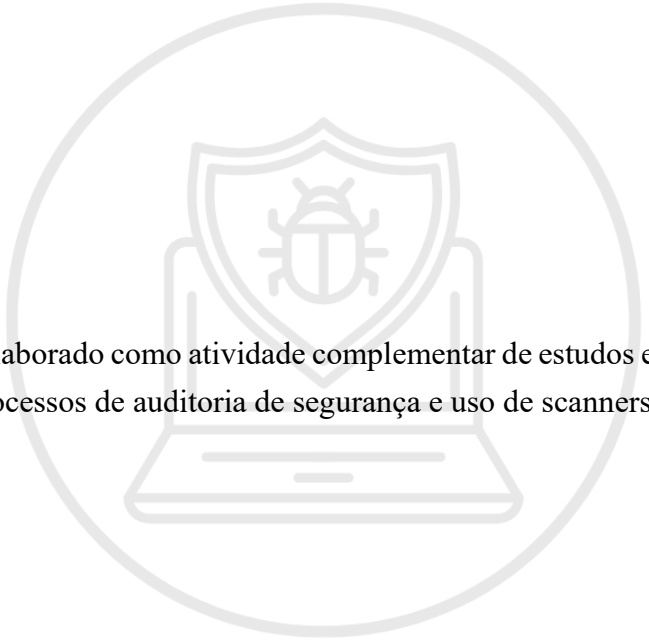
RELATÓRIO TÉCNICO - ANÁLISE DE VULNERABILIDADES COM NESSUS NO METASPLOITABLE

Kássia E. F. Guedes
UNIUBE - Universidade de Uberaba

Campina Grande, PB – 2026

RELATÓRIO TÉCNICO - ANÁLISE DE VULNERABILIDADES COM NESSUS NO METASPLOITABLE

Kássia E. F. Guedes



Relatório técnico elaborado como atividade complementar de estudos em cibersegurança, com foco prático em processos de auditoria de segurança e uso de scanners de vulnerabilidades.

Campina Grande, PB – 2026

SUMÁRIO

1 INTRODUÇÃO

O Nessus representa uma das ferramentas mais avançadas e confiáveis para detecção automatizada de vulnerabilidades em ambientes de TI, sendo amplamente adotado por profissionais de cibersegurança em auditorias e testes de penetração. Este relatório documenta a implementação prática de um laboratório controlado utilizando Kali Linux como plataforma de ataque e Metasploitable 2 como alvo vulnerável, seguindo rigorosamente o fluxograma projetual apresentado.

O estudo prático abrange desde a preparação do ambiente virtualizado até a instalação, configuração e execução do scanner Nessus Essentials. Diferentemente de soluções open-source como OpenVAS, o Nessus oferece maior estabilidade em distribuições Kali Linux, interface web intuitiva e atualizações automáticas de plugins de vulnerabilidade.

Este projeto contribui significativamente para o desenvolvimento de competências práticas em análise de vulnerabilidades, permitindo a identificação de falhas críticas (como CVE-2011-2523 vsftpd backdoor) e sua posterior validação via Metasploit Framework. O ambiente laboratorial simula condições reais de pentest, respeitando boas práticas de isolamento de rede (NAT/Host-Only) para garantir segurança durante os testes.

A metodologia adotada segue padrões profissionais de segurança da informação, com validação de serviços, análise detalhada de resultados por severidade e geração de relatórios formatados segundo normas ABNT, preparando o profissional para atuar em cenários corporativos reais de cibersegurança.

2 DESENVOLVIMENTO

2.1 Preparação do Ambiente Laboratorial

O projeto iniciou com a configuração de um ambiente virtual seguro, essencial para testes de cibersegurança. Foram utilizadas duas máquinas virtuais no VirtualBox:

Máquina	Sistema Operacional	Função no Projeto
Kali Linux	Kali Linux 2025.3	Hospedagem do Nessus
Metasploitable 2	Ubuntu 8.04 (Linux)	Alvo vulnerável para testes

Tabela 1 - Configuração das Máquinas Virtuais

Configuração de rede: Kali Linux com Adaptador 1 em modo NAT e Adaptador 2 em modo Rede interna, Metasploitable com Adaptador 1 em modo Host-Only garantindo isolamento total da rede externa.

2.2 Instalação do Nessus no Kali Linux

A instalação seguiu rigorosamente a documentação oficial da Tenable:

1. Atualização do sistema

```
sudo apt update && sudo apt upgrade -y
```

2. Download do pacote .deb mais recente

```
wget https://www.tenable.com/downloads/api/v2/pages/nessus/files/Nessus-latest-debian10_amd64.deb
```

3. Instalação do Nessus

```
sudo dpkg -i Nessus-latest-debian10_amd64.deb
```

4. Inicialização e ativação automática do serviço

```
sudo systemctl start nessusd.service

sudo systemctl enable nessusd.service
```

Validação dos serviços:

```
sudo systemctl status nessusd                    # Confirma: Active (running)

ss -lntp | grep :8834                            # Confirma: porta 8834 escutando
```

Acesso à interface web: https://localhost:8834/

2.3 Ativação e Sincronização de Plugins

Após o primeiro acesso, foi criada uma conta gratuita Tenable Nessus Essentials e realizada a autenticação:

1. Login via interface web → "Log in with Tenable"
2. Download automático dos plugins (50.000+ assinaturas CVE)
3. Tempo de sincronização: 25 minutos
4. Validação final: Settings → About → Status: "Plugins Loaded" ✓ "Feed Current" ✓

IMPORTANTE: Durante esse processo, não fechar a aba do navegador.

2.4 Configuração e Execução do Primeiro Scan

Descoberta do alvo:

1. No Metasploitable 2:

```
ip a
```

2. Resultado do IP anotado: **192.168.56.105**

Criação do scan básico na interface web:

1. Scans → New Scan → Basic Network Scan
2. Configurações:
    - Nome: Metasploitable
    - Targets (IPs): 192.168.56.105
    - Discovery: Port Scanning (Default)
    - Assessment: Padrão
3. Save → Launch

Tempo de execução: 12 minutos
Status final: Completed ✓

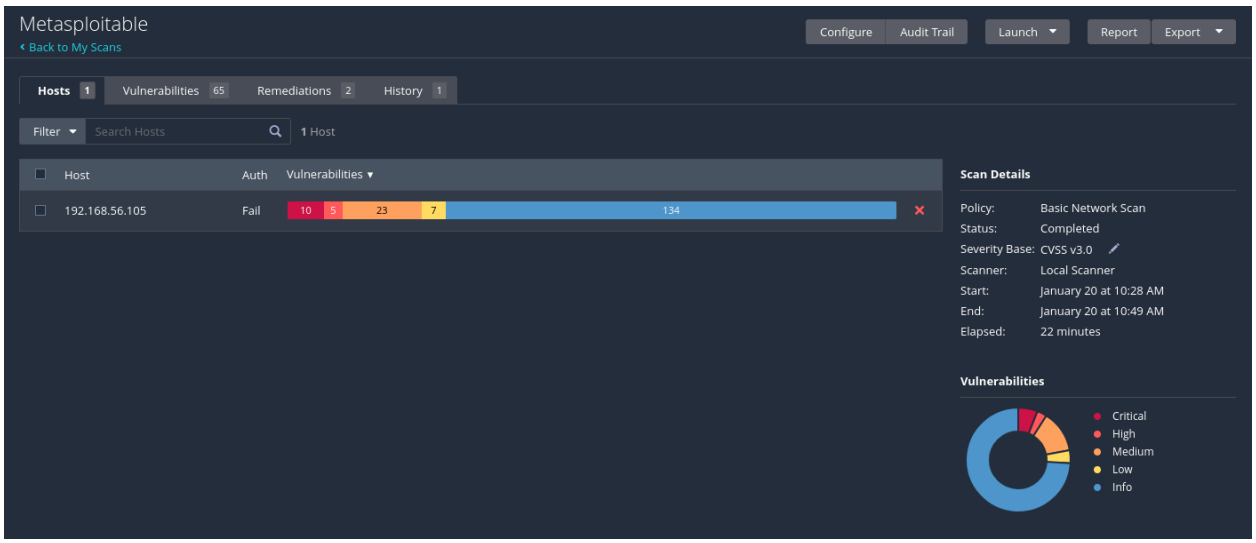


Imagem 1 – Dashboard do scan

2.5 Análise Preliminar dos Resultados

O scan identificou um total de **65 vulnerabilidades** no host analisado. Destas, **45 foram classificadas com severidade explícita** nas categorias Critical, High, Medium e Low:

Severidade	Quantidade
Critical	10
High	5
Medium	23
Low	7

Tabela 2 - Vulnerabilidades Identificadas

Embora o Nessus indique a identificação de 65 vulnerabilidades no host analisado, a soma visual das severidades exibidas (Critical, High, Medium e Low) totaliza apenas 45 achados. Essa diferença ocorre porque o valor apresentado como “Vulnerabilities” corresponde ao total de achados que o Nessus classifica como vulnerabilidades relevantes, independentemente de estarem todas associadas a uma severidade CVSS padrão exibida no gráfico. Parte desses achados inclui detecções relacionadas a más configurações, políticas fracas, uso de criptografia obsoleta ou outras condições de risco que não se enquadram diretamente nas categorias de severidade visualizadas. Dessa forma, o número 65 representa um total consolidado de vulnerabilidades, enquanto o gráfico reflete apenas aquelas categorizadas explicitamente entre Critical e Low.

Além das vulnerabilidades classificadas por severidade, o Nessus identificou 134 achados do tipo Informational (Info). Esses achados não representam vulnerabilidades exploráveis por si só, mas consistem em informações técnicas relevantes sobre o ambiente, como portas abertas, versões de serviços e sistemas operacionais, banners de serviços, configurações de rede,

enumeração de usuários ou compartilhamentos, e detalhes de certificados e protocolos. Embora não possuam pontuação CVSS, essas informações podem auxiliar significativamente um atacante durante a fase de reconhecimento e mapeamento do alvo, aumentando o contexto e a superfície de ataque. Por esse motivo, apesar de não serem contabilizadas como vulnerabilidades, os achados Informational devem ser considerados em análises de risco e em planos de endurecimento de segurança (hardening).

2.6 Consolidação e Classificação das Vulnerabilidades

A Tabela 3 apresenta a consolidação das principais vulnerabilidades identificadas durante o scan, organizadas de acordo com a severidade atribuída pelo Nessus (Critical, High, Medium e Info), bem como os respectivos serviços afetados, impactos potenciais e recomendações gerais de mitigação.

Devido ao elevado número de achados, optou-se por resumir e agrupar vulnerabilidades semelhantes (como falhas criptográficas e serviços fora de suporte), mantendo o foco naquelas que apresentam maior impacto à confidencialidade, integridade e disponibilidade do sistema.

Severidade	Vulnerabilidade	Serviço / Porta	Descrição Resumida	Impacto	Recomendação
Critical	Canonical Ubuntu Linux SEoL (8.04.x)	Sistema Operacional	SO fora de suporte desde 2013, sem patches de segurança.	Múltiplas vulnerabilidades conhecidas sem correção.	Atualizar para versão suportada do Ubuntu.
Critical	VNC Server 'password' Password	5900/tcp (vnc)	Servidor VNC protegido com senha fraca ('password').	Controle remoto total do sistema.	Definir senha forte e restringir acesso ao VNC.
Critical	Bind Shell Backdoor Detection	1524/tcp (wild_shell)	Shell aberta sem autenticação, executando comandos como root.	Comprometimento total do host.	Reinstalar sistema e investigar comprometimento.
Critical	Apache Tomcat SEoL (<= 5.5.x)	8180/tcp (www)	Tomcat fora de suporte desde 2012.	Múltiplas falhas conhecidas exploráveis.	Atualizar para versão suportada do Tomcat.
Critical	Apache Tomcat AJP Request Injection (Ghostcat)	8009/tcp (ajp13)	Leitura de arquivos e possível RCE via conector AJP.	Execução remota de código / vazamento de dados.	Atualizar Tomcat e restringir AJP.
Critical	Debian OpenSSL RNG Weakness (SSL)	25/tcp, 5432/tcp	Certificados gerados com entropia fraca.	Quebra de criptografia / MITM.	Regenerar chaves e certificados.
Critical	Debian OpenSSL RNG Weakness (SSH)	22/tcp (ssh)	Chaves SSH previsíveis.	Comprometimento de sessões SSH.	Regenerar chaves SSH.

High	ISC BIND Service Downgrade / Reflected DoS	53/udp (dns)	BIND vulnerável a downgrade e DoS refletido.	Interrupção de serviço / amplificação de ataques.	Atualizar BIND para versão corrigida.
High	Samba Badlock Vulnerability	445/tcp (cifs)	Downgrade de autenticação RPC em Samba.	Execução de chamadas arbitrárias.	Atualizar Samba.
High	NFS Shares World Readable	2049/tcp (rpc-nfs)	Compartilhamento NFS sem restrição de acesso.	Exposição de arquivos sensíveis.	Restringir exportações NFS.
High	SSL Medium Strength Ciphers (SWEET32)	25/tcp, 5432/tcp	Suporte a 3DES e cifras fracas.	Criptografia quebrável.	Desabilitar cifras fracas e 3DES.
Medium	Apache Tomcat Default Files	8180/tcp (www)	Arquivos e páginas padrão presentes.	Divulgação de informações.	Remover arquivos padrão.
Medium	ISC BIND Denial of Service	53/udp (dns)	Vulnerável a DoS por pacote malformado.	Indisponibilidade do serviço DNS.	Atualizar BIND.
Medium	HTTP TRACE / TRACK Methods Allowed	80/tcp (www)	Métodos HTTP perigosos habilitados.	Roubo de cookies / XST.	Desabilitar TRACE e TRACK.
Info	Apache Tomcat Detection	8180/tcp (www)	Detecção de versão do Tomcat.	Enumeração de serviços.	Ocultar banners e versões.

Tabela 3 – Consolidação das Vulnerabilidades por Severidade

Conforme observado na Tabela 3, o host analisado apresenta um estado severo de insegurança, com destaque para a presença de múltiplas vulnerabilidades críticas exploráveis remotamente, serviços fora de suporte e configurações inseguras por padrão.

Entre os achados mais relevantes, destacam-se:

- Presença de uma bind shell ativa sem autenticação (porta 1524/tcp)
- Servidor VNC protegido por senha fraca conhecida (“password”)
- Sistema operacional Ubuntu 8.04 e Apache Tomcat 5.5 fora de suporte
- Vulnerabilidades críticas no Apache Tomcat (Ghostcat – CVE-2020-1938)
- Uso de criptografia fraca e chaves previsíveis (Debian OpenSSL RNG)

Esses fatores indicam que, em um ambiente real, o sistema estaria totalmente comprometido ou em risco iminente de comprometimento.

2.7 Priorização dos Achados para Análise Detalhada

Considerando o elevado número de vulnerabilidades identificadas e a inviabilidade prática de explorar individualmente todos os achados, foi realizada uma etapa de priorização com base nos seguintes critérios:

- Severidade atribuída pelo Nessus (Critical e High)
- Possibilidade de exploração remota sem autenticação
- Impacto potencial sobre confidencialidade, integridade e disponibilidade
- Presença de provas de conceito públicas (PoC)
- Relevância didática para demonstração prática de risco

A partir desses critérios, foram selecionadas algumas vulnerabilidades críticas e de alto risco para análise mais aprofundada, dentre as quais se destacam:

- Apache Tomcat AJP Request Injection (Ghostcat – CVE-2020-1938)
- Bind Shell Backdoor Detection (porta 1524/tcp)
- VNC Server com senha fraca conhecida
- Debian OpenSSL Random Number Generator Weakness

Dentre essas, foi escolhida a vulnerabilidade Ghostcat (CVE-2020-1938) para validação prática, por permitir leitura arbitrária de arquivos e potencial execução remota de código.

2.8 Validação Prática de Vulnerabilidade Crítica (Proof of Concept)

A vulnerabilidade Ghostcat (CVE-2020-1938) está associada ao conector Apache JServ Protocol (AJP), utilizado pelo Apache Tomcat para comunicação interna entre componentes da aplicação. Em versões vulneráveis, quando o conector AJP encontra-se exposto indevidamente à rede, um atacante remoto não autenticado pode explorar a falha para realizar a leitura arbitrária de arquivos internos do servidor, comprometendo informações sensíveis e configurações críticas da aplicação.

A identificação inicial da vulnerabilidade ocorreu por meio do scanner Nessus, que apontou a exposição do serviço AJP na porta 8009/TCP, associada a uma versão vulnerável do Apache Tomcat, conforme apresentado na **Imagem 2**. Esse resultado indicou a possibilidade de exploração remota sem autenticação, justificando a realização de uma validação manual.

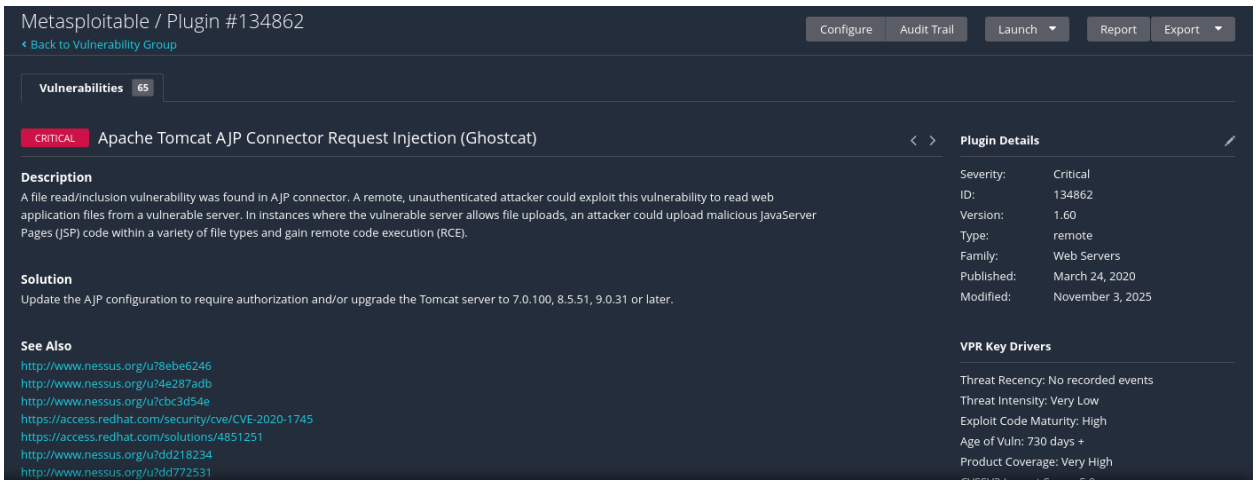


Imagem 2 – Ghostcat (CVE-2020-1938)

Na etapa seguinte, foi conduzida uma verificação ativa dos serviços expostos, confirmando que as portas 8009/TCP (AJP) e 8080/TCP (HTTP) encontravam-se acessíveis externamente, caracterizando uma configuração insegura do ambiente analisado (**Imagem 3**).

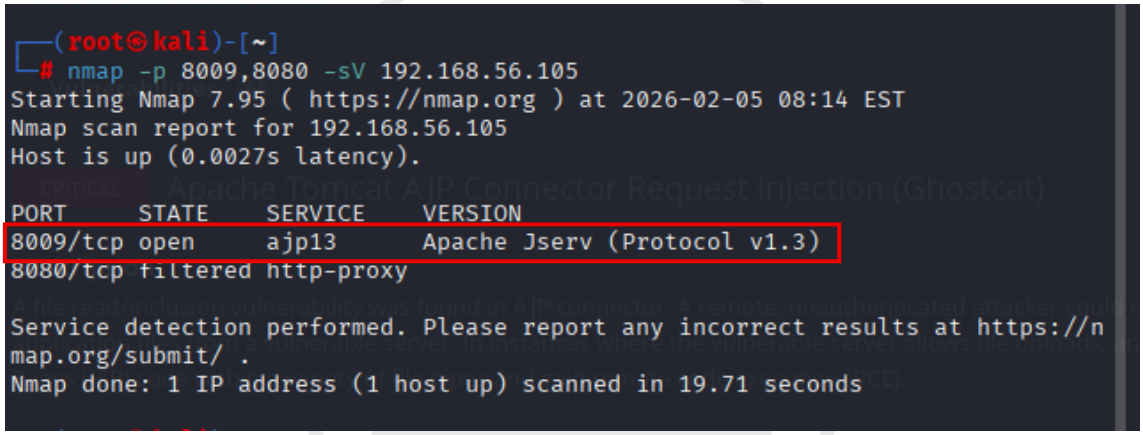


Imagem 3 – Terminal do mapeamento das portas 8009 e 8080

Com o objetivo de comprovar a existência e o impacto da vulnerabilidade, foi realizada uma exploração controlada utilizando o framework Metasploit, por meio do módulo específico para a Ghostcat (**Imagem 4**). A exploração permitiu a leitura arbitrária de arquivos internos da aplicação hospedada no servidor vulnerável, sem a necessidade de autenticação.

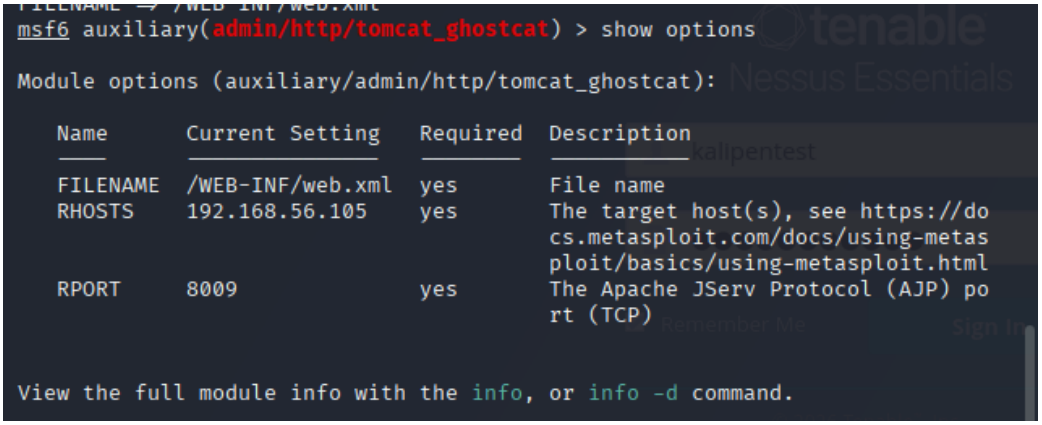


Imagem 4 – Módulo Ghostcat no Metasploit

Como evidência prática do impacto, foi realizada a leitura do arquivo de configuração web.xml, que contém informações sensíveis sobre a estrutura e o funcionamento da aplicação (**Imagem 5**). A obtenção desse arquivo confirma a exploração bem-sucedida da vulnerabilidade, demonstrando o comprometimento da confidencialidade das informações e validando o risco apontado na etapa de varredura automatizada.

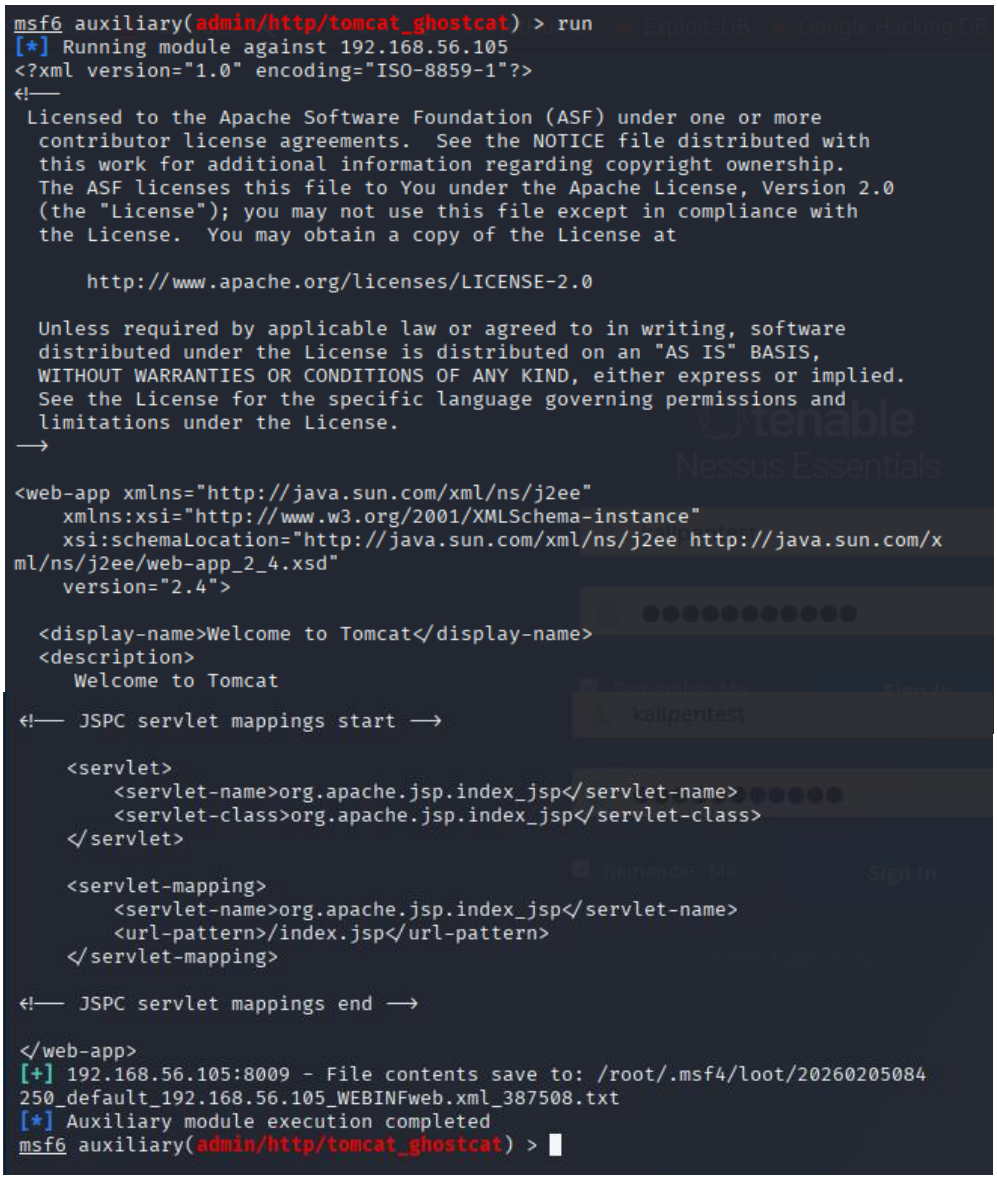


Imagem 5 – Leitura do arquivo **web.xml**

2.9 Análise de Impacto e Recomendações de Mitigação

A validação prática da vulnerabilidade Ghostcat (CVE-2020-1938) evidenciou que a exposição do conector Apache JServ Protocol (AJP) representa um risco significativo à segurança do ambiente analisado. A exploração realizada demonstrou que um atacante remoto, sem necessidade de autenticação prévia, é capaz de acessar arquivos internos da aplicação, os quais normalmente não estariam disponíveis externamente.

O impacto primário dessa vulnerabilidade está relacionado à **quebra da confidencialidade**, uma vez que arquivos sensíveis, como o *web.xml*, podem conter informações críticas sobre a estrutura da aplicação, nomes de classes, caminhos internos e, em alguns casos, credenciais ou parâmetros de conexão. Essas informações podem ser utilizadas como base para ataques subsequentes, ampliando a superfície de ataque do sistema.

Além disso, dependendo da configuração do servidor e das permissões atribuídas aos arquivos acessados, a falha pode contribuir para cenários mais severos, como **execução remota de código**, **modificação de arquivos internos** e **comprometimento completo do servidor**, afetando também a integridade e a disponibilidade dos serviços hospedados.

Diante desse cenário, recomenda-se a adoção das seguintes medidas de mitigação:

- Desativar o conector AJP caso não seja estritamente necessário ao funcionamento da aplicação;
- Restringir o acesso à porta 8009/TCP por meio de regras de firewall, permitindo conexões apenas de hosts autorizados;
- Atualizar o Apache Tomcat para versões que não sejam afetadas pela vulnerabilidade Ghostcat;
- Implementar práticas de *hardening* no servidor, reduzindo a exposição de serviços e portas desnecessárias;
- Realizar varreduras periódicas de vulnerabilidades e validações manuais para identificar falhas de configuração que não sejam mitigadas apenas por atualizações.

A análise reforça a importância de não se limitar apenas aos resultados de ferramentas automatizadas, destacando o valor da validação prática para a correta avaliação do risco real associado às vulnerabilidades identificadas.

3 CONSIDERAÇÕES FINAIS

O desenvolvimento deste trabalho possibilitou a aplicação prática de conceitos fundamentais de segurança da informação, especialmente no que se refere à identificação, análise e validação de vulnerabilidades em ambientes computacionais. Por meio do uso de ferramentas amplamente adotadas no mercado, como o Nessus e o Metasploit Framework, foi possível compreender de forma concreta como falhas de configuração e serviços expostos podem representar riscos significativos à segurança de um sistema.

A utilização de um scanner automatizado permitiu a identificação inicial de um elevado número de vulnerabilidades, evidenciando a importância de processos estruturados de análise e priorização. A partir dessa etapa, a escolha da vulnerabilidade Ghostcat (CVE-2020-1938) para validação prática demonstrou como um achado identificado automaticamente pode ser confirmado manualmente e explorado de forma controlada, reforçando a necessidade de complementar análises automatizadas com validações técnicas aprofundadas.

O Proof of Concept realizado evidenciou o impacto real da vulnerabilidade, demonstrando a possibilidade de acesso não autorizado a arquivos internos da aplicação, o que poderia servir como ponto de partida para ataques mais complexos em um ambiente produtivo. Essa etapa contribuiu para o entendimento do ciclo completo de um processo de avaliação de segurança, desde a detecção inicial até a análise de impacto e proposição de medidas de mitigação.

Por fim, o projeto contribuiu significativamente para o aprimoramento das habilidades técnicas e analíticas relacionadas à área de cibersegurança, proporcionando uma visão prática sobre testes de vulnerabilidade e a importância de boas práticas de configuração e manutenção de sistemas. A experiência adquirida reforça a relevância da segurança como elemento essencial no desenvolvimento e na operação de ambientes computacionais modernos.

REFERÊNCIAS

APACHE SOFTWARE FOUNDATION. **Documentação do Apache Tomcat**.

Disponível em: https://tomcat.apache.org/tomcat-9.0-doc/index.html.

MITRE CORPORATION. **CVE – Dicionário de Vulnerabilidades e Exposições Comuns**.

Disponível em: https://cve.mitre.org/.

RAPID7. **Documentação do Metasploit Framework**.

Disponível em: https://docs.rapid7.com/metasploit/.

KALI LINUX. **Documentação Oficial do Kali Linux**.

Disponível em: https://www.kali.org/docs/.

TENABLE. **Documentação do Nessus.**

Disponível em: https://pt-br.tenable.com/products/nessus.