Insper

Tecnologias Hacker

Prof. Dr. Rodolfo Avelino e João Eduardo

Quebra de senhas de Sistema Operacional

Objetivo do laboratório

- Entender como as senhas são armazenadas no Windows e Linux.
- Aprender a extrair hashes de senhas a partir dos arquivos SAM, SYSTEM (Windows) e /etc/shadow (Linux).
- Utilizar ferramentas como John the Ripper e Hashcat para quebrar hashes de senhas.

1. Como as Senhas são armazenadas no Windows?

Introdução ao SAM (Gerenciador de Contas de Segurança)

O que é o SAM?

O SAM (Security Account Manager) é o Gerenciador de Contas de Segurança do Windows que administra todas as contas de usuário e suas senhas. Ele funciona como um banco de dados onde todas as senhas são armazenadas na forma de hashes.

O LSA (Autoridade de Segurança Local) é responsável por verificar os logins dos usuários, comparando as senhas digitadas com o banco de dados mantido pelo SAM. O SAM começa a funcionar em segundo plano assim que o Windows é iniciado.

Insper

Onde encontrar o SAM:

- Arquivo: C:\Windows\System32\config
- No Registro do Windows: Abra o Editor de Registro e navegue até HKEY_LOCAL_MACHINE\SAM

Como as senhas são armazenadas no Windows?

Para entender como as senhas são salvas no Windows, precisamos conhecer diferentes protocolos de autenticação: LM, NTLM v1 e v2.

Autenticação LM

A autenticação LAN Manager (LM) foi desenvolvida pela IBM para os sistemas operacionais Windows da Microsoft. Atualmente, sua segurança é considerada vulnerável devido a:

- Divide a senha em dois blocos de sete caracteres cada
- Converte todos os caracteres para maiúsculas (não é sensível a maiúsculas/minúsculas)
- Usa criptografia DES de 56 bits, que hoje pode ser facilmente quebrada
- Facilita ataques de força bruta ou dicionário, pois elimina a necessidade de testar caracteres minúsculos

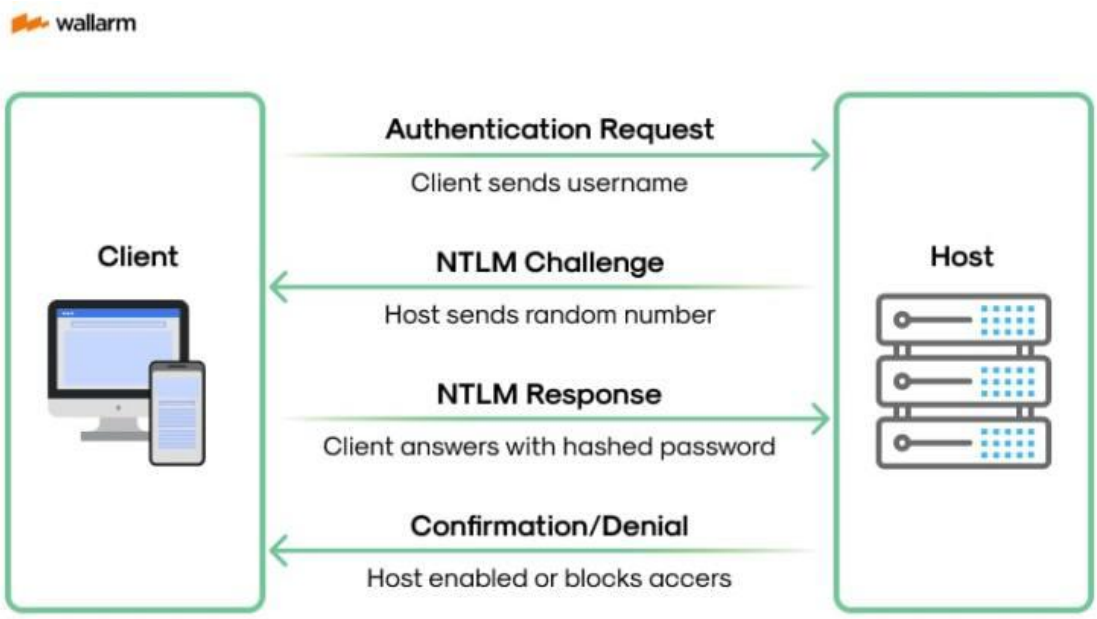
Autenticação NTLM

A autenticação NTLM foi desenvolvida para substituir o LM, que se mostrou inseguro. O NTLM é baseado em um mecanismo de desafio-resposta e utiliza três componentes:

1. **Nonce (desafio)**: string numérica aleatória
2. **Resposta**: resultado da criptografia
3. **Autenticação**: processo de validação

Funcionamento da autenticação NTLM:

1. O NTLM criptografa a senha e armazena o hash, descartando a senha original
2. O cliente envia o nome de usuário ao servidor
3. O servidor cria um nonce de 16 bytes e o envia ao cliente
4. O cliente criptografa o nonce usando o hash da senha e envia o resultado de volta
5. O Controlador de Domínio recebe o nome de usuário, nonce e resposta
6. O Controlador recupera o hash da senha do banco de dados SAM
7. Se o nonce e a resposta corresponderem, a autenticação é bem-sucedida



Insper

Diferenças entre NTLM v1 e NTLM v2

Característica	NTLM v1	NTLM v2
Algoritmo de hash	MD4	MD5
Tamanho do C/R	56 bits + 56 bits + 16 bits	128 bits
Algoritmo C/R	DES (modo ECB)	HMAC_MD5
Comprimento do valor C/R	64 bits + 64 bits + 64 bits	128 bits

Observação importante sobre o Windows 10

A partir do Windows 10 v1607, a Microsoft alterou o algoritmo, substituindo a cifra RC4 pelo AES. Essa mudança tornou obsoletas muitas ferramentas de extração que acessavam diretamente o SAM. Algumas ferramentas foram atualizadas para lidar com o novo método de criptografia, mas outras não conseguiram se adaptar.

Isso não significa que essas ferramentas não possam mais ser usadas, apenas que para sistemas Windows 10 mais recentes, é recomendável utilizar ferramentas atualizadas.

Insper

2. Ferramentas Utilizadas

Samdump2

- Ferramenta para extrair hashes de senhas dos arquivos SAM e SYSTEM.
- Faz parte do pacote samba no Linux.
- Comando simples e direto para extração de hashes.

John the Ripper

- Ferramenta de código aberto para quebra de senhas.
- Suporta hashes LM e NTLM.
- Disponível para Windows e Linux.

3. Passos para Quebra de senhas

Passo 1: Extrair os Hashes com Samdump2

1. Com os arquivos SAM e SYSTEM fornecidos, use o samdump2 para extrair os hashes. No comando a seguir o arquivo hashes.txt receberá os hashes:

```
┌──(root㉿kali)-[/home/kali/win]
└─# samdump2 SYSTEM SAM > hashes.txt
```

Isso gerará uma lista de hashes no formato: **usuário:ID:LM_hash:NTLM_hash:::**

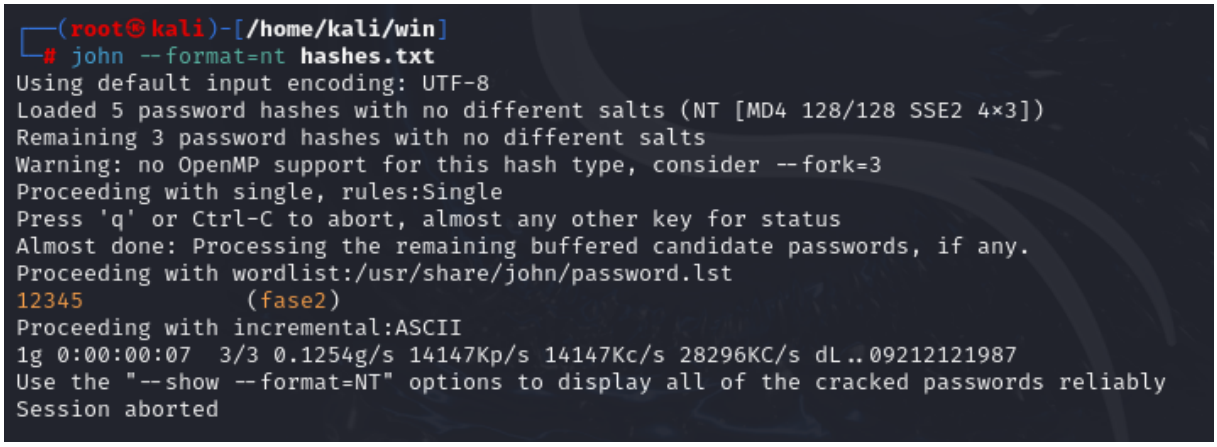
```
┌──(root㉿kali)-[/home/kali/win]
└─# cat hashes.txt
*disabled* Administrador:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
*disabled* Convidado:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
web1:1001:aad3b435b51404eeaad3b435b51404ee:59edfb15aec624e7ccf5c8c50682c649:::
HomeGroupUser$:1002:aad3b435b51404eeaad3b435b51404ee:4fb32bac80c878f17c5222469d57b790:::
fase2:1004:aad3b435b51404eeaad3b435b51404ee:7a21990fcd3d759941e45c490f143d5f:::
```

Insper

Passo 2: Quebrar os hashes

Execute o comando a seguir para quebrar os hashes NTLM:

john --format=nt hashes.txt



A saída do comando john, apresenta que a senha do usuário fase2 é 12345.

4. Como as senhas são administradas e armazenadas no GNU/Linux

Arquivo /etc/passwd

Características:

- O arquivo /etc/passwd contém informações básicas sobre os usuários do sistema.
- Cada linha representa um usuário e segue o formato:

nome_do_usuário:x:UID:GID:descrição:home:shell

- **nome_do_usuário**: Nome do usuário.
- **x**: Indica que a senha está armazenada no arquivo **/etc/shadow**.
- **UID**: Identificador único do usuário.
- **GID**: Identificador do grupo principal do usuário.
- **descrição**: Campo para informações adicionais (geralmente o nome completo).
- **home**: Diretório home do usuário.
- **shell**: Shell padrão do usuário.

Exemplo:

root:x:0:0:root:/root:/bin/bash

aluno:x:1000:1000:Aluno da Turma:/home/aluno:/bin/bash

Arquivo /etc/shadow

Características:

- O arquivo /etc/shadow armazena as senhas dos usuários em formato de hash.
- Cada linha representa um usuário e segue o formato:

nome_do_usuário:hash_senha:última_alt_senha:dias_min:dias_max:dias_aviso:dias_inat:data_expira:reservado

- **nome_do_usuário**: Nome do usuário.
- **hash_senha**: Hash da senha do usuário.
- **última_alt_senha**: Data da última alteração da senha (em dias desde 1º de janeiro de 1970).

- **dias_min**: Número mínimo de dias entre alterações de senha.
- **dias_max**: Número máximo de dias que a senha é válida.
- **dias_aviso**: Número de dias antes de expirar para avisar o usuário.
- **dias_inat**: Número de dias de inatividade após a expiração da senha.
- **data_expira**: Data de expiração da conta (em dias desde 1º de janeiro de 1970).
- **reservado**: Campo reservado para uso futuro.

Exemplo:

root:6randomsalt$hashedpassword:18395:0:99999:7:::

aluno:6randomsalt$hashedpassword:18395:0:99999:7:::

5. Identificando o Tipo de Hash no /etc/shadow

O campo hash_senha no arquivo /etc/shadow começa com um identificador que indica o algoritmo de hash utilizado. Aqui estão os principais:

Formato do Hash:

idsalt$hashedpassword

- **id**: Identificador do algoritmo de hash.
- **salt**: Valor aleatório usado para "temperar" o hash.
- **hashedpassword**: Hash da senha.

Identificadores Comuns:

Insper

SHA-512:

- Identificador: 6
- Exemplo: 6randomsalt$hashedpassword
- Algoritmo: SHA-512 (o mais comum em sistemas Linux modernos).

SHA-256:

- Identificador: 5
- Exemplo: 5randomsalt$hashedpassword
- Algoritmo: SHA-256.

MD5:

- Identificador: 1
- Exemplo: 1randomsalt$hashedpassword
- Algoritmo: MD5 (antigo e inseguro).

DES:

- Sem identificador (apenas o hash).
- Exemplo: hashedpassword
- Algoritmo: DES (muito antigo e inseguro).

Insper

4. Como Identificar o Tipo de Hash no Exercício

1. Abra o arquivo /etc/shadow fornecido.
2. Localize o campo hash_senha de um usuário.
3. Verifique o identificador no início do hash:
    - Se começar com 6, é **SHA-512**.
    - Se começar com 5, é **SHA-256**.
    - Se começar com 1, é **MD5**.
    - Se não tiver identificador, é **DES**.

Exemplo:

usuário1:6randomsalt$hashedpassword → SHA-512

usuário2:5randomsalt$hashedpassword → SHA-256

usuário3:1randomsalt$hashedpassword → MD5

usuário4:hashedpassword → DES

Insper

5. Quebrando a hash do arquivo shadow

Utilize o comando john diretamente no arquivo shadow fornecido para este exercício.

A saída do comando apresenta a senha 123456 para o usuário fase1.

Conclusão

Este exercício permitiu explorar técnicas e ferramentas para a quebra de senhas a partir de hashes extraídos de sistemas Windows (arquivos SAM e SYSTEM) e Linux (arquivo /etc/shadow). Utilizamos ferramentas como John the Ripper e samdump2 para extrair e quebrar hashes, demonstrando como é possível recuperar senhas quando estas são fracas ou comuns.

Pontos Importantes Destacados:

1. Hash é uma Cifra de Via Única:
    - Um hash é o resultado de uma função criptográfica que transforma uma senha em um código único e irreversível. Isso significa que, teoricamente, não é possível "reverter" o hash para obter a senha original. As ferramentas de quebra de senhas funcionam comparando

hashes gerados a partir de listas de senhas possíveis (wordlists) com o hash fornecido.

2. Ferramentas e Técnicas:
    - Além do John the Ripper, existem outras ferramentas poderosas, como o Hashcat, que utiliza GPUs para acelerar o processo de quebra de senhas.
    - Também existem páginas web e serviços online que permitem consultar hashes em bancos de dados pré-computados, como o CrackStation e o Hashes.com.
3. Dependência de Wordlists:
    - O sucesso da quebra de senhas depende diretamente da qualidade e abrangência das wordlists utilizadas. Listas como rockyou.txt contêm milhares de senhas comuns, mas senhas complexas e únicas podem ser praticamente impossíveis de quebrar sem o uso de técnicas avançadas ou recursos computacionais significativos.
4. Importância de Boas Práticas de Segurança:
    - Este exercício reforça a importância de utilizar senhas fortes, únicas e complexas, que dificultem ataques de dicionário ou força bruta.
    - Administradores de sistemas devem proteger arquivos como SAM, SYSTEM e /etc/shadow, além de implementar políticas de senhas robustas e autenticação de dois fatores (2FA).

A quebra de senhas é uma atividade que ilustra tanto as vulnerabilidades dos sistemas quanto a importância da criptografia e da segurança da informação. Embora ferramentas como John the Ripper e Hashcat sejam poderosas, elas dependem de falhas humanas, como a escolha de senhas fracas. Portanto, a conscientização sobre

Insper

boas práticas de segurança é fundamental para proteger dados e sistemas contra ataques.