Aula Prática

Segurança e Auditoria em Sistemas

Profa. Natássya Barlate Floro da Silva

Universidade Tecnológica Federal do Paraná -– Câmpus Cornélio Procópio

09 de Março de 2020

Roteiro

Ataque de Força Bruta

- Ataques de Força Bruta realizam tentativas com todas as combinações possíveis para desvendar uma senha ou chave.
- John the Ripper: software para realizar a quebra de chaves.
- HashSuite: alternativa para Windows.

Ataque de Força Bruta

- Instalação do John the Ripper no Ubuntu: sudo snap install john-the-ripper
- Baixar o arquivo Material_Aula_3.zip do Moodle, descompactá-lo e abrir o terminal no novo diretório.
- Os comandos serão executados com o comando `time` para sabermos o tempo que levou cada execução.
- Quebrar o hash usando ataque força bruta: `time john --format=raw-md5 hash_1.txt`
- Para observar a saída caso rode mais de uma vez, usar o comando: `john --show --format=raw-md5 hash_1.txt`
- Tirar PrintScreen e guardar como "Tentativa_1.png".

Ataque de Força Bruta

- Por que a quebra foi tão rápida?

Ataque de Força Bruta

- Por que a quebra foi tão rápida?
- Ao invés de termos um ataque de força bruta, foi realizado um ataque de dicionário.

Ataque de Força Bruta

- Por que a quebra foi tão rápida?
- Ao invés de termos um ataque de força bruta, foi realizado um ataque de dicionário.
- Observar arquivo password.lst: `cat /snap/john-the-ripper/current/run/password.lst`

Ataque de Força Bruta

- Por que a quebra foi tão rápida?
- Ao invés de termos um ataque de força bruta, foi realizado um ataque de dicionário.
- Observar arquivo password.lst: `cat /snap/john-the-ripper/current/run/password.lst`
- Ataque de dicionário utiliza uma lista de possíveis senhas/chaves para testar como soluções.

Ataque de Força Bruta

- Quebrar o hash usando ataque força bruta: `time john --format=raw-md5 hash_2.txt`
- Para observar a saída caso rode mais de uma vez, usar o comando: `john --show --format=raw-md5 hash_2.txt`
- Tirar PrintScreen e guardar como "Tentativa_2.png".

Ataque de Força Bruta

- Outras opções do modo *Increment*: `time john --increment=LowerNum --format=raw-md5 hash_3.txt`
- Para observar a saída caso rode mais de uma vez, usar o comando: `john --show --format=raw-md5 hash_3.txt`
- Tirar PrintScreen e guardar como "Tentativa_3.png".

Ataque de Força Bruta

- Próxima senha: `time john --increment=LowerNum --format=raw-md5 hash_4.txt`
- Para observar a saída caso rode mais de uma vez, usar o comando: `john --show --format=raw-md5 hash_4.txt`
- Tirar PrintScreen e guardar como "Tentativa_4.png".

Ataque de *Rainbow Tables*

- Ataques de *Rainbow Tables* utilizam uma tabela pré-computada com valores dos *hash*.
- Ocupam mais espaço de armazenamento, mas o processamento é mais rápido.
- Rainbow-Crack: permite a geração e determinação de *hashs* com *Rainbow Tables*.

Ataque de *Rainbow Tables*

- Entrar na pasta rainbowcrack-1.7-linux64 e abrir um terminal.
- Fornecer permissão de execução para os programas: `chmod +x rtgen rtsort rcrack`

Ataque de *Rainbow Tables*

- Geração de *Rainbow Tables*: `rtgen hash_algorithm charset plaintext_len_min plaintext_len_max table_index chain_len chain_num part_index`
  - hash_algorithm: algoritmo *hash* para geração.
  - charset: caracteres possíveis para a geração.
  - plaintext_len_min: tamanho mínimo do valor referente ao *hash*.
  - plaintext_len_max: tamanho máximo do valor referente ao *hash*.
  - table_index: seleciona a função de redução.
  - chain_len: tamanho da cadeia usada na geração. Quanto maior, maior o tempo para geração.
  - chain_num: número de cadeias calculadas.
  - part_index: usado para gerar tabelas menores em diferentes arquivos. Basta alterar esse número na geração.

Ataque de *Rainbow Tables*

- Gerar Rainbow Tables para caracteres alfa-numéricos minúsculos: `time ./rtgen md5 loweralpha-numeric 4 6 0 2100 8000000 0`

Ataque de *Rainbow Tables*

- Gerar Rainbow Tables para caracteres alfa-numéricos minúsculos: `time ./rtgen md5 loweralpha-numeric 4 6 0 2100 8000000 0`
- Ordenar a tabela gerada: `time ./rtsort .`

Ataque de *Rainbow Tables*

- Gerar Rainbow Tables para caracteres alfa-numéricos minúsculos: `time ./rtgen md5 loweralpha-numeric 4 6 0 2100 8000000 0`
- Ordenar a tabela gerada: `time ./rtsort .`
- Quebrar o *hash* para todos os arquivos: `time ./rcrack . -l ../hash_1.txt`
- Tire uma PrintScreen para cada arquivo e salve como "Tentativa_RT_N", onde N é o número correspondente a cada arquivo.

Ataque de *Rainbow Tables*

- Qual das duas técnicas utilizadas foi mais rápida?

Ataque de *Rainbow Tables*

- Qual das duas técnicas utilizadas foi mais rápida?
- Gere agora o seu próprio hash MD5 com o seguinte comando: `echo -n "suaString" | md5sum`

Ataque de *Rainbow Tables*

- Qual das duas técnicas utilizadas foi mais rápida?
- Gere agora o seu próprio hash MD5 com o seguinte comando: `echo -n ‘‘suaString’’ | md5sum`
- Salve o hash gerado em um arquivo e quebre-o usando as duas técnicas aprendidas. Tira uma PrintScreen e salve como “Tentativa_Final.png” para o ataque de força bruta e “Tentativa_RT_Final.png” para o ataque de *Rainbow Tables*.

Ataque de *Rainbow Tables*

- Compacte todas as imagens geradas na prática em um arquivo e submeta-o pelo Moodle.