PRINCÍPIO FUNDAMENTAL DA CONTAGEM: VIABILIDADE DE QUEBRA DE SENHA DE ARQUIVO COMPACTADO UTILIZANDO RECURSOS DOMÉSTICOS DE COMPUTAÇÃO

FUNDAMENTAL PRINCIPLE OF COUNTING: PASSWORD'S CRACKING FEASIBILITY OF COMPRESSED FILES USING HOME COMPUTER RESOURCES

PRINCIPIO FUNDAMENTAL DEL CONTEO: FACTIBILIDAD DE RUPTURA DE CONTRASEÑA DE ARCHIVO COMPRIMIDO CON EL USO DE RECURSOS INFORMÁTICOS DOMÉSTICOS

Ronaldo Renan Berdum[1]

Resumo

Este artigo analisa a viabilidade de quebra de senha de arquivo compactado utilizando recursos domésticos de computação. O estudo se justifica devido à confiança que temos em certos dispositivos de segurança que, por vezes, não são mais que uma falsa segurança. O objetivo central é revelar matematicamente a viabilidade para que um elemento adverso que não possua a senha do arquivo protegido consiga acessá-lo mediante tentativa e erro. A teoria prevê que toda e qualquer senha pode ser encontrada por tentativa e erro; a diferença está apenas no tempo necessário para tal feito. Foram empregados procedimentos de abordagem qualitativa. A técnica de obtenção de informações foi documental. Os instrumentos utilizados foram documentações e manuais de *softwares* específicos e conteúdo acadêmico relativo ao princípio fundamental da contagem. Este estudo fundamenta-se em revisão bibliográfica. A análise demonstrou que senhas curtas e com alfabetos pouco complexos podem ser encontradas em tempo relativamente curto. Senhas com oito caracteres, compostas apenas por algarismos numéricos, podem ser encontradas em poucos minutos de processamento computacional doméstico.

Palavras-chave: senha; força bruta; Pentest.

Abstract

This paper examines password's cracking feasibility of compressed files using home computer resources. The study is justified in one's reliance on certain security devices, which, however, oftentimes provides just a false sense of security. The main objective is to mathematically reveal the feasibility for an adversary without the password of a protected file can even so access it through trial and error method. The theory predict that any password can be found by trial and error; the difference lies only in the time required to do so. It were used qualitative approach procedures, and documentary technique to gather information, as specific softwares manuals and academic content regarding fundamental principle of counting. Therefore, this study is based on a literature review. The analysis showed that short passwords with not very complex alphabet were disclosed in relatively shorter time. An eight-characters password consisting only of numerical digits can be found in a few minutes of home computer processing.

Keywords: password; brute force attack; Pentest.

Resumen

Este artículo analiza la posibilidad de ruptura de contraseña de archivo comprimido con el uso de recursos informáticos domésticos. El estudio se justifica por la confianza que depositamos en determinados dispositivos de seguridad, que a menudo no ofrecen más que una falsa seguridad. El objetivo central es determinar en términos matemáticos la factibilidad de que un elemento adverso, que no disponga de la contraseña de un archivo comprimido logre acceder a él por ensayo y error. La teoría prevé que toda y cualquier contraseña puede ser descifrada por ensayo y error; la diferencia radica en el tiempo necesario para

[11] Bacharelando em Matemática no Centro Universitário Internacional Uninter. E-mail: ronaldo.renan@outlook.com

hacerlo. Para ello, se usaron procedimientos de investigación cualitativa. La técnica para recoger información fue documental. Los instrumentos utilizados fueron documentos y manuales de softwares específicos y contenido académico relativo al principio fundamental del conteo. El trabajo se fundamenta en revisión bibliográfica. El análisis reveló que contraseñas cortas y poco complejas pueden ser descifradas en un tiempo relativamente reducido. En pocos minutos de procesamiento informático doméstico, contraseñas con ocho caracteres, formadas exclusivamente por números, pueden ser encontradas.

Palabras-clave: contraseña; fuerza bruta; Pentest.

1 Introdução

A segurança no universo da informática é cercada de dúvidas. Por inúmeras vezes vemos notícias sobre vazamento de dados ou falhas em dispositivos de segurança. O senso comum entre os usuários de microcomputadores é que nenhum sistema é cem por cento seguro, porém, ainda se acredita que o investimento e o nível de conhecimento para a execução de ataques cibernéticos sejam restritos a entidades governamentais ou grandes grupos de cibercriminosos. Como exemplo, temos o livro *Fortaleza Digital*, do autor norte-americano Dan Brown (2005). Nesta obra de ficção, um supercomputador da Agência de Segurança Nacional dos Estados Unidos da América (NSA) consegue descobrir qualquer senha mediante ataque de força bruta e, consequentemente, acaba com o princípio da privacidade dos cidadãos norte-americanos. O conflito ético resume-se na frase do poeta romano do século I d.C, Juvenal, traduzida como "Quem guardará os guardiões?".

O presente artigo está inserido na linha de pesquisa de Ciência e Tecnologia, na temática de Matemática Computacional. A justificativa engloba o interesse social de se saber se determinado recurso computacional é realmente seguro. No ambiente informático algumas proteções podem ser interpretadas como invioláveis, mas, para as pessoas que não trabalham diretamente com tecnologia, fica difícil a diferenciação entre os sistemas de proteção. O estudo da viabilidade da quebra de senha de arquivo compactado trará uma luz sobre esse tipo de proteção e quais comportamentos diminuem ou aumentam sua eficácia. Usuários de microcomputadores, muitas vezes, não possuem conhecimento técnico especializado para julgar por si sós.

Um recurso muito utilizado na informática é a proteção de arquivos mediante senha, fundido ao recurso de compactação de arquivos; os arquivos compactados e protegidos por senha tornaram-se comuns. Empresas de telefonia e bancos utilizam recurso similar para o envio "seguro" de faturas e informativos. A senha por padrão consiste nos primeiros dígitos do CPF do cliente. Uma senha de poucos caracteres formada apenas por algarismos numéricos possui um intervalo pequeno de possibilidades e, consequentemente, é encontrada mais facilmente.

O objetivo geral é a análise da viabilidade da quebra de senha de arquivo compactado utilizando recursos domésticos de computação. Mediante a capacidade de processamento e *softwares* livres específicos, é possível realizar um ataque de força bruta. Ataque de força bruta é uma abordagem computacional em que não sabemos qual a senha de acesso de um determinado recurso, porém, sabemos que ela pertence a um intervalo de possibilidades. O ataque de força bruta testa todo o conjunto possível de senhas, encontrando a correta.

O objetivo específico é a definição do tempo necessário para a realização do ataque de força bruta. O *software* específico define uma velocidade de testes por segundo que o processamento computacional pode fazer. O princípio fundamental da contagem define o número máximo de possibilidades de senhas. Dividindo o total de senhas possíveis pelas tentativas realizadas por segundo temos a estimativa de tempo máximo para que a senha seja encontrada.

O artigo utilizará uma abordagem qualitativa. A técnica de obtenção de informações será do tipo documental. Os instrumentos utilizados serão as documentações e o manual de utilização do *software* livre *Fcrackzip*, do autor Marc Lehmann (2008), entre outros. O conteúdo acadêmico utilizado — referente ao princípio fundamental da contagem — foi retirado do livro *Análise Combinatória e Probabilidade*, de Lauro Igor Metz (2018). A definição do alfabeto segue regras do formato de codificação de letras nos computadores; literatura a respeito foi consultada no artigo de Paulo Feofiloff (2018). Informações educativas aos usuários de microcomputadores foram retiradas do documento *Cartilha de Segurança para Internet*, disponibilizado pelo Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil.

No primeiro momento apresenta-se o princípio fundamental da contagem e sua aplicação na definição do número de senhas possíveis conforme o tipo de alfabeto utilizado. Alguns exemplos são citados para melhor entendimento. Em seguida será estimado o tempo máximo para que a senha seja descoberta. A partir da velocidade de testes por segundo, podemos calcular qual o tempo necessário para que todas as possibilidades de senhas sejam verificadas. Senhas curtas e simples deverão ser encontradas rapidamente, entre minutos e horas. Senhas grandes e complexas exigirão maior tempo para que sejam encontradas.

2 Metodologia

O artigo utiliza abordagem qualitativa, pois analisa e interpreta as consequências das complexas ações do comportamento humano no resultado final. Segundo Cervo, Bervian e Silva (2006, p. 20), "os fatos humanos e sociais implicam maior complexidade do que os quantitativos ou físicos". A técnica de obtenção de informações será do tipo documental; os *softwares* livres possuem documentação elaborada pela comunidade de programadores independentes, onde todos podem contribuir com o desenvolvimento do projeto, utilizando e aprimorando o código-fonte conforme conhecimento e necessidade.

A coleta de dados foi realizada a partir do conjunto de programas que fazem parte do sistema operacional *Kali Linux*. Este sistema de *software* livre baseado em sistemas *Debian* conta com ferramentas desenvolvidas para a realização de testes de penetração, pesquisa em segurança, computação forense e engenharia reversa. Nosso foco será nas ferramentas de teste de penetração, *pentest* no termo em inglês. O sistema é mantido e disponibilizado gratuitamente na internet pela empresa norte-americana "Offsec Services Limited", que oferece cursos especializados para profissionais de segurança da informação.

Os documentos utilizados são de natureza técnica, compostos por manuais de utilização, listas de revisões e páginas de ajuda e exemplos, disponibilizados pelos *softwares* componentes do sistema operacional *Kali Linux*. O conteúdo dos documentos pode ser acessado de dentro do sistema operacional através do terminal de comandos ou pela interface gráfica. Também é possível acessar o conteúdo pela internet, nos endereços específicos de cada desenvolvedor.

3 Revisão bibliográfica

3.1 Princípio fundamental da contagem aplicado às senhas

Todos sabemos sobre a importância da matemática, e apesar de respeitar seu poder como solucionadora de problemas, dificilmente aplicamos sua teoria em atividades do dia a dia. A falta de uso afasta a proficiência, e a falta de proficiência afasta a capacidade de modelar problemas e encontrar soluções. Atividades simples como escolher o que vestir ou o que comer ganham rigor matemático com a aplicação do princípio fundamental da contagem. Entender os métodos de contagem ajuda na tomada de decisões e na compreensão de determinados processos.

Quando escolhemos uma senha definimos os caracteres que ocuparão as posições da palavra. Digamos que nossa senha será uma palavra contendo quatro caracteres. Pelo

princípio fundamental da contagem chegamos ao número de combinações possíveis fazendo a multiplicação das possibilidades de cada posição:

Exemplo 1: Multiplicação das possibilidades

POSSIBILIDADES DA POSIÇÃO 1	.	POSSIBILIDADES DA POSIÇÃO 2	.	POSSIBILIDADES DA POSIÇÃO 3	.	POSSIBILIDADES DA POSIÇÃO 4	=	NÚMERO DE SENHAS POSSÍVEIS

Fonte: O autor, 2022.

Os algarismos numéricos possuem dez diferentes caracteres, os números de “0” até “9”. Para uma palavra contendo quatro caracteres, onde cada posição pode assumir um dos dez algarismos, teremos um total de dez mil combinações possíveis, de “0000” até “9999”:

Exemplo 2: Multiplicação das possibilidades com dez caracteres.

$$10 \cdot 10 \cdot 10 \cdot 10 = 10000$$

Fonte: O autor, 2022.

O alfabeto latino possui vinte e seis letras, de “a” até “z”. Para uma palavra contendo quatro caracteres, onde cada posição pode assumir uma das vinte e seis letras, teremos um total de quatrocentos e cinquenta e seis mil novecentos e setenta e seis combinações possíveis, de “aaaa” até “zzzz”:

Exemplo 3: Multiplicação das possibilidades com vinte e seis caracteres.

$$26 \cdot 26 \cdot 26 \cdot 26 = 456976$$

Fonte: O autor, 2022.

O alfabeto latino composto por letras minúsculas e maiúsculas possui cinquenta e dois caracteres, de “a” minúsculo até “Z” maiúsculo. Para uma palavra contendo quatro caracteres de comprimento, onde cada posição pode assumir um dos cinquenta e dois caracteres do alfabeto, teremos um total de sete milhões trezentos e onze mil seiscentos e dezesseis combinações possíveis, de “aaaa” até “ZZZZ”:

Exemplo 4: Multiplicação das possibilidades com cinquenta e dois caracteres.

$$52 \cdot 52 \cdot 52 \cdot 52 = 7311616$$

Fonte: O autor, 2022.

A combinação dos algarismos numéricos com o alfabeto latino e suas letras minúsculas e maiúsculas resulta em um alfabeto misto de sessenta e dois caracteres, de “0” até “Z” maiúsculo, passando por combinações como “99aZ” e “ZZa0”, por exemplo. Para uma palavra contendo quatro caracteres de comprimento, onde cada posição pode assumir um dos sessenta e dois caracteres do alfabeto, teremos um total de quatorze milhões setecentos e setenta e seis mil trezentos e trinta e seis combinações possíveis, de “0000” até “ZZZZ”:

Exemplo 5: Multiplicação das possibilidades com sessenta e dois caracteres.

$$62 \cdot 62 \cdot 62 \cdot 62 = 14776336$$

Fonte: O autor, 2022.

Alguns provedores de serviços de informática definem obrigatoriamente o comprimento de senha a ser utilizado ou definem uma senha inicial padrão com o intuito de que seja alterada no futuro, tarefa raramente realizada pelo usuário.

Conhecendo previamente o comprimento da senha, o cálculo das possibilidades será mais simples. Para definirmos qual a quantidade máxima de senhas possíveis precisamos do número de caracteres que o alfabeto utilizado disponibiliza e do número de caracteres do comprimento da palavra. Segundo Metz (2018), a ideia de que determinado acontecimento ocorre em “n” etapas diferentes, e as etapas podem ocorrer de “k” maneiras diferentes, implica que o total de possibilidades “T” é o produto entre as quantidades de cada situação:

Equação 1: Produto das possibilidades.

$$T = k_1 \cdot k_2 \cdot k_3 \cdot \cdots \cdot k_n$$

Fonte: Metz (2018).

Para as senhas, as etapas “n” sempre ocorrem com o mesmo número de maneiras “k”, pois o alfabeto pode ser integralmente utilizado em cada posição, sem restrições de que um caractere seja repetido, logo, o total de possibilidade “T” será:

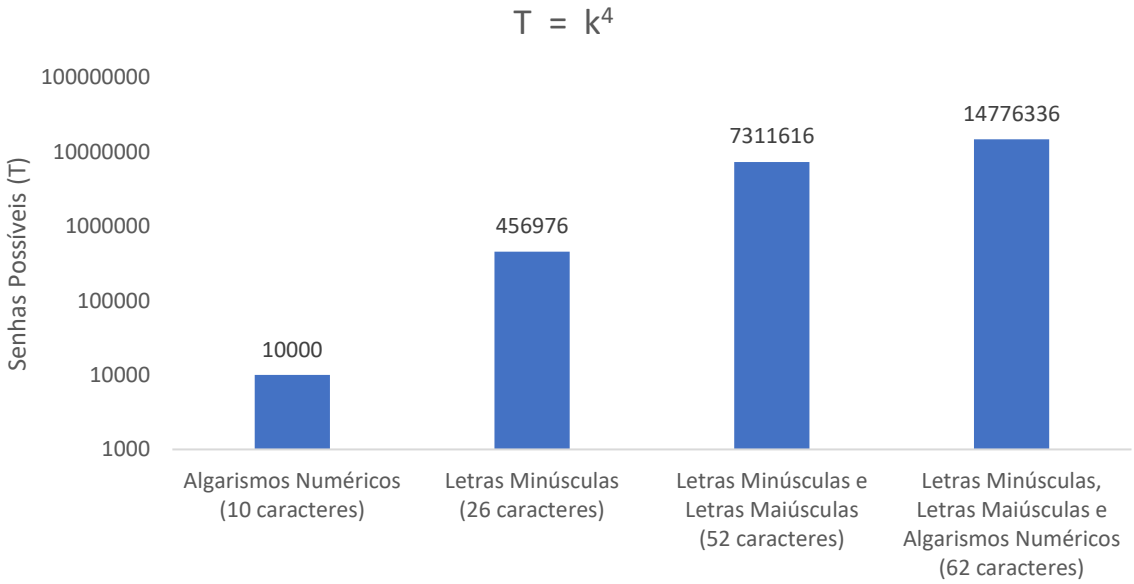
Equação 2: Equação para determinar T.

$$T = k^n$$

Fonte: O autor, 2022.

Onde, “T” é o número de senhas possíveis, “k” é o número de caracteres do alfabeto utilizado, e “n” é o número de caracteres do comprimento da palavra. Fica claro que quanto maior o número de caracteres que o alfabeto possui, maior será o número de senhas possíveis, conforme visualizamos no gráfico abaixo:

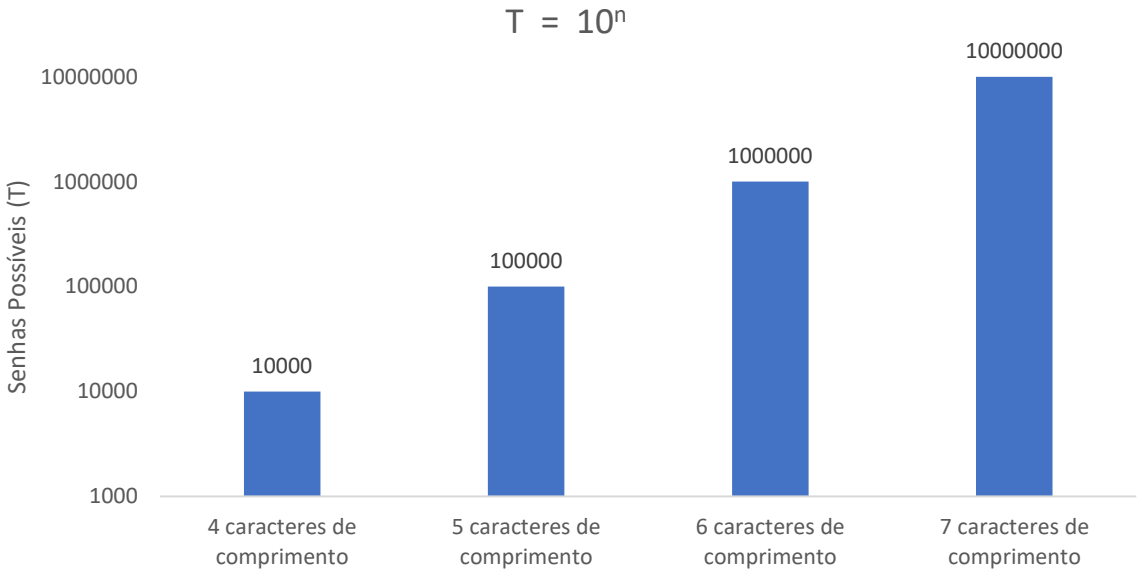
Gráfico 1: Senhas com quatro caracteres de comprimento.



Fonte: O autor, 2022.

A escala vertical utilizada no gráfico foi a logarítmica devido ao rápido aumento do número de senhas possíveis em função do aumento do número de caracteres do alfabeto utilizado. Também é válida a relação em que quanto maior o comprimento da senha para um mesmo alfabeto, maior o número de senhas possíveis, conforme visualizamos no gráfico abaixo:

Gráfico 2: Senhas com alfabeto de algarismos numéricos.



Fonte: O autor, 2022.

No gráfico acima é apresentado o número máximo de senhas utilizando os algarismos numéricos para um determinado comprimento de palavra. Serviços que utilizam os primeiros dígitos do CPF como padrão de senha enquadram-se neste cenário. Os algarismos numéricos não possuem diversidade de caracteres suficiente para gerar senhas seguras com poucos caracteres de comprimento.

Há também a possibilidade de que o comprimento da senha a ser encontrada não seja conhecido. Como exemplo temos as senhas de acesso aos serviços de correio eletrônico, que são escolhidas pelo próprio usuário. Segundo informações do cadastro de nova conta do servidor de e-mail gratuito Gmail (2022), a senha para o acesso ao serviço deve conter no mínimo oito e no máximo cem caracteres de comprimento, contendo apenas letras, números e caracteres de pontuação comuns (apenas caracteres “ASCII” padrão).

Nesta situação temos que “T” é igual à soma de todas as senhas possíveis de oito até cem caracteres de comprimento e “k” é o número de caracteres do alfabeto utilizado:

Equação 3: Produto das possibilidades para conta do Gmail.

$$T = k^8 + k^9 + k^{10} + \cdots + k^{100}$$

Fonte: O autor, 2022.

Podemos escrever a sequência de somas na forma de “somatório” e teremos a equação geral do número de senhas possíveis:

Equação 4: Equação geral para determinar T.

$$T = \sum_{i=m}^{n} k^i$$

Fonte: O autor, 2022.

Onde, “T” é o número de senhas possíveis, “k” é o número de caracteres do alfabeto utilizado, “m” é o mínimo comprimento de senha permitido e “n” é o máximo comprimento de senha permitido. Para o exemplo do Gmail temos:

Exemplo 6: Total de senhas para conta do Gmail.

$$T = \sum_{i=8}^{100} 95^i$$

$$T = \frac{95^{101} - 95^8}{94}$$

$$T = 5{,}98351 \times 10^{197}$$

Fonte: O autor, 2022.

O resultado da conta acima possui cento e noventa e oito dígitos, mostrando o quanto é grande a amplitude de senhas que podem ser escolhidas. Uma senha complexa e de comprimento elevado possui um total de combinações tão grande que implicará em uma quantidade de tempo inviável para a modalidade de ataque de força bruta.

3.2 Arquivo compactado protegido por senha

Quando pressionamos uma tecla do teclado, um sinal elétrico é transmitido para o computador. Este sinal será interpretado como um comando ou como um caractere, seguindo um catálogo preestabelecido (mapa de caracteres). Um caractere é um símbolo tipográfico usado para escrever texto em alguma língua. A língua portuguesa utiliza cento e vinte e sete caracteres e a inglesa utiliza noventa e quatro caracteres, por exemplo.

O código “ASCII” é um mapa de caracteres utilizado como padrão nos computadores originalmente desenvolvidos nos Estados Unidos da América. No código “ASCII” cada palavra possui 8 bits (1 byte), sendo que um dos bits não faz parte da codificação, sobrando 7 bits que permitem 128 posições diferentes de memória (2^7). Segundo Feofiloff (2018), apesar de 0 código “ASCII” possuir um mapa de caracteres com 128 posições, apenas 95 são caracteres utilizáveis, os demais são caracteres de controle

utilizados para tarefas específicas. Por não possuir letras com sinais diacríticos, não é possível utilizá-lo integralmente para a língua portuguesa.

Com a implementação do "HTML5" e a padronização do alfabeto "Unicode" no ambiente da rede mundial de computadores, agora temos um mapa com mais de um milhão de caracteres e com tamanho de palavra de 1 a 4 bytes. Cada caractere possui um "code point" único, que seria seu endereço no mapa. O caractere "A" maiúsculo corresponde ao "code point U+0041" e a cedilha minúscula "ç" corresponde ao "code point U+00E7", por exemplo. Conforme pretensão do projeto "Unicode", o conjunto dos caracteres que formam o "Alfabeto Unicode" contempla todos os caracteres de todas as línguas do mundo.

Para a realização do cálculo do maior número de senhas possíveis do arquivo compactado utilizaremos o alfabeto "ASCII" de noventa e cinco caracteres. Grande parte das senhas são geradas utilizando apenas este alfabeto. Apesar de não conter todos os caracteres especiais e os sinais diacríticos, ainda corresponde ao alfabeto utilizado para escrever as duzentas senhas mais utilizadas no Brasil em 2021, conforme relatório da empresa de segurança digital NordPass (2021).

A escolha da senha, além de respeitar a regra de complexidade, deve considearar a regra de confidencialidade. As informações utilizadas para sua criação não devem ser públicas. Segundo a *Cartilha de Segurança para Internet* (2020), deve-se evitar a utilização de dados pessoais, que podem ser obtidos em redes sociais, sequências de teclado e palavras que fazem parte de listas publicamente conhecidas.

Arquivos compactados ocupam menos espaço de armazenamento em disco, podendo ser transferidos mais rapidamente e armazenados em maior quantidade. Segundo informações do manual do usuário do *software WinRAR* (2022), o comprimento máximo da criptografia de senha de um arquivo compactado é de 127 caracteres. Senhas maiores serão truncadas para este comprimento. Não há limitação do comprimento mínimo, sendo, portanto, um caractere.

Temos então para o maior número de senhas possíveis do arquivo compactado a seguinte situação, onde "k" referente ao alfabeto é noventa e cinco, "m" referente ao mínimo comprimento de senha permitido que é um e "n" referente ao máximo comprimento de senha permitido que é cento e vinte e sete:

Exemplo 7: Total de senhas para arquivo compactado.

$$T = \sum_{i=1}^{127} 95^{i}$$

$$T = -\frac{95^1 - 95^{128}}{94}$$

$$T = 1{,}49794 \times 10^{251}$$

Fonte: O autor, 2022.

O resultado da conta acima possui duzentos e cinquenta e dois dígitos. Este é o valor máximo de combinações possíveis que podem ser utilizadas. Se todas estas combinações forem testadas mediante tentativa e erro, toda e qualquer senha poderá ser encontrada.

3.3 Quebra de senha mediante ataque de força bruta

Definido o total de senhas possíveis, falta definir a velocidade com que as tentativas são realizadas, para que o tempo máximo necessário para a quebra da senha seja encontrado. Segundo Lehmann (2008), o programa *Fcrackzip* realiza aproximadamente 204570 tentativas por segundo na modalidade de força bruta. Para um determinado número de senhas "T", e uma velocidade de testes por segundo "v", o tempo máximo necessário "s" para que uma senha seja encontrada será:

Equação 5: Equação para determinar s.

$$s = \frac{T}{v}$$

Fonte: O autor, 2022.

Para uma senha formada por algarismos numéricos com comprimento de quatro caracteres, temos dez mil combinações possíveis. Dividindo o número de combinações pela velocidade de testes do programa, temos que a senha correta será encontrada em até 48,88 milissegundos.

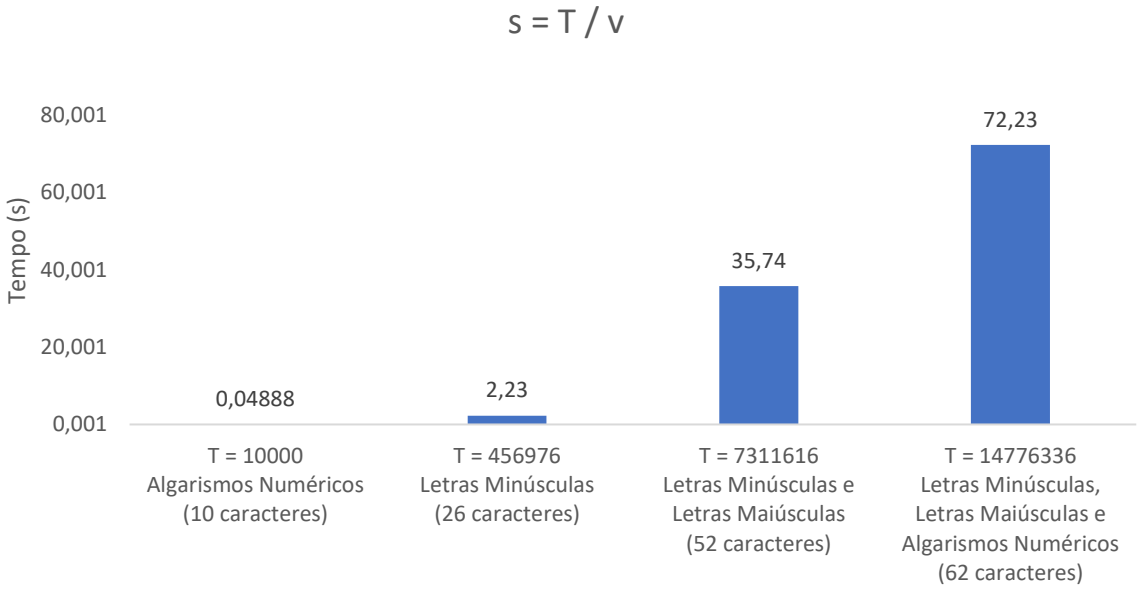
Para uma senha formada por letras minúsculas com comprimento de quatro caracteres, temos quatrocentos e cinquenta e seis mil novecentos e setenta e seis combinações possíveis. Dividindo o número de combinações pela velocidade de testes do programa, temos que a senha correta será encontrada em até 2,23 segundos.

Para uma senha formada por letras minúsculas e por letras maiúsculas com comprimento de quatro caracteres, temos sete milhões trezentos e onze mil seiscentos e dezesseis combinações possíveis. Dividindo o número de combinações pela velocidade de testes do programa, temos que a senha correta será encontrada em até 35,74 segundos.

Para uma senha formada por letras minúsculas, por letras maiúsculas e por algarismos numéricos com comprimento de quatro caracteres, temos quatorze milhões setecentos e setenta e seis mil trezentos e trinta e seis combinações possíveis. Dividindo o número de combinações pela velocidade de testes do programa, temos que a senha correta será encontrada em até 72,23 segundos.

Fica claro que quanto maior o número de senhas possíveis “T”, maior o tempo máximo necessário “s” para que a senha correta seja encontrada, conforme visualizamos no gráfico abaixo:

Gráfico 3: Tempo em segundos para encontrar senhas com quatro caracteres de comprimento.

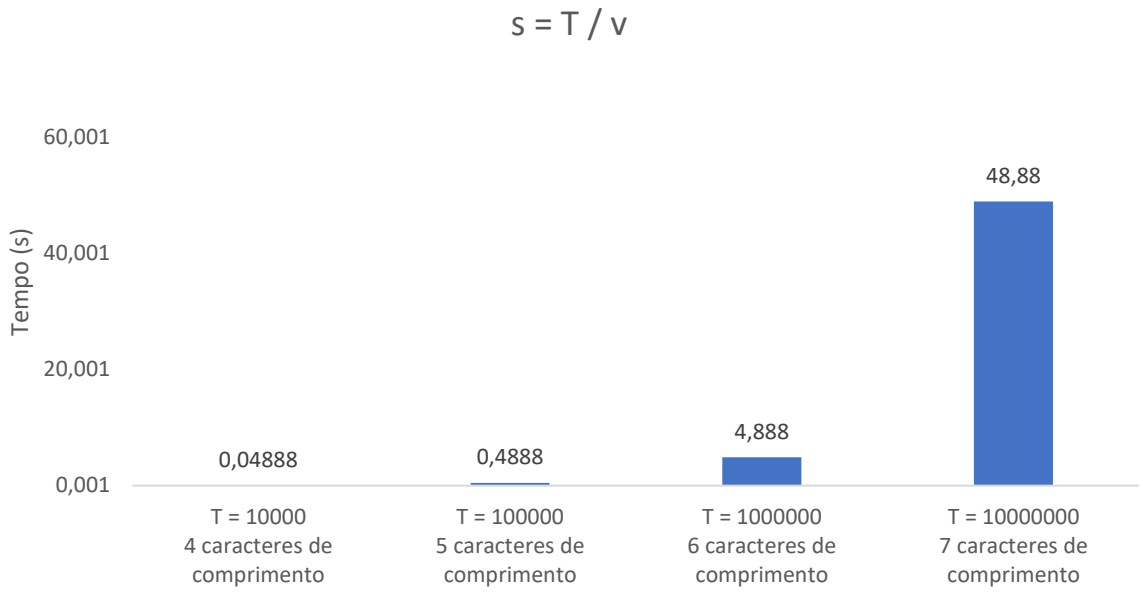


Fonte: O autor, 2022.

O tempo necessário para que a senha correta seja encontrada aumenta consideravelmente conforme aumenta a complexidade do alfabeto utilizado. O alfabeto de sessenta e dois caracteres contendo letras minúsculas, letras maiúsculas e algarismos numéricos possui apenas seis vezes mais caracteres do que o alfabeto contendo somente algarismos numéricos, porém o tempo necessário para encontrar a senha correta foi multiplicado por aproximadamente mil e quinhentas vezes.

Para as senhas compostas por algarismos numéricos, a cada caractere adicionado ao comprimento, o tempo máximo necessário para descobrir a senha é multiplicado por dez vezes, conforme visualizamos no gráfico abaixo:

Gráfico 4: Tempo em segundos para encontrar senhas de algarismos numéricos.



Fonte: O autor, 2022.

Como os algarismos numéricos possuem poucos caracteres em seu alfabeto, para uma senha considerada segura, são necessários muitos caracteres de comprimento. Considerando um tempo necessário de 1 milhão de anos, temos:

Exemplo 8: Comprimento de senha conforme o tempo.

$$1 \text{ milhão de anos} = 3{,}1536 \times 10^{13} \text{ segundos}$$

$$3{,}1536 \times 10^{13} = \frac{T}{204570}$$

$$T = 6{,}45132 \times 10^{18}$$

$$6{,}45132 \times 10^{18} = 10^{n}$$

$$n \cong 19$$

Fonte: O autor, 2022.

Para o tempo de pelo menos um milhão de anos para que a senha seja descoberta é necessário um comprimento de senha de dezenove caracteres. Na realidade cotidiana, senhas deste tamanho são raramente utilizadas. Segundo a *Cartilha de Segurança para Internet* (2020), uma senha considerada segura seria como o modelo “1 dia ainda verei os anéis de Saturno!!!”. Esta senha possui quarenta caracteres (na computação o espaço também é um caractere).

Podemos definir que o alfabeto utilizado para a escrita da senha foi o código “ASCII”, pois o alfabeto precisa contemplar os algarismos numéricos, as letras minúsculas,

as letras maiúsculas e os caracteres especiais, mas sem os caracteres com sinais diacríticos. Temos então a seguinte situação:

Exemplo 9: Exemplo de senha segura.

$$T = 95^{40}$$

$$T = 1{,}28512 \times 10^{79}$$

$$s = \frac{1{,}28512 \times 10^{79}}{204570}$$

$$s = 6{,}28206 \times 10^{73}$$

Fonte: O autor, 2022.

Para o número de combinações acima o programa demora mais de um trilhão de anos para encontrar a senha correta. Apesar de domesticamente não ser viável a descoberta desta senha pelo tempo necessário para a execução, existem supercomputadores com capacidade de realizar mais tentativas em menos tempo. Quando a abordagem de força bruta fica restrita a um número muito elevado de tentativas, geralmente o modo de ataque é substituído por abordagens mais rápidas, porém, com probabilidade menor de acerto.

4 Considerações finais

A análise demonstrou que senhas curtas e com alfabeto pouco complexo serão encontradas em intervalos de tempo relativamente curtos, sendo SIM viável a quebra de senha de arquivo compactado utilizando recursos domésticos de computação. Senhas com menos de oito caracteres e compostas apenas por letras minúsculas, um nome por exemplo, podem ser encontradas em menos de um dia de processamento computacional doméstico.

Conforme o conteúdo exposto, fica claro que a segurança no ambiente computacional é uma soma dos recursos de informática combinado com boas práticas por parte do usuário. Toda corrente é tão forte quanto o seu elo mais fraco. Se um componente for falho, é por essa falha que a segurança será comprometida.

Interessante observarmos que, para o mesmo recurso computacional, temos comportamentos distintos de segurança conforme o modo de utilização do usuário. O recurso de segurança é confiável, mas apenas se seu utilizador respeitar as normas de boa conduta com as senhas. Conforme a complexidade da senha aumenta, aumenta também o

tempo necessário para que o processamento computacional realize as tentativas necessárias.

O importante para a segurança é que a senha seja complexa o bastante para que não possa ser encontrada a ponto do seu conteúdo ter relevância. Do ponto de vista da teoria, toda e qualquer senha de criptografia, baseada em computadores binários, pode ser encontrada mediante ataque de força bruta. Uma civilização avançada futura pode ter acesso a um poder de processamento hoje considerado impossível e conseguir acesso a nossa fatura telefônica, mas daqui a milhares de anos, esta não será mais uma informação que precise ser protegida.

Referências

BROWN, Dan. **Fortaleza digital**. 1. ed. Guarulhos: Arqueiro, 2005.

CERT.BR. Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil. Cartilha de Segurança para Internet. Fascículo Senhas. 2021. Disponível em: https://cartilha.cert.br/fasciculos/#protecao-de-dados. Acesso em: 15 jan. 2022.

CERVO, Amado L.; BERVIAN, Pedro A.; SILVA, Roberto da. **Metodologia científica**. 6. ed. São Paulo: Pearson, 2006.

FEOFILOFF, Paulo. Unicode e UTF-8. IME - Instituto de Matemática e Estatística da Universidade de São Paulo. 27 de abril de 2018. Disponível em: https://www.ime.usp.br/~pf/algoritmos/apend/unicode.html. Acesso em: 08 jan. 2022.

GOOGLE LLC. Criar uma senha forte e uma conta mais segura - ajuda da conta do Google. 2022. Disponível em: https://support.google.com/accounts/answer/32040. Acesso em: 15 jan. 2022.

LEHMANN, Marc. Programa de código aberto FCrackZip. 08 de agosto de 2008. Disponível em: http://oldhome.schmorp.de/marc/fcrackzip.html ou via Sistema Operacional Kali Linux. Acesso em: 08 jan. 2022.

METZ, Lauro Igor. Análise combinatória e probabilidade. *In*: METZ, Lauro Igor (org.). **Análise combinatória**. Curitiba: InterSaberes, 2018. Cap. 1, p. 13-40.

NORDPASS - Top 200 most common passwords. 2021. Research. Nord Security 2022. Disponível em: https://nordpass.com/most-common-passwords-list/. Acesso em: 19 jan. 2022.

OFFSEC SERVICES LIMITED. Sistema Operacional Kali Linux para teste de penetração, segurança, computação forense e engenharia reversa. 2022. Disponível em: https://www.kali.org/docs/. Acesso em: 08 jan. 2022.

WINRAR. Manual do usuário – Win.rar GmbH. 2022. Disponível em: https://www.win-rar.com/start.html. Acesso em: 15 jan. 2022.