Universidade de São Paulo
Instituto de Matemática e Estatística
Bachalerado em Ciência da Computação

Rubens Douglas Roccia

Usuários respeitam as normas de criação de senhas seguras? Uma análise de datasets de senhas vazadas

São Paulo
Março de 2021

Usuários respeitam as normas de criação de senhas seguras? Uma análise de datasets de senhas vazadas

Monografia final da disciplina
MAC0499 – Trabalho de Formatura Supervisionado.

Supervisor: Prof. Dr. Roberto Hirata Junior

São Paulo
Março de 2021

Resumo

A autenticação por senha é de longe o método de autenticação mais amplamente utilizado. Por conta de seu grande valor, senhas são alvos de criminosos, tornando o vazamento de dados um grande problema atual. Este projeto tem por objetivo analisar volumosas bases de dados de credenciais vazadas utilizando o programa zxcvbn, para assim estudar os padrões de criação de senhas mais utilizados e mostrar os perigos de uma senha fraca. Ao fim das análises, foi possível observar que a entropia não é suficiente para estimar a força de uma senha, e que grande parte dessas senhas não seguem as normas de criação de senhas seguras.

Palavras-chave: senha, vazamento de dados, zxcvbn.

Abstract

Password authentication is by far the most widely used authentication method. Because of their great value, passwords are targets for criminals, making data leakage a major current problem. This project aims to analyze voluminous databases of leaked credentials using the software zxcvbn, in order to study the most used password creation patterns and show the dangers of a weak password. At the end of the analysis, it was possible to observe that entropy is not enough to estimate the strength of a password, and that most of these passwords do not follow the rules for creating secure passwords.

Keywords: password, data breach, zxcvbn.

Sumário

Capítulo 1

Introdução

Autenticações são imprescindíveis para a identificação de um usuário antes da autorização do acesso aos seus privilégios em um sistema. Em específico, a autenticação por senha é amplamente utilizada para o acesso de contas de e-mail, redes sociais e outros serviços. Para a proteção da privacidade desses usuários, a força da senha é um importante alvo de estudo do ramo da segurança da informação.

Por envolver dados pessoais e até bancários de pessoas, existe uma grande preocupação com o possível acesso à essas informações por indivíduos não autorizados. Os chamados *Data Breaches* consistem em vazamentos de informações confidenciais de um sistema, e trazem um grande impacto quando as senhas dos usuários são comprometidas.

Indivíduos mal intencionados podem se aproveitar das senhas vazadas para acessar informações sigilosas das vítimas, além de poder assumir a identidade de um usuário e pedir dinheiro em troca do acesso a conta. *Data Breaches* também prejudicam os responsáveis pelo sistema comprometido, que terão de lidar com a vulnerabilidade e arcar com as consequências do vazamento.

A motivação deste trabalho consiste em analisar estes *Data Breaches* com o intuito de trazer algo positivo para a área da segurança de dados. Este projeto tem como principal objetivo estudar os padrões de senhas utilizados pelos usuários, e de como estes se relacionam com as normas atuais de criação de senhas fortes. Que as análises aqui apresentadas esclareçam ao leitor a importância de uma senha segura, e que sirvam de apoio para estudos futuros.

1.1 Estrutura da monografia

O capítulo 2 descreve sistemas de autenticação no geral, e explica todo o processo de medição de força e armazenamento de uma senha.

O capítulo 3 aborda o tema *Data Breach*, trazendo a definição, história e impacto desse fenômeno. Também apresenta os *datasets* de senhas que serão estudados no projeto.

O capítulo 4 apresenta o software *zxcvbn*, o estimador de força de senha utilizado nas análises dos *datasets* do projeto.

O capítulo 5 traz as análises exploratórias das senhas, com gráficos e explicações dos resultados observados.

O capítulo 6 consiste nas conclusões do estudo realizado.

Capítulo 2

Senha

2.1 Autenticadores

Sistemas geralmente possuem contas com diferentes níveis de privilégio. Um *admin* terá acesso elevado à informações se comparado com um usuário comum. Certos dados pessoais serão restritos ao detentor da conta, e não poderão ser disponibilizados ao público.

Para que esses privilégios sejam respeitados dentro de um sistema, é necessário um ou mais **autenticadores** para a identificação do usuário, para então um sistema de autorização identificar suas permissões. A metodologia utilizada para autenticação pode envolver algo que o usuário **sabe**, **possui**, ou que é **inerente** à sua pessoa (FFIEC, 2005).

2.1.1 Segredo memorizável

Um método comum de autenticação é exigir que o usuário insira um segredo que o identifique.

Mais amplamente utilizado, a **senha** ou palavra-chave consiste num identificador composto por uma sequência de caracteres[1]. Levando em conta letras minúsculas, maiúsculas e dígitos, existem 56.800.235.584 possibilidades para uma senha padrão de 6 caracteres. Esse número aumenta para 735.091.890.625 se contar com todos os 95 caracteres imprimíveis ASCII. Apesar do grande número de combinações possíveis, senhas são os maiores alvos de ataques, dificultando a criação de uma senha realmente segura.

Originado com os caixas automáticos, o **PIN** é uma sequência memorizável que contém apenas dígitos. Existem 1.000.000 de possibilidades para um PIN de 6 dígitos, um número consideravelmente menor do que senhas que permitem mais tipos de caracteres.

Usado comumente para desbloquear *smartphones*, o **padrão de bloqueio** consiste em traçar um padrão na tela. Geralmente é apresentada uma figura com 9 pontos, e o usuário pode criar o padrão que quiser desde que contenha 4 ou mais pontos ligados. São 389.112 possibilidades de padrões com esse *layout*. Porém, são apenas 8.776 possibilidades ao usar um padrão de tamanho 4 ou 5, os tamanhos mais comuns. Por isso, padrões de bloqueio também estão sujeitos a ataques (Loge *et al.*, 2016).

Por ser simples e acessível, o método de segredo memorizável é utilizado massivamente, como para acessar emails, realizar transações bancárias e desbloquear celulares. Por conta disso, possuem grande valor para criminosos, sendo alvos de *hackers* e programas maliciosos.

[1] https://csrc.nist.gov/glossary/term/password

2.1.2 Biometria

Características físicas do usuário podem ser usadas como forma de identificá-las. O método biométrico de autenticação mais comum é por meio do **padrão da digital**, tecnologia utilizada por celulares modernos no lugar do PIN e do padrão de bloqueio.

Apesar de ser relativamente segura para um usuário comum, padrões de digitais podem ser roubados. Foi o que aconteceu com a ministra da defesa alemã, que teve a padrão de digital clonada por meio de fotos [2].

Há também estudos sobre a entropia dos padrões de digitais gerados por diferentes leitores, que pode ser comparado com a entropia de PINs e senhas (Young *et al.*). O cálculo da entropia de uma senha será explicado mais adiante.

2.1.3 Token

A autenticação pode ser feita por meio de um objeto ou dispositivo que a pessoa possui, como um documento de identificação.

Utilizando um cartão, um dispositivo de memória externa ou até um *software*, é possível inserir um *token*, que por sua vez armazena alguma informação que identifique o usuário. Pode ser uma palavra-chave ou até dados biométricos do usuário.

Uma técnica comum de *token* é a de gerar uma chave temporária para que o usuário insira além de suas credenciais. Isso dificulta muito um ataque de tentativa, visto que *tokens* dessa natureza expiram antes de ser possível adivinhar a chave gerada. Porém, apesar de ser um método seguro, o dispositivo do *token* pode ser roubado.

2.1.4 Autenticação multifator

Autenticação multifator consiste em utilizar dois ou mais métodos diferentes de autenticação. É uma estratégia empregada para garantir a segurança do usuário caso um dos meios de autenticação tenha sido comprometido.

Geralmente, *tokens* são utilizados como uma segundo fator de autenticação. Receber uma chave temporária por SMS para esse fim é bem comum, principalmente como forma do usuário que esqueceu a senha recuperar a conta. Porém, criminosos encontraram uma forma de burlar esse método com a técnica *SIM Swap*, que consiste em trocar o número de telefone de alguém para um aparelho em posse dos criminosos. Com o número de um usuário, é possível roubar a conta da vítima ao se aproveitar das opções de recuperação de conta do site. Foi o caso do C.E.O. do Twitter Jack Dorsey, que teve sua própria conta no site comprometida[3].

2.2 Armazenamento de senha

2.2.1 Métodos básicos

Uma importante etapa da autenticação por senha é o modo como ela será armazenada. Um método simples consiste em guardar a senha do usuário em texto plano no banco de dados. Porém, se algum agente malicioso conseguir acesso a essas informações, todas as senhas serão expostas, e a privacidade dos usuários será comprometida.

[2] https://www.bbc.com/news/technology-30623611
[3] https://www.nytimes.com/2019/09/05/technology/sim-swap-jack-dorsey-hack.html

Uma alternativa é criptografar a senha antes de armazená-la. Assim, mesmo que ocorra um vazamento de dados, as senhas estarão protegidas. Porém, é possível descriptografar essas senhas, ainda mais se a chave da criptografia também for comprometida.

2.2.2 Hashing

Hashing é o processo de passar a senha por uma função onde espera-se as seguintes propriedades: é computacionalmente inviável encontrar qualquer entrada que mapeie para qualquer saída pré-especificada; é computacionalmente inviável encontrar duas entradas distintas que mapeiem para a mesma saída[4].

Passar uma senha por uma dessas funções irá devolver o seu *hash*, que consiste numa sequência de caracteres de tamanho fixo a partir da qual é inviável retornar para a senha original.

Se os *hashes* das senhas vazarem, o *hacker* terá que *crackear* a senha. Isto é, tentar adivinhar a senha passando várias tentativas pela função *hash* até encontrar o *hash* correspondente.

Salt hashing é a técnica de mandar para a função *hash* não só a senha, mas também uma outra sequência de caracteres qualquer única do usuário chamada de *salt*. Assim, além da senha do usuário, o *hacker* também terá que conseguir ou adivinhar o *salt* enviado para a função hash, dificultando a quebra das senhas. E, mesmo que dois usuários tenham a mesma senha, quebrar uma delas não irá expor a outra caso os *salts* sejam diferentes para os dois usuários.

Outra técnica para dificultar a ação de *crackers* é conhecida como *pepper hashing*. Funciona como um *salt*, mas que não é armazenado junto com a senha.

2.2.3 Fast hash e Slow hash

Funções *hash* possuem utilidades além da autenticação de credenciais, como de checar a integridades de arquivos. Quando o objetivo de uma função *hash* é a velocidade e portabilidade, utiliza-se *fast hashs* como "SHA-1"e "MD5".

Contudo, também existem funções *hash* voltadas para a proteção dos dados contra *crackers*. Essas funções são desenvolvidas de forma que o cálculo do *hash* seja mais complicado e demorado. Alguns exemplos são "PBKDF2"e "bcrypt".

2.3 Parâmetros de segurança de senha

2.3.1 Entropia

Proposta originalmente no livro de Shannon (1948), *A Mathematical Theory of Communication*, a entropia na teoria da informação consiste na incerteza dos possíveis valores de uma variável randômica.

O grau de incerteza de uma senha pode ser calculado, assumindo a independência e uniformidade de cada caractere, a partir da seguinte fórmula:

$$H = log_2(b^l)$$

Onde b é o número de caracteres possíveis, l é o número de caracteres utilizados e H é a entropia da senha medida em bits (NIST, 2004).

[4] https://csrc.nist.gov/glossary/term/Cryptographic_hash_function

Os valores de b variam de acordo com o set de caracteres permitidos. Existem 26 caracteres de letras minúsculas e 26 de letras maiúsculas no alfabeto latino, além de 10 caracteres alfanuméricos. Os caracteres imprimíveis ASCII totalizam 95.

Uma senha com alto grau de incerteza é segura contra ataques *brute-force* simples, que testarão combinações de caracteres sem seguir uma heurística. Porém, isso é insuficiente nos tempos atuais, onde os programas de *cracking* de senhas como o *John the Ripper* utilizam ataques de dicionário, que levam em conta os padrões de criação de senhas comuns entre os usuários.

2.3.2 Medidores de força de senha

Muitos sites costumam integrar um medidor de força de senha ao cadastro de usuários para que estes criem senhas suficientemente seguras. São recomendações comuns desses medidores:

- Tamanho mínimo de 6 caracteres;
- Utilizar letras minúsculas, maiúsculas, dígitos e caracteres especiais;
- Não utilizar o nome de usuário na senha.

Naturalmente, obedecer essas regras adicionam entropia a senha, pois ela terá tamanho suficiente e todos os tipos de caracteres. A rigidez e o modo de apresentação dos medidores também são importantes influencias no comportamento dos usuários. Medidores mais lenientes parecem deixar as pessoas relutantes em escolherem uma senha julgada fraca (Ur *et al.*, 2012).

No entanto, senhas como "q1w2e3r4t5"e "P@ssword1"seriam consideradas fortes, mesmo sendo umas das mais comuns. Portanto, apenas essas dicas não são suficientes.

Medidores robustos devem, além de assegurar bons tamanho e entropia, comparar a senha com dicionários de senhas já vazadas, e também padrões comuns como sequências alfabéticas ou de teclas.

O padrão estabelecido pelo NIST, National Institute of Standards and Technology, leva em consideração a facilidade de memorização da senha por parte do usuário, em detrimento do hábito de exigir que as senhas contenham vários tipos de caracteres. Senhas grandes e memorizáveis serão seguras o suficiente para um usuário comum.

Capítulo 3

Data Breach

3.1 Definição e história

O termo *Data Breach* refere-se a um vazamento não autorizado de dados privados de um sistema.

Um dos primeiros *data breaches* foi o de DSW Shoe Warehouse em 2005. *Hackers* conseguiram acesso à base de dados da loja americana e obtiveram por volta de 1,4 milhões de números de cartão de crédito[1].

Já um dos primeiros grandes *data breaches* de senhas foi do RockYou! em 2009. As senhas estavam armazenadas em texto plano, e o serviço tinha várias falhas básicas de segurança[2].

Em 2013 aconteceu o grande *data breach* do Yahoo, que comprometeu cerca de 1 bilhão de usuários. Após uma série de vazamentos de dados desde 2012, o ataque ao site tornou-se um dos maiores *data breaches* da história[3].

Estes *datasets*, ou base de dados, costumam ser vendidos ou simplesmente liberados em fóruns. Como não é incomum reutilizar senhas, o impacto nos usuários afetados é grande.

3.2 Datasets utilizados

3.2.1 Collection #1

O Collection #1 consiste numa coleção de *datasets* volumosos de e-mails e senhas vazadas de diversas fontes, e é utilizado pelo site HaveIBeenPwned do especialista em segurança web Troy Hunt. Possui um total de mais de 2 bilhões de linhas de informação, com 21.222.975 senhas únicas[4].

Os arquivos da coleção são bases de dados de senhas vazadas feitas com foco na quantidade em detrimento da qualidade. Há disparidades na apresentação dos dados dentre os arquivos, senhas ainda no formato *hash*, *datasets* repetidos, e um grande volume de linhas de informação que traz limitações de espaço de armazenamento, tempo de análise e acurácia de gráficos. Portanto, certos *datasets* foram escolhidos dentre os demais para a confecção dos gráficos e análises:

- **Collection #1 BTC Combos**: Credenciais de plataformas de criptomoedas;

[1] https://www.nbcnews.com/id/wbna7550562
[2] https://www.theguardian.com/technology/blog/2009/dec/15/rockyou-hacked-passwords
[3] https://www.nytimes.com/2016/12/14/technology/yahoo-hack.html
[4] https://www.troyhunt.com/the-773-million-record-collection-1-data-reach/

  - 23.626.584 senhas analisadas.

- **Collection #1 RU Combo**: Credenciais de e-mails russos;

  - 40.081.692 senhas analisadas.

- **Collection #1 USA Combos**: Credenciais de e-mails americanos;

  - 37.926.300 senhas analisadas.

Boa parte das análises de padrões de criação de senha serão feitas com o Collection #1 BTC Combos. O estudo da influência da região de origem nas senhas será feito com os *datasets* Collection #1 RU Combo e Collection #1 USA Combos.

Esses arquivos totalizam 101.634.576 senhas com menos de 32 caracteres. Senhas com 32 ou mais caracteres não foram consideradas por limitação do *zxcvbn* e para evitar *hashes*.

3.2.2 Collection #2 - #5

Apesar de menos famoso, o resto da coleção de dados vazados contém um valor próximo de 1TB de dados. Esse conjunto massivo de credenciais, porém, precisaria ser tratado antes de servir como base de análise.

Os *datasets* escolhidos dessa coleção para as análises foram:

- **Collection #2 Shopping Combos**: Credenciais de sites de compras;

  - 21.609.384 senhas analisadas.

- **Collection #5 Game Combos**: Credenciais relacionadas a jogos.

  - 82.662.948 senhas analisadas.

Os arquivos totalizam 104.272.332 senhas analisadas. Tanto o Collection #2 Shopping Combos quanto o Collection #5 Game Combos serão utilizados no estudo de como o tipo de site influencia na criação de senhas.

Capítulo 4

zxcvbn

4.1 O que é o zxcvbn

A estimação da força da senha e os padrões utilizados em sua criação foram coletados por meio do programa *zxcvbn*, utilizado pelo *Dropbox*, que inspira-se em *crackers* de senhas reais para melhores resultados.

O algoritmo utiliza dicionários de senhas comuns, nomes e palavras populares nos Estados Unidos. Também leva em consideração repetições de caracteres, sequências alfabéticas e de teclas, datas, entre outros padrões.

De acordo com Dan Wheeler, criador do programa, um dos grandes problemas é o suporte apenas para palavras em inglês. Seria importante também o reconhecimento de frases comuns, e não apenas palavras[1].

4.2 Tipos de ataque

O *zxcvbn* possui quatro simulações de ataques. Estão listados em ordem crescente de acordo com a quantidade de ataques por segundo, que será inversamente proporcional à demora de crackeamento de uma senha.

4.2.1 100 ataques por hora

Simulação de um ataque online à um sistema que limita o número de tentativas de login por usuário.

4.2.2 10 ataques por segundo

Simulação de um ataque online que não possui limitador de tentativas de login, ou que este foi burlado de alguma forma. Por ser um ataque online, a velocidade ainda é limitada pela velocidade da internet.

4.2.3 10 mil ataques por segundo

Simulação de um ataque offline, ou seja, que não será limitado pela conexão de internet mas sim pelo poder de processamento do computador que está crackeando. Nessa situação, o hacker conseguiu acesso aos hashes das senhas.

[1] https://dropbox.tech/security/zxcvbn-realistic-password-strength-estimation

A simulação leva em conta a técnica de *slow hash*, que dificulta a quebra da senha.

4.2.4 10 bilhões de ataques por segundo

Simulação de um ataque offline sob senhas que passaram por *fast hash*, relativamente mais fáceis de serem quebradas que as de *slow hash*.

Com a tecnologia atual, é possível alcançar 10 bilhões de tentativas sem grandes dificuldades. Clusters de GPU, por sua vez, podem atingir até 350 bilhões de ataques por segundo[2].

4.3 Padrões

Diversos padrões de criação de senhas foram coletados pelo *zxcvbn*. Uma senha pode ter mais de um padrão de criação, como um nome e uma data de nascimento, por exemplo.

Em casos de partes da senha que se encaixem em mais de um padrão, o algoritmo tentará minimizar tanto a quantidade de padrões quanto a quantidade de tentativas para adivinhar a senha. Portanto, o programa tende a escolher palavras maiores e optar pelo padrão que quebre mais facilmente a parte da senha selecionada.

4.3.1 Dicionário

O algoritmo pesquisa partes da senha de no mínimo 2 caracteres nos seguintes dicionários:

- Lista da xato.net com as 10 milhões de senhas mais comuns (ex: 1234, senha);
- Nomes e sobrenomes do censo americano de 1990 (ex: bryan, pedro);
- Lista da Wikipedia de palavras comuns em inglês (ex: house, book);
- Palavras do Wiktionary relacionadas a filmes e séries de TV (ex: breaking, bad);

As principais informações retornadas pelo programa são:

- Parte da senha identificada em um dos dicionários;
- A palavra correspondente no dicionário;
- Em qual dos dicionários a palavra foi encontrada;
- O *rank* de frequência da palavra no dicionário;
- Se a palavra encontra-se invertida na senha ou não;
- Se a palavra utiliza *leet* ou não.

Leet consiste em substituir letras por símbolos parecidos, como substituir 'a' por '@' e 'e' por '3'. Caso uma alteração *leet* seja encontrada numa das palavras de dicionário, o programa retorna, dentre outras informações adicionais, quais caracteres foram substituídos.

[2]Notícia sobre o cluster de 25 GPUs: https://arstechnica.com/information-technology/2012/12/25-gpu-cluster-cracks-every-standard-windows-password-in-6-hours/

4.3.2 Sequência

Sequências crescentes ou decrescentes dos seguintes tipos de caracteres, considerando uma distância máxima de 5 na tabela Unicode:

- Dígitos de 0 até 9 (ex: 987, 048);
- Letras minúsculas de a até z (ex: abc, zyxw);
- Letras maiúsculas de A até Z (ex: ABC, ZYXW);
- Unicode, que engloba qualquer sequência da tabela Unicode que utilizar algum caractere fora do conjunto alfanumérico:
  - !"#$%&’
  - 4/*%

As principais informações retornadas pelo programa são:

- Parte da senha identificada como uma sequência;
- Em qual dos dicionários a palavra foi encontrada;
- Se a sequência encontrada segue a ordem ascendente ou decrescente da tabela Unicode.

4.3.3 Espacial

Uma sequência de no mínimo 3 caracteres que representem teclas espacialmente próximas. Considera os padrões de teclado QWERTY e DVORAK, além do Keypad comum e do Mac, mostrados na figura **4.1**. Alguns exemplos:

- asd
- rfv
- *-+9/87410

As principais informações retornadas pelo programa são:

- Parte da senha identificada como uma sequência espacial de teclas;
- O teclado comparado;
- Quantidade de voltas que o caminho nas teclas deu, começando com 1;
- Quantas teclas foram inseridas utilizando o *shift* (ex: trocar ’a’ por ’A’, trocar ’7’ por ’&’).

4.3.4 Regex

Todo número entre 1900 e 2019. Representam anos recentes, como possíveis datas de nascimento ou de outros acontecimentos.

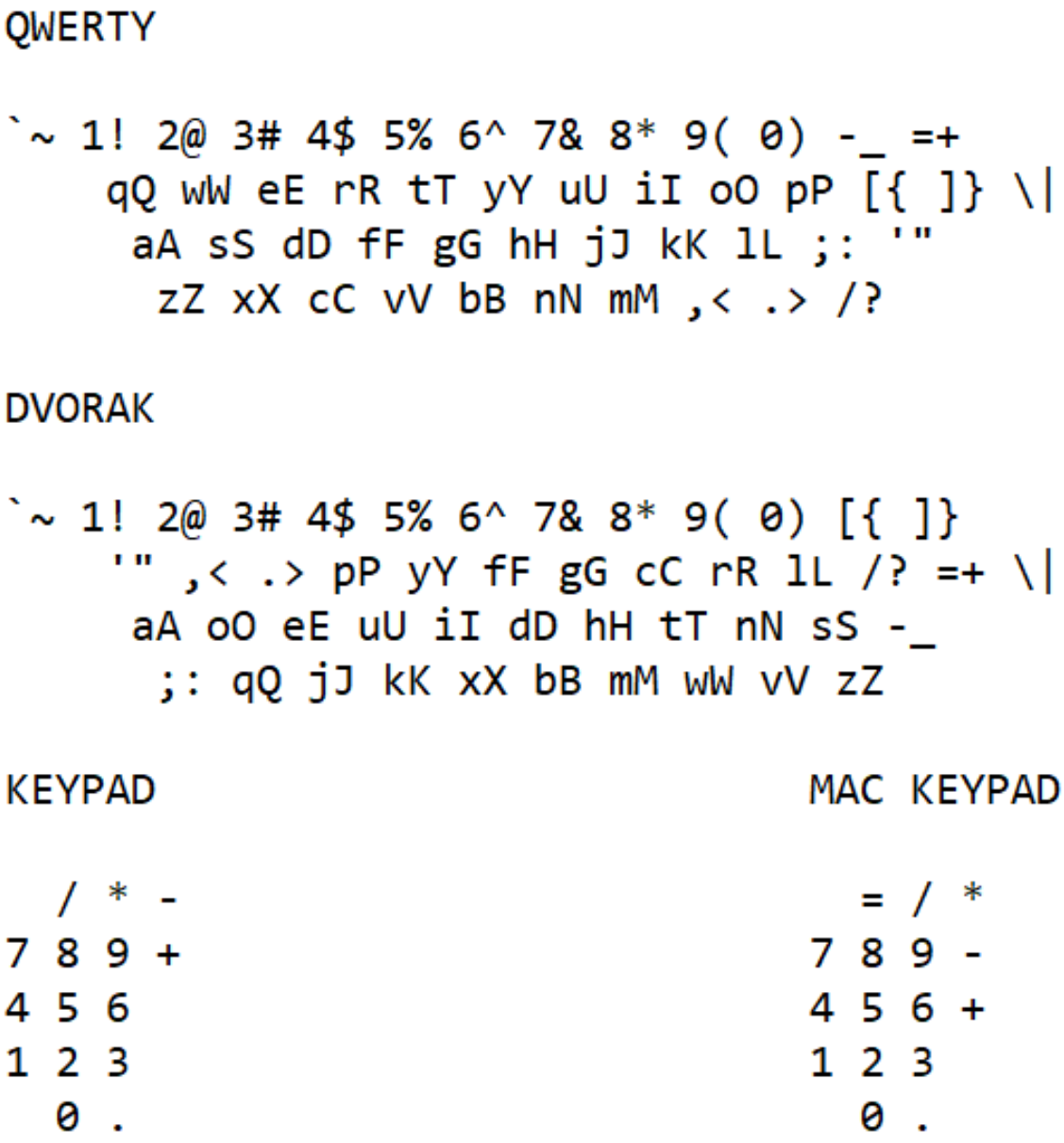


Figura 4.1: *Os quatro teclados considerados na identificação de padrões espaciais.*

4.3.5 Data

Qualquer número que se assemelhe a uma data, seguindo os seguintes critérios:

- Um dia entre 1 e 31;
- Um mês entre 1 e 12;
- Um ano entre 1000 e 2050;
- Considera tanto a presença quanto a ausência do zero à esquerda de dias e meses com apenas um dígito. Ou seja, considera 01/02/2019 e 1/2/2019.
- Anos podem ser representados apenas pela década, como ’99’ e ’17’.

  - Números entre ’00’ e ’50’ serão considerados como 20xx;
  - Números entre ’51’ e ’99’ serão considerados como 19xx;
  - Dígitos únicos serão considerados como 200x.

- Separadores podem ser ou não utilizados. São considerados os caracteres:
  - \
  - /
  - _
  - .
  - Espaço

4.3.6 Repetição

Repetição de um ou mais caracteres. Toda sequência de caracteres repetida terá seu próprio padrão, e pode ocorrer inclusive repetição de repetições (ex: ’aaaaoeaaaaoe’ é uma repetição de ’aaaaoe’, que por sua vez possui uma repetição de ’aaaa’).

As principais informações retornadas pelo programa são:

- Parte da senha identificada como uma repetição;
- A base da repetição, com seus padrões e respectivas informações;
- O número de vezes que a base é repetida.

4.3.7 Bruteforce

Padrão que não se encaixa em nenhum dos anteriores. Em outras palavras, sequências de caracteres que deverão ser quebradas por meio de buscas exaustivas, tentando todas as possibilidades.

Caracteres isolados se encaixam nesse padrão, visto que é fácil quebrar um caractere único. Sequências grandes de caracteres que se encaixam no padrão *bruteforce* são as mais demoradas para serem quebradas pela falta de heurística.

Capítulo 5

Análise exploratória das senhas

5.1 Tamanho das senhas

O tamanho de uma senha é um grande fator para sua segurança. Cada caractere aumenta exponencialmente a quantidade de combinações possíveis, dificultando o *crackeamento*.

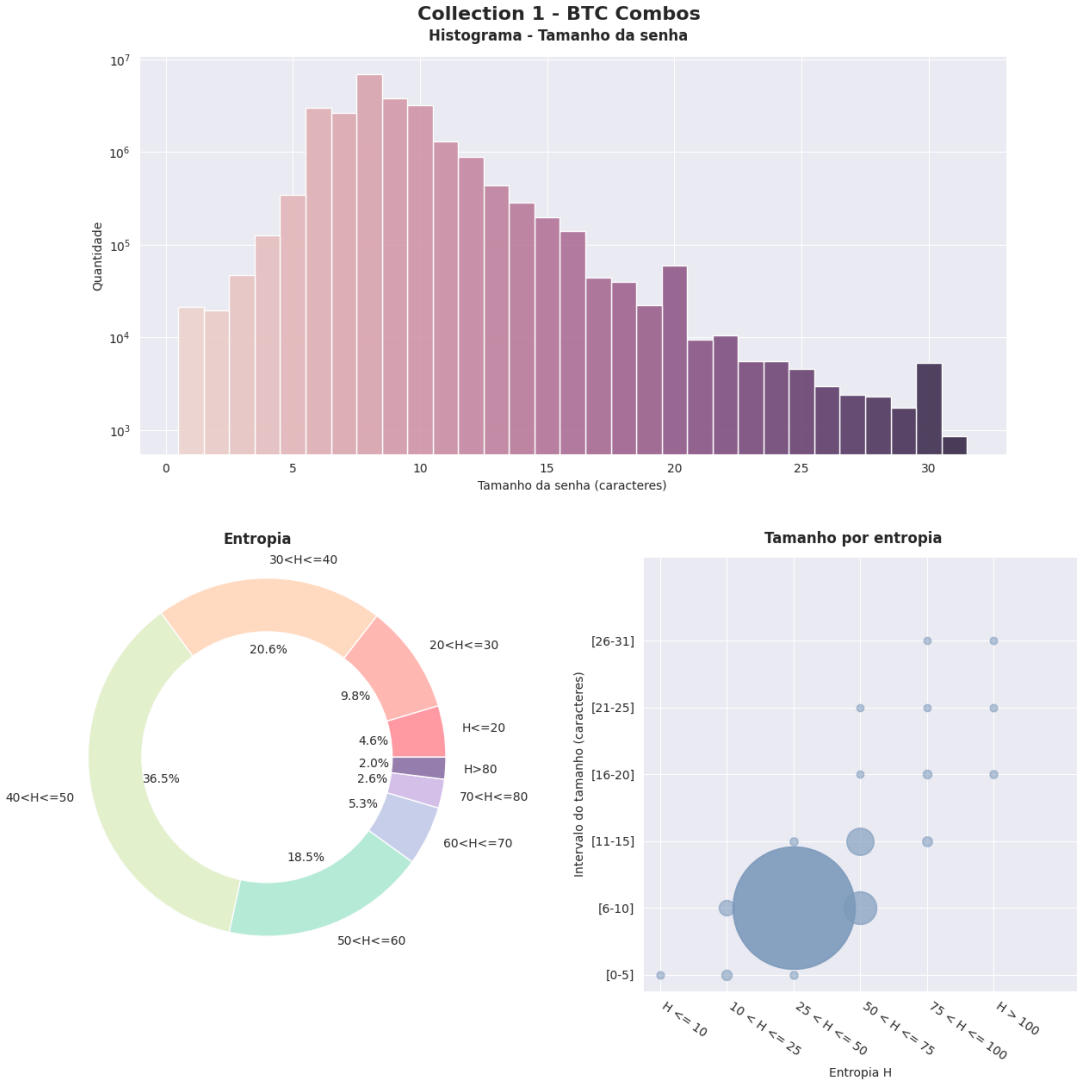


Figura 5.1: *Distribuição do tamanho e entropia das senhas do BTC Combos.*

Os gráficos da figura **5.1** mostram as distribuições dos tamanhos e entropias das senhas, bem como a importante relação entre elas. Como esperado, o diagrama de tamanho por entropia exibe uma curva crescente. A maioria das senhas analisadas possuem de 6 a 10 caracteres de tamanho, e uma entropia entre 25 e 50 bits. Esses valores de entropia representam, em teoria, uma força de senha mediana que não é facilmente quebrável nem completamente segura.

A figura **5.2**, no entanto, mostra que com apenas 10 ataques por segundo essas senhas seriam quebradas em menos de 1 dia. Simulando um ataque offline, os *hashes* de quase todas as senhas seriam quebrados em menos de 1 segundo.

Certas senhas entre 6 e 10 caracteres são suficientes para evitar o acesso de ataques online, onde o site limita o número de tentativas de login. Porém, caso os *hashes* dessas senhas acabem em um *data breach*, eles provavelmente serão quebrados facilmente. Pelas simulações, nem a técnica de *slow hash* demonstrou-se capaz de proteger essas senhas.

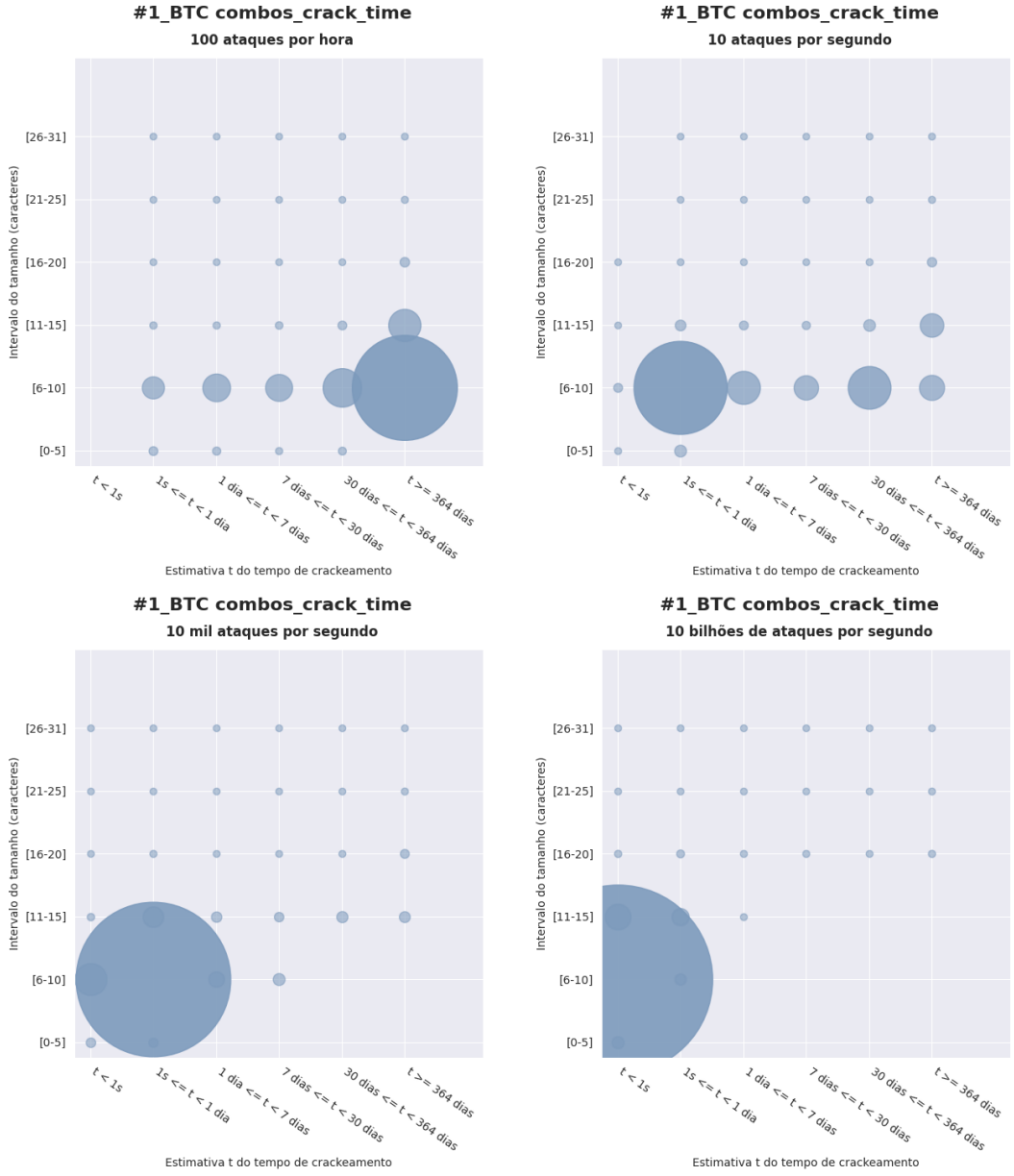


Figura 5.2: *Diagramas bolha do tamanho da senha por tempo de crackeamento.*

Apesar da maioria dos medidores de senha obrigarem o usuário a criar uma senha de no mínimo 6 caracteres, o hábito de utilizar senhas curtas ainda existe em quantidades alarmantes em sites que não possuem essa restrição.

O diagrama da figura **5.3** demonstra a baixa entropia dessas senhas curtas, justificando a facilidade de quebrá-las mostrada nos gráficos anteriores.

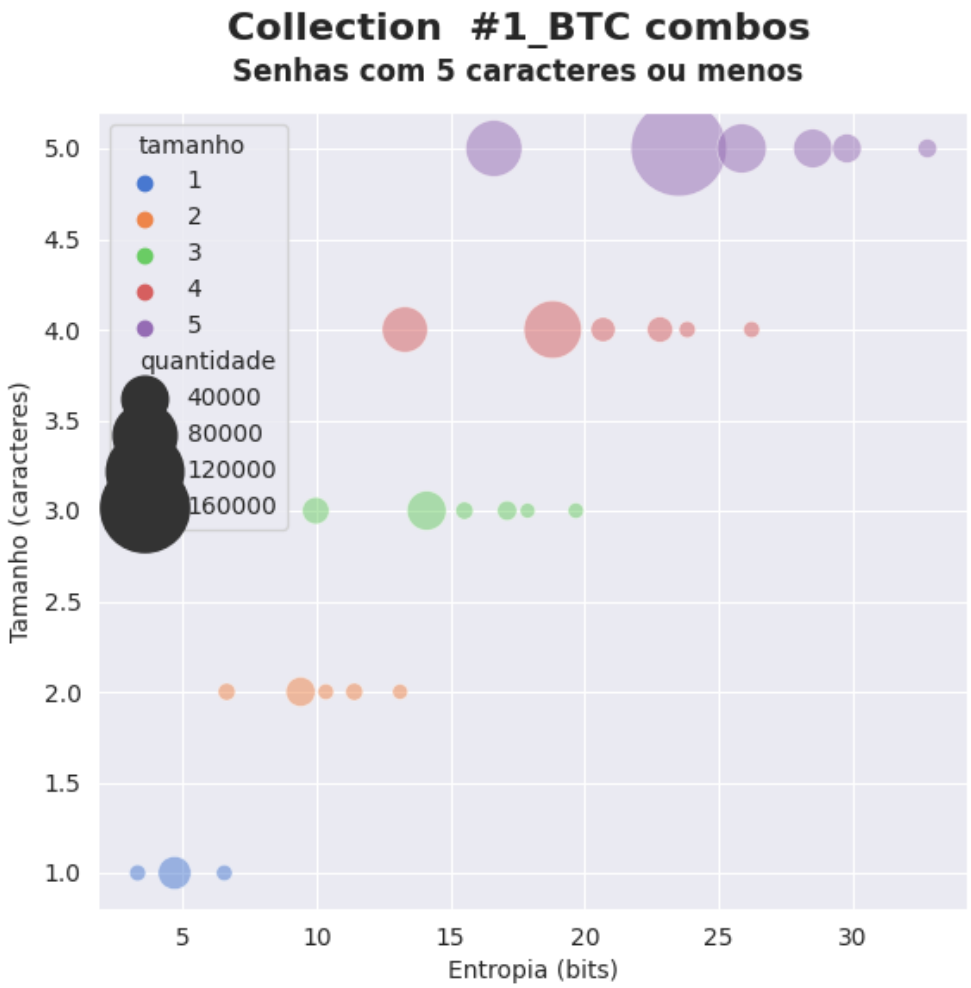


Figura 5.3: *Diagrama bolha do tamanho da senha por entropia. Apenas senhas com 5 caracteres ou menos foram analisadas.*

5.2 Caracteres utilizados

Utilizar diferentes tipos de caracteres na senha aumenta sua entropia, e é uma tática recomendada por muitos medidores de força de senha. Não é incomum sites obrigarem o usuário a utilizar letras minúsculas, maiúsculas, dígitos e caracteres especiais na criação da senha.

As figuras **5.4** e **5.5** mostram a distribuição dos diferentes tipos de caracteres no *dataset*. Dígitos e letras minúsculas predominam, enquanto letras maiúsculas e caracteres especiais compõem menos de 4% das senhas do *dataset*.

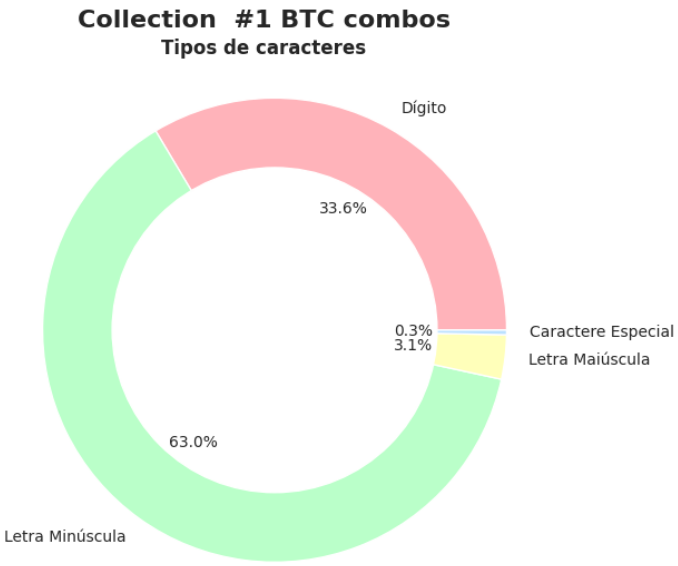


Figura 5.4: *Contagem de caracteres utilizados nas senhas.*

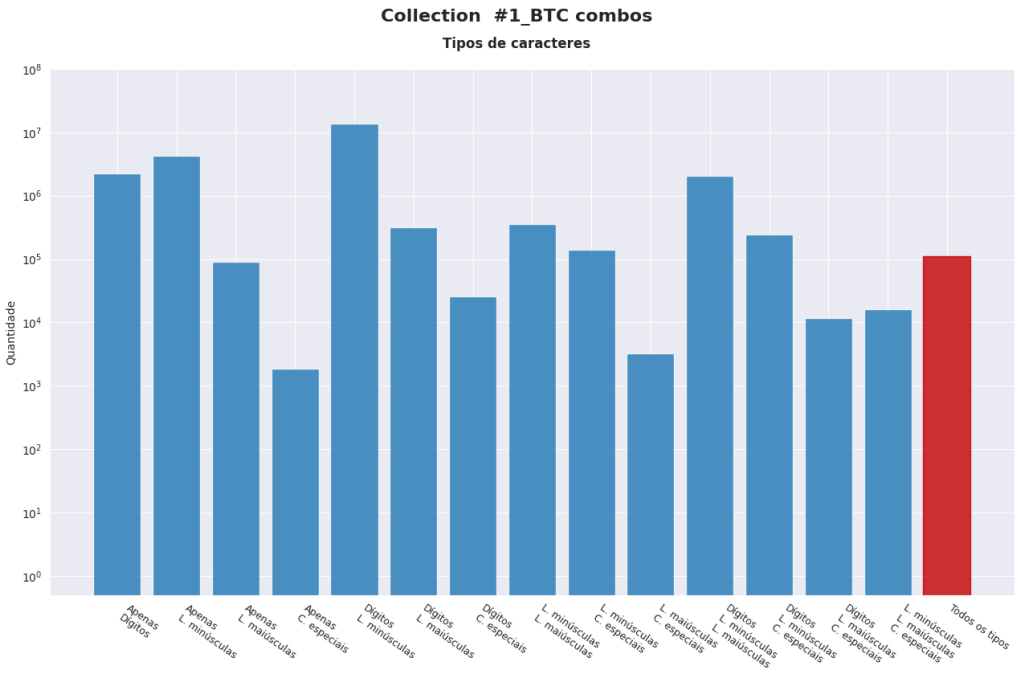


Figura 5.5: *Distribuição das senhas de acordo com os tipos de caracteres utilizados.*

5.2.1 Dígitos

Compondo grande parte das senhas, os dígitos normalmente são recomendados por medidores de força de senha para aumentar a entropia. Analisando os histogramas da figura **5.6**, é possível observar a presença de dígitos em senhas de todo tipo de força, mas principalmente nas mais fortes, mesmo sendo o tipo de caractere com menos entropia. Apesar de comum, o uso de dígitos ajuda quando não é feito de forma previsível.

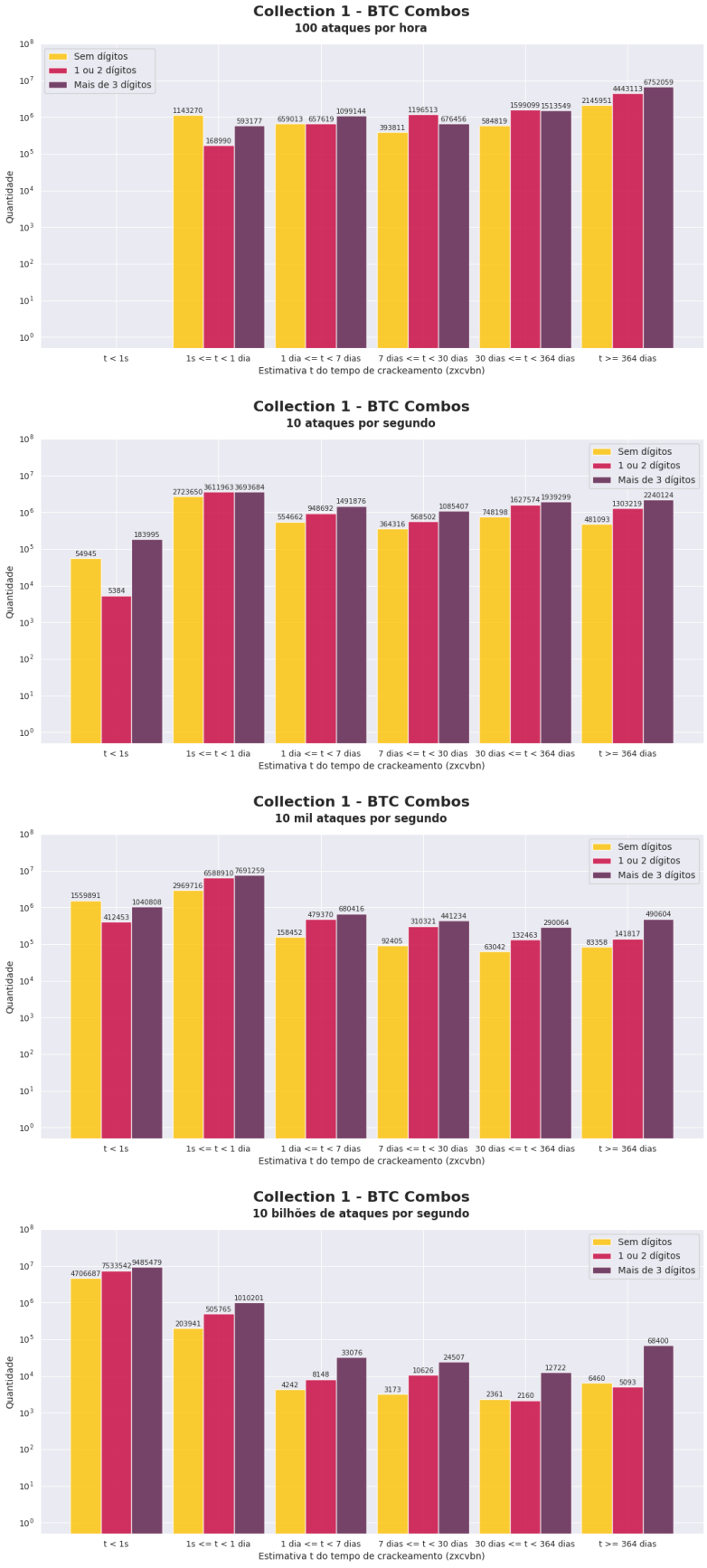


Figura 5.6: *Histogramas de tempo de crackeamento, destacando senhas com dígitos.*

Uma estratégia comum entre os usuários é a de adicionar dígitos apenas no final da senha. O ideal, porém, seria misturar dígitos com letras, visto que adicionar apenas no final é previsível. A tabela da figura **5.7** mostra os dígitos mais utilizados no final de senha, e o quanto esse padrão de criação de senha é comum: aproximadamente 11 milhões e meio de senhas seguiram esse método.

Collection #1 - BTC Combos Números no fim da senha mais comuns		
Número	**Quantidade**	**Porcentagem**
1	1.610.748	13,94%
123	548.800	4,75%
12	293.740	2,54%
2	254.080	2,20%
11	178.965	1,55%
3	134.976	1,17%
7	134.237	1,16%
13	127.836	1,11%
01	125.854	1,09%
10	111.774	0,97%
22	105.565	0,91%
1234	102.034	0,88%
5	100.595	0,87%
23	92.263	0,80%
21	85.575	0,74%
9	85.021	0,74%
4	83.932	0,73%
69	81.954	0,71%
8	81.443	0,70%
99	75.121	0,65%
...		
Total: 11.556.675		

Figura 5.7: *Números mais comuns no final das senhas.*

Apesar dos números mais comuns no final de senhas serem entre 1 e 2 dígitos, a figura **5.8** mostra alguns números comuns de maior tamanho. Consistem em repetições famosas, números relacionados a filmes como o '007', e datas prováveis de nascimento. O estudo dos padrões mostrará melhor a presença destes tipos de números em senhas.

Collection #1 - BTC Combos Números no fim da senha		
Número	Quantidade	Porcentagem
12345	49298	0,43%
666	43988	0,38%
777	42031	0,36%
007	41274	0,36%
2010	40731	0,35%
123456	39353	0,34%
101	37207	0,32%
2000	37182	0,32%
111	32556	0,28%
2009	30671	0,27%
2008	30422	0,26%
2012	27788	0,24%
2011	25852	0,22%
2007	23361	0,20%
2006	21424	0,19%
2002	21368	0,18%
2005	21301	0,18%
2001	20594	0,18%
1987	19914	0,17%
1995	19042	0,16%
...		
Total: 11.556.675		

Figura 5.8: *Outros números comuns encontrados no final de senhas.*

5.2.2 Letras maiúsculas

Letras maiúsculas são menos utilizadas em senhas do que os dígitos, sendo assim menos previsíveis. Os histogramas da figura **5.9** mostram como senhas fortes que possuem letras maiúsculas costumam utilizar mais do que duas.

A utilização de letras maiúsculas, apesar de reforçada pelos medidores de senha para aumentar a entropia, não só é pouca como também é feita de maneira previsível. São os casos de empregar letras maiúsculas no começo de um nome, ou então na palavra toda. A figura **5.10** mostra que um pouco mais de 30% das senhas com letras maiúsculas empregaram este tipo de caractere de forma previsível.

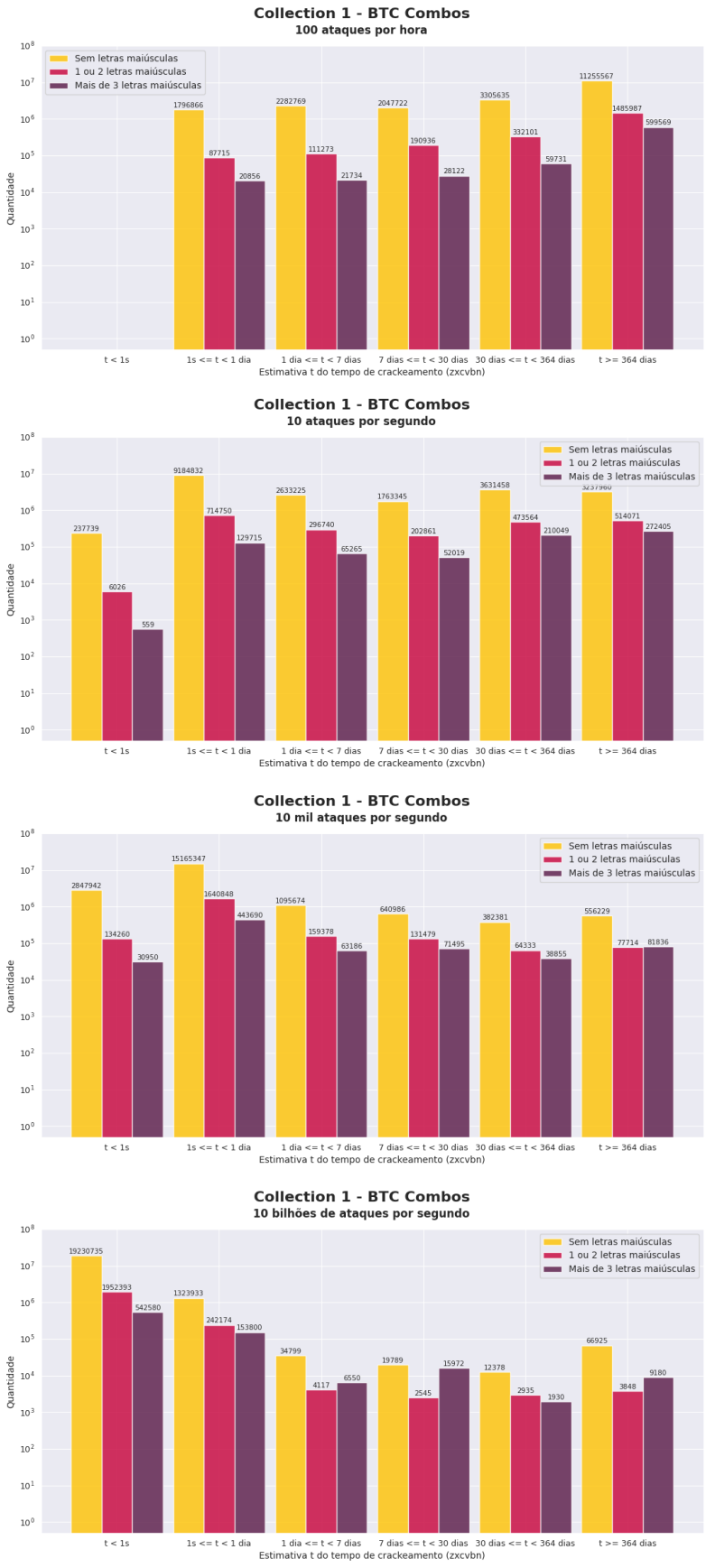


Figura 5.9: *Histogramas de tempo de crackeamento, destacando senhas com letras maiúsculas.*

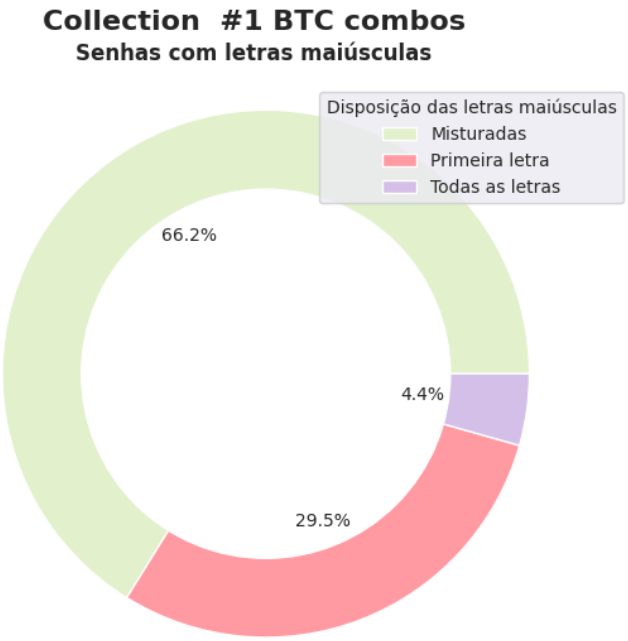


Figura 5.10: *Gráfico circular das senhas com letras maiúsculas e como elas foram utilizadas.*

5.2.3 Caracteres especiais

O tipo caractere especial compreende todo caractere imprimível que não pertença ao conjunto alfanumérico. Ou seja, acentuações, pontos, cifrões, entre outros símbolos. A tabela da figura **5.11** exibe os caracteres especiais encontrados com maior frequência nas senhas analisadas.

Collection #1 - BTC Combos Caracteres especiais		
Símbolo	**Quantidade**	**Porcentagem**
.	221.970	33,28%
!	158.732	23,80%
_	132.444	19,86%
-	106.199	15,92%
@	11.394	1,71%
&	8.970	1,34%
#	4.038	0,61%
$	3.852	0,58%
*	3.432	0,51%
:	1.765	0,26%
...		
Total: 666.948		

Figura 5.11: *Tabela com os caracteres especiais mais frequentes.*

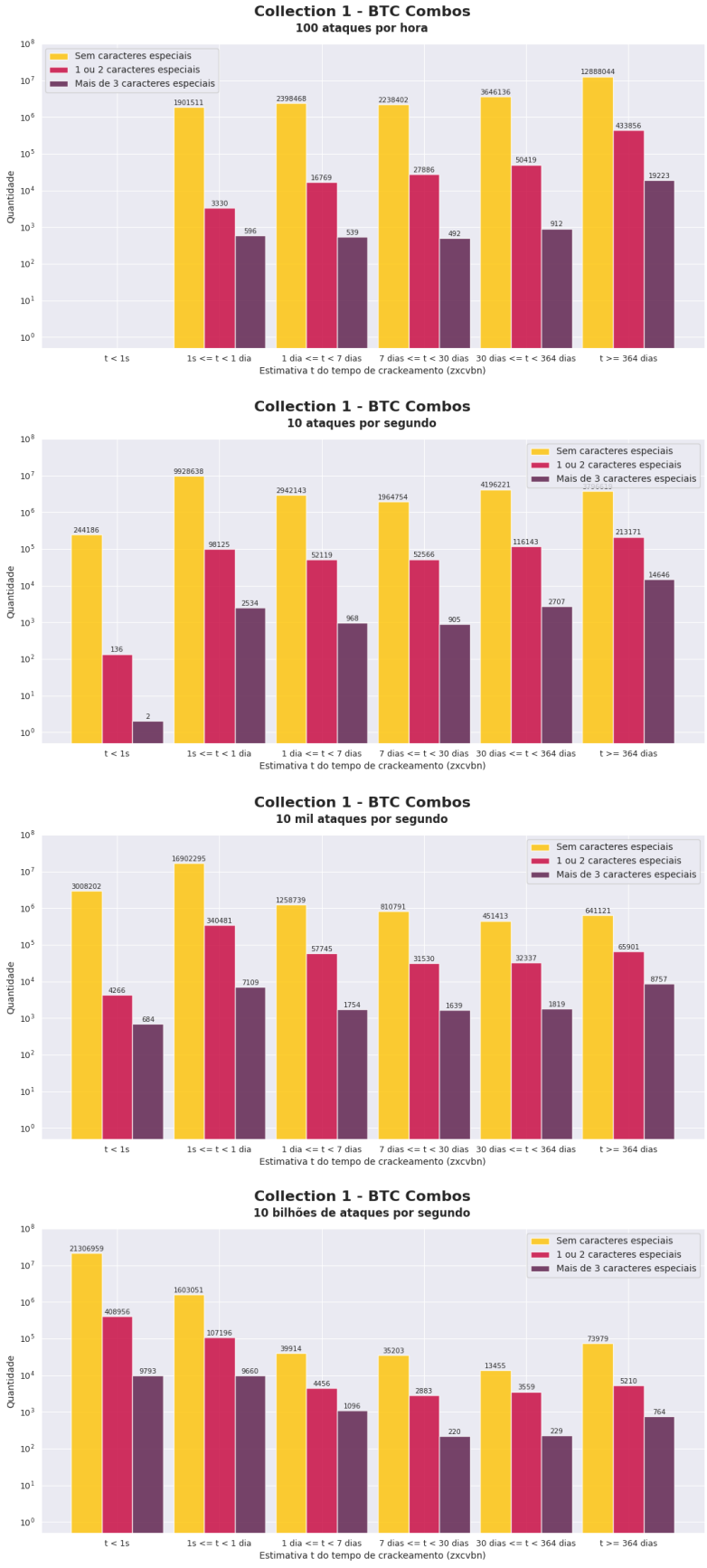


Figura 5.12: *Histogramas de tempo de crackeamento, destacando senhas com caracteres especiais.*

Por ser o tipo de caractere menos utilizado, é também o que mais adiciona força para uma senha. No entanto, até mesmo senhas com mais de dois caracteres especiais conseguiriam ser quebradas em menos de 1 dia com 100 ataques por hora, como indicado na figura **5.12**. O uso desses símbolos em certos *leets*, que será analisado depois, é um exemplo de uso previsível de um caractere especial.

5.3 Estudo de padrões nas senhas

Criações de senhas seguem certos padrões que podem ser explorados por programas de *cracking*. O histograma da figura **5.13** mostra a preferência de usuários por palavras de dicionário e datas. Sequências espaciais de teclado foram menos utilizadas que os demais padrões. Apesar da grande quantidade de padrões *Bruteforce*, não se pode afirmar que a maioria das senhas são seguras. Pequenas sequências de caracteres podem se encaixar no padrão Bruteforce e não adicionarem muita força em suas respectivas senhas.

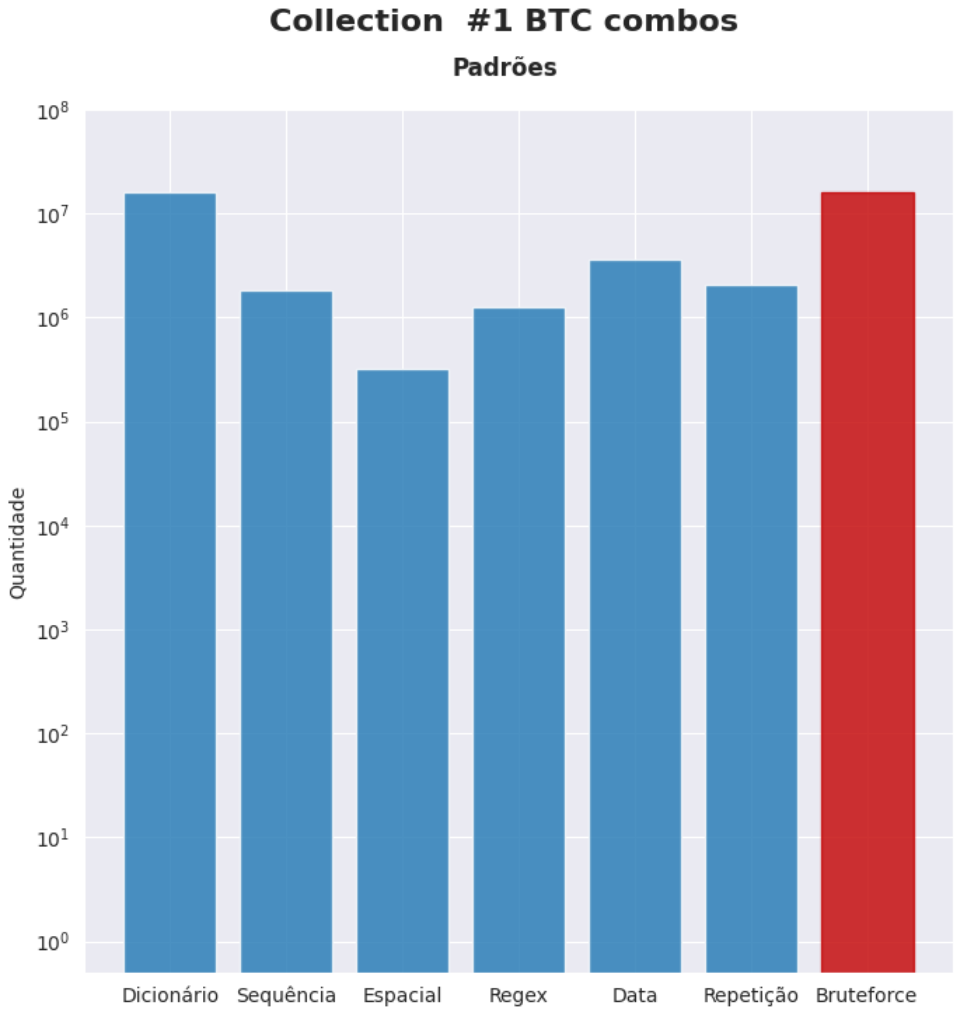


Figura 5.13: *Histograma dos padrões encontrados nas senhas.*

5.3.1 Dicionário

Dicionário foi o padrão mais encontrado dessas senhas. Observa-se, no entanto, que houve um grande número de ocorrências não só do dicionário de senhas comuns, mas também de todos os outros cinco, como pode-se observar no histograma da figura **5.14**. Nomes e palavras em inglês apareceram numa quantidade surpreendentemente grande.

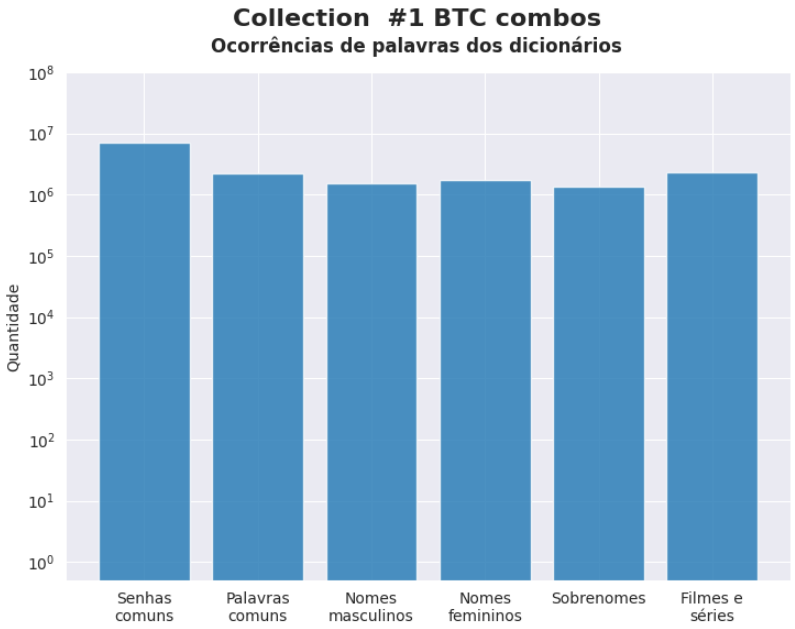


Figura 5.14: *Histograma das ocorrências de palavras dos diferentes dicionários.*

Algumas senhas apresentavam palavras dos dicionários invertidas. Apesar de ser um pouco menos previsível, inverter uma palavra não é o suficiente para enganar algoritmos de *cracking*.

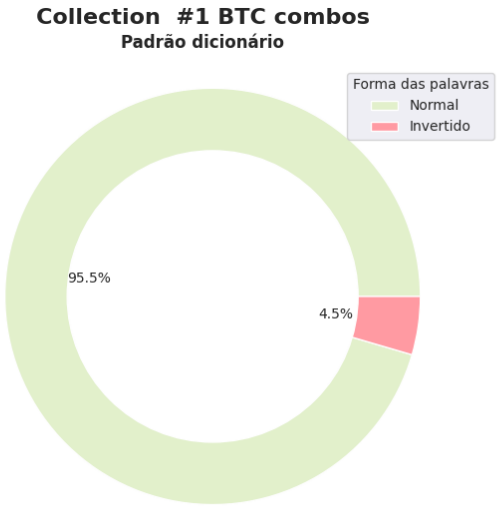


Figura 5.15: *Gráfico circular da ocorrência de palavras invertidas.*

Collection #1 - BTC Combos Dicionário: Senhas comuns		
Palavra	Quantidade	Porcentagem
123456	136.715	1,96%
1234	106.118	1,52%
12345	67.839	0,97%
123456789	65.605	0,94%
love	52.387	0,75%
password	45.304	0,65%
qwerty	38.861	0,56%
12345678	31.941	0,46%
ilove	27.221	0,39%
123123	24.732	0,36%
q1w2e3r4t5y6	23.836	0,34%
1q2w3e4r5t	19.101	0,27%
pass	18.123	0,26%
1234567	16.941	0,24%
master	16.679	0,24%
1qaz2wsx3edc	15.730	0,23%
dragon	15.622	0,22%
1234567890	15.494	0,22%
angel	15.075	0,22%
killer	14.423	0,21%
blue	14.061	0,20%
star	13.524	0,19%
super	13.353	0,19%
monkey	13.196	0,19%
password1	13.195	0,19%
111111	12.966	0,19%
iloveyou	12.871	0,18%
football	12.149	0,17%
soccer	11.811	0,17%
123321	11.723	0,17%
abc123	11.535	0,17%
shadow	11.452	0,16%
money	11.078	0,16%
dima	10.828	0,16%
fuck	10.130	0,15%
princess	10.114	0,15%
hello	9.716	0,14%
jordan	9.667	0,14%
baseball	9.576	0,14%
sexy	9.465	0,14%
000000	9.460	0,14%
azerty	9.267	0,13%
hunter	9.080	0,13%
1122	8.967	0,13%
qwerty123	8.905	0,13%
159753	8.820	0,13%
1q2w3e4r	8.569	0,12%
pink	8.544	0,12%
bear	8.228	0,12%
fire	8.094	0,12%
...		
Total: 6.965.119		

Collection #1 - BTC Combos Dicionário: Palavras comuns		
Palavra	Quantidade	Porcentagem
is	59.268	2,70%
as	55.955	2,55%
the	39.000	1,78%
at	30.225	1,38%
one	27.667	1,26%
an	26.030	1,19%
and	25.455	1,16%
are	24.508	1,12%
may	17.238	0,79%
its	16.818	0,77%
life	16.597	0,76%
in	16.118	0,73%
her	13.803	0,63%
red	13.268	0,60%
she	13.036	0,59%
has	13.007	0,59%
new	12.631	0,58%
july	11.960	0,54%
march	10.987	0,50%
san	10.319	0,47%
for	10.293	0,47%
or	10.240	0,47%
war	9.962	0,45%
art	9.522	0,43%
free	9.426	0,43%
rock	9.381	0,43%
was	8.839	0,40%
men	8.681	0,40%
doc	8.155	0,37%
per	8.043	0,37%
june	7.813	0,36%
power	7.692	0,35%
game	7.348	0,33%
age	7.249	0,33%
music	7.193	0,33%
top	7.081	0,32%
team	6.893	0,31%
best	6.624	0,30%
gold	6.579	0,30%
time	6.485	0,30%
air	6.408	0,29%
day	6.363	0,29%
had	6.126	0,28%
land	6.122	0,28%
led	6.039	0,28%
his	5.997	0,27%
of	5.874	0,27%
home	5.861	0,27%
family	5.853	0,27%
lord	5.691	0,26%
...		
Total: 2.195.509		

Figura 5.16: *Frequências das palavras identificadas como senhas comuns e palavras em inglês.*

Collection #1 - BTC Combos Dicionário: Nomes masculinos		
Palavra	Quantidade	Porcentagem
alex	29.403	1,88%
sam	21.036	1,35%
max	17.337	1,11%
dan	16.705	1,07%
ben	16.332	1,04%
tom	15.694	1,00%
daniel	15.012	0,96%
john	14.588	0,93%
david	14.248	0,91%
mike	14.082	0,90%
chris	13.552	0,87%
bob	13.190	0,84%
jack	13.124	0,84%
adam	12.020	0,77%
mark	11.806	0,75%
jay	11.751	0,75%
michael	11.588	0,74%
james	11.172	0,71%
tim	11.164	0,71%
ryan	10.251	0,66%
thomas	10.038	0,64%
joe	9.737	0,62%
jesus	9.726	0,62%
don	9.648	0,62%
andrew	9.428	0,60%
ray	9.352	0,60%
matt	9.125	0,58%
justin	9.123	0,58%
ken	9.102	0,58%
nick	9.032	0,58%
paul	8.590	0,55%
ian	8.412	0,54%
ron	8.171	0,52%
robert	8.041	0,51%
kevin	8.025	0,51%
ivan	7.798	0,50%
leo	7.718	0,49%
jose	7.629	0,49%
alan	7.389	0,47%
eric	7.327	0,47%
fred	7.223	0,46%
jon	7.218	0,46%
anthony	7.152	0,46%
joshua	7.100	0,45%
carlos	7.088	0,45%
bill	6.806	0,44%
jason	6.525	0,42%
andy	6.380	0,41%
tony	6.331	0,40%
jake	6.322	0,40%
...		
Total: 1.563.861		

Collection #1 - BTC Combos Dicionário: Nomes femininos		
Palavra	Quantidade	Porcentagem
anna	16.966	0,97%
ana	15.674	0,90%
kim	14.524	0,83%
ann	12.647	0,72%
eva	10.905	0,62%
rose	10.463	0,60%
sara	10.296	0,59%
maria	10.070	0,58%
lisa	8.938	0,51%
marie	8.857	0,51%
nicole	8.792	0,50%
amy	8.538	0,49%
roma	8.221	0,47%
jan	8.133	0,47%
anne	8.128	0,47%
april	8.099	0,46%
emma	7.976	0,46%
mary	7.957	0,46%
ella	7.849	0,45%
lola	7.845	0,45%
andrea	7.677	0,44%
sasha	7.510	0,43%
ida	7.477	0,43%
lena	7.132	0,41%
ashley	6.941	0,40%
mari	6.813	0,39%
tina	6.782	0,39%
nina	6.744	0,39%
sarah	6.728	0,39%
jessica	6.707	0,38%
michelle	6.119	0,35%
lina	6.111	0,35%
amanda	6.000	0,34%
pat	5.955	0,34%
tara	5.825	0,33%
olga	5.782	0,33%
kara	5.512	0,32%
lou	5.328	0,31%
kay	5.288	0,30%
jean	5.187	0,30%
lauren	5.089	0,29%
teri	5.021	0,29%
laura	4.953	0,28%
angela	4.863	0,28%
kate	4.792	0,27%
grace	4.675	0,27%
nova	4.665	0,27%
emily	4.635	0,27%
jennifer	4.625	0,26%
elena	4.589	0,26%
...		
Total: 1.746.821		

Figura 5.17: *Frequências das palavras identificadas como nomes masculinos e femininos.*

Collection #1 - BTC Combos Dicionário: Sobrenomes		
Palavra	Quantidade	Porcentagem
lee	33.821	2,47%
king	19.102	1,40%
black	13.118	0,96%
bell	13.025	0,95%
green	10.922	0,80%
martin	9.270	0,68%
hall	8.705	0,64%
chen	8.240	0,60%
ball	6.982	0,51%
taylor	6.818	0,50%
wolf	6.488	0,47%
morgan	6.475	0,47%
glass	6.210	0,45%
long	6.044	0,44%
pace	5.924	0,43%
wood	5.837	0,43%
bailey	5.813	0,42%
ford	5.605	0,41%
chan	5.584	0,41%
hill	5.533	0,40%
white	5.297	0,39%
moon	5.227	0,38%
alexander	5.189	0,38%
fox	5.119	0,37%
lucas	5.084	0,37%
west	4.982	0,36%
cole	4.922	0,36%
luna	4.602	0,34%
rich	4.565	0,33%
jackson	4.338	0,32%
snow	4.303	0,31%
link	4.301	0,31%
allen	4.204	0,31%
bird	4.118	0,30%
smith	4.113	0,30%
ross	4.083	0,30%
bass	4.047	0,30%
wang	3.981	0,29%
brown	3.898	0,28%
cooper	3.760	0,27%
mann	3.732	0,27%
khan	3.687	0,27%
berg	3.614	0,26%
jones	3.455	0,25%
miller	3.305	0,24%
young	3.272	0,24%
logan	3.260	0,24%
cash	3.216	0,24%
stone	3.191	0,23%
lara	3.187	0,23%
...		
Total: 1.367.992		

Collection #1 - BTC Combos Dicionário: Séries e filmes		
Palavra	Quantidade	Porcentagem
all	64.020	2,76%
it	53.395	2,30%
man	50.992	2,20%
i	40.618	1,75%
be	30.561	1,32%
god	24.604	1,06%
so	24.362	1,05%
baby	23.342	1,01%
to	22.563	0,97%
not	20.406	0,88%
can	17.808	0,77%
my	17.421	0,75%
me	16.018	0,69%
girl	15.849	0,68%
last	15.829	0,68%
big	15.712	0,68%
let	15.022	0,65%
get	13.410	0,58%
boy	13.260	0,57%
cool	12.920	0,56%
you	12.241	0,53%
ever	12.217	0,53%
see	11.869	0,51%
son	11.716	0,50%
too	11.250	0,48%
bad	10.236	0,44%
got	9.985	0,43%
go	9.874	0,43%
out	9.820	0,42%
no	9.773	0,42%
dark	9.191	0,40%
car	9.005	0,39%
mail	8.456	0,36%
kill	8.228	0,35%
do	7.967	0,34%
good	7.423	0,32%
say	7.410	0,32%
him	7.333	0,32%
mom	7.323	0,32%
happy	7.283	0,31%
will	7.203	0,31%
hot	7.141	0,31%
demon	7.070	0,30%
but	7.060	0,30%
live	6.638	0,29%
phone	6.478	0,28%
now	6.426	0,28%
sweet	6.377	0,27%
lady	6.297	0,27%
crazy	6.171	0,27%
...		
Total: 2.322.113		

Figura 5.18: *Frequências das palavras identificadas como sobrenomes e filmes e séries.*

5.3.2 Leet

Dentro do padrão dicionário, algumas palavras foram identificadas em senhas com a utilização de *leet*.

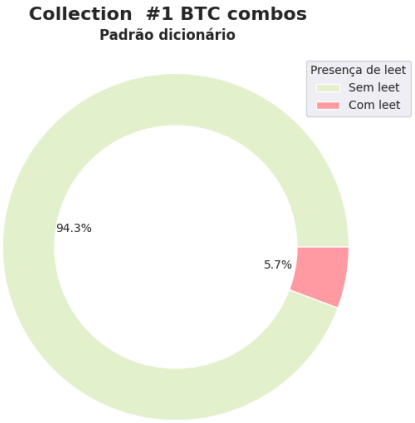


Figura 5.19: *Gráfico circular da ocorrência de leet.*

A figura **5.20** mostra os símbolos mais utilizados no lugar de letras de palavras de dicionário. O uso de dígitos nos *leets* foi bem mais frequente do que o uso de caracteres especiais.

Collection #1 - BTC Combos Substituições leet		
Substituição	**Quantidade**	**Porcentagem**
1 -> i	231.915	17,11%
3 -> e	209.616	15,47%
0 -> o	190.177	14,03%
5 -> s	181.571	13,40%
1 -> l	145.802	10,76%
4 -> a	141.075	10,41%
7 -> t	118.004	8,71%
7 -> l	49.764	3,67%
8 -> b	33.278	2,46%
9 -> g	18.249	1,35%
6 -> g	15.746	1,16%
2 -> z	14.041	1,04%
! -> i	3.390	0,25%
@ -> a	1.911	0,14%
$ -> s	717	0,05%
+ -> t	33	0,00%
| -> i	13	0,00%
(-> c	7	0,00%
| -> l	5	0,00%
< -> c	2	0,00%
% -> x	1	0,00%
Total: 1.355.317		

Figura 5.20: *Tabela com as substituições de letras por outros símbolos em palavras.*

5.3.3 Sequência

Sequências também foram amplamente utilizadas nas senhas do *dataset*. Como o esperado, sequências de dígitos predominaram, enquanto sequências de letras maiúsculas e símbolos mal foram utilizadas. Porém, aproximadamente 27% das sequências estavam na ordem decrescente, um número maior do que o esperado.

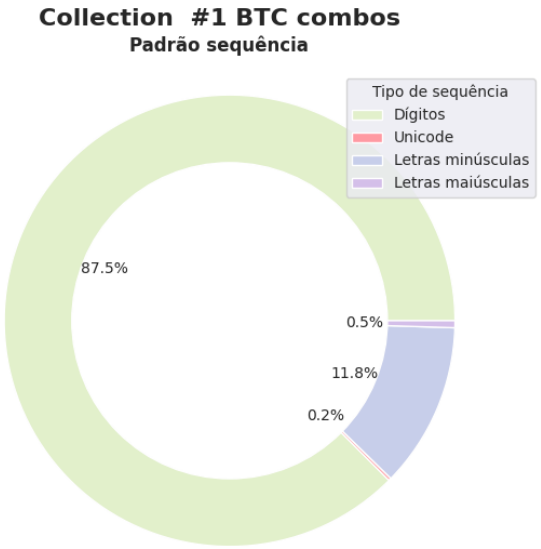


Figura 5.21: *Gráfico circular com a proporção dos tipos de sequências.*

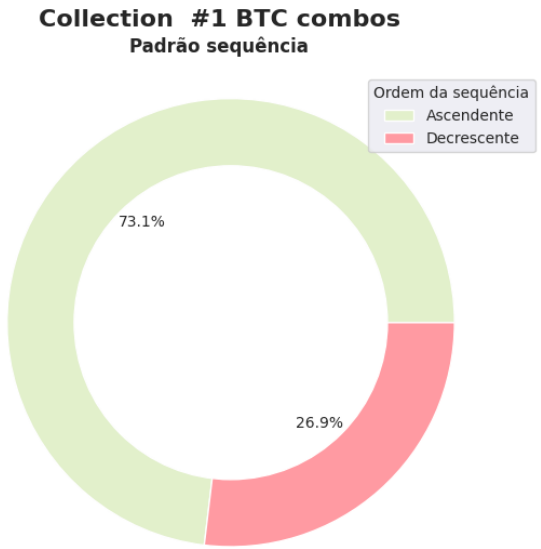


Figura 5.22: *Gráfico circular com a proporção das possíveis ordens de sequência.*

5.3.4 Espacial

O padrão espacial foi o menos utilizado na criação das senhas. Como esperado, os teclados "qwerty"e "keypad"foram os mais utilizados nas sequências espaciais. E a maior parte dos caminhos de teclas utilizados não deram mais de 2 voltas, tornando-os previsíveis.

A utilização de teclas *shift*, no entanto, apresentou uma distribuição fora do esperado. Naturalmente, a maioria das sequências espaciais não usou nenhuma tecla *shift*, mas as que usaram concentraram-se em 1 ou 4 teclas. Como foram números baixos de ocorrências, não se pode afirmar isso como um padrão recorrente em criações de senhas.

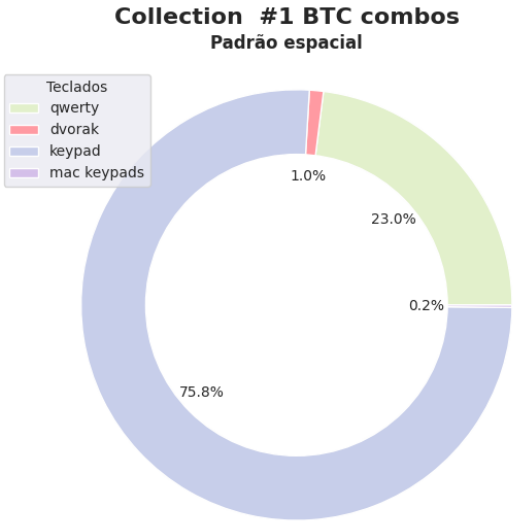


Figura 5.23: *Gráfico circular com a proporção dos teclados utilizados.*

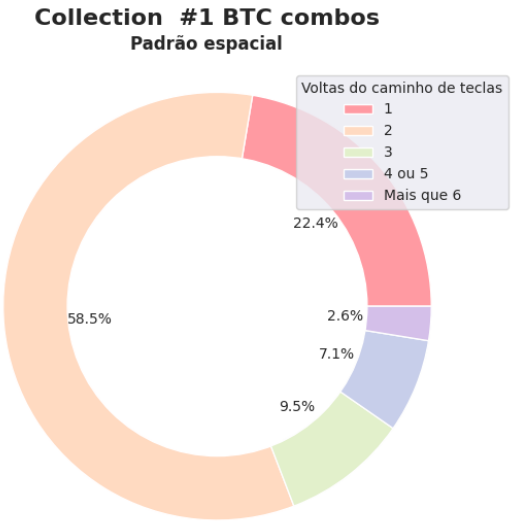


Figura 5.24: *Gráfico circular do número de voltas dos caminhos de teclas.*

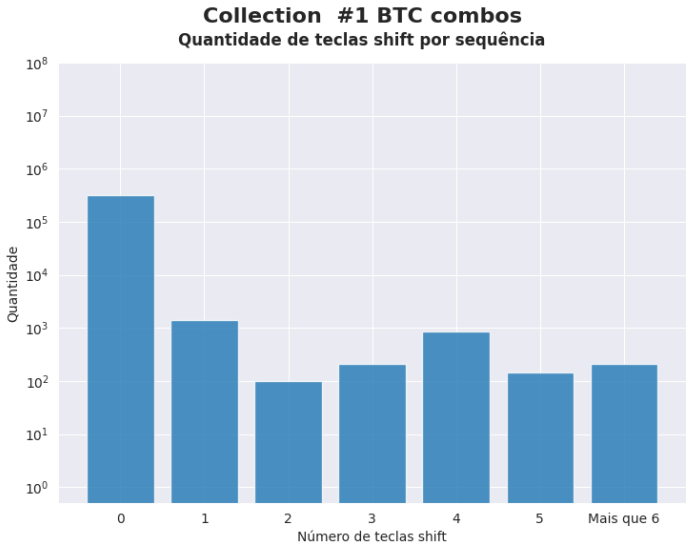


Figura 5.25: *Histograma do número de teclas shift utilizadas.*

5.3.5 Regex e data

Usar anos ou datas recentes na senha consiste num padrão bem frequente nas senhas analisadas. Aparecendo isolado ou junto com uma data, os anos encontrados não diferem muito entre os dois padrões.

Collection #1 - BTC Combos
Regex: anos

Ano	Quantidade	Porcentagem
2010	46.650	3,74%
2000	42.982	3,45%
2009	35.417	2,84%
2008	35.090	2,82%
2012	31.870	2,56%
2011	30.631	2,46%
2007	27.702	2,22%
2002	26.157	2,10%
1987	26.148	2,10%
2001	25.749	2,07%
2005	25.266	2,03%
2006	25.249	2,03%
1991	24.699	1,98%
1995	24.500	1,97%
1986	24.231	1,94%
1989	24.222	1,94%
1994	23.903	1,92%
1996	23.705	1,90%
1985	23.648	1,90%
2003	23.587	1,89%
1997	23.517	1,89%
1988	23.368	1,88%
1990	23.312	1,87%
1992	23.307	1,87%
1984	23.260	1,87%
2004	23.168	1,86%
1998	22.990	1,85%
1993	22.302	1,79%
1982	21.591	1,73%
1980	21.490	1,72%
...		
Total: 1.245.925		

Collection #1 - BTC Combos
Data: anos

Ano	Quantidade	Porcentagem
2000	119.329	3,35%
1989	67.838	3,09%
1988	63.163	2,88%
2001	62.830	2,86%
1990	59.010	2,69%
1985	58.178	2,65%
1986	57.640	2,63%
1987	57.575	2,62%
2008	56.786	2,59%
1991	55.505	2,53%
2002	55.254	2,52%
2009	54.791	2,50%
2005	54.395	2,48%
2003	54.268	2,47%
1984	54.084	2,46%
2007	53.261	2,43%
1999	52.987	2,41%
1998	52.323	2,38%
1983	51.504	2,35%
1995	51.349	2,34%
2004	50.766	2,31%
2006	50.373	2,29%
1996	50.180	2,29%
1994	49.896	2,27%
1993	48.624	2,21%
1997	48.115	2,19%
1992	47.251	2,15%
1982	46.778	2,13%
1980	44.556	2,03%
1981	44.320	2,02%
...		
Total: 3.563.545		

Figura 5.26: *Tabela com os anos encontrados nos padrões "regex"e "data".*

Apesar de se esperar uma distribuição parecida entre os dias e meses encontrados nas datas, houve uma concentração em certos números. Contudo, isso pode ter sido o resultado do modo como o zxcvbn divide a senha em padrões. O número 11, por exemplo, é muitas vezes considerado como um *leet* para "ll", diminuindo seu aparecimento em padrões de datas.

Collection #1 - BTC Combos
Data: dia

Dia	Quantidade	Porcentagem
1	226.051	6,34%
2	188.418	5,29%
8	161.294	4,53%
5	156.752	4,40%
9	156.447	4,39%
6	152.211	4,27%
7	145.566	4,08%
3	144.123	4,04%
4	143.280	4,02%
23	136.964	3,84%
25	127.904	3,59%
24	116.828	3,28%
13	114.092	3,20%
26	111.558	3,13%
14	105.700	2,97%
15	105.700	2,97%
27	104.527	2,93%
28	103.673	2,91%
22	95.921	2,69%
17	94.095	2,64%
16	93.416	2,62%
21	92.104	2,58%
29	91.775	2,58%
18	91.773	2,58%
19	90.802	2,55%
20	76.891	2,16%
30	72.924	2,05%
10	69.166	1,94%
12	67.410	1,89%
31	63.632	1,79%
11	62.548	1,76%

Total: 3.563.545

Collection #1 - BTC Combos
Data: mês

Mês	Quantidade	Porcentagem
1	359.239	10,08%
5	353.877	9,93%
2	345.040	9,68%
3	343.792	9,65%
8	330.585	9,28%
9	329.861	9,26%
6	327.756	9,20%
4	322.940	9,06%
7	322.005	9,04%
12	187.411	5,26%
10	176.629	4,96%
11	164.410	4,61%

Total: 3.563.545

Figura 5.27: *Tabela com os meses e dias encontrados em datas.*

5.3.6 Repetição

A maioria das repetições encontradas foram de dois dígitos. As diferentes sequências repetidas de caracteres não se acumularam de forma a indicar algum padrão importante de criação de senha.

5.3.7 Bruteforce

A maioria esmagadora das partes de senhas que entraram nesse padrão são caracteres isolados e sequências curtas de caracteres que não se encaixam em nenhum padrão. Existem, porém, grandes senhas de caracteres aleatorizados que não seguem nenhum dos outros padrões, e portanto são as senhas mais fortes para o tamanho delas.

5.4 Utilização do usuário na senha

Inserir partes do usuário ou email na senha apenas a tornam mais previsível, tanto que alguns sites proíbem essa prática. Para verificar as ocorrências dessa prática, foram coletados possíveis informações de usuário. Qualquer sequência de letras entre dígitos ou caracteres especiais foi considerada uma palavra, e portanto o possível nome ou apelido do usuário.

Enviar essas informações do usuário junto com a senha para análise, apesar de ter diminuído a força de algumas senhas, não fez diferença o suficiente para que seja facilmente percebido na comparação entre os histogramas de estimação de *crackeamento* de senha. Isso também decorre do fato de que o *zxcvbn* apenas verifica se a informação de usuário consta inteira na senha, e não parte dela.

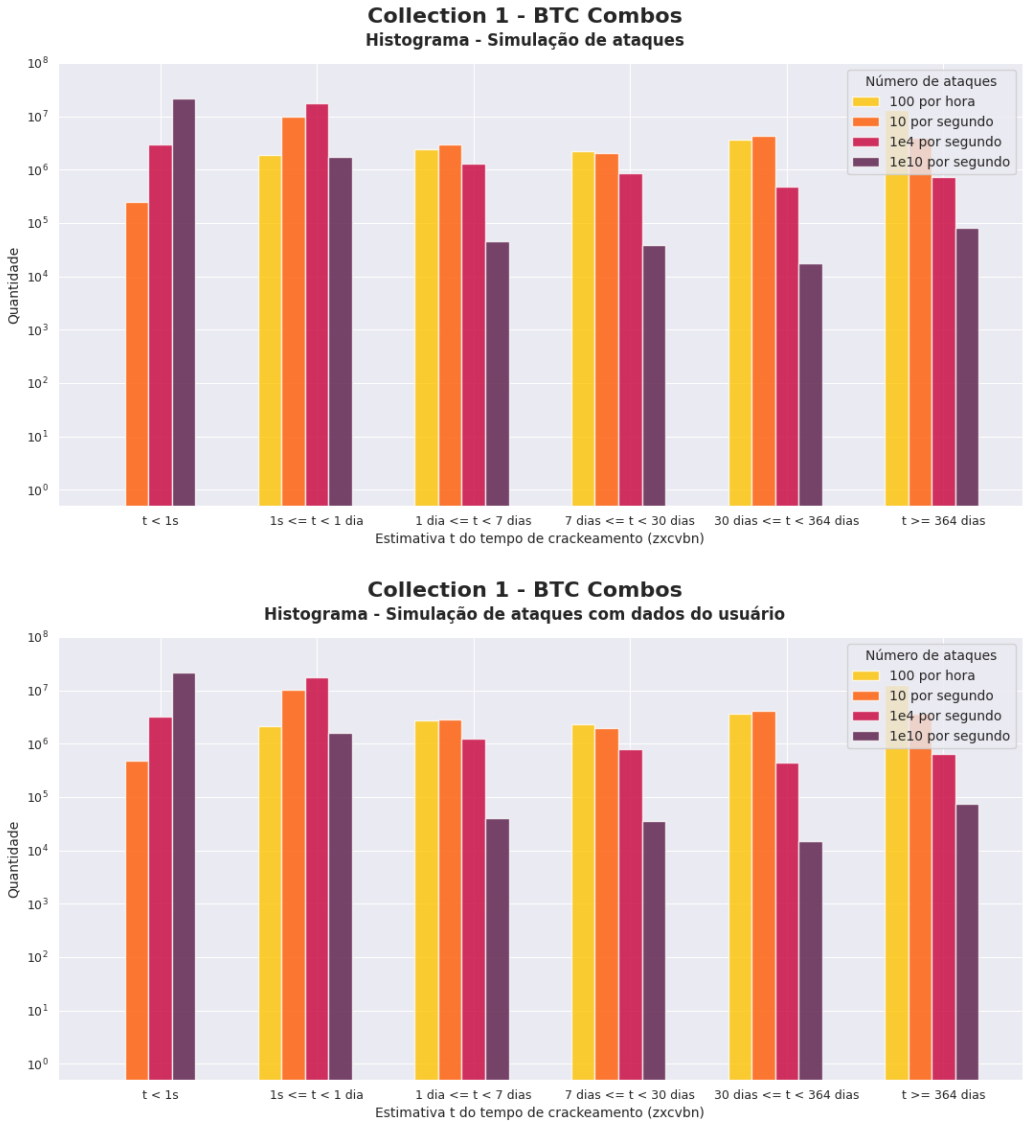
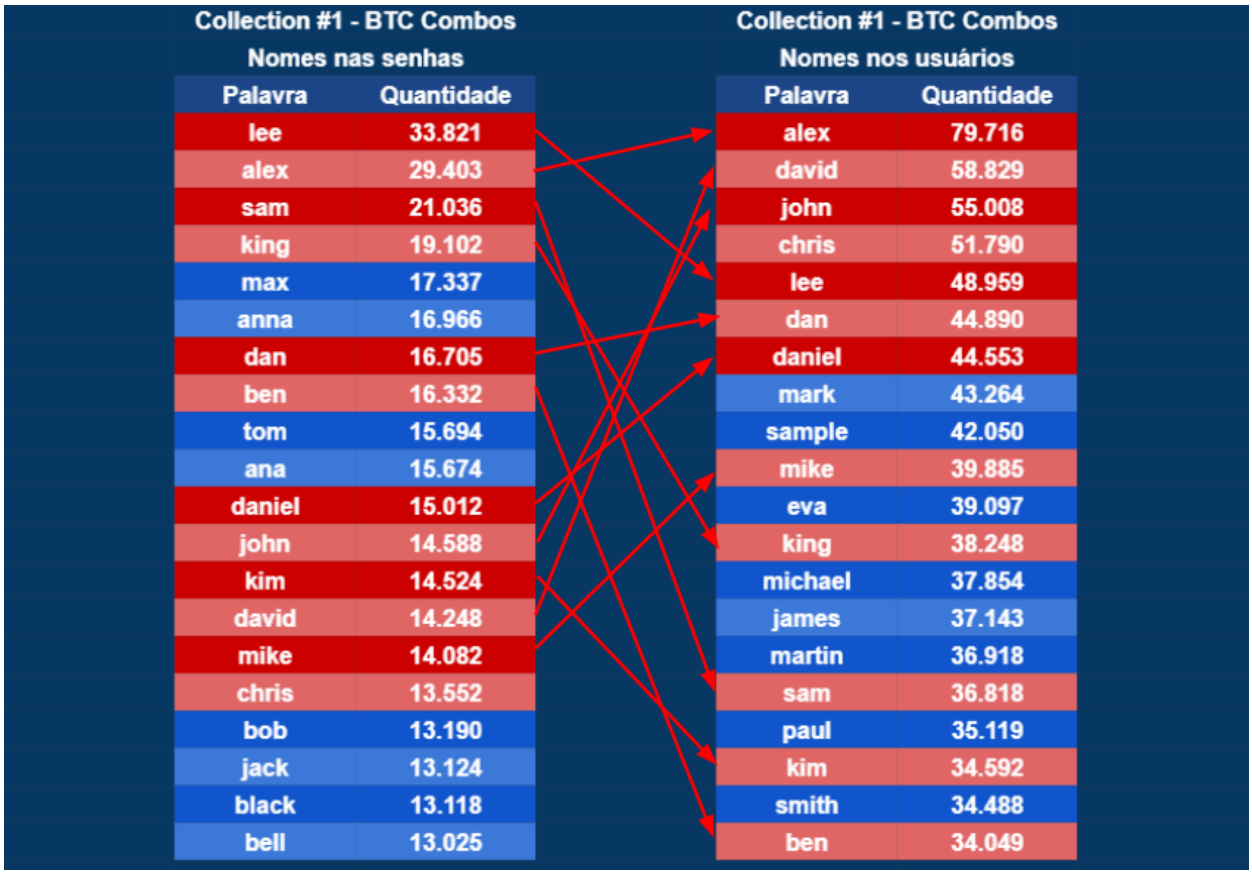


Figura 5.28: *Histograma dos tempos de crackeamento de senha sem e com o usuário.*

Um dado importante e também preocupante é a comparação entre os nomes mais comuns nas senhas com os nomes mais comuns nos usuários. Como mostrado na figura **5.29**, há correspondências entre as duas tabelas dos nomes mais comuns, indicando a utilização dos próprios nomes nas senhas.



Collection #1 - BTC Combos
Nomes nas senhas

Palavra	Quantidade
lee	33.821
alex	29.403
sam	21.036
king	19.102
max	17.337
anna	16.966
dan	16.705
ben	16.332
tom	15.694
ana	15.674
daniel	15.012
john	14.588
kim	14.524
david	14.248
mike	14.082
chris	13.552
bob	13.190
jack	13.124
black	13.118
bell	13.025

Collection #1 - BTC Combos
Nomes nos usuários

Palavra	Quantidade
alex	79.716
david	58.829
john	55.008
chris	51.790
lee	48.959
dan	44.890
daniel	44.553
mark	43.264
sample	42.050
mike	39.885
eva	39.097
king	38.248
michael	37.854
james	37.143
martin	36.918
sam	36.818
paul	35.119
kim	34.592
smith	34.488
ben	34.049

Figura 5.29: *Correspondências entre as tabelas de nomes mais comuns encontrados nas senhas e nos usuários.*

5.5 Propriedades dos datasets

Estudo de como as propriedades dos *datasets* influenciaram nas senhas.

5.5.1 País de origem

O artigo de AlSabah *et al.* (2018), *Your Culture is in Your Password: An Analysis of a Demographically-diverse Password Dataset*, mostra como a cultura transparece nas senhas. Como visto anteriormente, usuários costumam colocar informações sobre si em suas senhas, como datas de nascimento, nomes, entre outros dados.

Essa diferença pode ser percebida ao comparar os nomes mais frequentes dentre os *datasets* de credenciais da Rússia e dos Estados Unidos. Nomes como Ivan, Denis e Misha passam a se tornar mais comuns, justamente por serem mais comuns na Rússia do que nos Estados Unidos.

Esse fenômeno ressalta a importância de desenvolver medidores de senha que levem a localização em consideração. No *dataset* russo, por exemplo, foram encontradas letras do alfabeto russo, que então foram consideradas como caracteres especiais pelo zxcvbn, e assim não foram consideradas em padrões de palavras de dicionário.

Collection #1 - RU Combo
Nomes nas senhas

Palavra	Quantidade
alex	102.936
eva	63.413
anna	58.224
sasha	50.888
olga	47.560
lee	42.542
max	40.633
ivan	40.486
denis	39.457
elena	37.337
lena	36.049
anton	34.038
roma	30.809
vera	30.008
dan	28.716
roman	28.466
nova	27.822
sam	26.535
ann	25.581
alina	24.788
kim	22.556
nina	22.379
tim	21.980
ana	21.268
alena	21.261
ira	19.584
misha	18.653
lisa	18.644
tanya	18.027
mary	17.118

Collection #1 - USA Combos
Nomes nas senhas

Palavra	Quantidade
lee	58.386
alex	41.906
king	41.068
sam	35.819
john	32.105
dan	29.269
jay	27.976
kim	27.214
mike	27.135
anna	27.099
black	26.731
mark	25.952
james	25.609
jack	25.512
ryan	25.298
ana	24.989
david	24.834
max	24.028
chris	23.977
daniel	23.190
ann	23.184
tom	22.436
ben	21.922
michael	21.425
bob	20.308
rose	20.058
wang	20.002
ian	19.890
green	19.849
bell	19.421

Figura 5.30: *Comparação entre os 30 nomes mais comuns nas senhas dos dois datasets. Nomes da Rússia que não apareceram no top 30 dos EUA estão destacados de vermelho.*

5.5.2 Tipo de site

Foi verificado se o tipo de site do qual as senhas foram vazadas pode influenciar na força das senhas. Porém, surpreendentemente, os *datasets* analisados de diferentes tipos apresentaram uma distribuição bem parecida de força de senha.

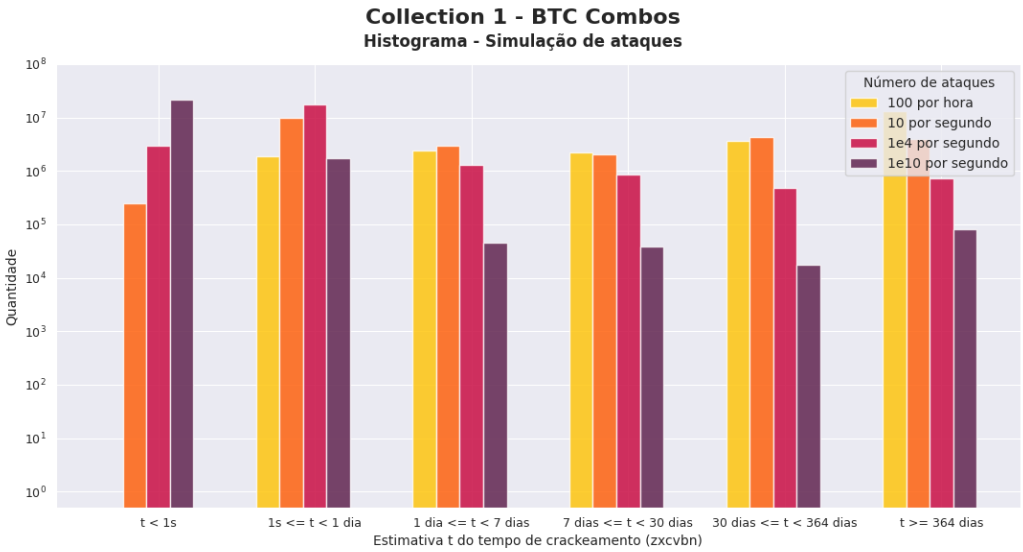


Figura 5.31: *Histograma de simulação de ataques no "BTC Combos".*

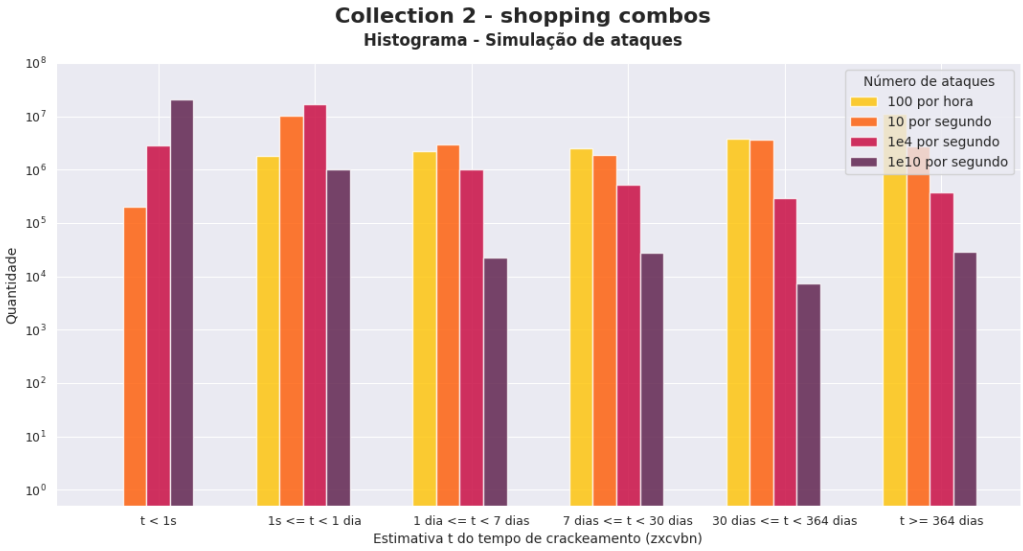


Figura 5.32: *Histograma de simulação de ataques no "shopping combos".*

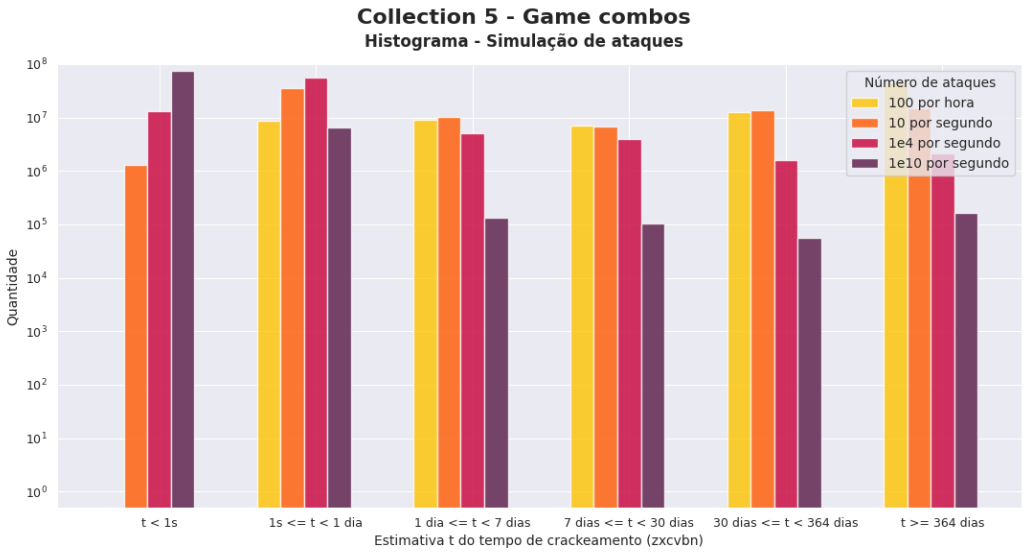


Figura 5.33: *Histograma de simulação de ataques no "Game combos".*

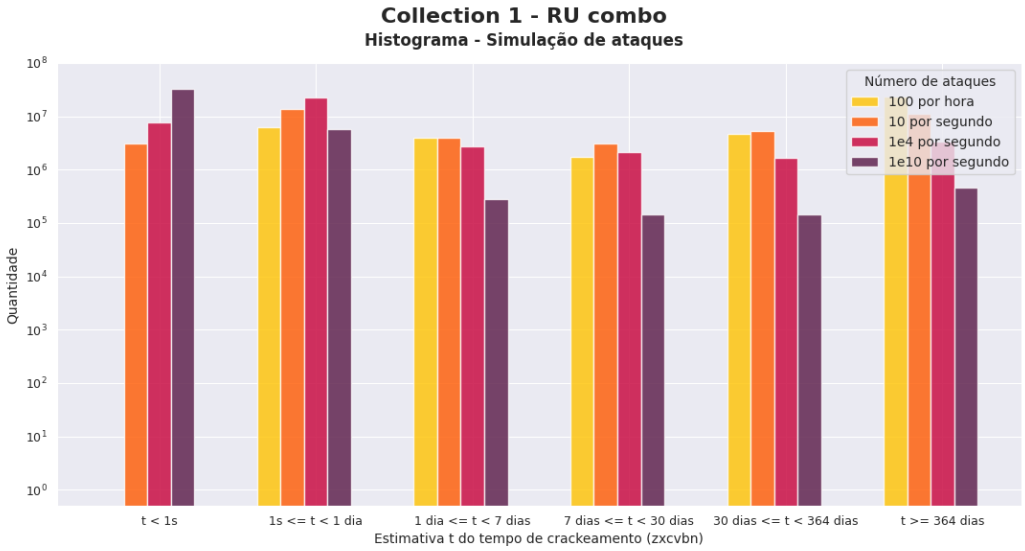


Figura 5.34: *Histograma de simulação de ataques no "RU combo".*

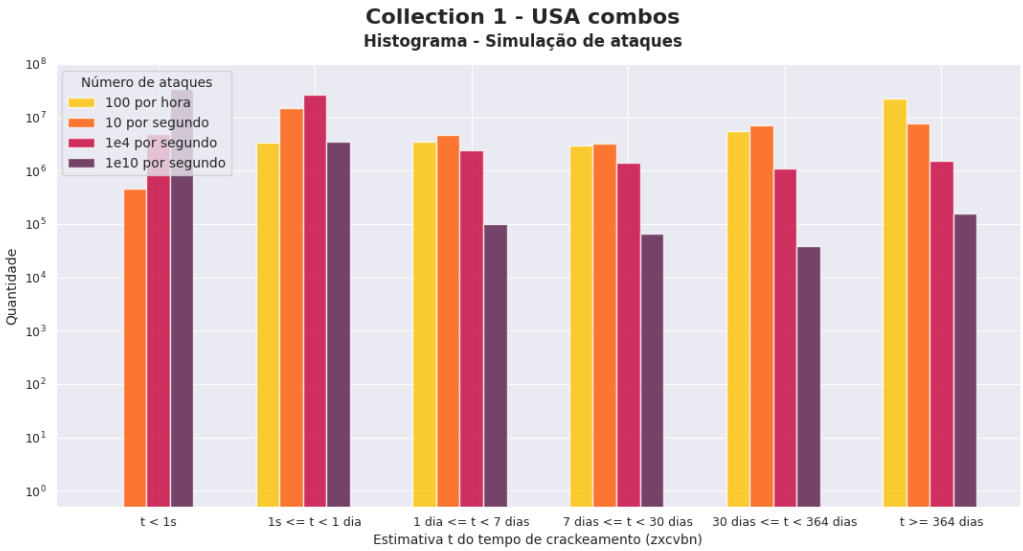


Figura 5.35: *Histograma de simulação de ataques no "USA combos".*

Capítulo 6

Conclusões

As análises das práticas comuns de criação de senha trouxeram reflexões importantes à respeito da segurança de uma senha. Apesar de boa parte das senhas apresentarem uma entropia de mais de 30 bits, a grande maioria das senhas seriam quebradas em menos de 1 dia nas simulações de 10 ataques por segundo, e em menos de 1 segundo nas simulações de 10 bilhões de ataques por segundo. Esses resultados demonstram o poder dos algoritmos de *cracking* atuais, e a necessidade de trocar de senha periodicamente para caso o *hash* ter sido vazado.

A frequência dos diferentes tipos de caracteres foi conforme o esperado, com letras maiúsculas e caracteres especiais sendo raramente usados. Ainda, todo tipo de caractere demonstrou a tendência de manter a mesma proporção de frequência tanto nas senhas fracas quanto nas fortes. Isso mostra que o que define uma senha fraca e uma senha forte vai além dos tipos de caracteres utilizados.

A tendência dos usuários de inserirem informações pessoais em suas senhas mostrou-se verdadeira nas análises dos padrões de senha. Surpreendentemente, foram encontradas palavras em inglês e nomes em quantidades tão altas quanto a aparição de senhas consideradas comuns. A concentração das datas em anos recentes indica a utilização de datas de nascimento nas senhas. E a diferença dos nomes encontrados de acordo com o país de origem do *dataset* ficou explícita.

Foram feitas tabelas com as ocorrências mais comuns de dígitos no final de senhas, palavras de dicionário, substituições *leet* e datas. Que elas sirvam de referência para os leitores se precaverem.

Que todos os gráficos e análises deste projeto guiem o leitor a proteger sua própria privacidade com práticas de criação de senhas melhores. No fim, é o usuário que decide a força de sua própria senha.

Referências Bibliográficas

AlSabah *et al.***(2018)** Mashael AlSabah, Gabriele Oligeri e Ryan Riley. Your culture is in your password: An analysis of a demographically-diverse password dataset. *Computers Security*, 77. doi: 10.1016/j.cose.2018.03.014. Citado na pág. 36

FFIEC(2005) Federal Financial Institutions Examination Council FFIEC. *Authentication in an Internet Banking Environment*. Citado na pág. 3

Loge *et al.***(2016)** Marte Loge, Markus D§rmuth e Lillian Rostad. On user choice for android unlock patterns. doi: 10.14722/eurousec.2016.23001. Citado na pág. 3

NIST(2004) NIST. *Electronic Authentication Guideline*. Citado na pág. 5

Shannon(1948) Claude E. Shannon. *A Mathematical Theory of Communication*. Citado na pág. 5

Ur *et al.***(2012)** Blase Ur, Patrick Gage Kelley, Saranga Komanduri, Joel Lee, Michael Maass, Michelle L. Mazurek, Timothy Passaro, Richard Shay, Timothy Vidas, Lujo Bauer, Nicolas Christin e Lorrie Faith Cranor. How does your password measure up? the effect of strength meters on password creation. Em *21st USENIX Security Symposium (USENIX Security 12)*, páginas 65–80, Bellevue, WA. USENIX Association. ISBN 978-931971-95-9. URL https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/ur. Citado na pág. 6

Young *et al.*() Matthew R. Young, Stephen J. Elliott, Catherine J. Tilton e James E. Goldman. Entropy of fingerprints. Citado na pág. 4