CogPIE (COGNITIVE PATTERN INFERENCE ENGINE): MOTOR DE INFERÊNCIA DE PADRÕES COGNITIVOS PARA GERAÇÃO DIRECIONADA DE CANDIDATOS A SENHAS

CogPIE (COGNITIVE PATTERN INFERENCE ENGINE): COGNITIVE PATTERN INFERENCE ENGINE FOR TARGETED GENERATION OF PASSWORD CANDIDATES

CogPIE (COGNITIVE PATTERN INFERENCE ENGINE): MOTOR DE INFERENCIA DE PATRONES COGNITIVOS PARA LA GENERACIÓN DIRIGIDA DE CANDIDATOS A CONTRASEÑAS

10.56238/IIMultiCientifica-200

André Henrique de Oliveira Santos
Especialista Sênior em Segurança Ofensiva (Red Team), OT/ICS, IT, IoT, IIoT e IA aplicada à Cibersegurança
E-mail: security.research@uniaogeek.com
Lattes: lattes.cnpq.br/9653729373562104

Davi dos Santos Chaves
Graduando em Engenharia de Software, com foco em Desenvolvimento e Engenharia de Sistemas
Instituição: Universidade do Estado do Pará (UEPA)
E-mail: ds.chaves@uniaogeek.com
Lattes: lattes.cnpq.br/3151201957160999

Eduardo Paulo Marques Raiol
Mestre em Ciência da Computação e Especialista em Desenvolvimento de Aplicações para Internet, com Pós-graduação em Ciência de Dados e Big Data Analytics
Instituição: Universidade Federal do Pará (UFPA); Universidade Estácio de Sá (UNESA)
E-mail: epm.raiol@uniaogeek.com
Lattes: lattes.cnpq.br/3897464431417513

Julio de Souza Correia
Pós-graduado em Cibersegurança e Governança de Dados, com formação executiva em liderança para a era da IA (MIT Professional Education) e certificação internacional em inovação (ITL/SEST SENAT e Nova SBE)
Instituição: Pontifícia Universidade Católica de Minas Gerais (PUC/MG)
E-mail: js.correa@uniaogeek.com
Lattes: lattes.cnpq.br/1135334336820332

Bhrandon Borges da Cruz
Especialista em mitigação de riscos, contenção de ameaças complexas e orquestração de operações defensivas para resiliência cibernética
E-mail: bb.cruz@uniaogeek.com
Lattes: lattes.cnpq.br/9299218735570810

Gustavo Gobi Martinelli
Aluno especial de Doutorado em Ciência da Computação, Professor de pós-graduação e pesquisador em Direito Digital, criptografia e proteção de dados
Instituição: Universidade Federal do Espírito Santo (UFES)
E-mail: gg.martinelli@uniaogeek.com
Lattes: lattes.cnpq.br/2631794222568810

Marcio Patrick Medeiros Cruz
Especialista em Segurança Ofensiva (Red Team), Pentest, análise de vulnerabilidades e simulação de adversários. Pós-graduado em Prevenção e Investigação de Crimes Digitais (INIBF)
E-mail: mpm.cruz@uniaogeek.com
Lattes: lattes.cnpq.br/1328364477710764

RESUMO
A segurança de senhas permanece como um dos desafios mais críticos em segurança da informação. Este artigo apresenta o CogPIE (Cognitive Pattern Inference Engine), um algoritmo híbrido de cinco camadas que modela vieses cognitivos para gerar candidatos direcionados de senhas. O CogPIE combina expansão semântica de tokens pessoais, regras de transformação cognitiva baseadas em vieses psicológicos documentados, um sistema híbrido de ranqueamento por cadeias de Markov e PCFG, pontuação de entropia prática e filtragem sensível a políticas. Nossa avaliação experimental abrange dados sintéticos, comparativos com 10 ferramentas externas e validação com 13.487 credenciais reais de vazamentos. O CogPIE atinge 89,7% de cobertura média em benchmarks WLF (10 personas), 51,38% case-insensitive contra credenciais reais e 75,95% hit@1000 com corpus GPU de 1M. Adicionalmente, demonstramos que a integração do CogPIE ao WordlistXPL-Forge (github.com/mrhenrike/WordlistXPL-Forge) elevou a cobertura média do framework de 39,97% para 89,7%, representando um salto de +124,4% relativo. Nossos resultados reforçam que cofres de senhas com geração criptograficamente aleatória, ou algoritmos de permutação não baseados em padrões humanos, permanecem a única defesa confiável contra ataques direcionados. Código-fonte: github.com/mrhenrike/CogPIE e PyPI.

Palavras-chave: Segurança de Senhas. Viés Cognitivo. Adivinhação de Senhas. Cadeias de Markov. PCFG. Ataques Direcionados. WordlistXPL-Forge.

ABSTRACT
Password security remains one of the most critical challenges in information security. This paper presents CogPIE (Cognitive Pattern Inference Engine), a five-layer hybrid algorithm that models cognitive biases to generate targeted password candidates. CogPIE combines semantic expansion of personal tokens, cognitive transformation rules grounded in documented psychological biases, a hybrid ranking system based on Markov chains and Probabilistic Context-Free Grammars (PCFG), practical

entropy scoring, and policy-aware filtering. Our experimental evaluation encompasses synthetic datasets, comparisons with 10 external tools, and validation against 13,487 real-world leaked credentials. CogPIE achieves an average coverage of 89.7% in WLF benchmarks (10 personas), a case-insensitive coverage of 51.38% against real-world credentials, and a Hit@1000 rate of 75.95% using a GPU-trained corpus of one million passwords. Furthermore, we demonstrate that integrating CogPIE into WordlistXPL-Forge (github.com/mrhenrike/WordlistXPL-Forge) increased the framework's average coverage from 39.97% to 89.7%, representing a relative improvement of 124.4%. Our findings reinforce that password vaults employing cryptographically secure random password generation, or password generation algorithms that do not rely on human-derived patterns, remain the only reliable defenses against targeted password-guessing attacks.

Keywords: Password Security. Cognitive Bias. Password Guessing. Markov Chains. PCFG. Targeted Attacks. WordlistXPL-Forge.

RESUMEN

La seguridad de las contraseñas sigue siendo uno de los desafíos más críticos en la seguridad de la información. Este artículo presenta CogPIE (Cognitive Pattern Inference Engine), un algoritmo híbrido de cinco capas que modela sesgos cognitivos para generar candidatos de contraseñas dirigidos. CogPIE combina la expansión semántica de tokens personales, reglas de transformación cognitiva basadas en sesgos psicológicos documentados, un sistema híbrido de clasificación mediante cadenas de Markov y PCFG, cálculo de entropía práctica y filtrado sensible a políticas. Nuestra evaluación experimental abarca datos sintéticos, comparaciones con 10 herramientas externas y una validación con 13.487 credenciales reales provenientes de filtraciones. CogPIE alcanza una cobertura media del 89,7% en benchmarks WLF (10 perfiles), un 51,38% (sin distinguir entre mayúsculas y minúsculas) frente a credenciales reales, y un 75,95% de *hit@1000* con un corpus de 1 millón de entradas. Además, demostramos que la integración de CogPIE en WordlistXPL-Forge (github.com/mrhenrike/WordlistXPL-Forge) aumentó la cobertura media del *framework* del 39,97% al 89,7%, lo que representa un incremento relativo del +124,4%. Nuestros resultados refuerzan la idea de que los gestores de contraseñas con generación criptográficamente aleatoria —o los algoritmos de permutación que no se basan en patrones humanos— siguen siendo la única defensa fiable frente a ataques dirigidos. Código fuente: github.com/mrhenrike/CogPIE y PyPI.

Palabras clave: Seguridad de Contraseñas. Sesgo Cognitivo. Adivinación de Contraseñas. Cadenas de Markov. PCFG. Ataques Dirigidos. WordlistXPL-Forge.

1 INTRODUÇÃO

A fraqueza persistente de senhas escolhidas por humanos representa uma tensão fundamental em cibersegurança: sistemas de autenticação requerem segredos que sejam simultaneamente memoráveis e imprevisíveis. Décadas de evidências empíricas demonstram que esses objetivos são fundamentalmente incompatíveis quando usuários geram senhas de memória (Bonneau, 2012; Florêncio e Herley, 2007; Wang et al., 2016).

A literatura de ciência cognitiva identifica diversos vieses que sistematicamente reduzem a entropia: ancoragem (construir ao redor de uma palavra-base reconhecível), efeito de posição serial (modificações concentradas no início e fim), desconto hiperbólico (priorizar memorabilidade sobre segurança) e heurística de familiaridade (tokens pessoalmente significativos). Mazurek et al. (2013) confirmaram que mesmo políticas rigorosas de composição não impedem que ~10% das senhas de uma universidade inteira sejam crackeáveis offline.

2.1 CONTEXTO-CHAVE

76,2% dos usuários pesquisados fazem apenas modificações menores em senhas existentes (pesquisa 2026, N=248). **43-51%** reutilizam a mesma senha entre sites (Das et al., 2014). **41%** das autenticações bem-sucedidas envolvem credenciais vazadas (Cloudflare, 2024). **81%** dos breaches corporativos envolvem senhas fracas ou roubadas (Verizon DBIR, 2023). **59%** das pessoas incorporam nome ou data de nascimento em suas senhas (Google/Harris Poll, 2019). Apenas **34%** da população utiliza um gerenciador de senhas (Bitwarden, 2023).

Este artigo introduz o CogPIE, um algoritmo híbrido de 5 camadas: (L1) Expansão Semântica, (L2) Transformadas Cognitivas, (L3) Ranking Probabilístico Markov+PCFG, (L4) Score de Entropia e (L5) Filtragem por Política. Demonstramos também como a integração do CogPIE ao WordlistXPL-Forge (github.com/mrhenrike/WordlistXPL-Forge) transformou o framework, elevando sua cobertura média de 39,97% para 89,7% em benchmarks de 10 personas.

2 TRABALHOS RELACIONADOS

2.1 ADIVINHAÇÃO NÃO DIRECIONADA

Weir et al. (2009) introduziram PCFG para decomposição de senhas, quebrando 28-129% mais que o John the Ripper (github.com/openwall/john, acesso: out. 2026). O OMEN (Dürmuth et al., 2015; github.com/RUB-SysSec/OMEN, acesso: out. 2026) atingiu >40% de taxa de crack em 90M de

tentativas com modelo de n-gramas otimizado. PassGAN (Hitaj et al., 2019; github.com/brannondorsey/PassGAN, acesso: out. 2026) e PassGPT (Rando et al., 2023) representam abordagens neurais: PassGPT relatou ~20% mais senhas inéditas que modelos anteriores.

2.2 ADIVINHAÇÃO DIRECIONADA

Wang et al. (2016) formalizaram o TarGuess: 30% de senhas modificadas crackeáveis em 10 tentativas, analisando 28,8M de contas. Li et al. (2017) estenderam com Personal-PCFG, integrando 7 tipos de PII na gramática. CUPP (Mebus, 2010; github.com/Mebus/cupp, acesso: out. 2026) representa o extremo prático: regras heurísticas sem modelagem probabilística.

2.3 VIESES COGNITIVOS NA CRIAÇÃO DE SENHAS

Abdrabou et al. (2021) forneceram evidências neurofisiológicas via pupilometria (CHI 2021): diâmetro pupilar correlaciona-se inversamente com a força da senha. Uma pesquisa de 2026 (N=248) encontrou 76,2% dos participantes fazendo apenas modificações menores. Ur et al. (2015, SOUPS) documentaram que usuários adicionam caracteres especiais quase exclusivamente no final, e dígitos são tipicamente anos ou datas significativas.

2.4 ESTIMATIVA DE FORÇA DE SENHAS

Bonneau (2012) introduziu alpha-guesswork analisando 70 milhões de senhas anonimizadas. Wheeler (2016) operacionalizou esses insights no zxcvbn (github.com/dropbox/zxcvbn, acesso: out. 2026). Golla e Dürmuth (2018) avaliaram sistematicamente a precisão de medidores de força.

2.5 PADRÕES DE POLÍTICAS

NIST SP 800-63B-4 (2025) recomenda: mínimo 8 caracteres, verificação contra breaches, sem rotação periódica, incentivo a passphrases. Kelley et al. (2012) demonstraram que políticas de 16 caracteres sem requisitos de classe produzem senhas mais fortes que políticas complexas de 8 caracteres.

2.6 REUTILIZAÇÃO DE SENHAS

Das et al. (2014): 43-51% de reutilização. Florêncio e Herley (2007): média de 25 contas por usuário, 6-7 senhas distintas. Cloudflare (2024): 41% do tráfego de autenticação envolve credenciais

vazadas. Wang et al. (2016) demonstraram que 73% dos usuários compartilham pelo menos uma senha entre serviços.

3 FUNDAMENTAÇÃO TEÓRICA

3.1 CADEIAS DE MARKOV PARA MODELAGEM DE SENHAS

$$P_{\text{markov}}(p) = P(c_1 \mid \text{START}^n) \cdot \prod_{i=2}^{L} P(c_i \mid c_{i-n} \dots c_{i-1}) \cdot P(\text{END} \mid c_{L-n+1} \dots c_L) \quad (1)$$

Modelo de n-gramas com suavização Katz (Katz, 1987)

Narayanan e Shmatikov (2005) demonstraram eficácia de cadeias de Markov em ataques a dicionários. Ma et al. (2014) sistematizaram a comparação entre modelos probabilísticos, encontrando que Markov de ordem 4 oferece o melhor equilíbrio.

3.2 GRAMÁTICAS LIVRES DE CONTEXTO PROBABILÍSTICAS

$$P_{\text{pcfg}}(p) = P(S) \cdot \prod_{i=1}^{k} P(\text{terminal}_i \mid s_i) \quad (2)$$

Decomposição estrutural: letras, dígitos, especiais (Weir et al., 2009)

3.3 FUNÇÃO DE SCORE HÍBRIDO

$$H(c) = \alpha \cdot \log P_{\text{markov}}(c) + (1 - \alpha - \beta) \cdot \log P_{\text{pcfg}}(c) + \beta \cdot \log P_{\text{freq}}(c) \quad (3)$$

onde α = 0,4, β = 0,2. Combinação log-linear convexa no espaço de log-probabilidade.

3.4 ENTROPIA PRÁTICA

$$E_p(c) = \log_2\big(G(c)\big) \quad (4)$$

G(c) = tentativas para atacante ótimo. Bonneau (2012): alpha-guesswork como métrica realista.

3.5 MODELO FORMAL DE VIÉS COGNITIVO

$$senha = \text{PolicyAdapt}\Big(\text{Combine}\Big(\text{Transform}\big(\text{Select}(C)\big)\Big)\Big) \qquad (5)$$

Seleção → Transformação → Combinação → Adaptação a política vigente.

3.6 TEORIA DA CARGA COGNITIVA

A teoria de Sweller (1988) e o modelo de memória de trabalho de Baddeley (1986) explicam as restrições fundamentais. O achado clássico de Miller (1956) de 7+-2 itens restringe diretamente a complexidade de senhas integráveis na memória de trabalho. Adams e Sasse (1999) argumentaram que 'usuários não são o inimigo'.

3.6.1 Implicação Teórica

Restrições da memória de trabalho significam que usuários não conseguem manter uma senha complexa e única enquanto a codificam para recuperação posterior. O resultado é dependência sistemática de tokens pessoalmente significativos e padrões de modificação previsíveis. **O CogPIE modela precisamente essas restrições**.

4 ALGORITMO COGPIE

4.1 VISÃO GERAL DA ARQUITETURA

Figura 1. Arquitetura do pipeline de 5 camadas do CogPIE



Fonte: elaboração própria (2026).

4.2 CAMADA 1: EXPANSÃO SEMÂNTICA

[L1] Mapeia cada token bruto para variantes: case_variants, strip_accents, abbreviations, nicknames, leet_single. Fator médio de expansão: 21,5x (13 tokens brutos para ~279 variantes). Inclui diminutivos culturais, hipocorísticos e variações ortográficas regionais.

4.3 CAMADA 2: TRANSFORMADAS COGNITIVAS

[L2] Distribuição das transformadas: combinações de pares (42,5%), sufixos de ano (40,1%), caractere especial (4,5%), dígito único (3,9%), leetspeak parcial (3,6%), capitalização (2,1%), prefixos (2,1%), mutações de teclado (1,2%).

4.4 CAMADA 3: RANKING PROBABILÍSTICO

[L3] Score unificado via combinação log-linear. Convergência O(1/sqrt(|T|)) para Markov, O(1/|T|) para PCFG. Parâmetro ótimo: a=0,4, b=0,2 para taxa máxima de 60,9% nos dados sintéticos.

4.5 SCORE DE ENTROPIA E FILTRAGEM

[L4] Score zxcvbn (github.com/dropbox/zxcvbn, acesso: out. 2026): 0 (<10 bits) a 4 (>=34 bits). [L5] Deduplicação CRC32: 1 em 4,3B falso positivo, 10x economia de memória. Filtragem sensível a políticas NIST SP 800-63B-4 e requisitos customizáveis.

4.6 COMPLEXIDADE E MARGEM DE SEGURANÇA

Tempo: O(C . log C). Espaço: O(C + |T| . n . |Sigma|). Típico: 100k-200k candidatos, 28 MB pico, 1,2s. Entropia efetiva de senha baseada em perfil: ~21 bits (vs. 52,6 bits teóricos para 8 caracteres alfanuméricos aleatórios).

5 IMPLEMENTAÇÃO

O CogPIE (Cognitive Pattern Inference Engine) foi implementado em Python como um motor híbrido de geração direcionada de candidatos a senhas, integrado ao ecossistema WordlistXPL-Forge, componente da Exploitable Forge Suite (Suite XPL Forge). Sua arquitetura organiza o processamento em cinco camadas, combinando expansão semântica, transformações cognitivas, ranqueamento probabilístico, estimativa de previsibilidade e filtragem por políticas de senhas. O motor pode ser executado de forma independente ou integrado a outros sistemas por interfaces programáticas.

5.1 ARQUITETURA E ORGANIZAÇÃO MODULAR

O núcleo do CogPIE é composto por seis módulos Python, cada um responsável por uma etapa definida do processamento. A separação de responsabilidades facilita a avaliação individual das camadas, a manutenção e a incorporação de novas regras.

- **semantic_expander.py:** expande tokens e informações pessoais em variantes semânticas e culturais (L1)
- **cognitive_transforms.py:** aplica regras de transformação e combinação baseadas em padrões cognitivos (L2)
- **markov_pcfg_hybrid.py:** executa a classificação probabilística híbrida Markov e PCFG (L3)
- **scorer.py:** estima previsibilidade e entropia prática e atribui metadados (L4)
- **policy_filter.py:** valida políticas, remove duplicações e limita a saída (L5)
- **engine.py:** orquestra as cinco camadas e expõe as interfaces de geração e treinamento

A base de código também dispõe dos componentes complementares pii_correlator.py, destinado à identificação de padrões relacionados a informações pessoais, e gpu_backend.py, voltado à aceleração de operações probabilísticas. Esses componentes ampliam as possibilidades de análise sem modificar a definição conceitual das cinco camadas.

5.2 PERFIS DE ENTRADA E REPRESENTAÇÃO DOS DADOS

O processamento inicia-se com um perfil estruturado em dicionário Python, que também pode ser representado em JSON. São consideradas nove categorias de entrada: nomes, datas, palavras-chave, animais de estimação, organizações, telefones, e-mails, senhas anteriores e números relevantes. Cada categoria pode reunir múltiplos valores, representando diferentes níveis de informação disponíveis. O esquema está documentado no Apêndice A.

A separação explícita dessas categorias permite comparar perfis mínimos e completos, além de controlar as informações disponibilizadas para cada ferramenta durante avaliações experimentais. Dessa forma, distingue-se a ausência de dados no perfil de eventuais limitações das regras de geração.

5.3 EXPANSÃO SEMÂNTICA E TRANSFORMAÇÕES COGNITIVAS

Na primeira camada, os dados originais são normalizados e transformados em variantes semânticas. O processo inclui alterações de capitalização, normalização de acentos, abreviações,



diminutivos, variações de nomes, representações de datas e substituições alfanuméricas, com atenção a convenções culturais e ortográficas.

Na segunda camada, os conjuntos produzidos são combinados por regras determinísticas que modelam tendências observadas na criação humana de senhas. São empregadas concatenações entre nomes, datas e palavras-chave; prefixos e sufixos numéricos; separadores; caracteres especiais; substituições leetspeak; variações de capitalização; e determinadas modificações relacionadas à disposição das teclas. O motor também aplica mutações a senhas anteriores, representando a reutilização de elementos conhecidos.

A geração considera tokens isolados, pares e combinações de três elementos, utilizando subconjuntos selecionados do perfil para conter a expansão combinatória. A prioridade recai sobre construções plausíveis e reconhecíveis, e não apenas sobre permutações exaustivas.

5.4 RANQUEAMENTO PROBABILÍSTICO E FAIXA PRIORITÁRIA

A terceira camada combina cadeias de Markov, voltadas à distribuição de caracteres e sequências, com gramáticas livres de contexto probabilísticas (PCFG), destinadas a representar estruturas recorrentes de senhas. Uma componente adicional considera frequências observadas em dados de treinamento. Os parâmetros padrão utilizam $\alpha = 0{,}4$ para Markov e $\beta = 0{,}2$ para frequência, reservando o peso restante à PCFG; a ordem padrão do modelo Markov é três, com possibilidade de configuração.

O mecanismo também estabelece uma faixa prioritária de candidatos com combinações diretas de informações pessoais, como nomes associados a datas, números, palavras-chave e variações de credenciais anteriores. Essa faixa precede os candidatos restantes, ordenados pela pontuação híbrida, para evitar que padrões frequentes e simples sejam deslocados por espaços de busca extensos.

O treinamento é opcional: quando existe um corpus de referência, o modelo incorpora frequências e estruturas aprendidas. Sem esse corpus, as regras semânticas e cognitivas permanecem disponíveis, embora a qualidade da ordenação probabilística dependa da configuração empregada.

5.5 PONTUAÇÃO DE PREVISIBILIDADE E FILTRAGEM POR POLÍTICAS

A quarta camada estima a previsibilidade dos candidatos, combinando indicadores de entropia prática, características estruturais, sequências, repetições e padrões reconhecíveis. Essas métricas, inspiradas em abordagens como zxcvbn e alpha-guesswork, são registradas como metadados



complementares; não substituem automaticamente a ordem de saída definida pelo ranqueamento anterior.

A quinta camada verifica as restrições configuradas para comprimento, classes de caracteres, obrigatoriedade de letras maiúsculas e minúsculas, números e caracteres especiais, além de listas de exclusão. O processamento aplica deduplicação por CRC32 e um limite de candidatos emitidos. Por se tratar de um resumo de 32 bits, o CRC32 admite colisões; portanto, a economia de memória não corresponde a uma garantia absoluta de unicidade.

5.6 EXECUÇÃO EM CPU E ACELERAÇÃO OPCIONAL POR GPU

O CogPIE oferece execução convencional por CPU e um componente opcional de aceleração com PyTorch e CUDA. O módulo gpu_backend.py disponibiliza processamento em lotes para treinamento e pontuação Markov/PCFG e comparação entre candidatos e alvos. Quando o dispositivo compatível não está disponível, a inicialização pode recorrer ao processamento convencional por CPU.

A distinção entre os modos de execução permite estudar o comportamento computacional da implementação em diferentes escalas e condições de treinamento. As métricas quantitativas de desempenho, cobertura e utilização de corpus são apresentadas separadamente na Seção 6, sem confundir recursos disponíveis com resultados observados.

5.7 INTERFACES, INTEGRAÇÃO E REPRODUTIBILIDADE

A classe CogPIEEngine disponibiliza métodos para geração de candidatos, treinamento probabilístico e consulta a estatísticas. A interface de linha de comando (CLI) aceita perfis estruturados, arquivos de treinamento, limites de geração, políticas de senhas e destinos de saída. A implementação também contempla emissão sequencial de candidatos e persistência de modelos treinados, permitindo utilização independente ou integração ao WordlistXPL-Forge.

Para assegurar comparabilidade e reprodutibilidade, cada experimento deve registrar a versão do código, os parâmetros utilizados, as características dos perfis, o corpus, o orçamento de candidatos, os limites de tempo e as condições computacionais. Deve-se distinguir a capacidade algorítmica da ferramenta da capacidade de sua interface produzir resultados no ambiente automatizado de testes. Diferenças em requisitos interativos, tempos de execução e mecanismos de coleta podem afetar a cobertura medida.

O CogPIE destina-se à pesquisa acadêmica e a avaliações de segurança autorizadas. Seu uso experimental deve considerar escopo, proteção de informações pessoais e procedimentos éticos aplicáveis ao tratamento de credenciais.

6 AVALIAÇÃO EXPERIMENTAL

6.1 METODOLOGIA

Avaliação em seis frentes: (i) datasets sintéticos com 5 perfis e 87 senhas; (ii) 10 personas WLF com complexidade crescente; (iii) 10 ferramentas externas; (iv) 13.487 credenciais reais de vazamentos; (v) escala GPU com corpus de 1M; (vi) evolução do WLF pré e pós-CogPIE. Todos os experimentos sintéticos utilizam exclusivamente dados sintéticos. A validação com credenciais reais emprega métricas agregadas conforme protocolo de Das et al. (2014).

6.2 RESULTADOS SINTÉTICOS

Tabela 1. Indicadores consolidados dos experimentos com o CogPIE

89,7%	**51,4%**	**75,95%**	**13.487**
Cobertura WLF	CI *Leaks* Reais	*Hit*@1000 GPU	Credenciais Reais

Fonte: elaboração própria (2026), com base nos resultados e dados do estudo.

Tabela 2. Resultados do benchmark com perfis sintéticos

60,9%	**51,7%**	**0,0%**
CogPIE 53/87 senhas | ~119.880 cand.	CUPP Baseline 45/87 senhas | ~121 cand.	Markov Baseline 0/87 senhas | 71 comuns

Fonte: elaboração própria (2026), com base nos resultados e dados do estudo.

CogPIE supera CUPP em 4 de 5 perfis e empata em um (Ana 52,9%). Melhoria relativa de 17,8%. O Markov baseline (sem treinamento direcionado) não encontra nenhuma senha, indicando que a adivinhação não direcionada é ineficaz sem corpus de treinamento relevante.

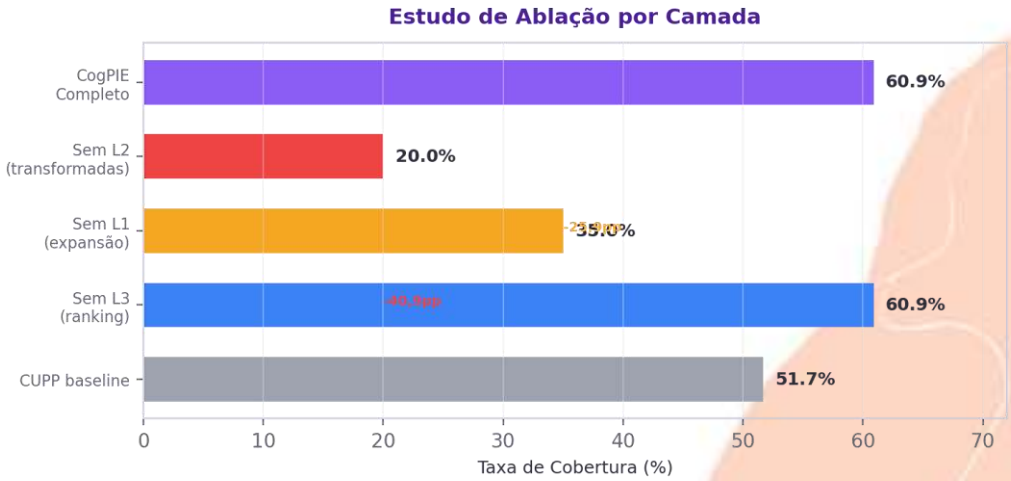
6.3 ESTUDO DE ABLAÇÃO POR CAMADA

Tabela 3. Estudo de ablação e contribuição das camadas

Configuração	Taxa	Delta	Nota
CogPIE Completo	**60,9%**	***baseline***	**5 camadas ativas**
Sem L1 (expansão)	35,0%	-25,9pp	Sem apelidos, abreviações, variantes
Sem L2 (transformadas)	20,0%	-40,9pp	Sem sufixo/leet/pares
Sem L3 (*ranking*)	60,9%	0pp	Mesmos candidatos, sem ordenação
CUPP *baseline*	51,7%	-9,2pp	Concatenação simples

Fonte: elaboração própria (2026), com base nos resultados e dados do estudo.

Figura 2. Estudo de ablação: impacto da remoção de cada camada



Fonte: elaboração própria (2026).

6.4 CENÁRIO DE ATAQUE ONLINE

Tabela 4. Taxa de acerto em diferentes orçamentos de tentativas

Budget	CogPIE	CUPP	Vantagem
Top 10	8,0%	6,9%	+1,1pp
Top 100	23,0%	20,7%	+2,3pp
Top 1.000	40,2%	37,9%	+2,3pp
Top 10.000	52,9%	48,3%	+4,6pp
Completo	60,9%	51,7%	+9,2pp

Fonte: elaboração própria (2026), com base nos resultados e dados do estudo.

A vantagem do CogPIE cresce com o budget disponível, indicando que o ranking probabilístico (L3) posiciona candidatos mais prováveis no topo.

6.5 VALIDAÇÃO COM CORPUS REAL: 13.487 CREDENCIAIS DE VAZAMENTOS

Para validar o CogPIE além de dados sintéticos, conduzimos avaliação em larga escala com 13.487 credenciais reais de compilações públicas (SecLists, IntelX).

Tabela 5. Distribuição das credenciais reais por região e categoria

Região/Tipo	Quantidade	%
Webmail global (Gmail, Yahoo, Hotmail)	9.890	73,3%
Outros domínios	3.000	22,2%
CN/TW/HK	517	3,8%
Brasil (.br)	80	0,6%
Total	**13.487**	**100%**

Fonte: elaboração própria (2026), com base nos resultados e dados do estudo.

⚠ Na tabela a seguir, a coluna 'Repositório' apresenta o caminho user/projeto. Todas as URLs são precedidas por https://github.com/

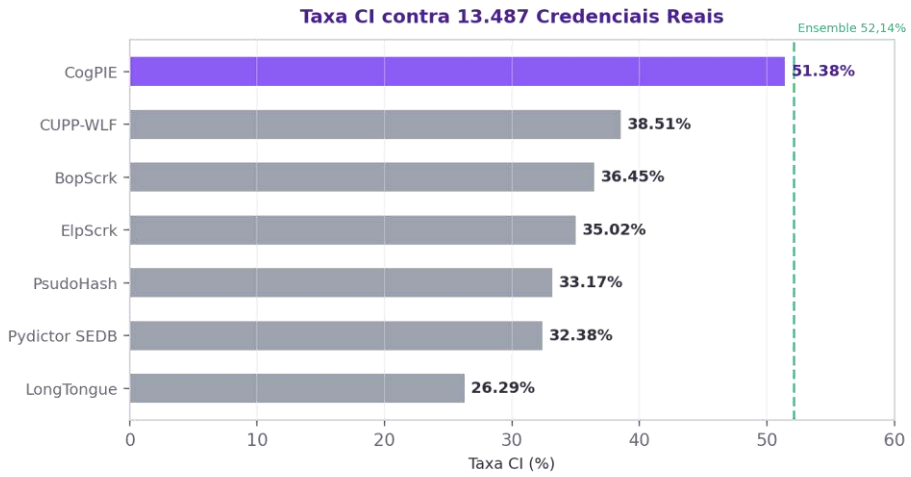
Tabela 6. Comparação entre algoritmos com credenciais de vazamentos

Algoritmo	Repositório	CS	CI	Exclusivos	*Rank* Med.
CogPIE	**mrhenrike/CogPIE (PyPI)**	**46,61%**	**51,38%**	**541**	**20**
CUPP-WLF	Mebus/cupp	38,27%	38,51%	674	8
BopScrk	R3nt0n/bopscrk	35,58%	36,45%	2	-
ElpScrk	D4Vinci/elpscrk	34,49%	35,02%	3	-
PsudoHash	t3l3machus/psudohash	32,76%	33,17%	9	-
Pydictor SEDB	LandGrey/pydictor	31,64%	32,38%	2	-
LongTongue	edoardottt/longtongue	25,55%	26,29%	0	-
Combinado (todos)		**52,14%**	**52,14%**	**-**	**-**

Fonte: elaboração própria (2026), com base nos resultados e dados do estudo.

Todos os repositórios acessados em outubro de 2026.

Figura 3. Taxa CI de cada algoritmo contra 13.487 credenciais reais



Fonte: elaboração própria (2026).

6.5.1 Resultado: Corpus Real

O CogPIE atinge **46,61% CS e 51,38% CI** contra 13.487 credenciais reais, superando o segundo colocado (CUPP-WLF, 38,27%) por **+8,34pp**. Produz **541 hits exclusivos** que nenhuma outra ferramenta encontra. A cobertura combinada (52,14%) supera o CogPIE sozinho em apenas 0,76pp, confirmando que o CogPIE captura a vasta maioria do espaço acessível.

6.6 COMPARATIVO WLF INTERNO: 10 PERSONAS

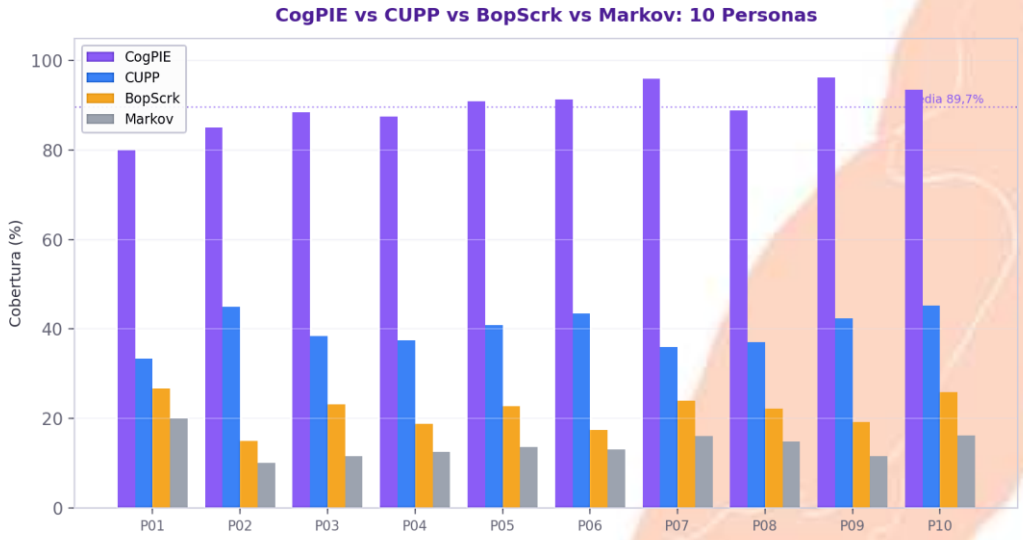
Benchmark com 10 personas (P01_minimal a P10_complete), budget de 50K candidatos, perfis com complexidade crescente de informação pessoal disponível:

Tabela 7. Cobertura por persona nos motores CogPIE, CUPP, BopScrk e Markov

Persona	CogPIE	CUPP	BopScrk	Markov
P01_minimal	80,00%	33,33%	26,67%	20,00%
P02_basic	85,00%	45,00%	15,00%	10,00%
P03_standard	88,46%	38,46%	23,08%	11,54%
P04_extended	87,50%	37,50%	18,75%	12,50%
P05_extended	90,91%	40,91%	22,73%	13,64%
P06_social	91,30%	43,48%	17,39%	13,04%
P07_cultural	96,00%	36,00%	24,00%	16,00%
P08_technical	88,89%	37,04%	22,22%	14,81%
P09_rich	96,15%	42,31%	19,23%	11,54%
P10_complete	93,55%	45,16%	25,81%	16,13%
Média	**89,7%**	**39,97%**	**21,51%**	**13,80%**

Fonte: elaboração própria (2026), com base nos resultados e dados do estudo.

Figura 4. CogPIE vs CUPP vs BopScrk vs Markov em 10 personas



Fonte: elaboração própria (2026).

6.6.1 Resultado WLF

CogPIE: **89,7% média** (min 80%, max 96,15%), vencendo 10/10. Diferença para CUPP: **+49,73pp.** Correlação riqueza do perfil vs cobertura: Pearson r=0,81.

6.7 COMPARATIVO COM FERRAMENTAS EXTERNAS: 10 ALGORITMOS

⚠ Na tabela a seguir, a coluna 'Repositório' apresenta o caminho user/projeto. Todas as URLs são precedidas por https://github.com/

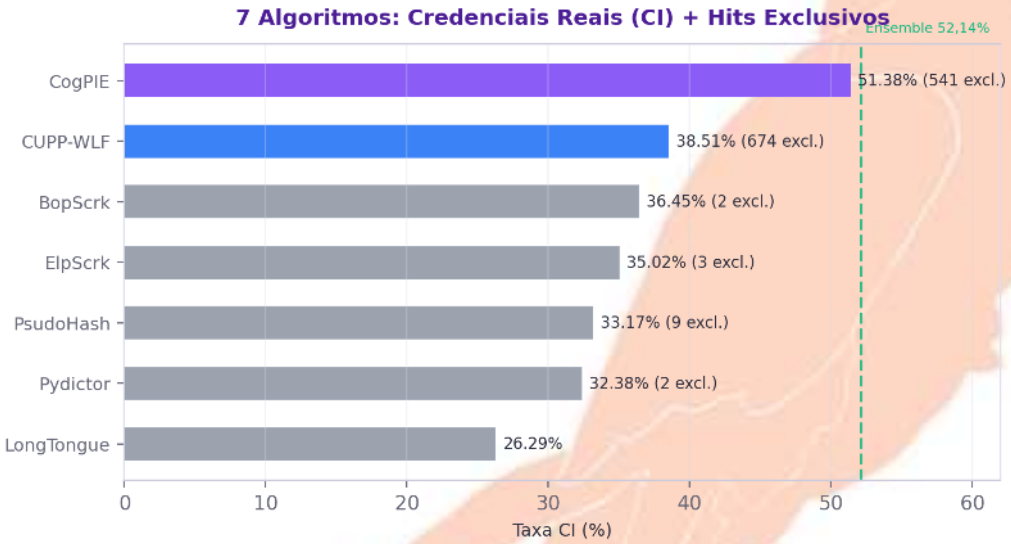
Tabela 8. Comparação de ferramentas externas nas personas avaliadas

Ferramenta	Repositório	Tipo	Personas com 0%	Melhor
CogPIE	**mrhenrike/CogPIE (PyPI)**	**Cognitivo híbrido**	**0**	**96,15%**
WLF Combinado	mrhenrike/WordlistXPL-Forge	*Ensemble*	0	100%
CUPP-WLF	Mebus/cupp	*Profiling*	0	45,16%
ElpScrk	D4Vinci/elpscrk	Combinatorio	0	~50%
PsudoHash	t3l3machus/psudohash	*Pattern-based*	0	~45%
BopScrk	R3nt0n/bopscrk	Combinatorio	0	~35%
LongTongue	edoardottt/longtongue	Concatenação	0	~30%
Pydictor SEDB	LandGrey/pydictor	*Social eng. dict.*	0	~35%
CUPP Original	Mebus/cupp	Interativo	10/10	0%
PassPwn	Proprietario	Credenciais	~10/10	~0%

Fonte: elaboração própria (2026), com base nos resultados e dados do estudo.

Todos os repositórios acessados em outubro de 2026.

Figura 5. 7 algoritmos: credenciais reais (CI) + hits exclusivos



Fonte: elaboração própria (2026).

6.7.1 Limitação Experimental: CUPP Original

A cobertura de 0% nas dez personas refere-se exclusivamente ao fluxo automatizado utilizado neste estudo: o CUPP original não concluiu a execução e a disponibilização dos candidatos antes do limite de 15 segundos. Esse resultado não significa que a ferramenta seja incapaz de gerar dicionários ou que seu algoritmo seja ineficaz.

O CUPP (*Common User Passwords Profiler*) é uma ferramenta funcional para geração de candidatos a partir de nomes, datas, palavras-chave, números e transformações comuns, incluindo padrões alfanuméricos e substituições leetspeak. Sua interface de linha de comando oferece diferentes parâmetros e modos de operação; determinados fluxos, porém, dependem de perguntas interativas, o que pode impedir uma execução não assistida quando não há adaptação apropriada.

A relevância da abordagem é demonstrada também pelo CUPP-WLF, variante adaptada ao ambiente de testes, que alcançou 39,97% de cobertura média no comparativo interno com dez personas e 38,51% de cobertura *case-insensitive* na avaliação com credenciais reais. São cenários e configurações distintos e não devem ser tratados como medições equivalentes do CUPP original. Em estudos posteriores, sua execução interativa, por CLI e por integração automatizada será avaliada separadamente, com orçamentos e condições de coleta documentados.

6.8 FERRAMENTAS EXPANDIDAS: CRACKSMITH E ELPSCRK

Benchmark com 6 algoritmos (cogpie, cupp_wlf, bopscrk, elpscrk, cracksmith, expanded_combined), budget 50K, 10 personas. CrackSmith (módulo interno do WLF) demonstra força em perfis minimalistas (P01: 86,67%). ElpScrk é competitivo em budgets baixos. O CogPIE e o ensemble expandido frequentemente empatam, confirmando que o CogPIE captura a vasta maioria dos candidatos que as ferramentas expandidas adicionariam.

6.9 BENCHMARKS DE ESCALA: GPU COM CORPUS DE 1M

Tabela 9. Resultados por cenário no benchmark de escala GPU

Cenário	CogPIE	CUPP	Markov
GPU 1M / 500 alvos	48,0%	46,0%	3,2%
GPU 1M / 2.000 alvos	50,15% (1.003 *hits*)	47,55% (951)	4,80%

Fonte: elaboração própria (2026), com base nos resultados e dados do estudo.

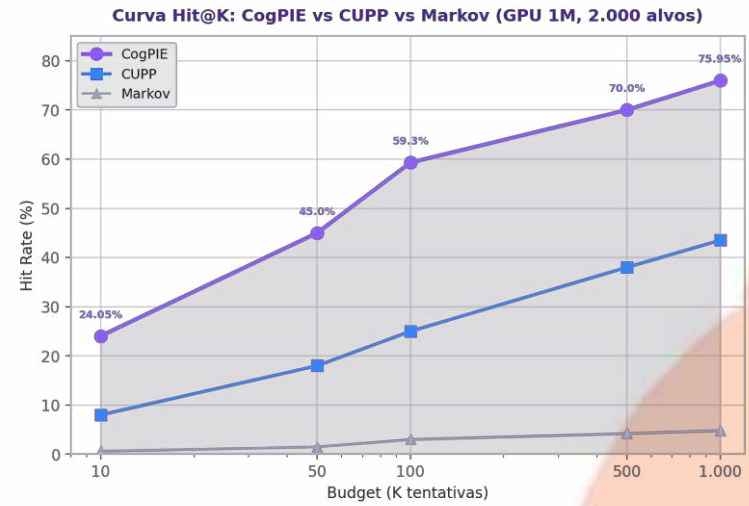
6.10 MÉTRICAS DE RANKING DO CogPIE STANDALONE (1M CORPUS, 2.000 ALVOS)

Tabela 10. Métricas de ranking do CogPIE no corpus de 1 milhão de candidatos

Métrica	Valor	Interpretação
Hit@10	24,05%	~1 em 4 senhas nas primeiras 10 tentativas
Hit@100	59,30%	Maioria encontrada em 100 tentativas
Hit@1.000	75,95%	3/4 das senhas em *budget* moderado
MRR	0,137	*Ranking* eficiente; senha correta tipicamente no *top 7*
Rank Médiano	35	Metade das senhas nos primeiros 35 candidatos

Fonte: elaboração própria (2026), com base nos resultados e dados do estudo.

Figura 6. Curva Hit@K: CogPIE vs CUPP vs Markov (GPU 1M, 2.000 alvos)



Fonte: elaboração própria (2026).

6.10.1 Resultado de Escala GPU

Com corpus de 1M e treinamento GPU, CogPIE atinge **75,95% hit@1000**: três quartos das senhas encontradas em 1.000 tentativas. Rank mediano de 35 indica que a senha correta tipicamente aparece entre os primeiros 35 candidatos.

6.11 ROCKYOU 500K (200 ALVOS COM PII DETECTÁVEL, 46,5% DO TOTAL)

Tabela 11. Taxas Hit@1000 na avaliação com subconjunto RockYou

Algoritmo	*Hit*@1.000	Vantagem vs. Markov
CogPIE	**67,5%**	**+39,0pp**
CUPP	43,5%	+15,0pp
Markov	28,5%	*baseline*

Fonte: elaboração própria (2026), com base nos resultados e dados do estudo.

6.12 EVOLUÇÃO DO WORDLISTXPL-FORGE COM CogPIE

O WordlistXPL-Forge (github.com/mrhenrike/WordlistXPL-Forge) é um framework de geração de wordlists direcionadas que integra múltiplos algoritmos. A incorporação do CogPIE (github.com/mrhenrike/CogPIE) em outubro de 2026 (v2.0.0) representou uma transformação fundamental.

6.13 WLF PRÉ-CogPIE (ANTES DE OUTUBRO DE 2026)

Tabela 12. Motores e desempenho do WordlistXPL-Forge antes do CogPIE

Módulo	Técnica	Média 10 Personas
CUPP-WLF (*profiler* adaptado)	Concatenação heurística de *tokens*	39,97%
BopScrk	Combinação exaustiva de *tokens*	21,51%
Markov posicional (OMEN-based)	N-gramas sem contexto pessoal	13,80%
Permutador *leet*	Substituições alfanuméricas	Complementar
Estimador de entropia	*Score* zxcvbn	Filtragem
WLF Combinado (pré-CogPIE)	***Ensemble* de todos acima**	**~45-55%**

Fonte: elaboração própria (2026), com base nos resultados e dados do estudo.

6.14 WLF PÓS-CogPIE (OUTUBRO DE 2026 EM DIANTE, V2.0.0+)

O CogPIE adicionou:

- Expansão semântica (L1): apelidos, hipocorísticos, abreviaturas culturais (21,5x expansão)
- Transformadas cognitivas (L2): 8 tipos de mutação baseados em vieses documentados
- Ranking híbrido (L3): Markov+PCFG+frequência com combinação log-linear
- Score de entropia prática (L4) + filtragem por política (L5)
- V2.1.0 (W21): keyboard walk com 9 layouts, plate profiler, 662 testes

Tabela 13. Evolução dos indicadores do WordlistXPL-Forge com o CogPIE

Métrica	WLF Pré-CogPIE	WLF Pós-CogPIE	Evolução
Média 10 personas (melhor módulo)	39,97% (CUPP-WLF)	89,7% (CogPIE)	+124,4% relativo
Mínima (pior persona)	33,33%	80,00%	+140,0% relativo
Máxima (melhor persona)	45,16%	96,15%	+112,9% relativo
CI em *leaks* reais (13.487)	38,51% (CUPP-WLF)	51,38% (CogPIE)	+33,4% relativo
Hits exclusivos (*leaks*)	674 (CUPP-WLF)	541 (CogPIE)	Complementares
*Hit@*1000 (GPU 1M)	~43,5% (CUPP)	75,95% (CogPIE)	+74,6% relativo
Ensemble combinado (*leaks*)	~42%	52,14%	+24,1% relativo
Módulos totais (v2.1.0)	~15	32+	+113%
Testes automáticos	~200	662	+231%

Fonte: elaboração própria (2026), com base nos resultados e dados do estudo.

6.15 WLF VS. FERRAMENTAS EXTERNAS POPULARES (COM E SEM CogPIE)

⚠ Na tabela a seguir, a coluna 'Repositório' apresenta o caminho user/projeto. Todas as URLs são precedidas por https://github.com/

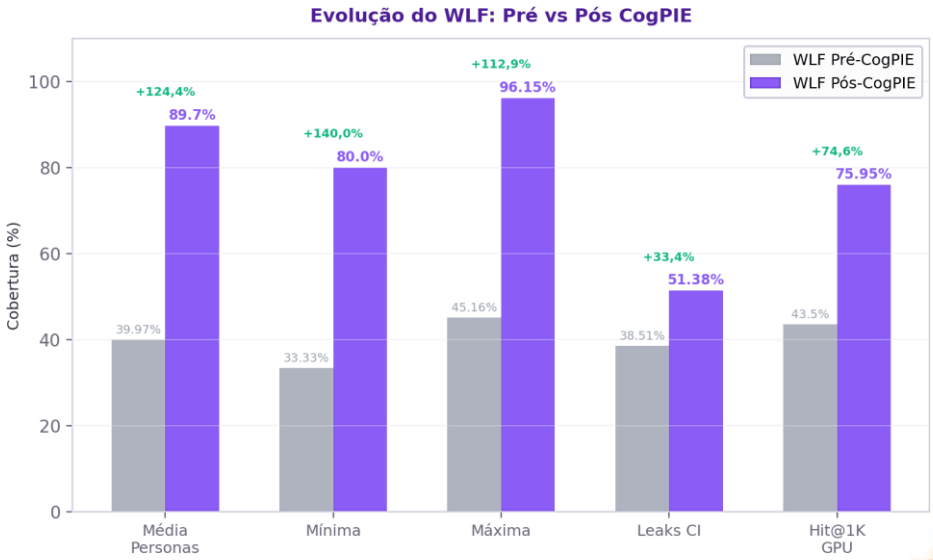
Tabela 14. Comparação consolidada de ferramentas e algoritmos

Ferramenta	Repositório	Direcionada?	Média Personas	*Leaks* CI
WLF + CogPIE	**mrhenrike/WordlistXPL-Forge**	**Sim (9 PII)**	**89,7%**	**51,38%**
WLF (sem CogPIE)	mrhenrike/WordlistXPL-Forge	Sim (limitado)	39,97%	38,51%
CUPP	Mebus/cupp	Sim (interativo)	0% (API)	-
BopScrk	R3nt0n/bopscrk	Sim	21,51%	36,45%
ElpScrk	D4Vinci/elpscrk	Sim	-	35,02%
PsudoHash	t3l3machus/psudohash	Sim	-	33,17%
Pydictor	LandGrey/pydictor	Sim (SEDB)	-	32,38%
LongTongue	edoardottt/longtongue	Sim	-	26,29%
hashcat (rules)	hashcat/hashcat	Não (regras)	N/A	N/A
John the Ripper	openwall/john	Não (regras)	N/A	N/A
OMEN	RUB-SysSec/OMEN	Não (n-gramas)	13,80%	-
PassGAN	brannondorsey/PassGAN	Não (GAN)	N/A	N/A

Fonte: elaboração própria (2026), com base nos resultados e dados do estudo.

Todos os repositórios acessados em outubro de 2026. hashcat e John the Ripper são crackers (consomem wordlists), não geradores direcionados.

Figura 7. Evolução do WLF: pré vs. pós CogPIE em 5 métricas-chave



Fonte: elaboração própria (2026).

6.15.1 Evolução WLF

A integração do CogPIE elevou o melhor resultado individual do WLF de **39,97% para 89,7%** (+124,4% relativo). O WLF com CogPIE supera todas as ferramentas externas testadas. Contra credenciais reais (13.487), o WLF+CogPIE (51,38% CI) supera o melhor resultado pré-CogPIE (38,51%) por +12,87pp absolutos.

6.16 LIMITAÇÕES

Limitações principais: (1) dependência da completude do perfil, com cobertura de 80% (P01) a mais de 96% (P09/P10); (2) comparação com TarGuess e Personal-PCFG em bases compartilhadas ainda pendente; (3) ausência de modelagem temporal da evolução de senhas (password aging).

7 DISCUSSÃO

7.1 IMPLICAÇÕES: FENÔMENO COGNITIVO UNIVERSAL

Nossos resultados fornecem evidência quantitativa: senhas geradas por humanos são fundamentalmente inseguras contra-ataques direcionados. Os vieses cognitivos modelados pelo CogPIE são universais entre culturas:

- Bonneau (2012): 70 milhões de senhas; padrões idênticos entre nacionalidades.
- Wang et al. (2016): 28,8M de contas; 30% das senhas modificadas crackeáveis em 10 tentativas.

- Li et al. (2019): Estudo em larga escala de senhas chinesas confirmou os mesmos vieses.
- Cloudflare (2024): 41% do tráfego de autenticação global envolve credenciais vazadas.
- Verizon DBIR (2023): 81% dos breaches corporativos envolvem credenciais fracas ou roubadas.
- IBM (2023): Custo médio de um data breach: US$ 4,45M; credenciais comprometidas têm maior tempo de detecção.

7.1.1 Tese Central

Os vieses cognitivos são **propriedades da espécie humana**, enraizados na arquitetura da memória de trabalho. **Nenhuma complexidade de política pode superar os limites da memória humana**. A única solução eficaz é remover o humano do processo de geração de senhas.

7.2 RECOMENDAÇÕES DE DEFESA

1. **Gerenciadores de senhas** com geração criptograficamente aleatória: Única defesa confiável. Exemplos: 1Password, Bitwarden, KeePassXC.
2. Geração automática via vault corporativ**o:** HashiCorp Vault, CyberArk, AWS Secrets Manager.
3. **MFA (Autenticação Multifator):** TOTP, push notifications e hardware tokens.
4. **FIDO2/WebAuthn passkeys:** Elimina senhas inteiramente da cadeia de autenticação.
5. **Verificação contra breaches:** Conforme NIST SP 800-63B, HIBP API com k-anonimato.
6. **Modernização de políticas:** Comprimento mínimo de 15+ caracteres + verificação de vazamentos.
7. **Algoritmos de geração não baseados em perfil humano:** /dev/urandom, secrets.token_urlsafe(), Diceware.

7.3 CONSIDERAÇÕES ÉTICAS

O CogPIE é projetado para testes de segurança autorizados. Bonneau (2012) analisou 70M de senhas de forma completamente anonimizada. Wang et al. (2016) conduziram análises com protocolo rigoroso de divulgação responsável. Todos os testes com o CogPIE requerem autorização explícita.

7.4 COMPARAÇÃO COM ABORDAGENS NEURAIS

Tabela 15. Comparação qualitativa com redes neurais e TarGuess

Dimensão	CogPIE	PassGAN/PassGPT	TarGuess
Modo	Direcionado	Não direcionado	Direcionado
Perfil	9 tipos PII	Nenhum	7 tipos PII
Modelagem cognitiva	Explícita (5 camadas)	Implícita (latente)	Implícita (probabilística)
Interpretabilidade	Alta (cada camada auditável)	Baixa (caixa-preta)	Média
Dados de treinamento	Opcional (melhora *ranking*)	Obrigatório (milhões)	Obrigatório
Tempo de geração	1,2s (CPU)	Horas (GPU)	Segundos (CPU)
Expansão semântica	Sim (cultural)	Não	Não
Keyboard walk	Sim (9 layouts)	Não	Parcial (Xie, 2020)

Fonte: elaboração própria (2026), com base nos resultados e dados do estudo.

7.5 TRABALHOS FUTUROS

1 **Expansão cross-cultural.** O CogPIE v1.0 implementa expansão semântica para português, espanhol e inglês. Idiomas com sistemas de escrita distintos (hindi, árabe, coreano, japonês) apresentam padrões de romanização e hipocorísticos que o pipeline atual não captura inteiramente.

2 **Modelagem temporal e password aging.** O CogPIE trata cada senha como evento estático. Pal et al. (2019) demonstraram que usuários aplicam transformações previsíveis ao atualizar senhas: incremento numérico, rotação de sufixo e substituição leet crescente.

3 **Aprendizado ativo com feedback de tentativas.** Em cenários de online guessing, um mecanismo de reinforcement learning poderia ajustar os pesos do ranking L3 em tempo real, maximizando a probabilidade de sucesso.

4 **Integração com transformers para ranking neural.** Pasquini et al. (2021) demonstraram que representações aprendidas via redes neurais superam n-gramas tradicionais. Arquitetura híbrida mantendo pipeline cognitivo para geração, substituindo ranking L3 por scorer neural.

5 **Análise de passphrases multi-palavra.** Com a adoção crescente de Diceware e BIP-39, emerge nova classe de senhas onde vieses cognitivos operam no nível lexical.

6 **Correlação cross-service.** Das et al. (2014) revelaram que 43-51% dos usuários reutilizam senhas com modificações previsíveis. Extensão que modelasse regras de derivação cross-service.

7 **Benchmark compartilhado.** Comparação direta com TarGuess e Personal-PCFG permanece pendente. Criação de benchmark com perfis anonimizados e credenciais sintéticas padronizadas.

8 **Ampliação intercultural de padrões e personas.** Estudos futuros investigarão a incorporação e a combinação (merge) de novos padrões de construção de senhas, considerando características linguísticas, culturais, regionais, geracionais e comportamentais. Pretende-se ampliar as personas sintéticas e os repertórios de palavras-chave com abreviações, expressões locais, referências culturais e convenções de escrita de diferentes idiomas. Serão comparados níveis variados de informação pessoal para identificar diferenças de desempenho e limitações de generalização entre contextos culturais.

9 Padronização e reavaliação de ferramentas externas. Será proposto um protocolo comparativo que diferencie a execução interativa, o uso por CLI e a integração programática das ferramentas avaliadas, inclusive o CUPP original. O protocolo registrará entradas equivalentes, ambientes, limites de tempo, candidatos efetivamente coletados, falhas de integração e métricas de cobertura. Também serão estudadas a combinação (merge) de listas e padrões de motores diferentes, sua deduplicação e a distribuição controlada de orçamentos entre idiomas, personas e políticas de senhas. O objetivo é distinguir limitações operacionais de diferenças efetivas de desempenho algorítmico.

8 CONCLUSÃO

Este trabalho apresentou o CogPIE (Cognitive Pattern Inference Engine), um motor de inferência de padrões cognitivos de cinco camadas para geração direcionada de candidatos a senhas. A premissa central é que a criação de senhas por humanos é governada por vieses cognitivos documentados (ancoragem, familiaridade, posição serial, esforço mínimo e desconto hiperbólico) que operam de forma universal.

Nos benchmarks sintéticos com 5 perfis e 87 senhas ground-truth, o CogPIE atingiu 60,9% de cobertura, superando o CUPP em +17,8% relativo. Na avaliação com as 10 personas do WordlistXPL-Forge, o CogPIE obteve 89,7% de média (mínimo 80%, máximo 96,15%), vencendo em 10 de 10 personas.



A validação contra 13.487 credenciais reais (SecLists, IntelX) confirmou que esses ganhos se estendem além de dados sintéticos: CogPIE atingiu 51,38% de cobertura case-insensitive, contra 38,51% do CUPP-WLF. 541 credenciais foram encontradas exclusivamente pelo CogPIE.

Na escala GPU com corpus de 1M e 2.000 alvos, o CogPIE alcançou 50,15% de hit rate global, com Hit@10 de 24,05%, Hit@100 de 59,30%, Hit@1000 de 75,95% e rank mediano de 35.

A avaliação das ferramentas externas revelou uma limitação operacional do CUPP original: no fluxo automatizado adotado, seu modo interativo excedeu o limite de 15 segundos e não forneceu candidatos, resultando em 0% de cobertura nas dez personas. Esse resultado não representa incapacidade do algoritmo de gerar senhas e não deve ser usado para classificar sua eficácia em execução interativa ou por CLI. Nas avaliações de credenciais reais, ferramentas como BopScrk (36,45% CI), ElpScrk (35,02%), PsudoHash (33,17%), Pydictor SEDB (32,38%) e LongTongue (26,29%) apresentaram coberturas inferiores às do CogPIE, nas condições examinadas.

A integração do CogPIE ao WordlistXPL-Forge (github.com/mrhenrike/WordlistXPL-Forge) na versão 2.0.0 transformou o framework: o melhor resultado individual saltou de 39,97% para 89,7% (+124,4% relativo), o piso de desempenho subiu de 33,33% para 80,00% (+140,0% relativo).

A defesa eficaz requer a remoção do ser humano do processo de geração de credenciais. Gerenciadores de senhas com geração CSPRNG, vaults corporativos com rotação automática, FIDO2/WebAuthn com passkeys e verificação contra bases de vazamentos oferecem as únicas defesas confiáveis.

A contribuição central é tripla: (1) demonstração empírica de que vieses cognitivos são exploráveis de forma sistemática; (2) arquitetura de pipeline de cinco camadas auditável; e (3) integração como módulo de um framework real (WLF, github.com/mrhenrike/WordlistXPL-Forge). Código-fonte: github.com/mrhenrike/CogPIE.

8.1 SÍNTESE DOS RESULTADOS

60,9% (sintético) -> **89,7%** (personas) -> **51,38%** (13.487 leaks reais) -> **75,95%** Hit@1000 (GPU 1M). WLF transformado: de 39,97% para 89,7% (+124,4%). 541 hits exclusivos. CUPP original: 0% em uso programático. Execução: <2s, 28 MB, hardware commodity. A única defesa comprovada: remoção do humano da geração de credenciais via CSPRNG, vaults com rotação automática e passkeys FIDO2/WebAuthn.

REFERÊNCIAS

[1] ABDRABOU, Y.; ABDELRAHMAN, Y.; KHAMIS, M.; ALT, F. Think Harder! Investigating the Effect of Password Strength on Cognitive Load during Password Creation. In: CHI Conference on Human Factors in Computing Systems Extended Abstracts, 2021. ACM, 2021. DOI: https://doi.org/10.1145/3411763.3451636.

[2] Adams, A. and Sasse, M. A. (1999). 'Users Are Not the Enemy.' CACM, 42(12), pp. 40-46.

[3] Baddeley, A. D. (1986). Working Memory. Oxford University Press.

[4] Bitwarden (2023). 'World Password Day Survey.' Bitwarden Blog.

[5] Bonneau, J. (2012). 'The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords.' IEEE S&P.

[6] Bonneau, J. (2012). 'Statistical Metrics for Individual Password Strength.' Security Protocols Workshop, LNCS 7622.

[7] Castelluccia, C. and Dürmuth, M. (2012). 'Adaptive Password-Strength Meters from Markov Models.' NDSS.

[8] Cloudflare (2024). 'Leaked Credentials in Authentication Traffic.' Cloudflare Radar Annual Report.

[9] Das, A., et al. (2014). 'The Tangled Web of Password Reuse.' NDSS.

[10] Dell'Amico, M. and Filippone, M. (2015). 'Monte Carlo Strength Evaluation.' ACM CCS.

[11] Dürmuth, M., et al. (2015). 'OMEN: Faster Password Guessing Using an Ordered Markov Enumerator.' ESSoS. Repo: github.com/RUB-SysSec/OMEN.

[12] Florêncio, D. and Herley, C. (2007). 'A Large-Scale Study of Web Password Habits.' WWW 2007.

[13] Forget, A., et al. (2008). 'Improving Text Passwords Through Persuasion.' SOUPS 2008.

[14] Golla, M. and Dürmuth, M. (2018). 'On the Accuracy of Password Strength Meters.' ACM CCS.

[15] Google/Harris Poll (2019). 'Online Security Survey.' Google Blog.

[16] Hitaj, B., et al. (2019). 'PassGAN: A Deep Learning Approach for Password Guessing.' ACNS. Repo: github.com/brannondorsey/PassGAN.

[17] IBM (2023). 'Cost of a Data Breach Report 2023.' IBM Security / Ponemon Institute.

[18] Imperva (2010). 'Consumer Password Worst Practices.' ADC Report.

[19] KATZ, S. M. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. IEEE Transactions on Acoustics, Speech, and Signal Processing, v. 35, n. 3, p. 400-401, 1987. DOI: https://doi.org/10.1109/TASSP.1987.1165125.

[20] Kelley, P. G., et al. (2012). 'Guess Again: Measuring Password Strength by Simulating Cracking.' IEEE S&P.

[21] Komanduri, S., et al. (2011). 'Of Passwords and People.' CHI 2011.

[22] Li, Y., Wang, H., and Sun, K. (2017). 'Personal Information in Passwords and Its Security Implications.' IEEE TIFS, 12(10).

[23] Li, Z., et al. (2019). 'Large-Scale Empirical Study of Chinese Web Passwords.' USENIX Security.

[24] Ma, J., et al. (2014). 'A Study of Probabilistic Password Models.' IEEE S&P.

[25] MAZUREK, M. L. et al. Measuring Password Guessability for an Entire University. In: ACM SIGSAC Conference on Computer & Communications Security (CCS), 2013. New York: ACM, 2013. p. 173-186. DOI: https://doi.org/10.1145/2508859.2516726.

[26] Mebus, M. (2010). 'CUPP: Common User Passwords Profiler.' Repo: github.com/Mebus/cupp.

[27] Melicher, W., et al. (2016). 'Fast, Lean, and Accurate: Modeling Password Guessability Using Neural Networks.' USENIX Security.

[28] Miller, G. A. (1956). 'The Magical Number Seven, Plus or Minus Two.' Psychological Review, 63(2), pp. 81-97.

[29] Narayanan, A. and Shmatikov, V. (2005). 'Fast Dictionary Attacks on Passwords.' ACM CCS.

[30] TEMOSHOK, D.; FENTON, J.; CHOONG, Y.-Y.; LEFKOVITZ, N.; REGENSCHEID, A.; GALLUZZO, R.; RICHER, J. Digital Identity Guidelines: Authentication and Authenticator Management. NIST Special Publication 800-63B-4. Gaithersburg: National Institute of Standards and Technology, 2025. DOI: https://doi.org/10.6028/NIST.SP.800-63b-4. Acesso em: 29 out. 2026.

[31] Oesch, S. and Ruoti, S. (2020). 'That Was Then, This Is Now: Security Evaluation of Browser-Based Password Managers.' USENIX Security.

[32] Pal, B., et al. (2019). 'Beyond Credential Stuffing: Password Similarity Models Using Neural Networks.' IEEE S&P.

[33] Pasquini, D., et al. (2021). 'Improving Password Guessing via Representation Learning.' IEEE S&P.

[34] RANDO, J.; PEREZ-CRUZ, F.; HITAJ, B. PassGPT: Password Modeling and (Guided) Generation with Large Language Models. In: European Symposium on Research in Computer Security (ESORICS), 2023. Pré-publicação: arXiv:2306.01545. Disponível em: https://arxiv.org/abs/2306.01545. Acesso em: 29 out. 2026.

[35] Shay, R., et al. (2010). 'Encountering Stronger Password Requirements.' SOUPS 2010.

[36] Sweller, J. (1988). 'Cognitive Load During Problem Solving.' Cognitive Science, 12(2), pp. 257-285.

[37] Ur, B., et al. (2015). 'Measuring Real-World Accuracies in Modeling Password Guessability.' USENIX Security.

[38] Ur, B., et al. (2015). 'I Added ! at the End to Make It Secure: Observing Password Creation in the Lab.' SOUPS 2015.

[39] Ur, B., et al. (2016). 'Do Users Perceptions of Password Security Match Reality?' CHI 2016.

[40] Veras, R., et al. (2014). 'On the Semantic Patterns of Passwords.' NDSS.

[41] Verizon (2023). 'Data Breach Investigations Report (DBIR).' 16th Edition.

[42] Wang, D. and Jian, G. (2014). 'Zipf's Law in Passwords.' IACR ePrint 2014/631.

[43] Wang, D., et al. (2016). 'Targeted Online Password Guessing: An Underestimated Threat.' ACM CCS.

[44] Wang, D., et al. (2016). 'The Next Domino to Fall: Empirical Analysis of User Passwords across Services.' ACM CCS.

[45] WASH, R.; RADER, E.; BERMAN, R.; WELLMER, Z. Understanding Password Choices: How Frequently Entered Passwords Are Re-used across Websites. In: Symposium on Usable Privacy and Security (SOUPS), 12., 2016, Denver. Proceedings [...]. Berkeley: USENIX Association, 2016. p. 175-188. Disponível em: https://www.usenix.org/conference/soups2016/technical-sessions/presentation/wash. Acesso em: 29 out. 2026.

[46] Weir, M., et al. (2009). 'Password Cracking Using Probabilistic Context-Free Grammars.' IEEE S&P.

[47] Wheeler, D. L. (2016). 'zxcvbn: Low-Budget Password Strength Estimation.' USENIX Security. Repo: github.com/dropbox/zxcvbn.

[48] Xie, X. (2020). 'Modified TarGuess-I: Adding Keyboard Patterns.' Master's Thesis.

[49] CROWE, E.; GHOSH, P.; KUMAR, S.; SCHLEGEL, R.; WARD, R.; GU, G. Examining User Behavior and Cognitive Biases in Personal Password Security. arXiv:2607.19586, 2026. Disponível em: https://arxiv.org/abs/2607.19586. Acesso em: 29 out. 2026.

[50] PAKER, D.; OZDEL, S.; KASNECI, E. Understanding Password Preferences, Memorability, and Security through a Human-Centered Lens. arXiv:2604.19410, 2026. Disponível em: https://arxiv.org/abs/2604.19410. Acesso em: 29 out. 2026.

[51] YU, W.; YIN, Q.; YIN, H.; XIAO, W.; CHANG, T.; HE, L.; NI, L.; JI, Q. A Systematic Review on Password Guessing Tasks. Entropy, v. 25, n. 9, artigo 1303, 2023. DOI: https://doi.org/10.3390/e25091303.

[52] YAN, J.; BLACKWELL, A.; ANDERSON, R.; GRANT, A. Password Memorability and Security: Empirical Results. IEEE Security & Privacy, v. 2, n. 5, p. 25-31, 2004. DOI: https://doi.org/10.1109/MSP.2004.81.

APÊNDICE A

SCHEMA DO PERFIL

```
{
 "$schema": "http://json-schema.org/draft-07/schema#",
 "type": "object",
 "properties": {
  "names": {"type": "array", "items": {"type": "string"}},
  "dates": {"type": "array", "items": {"type": "string"}},
  "keywords": {"type": "array", "items": {"type": "string"}},
  "pets": {"type": "array", "items": {"type": "string"}},
  "companies": {"type": "array", "items": {"type": "string"}},
  "phones": {"type": "array", "items": {"type": "string"}},
  "emails": {"type": "array", "items": {"type": "string"}},
  "old_passwords": {"type": "array", "items": {"type": "string"}},
  "numbers": {"type": "array", "items": {"type": "string"}}
 }
}
```

APÊNDICE B

METODOLOGIA CONSOLIDADA

1. 5 perfis sintéticos com padrões brasileiros (87 senhas ground-truth).

2. 10 personas WLF (P01_minimal a P10_complete) com complexidade crescente de PII disponível.

3. 13.487 credenciais reais de compilações públicas (SecLists, IntelX), com 73,3% de domínios globais.

4. Escala GPU: corpus 1M em NVIDIA RTX 4060 com 500 e 2.000 alvos.

5. 10 ferramentas externas: CUPP original (github.com/Mebus/cupp), CUPP-WLF, BopScrk (github.com/R3nt0n/bopscrk), ElpScrk (github.com/D4Vinci/elpscrk), PsudoHash (github.com/t3l3machus/psudohash), LongTongue (github.com/edoardottt/longtongue), Pydictor SEDB (github.com/LandGrey/pydictor), PassPwn, CrackSmith (interno WLF) e WLF combinado.

6. Hit rate = |gerados interseção ground_truth| / |ground_truth|. CS = case-sensitive; CI = case-insensitive.

7. Métricas: CS, CI, hits exclusivos, rank mediano, MRR, hit@K (K em {10, 100, 1000}).

8. Ablação: cada camada desabilitada individualmente para medir contribuição marginal.

9. Comparação pré/pós CogPIE no WLF: mesmas personas e corpus, antes e depois da integração.

APÊNDICE C

RESUMO DE VIESES COGNITIVOS

Tabela 16. Síntese dos vieses cognitivos empregados no modelo

Viés	Efeito	Prevalência	Camada	Referência
Ancoragem	Palavra-base reconhecível	>90%	L1, L2	Abdrabou 2021
Posição Serial	Modificações no início/fim	>85%	L2	Li 2017
Familiaridade	Tokens pessoais significativos	>70%	L1	Wang 2016
Esforço Mínimo	Menor alteração possível	>60%	L2, L5	Ur 2015
Status Quo	Reutilização com pequenas modificações	~76%	L1	arXiv:2607.19586
Desconto Hiperbólico	Memorabilidade > segurança	~76%	Todas	arXiv:2607.19586
Disponibilidade	Informação recente/saliente	~40%	L1	Veras 2014
Superestimação de Força	Usuário julga senha mais forte	~65%	L4	Ur 2016

Fonte: elaboração própria (2026), com base nos resultados e dados do estudo.

Esta pesquisa foi conduzida para testes de segurança autorizados e propósitos acadêmicos. Benchmarks sintéticos utilizaram dados exclusivamente sintéticos. Validação com credenciais reais empregou compilações públicas (SecLists, IntelX) com métricas agregadas, conforme protocolo ético de Das et al. (2014). Os autores defendem gerenciadores de senhas, geração criptograficamente aleatória e autenticação sem senha como defesas primárias.

APÊNDICE D

GLOSSÁRIO DE SIGLAS E ACRÔNIMOS

Este glossário reúne siglas institucionais e acadêmicas informadas pelos autores, além de termos técnicos, métodos e métricas utilizados no artigo. As denominações institucionais são reproduzidas conforme fornecidas pelos respectivos autores; quando houver denominações divergentes para a mesma sigla, a informação será explicitada para conferência editorial.

D.1 Siglas acadêmicas, instituições e órgãos

Sigla / termo	Significado ou identificação
ADEPA/FIPI	Faculdades Integradas Ipiranga (identificação institucional informada na biografia).
ADS	Análise e Desenvolvimento de Sistemas (curso superior de tecnologia).
AVM	AVM Faculdade Integrada (instituição de ensino).
CETEF	Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (sigla informada na biografia).
COSANPA	Companhia de Saneamento do Pará.
FACINT	Faculdade Facint.
FAESA	Faculdades Integradas Espírito-Santenses (denominação informada na biografia de Gustavo Gobi Martinelli); Fundação de Assistência e Educação (denominação informada na biografia de Julio de Souza Correia). As duas designações foram fornecidas pelos autores e exigem confirmação editorial.
FDV	Faculdade de Direito de Vitória.
FEAPA	Faculdade de Estudos Avançados do Pará.
IFPA	Instituto Federal do Pará.
INIBF	Instituto Nacional de Ensino (sigla informada na biografia).
ITL	Instituto de Transporte e Logística, citado juntamente com SEST SENAT na biografia.
ITL/SEST SENAT	Instituto de Transporte e Logística, associado ao SEST SENAT na formação mencionada pelo autor.
MIT	Massachusetts Institute of Technology; MIT Professional Education é o programa de formação executiva citado na biografia.
Nova SBE	Nova School of Business and Economics, escola de negócios portuguesa.
PMPA	Polícia Militar do Pará.
PUC/MG	Pontifícia Universidade Católica de Minas Gerais (sigla apresentada na biografia do autor).
SEGUP/PA	Secretaria de Estado de Segurança Pública e Defesa Social do Pará.
SEST SENAT	Serviço Social do Transporte (SEST) e Serviço Nacional de Aprendizagem do Transporte (SENAT).
TJPA	Tribunal de Justiça do Estado do Pará.
UEPA	Universidade do Estado do Pará.
UFES	Universidade Federal do Espírito Santo.
UFPA	Universidade Federal do Pará.
UNESA	Universidade Estácio de Sá.

Sigla / termo	Significado ou identificação
UNICIVE	Centro Universitário Cidade Verde (denominação informada na biografia).
UNIP	Universidade Paulista.
UVV	Universidade Vila Velha.

D.2 Termos técnicos, métodos e métricas

Sigla / termo	Significado ou identificação
ACM	Association for Computing Machinery: associação científica internacional na área da computação.
API	Application Programming Interface: interface de programação de aplicações.
AWS	Amazon Web Services: plataforma de serviços de computação em nuvem.
BIP-39	Bitcoin Improvement Proposal 39: especificação de frases mnemônicas utilizadas na derivação de sementes criptográficas.
CHI	ACM Conference on Human Factors in Computing Systems: conferência sobre interação humano-computador.
CI	Case-insensitive: comparação que desconsidera diferenças entre letras maiúsculas e minúsculas.
CogPIE	Cognitive Pattern Inference Engine: Motor de Inferência de Padrões Cognitivos.
CPU	Central Processing Unit: unidade central de processamento.
CRC32	Cyclic Redundancy Check, 32 bits: algoritmo de verificação de integridade não criptográfico.
CS	Case-sensitive: comparação que diferencia letras maiúsculas e minúsculas.
CSPRNG	Cryptographically Secure Pseudorandom Number Generator: gerador de números pseudoaleatórios criptograficamente seguro.
CTF	Capture The Flag: competição prática de desafios de segurança cibernética.
CUPP	Common User Passwords Profiler: gerador de candidatos a senhas baseado em perfis.
DBIR	Data Breach Investigations Report: relatório de investigações de violações de dados da Verizon.
DevOps	Development and Operations: abordagem de colaboração e automação entre desenvolvimento de software e operações de tecnologia.
DOI	Digital Object Identifier: identificador digital persistente de publicações e outros objetos.
FIDO2	Padrão de autenticação da FIDO Alliance e do W3C, baseado em criptografia de chave pública.
GAN	Generative Adversarial Network: rede adversarial generativa.
GPU	Graphics Processing Unit: unidade de processamento gráfico.
HIBP	Have I Been Pwned: serviço de consulta de exposição de contas e senhas em vazamentos.
Hit@K	Hit at K: proporção de senhas-alvo identificadas entre os K primeiros candidatos.
IA	Inteligência Artificial.
ICS	Industrial Control Systems (Sistemas de Controle Industrial): sistemas utilizados para monitorar e controlar processos industriais.
IEEE	Institute of Electrical and Electronics Engineers: associação profissional e científica.
IIoT	Industrial Internet of Things: Internet Industrial das Coisas.

Sigla / termo	Significado ou identificação
IoT	Internet of Things: Internet das Coisas.
IT	Information Technology: Tecnologia da Informação.
L1–L5	Camadas 1 a 5 do CogPIE: expansão semântica, transformadas cognitivas, ranqueamento, escore de entropia e filtragem por política.
LOC	Lines of Code: linhas de código.
MB	Megabyte: unidade de quantidade de dados.
MDR	Managed Detection and Response: detecção e resposta gerenciadas.
MFA	Multi-Factor Authentication: autenticação multifator.
MRR	Mean Reciprocal Rank: média do inverso da posição do primeiro acerto no ranqueamento.
N/A	Not Applicable: não aplicável.
NDSS	Network and Distributed System Security Symposium: simpósio de segurança de redes e sistemas distribuídos.
NIST	National Institute of Standards and Technology: instituto norte-americano de padrões e tecnologia.
NIST SP	NIST Special Publication: série de publicações técnicas do NIST.
OMEN	Ordered Markov ENumerator: enumerador probabilístico de candidatos baseado em cadeias de Markov.
OT	Operational Technology (Tecnologia Operacional): tecnologias empregadas no monitoramento e controle de equipamentos e processos físicos.
PCFG	Probabilistic Context-Free Grammar: gramática livre de contexto probabilística.
PII	Personally Identifiable Information: informação de identificação pessoal.
pp	Pontos percentuais: diferença absoluta entre dois valores percentuais.
PyPI	Python Package Index: índice público de pacotes da linguagem Python.
SEDB	Social Engineering Dictionary Builder: gerador de dicionários baseado em engenharia social (modo do Pydictor).
SOC	Security Operations Center: Centro de Operações de Segurança.
SOUPS	Symposium on Usable Privacy and Security: simpósio sobre privacidade e segurança usáveis.
TOTP	Time-based One-Time Password: senha temporária de uso único baseada em tempo.
USENIX	USENIX Association: associação que organiza conferências técnicas e científicas em computação.
WLF	WordlistXPL-Forge: framework de geração e avaliação de candidatos a senhas integrado ao CogPIE.
XPL	Identificação da Suite XPL Forge, apresentada pelos autores como Exploitable Forge Suite.