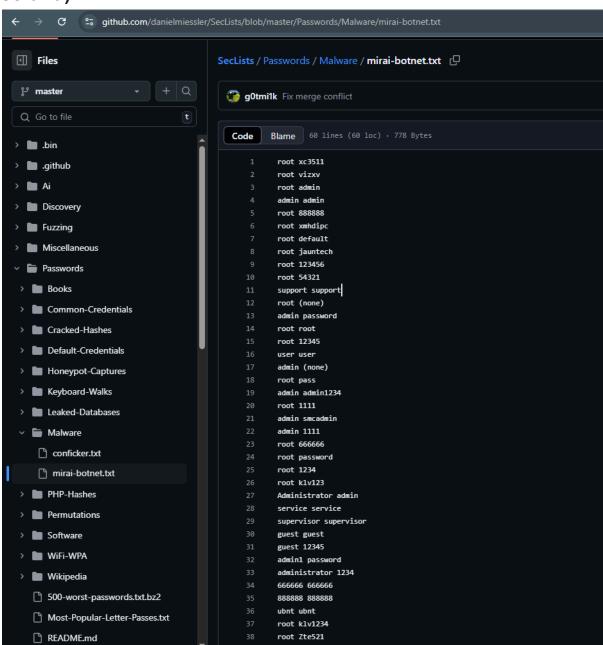
- 4. Neste exercício iremos estudar o impacto do uso de funções de hash sem salt para o armazenamento de senhas.
- a) Considere a lista de usuários e senhas usados pelo código malicioso Mirai para fazer força bruta em dispositivos IoT (https://github.com/danielmiessler/SecLists/blob/master/Passwords/Malware/mirai-botnet.txt). Escolha 10 senhas (segunda coluna).



54321
tech
1234
guest
xmhdipc
meinsm
hi3518
support
7te521

b) Use o OpenSSL ou qualquer ferramenta online para calcular o MD5 e o SHA-1 de cada uma das senhas. Dicas: use https://www.md5hashgenerator.com ou o seguinte comando: "echo -n "SENHA" | openssl dgst -md5". Verifique se o valor de hash gerado pelo OpenSSL realmente é o valor correto.

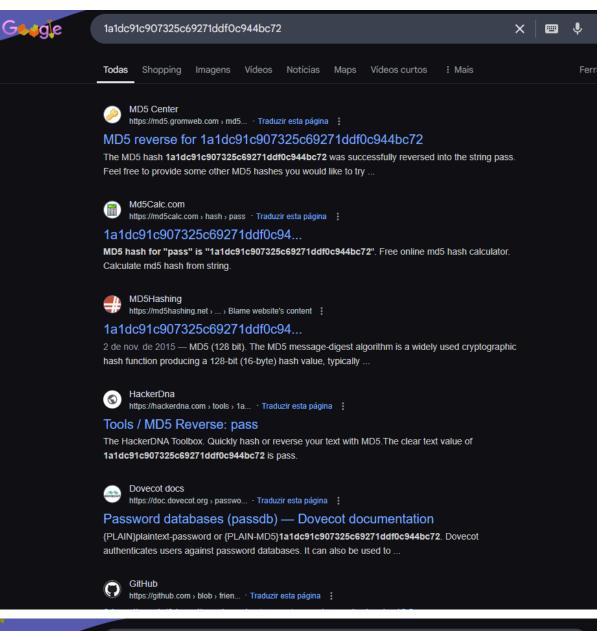
Your String	pass
MD5 Hash	1a1dc91c907325c69271ddf0c944bc72
SHA1 Hash	9d4e1e23bd5b727046a9e3b4b7db57bd8d6ee684

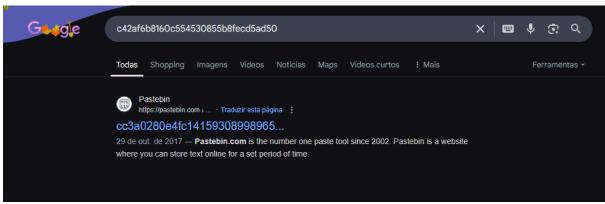
Your String	54321
MD5 Hash	c42af6b8160c554530855b8fecd5ad50
SHA1 Hash	0c3fd7f50369b5fdd7aac22f5f6ed03e713560b7
	'
Your String	tech
MD5 Hash	d9f9133fb120cd6096870bc2b496805b
SHA1 Hash	c95ee47689a0aaec70c3eb950244657722c69b1f
	•
Your String	1234
MD5 Hash	81dc9bdb52d04dc20036dbd8313ed055
SHA1 Hash	7110eda4d09e062aa5e4a390b0a572ac0d2c0220
'	
Your String	guest

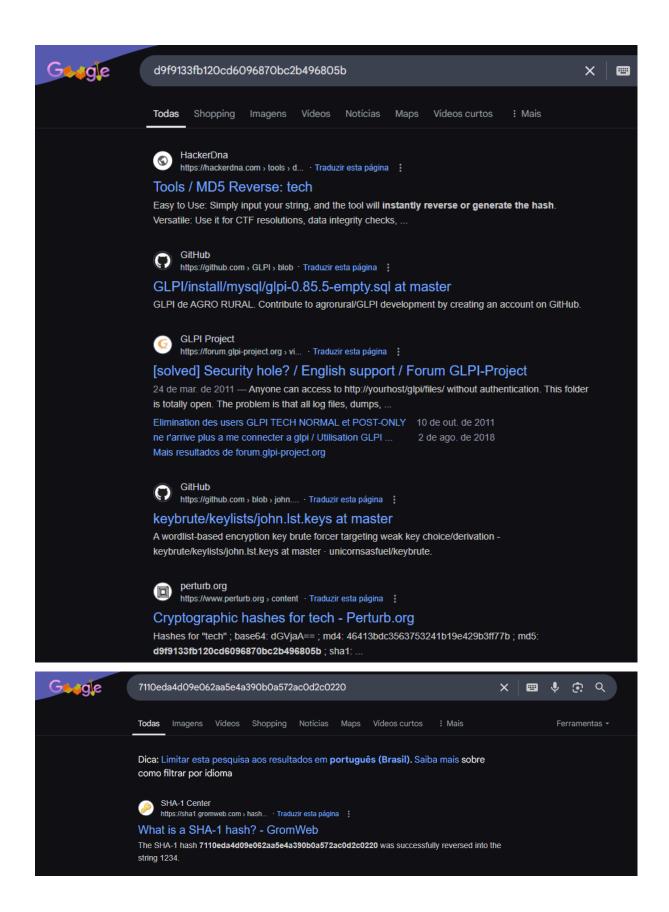
MD5 Hash	084e0343a0486ff05530df6c705c8bb4
SHA1 Hash	35675e68f4b5af7b995d9205ad0fc43842f16450
	'
Your String	xmhdipc
MD5 Hash	726927827f6c51a8995b49185ca8bc8e
SHA1 Hash	068767b389590dbb57dd3c5b7bc66971afcdf17d
Your String	meinsm
MD5 Hash	f985a3b18830f3566cdd29c242b53561
SHA1 Hash	ce7987ef8225cc1a2699a07901f59de7f09bdfed
	1
Your String	hi3518
MD5 Hash	bdfde4de898d679633fd1cf16af79242

SHA1 Hash	180a4def3cfd00b29905c73a437ced125211f233
	'
Your String	support
MD5 Hash	434990c8a25d2be94863561ae98bd682
SHA1 Hash	5bdcd3c0d4d24ae3e71b3b452a024c6324c7e4bb
Your String	Zte521
MD5 Hash	0ec3772a8336ecddd6b6c61f01c05244
SHA1 Hash	8df2ae0668218b74822da6e7de4d18b678230bd6

c) Faça buscas no Google usando como parâmetro o valor de hash obtido. A busca retorna resultados? Se sim, discuta os resultados.







Ao buscar os valores de hash no Google, vários deles apareceram com resultados, e em muitos casos dava até pra ver a senha original em sites de

consulta de hash. Isso mostra como usar só MD5 ou SHA-1, sem nenhum tipo de proteção como um salt, deixa as senhas bem vulneráveis. Mesmo sem ver a senha direto, qualquer pessoa consegue usar esses bancos de dados públicos pra descobrir o que tá por trás do hash. Por isso é importante usar algoritmos mais seguros como bcrypt ou Argon2, que já incluem mecanismos melhores para proteger a senha.

Mas não adianta só colocar salt, a senha também precisa ser forte. Isso quer dizer que ela tem que ter letras maiúsculas e minúsculas, números, símbolos e um tamanho mínimo. Senha simples tipo "1234" ou "admin" não adianta nada. E o salt também precisa ser bem feito, se for um número sequencial ou algo previsível, perde totalmente o sentido de proteção. Então o ideal é sempre combinar uma senha forte com um bom algoritmo e um salt aleatório e único pra cada usuário.

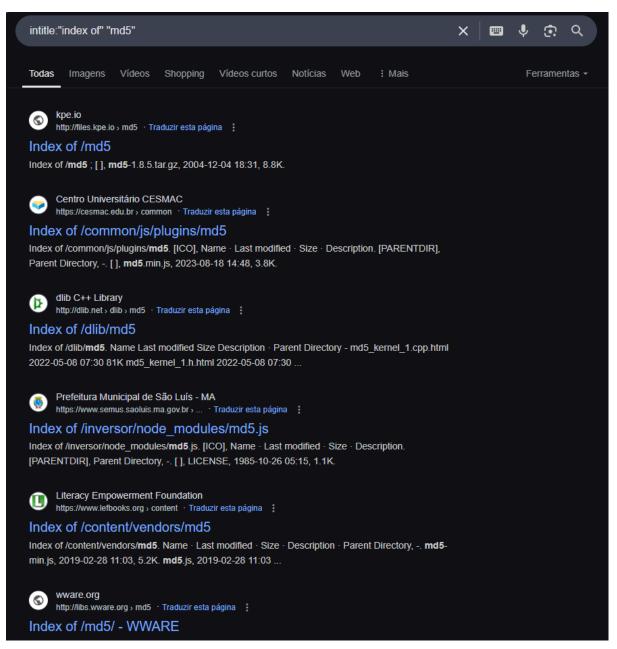
d) Tente melhorar/filtrar os resultados obtidos usando como exemplo diretivas/dorks

(https://www.avg.com/pt/signal/google-dorks) de busca do Google presentes na Google Hacking Database (https://www.exploit-db.com/google-hacking-database). Por exemplo, uma diretiva interessante seria buscar por hashes somente em arquivos .txt. Uma outra diretiva seria buscar por hashes em um determinado domínio e assim por diante. Use a sua criatividade aqui!

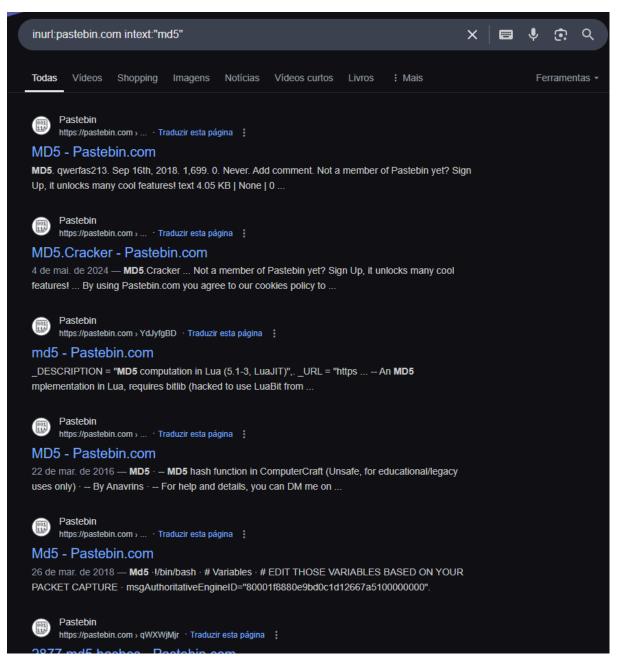
Exemplo de busca por hashes MD5 em arquivos .txt

filetype:txt intext:"1a1dc91c907325c69271ddf0c944bc72" Todas Shopping Imagens Vídeos Notícias Maps Vídeos curtos : Mais Dovecot docs https://doc.dovecot.org > _sources · Traduzir esta página : virtual users.rst.txt ... 1a1dc91c907325c69271ddf0c944bc72". UNIX UIDs ------ The most important thing you need to understand is that **Dovecot doesn't access the users' mails as ... GitHub https://github.com > blob > master - Traduzir esta página passwdMd5.txt - vieyang/md5F2F ... 1A1DC91C907325C69271DDF0C944BC72 0196F6C4F97DF3F48D570C23E46501AE 8A88344A8786439FEF8C7BDD49E69E31 413C8E4AC31A519DDBF6F0EC8048848A ... GitHub https://github.com > blob > master - Traduzir esta página php-malware-analysis/webshells/wso in depth/passwords 1a1dc91c907325c69271ddf0c944bc72 pass 6f238b53828dd00ada713972805cf1a3 pass0elpasso 43ae7930b90b7e345aad5267cc2f94a9 pass=nhzgrf ... University of Notre Dame https://www3.nd.edu > static > h... - Traduzir esta página hashes.txt ... 1a1dc91c907325c69271ddf0c944bc72 99c8ef576f385bc322564d5694df6fc2 327401d00875b45314447b206550b2b7 c0cdd82ce092b01267bdd88a8bfbb1f4 ... Packet Storm http://www2.packetstormsecurity.org > ... · Traduzir esta página jobberbase20subscribe-sql.txt You can use this script to verify if YOUR OWN instance is vulnerable. \$ bash verify.sh http://localhost/jobberbase/ admin:1a1dc91c907325c69271ddf0c944bc72 ... perlwatch.net https://perlwatch.net > VirtualUsers - Traduzir esta página

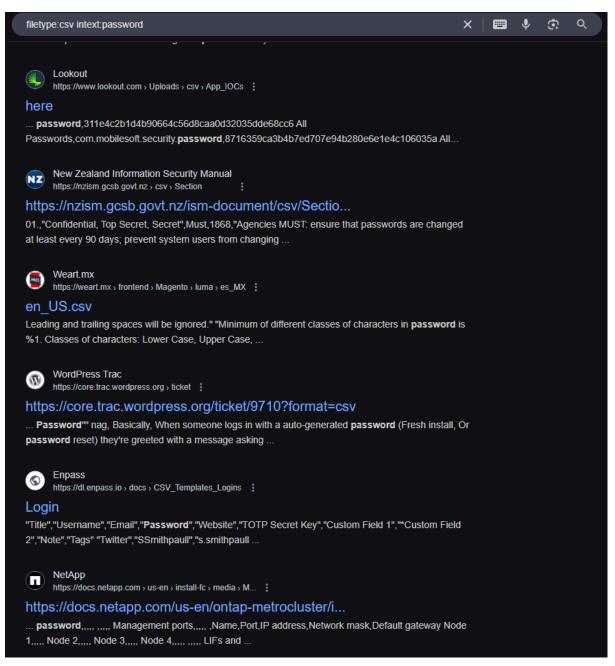
Buscar por hashes em pastas expostas



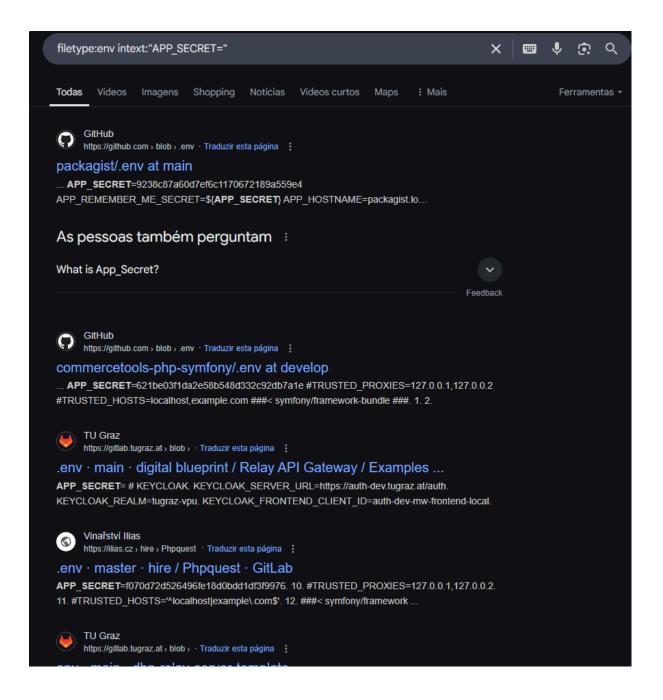
Procurar por senhas em sites que listam dumps



Dork de busca por listas de credenciais em CSV



Buscar por arquivos de config com senhas expostas



Usando dorks do Google, é possível refinar bastante as buscas por informações sensíveis como senhas e hashes. Por exemplo, conseguimos procurar diretamente por arquivos .txt ou .csv contendo hashes MD5 ou SHA-1, ou até restringir a busca para sites específicos. Isso mostra como dados mal protegidos podem ser facilmente localizados por qualquer pessoa com acesso à internet, sem nem precisar de ferramentas avançadas. Por isso, além de proteger as senhas com algoritmos seguros e salt, é essencial evitar expor arquivos sensíveis publicamente na web.