safepass.me

The smartest password filter

Documentation

A software solution by MATTA (https://www.trustmatta.com)

pwncheck version 1.0.32

Contents

safepass.me

The smartest password filter

About pwncheck

pwncheck is a self-contained, portable password audit tool for Active Directory. It enables quick, offline and effortless reporting on the prevalence of weak user passwords within your organisation.

pwncheck will flag weak passwords that have been reused across environments and/ or appear in online breached databases. It leverages a superset of a corpus of over 932M known to be compromised passwords (HIBP as of March 2024 - https://haveibeenpwned.com/Passwords).

Unlike other password audit tools, its strength lies in its speed (no need for GPUs), simplicity (no risky or cumbersome hash extraction process) and straightforward set-up (portable installation - completely offline and self-contained, no external API calls!); pwncheck can be deployed rapidly and there is no need to spend hours configuring it to obtain readily actionable intelligence.

Before you start

Overview

At safepass.me, we know your time is precious so we have worked hard to ensure that the user experience is as streamlined as possible; you will get results from pwncheck within seconds.

It’s as simple as extracting the provided ZIP file and running the portable executable!

Requirements

Choice of the computer you will run the software from

pwncheck will fetch confidential and sensitive data from your Active Directory domain (including password hashes!) using highly privileged credentials. While it attempts to do so in the most secure manner (using a kerberos authenticated secure channel), will attempt to sanitise its own memory upon termination and won't store any of the sensitive data to disk, thoughts should be put as to where you want the process to take place from.

We recommend that you run pwncheck either from a PAW (Priviliged Access Workstation - https://docs.microsoft.com/en-us/windows-server/identity/securing-privileged-access/privileged-access-workstations) or a Domain Controller.

Supported Windows Versions / Software dependencies

pwncheck requires .NET framework 4.7.1 (or later) to be installed in order to function. You can obtain the latest version of the framework at:

- https://dotnet.microsoft.com/download/dotnet-framework

Disk Space

pwncheck takes up minimal disk space (less than 3GiB on disk!), plus the size of any custom wordlist file you might want to add. The ZIP file containing the program is deceptively small: the actual database required by the software will be automatically downloaded on the first run.

Memory

pwncheck has a very small memory footprint of less than 3GiB of virtual memory, which sets it apart from the competition.

Network

pwncheck will require access to a domain controller (in all cases) and may require limited access to internet resources (see dedicated section below about Internet Connectivity) to provide additional features.

Installation

Download the software

The current version of pwncheck can be downloaded from our CDN using the following link:

- https://pwncheck.me/download

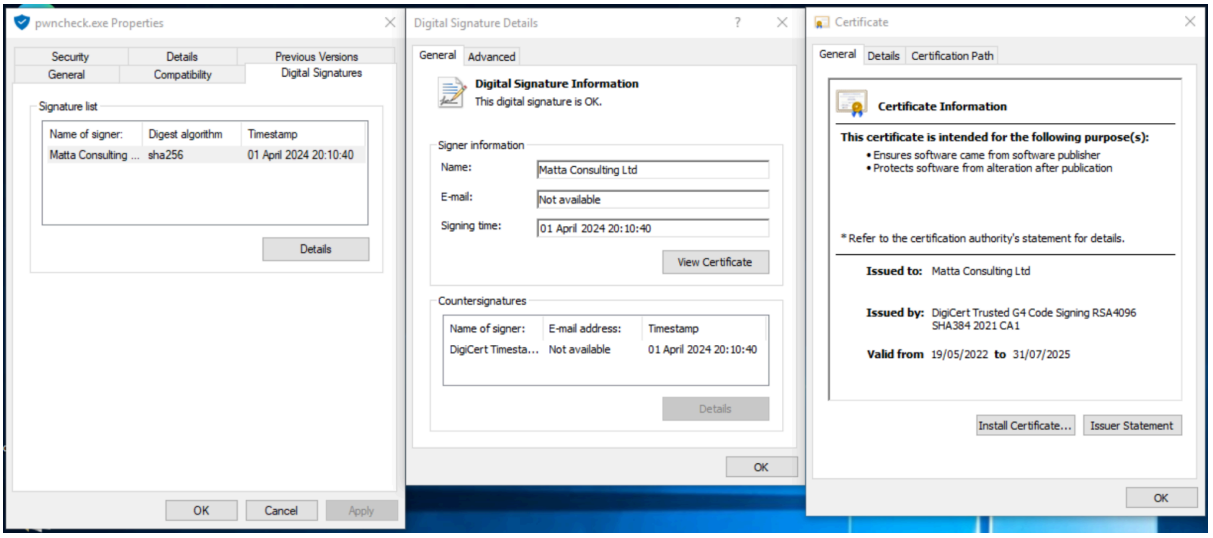
Extract the ZIP file

The software can be extracted to any suitable, writable path.

Verify its digital signature

pwncheck is distributed as a Windows Executable (EXE). The binary and all of its dependencies (DLLs) are signed by **Matta Consulting Ltd**, a leading UK-registered, boutique security consultancy outfit established in 2001 that is in effect the legal structure behind safepass.me.

In order to verify the integrity and authenticity of your safepass.me build, you should validate the authenticode signature of the MSI installer. You can do so easily by right clicking on the MSI file you have just downloaded and selecting *Properties -> Digital Signatures*



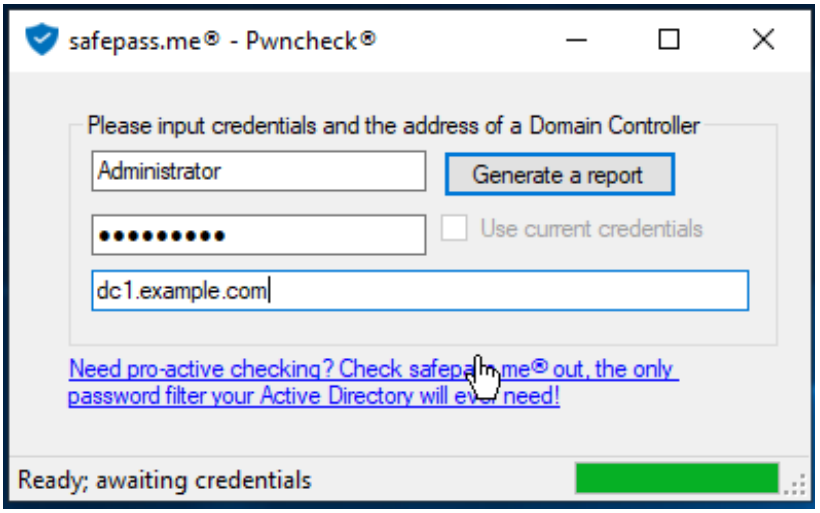
Should the signature verification fail, please report the error to **support@safepass.me** and **DO NOT RUN THE SOFTWARE !**

Common cause for signature verification failures include:

- Inaccurate time on the host
- Missing or Untrusted Certificate Authority
- Corrupted, mangled or otherwise tampered with download
- Missing windows patches that may impact the availability of cryptographic primitives used for the authenticode validation

Run the software

The software is now ready to function!



Advanced Configuration

pwncheck does not require any configuration per se: we have made it extremely straightforward to deploy and use out of the box.

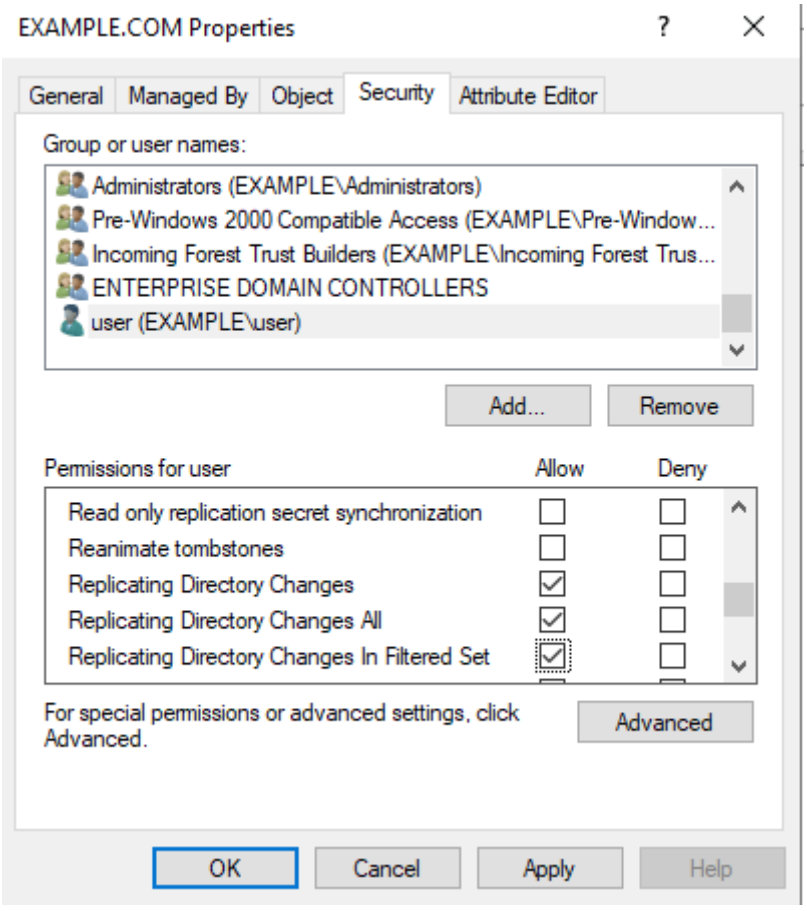
There are some advanced options you may want to use however, and are documented below.

Minimum privileges required

The user account used by pwncheck requires three specific permissions in order to be usable:

- Replicating Directory Changes
- Replicating Directory Changes All
- Replicating Directory Changes in Filtered Set

These can be granted using the GUI as follows:



Or alternatively, from PowerShell using a script such as:

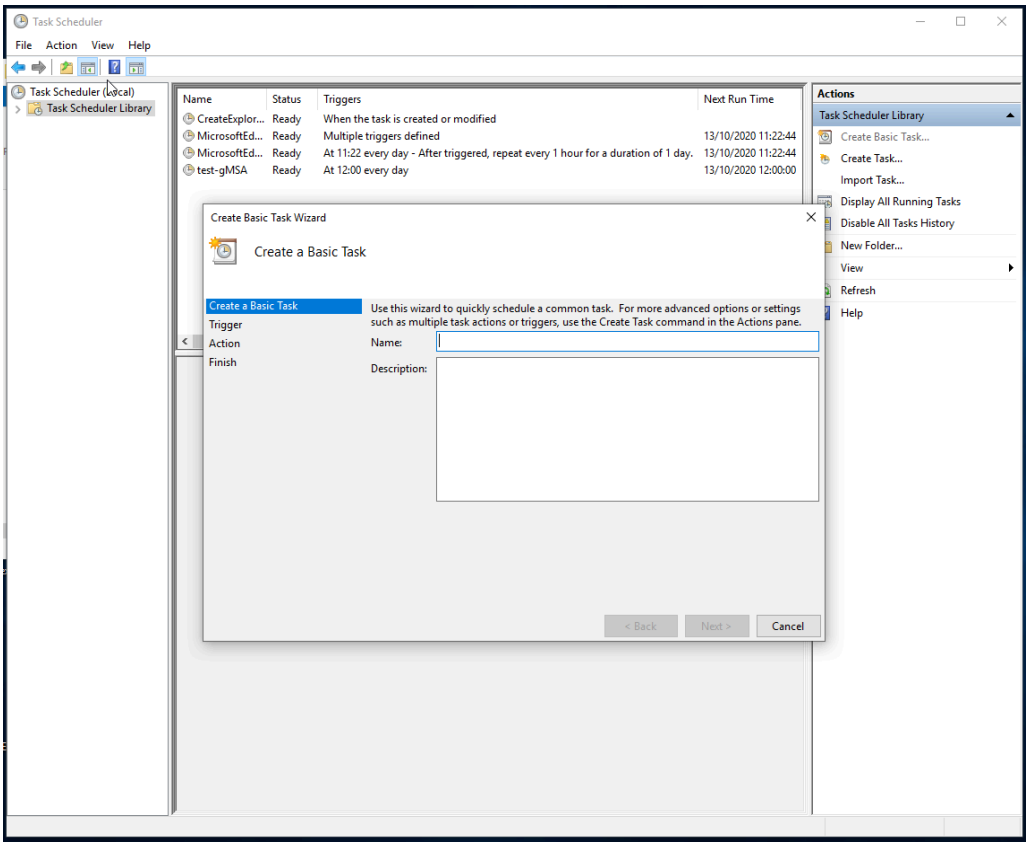
```
$Account = "pwncheck"
$RootDSE = [ADSI]"LDAP://RootDSE"
$DefaultNamingContext = $RootDse.defaultNamingContext

$cmd = "dsacls '$DefaultNamingContext' /G '`"$Account`":CA;`"Replicating Directory Changes`";'"
Invoke-Expression $cmd
$cmd = "dsacls '$DefaultNamingContext' /G '`"$Account`":CA;`"Replicating Directory Changes All`";'"
Invoke-Expression $cmd
$cmd = "dsacls '$DefaultNamingContext' /G '`"$Account`":CA;`"Replicating Directory Changes In Filtered Set`";'"
Invoke-Expression $cmd
```

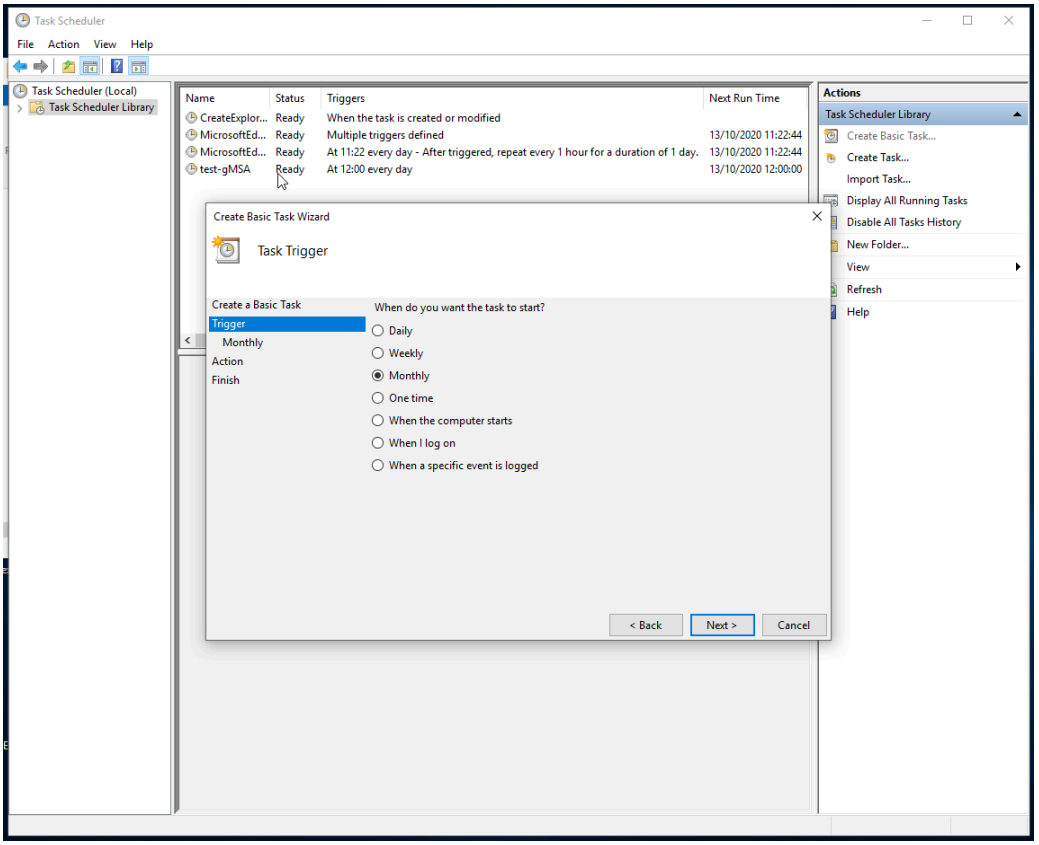
Running pwncheck non-interactively (as a scheduled task)

pwncheck can be run non-interactively as a scheduled task. It expects one single parameter as an argument (the FQDN to target: either the FQDN of a DC or a domain name). The tool's output will be logged to the pwncheck.log file in the pwncheck folder. Pwncheck will return with a non-zero exit code if it fails.

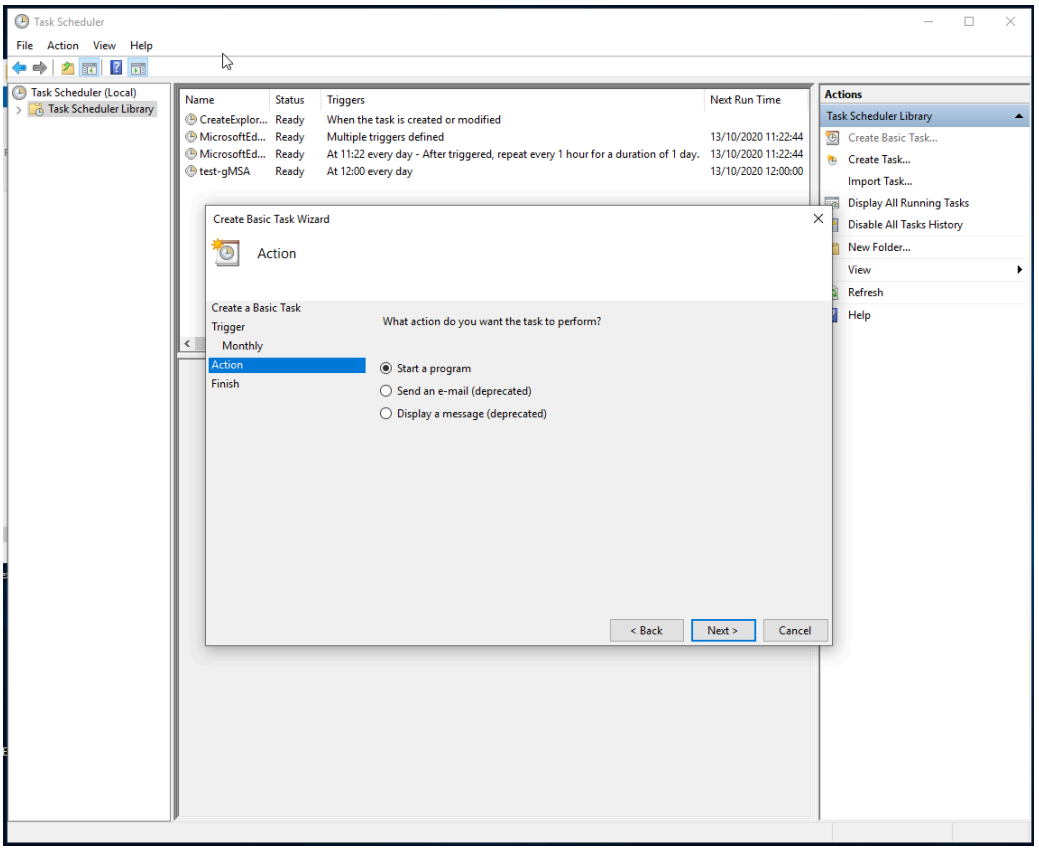
The scheduled task will have to run using a privileged account (see previous section). We recommend that you setup a dedicated account for that purpose (possibly a gMSA - https://docs.microsoft.com/en-us/windows-server/security/group-managed-service-accounts/group-managed-service-accounts-overview to ensure that its credentials are rotated automatically). The scheduled task itself can be setup easily using the GUI wizard as follows:



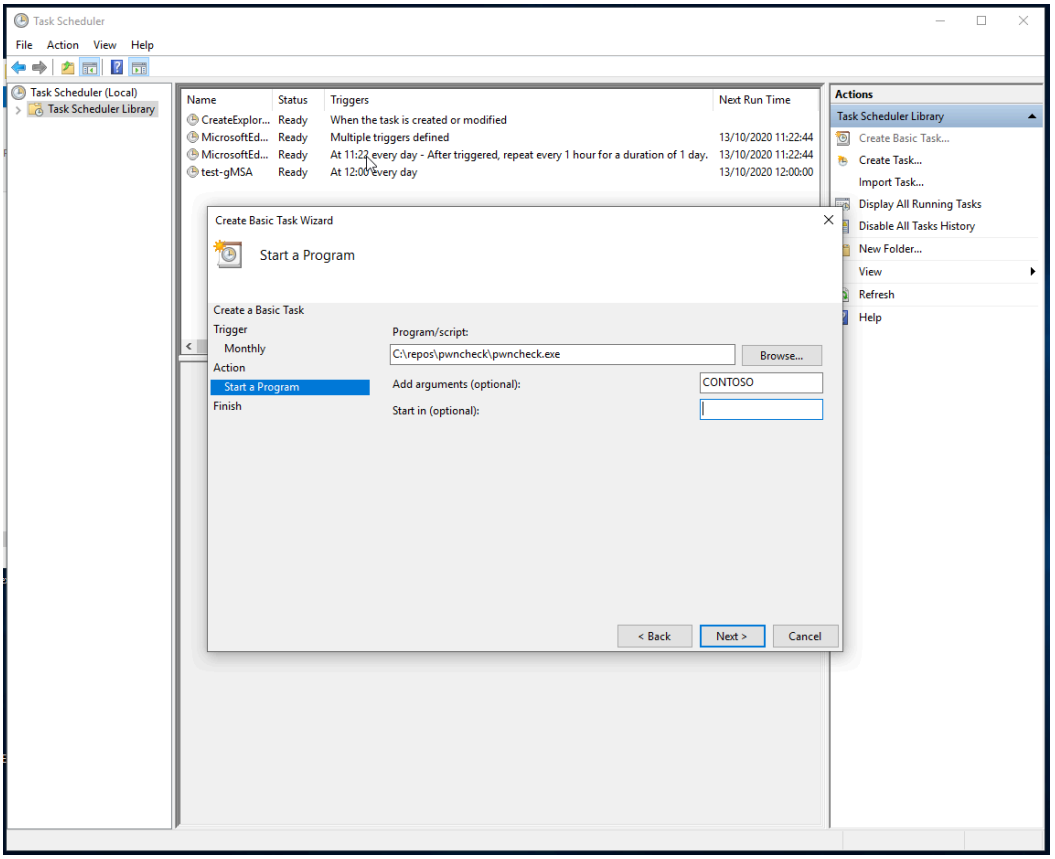
Choose a periodicity:



Select “start a program”:



Specify the path to “pwncheck.exe” and the FQDN of the domain controllers you’d like to perform the audit from (or alternatively the domain name).



That’s it! You should see a new report being created under the reports directory when the scheduled task runs.

Upgrading pwncheck and its database non-interactively

You can upgrade pwncheck non interactively by passing the ‘--upgrade’ command line argument to the tool. Similarly you can upgrade the database by using ‘--download’.

If the database is current and the tool is up to date both will return an exit code of zero.

Changing the location of the reports directory

pwncheck will create a new directory (under the reports directory) for each new report. You may change the reports directory location by using a symlink or a junction (mklink) as follows:

```
rmdir \path\to\pwncheck\reports
mklink /d \new\path\ \path\to\pwncheck\reports
```

Sharing the report folder with a different set of users that would have less privileges is of course possible.

Adding additional dictionaries

pwncheck can check for additional words from custom dictionaries that you may want to provide (threat intel feed, custom requirement, …).

There are three different ways of adding a new dictionary:

- drag & drop it onto the interface (or pass a .txt file as the first argument of pwncheck non-interactively). Using this method the data will be pre-processed into our optimised format and you could use very large dictionaries.
- in the ‘additional’ folder, any *.txt file will be scanned (runtime may be affected by very large files: you may want to pre-process them instead)
- in the ‘additional’ folder, any *.ntlm file will be scanned for hex-represented NTLM hashes

Any dictionary that is found will be used for subsequent scans. If you would like to remove a dictionary you can delete the corresponding file from the additional folder in pwncheck's root directory.

Reporting-wise, the result of the scan from those new dictionaries will be exported to the data.csv file. Each dictionary will have its own field and unlike for HIBP, the plaintext of the password will be displayed if a hit is found.

Post-processing pwncheck reports

pwncheck calls a post-processing hook that enables for further action to be undertaken automatically when a new report is issued.

The hook is in scripts/post-hook.bat and can be customised as required. If absent, it will get re-created upon startup and won’t be overwritten on upgrade. The script will run within the context of the user running pwncheck (that may not be the privileged account whose credentials were entered manually) and will have the newly created report’s folder as current directory.

By default, safepass.me customers with an Enterprise subscription, get their reports automatically unlocked (provided internet connectivity is working) using this functionality. This is achieved by the following line in the script:

```
@FOR %%r in (*.zip) do ..\..\pwncheck %%r
```

Other actions could be undertaken, like emailing the results, logging in a JIRA ticket or even programmatically taking action against offending accounts (locking those accounts out, force-changing their passwords, ...).

Internet connectivity

pwncheck doesn't require internet connectivity to function but if it does detect one, it will use it to provide additional functionality such as:

- automatic database update
- automatic software update
- automatic report unlocking

Configuring a firewall

To use the online features of pwncheck, the following two URLs should be reachable:

- https://downloads.safepass.me
- https://portal.safepass.me

Please note that while those FQDNs are stable, they both point to cloud infrastructure that may resolve to different IP addresses. We therefore recommend that you do not whitelist by IP address but instead use SNI inspection or IP rules linked to DNS lookups to hole-punch.

Configuring a proxy server

pwncheck can use an external proxy server, provided you manually configure it to do so.

Assuming that you have a proxy server that doesn't expect any authentication, you can create the following **pwncheck.exe.config** file in the pwncheck folder:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.1"/>
  </startup>
  <system.net>
    <defaultProxy useDefaultCredentials="False">
      <proxy proxyaddress="http://my-proxy-server.local:3128/"/>
    </defaultProxy>
  </system.net>
</configuration>
```

Documentation on other combinations (using system settings, using authentication, …) is available at:

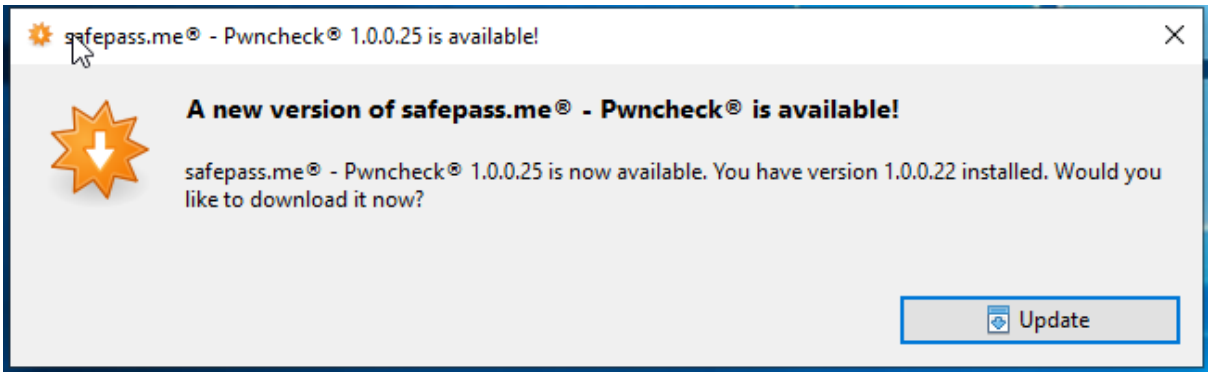
- https://docs.microsoft.com/en-us/dotnet/framework/network-programming/proxy-configuration
- https://docs.microsoft.com/en-us/dotnet/framework/configure-apps/file-schema/network/defaultproxy-element-network-settings

Fully offline usage

pwncheck can run fully offline provided you pre-fetch its offline database. Download it from https://downloads.safepass.me/data/HIBPv9.db and place it in the pwncheck folder before running the software.
Alternatively you can start the program non interactively and pass '--download' as its first and only parameter.

Auto-Updates

Pwncheck will auto-update on launch provided internet connectivity is available **AND** the software is run interactively.



We highly recommend that you consider running our sister tool, safepass.me (https://safepass.me) in conjunction to ensure that weak passwords are proactively prevented from being set in your environment (as opposed to being reactively acted upon).

Support

pwncheck is generally hassle-free software: it should just work!

Extended support options are available to our safepass.me Enterprise customers (you can get in touch with your account manager for full information).

safepass.me

The smartest password filter

Privacy

We take your information and privacy very seriously. We've been running an IT security company for almost two decades and have seen terribly insecure setups so we're extra aware of how important this is; that's why we collect the absolute minimum information required to deliver the best service possible.

This is what we collect to enable and enforce licensing:

- Number of active users of the domain (user account objects that are active, not disabled).
- Active Directory domain GUID
- NETBIOS Name of the computer the software is running on
- The version number of the software
- The IP address the software is used from (as seen from the Internet)
- The date the software was ran and a report generated

We believe it's of paramount importance to also make it crystal clear that we have built the tool to check your passwords completely offline. None of your users' passwords are ever sent outside of your network nor do we collect any password statistics.

Just like every other site we use cookies to provide a customised experience to our website by using analytics to give us insight on what information is most valuable to our visitors so we can ensure we can make improvements over time. We only use this information internally and we never share any of this information with any third party (unless we have to) nor do we monetise on your data.

A comprehensive privacy policy on the cookies we set and how they are used can be found on our website at https://safepass.me/

CSV fields

"SamAccountName" https://learn.microsoft.com/en-us/windows/win32/adschema/a-samaccountname
"DisplayName" https://learn.microsoft.com/en-us/windows/win32/adschema/a-displayname
"UPN" https://learn.microsoft.com/en-us/windows/win32/adschema/a-userprincipalname
"CanonicalName"
"DistinguishedName"
"ParentOU"
"Enabled" https://learn.microsoft.com/en-us/windows/win32/adschema/a-enabled
"LastLogonDate" https://learn.microsoft.com/en-us/windows/win32/adschema/a-lastlogon
"LastPasswordChange" https://learn.microsoft.com/en-us/windows/win32/adschema/a-pwdlastset
"SetToChangeAtNextLogon"
"LockedOut" https://learn.microsoft.com/en-us/windows/win32/adschema/a-lockouttime
"AdminCount" https://learn.microsoft.com/en-us/windows/win32/adschema/a-admincount
"RestrictedWorkstations"
https://learn.microsoft.com/en-us/windows/win32/adschema/a-userworkstations
"AccountExpires" https://learn.microsoft.com/en-us/windows/win32/adschema/a-accountexpires
"NoPasswordRequired"
"PasswordExpired"
"EmptyPassword" triggers when no password is set for the user
"HasLMHash" whether the account has any LM hash set (including in password history) that an attacker could leverage in a bruteforce attack
"NTHashGroup" if the number here matches on more than one row, the two accounts share the same password
"NeedsAttention" When either: 1) LM hashes were found 2) there is a hit on HIBP 3) the password is shared on more than one account 3) no password is required
"AdminDelegated"
https://www.ired.team/offensive-security-experiments/active-directory-kerberos-abuse/abusing-kerberos-constrained-delegation#prerequisites
"PasswordNeverExpires"
"PasswordUsesReversibleEncryption"
"AESKeysMissing" Accounts set up using older functional AD levels and as such have no AES keys, these will use weaker encryption methods. See below for details.
“Owner” SID or name of the owner of the user object.
“Created” timestamp of when the user object was created (UTC)

Kerberos related flags:

"msDS-supportedEncryptionTypes" see https://techcommunity.microsoft.com/t5/core-infrastructure-and-security/decrypting-the-selection-of-supported-kerberos-encryption-types/ba-p/1628797 For convenience pwncheck parses the field into the following: "PreAuthNotRequired" "DESOnly" "DESAllowed" "RC4Allowed" ”AES128Allowed" "AES256Allowed"

Whether a machine account uses the default password of “hostname$”

"IsDefault"
And all dictionaries will have their own column.

Changelog

1.0.32: April 2024

- Fix a bug that would lead to an exception being thrown (and the reporting process interrupted) while populating the “Created” column if the user running pwncheck lacked the permissions to read the “whenCreated” LDAP attribute.
- Ensure that the name related to the SID is shown in “Owner” when it is known

1.0.31: March 2024

- Add two new fields in CSV: “Owner” and “Created” to reflect who owns the user-object (usually the ‘domain admins’ group) and when it was created
- Introduce three new headless parameters (--download --upgrade and ‘c:\path\to\verylargedict.txt’); document them. Pwncheck will return a non-zero code if it fails unexpectedly.
- Fix a bug related to the parsing of "msDS-supportedEncryptionTypes”
- Return a readable error message rather than a stack trace if it is an authentication failure
- Refactor the logging code (pwncheck.log) and fix a race condition affecting post-processing hooks
- Switch to a newer EV certificate
- Update the database to HIBPv10 (dump of 20240330 ~ 932M entries)

Known issues

- The ‘--upgrade’ flag has been reported to require an interactive session
- FQDNs of DCs or domain names are required, bare IP addresses will not work