Namespace CompromisedCredentialsChecker

Classes

Checker

.NET package for V3 API of https://haveibeenpwned.com/

HIBPBreach

Breach information from the HaveIBeenPwned API

HIBPBreachName

A Pascal-cased name representing the breach which is unique across all other breaches. This value never changes and may be used to name dependent assets (such as images) but should not be shown directly to end users.

HIBPPaste

Paste information from the HaveIBeenPwned API

HIBPPastes

List of pastes with details

HIBPSubscribedDomain

Subscribed domain information from the HaveIBeenPwned API

HIBPSubscriptionStatus

SubscriptionStatus from the HaveIBeenPwned API

Class Checker

Namespace: CompromisedCredentialsChecker

Assembly: CompromisedCredentialsChecker.dll

.NET package for V3 API of https://haveibeenpwned.com/

```
public class Checker
```

Inheritance

object ← Checker

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Methods

CheckPastes(string, string, string)

Check for pastes that have been found that include this email address

```
public static HIBPPastes CheckPastes(string ApiKey, string UserAgent, string emailAddress)
```

Parameters

`ApiKey` string

API Key from https://haveibeenpwned.com/API/Key

`UserAgent` string

String to indicate what application is using the API

`emailAddress` string

Email address to be searched for

Returns

HIBPPastes

List of pastes with details

GetAllBreaches(string, string, string, bool)

Get a list of all of the breaches in the system

```
public static List<HIBPBreach> GetAllBreaches(string ApiKey, string UserAgent, string
DomainFilter = "", bool IsSpamList = false)
```

Parameters

`ApiKey` string

API Key from https://haveibeenpwned.com/API/Key

`UserAgent` string

String to indicate what application is using the API

`DomainFilter` string

If supplied, only breaches against the domain are returned.

`IsSpamList` bool

Filters the result set to only breaches that either are or are not flagged as a spam list.

Returns

List<HIBPBreach>

GetAllDataClasses(string, string)

Get all of the data classes in the system

```
public static List<string> GetAllDataClasses(string ApiKey, string UserAgent)
```

Parameters

`ApiKey` string

API Key from https://haveibeenpwned.com/API/Key

`UserAgent` string

String to indicate what application is using the API

Returns

List<string>

GetBreachedEmailsForDomain(string, string, string)

Determine all the breaches for email addresses for a specific domain.

```
public static dynamic GetBreachedEmailsForDomain(string ApiKey, string UserAgent,
string Domain)
```

Parameters

`ApiKey` string

API Key from https://haveibeenpwned.com/API/Key

`UserAgent` string

String to indicate what application is using the API

`Domain` string

Email address to be searched for

Returns

dynamic

All email addresses on a given domain and the breaches they've appeared in can be returned via the domain search API. Only domains that have been successfully added to the domain search dashboard after verifying control can be searched.

GetBreachesForEmailAddress(string, string, string, bool, string, bool)

Determine all the breaches the email address has been involved in.

```
public static List<HIBPBreach> GetBreachesForEmailAddress(string ApiKey, string UserAgent,
string EmailAddress, bool NamesOnly = true, string DomainFilter = "", bool ExcludeUnverified
= false)
```

Parameters

`ApiKey` string

API Key from https://haveibeenpwned.com/API/Key

`UserAgent` string

String to indicate what application is using the API

`EmailAddress` string

Email address to be searched for

`NamesOnly` bool

If true, only the names of the breaches are returned. If False, all breach data returned. Default is true and returns all information about the breaches

`DomainFilter` string

If supplied, only breaches against the domain are returned.

`ExcludeUnverified` bool

If true, this excludes breaches that have been flagged as "unverified". By default, both verified and unverified breaches are returned if this parameter not included or passed in as true

Returns

List<HIBPBreach>

Array of breaches that the email address has been involved in. If the number of breaches is 0 (zero) than the email address has not been involved in a breach

GetMostRecentBreachAdded(string, string)

Get the most recently added breach

```
public static HIBPBreach GetMostRecentBreachAdded(string ApiKey, string UserAgent)
```

Parameters

`ApiKey` string

API Key from https://haveibeenpwned.com/API/Key

`UserAgent` string

String to indicate what application is using the API

Returns

HIBPBreach

GetSingleBreachedSiteByName(string, string, string)

Get a breach by name

```
public static HIBPBreach GetSingleBreachedSiteByName(string ApiKey, string UserAgent,
string BreachName)
```

Parameters

`ApiKey` string

API Key from https://haveibeenpwned.com/API/Key

`UserAgent` string

String to indicate what application is using the API

`BreachName` string

Name of the breach from the list of breaches

Returns

HIBPBreach

GetSubscribedDomains(string, string)

Get a list of all domains that the API has subscribed to for breach notifications

```
public static List<HIBPSubscribedDomain> GetSubscribedDomains(string ApiKey,
string UserAgent)
```

Parameters

`ApiKey` string

API Key from https://haveibeenpwned.com/API/Key

`UserAgent` string

String to indicate what application is using the API

Returns

List<HIBPSubscribedDomain>

GetSubscriptionStatus(string, string)

Get details of the current subscription

```
public static HIBPSubscriptionStatus GetSubscriptionStatus(string ApiKey, string UserAgent)
```

Parameters

`ApiKey` string

API Key from https://haveibeenpwned.com/API/Key

`UserAgent` string

String to indicate what application is using the API

Returns

HIBPSubscriptionStatus

PasswordCheck(string, string, string)

Determine if the password has been found in a hack

```
public static long PasswordCheck(string ApiKey, string UserAgent, string PlainPassword)
```

Parameters

`ApiKey` string

API Key from https://haveibeenpwned.com/API/Key

`UserAgent` string

String to indicate what application is using the API

`PlainPassword` string

The password to be checked

Returns

long

The number of data breaches the password has been found in

Class HIBPBreach

Namespace: CompromisedCredentialsChecker

Assembly: CompromisedCredentialsChecker.dll

Breach information from the HaveIBeenPwned API

```
public class HIBPBreach
```

Inheritance

object ← HIBPBreach

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

AddedDate

The date and time (precision to the minute) the breach was added to the system in ISO 8601 format.

```
public DateTime AddedDate { get; set; }
```

Property Value

DateTime

BreachDate

The date (with no time) the breach originally occurred on in ISO 8601 format. This is not always accurate — frequently breaches are discovered and reported long after the original incident. Use this attribute as a guide only.

```
public DateTime BreachDate { get; set; }
```

Property Value

DateTime

DataClasses

This attribute describes the nature of the data compromised in the breach and contains an alphabetically ordered string array of impacted data classes.

```
public List<string> DataClasses { get; set; }
```

Property Value

List <string>

Description

Contains an overview of the breach represented in HTML markup. The description may include markup such as emphasis and strong tags as well as hyperlinks.

```
public string Description { get; set; }
```

Property Value

string

Domain

The domain of the primary website the breach occurred on. This may be used for identifying other assets external systems may have for the site.

```
public string Domain { get; set; }
```

Property Value

string

IsFabricated

Indicates that the breach is considered unverified. An unverified breach may not have been hacked from the indicated website. An unverified breach is still loaded into HIBP when there's sufficient confidence that a significant portion of the data is legitimate.

```
public bool IsFabricated { get; set; }
```

Property Value

bool

IsMalware

Indicates if the breach is sourced from malware. This flag has no impact on any other attributes, it merely flags that the data was sourced from a malware campaign rather than a security compromise of an online service.

```
public bool IsMalware { get; set; }
```

Property Value

bool

IsRetired

Indicates if the breach has been retired. This data has been permanently removed and will not be returned by the API.

```
public bool IsRetired { get; set; }
```

Property Value

bool

IsSensitive

Indicates if the breach is considered sensitive. The public API will not return any accounts for a breach flagged as sensitive.

```
public bool IsSensitive { get; set; }
```

Property Value

bool

IsSpamList

Indicates if the breach has been retired. This data has been permanently removed and will not be returned by the API.

```
public bool IsSpamList { get; set; }
```

Property Value

bool

IsSubscriptionFree

Indicates if the breach is subscription free. This flag has no impact on any other attributes, it is only used when running a domain search where a sufficiently sized subscription isn't present.

```
public bool IsSubscriptionFree { get; set; }
```

Property Value

bool

IsVerified

Indicates that the breach is considered unverified. An unverified breach may not have been hacked from the indicated website. An unverified breach is still loaded into HIBP when there's sufficient confidence that a significant portion of the data is legitimate.

```
public bool IsVerified { get; set; }
```

Property Value

bool

LogoPath

A URI that specifies where a logo for the breached service can be found. Logos are always in PNG format.

```
public string LogoPath { get; set; }
```

Property Value

string

ModifiedDate

The date and time (precision to the minute) the breach was modified in ISO 8601 format. This will only differ from the AddedDate attribute if other attributes represented here are changed or data in the breach itself is changed (i.e. additional data is identified and loaded). It is always either equal to or greater then the AddedDate attribute, never less than.

```
public DateTime ModifiedDate { get; set; }
```

Property Value

DateTime

Name

A Pascal-cased name representing the breach which is unique across all other breaches. This value never changes and may be used to name dependent assets (such as images) but should not be shown directly to end users (see the "Title" attribute instead).

```
public string Name { get; set; }
```

Property Value

string

PwnCount

The total number of accounts loaded into the system. This is usually less than the total number reported by the media due to duplication or other data integrity issues in the source data.

```
public int PwnCount { get; set; }
```

Property Value

int

Title

A descriptive title for the breach suitable for displaying to end users. It's unique across all breaches but individual values may change in the future (i.e. if another breach occurs against an organisation already in the system). If a stable value is required to reference the breach, refer to the "Name" attribute instead.

```
public string Title { get; set; }
```

Property Value

string

Class HIBPBreachName

Namespace: CompromisedCredentialsChecker

Assembly: CompromisedCredentialsChecker.dll

A Pascal-cased name representing the breach which is unique across all other breaches. This value never changes and may be used to name dependent assets (such as images) but should not be shown directly to end users.

```
public class HIBPBreachName
```

Inheritance

object ← HIBPBreachName

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

Name

A Pascal-cased name representing the breach which is unique across all other breaches. This value never changes and may be used to name dependent assets (such as images) but should not be shown directly to end users.

```
public string Name { get; set; }
```

Property Value

string

Class HIBPPaste

Namespace: CompromisedCredentialsChecker

Assembly: CompromisedCredentialsChecker.dll

Paste information from the HaveIBeenPwned API

```
public class HIBPPaste
```

Inheritance

object ← HIBPPaste

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

Date

The date and time (precision to the second) that the paste was posted. This is taken directly from the paste site when this information is available but may be null if no date is published.

```
public DateTime? Date { get; set; }
```

Property Value

DateTime?

EmailCount

The number of emails that were found when processing the paste. Emails are extracted by using the regular expression \b[a-zA-Z0-9.-+]+@[a-zA-Z0-9.-]+.[a-zA-Z]+\b

```
public int EmailCount { get; set; }
```

Property Value

int

Id

The ID of the paste as it was given at the source service. Combined with the "Source" attribute, this can be used to resolve the URL of the paste.

```
public string Id { get; set; }
```

Property Value

string

Source

The paste service the record was retrieved from. Current values are: Pastebin, Pastie, Slexy, Ghostbin, QuickLeak, JustPaste, AdHocUrl, PermanentOptOut, OptOut

```
public string Source { get; set; }
```

Property Value

string

Title

The title of the paste as observed on the source site. This may be null and if so will be omitted from the response.

```
public string Title { get; set; }
```

Property Value

string

Class HIBPPastes

Namespace: CompromisedCredentialsChecker

Assembly: CompromisedCredentialsChecker.dll

List of pastes with details

```
public class HIBPPastes : List<HIBPPaste>, IList<HIBPPaste>, ICollection<HIBPPaste>,
IReadOnlyList<HIBPPaste>, IReadOnlyCollection<HIBPPaste>, IEnumerable<HIBPPaste>, IList,
ICollection, IEnumerable
```

Inheritance

object ← List<HIBPPaste> ← HIBPPastes

Implements

IList<HIBPPaste>, ICollection<HIBPPaste>, IReadOnlyList<HIBPPaste>, IReadOnlyCollection<HIBPPaste>, IEnumerable<HIBPPaste>, IList, ICollection, IEnumerable

Inherited Members

List<HIBPPaste>.Add(HIBPPaste) , List<HIBPPaste>.AddRange(IEnumerable<HIBPPaste>) , List<HIBPPaste>.AsReadOnly() , List<HIBPPaste>.BinarySearch(int, int, HIBPPaste, IComparer<HIBPPaste>) , List<HIBPPaste>.BinarySearch(HIBPPaste) , List<HIBPPaste>.BinarySearch(HIBPPaste, IComparer<HIBPPaste>) , List<HIBPPaste>.Clear() , List<HIBPPaste>.Contains(HIBPPaste) , List<HIBPPaste>.ConvertAll<TOutput>(Converter<HIBPPaste, TOutput>) , List<HIBPPaste>.CopyTo(int, HIBPPaste[], int, int) , List<HIBPPaste>.CopyTo(HIBPPaste[]) , List<HIBPPaste>.CopyTo(HIBPPaste[], int) , List<HIBPPaste>.EnsureCapacity(int) , List<HIBPPaste>.Exists(Predicate<HIBPPaste>) , List<HIBPPaste>.Find(Predicate<HIBPPaste>) , List<HIBPPaste>.FindAll(Predicate<HIBPPaste>) , List<HIBPPaste>.FindIndex(int, int, Predicate<HIBPPaste>) , List<HIBPPaste>.FindIndex(int, Predicate<HIBPPaste>) , List<HIBPPaste>.FindIndex(Predicate<HIBPPaste>) , List<HIBPPaste>.FindLast(Predicate<HIBPPaste>) , List<HIBPPaste>.FindLastIndex(int, int, Predicate<HIBPPaste>) , List<HIBPPaste>.FindLastIndex(int, Predicate<HIBPPaste>) , List<HIBPPaste>.FindLastIndex(Predicate<HIBPPaste>) , List<HIBPPaste>.ForEach(Action<HIBPPaste>) , List<HIBPPaste>.GetEnumerator() , List<HIBPPaste>.GetRange(int, int) , List<HIBPPaste>.IndexOf(HIBPPaste) , List<HIBPPaste>.IndexOf(HIBPPaste, int) , List<HIBPPaste>.IndexOf(HIBPPaste, int, int) ,

List<HIBPPaste>.Insert(int, HIBPPaste) , List<HIBPPaste>.InsertRange(int, IEnumerable<HIBPPaste>) ,
List<HIBPPaste>.LastIndexOf(HIBPPaste) , List<HIBPPaste>.LastIndexOf(HIBPPaste, int) ,
List<HIBPPaste>.LastIndexOf(HIBPPaste, int, int) , List<HIBPPaste>.Remove(HIBPPaste) ,
List<HIBPPaste>.RemoveAll(Predicate<HIBPPaste>) , List<HIBPPaste>.RemoveAt(int) ,
List<HIBPPaste>.RemoveRange(int, int) , List<HIBPPaste>.Reverse() ,
List<HIBPPaste>.Reverse(int, int) , List<HIBPPaste>.Slice(int, int) , List<HIBPPaste>.Sort() ,
List<HIBPPaste>.Sort(IComparer<HIBPPaste>) , List<HIBPPaste>.Sort(Comparison<HIBPPaste>) ,
List<HIBPPaste>.Sort(int, int, IComparer<HIBPPaste>) , List<HIBPPaste>.ToArray() ,
List<HIBPPaste>.TrimExcess() , List<HIBPPaste>.TrueForAll(Predicate<HIBPPaste>) ,
List<HIBPPaste>.Capacity , List<HIBPPaste>.Count , List<HIBPPaste>.this[int] ,
object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() ,
object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Class HIBPSubscribedDomain

Namespace: CompromisedCredentialsChecker

Assembly: CompromisedCredentialsChecker.dll

Subscribed domain information from the HaveIBeenPwned API

```
public class HIBPSubscribedDomain
```

Inheritance

object ← HIBPSubscribedDomain

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

DomainName

The full domain name that has been successfully verified.

```
public string DomainName { get; set; }
```

Property Value

string

NextSubscriptionRenewal

The date and time the current subscription ends in ISO 8601 format. The PwnCountExcludingSpamListsAtLastSubscriptionRenewal value is locked in until this time (will be null if there have been no subscriptions).

```
public DateTime NextSubscriptionRenewal { get; set; }
```

Property Value

DateTime

PwnCount

The total number of breached email addresses found on the domain at last search (will be null if no searches yet performed).

```
public int PwnCount { get; set; }
```

Property Value

int

PwnCountExcludingSpamLists

The number of breached email addresses found on the domain at last search, excluding any breaches flagged as a spam list (will be null if no searches yet performed).

```
public int PwnCountExcludingSpamLists { get; set; }
```

Property Value

int

PwnCountExcludingSpamListsAtLastSubscriptionRenewal

The total number of breached email addresses found on the domain when the current subscription was taken out (will be null if no searches yet performed). This number ensures the domain remains searchable throughout the subscription period even if the volume of breached accounts grows beyond the subscription's scope.

```
public object PwnCountExcludingSpamListsAtLastSubscriptionRenewal { get; set; }
```

Property Value

object

Class HIBPSubscriptionStatus

Namespace: CompromisedCredentialsChecker

Assembly: CompromisedCredentialsChecker.dll

SubscriptionStatus from the HaveIBeenPwned API

```
public class HIBPSubscriptionStatus
```

Inheritance

object ← HIBPSubscriptionStatus

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

Description

A human readable sentence explaining the scope of the subscription.

```
public string Description { get; set; }
```

Property Value

string

DomainSearchMaxBreachedAccounts

The size of the largest domain the subscription can search. This is expressed in the total number of breached accounts on the domain, excluding those that appear solely in spam list.

```
public int DomainSearchMaxBreachedAccounts { get; set; }
```

Property Value

int

Rpm

The rate limit in requests per minute. This applies to the rate the breach search by email address API can be requested.

```
public int Rpm { get; set; }
```

Property Value

int

SubscribedUntil

The date and time the current subscription ends in ISO 8601 format.

```
public DateTime SubscribedUntil { get; set; }
```

Property Value

DateTime

SubscriptionName

The name representing the subscription being either "Pwned 1", "Pwned 2", "Pwned 3" or "Pwned 4".

```
public string SubscriptionName { get; set; }
```

Property Value

string