CredEntial sensitive information type (SIT) definitions

Contents

[ASP.NET Machine Key 2](#_Toc102984354)

[Amazon S3 Client Secret Access Key 3](#_Toc102984355)

[Azure Subscription Management Certificate 4](#_Toc102984356)

[Azure SQL Connection String 6](#_Toc102984357)

[Azure Service Bus Shared Access Signature 7](#_Toc102984358)

[Azure Redis Cache Connection String Password 9](#_Toc102984359)

[Azure IoT Shared Access Key 10](#_Toc102984360)

[Azure Storage Account Shared Access Signature 12](#_Toc102984361)

[Azure Storage Account Shared Access Signature for High Risk Resources 13](#_Toc102984362)

[Azure Logic App Shared Access Signature 15](#_Toc102984363)

[Azure Storage Account Access Key 16](#_Toc102984364)

[Azure COSMOS DB Account Access Key 18](#_Toc102984365)

[Azure App Service Deployment Password 19](#_Toc102984366)

[Azure DevOps Personal Access Token 20](#_Toc102984367)

[Azure DevOps App Secret 21](#_Toc102984368)

[Azure Function Master / API Key 23](#_Toc102984369)

[Azure Shared Access Key / Web Hook Token 24](#_Toc102984370)

[Azure AD Client Access Token 26](#_Toc102984371)

[Azure AD User Credentials 28](#_Toc102984372)

[Azure AD Client Secret 30](#_Toc102984373)

[Azure Bot Service App Secret 31](#_Toc102984374)

[Azure Databricks Personal Access Token 33](#_Toc102984375)

[Azure Container Registry Access Key 34](#_Toc102984376)

[Azure Batch Shared Access Key 35](#_Toc102984377)

[Azure SignalR Access Key 36](#_Toc102984378)

[Azure EventGrid Access Key 37](#_Toc102984379)

[Azure Machine Learning Web Service API Key 38](#_Toc102984380)

[Azure Cognitive Search API Key 39](#_Toc102984381)

[Azure Cognitive Service Key 41](#_Toc102984382)

[Azure Maps Subscription Key 42](#_Toc102984383)

[Azure Bot Framework Secret Key 43](#_Toc102984384)

[Client Secret / Api Key 45](#_Toc102984385)

[General Password 49](#_Toc102984386)

[General Symmetric Key 55](#_Toc102984387)

[GitHub Personal Access Token 56](#_Toc102984388)

[Google API key 58](#_Toc102984389)

[Http Authorization Header 59](#_Toc102984390)

[Microsoft Bing Maps Key 60](#_Toc102984391)

[Slack Access Token 61](#_Toc102984392)

[User Login Credentials 63](#_Toc102984393)

[X.509 Certificate Private Key 65](#_Toc102984394)

# ASP.NET Machine Key

Format:

Symmetric keys in XML configuration

Pattern:

Variant symmetric key formats in XML e.g.

... <machineKey decryptionKey="\*\*\*\*\*\*\*\* ...

... <machineKey validationKey="\*\*\*\*\*\*\*\* ...

...

Samples:

<machineKey validationKey="ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789" decryptionKey="ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789"...

Checksum:

No

Description:

This SIT "ASP.NET Machine Key" is designed to match the security information that's used to encrypt or hash data in ASP.NET forms authentication and view state (https://docs.microsoft.com/en-us/dotnet/api/system.web.security.machinekey?view=netframework-4.8). To do this, it uses several primary resources:

Patterns of Symmetric key context in xml files.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKeyContextInXml:

password

key

connectionstring

------------------------------------------------------------------------

# Amazon S3 Client Secret Access Key

Format:

A 40-character combination of letters, digits, and special characters

Pattern:

Any combination of 40

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

e.g.

abcdefghijklmnopqrst0123456789/+ABCDEFGH

Samples:

string AmazonWebServicesSecretToken = "abcdefghijklmnopqrst0123456789/+ABCDEFGH";

Checksum:

No

Description:

This SIT "Amazon S3 Client Secret Access Key" is designed to match the security information that's used to access Amazon Web Services (https://docs.aws.amazon.com/toolkit-for-eclipse/v1/user-guide/setup-credentials.html). To do this, it uses several primary resources:

Patterns of Base64 encoded 240 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey240:

secret

key

------------------------------------------------------------------------

# Azure Subscription Management Certificate

Format:

Up to 20,000-character combination of letters, digits, and special characters

Pattern:

Any combination of

up to 20,000

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

Up to 2

equal signs (=)

e.g.

MIIKcQIBAzCCCi0GCSqGSIb3DQEHAaCCCh4EggoaMIIKFjCCBg8GCSqGSIb3DQEHAaCCBgAEggX8MIIF+DCCBfQGCyqGSIb3DQEM…

Samples:

<Subscription id="..." ManagementCertificate="MIIPuQIBGSIb3DQEHAaC..."

Checksum:

Yes

Description:

This SIT "Azure Subscription Management Certificate" is designed to match the security information that's used to authenticate with the classic deployment model provided by Azure. Many programs and tools (such as Visual Studio or the Azure SDK) use these certificates to automate configuration and deployment of various Azure services (https://docs.microsoft.com/en-us/azure/azure-api-management-certs). To do this, it uses several primary resources:

Patterns of Base64 encoded string literal.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_Base64EncodedStringLiteral:

MII

------------------------------------------------------------------------

# Azure SQL Connection String

Format:

Up to 20,000-character combination of letters, digits, and special characters

or

A pair of username and password used in general authentication process

Pattern:

Any combination of

up to 20,000

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

Up to 2

equal signs (=)

e.g.

MIIKcQIBAzCCCi0GCSqGSIb3DQEHAaCCCh4EggoaMIIKFjCCBg8GCSqGSIb3DQEHAaCCBgAEggX8MIIF+DCCBfQGCyqGSIb3DQEM…

or

Variant username and password formats e.g.

username=...;password=\*\*\*\*\*\*\*\*;

user id=...;password=\*\*\*\*\*\*\*\*;

uid=...;pwd=\*\*\*\*\*\*\*\*;

DB\_USER=...;DB\_PASS=\*\*\*\*\*\*\*\*;

Service Account=...;Password=\*\*\*\*\*\*\*\*;

...

Samples:

server=...database.windows.net;database=...;user=...;pwd=ZYXWVU\_2;

<add key="ConnectionString" value="server=tcp:ZYXWVU\_1.database.windows.net;database=ZYXWVU\_1;user=ZYXWVU\_1;password=ZYXWVU\_2;"

Checksum:

Yes

Description:

This SIT "Azure SQL Connection String" is designed to match the security information that's used to connect to Azure SQL Databases (https://docs.microsoft.com/en-us/azure/sql-database/sql-database-aad-authentication-configure). To do this, it uses several primary resources:

Patterns of Base64 encoded string literal.

Patterns of Plain-text username and password.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

Patterns of mockup values, redactions, and placeholders.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_Base64EncodedStringLiteral:

MII

Keyword\_LoginCredentials:

password

pw

DB\_

------------------------------------------------------------------------

# Azure Service Bus Shared Access Signature

Format:

A 44-character combination of letters, digits, and special characters

or

Up to 76-character combination of letters, digits, and special characters

Pattern:

Any combination of 43

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

ends with an equal sign (=)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

or

Any combination of 43 to 73

a-z (not case-sensitive)

0-9

or percent signs (%)

ends with a suffix '%3d' (not case-sensitive)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789%2F%2BABCDE%3D

Samples:

Endpoint=sb://...servicebus.windows.net...SharedAccessKey=abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

ServiceBusNamespace...SharedAccessPolicy...Key=abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

Checksum:

No

Description:

This SIT "Azure Service Bus Shared Access Signature" is designed to match the security information that's used to grant a user access to Azure Service Bus resources with specific rights (https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-authentication-and-authorization). To do this, it uses several primary resources:

Patterns of Base64 encoded 256 bits symmetric key.

Patterns of URL Encoded 256 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName, Id.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey256:

SharedAccessKey

AccountKey

Keyword\_SymmetricKey256UrlEncoded:

sig=

key

token

secret

password

------------------------------------------------------------------------

# Azure Redis Cache Connection String Password

Format:

Up to 20,000-character combination of letters, digits, and special characters

or

A 44-character combination of letters, digits, and special characters

Pattern:

Any combination of

up to 20,000

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

Up to 2

equal signs (=)

e.g.

MIIKcQIBAzCCCi0GCSqGSIb3DQEHAaCCCh4EggoaMIIKFjCCBg8GCSqGSIb3DQEHAaCCBgAEggX8MIIF+DCCBfQGCyqGSIb3DQEM…

or

Any combination of 43

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

ends with an equal sign (=)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

Samples:

HostName=ZYXWVU\_1.redis.cache.windows.net;Password=abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

Checksum:

Yes

Description:

This SIT "Azure Redis Cache Connection String Password" is designed to match the security information that's used to connect to Azure Cache for Redis servers (https://docs.microsoft.com/en-us/azure/azure-cache-for-redis/). To do this, it uses several primary resources:

Patterns of Base64 encoded string literal.

Patterns of Base64 encoded 256 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName, Id.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_Base64EncodedStringLiteral:

MII

Keyword\_SymmetricKey256:

SharedAccessKey

AccountKey

------------------------------------------------------------------------

# Azure IoT Shared Access Key

Format:

A 44-character combination of letters, digits, and special characters

Pattern:

Any combination of 43

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

ends with an equal sign (=)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

Samples:

HostName=...azure-devices.net...SharedAccessKeyName=...SharedAccessKey=abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=, iotHub...abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

iotHub...abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

Checksum:

No

Description:

This SIT "Azure IoT Shared Access Key" is designed to match the security information that's used to authenticate Azure IoT devices and services (https://docs.microsoft.com/en-us/azure/iot-fundamentals/iot-security-deployment). To do this, it uses several primary resources:

Patterns of Base64 encoded 256 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName, Id.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey256:

SharedAccessKey

AccountKey

------------------------------------------------------------------------

# Azure Storage Account Shared Access Signature

Format:

A 44-character combination of letters, digits, and special characters

or

Up to 76-character combination of letters, digits, and special characters

Pattern:

Any combination of 43

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

ends with an equal sign (=)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

or

Any combination of 43 to 73

a-z (not case-sensitive)

0-9

or percent signs (%)

ends with a suffix '%3d' (not case-sensitive)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789%2F%2BABCDE%3D

Samples:

...core.windows.net...sr=...sv=...st=...se=...sp=...sig=abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

https://...core.windows.net...sig=abcdefghijklmnopqrstuvwxyz0123456789%2F%2BABCDE%3D

Checksum:

No

Description:

This SIT "Azure Storage Account Shared Access Signature" is designed to match the security information that's used to grant restricted access rights to Azure Storage resources (https://docs.microsoft.com/en-us/rest/api/storageservices/delegate-access-with-shared-access-signature). To do this, it uses several primary resources:

Patterns of Base64 encoded 256 bits symmetric key.

Patterns of URL Encoded 256 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName, Id.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey256:

SharedAccessKey

AccountKey

Keyword\_SymmetricKey256UrlEncoded:

sig=

key

token

secret

password

------------------------------------------------------------------------

# Azure Storage Account Shared Access Signature for High Risk Resources

Format:

A 44-character combination of letters, digits, and special characters

or

Up to 76-character combination of letters, digits, and special characters

Pattern:

Any combination of 43

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

ends with an equal sign (=)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

or

Any combination of 43 to 73

a-z (not case-sensitive)

0-9

or percent signs (%)

ends with a suffix '%3d' (not case-sensitive)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789%2F%2BABCDE%3D

Samples:

https://ZYXWVU\_1.core.windows.net/blob/ZYXWVU\_1.cspkg?sig=abcdefghijklmnopqrstuvwxyz0123456789%2F%2BABCDE%3D

Checksum:

No

Description:

This SIT "Azure Storage Account Shared Access Signature for High Risk Resources" is designed to match the security information that's used to grant restricted access rights to Azure Storage resources (https://docs.microsoft.com/en-us/rest/api/storageservices/delegate-access-with-shared-access-signature). To do this, it uses several primary resources:

Patterns of Base64 encoded 256 bits symmetric key.

Patterns of URL Encoded 256 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName, Id.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey256:

SharedAccessKey

AccountKey

Keyword\_SymmetricKey256UrlEncoded:

sig=

key

token

secret

password

------------------------------------------------------------------------

# Azure Logic App Shared Access Signature

Format:

Up to 76-character combination of letters, digits, and special characters

Pattern:

Any combination of 43 to 73

a-z (not case-sensitive)

0-9

or percent signs (%)

ends with a suffix '%3d' (not case-sensitive)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789%2F%2BABCDE%3D

Samples:

https://test.logic.azure.com/ZYXWVU\_1;.?..sig=abcdefghijklmnopqrstuvwxyz0123456789%2F%2BABCDE%3D

Checksum:

No

Description:

This SIT "Azure Logic App Shared Access Signature" is designed to match the security information that's used to grant access to a request endpoint on Azure Logic Apps (https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-securing-a-logic-app?tabs=azure-portal). To do this, it uses several primary resources:

Patterns of URL Encoded 256 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey256UrlEncoded:

sig=

key

token

secret

password

------------------------------------------------------------------------

# Azure Storage Account Access Key

Format:

Up to 20,000-character combination of letters, digits, and special characters

or

An 88-character combination of letters, digits, and special characters

Pattern:

Any combination of

up to 20,000

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

Up to 2

equal signs (=)

e.g.

MIIKcQIBAzCCCi0GCSqGSIb3DQEHAaCCCh4EggoaMIIKFjCCBg8GCSqGSIb3DQEHAaCCBgAEggX8MIIF+DCCBfQGCyqGSIb3DQEM…

or

Any combination of 86

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

ends with two equal signs (=)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEabcdefghijklmnopqrstuvwxyz0123456789/+ABCDE==

Samples:

Endpoint=...table.core.windows.net;AccountName=...;AccountKey=abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEabcdefghijklmnopqrstuvwxyz0123456789/+ABCDE==

AccountName=...;AccountKey=abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEabcdefghijklmnopqrstuvwxyz0123456789/+ABCDE==;EndpointSuffix=...;

Account...PrimaryKey=abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEabcdefghijklmnopqrstuvwxyz0123456789/+ABCDE==;

SecondaryKey=abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEabcdefghijklmnopqrstuvwxyz0123456789/+ABCDE==

Checksum:

Yes

Description:

This SIT "Azure Storage Account Access Key" is designed to match the security information that's used to make request against Azure Storage services (e.g. Blob, Queue, Table and File services) (https://docs.microsoft.com/en-us/rest/api/storageservices/authorize-with-shared-key). To do this, it uses several primary resources:

Patterns of Base64 encoded string literal.

Patterns of Base64 encoded 512 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName, Id, AccountName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_Base64EncodedStringLiteral:

MII

Keyword\_SymmetricKey512:

SharedAccessKey

AccountKey

------------------------------------------------------------------------

# Azure COSMOS DB Account Access Key

Format:

An 88-character combination of letters, digits, and special characters

Pattern:

Any combination of 86

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

ends with two equal signs (=)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEabcdefghijklmnopqrstuvwxyz0123456789/+ABCDE==

Samples:

AccountEndpoint=https://...documents.azure.com...;AccountKey=abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEabcdefghijklmnopqrstuvwxyz0123456789/+ABCDE==

DocDbConnectionStr...abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEabcdefghijklmnopqrstuvwxyz0123456789/+ABCDE==

Checksum:

No

Description:

This SIT "Azure COSMOS DB Account Access Key" is designed to match the security information that's used to provide access to administrative resources for Azure COSMOS Database accounts (https://docs.microsoft.com/en-us/azure/cosmos-db/secure-access-to-data). To do this, it uses several primary resources:

Patterns of Base64 encoded 512 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName, Id, AccountName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey512:

SharedAccessKey

AccountKey

------------------------------------------------------------------------

# Azure App Service Deployment Password

Format:

A 60-character combination of letters, digits, and special characters

Pattern:

Any combination of 60

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEFGHIJKLMNOPQRSTUV

Samples:

userPWD=abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEFGHIJKLMNOPQRSTUV;

PublishingPassword=abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEFGHIJKLMNOPQRSTUV...

Checksum:

No

Description:

This SIT "Azure App Service Deployment Password" is designed to match the security information that's used to secure Azure App Sevice deployment from a local computer (https://docs.microsoft.com/en-us/azure/app-service/deploy-configure-credentials). To do this, it uses several primary resources:

Patterns of Base64 encoded 360 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey360:

password

pw

------------------------------------------------------------------------

# Azure DevOps Personal Access Token

Format:

A 52-character combination of letters, digits, and special characters

Pattern:

Any combination of 52

a-z or A-Z (case-sensitive)

or 2-7

e.g.

ntpi2ch67ci2vjzcohglogyygwo5fuyl365n2zdowwxhsys6jnoa

Samples:

URL="\*\*\*.visualstudio.com"; PAT = "ntpi2ch67ci2vjzcohglogyygwo5fuyl365n2zdowwxhsys6jnoa"

Checksum:

Yes

Description:

This SIT "Azure DevOps Personal Access Token" is designed to match the security information that's used as an alternate password to authenticate into Azure DevOps (https://docs.microsoft.com/en-us/azure/devops/organizations/accounts/use-personal-access-tokens-to-authenticate?view=azure-devops). To do this, it uses several primary resources:

Patterns of Base32 encoded 256 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey256B32:

pat

token

ado

vsts

azuredevops

visualstudio.com

dev.azure.com

------------------------------------------------------------------------

# Azure DevOps App Secret

Format:

A 52-character combination of letters, digits, and special characters

Pattern:

Any combination of 52

a-z or A-Z (case-sensitive)

or 2-7

e.g.

ntpi2ch67ci2vjzcohglogyygwo5fuyl365n2zdowwxhsys6jnoa

Samples:

AdoAppId="ZYXWVU\_1;";AdoAppSecret="ntph2ch67ciqunzcohglogyygwo5fuyl365n4zdowwxhsys6jnoa"

Checksum:

Yes

Description:

This SIT "Azure DevOps App Secret" is designed to match the security information that's used to authenticate web app users for Azure DevOps REST API access (https://docs.microsoft.com/en-us/azure/devops/integrate/get-started/authentication/oauth?view=azure-devops). To do this, it uses several primary resources:

Patterns of Base32 encoded 256 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey256B32:

pat

token

ado

vsts

azuredevops

visualstudio.com

dev.azure.com

------------------------------------------------------------------------

# Azure Function Master / API Key

Format:

A 56-character combination of letters, digits, and special characters

or

Up to 90-character combination of letters, digits, and special characters

Pattern:

Any combination of 54

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

ends with two equal signs (=)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEFGHIJKLMNOP==

or

Any combination of 54 to 84

a-z (not case-sensitive)

0-9

or percent signs (%)

ends with a suffix '%3d%3d' (not case-sensitive)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789%2F%2BABCDEF0123456789%3D%3D

Samples:

https://...azurewebsites.net/api/...?code=abcdefghijklmnopqrstuvwxyz0123456789%2F%2BABCDEF0123456789%3D%3D...

ApiEndpoint = ...azurewebsites.net/api/...; ApiKey=abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEFGHIJKLMNOP==...

x-functions-key: abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEFGHIJKLMNOP==

Checksum:

No

Description:

This SIT "Azure Function Master / API Key" is designed to match the security information that's used to request Azure Function API when its authorization level is set a value other than anonymous (https://docs.microsoft.com/en-us/azure/azure-functions/functions-how-to-use-azure-function-app-settings?tabs=portal#get-your-function-access-keys). To do this, it uses several primary resources:

Patterns of Base64 encoded 320 bits symmetric key.

Patterns of URL Encoded 320 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey320:

code=

key

Keyword\_SymmetricKey320UrlEncoded:

code=

key

------------------------------------------------------------------------

# Azure Shared Access Key / Web Hook Token

Format:

A 44-character combination of letters, digits, and special characters

or

Up to 76-character combination of letters, digits, and special characters

Pattern:

Any combination of 43

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

ends with an equal sign (=)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

or

Any combination of 43 to 73

a-z (not case-sensitive)

0-9

or percent signs (%)

ends with a suffix '%3d' (not case-sensitive)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789%2F%2BABCDE%3D

Samples:

PrimaryKey=abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=;

Checksum:

No

Description:

This SIT "Azure Shared Access Key / Web Hook Token" is designed to match the security information that's used to access general Azure resources with restricted permission (https://docs.microsoft.com/en-us/azure/notification-hubs/notification-hubs-push-notification-security#security-claims). To do this, it uses several primary resources:

Patterns of Base64 encoded 256 bits symmetric key.

Patterns of URL Encoded 256 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName, Id.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey256:

SharedAccessKey

AccountKey

Keyword\_SymmetricKey256UrlEncoded:

sig=

key

token

secret

password

------------------------------------------------------------------------

# Azure AD Client Access Token

Format:

Up to 10,000-character combination of letters, digits, and special characters

or

A client secret or refresh token used in OAuth2.0 protocol

or

Up to 1,000-character combination of letters, digits, and special characters

Pattern:

Any combination of

up to 10,000

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

Up to 2

equal signs (=)

e.g.

"VersionProfile": null, "TokenCache": { "CacheData": "AgAAAAIAAACZAWh0dHBzOi8vbG9naW4ubWljcm9zb2…

or

Variant client secret or refresh token formats e.g.

ClientSecret:\*\*\*\*\*\*\*\*

AppSecret=\*\*\*\*\*\*\*\*

ConsumerKey:=\*\*\*\*\*\*\*\*

Refresh\_Token:\*\*\*\*\*\*\*\*

or

3 letters: eyJ (case-sensitive)

Any combination of

up to 1,000

a-z (not case-sensitive)

0-9

dashes (-)

underlines (\_)

or dots (.)

e.g.

eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Ing0Nzh4eU9wbHNNMUg3TlhrN1N4MTd4MXVwYyIsImtpZCI6Ing0Nzh4…

Samples:

Authroization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJS...

Checksum:

Yes

Description:

This SIT "Azure AD Client Access Token" is designed to match the security information that's contains claims that one can use in Azure Active Directory B2C (Azure AD B2C) to identify the granted permissions to Azure resources (https://docs.microsoft.com/en-us/azure/active-directory-b2c/active-directory-b2c-access-tokens). To do this, it uses several primary resources:

Patterns of Azure PowerShell Token Cache.

Patterns of Client secret context.

Patterns of Json Web Token.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_AzurePowerShellTokenCache:

tokencache

Keyword\_ClientSecretContext:

secret

token

auth

securestring

key

Keyword\_JsonWebToken:

eyJ

------------------------------------------------------------------------

# Azure AD User Credentials

Format:

A pair of username and password related to \*.onmicrosoft.com domain

or

Plain-text password used in code snippets

or

Plain-text password used in XML configuration

Pattern:

Variant username and password formats e.g.

username=...password=\*\*\*\*\*\*\*\*

/user:.../pass:\*\*\*\*\*\*\*\*

SharePointOnlineAuthenticatedContext...

sign\_in...

...

or

Variant password formats in code snippets e.g.

... new X509Certificates2( ...

... ConvertTo-SecureString -String \*\*\*\*\*\*\*\*...

... password = "\*\*\*\*\*\*\*\*"...

... "password" : "\*\*\*\*\*\*\*\*"...

... UserPasswordCredential( ...

...

or

Variant password formats in XML e.g.

... <secret>\*\*\*\*\*\*\*\*</secret> ...

... <password>\*\*\*\*\*\*\*\*</password> ...

... <setting name="password" value="\*\*\*\*\*\*\*\*" > ...

... <setting name="password">\*\*\*\*\*\*\*\*</setting> ...

... <setting name="password"><value>\*\*\*\*\*\*\*\*</value></setting> ...

...

Samples:

username=user@cortso.onmicrosoft.com;password=ZYXWVU\_1;

Checksum:

No

Description:

This SIT "Azure AD User Credentials" is designed to match the security information that's used as individual user passwords to authenticate against Azure Active Directory (https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-users-reset-password-azure-portal). To do this, it uses several primary resources:

Patterns of Plain-text username and password for Azure AD tenant.

Patterns of Password context in code.

Patterns of Password context in XML.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_AzureActiveDirectoryLoginCredentials:

password

pw

userpass

credentials

cmdkey

Authenti

sign\_in

Keyword\_PasswordContextInCode:

key

x509c

credential

password

pw

securestring

Keyword\_PasswordContextInXml:

userpass

password

pw

connectionstring

key

credential

token

sas

secret

------------------------------------------------------------------------

# Azure AD Client Secret

Format:

Up to 40-character combination of letters, digits, and special characters

Pattern:

Any combination of

up to 40

a-z (not case-sensitive)

0-9

dashes (-)

underlines (\_)

dots (.)

or tilde accents (~)

e.g.

abc7Q~defghijklmnopqrs0t123456789-\_.~

Samples:

"AppId=6976bfbe-616b-403d-aa0a-1265677ef31c;AppSecret="aEZ7Q~l6DpVLZsyoEGxDee-wbiWwL-lU7W6-y"

Checksum:

Yes

Description:

This SIT "Azure AD Client Secret" is designed to match the security information that's used to secure Azure Active Directory service principals (https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/service-accounts-principal). To do this, it uses several primary resources:

Patterns of Client Secret with specific format.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_AppSecret:

secret

password

key

------------------------------------------------------------------------

# Azure Bot Service App Secret

Format:

Up to 40-character combination of letters, digits, and special characters

Pattern:

Any combination of

up to 40

a-z (not case-sensitive)

0-9

dashes (-)

underlines (\_)

dots (.)

or tilde accents (~)

e.g.

abc7Q~defghijklmnopqrs0t123456789-\_.~

Samples:

"test.azurewebsites.net/api/messages;AppId=6976bfbe-616b-403d-aa0a-1265677ef31c;AppSecret="abcdefghijklmnopqrstuvwxy~-\_0z"

Checksum:

Yes

Description:

This SIT "Azure Bot Service App Secret" is designed to match the security information that's used to establish secure communication between an Azure Bot, WebChat channels and client applications (https://docs.microsoft.com/en-us/azure/bot-service/bot-builder-concept-authentication-types?view=azure-bot-service-4.0). To do this, it uses several primary resources:

Patterns of Client Secret with specific format.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_AppSecret:

secret

password

key

------------------------------------------------------------------------

# Azure Databricks Personal Access Token

Format:

A 32-character combination of letters and digits

Pattern:

Any combination of 32

a-f or A-F (case-sensitive)

or 0-9

e.g.

abcdef0123456789abcdef0123456789

Samples:

ZYXWVU\_1.azuredatabricks.net;PAT=dapiabcdef0123456789abcdef0123456789;

Checksum:

No

Description:

This SIT "Azure Databricks Personal Access Token" is designed to match the security information that's used to authenticate to the Azure Databricks REST API (https://docs.microsoft.com/en-us/azure/databricks/administration-guide/access-control/tokens). To do this, it uses several primary resources:

Patterns of Hex encoded 128 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey128Hex:

dapi

key

secret

token

password

pw

------------------------------------------------------------------------

# Azure Container Registry Access Key

Format:

A 32-character combination of letters, digits, and special characters

Pattern:

Any combination of 32

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

e.g.

abcdefghijklmnopqr0123456789/+AB

Samples:

host.azurecr.io/ #docker password: abcdefghijklmnopqr0123456789/+AB

Checksum:

No

Description:

This SIT "Azure Container Registry Access Key" is designed to match the security information that's used to access Azure Container Registry services as an admin account (https://docs.microsoft.com/en-us/azure/container-registry/container-registry-authentication#admin-account). To do this, it uses several primary resources:

Patterns of Base64 encoded 192 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey192:

password

-p

azurecr

------------------------------------------------------------------------

# Azure Batch Shared Access Key

Format:

A 44-character combination of letters, digits, and special characters

Pattern:

Any combination of 43

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

ends with an equal sign (=)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

Samples:

Account=host.batch.azure.net;AccountKey=abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=;

Checksum:

No

Description:

This SIT "Azure Batch Shared Access Key" is designed to match the security information that's used to access Azure Batch accounts (https://docs.microsoft.com/en-us/azure/batch/security-best-practices). To do this, it uses several primary resources:

Patterns of Base64 encoded 256 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName, Id.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey256:

SharedAccessKey

AccountKey

------------------------------------------------------------------------

# Azure SignalR Access Key

Format:

A 44-character combination of letters, digits, and special characters

Pattern:

Any combination of 43

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

ends with an equal sign (=)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

Samples:

host: host.service.signalr.net; accesskey: abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=;

Checksum:

No

Description:

This SIT "Azure SignalR Access Key" is designed to match the security information that's used to authenticate Azure SignalR clients when requests are made to the service (https://docs.microsoft.com/en-us/azure/azure-signalr/signalr-howto-key-rotation). To do this, it uses several primary resources:

Patterns of Base64 encoded 256 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName, Id.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey256:

SharedAccessKey

AccountKey

------------------------------------------------------------------------

# Azure EventGrid Access Key

Format:

A 44-character combination of letters, digits, and special characters

Pattern:

Any combination of 43

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

ends with an equal sign (=)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

Samples:

host: host.eventgrid.azure.net; accesskey: abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=;

Checksum:

No

Description:

This SIT "Azure EventGrid Access Key" is designed to match the security information that's used to authenticate an application publishing events to Azure Event Grid resources (topics and domains) (https://docs.microsoft.com/en-us/azure/event-grid/get-access-keys). To do this, it uses several primary resources:

Patterns of Base64 encoded 256 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName, Id.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey256:

SharedAccessKey

AccountKey

------------------------------------------------------------------------

# Azure Machine Learning Web Service API Key

Format:

An 88-character combination of letters, digits, and special characters

Pattern:

Any combination of 86

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

ends with two equal signs (=)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEabcdefghijklmnopqrstuvwxyz0123456789/+ABCDE==

Samples:

host: host.azureml.net/services/6976bfbe-616b-403d-aa0a-1265677ef31c/workspaces/6976bfbe-616b-403d-aa0a-1265677ef31c/; apikey: abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEabcdefghijklmnopqrstuvwxyz0123456789/+ABCDE==;

Checksum:

No

Description:

This SIT "Azure Machine Learning Web Service API Key" is designed to match the security information that's used to connect to Azure Machine Learning Web services (https://docs.microsoft.com/en-us/azure/machine-learning/classic/consume-web-services). To do this, it uses several primary resources:

Patterns of Base64 encoded 512 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName, Id, AccountName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey512:

SharedAccessKey

AccountKey

------------------------------------------------------------------------

# Azure Cognitive Search API Key

Format:

A 32-character combination of letters and digits

Pattern:

Any combination of 32

a-f or A-F (case-sensitive)

or 0-9

e.g.

abcdef0123456789abcdef0123456789

Samples:

host: host.search.windows.net; apikey: abcdef0123456789abcdef0123456789;

Checksum:

No

Description:

This SIT "Azure Cognitive Search API Key" is designed to match the security information that's used to authenticate inbound requests to Azure Cognitive Search APIs (https://docs.microsoft.com/en-us/azure/search/search-security-api-keys). To do this, it uses several primary resources:

Patterns of Hex encoded 128 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey128Hex:

dapi

key

secret

token

password

pw

------------------------------------------------------------------------

# Azure Cognitive Service Key

Format:

A 32-character combination of letters and digits

Pattern:

Any combination of 32

a-f or A-F (case-sensitive)

or 0-9

e.g.

abcdef0123456789abcdef0123456789

Samples:

cognitiveservices.azure.com...apikey=abcdef0123456789abcdef0123456789;

api.cognitive.microsoft.com...apikey=abcdef0123456789abcdef0123456789;

Checksum:

No

Description:

This SIT "Azure Cognitive Service Key" is designed to match the security information that's used to authenticate requests to Azure Cognitive Services (https://docs.microsoft.com/en-us/azure/search/search-security-api-keys). To do this, it uses several primary resources:

Patterns of Hex encoded 128 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey128Hex:

dapi

key

secret

token

password

pw

------------------------------------------------------------------------

# Azure Maps Subscription Key

Format:

A 43-character combination of letters, digits, and special characters

Pattern:

Any combination of 43

a-z (not case-sensitive)

0-9

dashes (-)

or underlines (\_)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789-\_ABCDE

Samples:

host: atlas.microsoft.com; key: abcdefghijklmnopqrstuvwxyz0123456789-\_ABCDE;

Checksum:

No

Description:

This SIT "Azure Maps Subscription Key" is designed to match the security information that's used to access resources in Azure Maps accounts (https://docs.microsoft.com/en-us/azure/azure-maps/how-to-manage-authentication). To do this, it uses several primary resources:

Patterns of Base64 URL encoded 256 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey256Url:

key

microsoft.maps

------------------------------------------------------------------------

# Azure Bot Framework Secret Key

Format:

A 55-character combination of letters, digits, and special characters

or

A 63-character combination of letters, digits, and special characters

Pattern:

Any combination of 55

a-z (not case-sensitive)

0-9

underlines (\_)

or dots (.)

e.g.

abcdefghijklmnopqrstuvwxyz.0123456789\_ABCDEabcdefghijkl

or

Any combination of 11

a-z (not case-sensitive)

0-9

dashes (-)

or underlines (\_)

a dot

Any combination of 3

a-z (not case-sensitive)

0-9

dashes (-)

or underlines (\_)

a dot

Any combination of 3

a-z (not case-sensitive)

0-9

dashes (-)

or underlines (\_)

a dot

Any combination of 43

a-z (not case-sensitive)

0-9

dashes (-)

or underlines (\_)

e.g.

abcdefghijk.lmn.opq.rstuvwxyz0123456789-\_ABCDEFGHIJKLMNOPQRSTUV

Samples:

host: webchat.botframework.com/?s=abcdefghijk.lmn.opq.rstuvwxyz0123456789-\_ABCDEFGHIJKLMNOPQRSTUV&

Checksum:

No

Description:

This SIT "Azure Bot Framework Secret Key" is designed to match the security information that's used to connect to WebChat channels from Azure Bot services (https://docs.microsoft.com/en-us/azure/bot-service/bot-service-channel-connect-webchat?view=azure-bot-service-4.0). To do this, it uses several primary resources:

Patterns of Base64 URL encoded 328 bits symmetric key.

Patterns of Base64 URL encoded 360 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey328Url:

botframework

key

Keyword\_SymmetricKey360Url:

botframework

key

------------------------------------------------------------------------

# Client Secret / Api Key

Format:

A client secret or refresh token used in OAuth2.0 protocol

or

A 24-character combination of letters, digits, and special characters

or

A 32-character combination of letters and digits

or

A 40-character combination of letters and digits

or

A 44-character combination of letters, digits, and special characters

or

A 56-character combination of letters, digits, and special characters

or

An 88-character combination of letters, digits, and special characters

Pattern:

Variant client secret or refresh token formats e.g.

ClientSecret:\*\*\*\*\*\*\*\*

AppSecret=\*\*\*\*\*\*\*\*

ConsumerKey:=\*\*\*\*\*\*\*\*

Refresh\_Token:\*\*\*\*\*\*\*\*

or

Any combination of 22

a-z (not case-sensitive)

digits, forward slashes, or plus signs

ends with two equal signs (=)

e.g.

abcdefgh0123456789/+AB==

or

Any combination of 32

a-f or A-F (case-sensitive)

or 0-9

e.g.

abcdef0123456789abcdef0123456789

or

Any combination of 40

a-f or A-F (case-sensitive)

or 0-9

e.g.

abcdef0123456789abcdef0123456789abcdef01

or

Any combination of 43

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

ends with an equal sign (=)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

or

Any combination of 54

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

ends with two equal signs (=)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEFGHIJKLMNOP==

or

Any combination of 86

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

ends with two equal signs (=)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEabcdefghijklmnopqrstuvwxyz0123456789/+ABCDE==

Samples:

client\_secret=abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

ida:password=abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

ida:...issuer...Api...abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

Namespace...ACS...Issuer...abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

IssuerName...IssuerSecret=abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

App\_Secret=abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEabcdefghijklmnopqrstuvwxyz0123456789/+ABCDE==

Checksum:

No

Description:

This SIT "Client Secret / Api Key" is designed to match the security information that's known only to the OAuth application and the authorization server to exchange for an access token at runtime (https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-how-applications-are-added). To do this, it uses several primary resources:

Patterns of Client secret context.

Patterns of Base64 encoded 128 bits symmetric key.

Patterns of Hex encoded 128 bits symmetric key.

Patterns of Hex encoded 160 bits Symmetric Key.

Patterns of Base64 encoded 256 bits symmetric key.

Patterns of Base64 encoded 320 bits symmetric key.

Patterns of Base64 encoded 512 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName, Id, AccountName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_ClientSecretContext:

secret

token

auth

securestring

key

Keyword\_SymmetricKey128:

secret

key

password

pw

Keyword\_SymmetricKey128Hex:

dapi

key

secret

token

password

pw

Keyword\_SymmetricKey160Hex:

token

Keyword\_SymmetricKey256:

SharedAccessKey

AccountKey

Keyword\_SymmetricKey320:

code=

key

Keyword\_SymmetricKey512:

SharedAccessKey

AccountKey

------------------------------------------------------------------------

# General Password

Format:

Up to 20,000-character combination of letters, digits, and special characters

or

A pair of username and password used in URL

or

Login credentials used in command lines

or

Plain-text password used in code snippets

or

Plain-text password used in script

or

Plain-text password used in XML configuration

or

A 24-character combination of letters, digits, and special characters

or

A 32-character combination of letters and digits

or

A 32-character combination of letters, digits, and special characters

or

A 44-character combination of letters, digits, and special characters

or

An 88-character combination of letters, digits, and special characters

Pattern:

Any combination of

up to 20,000

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

Up to 2

equal signs (=)

e.g.

MIIKcQIBAzCCCi0GCSqGSIb3DQEHAaCCCh4EggoaMIIKFjCCBg8GCSqGSIb3DQEHAaCCBgAEggX8MIIF+DCCBfQGCyqGSIb3DQEM…

or

Variant URL username and password formats e.g.

https://username:\*\*\*\*\*\*\*\*@contoso.com/...

ftp://username:\*\*\*\*\*\*\*\*@contoso.com:20/...

...

or

Variant command line login credentials formats e.g.

... -u username:\*\*\*\*\*\*\*\* ...

... -u username -p \*\*\*\*\*\*\*\* ...

... /f ... /p \*\*\*\*\*\*\*\*...

... -Password \*\*\*\*\*\*\*\*...

... -U username%\*\*\*\*\*\*\*\* ...

... -secrets:\*\*\*\*\*\*\*\* ...

...

or

Variant password formats in code snippets e.g.

... new X509Certificates2( ...

... ConvertTo-SecureString -String \*\*\*\*\*\*\*\*...

... password = "\*\*\*\*\*\*\*\*"...

... "password" : "\*\*\*\*\*\*\*\*"...

... UserPasswordCredential( ...

...

or

Variant password formats in script e.g.

... -password "\*\*\*\*\*\*\*\*" ...

... -UserPassword "\*\*\*\*\*\*\*\*" ...

... password = \*\*\*\*\*\*\*\*...

... password: \*\*\*\*\*\*\*\*...

...

or

Variant password formats in XML e.g.

... <secret>\*\*\*\*\*\*\*\*</secret> ...

... <password>\*\*\*\*\*\*\*\*</password> ...

... <setting name="password" value="\*\*\*\*\*\*\*\*" > ...

... <setting name="password">\*\*\*\*\*\*\*\*</setting> ...

... <setting name="password"><value>\*\*\*\*\*\*\*\*</value></setting> ...

...

or

Any combination of 22

a-z (not case-sensitive)

digits, forward slashes, or plus signs

ends with two equal signs (=)

e.g.

abcdefgh0123456789/+AB==

or

Any combination of 32

a-f or A-F (case-sensitive)

or 0-9

e.g.

abcdef0123456789abcdef0123456789

or

Any combination of 32

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

e.g.

abcdefghijklmnopqr0123456789/+AB

or

Any combination of 43

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

ends with an equal sign (=)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

or

Any combination of 86

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

ends with two equal signs (=)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEabcdefghijklmnopqrstuvwxyz0123456789/+ABCDE==

Samples:

UserName=...;Passwpod=abcdefgh0123456789/+AB==;

tool.exe ...-u ... -p..."ZYXWVU\_2"...

<secret>ZYXWVU\_3</secret>

NetworkCredential(..., ZYXWVU\_2)

net use .../u:redmond... /p ZYXWVU\_2

schtasks.../ru ntdev.../rp ZYXWVU\_2

RemoteUserNameParameter:...;;RemotePasswordParameter:\*\*\*;;

Checksum:

Yes

Description:

This SIT "General Password" is designed to match the security information that's like usernames and passwords used in general login process (https://docs.microsoft.com/en-us/azure/key-vault/quick-create-portal). To do this, it uses several primary resources:

Patterns of Base64 encoded string literal.

Patterns of User Login Credentials in URL.

Patterns of Password context in command line.

Patterns of Password context in code.

Patterns of Password context in script.

Patterns of Password context in XML.

Patterns of Base64 encoded 128 bits symmetric key.

Patterns of Hex encoded 128 bits symmetric key.

Patterns of Base64 encoded 192 bits symmetric key.

Patterns of Base64 encoded 256 bits symmetric key.

Patterns of Base64 encoded 512 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName, Id, AccountName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_Base64EncodedStringLiteral:

MII

Keyword\_LoginCredentialsInUrl:

://

Keyword\_PasswordContextInCmdLine:

certutil

zdbg

secret

VSTS\_TOKEN

curl

powershell

ps1

-u

Smc

AutoLogon

ldifde

Rclone

--env

SignTool

winexe

net

Keyword\_PasswordContextInCode:

key

x509c

credential

password

pw

securestring

Keyword\_PasswordContextInScript:

secret

password

pw

Keyword\_PasswordContextInXml:

userpass

password

pw

connectionstring

key

credential

token

sas

secret

Keyword\_SymmetricKey128:

secret

key

password

pw

Keyword\_SymmetricKey128Hex:

dapi

key

secret

token

password

pw

Keyword\_SymmetricKey192:

password

-p

azurecr

Keyword\_SymmetricKey256:

SharedAccessKey

AccountKey

Keyword\_SymmetricKey512:

SharedAccessKey

AccountKey

------------------------------------------------------------------------

# General Symmetric Key

Format:

A 44-character combination of letters, digits, and special characters

or

An 88-character combination of letters, digits, and special characters

Pattern:

Any combination of 43

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

ends with an equal sign (=)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

or

Any combination of 86

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

ends with two equal signs (=)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEabcdefghijklmnopqrstuvwxyz0123456789/+ABCDE==

Samples:

sse-c-key=abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

aes256...startDate...abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

SymmetricKey...abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

password...abcdefghijklmnopqrstuvwxyz0123456789/+ABCDE=

secret...abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEabcdefghijklmnopqrstuvwxyz0123456789/+ABCDE==

Checksum:

No

Description:

This SIT "General Symmetric Key" is designed to match the security information that's used in general authentication process (https://docs.microsoft.com/en-us/dotnet/api/system.security.cryptography.aes?view=net-5.0). To do this, it uses several primary resources:

Patterns of Base64 encoded 256 bits symmetric key.

Patterns of Base64 encoded 512 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName, Id, AccountName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey256:

SharedAccessKey

AccountKey

Keyword\_SymmetricKey512:

SharedAccessKey

AccountKey

------------------------------------------------------------------------

# GitHub Personal Access Token

Format:

A 40-character combination of letters, digits, and special characters

or

A pair of username and password used in URL

or

A 40-character combination of letters and digits

Pattern:

A token prefix (case-sensitive) 'ghp\_', 'gho\_', 'ghu\_', 'ghs\_', or 'ghr\_'

Any combination of 36

a-z (not case-sensitive)

or 0-9

e.g.

ghp\_abcdefghijklmnopqrstuvwxyzABCD012345

or

Variant URL username and password formats e.g.

https://username:\*\*\*\*\*\*\*\*@contoso.com/...

ftp://username:\*\*\*\*\*\*\*\*@contoso.com:20/...

...

or

Any combination of 40

a-f or A-F (case-sensitive)

or 0-9

e.g.

abcdef0123456789abcdef0123456789abcdef01

Samples:

pat=v1.abcdef0123456789abcdef0123456789abcdef01

https://user:abcdef0123456789abcdef0123456789abcdef01@github.com

Checksum:

Yes

Description:

This SIT "GitHub Personal Access Token" is designed to match the security information that's used as an alternate password for authentication to GitHub when using GitHub API or the command line (https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token). To do this, it uses several primary resources:

Patterns of Identifiable GitHub PAT.

Patterns of User Login Credentials in URL.

Patterns of Hex encoded 160 bits Symmetric Key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

Patterns of mockup values, redactions, and placeholders.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_GitHubPatIdentifiableSecret:

gh\_

Keyword\_LoginCredentialsInUrl:

://

Keyword\_SymmetricKey160Hex:

token

------------------------------------------------------------------------

# Google API key

Format:

A 39-character combination of letters, digits, and special characters

Pattern:

A token prefix (case-sensitive) 'AIza'

Any combination of 35

a-z (not case-sensitive)

0-9

dashes (-)

underlines (\_)

or backward slashes (\)

e.g.

AIzaefgh0123456789\_-ABCDEFGHIJKLMNOPQRS

Samples:

apiKey=AIzaefgh0123456789\_-ABCDEFGHIJKLMNOPQRS;

Checksum:

No

Description:

This SIT "Google API key" is designed to match the security information that's used as a simple encrypted string that identifies a Google REST API client without any principal that are used to associate API requests with your project for quota and billing (https://cloud.google.com/docs/authentication/api-keys). To do this, it uses several primary resources:

Patterns of Base64 encoded 210 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey210:

AIza

------------------------------------------------------------------------

# Http Authorization Header

Format:

An authorization header used in HTTP request

Pattern:

Variant authentication header formats e.g.

authorization: basic \*\*\*\*\*\*\*\*

authorization: bearer \*\*\*\*\*\*\*\*

authorization: digest \*\*\*\*\*\*\*\*

authorization: negotiate \*\*\*\*\*\*\*\*

...

Samples:

Authorization: Basic ABCDEFGHIJKLMNOPQRS0123456789...;

Authorization: Digest ABCDEFGHIJKLMNOPQRS0123456789...;

Checksum:

No

Description:

This SIT "Http Authorization Header" is designed to match the security information that's used in the header of an HTTP request for authentication and authorization (https://docs.microsoft.com/en-us/dotnet/api/system.net.http.headers.httprequestheaders.authorization?view=netframework-4.8). To do this, it uses several primary resources:

Patterns of Http authorization header.

Patterns of CredentialName, CredentialFeatures, ResourceType.

Patterns of mockup values, redactions, and placeholders.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_HttpAuthorizationHeader:

authorization

------------------------------------------------------------------------

# Microsoft Bing Maps Key

Format:

A 64-character combination of letters, digits, and special characters

Pattern:

Any combination of 64

a-z (not case-sensitive)

0-9

underlines (\_)

or dashes (-)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789-\_ABCDEabcdefghijklmnopqrstu

Samples:

bingMapsKey=abcdefghijklmnopqrstuvwxyz0123456789-\_ABCDEabcdefghijklmnopqrstu

...bing.com/api/maps/...key=abcdefghijklmnopqrstuvwxyz0123456789-\_ABCDEabcdefghijklmnopqrstu

...dev.virtualearth.net/...key=abcdefghijklmnopqrstuvwxyz0123456789-\_ABCDEabcdefghijklmnopqrstu

Checksum:

No

Description:

This SIT "Microsoft Bing Maps Key" is designed to match the security information that's used to call Bing Maps API (https://docs.microsoft.com/en-us/bingmaps/getting-started/bing-maps-dev-center-help/getting-a-bing-maps-key). To do this, it uses several primary resources:

Patterns of Base64 URL encoded 384 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SymmetricKey384Url:

virtualearth

api/maps

key

------------------------------------------------------------------------

# Slack Access Token

Format:

Up to 34-character combination of letters, digits, and special characters

Pattern:

A token prefix (case-sensitive) 'xoxp-', 'xoxb-', 'xoxa-', 'xoxr-', 'xoxo-', 'xoxs-' or 'xoxe-'

Any combination of

up to 29 a-z (not case-sensitive)

0-9

or dashes (-)

e.g.

xoxp-abcdef-abcdef-abcdef-abcdef

Samples:

slack\_token=xoxp-abcdef-abcdef-abcdef-abcd

slack\_token=xoxb-abcdef-abcdef

slack\_token=xoxa-2-abcdef-abcdef-abcdef-abcdef

slack\_token=xoxr-abcdef-abcdef-abcdef-abcdef

Checksum:

No

Description:

This SIT "Slack Access Token" is designed to match the security information that's used to access Slack platform functionalities (e.g. Bot tokens, User tokens and App-level tokens) (https://api.slack.com/docs/token-types). To do this, it uses several primary resources:

Patterns of Slack user/bot/workspace token.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

Patterns of mockup values, redactions, and placeholders.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_SlackTokens:

xox

---------------------------------------------------------------------------------------

# User Login Credentials

Format:

A pair of username and password used in general authentication process

or

A pair of username and password used in PuTTY connection manager

or

Plain-text password used in code snippets

or

An 88-character combination of letters, digits, and special characters

Pattern:

Variant username and password formats e.g.

username=...;password=\*\*\*\*\*\*\*\*;

user id=...;password=\*\*\*\*\*\*\*\*;

uid=...;pwd=\*\*\*\*\*\*\*\*;

DB\_USER=...;DB\_PASS=\*\*\*\*\*\*\*\*;

Service Account=...;Password=\*\*\*\*\*\*\*\*;

...

or

An XML element <login>

An embeded XML element <login>

Inner XML content

An embeded XML element </login>

An embeded XML element <password>

Inner XML content

An embeded XML element </password>

An XML element </login>

e.g.

<login> <login>ZYXWVU\_1</login> <password>ZY…

or

Variant password formats in code snippets e.g.

... new X509Certificates2( ...

... ConvertTo-SecureString -String \*\*\*\*\*\*\*\*...

... password = "\*\*\*\*\*\*\*\*"...

... "password" : "\*\*\*\*\*\*\*\*"...

... UserPasswordCredential( ...

...

or

Any combination of 86

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

ends with two equal signs (=)

e.g.

abcdefghijklmnopqrstuvwxyz0123456789/+ABCDEabcdefghijklmnopqrstuvwxyz0123456789/+ABCDE==

Samples:

{ "user": "...", "password": "ZYXWVU\_2" }

Checksum:

No

Description:

This SIT "User Login Credentials" is designed to match the security information that's used in general user login process (https://docs.microsoft.com/en-us/azure/key-vault/quick-create-portal). To do this, it uses several primary resources:

Patterns of Plain-text username and password.

Patterns of Plain-text username and password in PuTTYcm database file.

Patterns of Password context in code.

Patterns of Base64 encoded 512 bits symmetric key.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName, Id, AccountName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_LoginCredentials:

password

pw

DB\_

Keyword\_LoginCredentialsPutty:

login

Keyword\_PasswordContextInCode:

key

x509c

credential

password

pw

securestring

Keyword\_SymmetricKey512:

SharedAccessKey

AccountKey

------------------------------------------------------------------------

# X.509 Certificate Private Key

Format:

Up to 20,000-character combination of letters, digits, and special characters

or

Up to 40-character combination of upper-case letters, space, and dashes

Pattern:

Any combination of

up to 20,000

a-z (not case-sensitive)

0-9

forward slashes (/)

or plus signs (+)

Up to 2

equal signs (=)

e.g.

MIIKcQIBAzCCCi0GCSqGSIb3DQEHAaCCCh4EggoaMIIKFjCCBg8GCSqGSIb3DQEHAaCCBgAEggX8MIIF+DCCBfQGCyqGSIb3DQEM…

or

5 dashes (-)

Any combination of

up to 30

A-Z (case-sensitive)

spaces

5 dashes (-)

e.g.

-----BEGIN PRIVATE KEY-----

Samples:

���������������� (binary certificate file: \*.pfx, \*.key...)

-----BEGIN PRIVATE KEY----- MIIPuQIBAzCCD38GCSqGSIb3DQEH...

-----BEGIN RSA PRIVATE KEY----- ��������������� ...

-----BEGIN DSA PRIVATE KEY----- MIIPuQIBAzCCD38GCSqGSIb3DQEH...

-----BEGIN EC PRIVATE KEY----- ��������������� ...

-----BEGIN OPENSSH PRIVATE KEY----- MIIPuQIBAzCCD38GCSqGSIb3DQEH...

certificate = "MIIPuQIBAzCCD38GCSqGSIb3DQEH..."

Checksum:

Yes

Description:

This SIT "X.509 Certificate Private Key" is designed to match the security information that's used as a private component in SSL certificates (https://docs.microsoft.com/en-us/azure/key-vault/certificate-scenarios). To do this, it uses several primary resources:

Patterns of Base64 encoded string literal.

Patterns of Certificate private key header.

Patterns of CredentialName, CredentialFeatures, AccountIdentityName, AccountIdentityValue, ResourceType, ResourceName.

Patterns of mockup values, redactions, and placeholders.

A dictionary of vocabulary.

The patterns are designed to match credentials with reasonable confidence, not credentials format examples. So mockup values, redacted values, and placeholders (e.g. credential type or usage descriptions) in the position where a secret value should present will not be matched.

Keywords:

Keyword\_Base64EncodedStringLiteral:

MII

Keyword\_CertificatePrivateKeyHeader:

key

------------------------------------------------------------------------