SecuGen

SecuGen WebAPI1toN Programmer's Manual

SDK version 1.1

Contents

1. Introduction

SecuGen WebAPI1:N is a combination of two of applications from SecuGen. WebAPI1:N is used to capture fingerprints with the ability to store the template into both an in memory and a typical database. SecuSearch, the desktop product, has a .NET framework interface, that has been altered to allow a .NET core interface to be used to as the in memory database. So, the combination of these two products allow for web sites to be able to do a 1:N search to identify a single user with no other identification qualifiers.

WebAPI1:N is built on the current .NET core technologies of Microsoft. Therefore, the "views" code is mostly a javascript based coding, with C# code written at the server side to manage the template handling and web site overhead.

The enrolment process is mostly a "capture fingerprint" call that will cause the fingerprint sensor to capture an image from the user. The returned data of the capture process can the provide the application with both user image and user template to be stored at the server side as enrolment data.

For the verification of the user, a similar "capture fingerprint" call is made. Then, WebAPI1:N provides two ways on which to verify the user. Firstly, if the user has provided some form of identification, then the application can perform a simple client side 1 to 1 matching to determine user verification. Secondly, the application can do a server side search of the database, and find the single user with no other identifications to allow a 1:N search.

Main Features

- Easy to integrate fingerprint capture, enrollment, and matching functions
- Works with most modern web browsers
- Supports JavaScript
- Utilizes RESTful web service and JSON objects
- No Java runtimes needed
- No browser plug-ins needed
- Small client software
- SecuSearch libraries installed on the server side

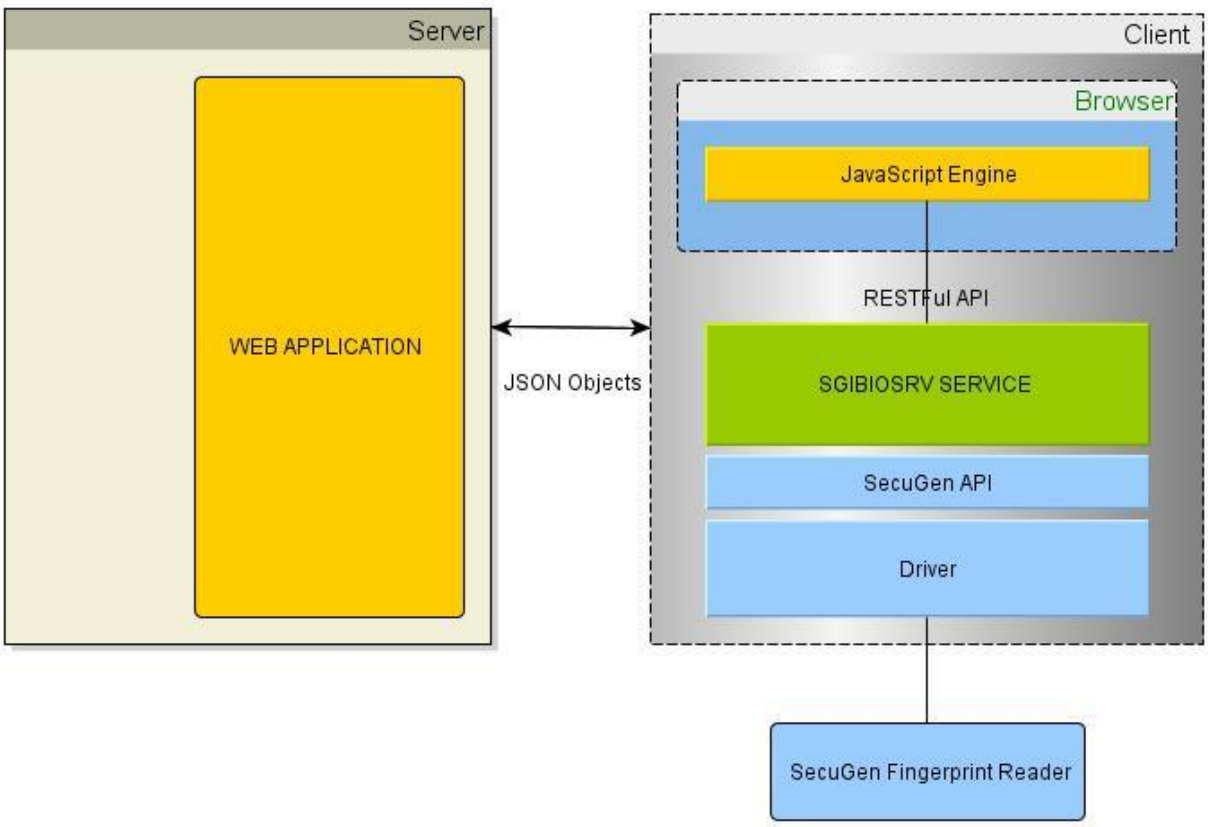
Fingerprint Functions Provided

SecuGen WebAPI provides simple web service calls to the WebAPI Client application to capture fingerprint data and create a fingerprint template in a single method.

- Capture – single finger
- Capture and enroll – single finger
- Client side 1:1 Match
- Server side 1:N Match
- Server side search results that include a candidate result

License Requirements

A license key for each domain that hosts your web application will be needed. If no license key is installed, the web service will work for a limited period of 60 days. Please contact your SecuGen Representative for information about licensing and pricing.



For the SecuSearch in memory database, there is 1,000 templates that can be stored in the database for free. Once that threshold is reached, then a SecuSearch license will be required to place any more templates in the database. In order to get a license, a utility will be needed from the web server

2. Installation and Requirements

Client Side Installation:

System Requirements

- Windows 10 or later, 32-bit or 64-bit
- Memory 4 GB minimum

Supported SecuGen Fingerprint Readers

- Hamster Air (HU-AIR)
- Hamster Pro (HUPx)
- Hamster Pro 10 (HU10)
- Hamster Pro 20 (HU20-A, HU20-AP, HU20)
- Hamster Pro 30 (HU30)
- Hamster IV (HSDU04P)
- Hamster Plus (HSDU03P)

Drivers for Fingerprint Reader

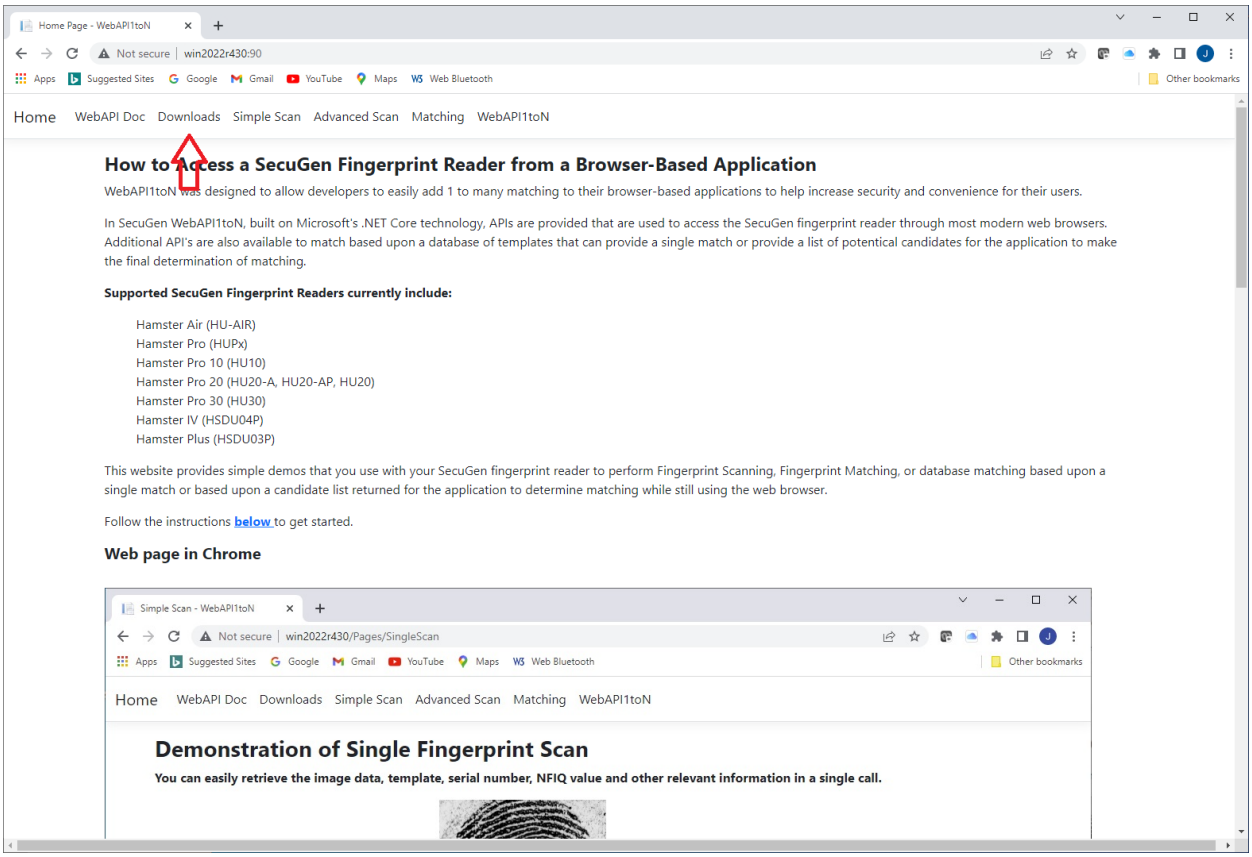
This product is built on top of the drivers for the supported SecuGen fingerprint readers listed above. It is recommended that the latest driver be installed. The drivers can be installed by one of two ways:

(A) Plug in the SecuGen fingerprint reader and the driver will automatically download and install via Windows Update, or

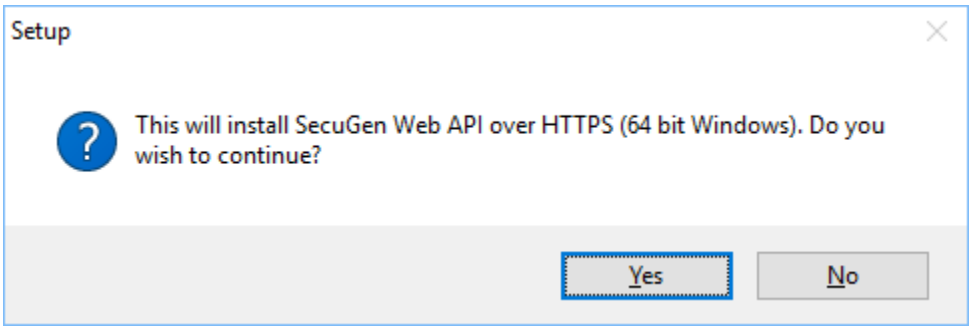
(B) Go to https://secugen.com/download, download and manually install the latest WBF driver.

WebAPI1toN Client Application

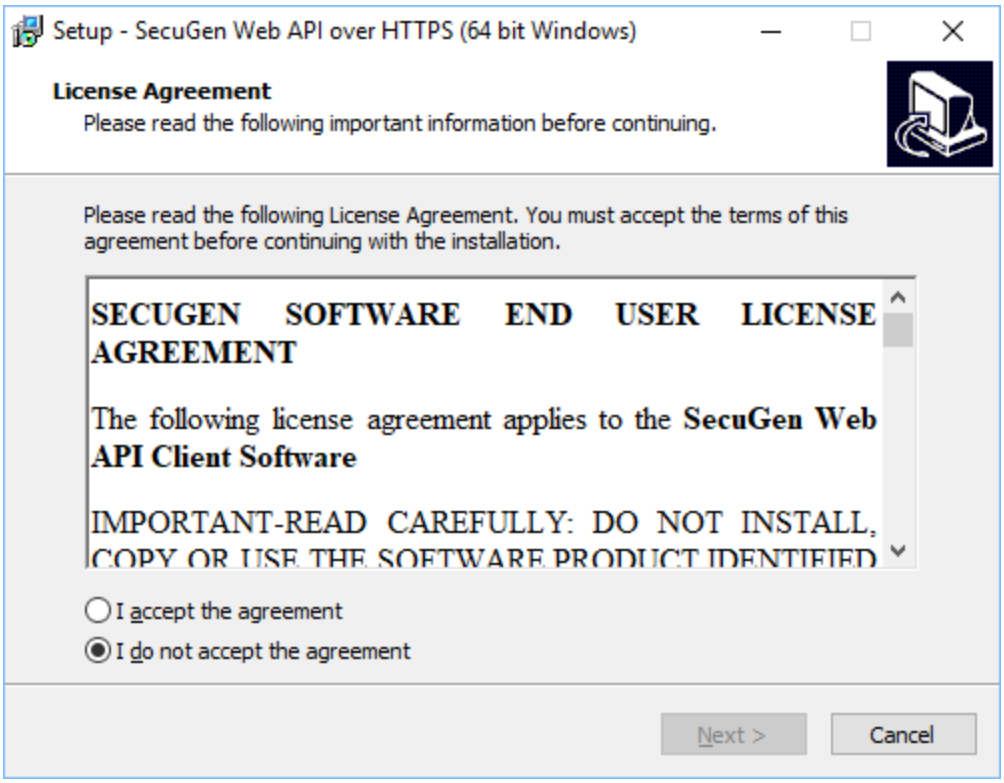
To download the WebAPI client application (SgiBioSrv), go to: https://WebAPI1toN.secugen.com/ and click on the appropriate “Downloads” link for the 32-bit or 64-bit SSL client.



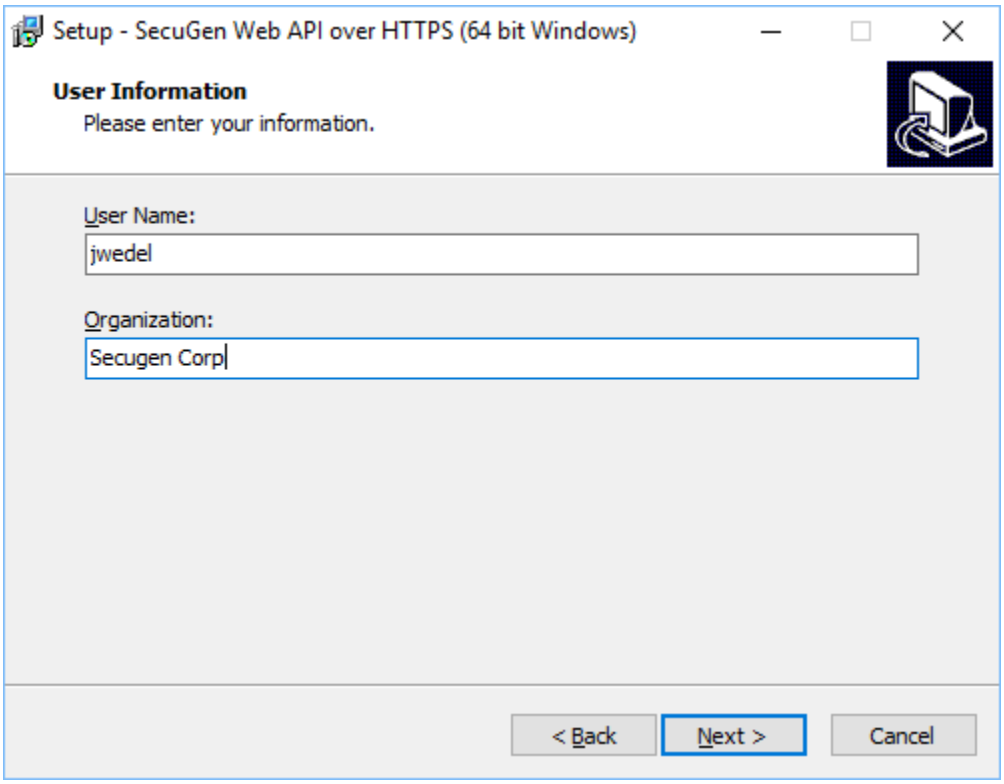
Run the downloaded zip file.



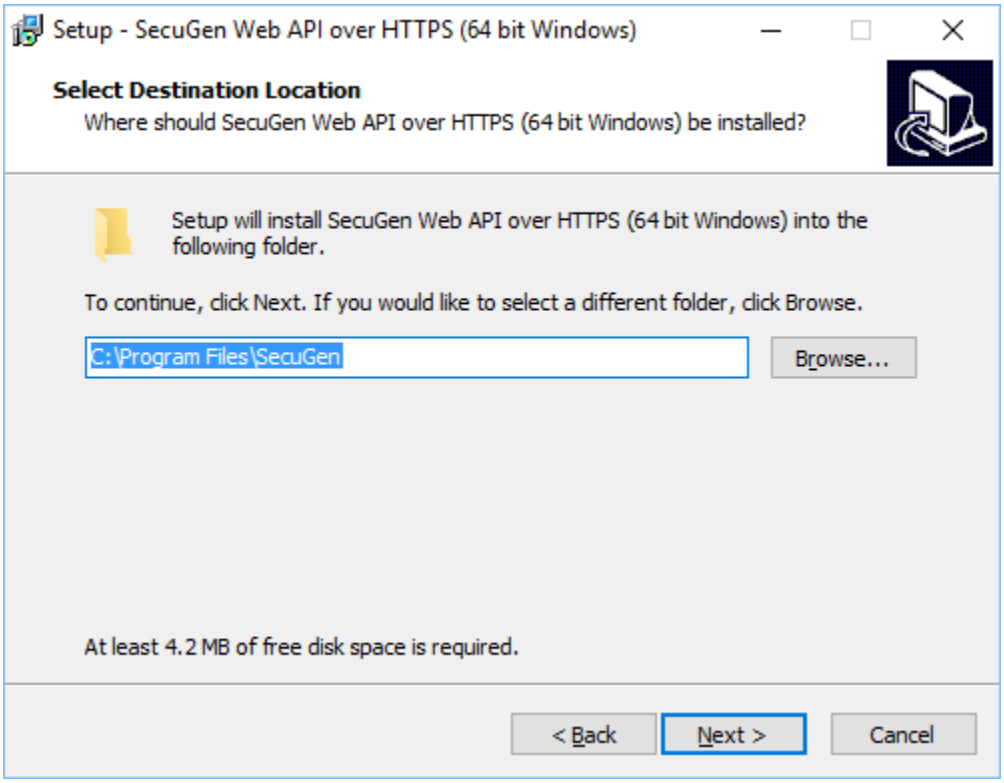
Click Yes to continue with installation.



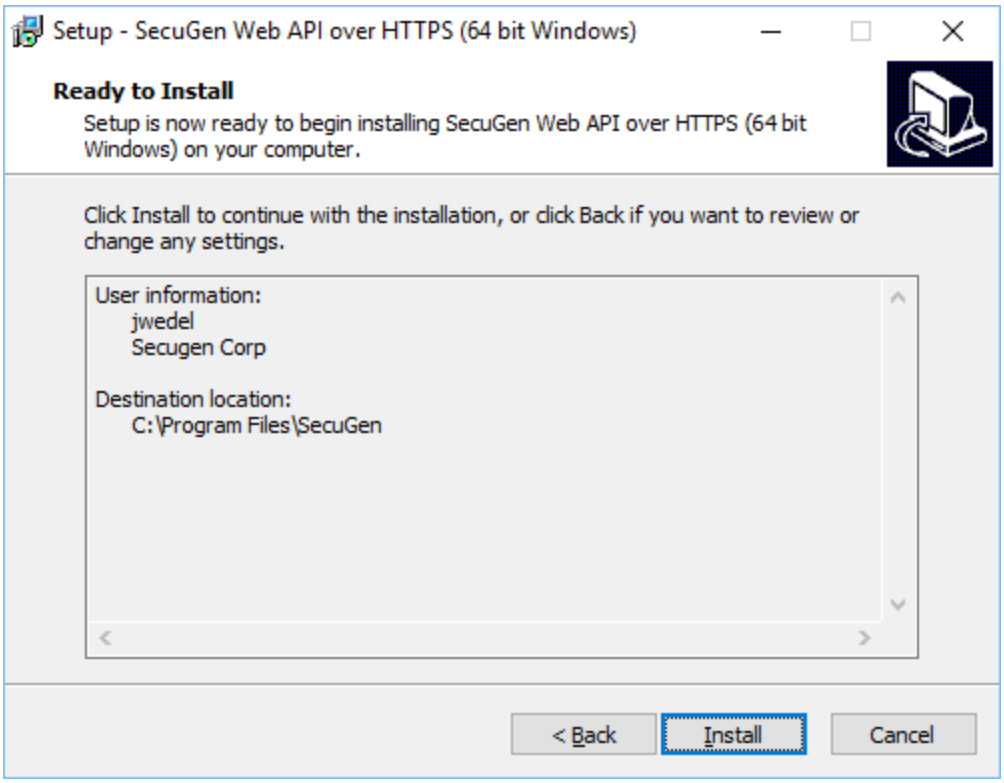
Click “I accept the agreement” and click Next.



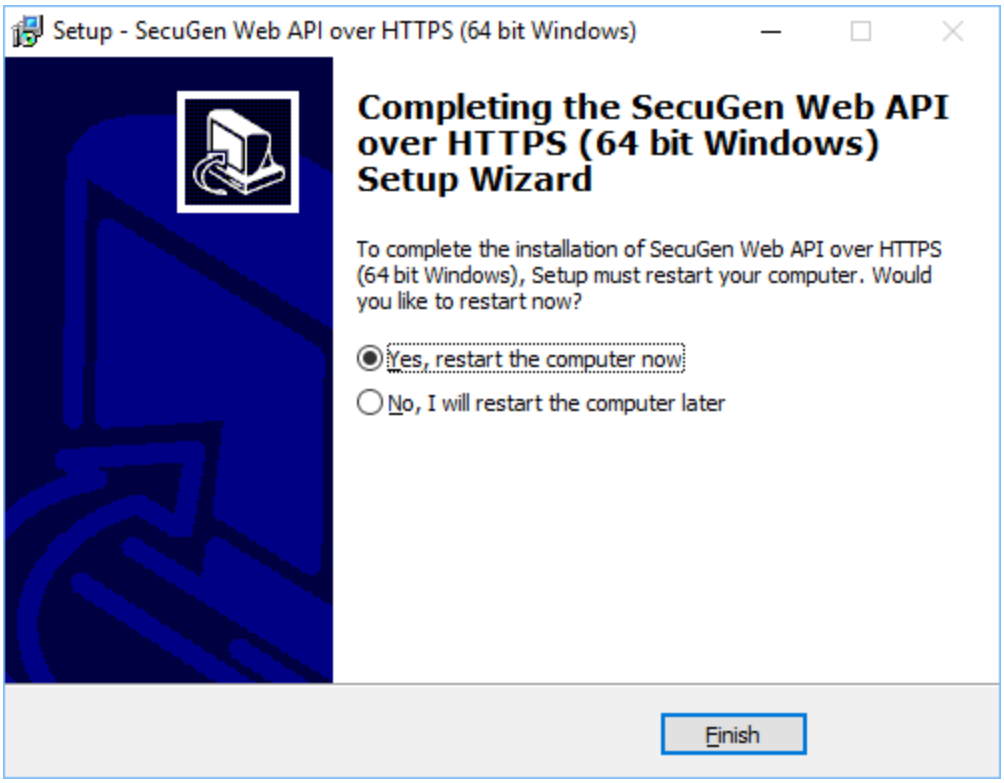
Fill in the User Name field along with the appropriate Organization. The User Name field is prepopulated with the name of the current Window user. Click Next.



A default location is listed but can be changed if desired. Click Next.



Review the summary information and click Install to continue.

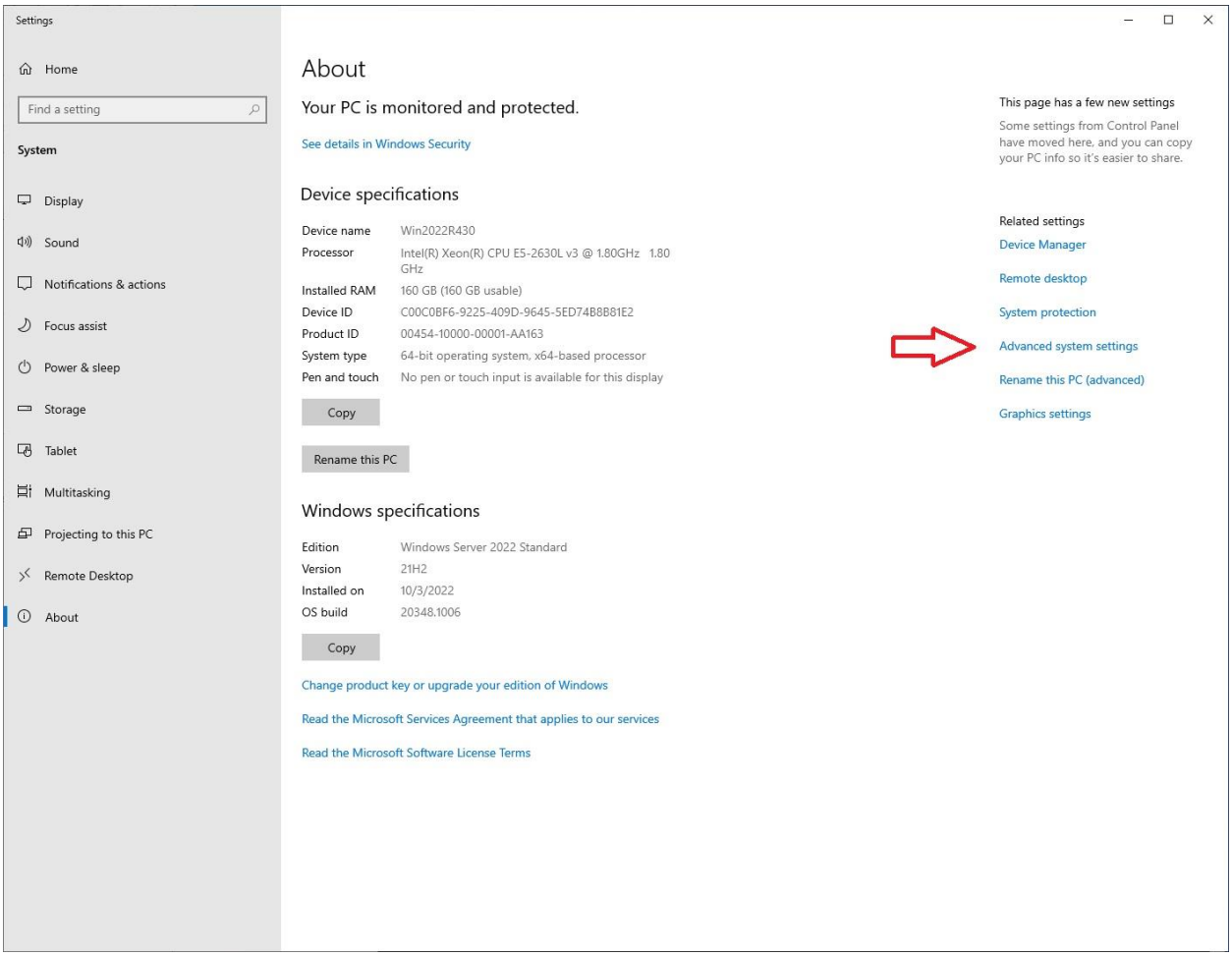


Select Yes and click Finish to complete the installation.

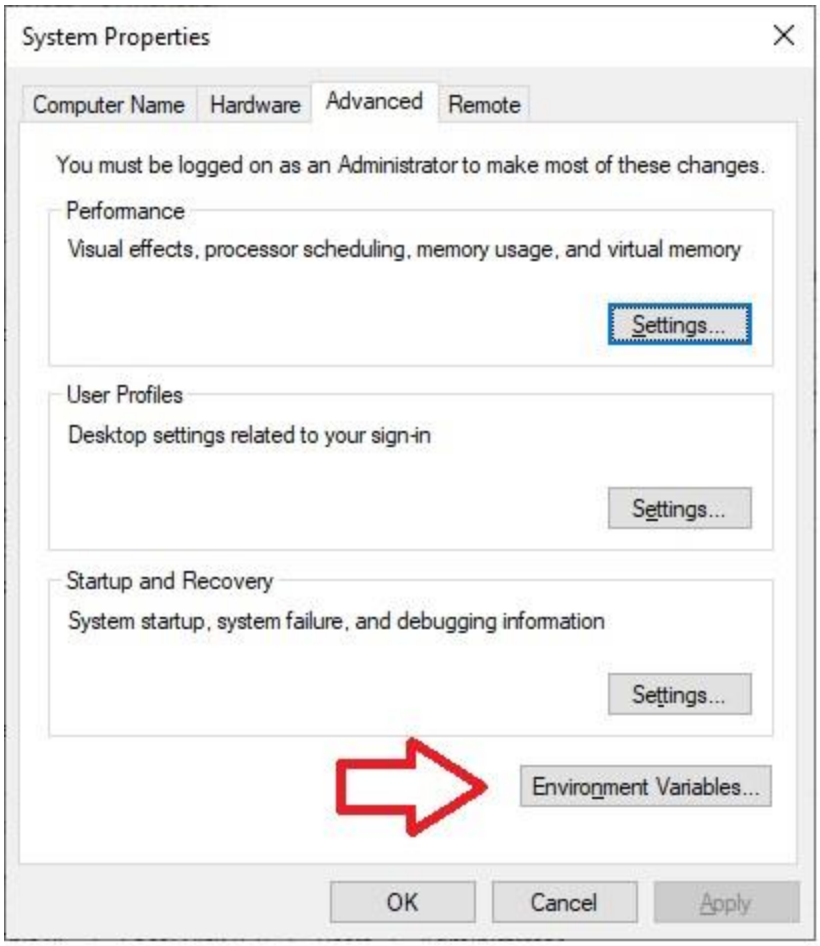
Server Side Installation:

The libraries for SecuSearch must be in the Windows Path in order for any custom web site to pick up and use the SecuSearch engine:

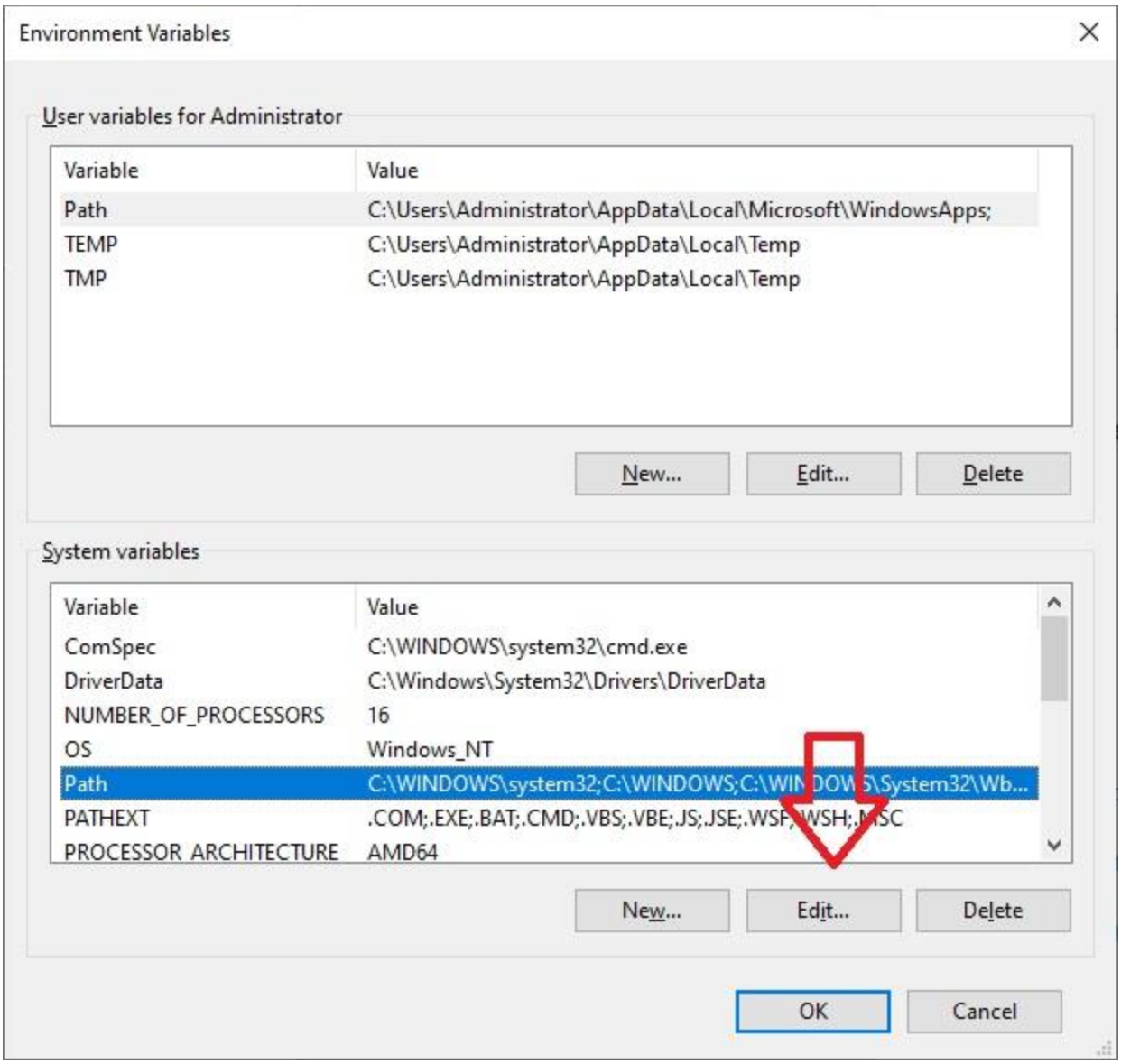
1. To add a specific directory to the Windows path, go to the Settings window, then click on the 'Advanced system settings'.



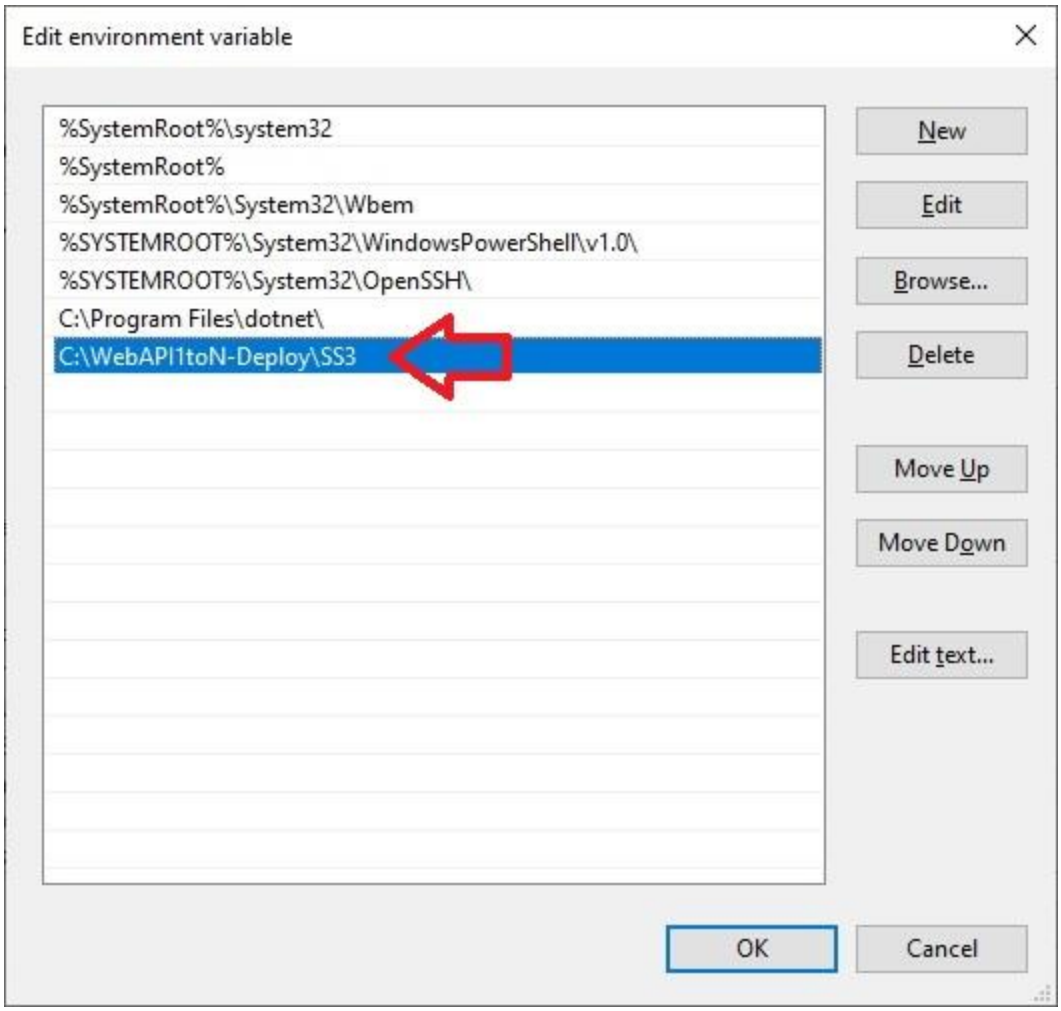
2. Go to the 'Advanced' tab and click on Environment Variables button:



3. On the 'System variables' section, find and highlight the 'Path' variable, next click on 'Edit' button.

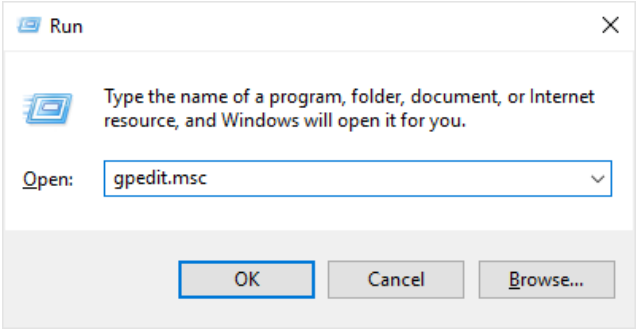


4. Add the appropriate path on where the SecuSearch libraries are going to be placed on the Web Server.

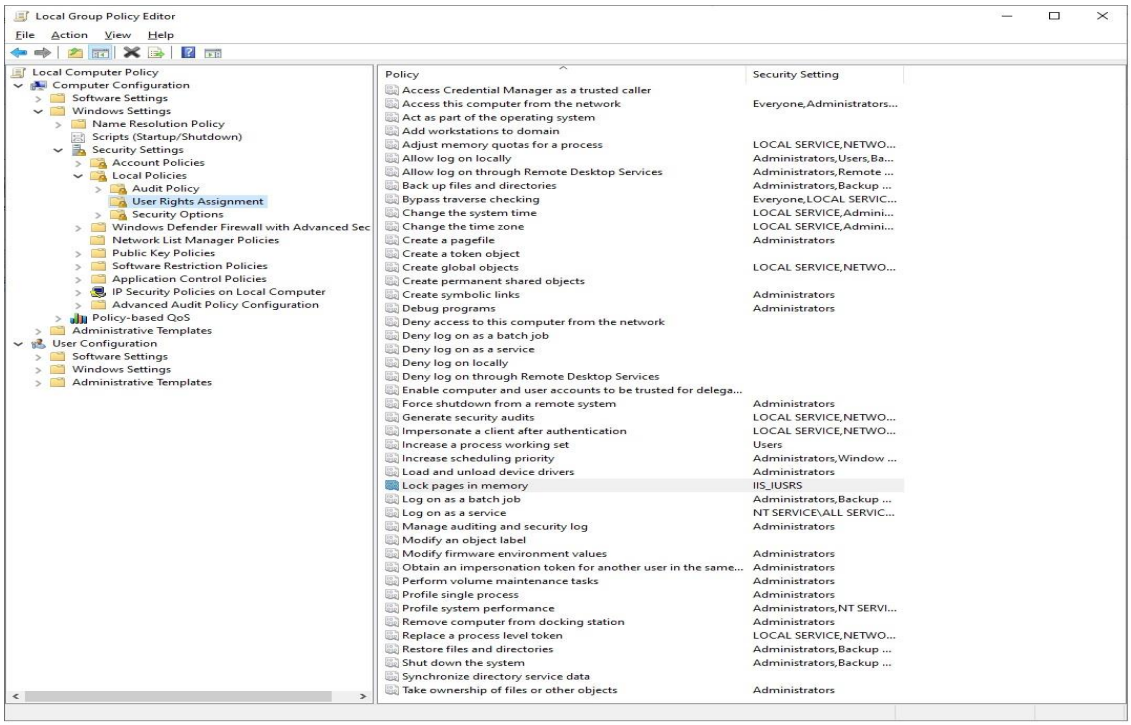


The SecuSearch engine requires contiguous memory to be locked to function correctly. The Windows system policy must be configured as shown.

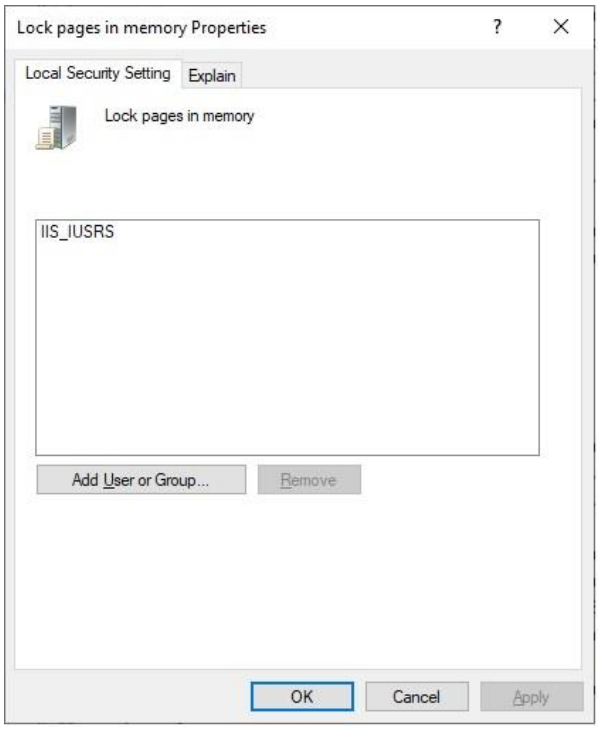
1. Launch ***gpedit.msc***.



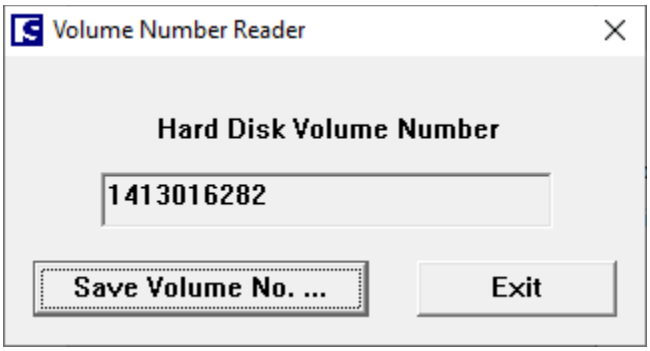
2. Navigate to ***Computer Configuration-> Windows Settings -> Security Settings -> User Rights Assignment***.
3. Double-click "***Lock pages in memory***" and add the user or group that will be running the SecuSearch Engine.


Local Group Policy Editor
File Action View Help
Local Computer Policy
Computer Configuration
Software Settings
Windows Settings
Name Resolution Policy
Scripts (Startup/Shutdown)
Security Settings
Account Policies
Local Policies
Audit Policy
User Rights Assignment
Security Options
Windows Defender Firewall with Advanced Sec
Network List Manager Policies
Public Key Policies
Software Restriction Policies
Application Control Policies
IP Security Policies on Local Computer
Advanced Audit Policy Configuration
Policy-based QoS
Administrative Templates
User Configuration
Software Settings
Windows Settings
Administrative Templates
Policy
Security Setting
Access Credential Manager as a trusted caller
Access this computer from the network
Everyone,Administrators...
Act as part of the operating system
Add workstations to domain
Adjust memory quotas for a process
LOCAL SERVICE,NETWO...
Allow log on locally
Administrators,Users,Ba...
Allow log on through Remote Desktop Services
Administrators,Remote ...
Back up files and directories
Administrators,Backup ...
Bypass traverse checking
Everyone,LOCAL SERVIC...
Change the system time
LOCAL SERVICE,Admini...
Change the time zone
LOCAL SERVICE,Admini...
Create a pagefile
Administrators
Create a token object
Create global objects
LOCAL SERVICE,NETWO...
Create permanent shared objects
Create symbolic links
Administrators
Debug programs
Administrators
Deny access to this computer from the network
Deny log on as a batch job
Deny log on as a service
Deny log on locally
Deny log on through Remote Desktop Services
Enable computer and user accounts to be trusted for delega...
Force shutdown from a remote system
Administrators
Generate security audits
LOCAL SERVICE,NETWO...
Impersonate a client after authentication
LOCAL SERVICE,NETWO...
Increase a process working set
Users
Increase scheduling priority
Administrators,Window ...
Load and unload device drivers
Administrators
Lock pages in memory
IIS_IUSRS
Log on as a batch job
Administrators,Backup ...
Log on as a service
NT SERVICE\ALL SERVIC...
Manage auditing and security log
Administrators
Modify an object label
Modify firmware environment values
Administrators
Obtain an impersonation token for another user in the same...
Administrators
Perform volume maintenance tasks
Administrators
Profile single process
Administrators
Profile system performance
Administrators,NT SERVI...
Remove computer from docking station
Administrators
Replace a process level token
LOCAL SERVICE,NETWO...
Restore files and directories
Administrators,Backup ...
Shut down the system
Administrators,Backup ...
Synchronize directory service data
Take ownership of files or other objects
Administrators

4. Add the user or group that will be running the SecuSearch application by selecting "***Add User or Group…***". Only users with this privilege can start the SecuSearch engine.



5. Installing the SecuSearch license (for more then 1,000 template storage)
   In windows explorer, navigate to the SS3\MakeLicense. Run the utility: VolNoReader.exe.



   "Save Volume No." down to disk, and send the file to your SecuGen representative.

   They will return a license.dat file back to you. Drop this file into a location, that can be referenced by the web server. Take the complete path + "license.dat" and drop that complete string into the initialization call of secusearch.

```
public SystemSecuSearch3()
{
    _SS3 = new SecuSearch();
    SSParam param = new SSParam();
    param.CandidateCount = 10;
```

```
param.Concurrency = 0;
param.LicenseFile = "";
param.EnableRotation = true;
_SSParam = param;

_SS3.InitializeEngine(param);
   }
```

6. Reboot the web server.

3. SGIFPCapture

The SGIFPCapture service returns fingerprint data, details of the fingerprint reader, and the extracted template to the caller as a JSON object.

Parameters

The following table describes the parameters and their permissible values that can be passed to the service. All the parameters are optional, and their defaults are described below.

Name	Type	Description
Licstr	STRING	This is the license key provided for the domain. If not provided, the Web server will work for only a limited period of 60 days.
FakeDetection	INTEGER	Specifies the level of fake finger detection. At higher levels of fake finger detection, it becomes more difficult for users to successfully authenticate. Not all SecuGen fingerprint devices support all levels of fake finger detection, and some devices don't support any level of fake finger detection. Default value is 0 (OFF).
Timeout	INTEGER	Specifies the timeout in milliseconds to wait for the sensing operation to complete. If fingerprint image is not captured within this time (i.e. user does not place the finger on the reader), then an error is generated. Default value is 10000.
TemplateFormat	STRING	Extractor template format is specified in this parameter. Currently supported values are "ISO" and "ANSI". Default is "ISO".
ImageWSQRate	FLOAT	Wavelet Scalar Quantization specification compression ratio. 2.25=WSQ_BITRATE_5_TO_1; 0.75=WSQ_BITRATE_15_TO_1; 0=No compression (WSQImage is not returned)

JSON Object Returned

The web service returns a JSON formatted object that contains the following fields.

Name	Type	Description
ErrorCode	INTEGER	Integer value indicating error if any. Value of 0 indicates no error. Non-zero error code indicates various errors described in this document elsewhere. ***You must check this value before accessing other fields of JSON object. If this is not 0, then other fields are NULL or undefined.***
Manufacturer	STRING	SecuGen
Model	STRING	String indicating model of fingerprint reader
SerialNumber	STRING	String containing the unique serial number of the connected reader
ImageWidth	INTEGER	Integer value indicating the width of the fingerprint image in pixels
ImageHeight	INTEGER	Integer value indicating the height of the fingerprint image in pixels
ImageDPI	INTEGER	Integer value indicating resolution of the fingerprint image in dots per inch.

Name	Type	Description
ImageQuality	INTEGER	Integer value indicating the quality of the image captured. It will always be equal to or higher than the quality parameter passed as argument.
NFIQ	INTEGER	Integer value, NIST Finger Image Quality number from 1 – 5, where 1 is best and 5 is worst
ImageDataBase64	STRING	String containing actual raw image data encoded as a base64 string
BMPBase64	STRING	String value fingerprint image in .BMP format encoded as a base64 string, useful for display in browser using data element in image tag
TemplateBase64	STRING	String value containing fingerprint template encoded as a base64 format. This value is encrypted if called with session key parameter.
WSQImageSize	INTEGER	WSQ Image size returned in bytes
WSQImage	STRING	String containing encoded Wavelet Scalar Image compressed into a base64 string

5. SGIMatchScore

The SGIMatchScore service takes two templates as input, compares them with each other, and returns a matching score back to the calling application. This web method is considered to be a 1 to 1 matching call.

Parameters

The following table describes the parameters and their permissible values that can be passed to the service. All the parameters are optional, and their defaults are described below.

Name	Type	Description
Licstr	STRING	This is the license key provided for the domain. If not provided, the Web server will work for only a limited period of 60 days.
Template1	STRING	String value containing template encoded as base 64 format. This value is encrypted if called with session key parameter.
Template2	STRING	String value containing template encoded as base 64 format. This value is encrypted if called with session key parameter.
TemplateFormat	STRING	Extractor template format is specified in this parameter. Currently supported values are “ISO” and “ANSI”. Default is “ISO”.

JSON Object Returned

The web service returns a JSON formatted object that contains the following fields.

Name	Type	Description
ErrorCode	INTEGER	Integer value indicating error if any. Value of 0 indicates no error. Non-zero error code indicates various errors described in this document elsewhere. ***You must check this value before accessing other fields of JSON object. If this is not 0, then other fields are NULL or undefined.***
MatchingScore	INTEGER	Integer value indicating the matching score of the 2 templates previously captured. Matching score is 0 – 199, with 199 being the very close to an identical match.

6. Integration of SecuSearch into a web based application

SecuSearch was a designed to be a parallel product to a typical database. SecuSearch is meant to traverse the stored templates, and seeks out the single match or return a candidate list back to the calling application. It does require the in memory database to be saved to disk periodically. This avoids the catastrophic events that require a complete rebuild of the in memory database.

SecuSearch administration:

SSError = InitializeEngine(SSParam)

Initialize engine with an already populated structure.

Name	Type	Description
Param	SSParam	Structure that identifies search engine characteristics: number of candidate's to return, Concurrency, LicenseFile location, EnableRotation {true/false}.

SSError = TerminateEngine()

Ends current processing, allowing SecuSearch engines to deallocate memory and unload libraries.

SSError = LoadFPDB(FileName)

Load fingerprint in memory database from the path and file name from parameter. Example: Fingerprint.tbd

Name	Type	Description
FileName	String	Null terminated path plus file name of the template database on which to load.

SSError = SaveFPDB(FileName)

Save fingerprint in memory database from the path and file name from parameter. Example: Fingerprint.tbd

Name	Type	Description
FileName	String	Null terminated path plus file name of the template database on which to save.

SSError = ClearFPDB()

Removes all previously registered fingerprint templates from the SecuSearch DB.

IntPtr = GetVersion()

Returns a 4 byte pointer to an string representing the version of SecuSearch.

SSError = GetFPCount(count)

Get a count of the number of templates that are stored in the in memory database.

Name	Type	Description
Count	UInt64	Reference variable that will contain the total number of templates that are currently stored within the in-memory database.

SSError = GetEngineParam(SSParam)

Returns the engine params of the in memory database.

Name	Type	Description
Param	SSParam	Reference structure that will contain the parameters of the in memory database.

SSError = GetIDList(idList, fpCount, count)

Gets the number of views within the template that is passed in.

Name	Type	Description
idList	Uint32[]	Reference list that will be returned of Id's within the in memory database.
fpCount	Int32	Count of elements in idList.
count	Int32	Reference to count of templates in in memory database.

Template API's:

SSError = GetNumberOfView(template, templateType, numberOfView)

Gets the number of views within the template that is passed in.

Name	Type	Description
template	Byte[]	Incoming template that will be analyized.
templateType	SSTemplateType	Enumeration of either ANSI 378 or ISO19794.
numberOfView	UInt32	Returns the total number of views that are associated with the incoming template.

SSError = ExtractTemplate(template, templateType, indexOfView, sgTemplate)

Extracts a single view of template into a SG400 template that has only one view.

Name	Type	Description
template	Byte[]	Incoming template to be converted.
templateType	SSTemplateType	Enumeration of either ANSI378 or ISO19794.
indexOfView	UInt32	View number of the template to be extracted.
sgTemplate	Byte[]	Reference to byte array to received the converted template.

SSError = GetTemplate(templateId, sgTemplate)

Retrieves the template that is stored in the in memory database.

Name	Type	Description
templateId	UInt32	TemplateId to be retrieved.
template	Byte[]	Template data to be received.

Enrolment/Registration:

As users are enrolled in and temples are sent up to the server for storage and further processing, they must be entered into the SecuSearch product. The act of registering a single template is somewhat straightforward and can be done by calling the SecuSearch API – RegisterFP -. Additionally, SecuSearch has the ability to enter into the in memory database, a batch of fingerprints, -- RegisterFPBatch --.

SSError = RegisterFP(sgTemplate, templateId)

Register a template into the in memory database.

Name	Type	Description
sgTemplate	Byte[]	Template in the SG400 format that will be sent into SecuSearch for storage.
templateId	UInt32	Specifies the integer id of the incoming template. This is used as an identification of the template during searches.

SSError = RegisterFPBatch(sgTemplatePair, count)

Register a set of id's/sgTemplates into the in memory database.

Name	Type	Description
sgTemplatePair	SSIdTemplatePair	Incoming structure that contains id's/sgTemplate pairs to be registered into the in memory database.
count	UInt64	Count of id's/sgTemplates to be registered into the in memory database.

Removal of Templates from database:

SSError = RemoveFP(templateId)

Register a template into the in memory database.

Name	Type	Description
templateId	UInt32	Specifies the integer id of the template to be removed.

SSError = RemoveFPBatch(Templateids, count)

Register a set of id's/sgTemplates into the in memory database.

Name	Type	Description
templateIDs	UInt32	Array of templateIds of templates to be removed from in memory database.
Count	UInt64	Count of id's to be removed from the in memory database.

Search Templates from database:

SSError = SearchFP(template, candList)

Search in memory database for a list of potential matching candidates.

Name	Type	Description
Template	Byte[]	Incoming template to be converted.
candList	SSCandList	Reference to structure SSCandList that is populated from SecuSearch call into the in memory database.

SSError = IdentifyFP(sgTemplate, secuLevel, templateid)

Search in memory database for a matching template that is equal or above the security level passed in.

Name	Type	Description
sgTemplate	Byte[]	Incoming template on which is used to find a match or not find one.

secuLevel	SSConfLevel	Enumeration of Int32 security level from 0 to 9.
templateId	UInt32	Reference to template id of the closest matching Id of the template that is passed in.

Error Codes

ERROR CODE	DESCRIPTION
0	No error
1	Creation failed (fingerprint reader not correctly installed or driver files error)
2	Function failed (wrong type of fingerprint reader or not correctly installed)
3	Internal (invalid parameters to sensor API)
5	DLL load failed
6	DLL load failed for driver
7	DLL load failed for algorithm
51	System file load failure
52	Sensor chip initialization failed
53	Sensor line dropped
54	Timeout
55	Device not found
56	Driver load failed
57	Wrong image
58	Lack of bandwidth
59	Device busy
60	Cannot get serial number of the device
61	Unsupported device
101	Very low minutiae count
102	Wrong template type
103	Invalid template
104	Invalid template
105	Could not extract features
106	Match failed
1000	No memory
4000	Invalid parameter passed to service
2000	Internal error
3000	Internal error extended
6000	Certificate error cannot decode
10001	License error
10002	Invalid domain
10003	License expired
10004	WebAPI may not have received the origin header from the browser