Security

Hanwha

SUNAPI

v2.6.2
2023-04-07

Contact Information

Hanwha Vision Co., Ltd.
Hanwha Vision 6, Pangyo-ro 319beon-gil, Bundang-gu, Seongnam-si, Gyeonggi-do, 13488, KOREA
www.hanwhavision.com

Hanwha Vision America
500 Frank W. Burr Blvd. Suite 43 Teaneck, NJ 07666
hanwhavisionamerica.com

Hanwha Vision Europe
Heriot House, Heriot Road, Chertsey, Surrey, KT16 9DT, United Kingdom
hanwhavision.eu

Hanwha Vision Middle East FZE
Jafza View 18, Office 2001-2003, Po Box 263572, Jebel Ali Free Zone, Dubai, United Arab Emirates
www.hanwhavision.com/ar

Table of Contents

Chapter 1. Overview

1.1. Description

Hanwha Vision provides network security and authentication methods to support secure data transfers. **security.cgi** configures the general security settings for Hanwha Vision video surveillance devices.

The following submenus of security.cgi are used for network security settings:

- **ipfilter**: Requests and configures IP address filtering to block or allow certain IP addresses.
- **802Dot1x**: Requests and configures the parameters required for accessing an 802.1x protected network.**rtsp**: Requests and selects the RTSP authentication method.
- **ssl**: Requests and configures the SSL (Secure Socket Layer) settings.
- **guest**: Requests and enables/disables guest logins.
- **users**: Requests, adds and deletes system users and sets access permissions.
- **usergroups**: Requests, adds and deletes system user groups and sets access permissions for the user groups.
- **authority**: Sets several access permissions and auto logout times.
- **additionalpassword**: Sets multiple passwords for the user.
- **rsa**: Requests public RSA key
- **camerausers**: Configures to try connecting cameras with the corresponding IDs and passwords when NVR discovers them or test connection has been found.
- **clienthttpsstatus**: Requests the mutual authentication status of the client.
- **tlsversion:** Requests and configures the TLS (Transport Layer Security) settings.
- **cameravalidationstatus**: Requests the camera validation status from NVR.
- **cacertificate**: Configures and handles the CA certificate used in a device.

NOTE It is highly recommended not to use plain text passwords in the submenus. Password encryption should be used. Refer to the Application Programming Guide Document for details.

Chapter 2. IP Address Filtering

2.1. Description

The **ipfilter** submenu configures IP address filtering. This submenu controls the device access based IP filter settings. The feature provides both Allow and Deny filtering types. Deny filtering allows all IP addresses except the listed addresses, while Allow filtering blocks all IP addresses except the listed addresses.

Access level

Action	Camera	NVR
view	Admin	User
set	Admin	User
add, update	Admin	User
remove	Admin	User

2.2. Syntax

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=
ipfilter&action=<value>[&<parameter>=<value>...]
```

2.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view				Reads the IP address filtering settings.
set	AccessType	REQ, RES	<enum> Allow, Deny	Access type • Allow: Allows access to listed IP addresses • Deny: Blocks access to listed IP addresses **Note** **AccessType** must be sent together with the **set** action.
	IPType	REQ, RES	<enum>	IP type

Action	Parameter	Request/ Response	Type/ Value	Description
add, update	IPIndex	REQ, RES	<int>	Position of a IP address in the list **Note** **IPType**, **IPIndex**, and **Enable** must be sent together for the **update** action.
	IPType	REQ, RES	<enum> IPv4, IPv6	IP type The IP type is either IPv4 or IPv6. **Note** **IPType**, **Address**, and **Enable** parameters must be sent together for the **add** action.
	Address	REQ, RES	<string>	IP address to be configured for device access IPv4 and IPv6 are both allowed.
	Mask	REQ, RES	<int>	Netmask for the IP address
	Enable	REQ, RES	<bool> True, False	Whether to apply IPv4 or IPv6 address filtering
remove	IPType	REQ	<enum> IPv4, IPv6	IP type that is to be removed. **Note** **IPType** and **IPIndex** must be sent together for the **remove** action.
	IPIndex	REQ	<int>	IP Index that is to be removed.

2.4. Examples

2.4.1. Getting the current IP filtering settings

REQUEST

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=ipfilter&action=view
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
"ipfilter.IPType.IPIndex=Address/Mask/Enable"
AccessType=Deny
ipfilter.IPv4.1=192.168.75.135/32/False
ipfilter.IPv6.1=2001:1:1:1:1:1:1:1/128/True
```

JSON RESPONSE (For Camera)

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "AccessType": "Deny",
    "IPFilters": [
        {
            "IPType": "IPv4",
            "Filters": [
                {
                    "IPIndex": 1,
                    "Address": "192.168.75.135",
                    "Mask": 32,
                    "Enable": false
                }
            ]
        },
        {
            "IPType": "IPv6",
            "Filters": [
                {
                    "IPIndex": 1,
                    "Address": "2001:1:1:1:1:1:1:1",
                    "Mask": 128,
                    "Enable": true
                }
            ]
        }
    ]
}
```

JSON RESPONSE (For NVR)

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "IPFilters": [
        {
            "IPType": "IPv4",
            "AccessType": "Deny",
            "Filters": [
                {
                    "IPIndex": 1,
                    "Address": "192.168.75.135",
                    "Mask": 32,
                    "Enable": false
                }
            ]
        },
        {
            "IPType": "IPv6",
            "AccessType": "Allow",
            "Filters": [
                {
                    "IPIndex": 1,
                    "Address": "2001:1:1:1:1:1:1:1",
                    "Mask": 128,
                    "Enable": true
                }
            ]
        }
    ]
}
```

2.4.2. Setting an IPv6 address and enabling filtering

To add the IP address to be filtered with the **add** action, the **IPType**, **Address**, and **Enable** parameters must be set.

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=ipfilter&action=add&&IPType=IPv6&Address=fe80::209
1:18ff:fe71:1111&Mask=32&Enable=True
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
OK
IPIndex=1
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "Response": "Success"
}
```

2.4.3. Updating an IPv6 address and disabling filtering

To modify a filtered address via the **update** action, the **IPType**, **IPIndex**, and **Enable** parameters must be set.

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=ipfilter&action=update&IPIndex=1&IPType=IPv6&Addre
ss=fe80::2091:18ff:fe71:1111&Mask=32&Enable=False
```

2.4.4. Deleting an IPv4 address from IP filtering list

To remove a filtered address with the **remove** action, the **IPType** and **IPIndex** parameters must be set.

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=ipfilter&action=remove&IPIndex=1&IPType=IPv4
```

Chapter 3. 802.1x Setup

3.1. Description

The **802Dot1x** submenu requests and configures the parameters required for accessing an 802.1x protected network.

Access level

Action	Camera	NVR
view	Admin	User
set	Admin	User
install	Admin	User
remove	Admin	User

3.2. Syntax

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=
802Dot1x&action=<value>[&<parameter>=<value>...]
```

3.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view				Reads the 802.1x settings
	Status	RES	<enum> Unauthorized, Authorized, Stopped	Status (read-only)
	CACertificateInstalled	RES	<bool> True, False	Whether or not a CA certificate is installed
	ClientCertificateInstalled	RES	<bool> True, False	Whether or not a public certificate containing the client certificate key is installed
	ClientPrivateKeyInstalled	RES	<bool> True, False	Whether or not a public certificate containing the client private key is installed
	IsPasswordSet	RES	<bool> True, False	Whether or not the EAPOL password is set
set	InterfaceName	REQ, RES	<string>	Interface name (read-only for network cameras)

Action	Parameter	Request/ Response	Type/ Value	Description
	Enable	REQ, RES	<bool> True, False	Whether to activate 802.1x mode
	EAPOLVersion	REQ, RES	<enum> 1, 2	EAPOL (Extensible Authentication Protocol over LAN) version
	EAPOLId	REQ, RES	<string>	EAPOL ID
	EAPOLPassword	REQ, RES	<string>	EAPOL password
	IsEAPOLPasswordEncrypted	REQ	<bool> True, False	If set to true, EAPOLPassword is encrypted using the public key provided by the **rsa** submenu of **security**.cgi and sent as post payload. Refer to the Application Programmer's Guide.
	ClientCertificateInUse	REQ, RES	<string>	Set or get the client certificate to (in) use
	CACertificateInUse	REQ,RES	<string>	Set or get the CA certificate to (in) use
	EAPOLType	REQ, RES	<enum> EAP-TLS, LEAP	EAPOL type
install	InterfaceName	REQ	<string>	Interface name NVR ONLY
remove	CertificateType	REQ	<enum> CACertificate, ClientCertificate, ClientPrivateKey	Certificate type **Note** **CertificateType** must be sent together with the **remove** action
	InterfaceName	REQ	<string>	Interface name NVR ONLY

3.4. Examples

3.4.1. Getting the current 802.1x settings

REQUEST

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=802Dot1x&action=view
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
InterfaceName=1a5a97d2-464a-4222-91c6-140ff36b82b6
Enable=False
EAPOLType=EAP-TLS
Status=Stopped
CACertificateInstalled=False
ClientCertificateInstalled=False
ClientPrivateKeyInstalled=False
ClientCertificateInUse=
CACertificateInUse=
EAPOLVersion=1
EAPOLId=test
EAPOLPassword=test
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "IEEE802Dot1x": [
        {
            "InterfaceName": "NetworkInterface1",
            "Enable": false,
            "Status": "Stopped",
            "CACertificateInstalled": false,
            "ClientCertificateInstalled": false,
            "ClientPrivateKeyInstalled": false,
            "ClientCertificateInUse": "",
            "CACertificateInUse": "",
            "EAPOLVersion": "1",
            "EAPOLId": "",
            "EAPOLPassword": "",
            "EAPOLType": "EAP-TLS",
```

```
            "IsPasswordSet": false
        }
    ]
}
```

3.4.2. Setting the EAPOL

Setting the EAPOL version to '1', the EAPOL ID to 'test' and the EAPOL password to '123'

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=802Dot1x&action=set&EAPOLVersion=1&EAPOLId=test&EA
POLPassword=123
```

The following request example is for NVR only.

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=802Dot1x&action=set&InterfaceName=Network2&EAPOLVe
rsion=1&EAPOLId=test& EAPOLPassword=123
```

3.4.3. Installing an 802.1x certificate

REQUEST

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=802Dot1x&action=install
```

When requesting for an 802.1x certificate to be installed, data should be sent via the POST method in the following format.

The certtype value is the CACertificate, ClientCertificate or ClientPrivateKey. The certlength value is the data size of the certificate. The certdata value is certificate data.

```
<SetData802Dot1x>
    <PublicCertType>certtype</PublicCertType>
    <CertLength>certlength</CertLength>
    <CertData>certdata</CertData>
</SetData802Dot1x>
```

NOTE

Now ssl and cacertificate handles all the certificates used in a device. So we recommend using those submenus to install new certificates and just use **ClientCertificateInUse** and

> **CACertificateInUse** parameter to set the certificates that you want to use.

CURL command

802.1x certificate install can be tested with CURL as below. To learn about CURL, please refer to http://curl.haxx.se.

> **NOTE** To get a JSON response add the -H ted with CURL as shown below in the header of the request.

```
curl -v --digest -u <userid>:<password> --data-urlencode @802dot1x.xml
"http://<Device IP>/stw-cgi/security.cgi?msubmenu=802Dot1x&action=install"
-H "Expect:"
```

(The following example is for NVR only.)

```
curl --digest -u <userid>:<password> "http://<Device IP>/stw-
cgi/security.cgi?msubmenu=802Dot1x&action=install&InterfaceName=Network2" -H
"Expect:" --data-urlencode @802dot1x.xml
```

The above command will produce a request to the device as below:

```
POST /stw-cgi/security.cgi?msubmenu=802Dot1x&action=install HTTP/1.1
Content-Length: 1985
Content-Type: application/x-www-form-urlencoded
```

TEXT RESPONSE

```
OK
```

JSON RESPONSE

```
{
    "Response": "Success"
}
```

3.4.4. Deleting a certificate

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=802Dot1x&action=remove&CertificateType=CACertifica
```

```
te
```

The following request example is for NVR only.

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=802Dot1x&action=remove&InterfaceName=Network2&Cert
ificateType=CACertificate
```

Chapter 4. RTSP Authentication

4.1. Description

The **rtsp** submenu selects the RTSP authentication method.

Access level

Action	Camera	NVR
view	Admin	User
set	Admin	User

4.2. Syntax

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=
rtsp&action=<value>[&<parameter>=<value>...]
```

4.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view				Reads the RTSP authentication settings.
	RTSPAuthentication	RES	<enum> Anonymous, Protected	Whether to allow anonymous access without login for RTSP URL requests
set	RTSPAuthentication	REQ, RES	<enum> Anonymous, Protected	Whether to allow anonymous access without logging in for RTSP URL requests (read-only for NVR) **Note** **RTSPAuthentication** must be sent together with the **set** action.

4.4. Examples

4.4.1. Getting the current RTSP authentication settings

REQUEST

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=rtsp&action=view
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
RTSPAuthentication=Protected
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "RTSPAuthentication": "Protected"
}
```

4.4.2. Enabling anonymous RTSP authentication

To use the **set** action, the **RTSPAuthentication** parameter must be set at the same time.

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=rtsp&action=set&RTSPAuthentication=Anonymous
```

Chapter 5. SSL (HTTPS) Settings

5.1. Description

The **ssl** submenu configures the SSL (Secure Socket Layer) settings.

Access level

Action	Camera	NVR
view	Admin	User
set	Admin	User
install	Admin	User
remove	Admin	User

5.2. Syntax

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=
ssl&action=<value>[&<parameter>=<value>...]
```

5.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view				Reads the SSL (HTTPS) settings
	PublicCertificateInstalled	RES	<bool> True, False	Whether or not a public certificate is installed (read-only)
	PublicCertificateName	RES	<csv>	Public certificate name
	UpdateDeviceHostname	RES	<bool>	Whether device hostname is updated
	CertificateInUse	RES	<string>	Name of certificate currently in use
	Certificate.#.CertificateName	RES	<string>	Certificate name
	Certificate.#.Type	RES	<enum> Unique, SelfSigned, Public	Certificate type
	Certificate.#.Subject	RES	<string>	Subject of the certificate
	Certificate.#.SubjectAlternativeName	RES	<string>	Subject alternative name (SAN)

Action	Parameter	Request/ Response	Type/ Value	Description
	Certificate.#.Issuer	RES	<string>	Issuer
	Certificate.#.IssueDate	RES	<string>	Issued date
	Certificate.#.ExpiryDate	RES	<string>	Expiry date
	Certificate.#.Version	RES	<string>	Version
	Certificate.#.SerialNumber	RES	<string>	Serial number
	Certificate.#.Signature	RES	<string>	Signature
	Certificate.#.Thumbprint	RES	<string>	Thumbprint
	Certificate.#.IsRemovable	RES	<bool>	Whether the certificate can be deleted
	Certificate.#.IsEncrypted	RES	<bool>	Whether the certificate is encrypted
add	CertificateName	REQ	<string>	The name of the certificate
	Type	REQ	<enum> SelfSigned	The type of the certificate • SelfSigned: Access using a built-in certificate from the device
	CommonName	REQ	<string>	Common name (CN)
	SubjectAlternativeName	REQ	<string>	Subject alternative name (SAN)
	ExpiryDate	REQ	<string>	Certificate expiry date
	Country	REQ	<string>	Country © **Note** ISO-3166-1 alpha-2 codes
	Province	REQ	<string>	Province (ST)
	Location	REQ	<string>	Location (L)
	Organization	REQ	<string>	Organization (O)
	Division	REQ	<string>	Division (OU)

Action	Parameter	Request/ Response	Type/ Value	Description
	EmailID	REQ	<string>	E-mail address **Note** Multiple e-mails are separated by commas
set	Policy	REQ, RES	<enum> HTTP, HTTPSProprietary, HTTPSPublic, HTTPandHTTPSProprietary, HTTPandHTTPSPublic	Select the SSL method: • HTTP: Access using only HTTP • HTTPSProprietary: Access using a built-in certificate in the device • HTTPSPublic: Access using a certificate that the user has installed (HTTPSPublic mode is valid only when the certificate is installed) • HTTPandHTTPSProprietary: HTTP and HTTPSProprietary mode mode • HTTPandHTTPSPublic: HTTP and HTTPSPublic mode
	UpdateDeviceHostname	REQ	<bool> True, False	Whether or not device hostname is updated
	CertificateInUse	REQ	<string>	Name of certificate currently in use
	ClientCertificateAuthenticationEnable	REQ, RES	<bool> True, False	Whether or not mutual authentication is enabled

Action	Parameter	Request/ Response	Type/ Value	Description
	ClientCertificateAuthenticationMode	REQ, RES	<enum> ALLOW_ALL_CERT, ALLOW_CERT_FROM_KNOWN_CA, ALLOW_CERT_FROM_VALID_CLIENT_AND_KNOWN_CA	Select the mode of certificate authentication • ALLOW_ALL_CERT: Allows all connections • ALLOW_CERT_FROM_KNOWN_CA: Allows only mutually authenticated connections • ALLOW_CERT_FROM_VALID_CLIENT_AND_KNOWN_CA: Allows only mutually authenticated connections (including Device ID authentication) **Note** Only works if **Policy** is HTTPSProprietary.
remove	CertificateName	REQ	<string>	Certificate name
install				POST method

5.4. Examples

5.4.1. Getting the current SSL settings

REQUEST

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=ssl&action=view
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
Policy=HTTPandHTTPSProprietary
PublicCertificateInstalled=False
SelfSignedCertificateInstalled=True
PublicCertificateName=
UpdateDeviceHostname=False
CertificateInUse=CA
```

```
ClientCertificateAuthenticationEnable=False
ClientCertificateAuthenticationMode=ALLOW_ALL_CERT
Certificate.1.CertificateName=HTW_default
Certificate.1.Type=Unique
Certificate.1.Subject=/C=KR/O=Hanwha Vision/OU=Security
Device/CN=00091867F9BC.hanwhavision.com/serialNumber=5f55694d-f741-41a6-
9c6f-ceb4b1ee124c
Certificate.1.SubjectAlternativeName=00091867F9BC.hanwhavision.com
Certificate.1.Issuer=/C=KR/O=Hanwha Vision/OU=Security Solution/CN=Hanwha
Vision Private Root CA 2
Certificate.1.IssueDate=Dec 1 10:05:06 2020 GMT
Certificate.1.ExpiryDate=Nov 24 10:05:06 2050 GMT
Certificate.1.Version=V3
Certificate.1.SerialNumber=00 12 35 8F
Certificate.1.Signature=sha256WithRSAEncryption
Certificate.1.Thumbprint=e268cf387b1a19c5ea3c04be2f24a2053069d8742b9741f1d03
4937872ce8538
Certificate.1.IsRemovable=False
Certificate.1.IsEncrypted=False
Certificate.2.CertificateName=test
Certificate.2.Type=Public
Certificate.2.Subject=/C=FR/ST=Radius/O=Example
Inc./CN=user@example.org/emailAddress=user@example.org
Certificate.2.SubjectAlternativeName=-
Certificate.2.Issuer=/C=FR/ST=Radius/L=Somewhere/O=Example
Inc./emailAddress=admin@example.org/CN=Example Certificate Authority
Certificate.2.IssueDate=Jul 1 04:36:15 2021 GMT
Certificate.2.ExpiryDate=May 10 04:36:15 2031 GMT
Certificate.2.Version=V3
Certificate.2.SerialNumber=00
Certificate.2.Signature=sha256WithRSAEncryption
Certificate.2.Thumbprint=2b58473f4655282053630336ad25a84eae3e53155030a6fa10e
dc3576feec51d
Certificate.2.IsRemovable=True
Certificate.2.IsEncrypted=True
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "Policy": "HTTPandHTTPSProprietary",
    "PublicCertificateInstalled": false,
    "SelfSignedCertificateInstalled": true,
    "PublicCertificateName": "",
    "UpdateDeviceHostname": false,
    "CertificateInUse": "CA",
    "ClientCertificateAuthenticationEnable": false,
    "ClientCertificateAuthenticationMode": "ALLOW_ALL_CERT",
    "Certificates": [
        {
            "CertificateName": "HTW_default",
            "Type": "Unique",
            "Subject": "/C=KR/O=Hanwha Vision/OU=Security
Device/CN=00091867F9BC.hanwhavision.com/serialNumber=5f55694d-f741-41a6-
9c6f-ceb4b1ee124c",
            "SubjectAlternativeName": "00091867F9BC.hanwhavision.com",
            "Issuer": "/C=KR/O=Hanwha Vision/OU=Security Solution/CN=Hanwha
Vision Private Root CA 2",
            "IssueDate": "Dec 1 10:05:06 2020 GMT",
            "ExpiryDate": "Nov 24 10:05:06 2050 GMT",
            "Version": "V3",
            "SerialNumber": "00 12 35 8F ",
            "Signature": "sha256WithRSAEncryption",
            "Thumbprint":
"e268cf387b1a19c5ea3c04be2f24a2053069d8742b9741f1d034937872ce8538",
            "IsRemovable": false,
            "IsEncrypted": false
        },
        {
            "CertificateName": "test",
            "Type": "Public",
            "Subject": "/C=FR/ST=Radius/O=Example
Inc./CN=user@example.org/emailAddress=user@example.org",
            "SubjectAlternativeName": "-",
            "Issuer": "/C=FR/ST=Radius/L=Somewhere/O=Example
Inc./emailAddress=admin@example.org/CN=Example Certificate Authority",
            "IssueDate": "Jul 1 04:36:15 2021 GMT",
            "ExpiryDate": "May 10 04:36:15 2031 GMT",
            "Version": "V3",
            "SerialNumber": "00",
```

```
            "Signature": "sha256WithRSAEncryption",
            "Thumbprint":
"2b58473f4655282053630336ad25a84eae3e53155030a6fa10edc3576feec51d",
            "IsRemovable": true,
            "IsEncrypted": true
        }
    ]
}
```

5.4.2. Enabling HTTPS

To use the **set** action, the **Policy** parameter must be set at the same time.

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=ssl&action=set&Policy=HTTPSPublic
```

5.4.3. Installing a certificate

REQUEST

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=ssl&action=install
```

When requesting a certificate to be installed, data should be sent via the POST method in the following format.

The certname value is the certificate name. The certlength value is the certificate data size. The certdata value is the certificate data. The keylength is the key data size. The keydata value is the key data.

```
<SetHTTPSData>
    <PublicCertName>certname</PublicCertName>
    <CertLength>certlength</CertLength>
    <CertData>certdata</CertData>
    <KeyLength>keylength</KeyLength>
    <KeyData>keydata</KeyData>
</SetHTTPSData>
```

CURL command

The certificate install can be tested with CURL as below. To learn about CURL, please refer to http://curl.haxx.se.

NOTE To get a JSON response, add the -H "Accept: application/json" header to the request.

```
curl -v --digest -u <userid>:<password> --data-urlencode @ssl.xml
"http://<Device IP>/stw-cgi/security.cgi?msubmenu=ssl&action=install" -H
"Expect:"
```

The above command will produce a request to the device that looks like below:

```
POST /stw-cgi/security.cgi?msubmenu=ssl&action=install HTTP/1.1
Content-Length: 3775
Content-Type: application/x-www-form-urlencoded
```

TEXT RESPONSE

```
OK
```

JSON RESPONSE

```
{
    "Response": "Success"
}
```

5.4.4. Add a self-signed certificate

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=ssl&action=add&CertificateName=certName&Type=SelfS
igned&CommonName=192.168.75.123&SubjectAlternativeName=domain.com,testdom.co
m&ExpiryDate=2021-09-
09&Country=KR&Province=Gyeonggi&Location=Bundang&Organization=Hanwha&Divisio
n=RSDT&EmailID=test@hanwha.com
```

TEXT RESPONSE

```
OK
```

JSON RESPONSE

```
{
    "Response": "Success"
```

```
}
```

5.4.5. Use a specific certificate

To delete a certificate with **remove** action, the **CertificateName** parameter must be set at the same time.

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=ssl&action=remove&CertificateName=certname
```

5.4.6. Deleting a certificate

To delete a certificate with the **remove** action, the **CertificateName** parameter must be set at the same time.

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=ssl&action=remove&CertificateName=certname
```

Chapter 6. Guest User Login

6.1. Description

The **guest** submenu enables or disables guest logins.

NOTE | This chapter applies to the network cameras only.

Access level

Action	Camera
view	Admin
set	Admin

6.2. Syntax

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=
guest&action=<value>[&<parameter>=<value>...]
```

6.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view				Reads the guest user login settings.
set	Enable	REQ, RES	<bool> True, False	Enables or disables guest logins. **Note** **Enable** must be sent together with the **set** action.

6.4. Examples

6.4.1. Getting the current guest login setting

REQUEST

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=guest&action=view
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
```

```
<Body>
```

```
Enable=False
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "Enable": false
}
```

6.4.2. Enabling guest login

To use the **set** action, the **Enable** parameter must be set at the same time.

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=guest&action=set&Enable=True
```

Chapter 7. User Configuration

7.1. Description

The **users** submenu adds and deletes system users and sets access permissions.

Access level

Action	Camera	NVR
view	User	User
add, update	Admin	Admin
remove	Admin	Admin

7.2. Syntax

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=
users&action=<value>[&<parameter>=<value>...]
```

7.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view	UserID	REQ	<string>	User ID 'admin' cannot be used as the **UserID** value.
add, update	Index	REQ, RES	<int>	Index of the registered user account. **Note** **Index**, **UserID**, and **Password** must be sent together for the **update** action.
	UserID	REQ, RES	<string>	User ID **Note** **UserID** and **Password** must be sent together with the **add** action.
	UserName	REQ, RES	<string>	User name This parameter is for users only; the admin user name cannot be changed. NVR ONLY

Action	Parameter	Request/ Response	Type/ Value	Description
	Enable	REQ, RES	<bool> True, False	Whether to activate or deactivate the user account. CAMERA ONLY
	VideoProfileAccess	REQ, RES	<bool> True, False	Whether to grant video profile permission. If **VideoProfileAccess** is set to False, only the default profile will be allowed when accessing video. If **VideoProfileAccess** is set to True, all profiles are allowed when accessing video. CAMERA ONLY
	PTZAccess	REQ, RES	<bool> True, False	Whether to grant PTZ control permission. For PTZ models,the values for **PTZAccess**, **AlarmOutputAccess**, and **AudioOutAccess** should be the same. CAMERA ONLY
	AudioInAccess	REQ, RES	<bool> True, False	Whether to grant audio input control permission. CAMERA ONLY
	AudioOutAccess	REQ, RES	<bool> True, False	Whether to grant audio output control permission. For PTZ models, the values for **PTZAccess**, **AlarmOutputAccess**, and **AudioOutAccess** should be the same. CAMERA ONLY
	AlarmOutputAccess	REQ, RES	<bool> True, False	Whether to grant alarm output control permission. For PTZ models, values for **PTZAccess**, **AlarmOutputAccess**, and **AudioOutAccess** should be the same. CAMERA ONLY

Action	Parameter	Request/ Response	Type/ Value	Description
	ViewerAccess	REQ, RES	<bool> True, False	Whether to grant access permission to a viewer **ViewerAccess** is valid only when UserID is NOT set to Admin. NVR ONLY
	AdminAccess	REQ, RES	<bool> True, False	Whether or not to grant admin permission. If this parameter is True, it means that user has full access rights (e.g. administrator). CAMERA ONLY
	PrivacyAreaAccess	REQ, RES	<bool> True, False	Whether to grant access pertmission to create/remove privacy areas If this parameter is True, the user can create a new privacy area or configure one. PTZAccess will be enbled automatically when the device supports PTZ. CAMERA ONLY
	Password	REQ, RES	<string>	User password
	IsPasswordEncrypted	REQ	<bool> True, False	When set to true, password is encrypted using the public key provided by the **rsa** submenu of **security**.cgi, and sent as post payload. Refer to the Application Programmer's Guide.
	GroupID	REQ, RES	<string>	Group ID **GroupID** is invalid, if **UserLevel** is set to Admin, NVR ONLY
	UserLevel	RES	<enum> Guest, Admin, User	User level If **UserLevel** is set to Admin for NVR, **GroupID** is invalid. NVR ONLY

Action	Parameter	Request/ Response	Type/ Value	Description
remove	Index	REQ	<int>	Index number of the user that is to be removed **Note** Either **Index** or **UserID** must be sent together with the **remove** action. CAMERA ONLY
	UserID	REQ	<string>	User ID that is to be removed CAMERA ONLY

Note
To find out the max users supported by the device, refer to the Attributes/Security/Limit/MaxUser attribute in the device attributes section.

7.4. Examples

7.4.1. Getting current users settings

REQUEST

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=users&action=view
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
"Users.Index=UserID/Password/Enable/VideoProfileAccess/PTZAccess/AudioInAcce
ss/AudioOutAccess/AlarmOutputAccess/AdminAccess/PrivacyAreaAccess"
Users.0=admin//True/True/True/True/True/True/True/True
Users.1=user1//False/False/False/False/False/False/False/False
Users.2=user2//False/False/False/False/False/False/False/False
Users.3=user3//False/False/False/False/False/False/False/False
Users.4=user4//False/False/False/False/False/False/False/False
Users.5=user5//False/False/False/False/False/False/False/False
Users.6=user6//False/False/False/False/False/False/False/False
Users.7=user7//False/False/False/False/False/False/False/False
```

```
Users.8=user8//False/False/False/False/False/False/False/False
Users.9=user9//False/False/False/False/False/False/False/False
Users.10=user10//False/False/False/False/False/False/False/False
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```


```
{
    "Users": [
        {
            "Index": 0,
            "UserID": "admin",
            "Password": "",
            "Enable": true,
            "VideoProfileAccess": true,
            "PTZAccess": true,
            "AudioInAccess": true,
            "AudioOutAccess": true,
            "AlarmOutputAccess": true,
            "AdminAccess": true,
            "PrivacyAreaAccess": true
        },
        {
            "Index": 1,
            "UserID": "user1",
            "Password": "",
            "Enable": false,
            "VideoProfileAccess": false,
            "PTZAccess": false,
            "AudioInAccess": false,
            "AudioOutAccess": false,
            "AlarmOutputAccess": false,
            "AdminAccess": false,
            "PrivacyAreaAccess": false
        },
        {
            "Index": 2,
            "UserID": "user2",
```

```
            "Password": "",
            "Enable": false,
            "VideoProfileAccess": false,
            "PTZAccess": false,
            "AudioInAccess": false,
            "AudioOutAccess": false,
            "AlarmOutputAccess": false,
            "AdminAccess": false,
            "PrivacyAreaAccess": false
        },
        {
            "Index": 3,
            "UserID": "user3",
            "Password": "",
            "Enable": false,
            "VideoProfileAccess": false,
            "PTZAccess": false,
            "AudioInAccess": false,
            "AudioOutAccess": false,
            "AlarmOutputAccess": false,
            "AdminAccess": false,
            "PrivacyAreaAccess": false
        },
        {
            "Index": 4,
            "UserID": "user4",
            "Password": "",
            "Enable": false,
            "VideoProfileAccess": false,
            "PTZAccess": false,
            "AudioInAccess": false,
            "AudioOutAccess": false,
            "AlarmOutputAccess": false,
            "AdminAccess": false,
            "PrivacyAreaAccess": false
        },
        {
            "Index": 5,
            "UserID": "user5",
            "Password": "",
            "Enable": false,
```

```
            "VideoProfileAccess": false,
            "PTZAccess": false,
            "AudioInAccess": false,
            "AudioOutAccess": false,
            "AlarmOutputAccess": false,
            "AdminAccess": false,
            "PrivacyAreaAccess": false
        },
        {
            "Index": 6,
            "UserID": "user6",
            "Password": "",
            "Enable": false,
            "VideoProfileAccess": false,
            "PTZAccess": false,
            "AudioInAccess": false,
            "AudioOutAccess": false,
            "AlarmOutputAccess": false,
            "AdminAccess": false,
            "PrivacyAreaAccess": false
        },
        {
            "Index": 7,
            "UserID": "user7",
            "Password": "",
            "Enable": false,
            "VideoProfileAccess": false,
            "PTZAccess": false,
            "AudioInAccess": false,
            "AudioOutAccess": false,
            "AlarmOutputAccess": false,
            "AdminAccess": false,
            "PrivacyAreaAccess": false
        },
        {
            "Index": 8,
            "UserID": "user8",
            "Password": "",
            "Enable": false,
            "VideoProfileAccess": false,
            "PTZAccess": false,
```

```
            "AudioInAccess": false,
            "AudioOutAccess": false,
            "AlarmOutputAccess": false,
            "AdminAccess": false,
            "PrivacyAreaAccess": false
        },
        {
            "Index": 9,
            "UserID": "user9",
            "Password": "",
            "Enable": false,
            "VideoProfileAccess": false,
            "PTZAccess": false,
            "AudioInAccess": false,
            "AudioOutAccess": false,
            "AlarmOutputAccess": false,
            "AdminAccess": false,
            "PrivacyAreaAccess": false
        },
        {
            "Index": 10,
            "UserID": "user10",
            "Password": "",
            "Enable": false,
            "VideoProfileAccess": false,
            "PTZAccess": false,
            "AudioInAccess": false,
            "AudioOutAccess": false,
            "AlarmOutputAccess": false,
            "AdminAccess": false,
            "PrivacyAreaAccess": false
        }
    ]
}
```

The following response example is for NVR only.

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
```

```
<Body>
```

```
"Users.Index=UserID/UserName/Password/GroupID/UserLevel/ViewerAccess"
Users.0=admin/admin/7i8o9p0[//Admin/True
Users.1=tarak/tarak/7i8o9p0[/Group 1/User/True
Users.2=gopal/gopal/7i8o9p0[/Group 2/User/True
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "Users": [
        {
            "Index": 0,
            "UserID": "admin",
            "UserName": "admin",
            "Password": "7i8o9p0[",
            "GroupID": "",
            "ViewerAccess": true,
            "UserLevel": "Admin"
        },
        {
            "Index": 1,
            "UserID": "tarak",
            "UserName": "tarak",
            "Password": "7i8o9p0[",
            "GroupID": "Group 1",
            "ViewerAccess": true,
            "UserLevel": "User"
        },
        {
            "Index": 2,
            "UserID": "anumolu",
            "UserName": "anumolu",
            "Password": "7i8o9p0[",
            "GroupID": "Group 2",
            "ViewerAccess": true,
```

```
            "UserLevel": "User"
        }
    ]
}
```

7.4.2. Getting ‘user1’

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=users&action=view&UserID=user1
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
"Users.Index=UserID/Password/Enable/VideoProfileAccess/PTZAccess/AudioInAcce
ss/AudioOutAccess/AlarmOutputAccess/AdminAccess/PrivacyAreaAccess"
Users.1=user1//False/False/False/False/False/False/False/False
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "Users": [
        {
            "Index": 1,
            "UserID": "user1",
            "Password": "",
            "Enable": false,
            "VideoProfileAccess": false,
            "PTZAccess": false,
            "AudioInAccess": false,
            "AudioOutAccess": false,
            "AlarmOutputAccess": false,
```

```
            "AdminAccess": false,
            "PrivacyAreaAccess": false
        }
    ]
}
```

7.4.3. Adding a user

This is an example in which a user is added that has permission to access all video profiles but without PTZ control.

To add a user, the **UserID** and **Password** parameters must be set together.

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=users&action=add&UserID=user2&Password=123&VideoPr
ofileAccess=True&PTZAccess=False
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
OK
Index=1
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "Response": "Success"
}
```

The following example is for NVR only.

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=users&action=add&UserId=user1&UserName=username1&P
assword=password1&GroupId=group1&UserLevel=User&ViewerAccess=False
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
OK
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "Response": "Success"
}
```

7.4.4. Updating ‘user2’ to give permission for PTZ control

To update user information for the cameras, the **UserID**, **Index**, and **Password** parameters must be sent together.

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=users&action=update&UserID=user2&Index=2&Password=
123&VideoProfileAccess=True&PTZAccess=True
```

The following request example is for NVR only.

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=users&action=update&UserId=admin&Password=1111aaa
```

7.4.5. Updating password

When a user (including admin users) wants to change their current password, it's recommended to use an encrypted password using the RSA PublicKey retrievable from the **rsa** submenu.

if below attribute is supported, current password verification is supported. Value can be 'Optional' or 'Mandatory', if its optional current password verification is optional, if its mandatory only below JSON format can be used for changing password.

```
<attribute name="CurrentPasswordVerification" type="enum" value="Optional"
accesslevel="suser"/>
```

JSON FORMAT (This should be encrypted using the RSA public key and base64 encoding.)

```
{
"CurrentPassword":"q1w2e3r4t5!!",
"NewPassword":"qwerty99!!"
}
```

The post body is sent in the below format,
Base64(RSAEncrypt(JsonString))

If above attribute is not supported, only new password can be encypted using the RSA public key, (example if new password is "qwerty99!!" then only this password needs to be encrypted)

The post body is sent in the below format,
Base64(RSAEncrypt(newpassword))

REQUEST

```
http://192.168.75.196/stw-
cgi/security.cgi?msubmenu=users&action=update&&UserID=user1&Index=1&IsPasswo
rdEncrypted=True
```

POST BODY

```
gH3EsB2O9IfcNY02fQLyOMYA//3ES1nGDT2UUInzN51i0Bw849tuJsdASRWrJt5P+oC1mH05vt2W
7VqICpTZSK4bg578nonpBbv3uTtsTzMyqDrK71hNsblcISZSDTYLa61IeawC88X1//0UTsIHUtT4
XKaiEEAyMFXo7E12dj3itf1ySj//Emo7zq321bRqE3ELOxKQnNFJi7DQf92gHcCN3Tr1WosK3vIA
uZNiOD2txsEIBPxWdS+4RGZNAKzDy62yrJaoIe1lvU3xLA94KH7mGgtsebdXx1X6xay0wTyEtBf8
/TSt2in/PHKx8ZGJxqTq0eSZDsf40nFXkl5K7w==
```

If an admin is changing the password of a normal user, verifying the current password is not required. Only the password string can be encrypted using the RSA public key and be sent in the POST body.

NOTE | Plain text password updates will soon be deprecated.

7.4.6. Deleting ‘user2’

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=users&action=remove&UserID=user2
```

Deleting ‘user2’ via the user’s index number

Assuming the 'user2' index is '3', user2 can be deleted by using the index number.

REQUEST

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=users&action=remove&Index=3
```

Chapter 8. User Group Configuration

8.1. Description

The **usergroups** submenu adds and deletes system user groups and sets access permissions for the user groups.

> **NOTE** This chapter applies to NVR only.

Access level

Action	NVR
view	User
add, update	Admin
remove	Admin

8.2. Syntax

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=
usergroups&action=<value>[&<parameter>=<value>...]
```

8.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view	UserGroupID	REQ	<string>	User group ID
add, update	Index	REQ, RES	<int>	User group index **Note** **Index** must be sent for the **update** action.
	UserGroupID	REQ, RES	<string>	User group ID
	LiveChannel	REQ, RES	<csv> #, None	Live channel • None: Disables all channels
	SearchChannel	REQ, RES	<csv> #, None	Search channel • None: Disables all channels
	BackupChannel	REQ, RES	<csv> #, None	Backup channel • None: Disables all channels

Action	Parameter	Request/ Response	Type/ Value	Description
	PTZAccess	REQ, RES	<bool> True, False	Whether to grant permission for PTZ control **PTZAccess** is valid only when **LiveChannel** is enabled.
	RecordStartAccess	REQ, RES	<bool> True, False	Whether to grant access permission to start recording
	RecordStopAccess	REQ, RES	<bool> True, False	Whether to grant access permission to stop recording
	AlarmOutputAccess	REQ, RES	<bool> True, False	Whether to grant permission to access the alarm output
	ShutdownAccess	REQ, RES	<bool> True, False	Whether to grant permission to access shutdown
	MenuAccess	REQ, RES	<csv> System, Device, Record, Event, Network, None	Accessible menu
	SystemMenuAccess	REQ, RES	<csv> DateTimeLanguage, SystemManagement, SystemLog, EventLog, BackupLog, Holiday, None	Accessible system menu
	DeviceMenuAccess	REQ, RES	<csv> CameraRegistration, CameraSetup, LiveSetup, ChannelSetup, DeviceFormat, iSCSI, RAID, HDDAlarm, Monitor, POSConf, POSEventConf, None	Accessible device menu
	RecordMenuAccess	REQ, RES	<csv> RecordingSchedule, NvrRecordSetup, NetCamRecordSetup, RecordOption, None	Accessible record menu

Action	Parameter	Request/ Response	Type/ Value	Description
	EventMenuAccess	REQ, RES	<csv> NvrSensorDetection, NetCamSensorDetection, NvrEventDetection, NetCamEventDetection, VideoLossDetection, AlarmSchedule, Gsensor, None	Accessible event menu
	NetworkMenuAccess	REQ, RES	<csv> NetworkInterface, NetworkPort, DDNS, IPFilter, SSL, 802.1x, LiveStreaming, SMTP, EventMail, GroupAndRecipientEmail, SNMP, DHCPServer,MTS, None	Accessible network menu
remove	Index	REQ	<int>	User group index **Note** Either **Index** or **UserGroupID** must be sent together with the **remove** action.
	UserGroupID	REQ	<string>	User group ID that is to be removed

8.4. Examples

8.4.1. Getting current user group settings

REQUEST

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=usergroups&action=view
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
Index.1.UserGroupID=Group 1
Index.1.LiveChannel=0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,
22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,4
7,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63
Index.1.SearchChannel=0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,2
1,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46
,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63
Index.1.BackupChannel=0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,2
1,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46
,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63
Index.1.MenuAccess=System,Device,Record,Event,Network
Index.1.SystemMenuAccess=DateTimeLanguage,SystemManagement,SystemLog,BackupL
og,EventLog
Index.1.DeviceMenuAccess=CameraRegistration,CameraSetup,LiveSetup,ChannelSet
up,DeviceFormat,HDDAlarm,iSCSI,RAID,Monitor
Index.1.RecordMenuAccess=RecordingSchedule,NvrRecordSetup,NetCamRecordSetup,
RecordOption
Index.1.EventMenuAccess=NvrSensorDetection,NetCamSensorDetection,NvrEventDet
ection,NetCamEventDetection,VideoLossDetection,AlarmSchedule
Index.1.NetworkMenuAccess=NetworkInterface,NetworkPort,DDNS,IPFilter,SSL,802
.1x,LiveStreaming,SMTP,EventMail,GroupAndRecipientEmail,SNMP,DHCPServer
Index.1.RecordStartAccess=True
Index.1.RecordStopAccess=True
Index.1.PTZAccess=True
Index.1.AlarmOutputAccess=True
Index.1.ShutdownAccess=True
Index.2.UserGroupID=Group 2
Index.2.LiveChannel=None
Index.2.SearchChannel=None
Index.2.BackupChannel=None
Index.2.MenuAccess=None
Index.2.SystemMenuAccess=None
Index.2.DeviceMenuAccess=None
Index.2.RecordMenuAccess=None
Index.2.EventMenuAccess=None
Index.2.NetworkMenuAccess=None
Index.2.RecordStartAccess=False
Index.2.RecordStopAccess=False
Index.2.PTZAccess=False
Index.2.AlarmOutputAccess=False
```

```
Index.2.ShutdownAccess=False
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "UserGroups": [
        {
            "Index": 1,
            "UserGroupID": "Group 1",
            "LiveChannel": [
                "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
                "10", "11", "12", "13", "14", "15", "16", "17", "18", "19",
                "20", "21", "22", "23", "24", "25", "26", "27", "28", "29",
                "30", "31", "32", "33", "34", "35", "36", "37", "38", "39",
                "40", "41", "42", "43", "44", "45", "46", "47", "48", "49",
                "50", "51", "52", "53", "54", "55", "56", "57", "58", "59",
                "60", "61", "62", "63"
            ],
            "SearchChannel": [
                "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
                "10", "11", "12", "13", "14", "15", "16", "17", "18", "19",
                "20", "21", "22", "23", "24", "25", "26", "27", "28", "29",
                "30", "31", "32", "33", "34", "35", "36", "37", "38", "39",
                "40", "41", "42", "43", "44", "45", "46", "47", "48", "49",
                "50", "51", "52", "53", "54", "55", "56", "57", "58", "59",
                "60", "61", "62", "63"
            ],
            "BackupChannel": [
                "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
                "10", "11", "12", "13", "14", "15", "16", "17", "18", "19",
                "20", "21", "22", "23", "24", "25", "26", "27", "28", "29",
                "30", "31", "32", "33", "34", "35", "36", "37", "38", "39",
                "40", "41", "42", "43", "44", "45", "46", "47", "48", "49",
                "50", "51", "52", "53", "54", "55", "56", "57", "58", "59",
                "60", "61", "62", "63"
            ],
            "MenuAccess": [
```

```
                "System",
                "Device",
                "Record",
                "Event",
                "Network"
            ],
            "SystemMenuAccess": [
                "DateTimeLanguage",
                "SystemManagement",
                "SystemLog",
                "BackupLog",
                "EventLog"
            ],
            "DeviceMenuAccess": [
                "CameraRegistration",
                "CameraSetup",
                "LiveSetup",
                "ChannelSetup",
                "DeviceFormat",
                "HDDAlarm",
                "iSCSI",
                "RAID",
                "Monitor"
            ],
            "RecordMenuAccess": [
                "RecordingSchedule",
                "NvrRecordSetup",
                "NetCamRecordSetup",
                "RecordOption"
            ],
            "EventMenuAccess": [
                "NvrSensorDetection",
                "NetCamSensorDetection",
                "NvrEventDetection",
                "NetCamEventDetection",
                "VideoLossDetection",
                "AlarmSchedule"
            ],
            "NetworkMenuAccess": [
                "NetworkInterface",
                "NetworkPort",
```

```
                "DDNS",
                "IPFilter",
                "SSL",
                "802.1x",
                "LiveStreaming",
                "SMTP",
                "EventMail",
                "GroupAndRecipientEmail",
                "SNMP",
                "DHCPServer"
            ],
            "RecordStartAccess": true,
            "RecordStopAccess": true,
            "PTZAccess": true,
            "AlarmOutputAccess": true,
            "ShutdownAccess": true
        },
        {
            "Index": 2,
            "UserGroupID": "Group 2",
            "LiveChannel": [
                "None"
            ],
            "SearchChannel": [
                "None"
            ],
            "BackupChannel": [
                "None"
            ],
            "MenuAccess": [
                "None"
            ],
            "SystemMenuAccess": [
                "None"
            ],
            "DeviceMenuAccess": [
                "None"
            ],
            "RecordMenuAccess": [
                "None"
            ],
```

```
            "EventMenuAccess": [
                "None"
            ],
            "NetworkMenuAccess": [
                "None"
            ],
            "RecordStartAccess": false,
            "RecordStopAccess": false,
            "PTZAccess": false,
            "AlarmOutputAccess": false,
            "ShutdownAccess": false
        }
    ]
}
```

8.4.2. Getting ‘Group 1’ user group settings

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=usergroups&action=view&UserGroupID=Group 1
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
Index.1.UserGroupID=Group 1
Index.1.LiveChannel=0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,
22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,4
7,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63
Index.1.SearchChannel=0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,2
1,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46
,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63
Index.1.BackupChannel=0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,2
1,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46
,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63
Index.1.MenuAccess=System,Device,Record,Event,Network
Index.1.SystemMenuAccess=DateTimeLanguage,SystemManagement,SystemLog,BackupL
og,EventLog
```

```
Index.1.DeviceMenuAccess=CameraRegistration,CameraSetup,LiveSetup,ChannelSetup,DeviceFormat,HDDAlarm,iSCSI,RAID,Monitor
Index.1.RecordMenuAccess=RecordingSchedule,NvrRecordSetup,NetCamRecordSetup,RecordOption
Index.1.EventMenuAccess=NvrSensorDetection,NetCamSensorDetection,NvrEventDetection,NetCamEventDetection,VideoLossDetection,AlarmSchedule
Index.1.NetworkMenuAccess=NetworkInterface,NetworkPort,DDNS,IPFilter,SSL,802.1x,LiveStreaming,SMTP,EventMail,GroupAndRecipientEmail,SNMP,DHCPServer
Index.1.RecordStartAccess=True
Index.1.RecordStopAccess=True
Index.1.PTZAccess=True
Index.1.AlarmOutputAccess=True
Index.1.ShutdownAccess=True
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "UserGroups": [
        {
            "Index": 1,
            "UserGroupID": "Group 1",
            "LiveChannel": [
                "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10",
"11", "12",
                "13", "14", "15", "16", "17", "18", "19", "20", "21", "22",
"23",
                "24", "25", "26", "27", "28", "29", "30", "31", "32", "33",
"34",
                "35", "36", "37", "38", "39", "40", "41", "42", "43", "44",
"45",
                "46", "47","48", "49", "50", "51", "52", "53", "54", "55",
"56",
                "57", "58", "59", "60", "61", "62", "63"
            ],
            "SearchChannel": [
                "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10",
"11", "12",
```

```
                "13", "14", "15", "16", "17", "18", "19", "20", "21", "22",
"23",
                "24", "25", "26", "27", "28", "29", "30", "31", "32", "33",
"34",
                "35", "36", "37", "38", "39", "40", "41", "42", "43", "44",
"45",
                "46", "47","48", "49", "50", "51", "52", "53", "54", "55",
"56",
                "57", "58", "59", "60", "61", "62", "63"
            ],
            "BackupChannel": [
                "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10",
"11", "12",
                "13", "14", "15", "16", "17", "18", "19", "20", "21", "22",
"23",
                "24", "25", "26", "27", "28", "29", "30", "31", "32", "33",
"34",
                "35", "36", "37", "38", "39", "40", "41", "42", "43", "44",
"45",
                "46", "47","48", "49", "50", "51", "52", "53", "54", "55",
"56",
                "57", "58", "59", "60", "61", "62", "63"
            ],
            "MenuAccess": [
                "System",
                "Device",
                "Record",
                "Event",
                "Network"
            ],
            "SystemMenuAccess": [
                "DateTimeLanguage",
                "SystemManagement",
                "SystemLog",
                "BackupLog",
                "EventLog"
            ],
            "DeviceMenuAccess": [
                "CameraRegistration",
                "CameraSetup",
                "LiveSetup",
```

```
                "ChannelSetup",
                "DeviceFormat",
                "HDDAlarm",
                "iSCSI",
                "RAID",
                "Monitor"
            ],
            "RecordMenuAccess": [
                "RecordingSchedule",
                "NvrRecordSetup",
                "NetCamRecordSetup",
                "RecordOption"
            ],
            "EventMenuAccess": [
                "NvrSensorDetection",
                "NetCamSensorDetection",
                "NvrEventDetection",
                "NetCamEventDetection",
                "VideoLossDetection",
                "AlarmSchedule"
            ],
            "NetworkMenuAccess": [
                "NetworkInterface",
                "NetworkPort",
                "DDNS",
                "IPFilter",
                "SSL",
                "802.1x",
                "LiveStreaming",
                "SMTP",
                "EventMail",
                "GroupAndRecipientEmail",
                "SNMP",
                "DHCPServer"
            ],
            "RecordStartAccess": true,
            "RecordStopAccess": true,
            "PTZAccess": true,
            "AlarmOutputAccess": true,
            "ShutdownAccess": true
        }
```

```
    ]
}
```

8.4.3. Adding a user group

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=usergroups&action=add&UserGroupID=Group3&LiveChann
el=1,11,21&SearchChannel=2,12,22&BackupChannel=3,13,33
```

8.4.4. Updating a user group

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=usergroups&action=update&UserGroupID=Group3&Backup
Channel=13,21,3
```

8.4.5. Removing a user group

A user group can be deleted with the index number or the user group ID.

REQUEST

```
http://<Device IP>/ stw-
cgi/security.cgi?msubmenu=usergroups&action=remove&Index=3
```

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=usergroups&action=remove&UserGroupID=Group3
```

Chapter 9. Authority

9.1. Description

The **authority** submenu sets the access permissions.

> **NOTE** This chapter applies to NVR only.

Access level

Action	NVR
view	User
set	Admin

9.2. Syntax

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=
authority&action=<value>[&<parameter>=<value>...]
```

9.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view				Reads the authority settings
set	LiveAccess	REQ, RES	<bool> True, False	Whether to grant permission for live access
	PTZAccess	REQ, RES	<bool> True, False	Whether to grant permission for PTZ control
	RemoteAlarmOutput	REQ, RES	<bool> True, False	Whether to grant permission for remote alarm output control
	ShutDown	REQ, RES	<bool> True, False	Whether to grant permission to shut down
	RecordStartAccess	REQ, RES	<bool> True, False	Whether to grant permission for record start control
	RecordStopAccess	REQ, RES	<bool> True, False	Whether to grant permission for record stop control
	SearchAccess	REQ, RES	<bool> True, False	Whether to grant permission for search control
	BackupAccess	REQ, RES	<bool> True, False	Whether to grant permission for backup access

Action	Parameter	Request/ Response	Type/ Value	Description
	NetworkAccess	REQ, RES	<bool> True, False	Whether to grant permission for network access
	WebviewerAccess	REQ, RES	<bool> True, False	Whether to grant permission for Web viewer access
	AutoLogoutTime	REQ, RES	<enum> Off, 1m, 2m, 3m, 5m, 10m	Auto logout time
	IDManualInput	REQ, RES	<bool> True, False	Whether to grant permission to manually input the ID

9.4. Examples

9.4.1. Getting current permission settings

REQUEST

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=authority&action=view
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
LiveAccess=True
PTZAccess=False
RemoteAlarmOutput=False
ShutDown=False
RecordStopAccess=False
SearchAccess=False
BackupAccess=False
RecordStartAccess=False
NetworkAccess=True
WebviewerAccess=True
IDManualInput=False
AutoLogoutTime=OFF
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "LiveAccess": true,
    "PTZAccess": false,
    "RemoteAlarmOutput": false,
    "ShutDown": false,
    "RecordStopAccess": false,
    "SearchAccess": false,
    "BackupAccess": false,
    "RecordStartAccess": false,
    "NetworkAccess": true,
    "WebviewerAccess": true,
    "IDManualInput": false,
    "AutoLogoutTime": "OFF"
}
```

9.4.2. Setting the access permission

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=authority&action=set&LiveAccess=False&PTZAccess=Fa
lse&RemoteAlarmOutput=False&ShutDown=False&RecordStopAccess=False&SearchAcce
ss=False&BackupAccess=false
```

Chapter 10. Additional Password

10.1. Description

The **additionalpassword** submenu sets multiple passwords for the user.

> **NOTE** This chapter applies to NVR only.

Access level

Action	NVR
view	User
add, update	Admin
remove	Admin

10.2. Syntax

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=
additionalpassword&action=<value>[&<parameter>=<value>...]
```

10.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view	UserID	REQ	<string>	User ID
add/update	UserID	REQ, RES	<string>	User ID
	Enable	REQ, RES	<bool> True, False	• To enable a multi password rule
	PasswordIndex	REQ, RES	<int>	• Index for password
	Password	REQ, RES	<string>	• Additional password
	IsPasswordEncrypted	REQ	<bool> True, False	• If set to true, Password is encrypted using the public key provided by the **rsa** submenu of **security**.cgi, and sent as post payload. Refer to the Application Programmer's Guide.
remove	UserID	REQ	<string>	User ID

Action	Parameter	Request/ Response	Type/ Value	Description
	PasswordIndex	REQ	<csv>	Password Index that is to be removed **Note** All passwords will be removed for the corresponding user if PasswordIndex is not provided.

10.4. Examples

10.4.1. Getting additional password settings for all users

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=additionalpassword&action=view
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
UserID.User1.GroupID=Group1
UserID.User1.Enable=True
UserID.User2.GroupID=Group2
UserID.User2.Enable=False
UserID.User3.GroupID=Group2
UserID.User3.Enable=True
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "AdditionalPasswords": [
        {
            "UserID": "User1",
```

```
            "GroupID": "Group1",
            "Enable": true
        },
        {
            "UserID": "User2",
            "GroupID": "Group2",
            "Enable": false
        },
        {
            "UserID": "User3",
            "GroupID": "Group2",
            "Enable": true
        }
    ]
}
```

10.4.2. Getting additional password settings for user "User1"

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=additionalpassword&action=view&UserID=User1
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
UserID.User1.GroupID=Group1
UserID.User1.Enable=True
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "AdditionalPasswords": [
```

```
        {
            "UserID": "User1",
            "GroupID": "Group1",
            "Enable": true
        }
    ]
}
```

10.4.3. Adding a Password for a User

REQUEST

```
http://<Device IP>/stw-cgi/security.cgi?
msubmenu=additionalpassword&action=add&UserID=User1&Password=12345678
```

10.4.4. Updating Password

REQUEST

```
http://<Device IP/stw-
cgi/security.cgi?msubmenu=additionalpassword&action=update&UserID=User1&Pass
wordindex=1&Password=q1w2e3r4t5!
```

10.4.5. Removing all Passwords for a User

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=additionalpassword&action=remove&UserID=User1
```

10.4.6. Removing Password Index 1,2,3

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=additionalpassword&action=remove&UserID=User1&Pass
wordIndex=1,2,3
```

Chapter 11. Getting public key

11.1. Description

The **rsa** submenu is used to get the public key from the device. This can be used to encrypt the password information sent to the device.

> **NOTE** This chapter applies to network cameras only.

Access level

Action	Camera
view	Admin

11.2. Syntax

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=
rsa&action=<value>[&<parameter>=<value>...]
```

11.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view	PublicKeyFormat	REQ	<enum> PKCS1, X509	Optional request parameter to specify the RSA public key format. If not specified, the PKCS1 format is served.
	PublicKey	RES	<string>	RSA public key

11.4. Examples

> **NOTE** Please refer to the application programming guide for information on how to use this public key to encrypt passwords.

REQUEST

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=rsa&action=view
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
```

```
<Body>
```

```
PublicKey=-----BEGIN RSA PUBLIC KEY-----
MIIBCgKCAQEAugtcj1ql+mvkzK60Hph3an0Z/rdtZ/NF84m0TAsQuiDheGnN7dYJ
nZfRit5PcdugQO7XAkVq9DBy6kWrgrMqlzS9PwwEN7cBgFmyU/yJvpnZNrxlDLFB
ELlXgEYVih5yTSSoa6uWy8cSnGrnY1Ywymh8JGvuk0xZFc09eBCnIogQqydQb1AP
OnUqTx5JaCdnitYekHRWdyh1XY3wJnV6Ykb8hfnwzhrbz4P2bOCTW/ISE5hl2qvP
WrSBk+EEH2Wfcwfu2785iyu6mDzoCT64XcyOgscLkLhkg2IAXhoel+TNDdN0zzOX
dIpDg2Vi4s3ObPG4KSLnStXJJYIDx3CrRwIDAQAB
-----END RSA PUBLIC KEY-----
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "PublicKey": "-----BEGIN RSA PUBLIC KEY-----
\nMIIBCgKCAQEAugtcj1ql+mvkzK60Hph3an0Z/rdtZ/NF84m0TAsQuiDheGnN7dYJ\nnZfRit5P
cdugQO7XAkVq9DBy6kWrgrMqlzS9PwwEN7cBgFmyU/yJvpnZNrxlDLFB\nELlXgEYVih5yTSSoa6
uWy8cSnGrnY1Ywymh8JGvuk0xZFc09eBCnIogQqydQb1AP\nOnUqTx5JaCdnitYekHRWdyh1XY3w
JnV6Ykb8hfnwzhrbz4P2bOCTW/ISE5hl2qvP\nWrSBk+EEH2Wfcwfu2785iyu6mDzoCT64XcyOgs
cLkLhkg2IAXhoel+TNDdN0zzOX\ndIpDg2Vi4s3ObPG4KSLnStXJJYIDx3CrRwIDAQAB\n-----
END RSA PUBLIC KEY-----\n"
}
```

Example of when key format is specified.

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=rsa&action=view&PublicKeyFormat=X509
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
PublicKey=-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxp5XPQay2FVUJZXCV2K2
7Wv7uLtZ3vIwsSiAHVJZwSmAQV7H23ElmBCNEWCk96mdjonZnHpQ0mWj3hsDkO48
qGnELbsrqfTuUF5Ulze7+f34aX/Mg9pwbOruZE3CRbcsxc2JTTbm0sLoVnSV7pPn
Lg/r4dzp7ll3fL4WfKere/sXmRdeZ+2ugVzrCGSovOX4madkAtwCEszOZedIWe85
AkDN42Aw11sknn66EkDZAMVrpI5g0nfrdUYTKxh/e+LAVOfMSHdFaMht4rSTaXN7
z+RxPh5Ro0UN5Ha9buNtiXUB4VkjV440/Be13njHt+d5HZduGh0WhggIjgyTJGsN
7wIDAQAB
-----END PUBLIC KEY-----
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxp5XPQay2FVUJZXCV2K2\n7Wv7uLtZ
3vIwsSiAHVJZwSmAQV7H23ElmBCNEWCk96mdjonZnHpQ0mWj3hsDkO48\nqGnELbsrqfTuUF5Ulz
e7+f34aX/Mg9pwbOruZE3CRbcsxc2JTTbm0sLoVnSV7pPn\nLg/r4dzp7ll3fL4WfKere/sXmRde
Z+2ugVzrCGSovOX4madkAtwCEszOZedIWe85\nAkDN42Aw11sknn66EkDZAMVrpI5g0nfrdUYTKx
h/e+LAVOfMSHdFaMht4rSTaXN7\nz+RxPh5Ro0UN5Ha9buNtiXUB4VkjV440/Be13njHt+d5HZdu
Gh0WhggIjgyTJGsN\n7wIDAQAB\n-----END PUBLIC KEY-----\n"
}
```

Chapter 12. Configure default camera user credentials in NVR

12.1. Description

The **camerausers** submenu is used to configure camera's default set of username and password in NVR. If NVR discovers some cameras or does connection test, it automatically tries to connect with camera with these configured user credentials.

> **NOTE** This chapter applies to NVR only.

Access level

Action	NVR
view	User
add/update	Admin
remove	Admin

12.2. Syntax

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=
camerausers&action=<value>[&<parameter>=<value>...]
```

12.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view	UserID	REQ	<string>	
	IsInitPasswordSet	RES	<bool> True, False	Initial password status whether it is set or not
add/update	Index	REQ, RES	<int>	Index to be added
	UserID	REQ, RES	<string>	ID to try connecting with its camera
	Password	REQ, RES	<string>	Password to try connecting with its camera
	IsPasswordEncrypted	REQ	<bool> True, False	When this is set to true, password is encrypted using the public key obtained from the **rsa** submenu of security.cgi, and sent as payload content for the POST command.

Action	Parameter	Request/ Response	Type/ Value	Description
set	InitPassword	REQ	<string>	Initializes camera with this password
	IsPasswordEncrypted	REQ	<bool> True, False	When this is set to True, InitPassword parameter is not sent and instead the password is encrypted using the public key obtained from the **rsa** submenu of security.cgi, and sent as payload content for the POST command.
remove	Index	REQ	<int>	Index to be removed

12.4. Examples

12.4.1. Viewing a user

REQUEST

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=camerausers&action=view
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "Users": [
        {
            "Index": 0,
            "UserID": "admin"
        },
        {
            "Index": 1,
            "UserID": "admin"
        },
        {
            "Index": 2,
            "UserID": "root"
        }
    ]
```

```
}
```

NOTE Please refer to the application programming guide for information on how to use this public key to encrypt passwords.

12.4.2. Adding a user

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=camerausers&action=add&UserId=admin&password=12345
678&IsPasswordEncrypted=false
```

12.4.3. Updating a user

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=camerausers&action=update&Index=0&UserId=admin&pas
sword=12345678&IsPasswordEncrypted=False
```

12.4.4. Removing a user

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=camerausers&action=remove&Index=1
```

Chapter 13. Getting Client Mutual Authenticate Status

13.1. Description

The **clienthttpsstatus** submenu is used to get the client's mutual authentication status from the device.

> **NOTE**
> This chapter applies to network cameras only.
> Attribute that checks for the client certificate authentication supports:
> "attributes/Security/Support/ClientCertificateAuthentication"

Access level

Action	Camera
view	User

13.2. Syntax

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=
clienthttpsstatus&action=<value>[&<parameter>=<value>...]
```

13.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view				Read client's mutual certificate status

Action	Parameter	Request/ Response	Type/ Value	Description
	ClientHttpsStatus	RES	<enum> NO_HTTPS, HTTPS_WITHOUT_CLIENT_CERT, HTTPS_WITH_INVALID_CLIENT_CERT, HTTPS_WITH_VALID_CLIENT_CERT	Current client's HTTPS connection status • NO_HTTPS: The client uses no HTTPS • HTTPS_WITHOUT_CLIENT_CERT: The client uses HTTPS, but the device does not use mutual authentication • HTTPS_WITH_INVALID_CLIENT_CERT: The device has mutual authentication enabled, but the client's certificate is invalid • HTTPS_WITH_VALID_CLIENT_CERT: The device has mutual authentication enabled and the client's certificate is valid

13.4. Examples

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=clienthttpsstatus&action=view
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

ClientHttpsStatus=NO_HTTPS

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

{

"ClientHttpsStatus": "NO_HTTPS"

}

Chapter 14. Getting TLS Configuration

14.1. Description

The **tlsversion** submenu is used to get the TLS configuration from the device.

> **NOTE** This chapter applies to network cameras only.

Access level

Action	Camera
view	Admin
set	Admin

14.2. Syntax

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=
tlsversion&action=<value>[&<parameter>=<value>...]
```

14.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view				Reads TLS configuration.
	Version	REQ	<csv> TLSv1_0, TLSv1_1, TLSv1_2, TLSv1_3	TLS Version Getting the configuration of specific TLS version
	Version.#.SupportedCipherMode	RES	<csv> Compatible, Secure	Supported cipher mode • Compatible: Supports a compatible cipher suite • Secure: Supports a secure cipher suite
set	Version.#.Enable	REQ, RES	<bool> True, False	Enables or disables TLS version If a specific TLS version to be enabled does not support Secure mode, **CipherMode** will automatically change to Compatible mode

Action	Parameter	Request/ Response	Type/ Value	Description
	CipherMode	REQ, RES	<enum> Compatible, Secure	Cipher mode • Compatible: Device uses a compatible cipher suite • Secure: Device uses a more secure cipher suite To use a specific mode, all versions must support the mode.

14.4. Examples

14.4.1. Getting all versions of the TLS configurations

REQUEST

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=tlsversion&action=view
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
Version.TLSv1_0.Enable=False
Version.TLSv1_0.SupportedCipherModes=Compatible
Version.TLSv1_1.Enable=False
Version.TLSv1_1.SupportedCipherModes=Compatible
Version.TLSv1_2.Enable=True
Version.TLSv1_2.SupportedCipherModes=Compatible,Secure
Version.TLSv1_3.Enable=True
Version.TLSv1_3.SupportedCipherModes=Compatible,Secure
CipherMode=Secure
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "Versions": [
        {
            "Version": "TLSv1_0",
            "Enable": false,
            "SupportedCipherModes": [
                "Compatible"
            ]
        },
        {
            "Version": "TLSv1_1",
            "Enable": false,
            "SupportedCipherModes": [
                "Compatible"
            ]
        },
        {
            "Version": "TLSv1_2",
            "Enable": true,
            "SupportedCipherModes": [
                "Compatible",
                "Secure"
            ]
        },
        {
            "Version": "TLSv1_3",
            "Enable": true,
            "SupportedCipherModes": [
                "Compatible",
                "Secure"
            ]
        }
    ],
    "CipherMode": "Secure"
}
```


14.4.2. Getting ‘TLSv1_3’ configuration

REQUEST

```
http://<Device IP>/stw-
```

```
cgi/security.cgi?msubmenu=tlsversion&action=view&Version=TLSv1_3
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
Version.TLSv1_3.Enable=True
Version.TLSv1_3.SupportedCipherModes=Compatible,Secure
CipherMode=Secure
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "Versions": [
        {
            "Version": "TLSv1_3",
            "Enable": true,
            "SupportedCipherModes": [
                "Compatible",
                "Secure"
            ]
        }
    ],
    "CipherMode": "Secure"
}
```

14.4.3. Enabling TLS v1.2 and v1.3

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=tlsversion&action=set&Version.TLSv1_2.Enable=True&
Version.TLSv1_3.Enable=True
```

14.4.4. Setting cipher mode to compatible mode

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=tlsversion&action=set&CipherMode=Compatible
```

Chapter 15. Getting Camera's validation status from NVR

15.1. Description

The **cameravalidationstatus** submenu is used to get the camera's validation status from NVR.

> **NOTE** This chapter applies to NVR only.

Access level

Action	NVR
view	User

15.2. Syntax

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=
cameravalidationstatus&action=<value>[&<parameter>=<value>...]
```

15.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view				Reads camera validation status
	Channel.#.Connected	RES	<bool> True, False	Camera connection status
	Channel.#.CameraValidationStatus	RES	<enum> UNKNOWN, HTTP, OTHER_CERT, CHANGED_CERT, INVALID_DEVICE_CERT, VALID_DEVICE_CERT	Current channels connection status **Note** Valid status is only supported for cameras connected using SUNAPI in NVR

15.4. Examples

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=cameravalidationstatus&action=view
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
Channel.0.Connected=True
Channel.0.CameraValidationStatus=HTTP
Channel.1.Connected=True
Channel.1.CameraValidationStatus=HTTP
Channel.2.Connected=True
Channel.2.CameraValidationStatus=HTTP
....
Channel.3.Connected=False
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "cameravalidationstatus": [
        {
            "Channel": 0,
            "Connected": true,
            "CameraValidationStatus": "HTTP"
        },
        {
            "Channel": 1,
            "Connected": true,
            "CameraValidationStatus": "HTTP"
        },
        {
            "Channel": 2,
            "Connected": true,
            "CameraValidationStatus": "HTTP"
        },
        ...
    ]
```

```
}
```

Chapter 16. CA Certificate Settings

16.1. Description

The **cacertificate** submenu handles CA certificates.

Access level

Action	Camera
view	Admin
install	Admin
remove	Admin

16.2. Syntax

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=
cacertificate&action=<value>[&<parameter>=<value>...]
```

16.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view				Reads the SSL (HTTPS) settings
	Certificate.#.CertificateName	RES	<string>	Certificate name
	Certificate.#.Type	RES	<enum> Unique, Public, SelfSigned	Type of the certificate
	Certificate.#.Subject	RES	<string>	Subject of the certificate
	Certificate.#.SubjectAlternativeName	RES	<string>	Subject alternative name (SAN)
	Certificate.#.Issuer	RES	<string>	Issuer
	Certificate.#.IssueDate	RES	<string>	Issued date
	Certificate.#.ExpiryDate	RES	<string>	Expiry date
	Certificate.#.Version	RES	<string>	Version
	Certificate.#.SerialNumber	RES	<string>	Serial number
	Certificate.#.Signature	RES	<string>	Signature

Action	Parameter	Request/ Response	Type/ Value	Description
	Certificate.#.Thumbprint	RES	<string>	Thumbprint
	Certificate.#.IsRemovable	RES	<bool>	Whether the certificate can be deleted
remove	CertificateName	REQ	<string>	Certificate name
install				POST method

16.4. Examples

16.4.1. Getting the CA certificates

REQUEST

```
http://<Device IP>/stw-cgi/security.cgi?msubmenu=cacertificate&action=view
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
Certificate.1.CertificateName=HTW_rootca
Certificate.1.Type=Unique
Certificate.1.Subject=/C=KR/O=Hanwha Vision/OU=Security Solution/CN=Hanwha
Vision Private Root CA 2
Certificate.1.SubjectAlternativeName=-
Certificate.1.Issuer=/C=KR/O=Hanwha Vision/OU=Security Solution/CN=Hanwha
Vision Private Root CA 2
Certificate.1.IssueDate=Feb 17 04:15:52 2020 GMT
Certificate.1.ExpiryDate=Feb 2 04:15:52 2080 GMT
Certificate.1.Version=V3
Certificate.1.SerialNumber=00 85 BE B7 49 3A 83 3A E7
Certificate.1.Signature=sha256WithRSAEncryption
Certificate.1.Thumbprint=3676ae9bd6ebb4f3543b00c08898d17b7fa96b7e61726a4bd5f
60c09062c4fce
Certificate.1.IsRemovable=False
Certificate.2.CertificateName=helloCA
Certificate.2.Type=Public
Certificate.2.Subject=/C=FR/ST=Radius/L=Somewhere/O=Example
Inc./emailAddress=admin@example.org/CN=Example Certificate Authority
```

```
Certificate.2.SubjectAlternativeName=-
Certificate.2.Issuer=/C=FR/ST=Radius/L=Somewhere/O=Example
Inc./emailAddress=admin@example.org/CN=Example Certificate Authority
Certificate.2.IssueDate=Mar 9 01:46:25 2020 GMT
Certificate.2.ExpiryDate=Jan 16 01:46:25 2030 GMT
Certificate.2.Version=V3
Certificate.2.SerialNumber=00 62 0A 2B 24 D9 7E DD 53 B2 B5 F6 71 DF 72 92
08 F0 7F 25 9A
Certificate.2.Signature=sha256WithRSAEncryption
Certificate.2.Thumbprint=68b8a185ae17600815b25cd1c42ecbbd62f2b962a62787376fc
53f9efb211391
Certificate.2.IsRemovable=True
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
    "Certificates": [
        {
            "CertificateName": "HTW_rootca",
            "Type": "Unique",
            "Subject": "/C=KR/O=Hanwha Vision/OU=Security Solution/CN=Hanwha
Vision Private Root CA 2",
            "SubjectAlternativeName": "-",
            "Issuer": "/C=KR/O=Hanwha Vision/OU=Security Solution/CN=Hanwha
Vision Private Root CA 2",
            "IssueDate": "Feb 17 04:15:52 2020 GMT",
            "ExpiryDate": "Feb 2 04:15:52 2080 GMT",
            "Version": "V3",
            "SerialNumber": "00 85 BE B7 49 3A 83 3A E7 ",
            "Signature": "sha256WithRSAEncryption",
            "Thumbprint":
"3676ae9bd6ebb4f3543b00c08898d17b7fa96b7e61726a4bd5f60c09062c4fce",
            "IsRemovable": false
        },
        {
            "CertificateName": "helloCA",
            "Type": "Public",
```

```
            "Subject": "/C=FR/ST=Radius/L=Somewhere/O=Example
Inc./emailAddress=admin@example.org/CN=Example Certificate Authority",
            "SubjectAlternativeName": "-",
            "Issuer": "/C=FR/ST=Radius/L=Somewhere/O=Example
Inc./emailAddress=admin@example.org/CN=Example Certificate Authority",
            "IssueDate": "Mar 9 01:46:25 2020 GMT",
            "ExpiryDate": "Jan 16 01:46:25 2030 GMT",
            "Version": "V3",
            "SerialNumber": "00 62 0A 2B 24 D9 7E DD 53 B2 B5 F6 71 DF 72 92
08 F0 7F 25 9A ",
            "Signature": "sha256WithRSAEncryption",
            "Thumbprint":
"68b8a185ae17600815b25cd1c42ecbbd62f2b962a62787376fc53f9efb211391",
            "IsRemovable": true
        }
    ]
}
```

16.4.2. Installing CA certificate

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=cacertificate&action=install
```

When requesting a certificate to be installed, data should be sent via the POST method in the following format.

The certname value is the certificate name. The certlength value is the certificate data size. The certdata value is the certificate data. The keylength is the key data size. The keydata value is the key data.

```
<SetHTTPSData>
    <PublicCertName>certname</PublicCertName>
    <CertLength>certlength</CertLength>
    <CertData>certdata</CertData>
</SetHTTPSData>
```

CURL command

The certificate installation can be tested with CURL as below. To learn about CURL, please refer to http://curl.haxx.se.

NOTE To get a JSON response, add the -H "Accept: application/json" header to the request.

```
curl -v --digest -u <userid>:<password> --data-urlencode @cert.xml
"http://<Device IP>/stw-
cgi/security.cgi?msubmenu=cacertificate&action=install" -H "Expect:"
```

The above command will produce a request to the device that appears as below:

```
POST /stw-cgi/security.cgi?msubmenu=ssl&action=install HTTP/1.1
Content-Length: 3775
Content-Type: application/x-www-form-urlencoded
```

TEXT RESPONSE

```
OK
```

JSON RESPONSE

```
{
    "Response": "Success"
}
```

16.4.3. Deleting a certificate

To delete a certificate with the **remove** action, the **CertificateName** parameter must be set at the same time.

REQUEST

```
http://<Device IP>/stw-
cgi/security.cgi?msubmenu=cacertificate&action=remove&CertificateName=certna
me
```