SharkTrust Protocol Specification

1 Introduction

1.1 SharkTrust, a Combined DNS and PKI Service

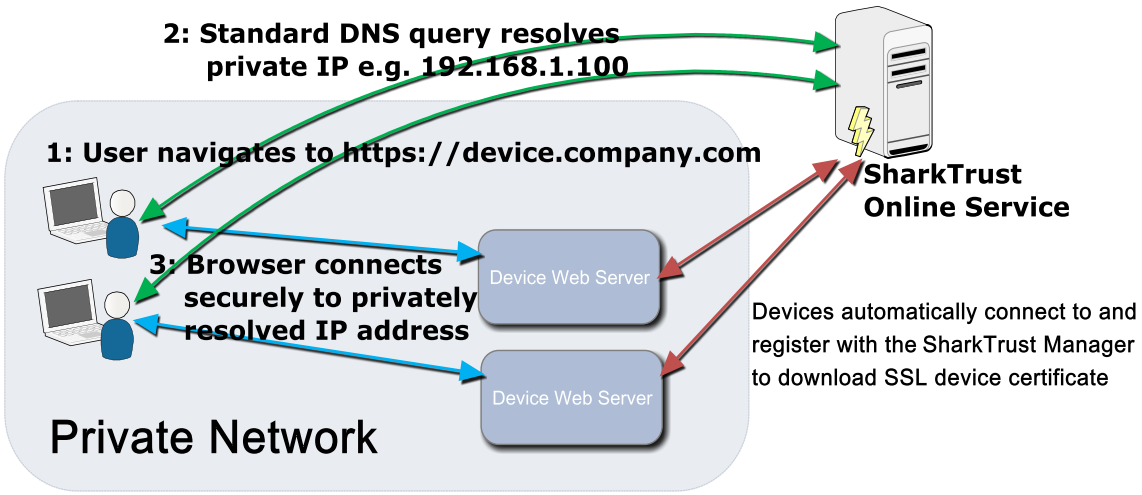
SharkTrust is a solution that includes DNS services and automatic certificate management for devices functioning as web servers. The SharkTrust service enables users, with limited to no understanding of Public Key Infrastructure (PKI), to trust and securely access all devices on the local network. The SharkTrust service completely automates the Public Key Infrastructure.

SharkTrust lets device manufacturers integrate our automatic DNS and certificate management solution as an option for the end customer. The SharkTrust solution, which is designed for resolving DNS for private networks and automates certificate management for devices, provides an easy to use and configuration less solution for the device manufacturer's end customers.

An added benefit of SharkTrust is that web server enabled devices will not need a manually configured static IP address when using the SharkTrust service but can instead connect to the company's network and get a dynamic IP using DHCP. The device then connects to the online SharkTrust service, registers the private IP address, and downloads the SSL certificate. The user (end customer) learns about new devices by connecting to the online SharkTrust manager's web interface, which lists all registered devices connected to the end customer's private network. A user can then simply click the link to one of the private devices which makes the browser connect securely to the local device.

1.3 Using the SharkTrust Service

In short, a device with an embedded web server is connected to a private network with Internet access. The device connects to the online SharkTrust service and downloads an SSL certificate for use with its internal web server. The SSL certificate downloaded by the device is signed for a domain name, and to make sure the browser trusts the device, the online SharkTrust service provides DNS services for the connected devices. As an example, a device may have the name "device" and the domain name may be "company.com". The fully qualified name for the device will then be device.company.com and a browser may navigate to the device's internal web server by navigating to https://device.company.com



The SharkTrust service groups a set of devices into a group called a zone. Each zone registered with the SharkTrust service is a domain name, and each domain name may be used by multiple end customers.

A zone used by multiple end customers must be managed by the device manufacturer. A device manufacturer may also let the end customers manage their own zones. This feature is particularly useful when you have large companies as customers.

1.3 Real Time Logic's Online SharkTrust Testing Service

Real Time Logic provides a testing service device manufacturers may use for testing the SharkTrust service and for development purposes.

Use the service as follows:

1. Use an existing domain name and change the name servers to ns1.realtimelogic.com and ns2.realtimelogic.com. Alternatively, register a free domain name for testing purposes.
2. Wait till the DNS settings replicate across the Internet.
3. Navigate to https://sharktrust.realtimelogic.com/, sign up, and register your domain.
4. Download the SharkTrust client example SharkTrust-Client.c.

1.4 The Binary Protocol

The SharkTrust Binary Protocol is designed to be easy to implement in any type of device. Although the protocol is initiated over HTTPS, an HTTP client stack is not required in the device since HTTP is only used for the request. The SharkTrust server response is sent as a binary protocol without adding an HTTP header, thus making it very easy to implement in a device using C code. You may also enable a standard HTTP response as explained in section 5. SharkTrust security is discussed in section 8.

2 Acronyms and Definitions:

- **TCP/IP:** Transmission Control Protocol/Internet Protocol.
- **HTTP:** Hypertext Transfer Protocol
- **HTTP(s):** HTTP over SSL or HTTP Secure
- **Zone Administrator:** an administrator for a company (customer) using the SharkTrust service. The zone is a registered domain name owned by the customer and used with the SharkTrust service.
- **SharkTrust Team Member:** a root user with full access to the SharkTrust service and all zones managed by the service.
- **Uninitialized:** the initial state of the device. The device remains in this state until it receives a DeviceKey via the Register command.
- **Active**: the device has a DeviceKey and is in the active state.

3 SharkTrust Binary Protocol Overview

Protocol

- Device to SharkTrust service (request) is sent as an HTTPS header. The request is sent to https://certA.realtimelogic.com/device/. The HTTPS server at this end point is signed with Real Time Logic's RSA root certificate and with Real Time Logic's EC root certificate. The device should validate this endpoint by storing either Real Time Logic's RSA root certificate or Real Time Logic's EC root certificate in the device.

- SharkTrust service to device (response) uses a binary protocol. All binary responses start with the following preamble:

Four Byte Packet Header for Binary Response:

1 byte	0xFF (Header response ID 1)
1 byte	0x55 (Header response ID 2)
1 byte	Response status
1 byte	Reserved. Data must be ignored.

The response starts with the sequence 0xFF,0x55 to indicate that this is a SharkTrust binary response. Any other sequence indicates that the device is not communicating with the SharkTrust service. A proxy may, for example, interfere with the SharkTrust communication.

The response data length is 4 bytes if the status code is a non zero value (an error). The response data length depends on the message type when the response status is zero (success).

Response Codes:

0. **Success:** See message types below for remaining response data.
1. **Forbidden:** The client used an unknown or invalid X-Key.
2. **Unknown:** The client used an unknown or invalid X-Dev ID. The client must send a new registration request.
3. **Processing:** SharkTrust is working on the certificate. This response code may be sent if SharkTrust does not have a certificate when the client sends a GetCertificate request. The client should poll for updates, by sending GetCertificate messages, with a one to two minute poll interval.
4. **ServerError**: SharkTrust is in maintenance mode. The device should repeatedly retry connecting with a timeout value of no shorter than five minutes.
5. **ClientError:** the server did not understand the data sent from the client (the device).

4 Message Types

You may use any of the following message types when communicating with the online SharkTrust service; however, the only two required message types are Register and GetCertificate.

Register

The Register command is sent by an uninitialized device. The Register command registers the device with the SharkTrust service.

HTTP Request Headers:

X-Command	Register
X-Key	The 64 byte long RegistrationKey received after signing up for the SharkTrust service. The key is unique to the registered domain name.
X-Name	The sub domain name unique to the device. The SharkTrust service may require the device to use another name if the name is already in use. See the response below for details.
X-Info	Optional information describing the product in more detail. This information is made available to the Zone Administrator and to registered users of the zone.
X-IpAddress	The IP address of the device as registered by the device's Ethernet port either by static configuration or DHCP. The format must be such that it can be used by DNS Bind -- e.g. 192.168.1.100

Binary Response:

4 bytes	Binary header including status=0
20 bytes	A private DeviceKey (X-Dev) that must be used for subsequent communication with the SharkTrust service. The key is a hex representation of a 10 byte binary key and can be used "as is" when sending HTTP requests to the server. The device must persistently and securely store the key in device memory and change device state from uninitialized to active.
2 bytes	16 bit number: Length of the next record (sub domain name)
N bytes	The sub domain name selected by the SharkTrust service. This name will be the same as X-Name if the name has not been previously registered by another device. This is a null-terminated string.

GetCertificate

Download the certificate and private key from the SharkTrust service. This command also includes the SetIpAddress command, thus updating the DNS if required. The SharkTrust service requires the device to be in "active" state when sending this command.

HTTP Request Headers:

X-Command	GetCertificate
X-Key	The 64 byte long RegistrationKey received after signing up for the SharkTrust service.
X-Dev	The 20 byte long DeviceKey received as a response to the Register command.
X-CertType	Shark | X509. Return a combined binary SharkSSL certificate and private key or return an X.509 PEM certificate and a PEM formatted private key. This header is optional and defaults to "Shark" (SharkSSL) certificate.
X-IpAddress	The IP address of the device as registered by the device's Ethernet port either by static configuration or DHCP. The format must be such that it can be used by DNS Bind -- e.g. 192.168.1.100

Binary Response (when CertType is Shark):

4 bytes	Binary header including status=0
4 bytes	32 bit number: how many days in seconds till the certificate expires and until a new GetCertificate request must be sent.
2 bytes	16 bit number: Length of the next record (SharkSSL certificate)
N bytes	Binary SharkSSL certificate

Binary Response (when CertType is X509):

4 bytes	Binary header including status=0
4 bytes	32 bit number: how many days in seconds till the certificate expires and until a new GetCertificate request must be sent.
2 bytes	16 bit number: Length of the next record (X.509 certificate)
N bytes	X.509 certificate
2 bytes	16 bit number: Length of the next record (X.509 private key)
N bytes	X.509 private key

SetIpAddress

Send the device IP address to the SharkTrust service and update the DNS. This command may be used as an alternative to command GetCertificate when the client is using DHCP and needs to update the IP address, but does not need a new certificate.

HTTP Request Headers:

X-Command	SetIpAddress
X-Key	The 64 byte long RegistrationKey received after signing up for the SharkTrust service.
X-Dev	The 20 byte long DeviceKey received as a response to the Register command.
X-IpAddress	The IP address of the device as registered by the device's Ethernet port either by static configuration or DHCP. The format must be such that it can be used by DNS Bind -- e.g. 192.168.1.100

Binary Response:

4 bytes	Binary header including status=0

GetWAN

You may query the online service for the WAN address in any device state.

HTTP Request Headers:

X-Command	GetWAN
X-Key	The 64 byte long RegistrationKey received after signing up for the SharkTrust service.

Binary Response:

4 bytes	Binary header including status=0
2 bytes	16 bit number: Length of the next record (IP address)
N bytes	The IP address in string notification

GetDN

Get the full Domain Name (DN), including the device's sub domain name. This request requires that the device has successfully downloaded a certificate using the GetCertificate command. The request will otherwise fail.

HTTP Request Headers:

X-Command	GetDN
X-Key	The 64 byte long RegistrationKey received after signing up for the SharkTrust service.
X-Dev	The 20 byte long DeviceKey received as a response to the Register command.

Binary Response:

4 bytes	Binary header including status=0
2 bytes	16 bit number: Length of the next record (domain name)
N bytes	The full domain name

5 Enable Standard HTTP Response

The SharkTrust service, by default, sends the binary response immediately without sending an HTTP response header. You may enable a standard HTTP response by setting the header X-Response to HTTP-BIN. The SharkTrust service will then send a standard HTTP response with the HTTP code set to 202 followed by the binary data. The purpose with this feature is to make it easy to use the SharkTrust service in devices that include an HTTP stack and where the developer chooses to use the HTTP client as a method for communicating with the online SharkTrust service.

Additional HTTP Header parameter:

X-Response	HTTP-BIN

6 SharkTrust Service URL Used By Devices

The device gets the certificate and registers it's IP address with the online SharkTrust service by connecting to **https://[online-service]/device/**, where "online-service" is the domain name used by your online service. The online testing service provided by Real Time Logic is configured to use two URLs:

1. https://SharkTrust.realtimelogic.com/device/
   RSA certificate signed by Let's Encrypt.
2. https://SharkTrustEC.realtimelogic.com/device/
   EC Certificate signed by Real Time Logic's EC root certificate.

7 Networks Requiring Proxy Configurations

Some company networks may require that all Internet connectivity is routed via a proxy. All SharkTrust commands may be routed via a proxy since the SharkTrust protocol is based on HTTPS. You may choose to include proxy/socks configuration as an option in your product; however, a proxy requires configuration including credentials for proxy authentication. Such configuration defeats the purpose with the hassle free SharkTrust concept and makes it tedious and difficult for the end user to use the product. A solution for device manufacturers is to instead suggest for their end customers to have a separate network for devices, in which no proxy configuration is necessary.

8 SharkTrust Security Considerations

The SharkTrust service is designed to provide a secure mechanism for certificate management without requiring PKI configuration. The security is provided by the SharkTrust service in combination with the use of the SharkTrust binary protocol. Each device gets its own unique certificate/private-key combo, in which the certificate is signed for the device's sub-domain name. The online service makes sure there are no sub-domain name conflicts and security issues.

The SharkTrust binary protocol's security works as follows. The device (client) uses public-key cryptography to authenticate the SharkTrust service. The SharkTrust service uses a "zone key" in combination with a "device-key" to authenticate the device. The device must have the zone-key pre-installed in the device when the device manufacturer is in control of the zone. Otherwise, the end customer must configure the zone-key for the domain used. A device must initially register with the online service, using a valid zone-key, and download a device-key. The device-key must be stored in persistent device memory (or file system).

Device Security Considerations

The zone-key (X-Key) and device-key (X-Dev) must be stored in a location on the device that prevents extraction via direct physical attacks. See the Securing Edge Node whitepaper for suggestions on how to keep information private in a device.

The downloaded certificate does not need to be protected, but the associated private key must be protected in the same way as the zone-key and device-key if the designer chooses to store the certificate/private-key combo persistently on the device. Our suggestion is to download the certificate/private-key when the device boots and to store the certificate/private-key in RAM memory only.

9 Design Considerations

A device should preferably have logic that creates a unique sub-domain name, which does not conflict with other devices on the same zone. A suggestion is to use the company's unique part of the MAC address as the sub-domain name. The online service makes sure no device can use a name that is already registered and will rename the sub-domain name if a conflict is found. The sub-domain name in the Register command response message is set to the name selected by the online service. If you hard code the name to, for example, 'device', the online service will name each device as 'device1', 'device2', and so on.

A device only needs to implement the two commands, Register and GetCertificate. Register is sent when the device has no device-key or if the online service has invalidated the key and sends an "Unknown" error as response to GetCertificate. The GetCertificate may be sent each time the device boots to simplify device logic. The downloaded certificate/key combo does not need to be stored in persistent memory. A device that is powered on for more than the certificate validity must send a new GetCertificate request before the certificate expires. Let's Encrypt signed certificates are valid for 90 days. The online service updates the certificate 22 days prior to expiration date. A device can create a rudimentary clock, for example via interrupts, that decrements the expiration time provided in the GetCertificate response.