Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 31, 2018

E. Lear
O. Friel
N. Cam-Winget
Cisco Systems

June 29, 2018

Bootstrapping Key Infrastructure over EAP
draft-lear-eap-teap-brski-00

Abstract

In certain environments, in order for a device to establish any layer three communications, it is necessary for that device to be properly credentialed. This is a relatively easy problem to solve when a device is associated with a human being and has both input and display functions. It is less easy when the human, input, and display functions are not present. To address this case, this memo specifies extensions to the Tunnel Extensible Authentication Protocol (TEAP) method that leverages Bootstrapping Remote Secure Key Infrastructures (BRSKI) in order to provide a credential to a device at layer two.

Table of Contents

1. Introduction

[I-D.ietf-anima-bootstrapping-keyinfra] (BRSKI) specifies a means to provision credentials to be used as credentials to operationally access networks. It was designed as a standalone means where some limited access to an IP network is already available. This is not always the case. For example, IEEE 802.11 networks generally require authentication prior to any form of address assignment. While it is

possible to assign an IP address to a device on some form of an open network, or to accept some sort of default credential to establish initial IP connectivity, the steps that would then follow might well require that the device is placed on a new network, requiring reseting all layer three parameters.

A more natural approach in such cases is to more tightly bind the provisioning of credentials with the authentication mechanism. One such way to do this is to make use of the Extensible Authentication Protocol (EAP) [RFC3748] and the Tunnel Extensible Authentication Protocol (TEAP) method [RFC7170]. Thus we define new TEAP Type-Length-Value (TLV) objects that can be used to transport the BRSKI protocol messages within the context of a TEAP TLS tunnel.

[RFC7170] discusses the notion of provisioning peers. Several different mechanisms are available. Section 3.8 of that document acknowledges the concept of not initially authenticating the outer TLS session so that provisioning may occur. In addition, exchange of multiple TLV messages between client and EAP server permits multiple provisioning steps.

1.1. Terminology

The reader is presumed to be familiar with EAP terminology as stated in [RFC3748]. In addition, the following terms are commonly used in this document.

- BRSKI: Bootstrapping Remote Secure Key Infrastructures, as defined in [I-D.ietf-anima-bootstrapping-keyinfra]. The term is also used to refer to the flow described in that document.
- EST: Enrollment over Secure Transport, as defined in [RFC7030].
- voucher: a signed json object as defined in [RFC8366].

2. TEAP BRSKI Architecture

The TEAP BRSKI architecture is illustrated in Section 3. The device talks to the TEAP server via the Authenticator as per any normal EAP exchange. There is no need for an inner EAP method server, and there is no explicit EAP method type defined for BRSKI.

The architecture illustrated shows the TEAP server and Registrar function as being two logically separate entities, however the BRSKI Registrar functionality may be integrated into the TEAP server. The device is not explicitly aware of where the Registrar functionality is deployed when executing BRSKI inside a TEAP tunnel. Note that the device may connect directly to the Registrar for the purposes of

certificate reenrollment, but this happens outside the context to 801.1X and TEAP authentication.

The Registrar in turn communicates with the BRSKI MASA service for the purposes of getting signed vouchers. [[TODO: I guess we should mention TEAP server talking to vendor default Registrar in the cloud]]

The Registrar also comunicates with a Certificate Authority in order to issue LDevIDs. The architecture shows the Registrar and CA as being two logically separate entities, however the CA may be integrated into the Registrar. The device is not explicitly aware of whether the CA and Registrar functions are integrated.

```
+--------+     +---------+     +--------+     +---------+     +------+
|        |     |         |     |        |     |         |<--->| MASA |
|        |     | Authen- |     |  TEAP  |     | BRSKI   |     +------+
| Device |<--->| ticator |<--->| server |<--->|Registrar|
|        |     |         |     |        |     |         |     +------+
|        |     |         |     |        |     |         |<--->|  CA  |
+--------+     +---------+     +--------+     +---------+     +------+
```

3. BRSKI Bootstrap and Enroll Operation

This section summarises the current BRSKI operation. The BRSKI flow assumes the device has an IDevID and has a manufacturer installed trust anchor that can be used to validate the BRSKI voucher. The BRSKI flow compromises serveral main steps from the perspective of the device:

- Step 1: Device discovers the Registrar

- Step 2: Device establishes provisional TLS connection to Registrar

- Step 3: Device sends voucher request message and receives signed voucher response

- Step 4: Device validates voucher and validates provisional TLS connection to Registrar

- Step 5: Device sends Voucher Status Telemetry message

- Step 6: Device downloads additional local domain CA information

- Step 7: Device downloads Certificate Signing Reqeust (CSR) attributes

o Step 8: Device does an EST enroll to obtain an LDevID

o Step 9: Device sends Enrollment Status Telemetry message

o Step 10: Device periodically reenrolls via EST to refresh its LDevID

Most of the operational steps require the device, and thus its internal state machine, to automatically complete the next step without being explicitly instructed to do so by the Registrar. For example, the Registrar does not explicitly tell the device to download additional local domain CA information, or to do an EST enroll to obtain an LDevID.

3.1. Executing BRSKI in a TEAP Tunnel

This section outlines how the main BRSKI steps outlined above map to TEAP, and how BRSKI and enrollment can be accomplished inside a TEAP TLS tunnel. The following new TEAP TLVs are introduced:

o BRSKI-VoucherRequest

o BRSKI-Voucher

o BRSKI-VoucherStatus

o EnrollmentStatus

o CSR-Attributes

The following steps outline how the above BRSKI flow maps to TEAP.

o Step 1: Device discovers the Registrar

When BRSKI is executed in a TEAP tunnel, the device exchanges BRSKI TLVs with the TEAP server. The discovery process for devices is therefore the standard wired or wireless LAN EAP server discovery process. The discovery processes outlined in section 4 of [I-D.ietf-anima-bootstrapping-keyinfra] are not required for initial discovery of the Registrar.

o Step 2: Device establishes provisional TLS connection to Registrar

The device establishes an outer TEAP tunnel with the TEAP server and does not validate the server certificate. The device presents its LDevID as its identity certificate if it has a valid LDevID, otherwise it presents its IDevID. Server policy may also be used to

control which certificate the device is allowed present, as described in section Section 4.

If the presented credential is sufficient to grant access, the TEAP server can return an EAP-Success immediately. The device may still send a BRSKI-RequestVoucher TLV in response to the EAP-Success if it does not have, but requires, trust anchors for validating the TEAP server certificate.

If the TEAP server requires that the device execute a BRSKI flow, it sends a Request-Action TLV that includes a BRSKI-VoucherRequest TLV. For example, if the device presented its IDevID but the TEAP server requires an LDevID.

The TEAP server may also require the device to reenroll, for example, if the device presented a valid LDevID that is very closed to expiration. The server may instruct a device to reenroll by sending a Request-Action TLV that includes a zero byte length PKCS#10 TLV.

- Step 3: Device sends voucher request message and receives signed voucher response

The device sends a BRSKI-RequestVoucher TLV to the TEAP server. The TEAP server forwards the RequestVoucher message to the MASA server, and the MASA server replies with a signed voucher. The TEAP server sends a BRSKI-Voucher TLV to the device.

If the MASA server does not issue a signed voucher, the TEAP server sends an EAP-Error TLV with a suitable error code to the device.

- Step 4: Device validates voucher and validates provisional TLS connection to Registrar

The device validates the signed voucher using its manufacturer installed trust anchor, and uses the CA information in the voucher to validate the outer TEAP TLS connection to the TEAP server.

If the device fails to validate the voucher, or fails to validate the outer TEAP TLS connection, then it sends a BRSKI-VoucherStatus TLV indicating failure to the TEAP server.

- Step 5: Device sends Voucher Status Telemetry message

On successfully validating the voucher and outer TEAP TLS connection, the device sends a BRSKI-VoucherStatus TLV to the TEAP server.

Once step 5 is complete, the device has completed the BRSKI flow and has established trust with the network.

- Step 6: Device downloads additional local domain CA information

On completion of the BRSKI flow, the device SHOULD send a Trusted-Server-Root TLV to the TEAP server in order to discover additional local domain CAs.

- Step 7: Device downloads CSR attributes

On completion of step 6, the device MUST send a CSR-Attributes TLV to the TEAP server in order to discover the correct fields to include when it enrolls to get an LDevID.

- Step 8: Device does an EST enroll to obtain an LDevID

When executing the BRSKI flow inside a TEAP tunnel, the device does not directly leverage EST when doing its initial enroll. Instead, the device uses the existing TEAP PKCS#10 and PCKS#7 TEAP mechanisms.

Once the BRSKI flow is complete, the device can now send a PKCS#10 TLV to enroll and request an LDevID. If the TEAP server instructed the device to start the BRSKI flow via a Request-Action TLV that includes a BRSKI-RequestVoucher TLV, then the device MUST send a PKCS#10 in order to start the enroll process. The TEAP server will handle the PKCS#10 and ultimately return a PKCS#7 including an LDevID to the device.

If the TEAP server granted the device access on completion of the outer TEAP TLS tunnel in step 2 without sending a Request-Action TLV, the device does not have to send a PKCS#10 to enroll.

- Step 9: Device sends Enrollment Status Telemetry message

On completion of certificate enrollment, the device sends an EnrollmentStatus TLV to the TEAP server.

- Step 10: Device periodically reenrolls to refresh its LDevID

When a device's LDevID is close to expiration, there are two options for re-enrollment in order to obtain a fresh LDevID. As outlined in Step 2 above, the TEAP server may instruct the device to reenroll by sending a Request-Action TLV including a PKCS#10 TLV. If the TEAP server explicilty instructs the device to reenroll via these TLV exchange, then the device MUST send a PKCS#10 to reenroll and request a fresh LDevID.

However, the device should be capabale of autonomic reenrollment if it determines that its LDevID is close to expiration wihtout waiting

for explicit instruction from the TEAP server. There are two options to do this.

Option 1: The device reenrolls for a new LDevID directly with the EST CA outside the context of the 802.1X TEAP authentication flow. The device uses the Registrar discovery mechanisms oulined in [I-D.ietf-anima-bootstrapping-keyinfra] to discover the Registrar and the device sends the EST reenroll messages to the discovered Registrar endpoint. No new TEAP TLVs are defined to facilitate discover of the Registrar or EST endpoints inside the context of the TEAP tunnel.

Option 2: When the device is perfroming a periodic 802.1X authentication using its current LDevID, it reenrolls for a new LDevID by sending a PKCS#10 TLV inside the TEAP TLS tunnel.

4. PKI Certificate Authority Considerations

Careful consideration must be given to PKI certificate authority handling when:

- Establishing the TEAP tunnel

- Establishing trust using BRSKI

These are described in more detail here.

4.1. TEAP Tunnel Establishment

The client sends its ClientHello to initiate TLS tunnel establishment. It is possible for the TEAP server to restrict the certificates that the client can use for tunnel establishment by including a list of CA distinguished names in the certificate_authorities field in the CertificateRequest message. Network operators may want to do this in order to restrict netwok access to clients that have a certificate signed by one of a small set of trusted manufacturer/supplier CAs. If the client has both an IDevID and an LDevID, the client should present the LDevID in preference to its IDevID if allowed by server policy.

In practice, network operators will likely want to onboard devices from a large number of device manufacturers, with each manufacturer using a different root CA when issuing IDevIDs. If the number of different manufacturer root CAs is large, this could result in very large TLS handshake messages. Operators may prefer to include no CAs in the certificate_authorities field thus allowing devices to present IDevIDs signed by any CA when establishing the TEAP tunnel, and instead enforce policy at LDevID enrollment time.

It is recommended that the client validate the certificate presented by the server in the server's Certificate message, but this may not be possible for bootstrapping clients that do not have appropriate trust anchors installed yet. If the client is bootstrapping and has not yet completed a BRSKI flow, it will not have trust anchors installed yet, and thus will not be able to validate the server's certificate. The client must however note the certificate presented by the server for (i) inclusion in the BRSKI-RequestVoucher TLV and for (ii) validation once the client has discovered the local domain trust anchors.

If the client does not present a suitable certificate to the server, the server MUST terminate the connection and fail the EAP request.

On establishment of the outer TLS tunnel, the TEAP server will make a policy decision on next steps. Possible policy decisions include:

- Option 1: Server grants client full network access and returns EAP-Success. This will typically happen when the client presents a valid LDevID. Network policy may grant client network access based on IDevID without requiring the device to enroll to obtain an LDevID.

- Option 2: Server requires that client perform a full BRSKI flow, and then enroll to get an LDevID. This will typically happen when the client presents a valid IDevID and network policy requires all clients to have LDevIDs. The server sends a Request-Action TLV that includes a BRSKI-RequestVoucher TLV to the client to instruct it to start the BRSKI flow.

- Option 3: Server requires that the client reenroll to obtain a new LDevID. This could happen when the client presents a valid LDevID that is very close to expiration time, or the server's policy requires an LDevID update. The server sends an Action-Request TLV including a PKCS#10 TLV to the client to instruct it to reenroll.

4.2. BRSKI Trust Establishment

If the server requires that client perform a full BRSKI flow, it sends a Request-Action TLV that includes a zero byte length BRSKI-RequestVoucher TLV to the client. The client sends a new BRSKI-RequestVoucher TLV to the server, which contains all data specified in [I-D.ietf-anima-bootstrapping-keyinfra] section 5.2. The client includes the server certificate it received in the server's Certificate message during outer TLS tunnel establishment in the proximity-registrar-cert field. The client signs the request using its IDevID.

The server includes all additional information as required by [I-D.ietf-anima-bootstrapping-keyinfra] section 5.4 and signs the request prior to forwarding to the MASA.

The MASA responds as per [I-D.ietf-anima-bootstrapping-keyinfra] section 5.5. The response may indicate failure and the server should react accordingly to failures by sending a failure response to the client, and failing the TEAP method.

If the MASA replies with a signed voucher and a successful result, the server then forwards this response to the client in a BRSKI-Voucher TLV.

When the client receives the signed voucher, it validates the signature using its built in trust anchor list, and extracts the pinned-domain-cert field. The client must use the CA included in the pinned-domain-cert to validate the certificate that was presented by the server when establishing the outer TLS tunnel. If this certificate validation fails, the client must fail the TEAP request and not connect to the network.

If the client successfully validates the server certificate, then it must send voucher status telemetry, as required by [I-D.ietf-anima-bootstrapping-keyinfra] section 5.6 to the server, in a new BRSKI-VoucherTelemetry TLV.

5. Channel and Crypto Binding

As the TEAP BRSKI flow does not define or require an inner EAP method, there is no explicit need for exchange of Channel-Binding TLVs between the device and the TEAP server.

The TEAP BRSKI TLVs are expected to occur at the beginning of the TEAP Phase 2 and MUST occur before the final Crypto-Binding TLV. This draft does not exclude the possibility of having other EAP methods occur following the TEAP BRSKI TLVs and as such, the Crypto-Binding TLV process rules as defined in [RFC7170] apply.

6. Protocol Flows

This section outlines protocol flows that map to the 3 server policy options described in section Section 4.1. The protocol flows illustrate a TLS1.2 exchange.

[[TODO]] MISSING EAP Start/identity/Crypto binding, etc. Flows illustrative, but not 100% accurate.

6.1. TEAP Server Grants Access

```
   +--------+             +-------------+     +------+
   | Client |             | TEAP-Server |     | MASA |
   +--------+             +-------------+     +------+
       |                          |                 |
       |  ClientHello             |                 |
       |------------------------->|                 |
       |  ServerHello,            |                 |
       |  Certificate(1),         |                 |
       |  ServerKeyExchange,      |                 |
       |  CertificateRequest(2),  |                 |
       |  ServerHelloDone,        |                 |
       |<-------------------------|                 |
       |  Certificate(3),         |                 |
       |  ClientKeyExchange,      |                 |
       |  CertificateVerify,      |                 |
       |  ChangeCipherSpec,       |                 |
       |  Finished                |                 |
       |------------------------->|                 |
       |  ChangeCipherSpec,       |                 |
       |  Finished                |                 |
       |<-------------------------|                 |
       |  Crypto-Binding TLV      |                 |
       |<-------------------------|                 |
       |  Crypto-Binding TLV      |                 |
       |------------------------->|                 |
       |  Result TLV              |                 |
       |<-------------------------|                 |
       |  Result TLV              |                 |
       |------------------------->|                 |
       |      EAP-Success         |                 |
       |<-------------------------|                 |
```

Figure 1: TEAP Server Grants Access

Notes:

(1) If the client has completed the BRSKI flow yet, it must validate the Certificate received from the server. If the client has not yet completed the BRSKI flow, then it provisionally accepts the server Certificate and must validate it later once BRSKI is complete.

(2) The server may include certificate_authorities field in the CertificateRequest message in order to restrict the identity certificates that the device is allowed present.

(3) The device will present its LDevID, if it has one, in preference to its IDevID, if allowed by server policy.

6.2. TEAP Server Instructs Client to Perform BRSKI Flow

```
   +--------+             +-------------+     +------+
   | Client |             | TEAP-Server |     | MASA |
   +--------+             +-------------+     +------+
       |                         |                |
       |      ClientHello        |                |
       |------------------------>|                |
       |   ServerHello, etc.(1)  |                |
       |<------------------------|                |
       |     etc., Finished      |                |
       |------------------------>|                |
       |     etc., Finished      |                |
       |<------------------------|                |
       | BRSKI-RequestAction(2)  |                |
       |<------------------------|                |
       | BRSKI-RequestVoucher(3) |                |
       |------------------------>| RequestVoucher |
       |                         |--------------->|
       |                         |    Voucher     |
       |     BRSKI-Voucher(4)    |<---------------|
       |<------------------------|                |
       |  BRSKI-VoucherStatus    |                |
       |------------------------>|                |
       |  Trusted-Server-Root    |                |
       |------------------------>|                |
       |  Trusted-Server-Root    |                |
       |<------------------------|                |
       |     CSR-Attributes      |                |
       |------------------------>|                |
       |     CSR-Attributes      |                |
       |<------------------------|                |
       |         PKCS#10         |                |
       |------------------------>|                |
       |         PKCS#7          |                |
       |<------------------------|                |
       |    EnrollmentStatus     |                |
       |------------------------>|                |
       |  Crypto-Binding TLV     |                |
       |<------------------------|                |
       |  Crypto-Binding TLV     |                |
       |------------------------>|                |
       |  Result TLV             |                |
       |<------------------------|                |
       |  Result TLV             |                |
```

```
|------------------------>|                |
|      EAP-Success        |                |
|<------------------------|                |
```

Figure 2: TEAP Server Instructs Client to Perform BRSKI Flow

Notes:

(1) If the client has not yet completed the BRSKI flow, then it provisionally accepts the server certificate and must validate it later once BRSKI is complete.

(2) The server instructs the client to start the BRSKI flow by sending a RequestAction TLV that includes a BRSKI-RequestVoucher TLV.

(3) The client includes the certificate it received from the server in the RequestVoucher message.

(4) Once the client receives and validates the voucher signed by the MASA, it must verify the certificate it previously received from the server.

6.3. TEAP Server Instructs Client to Reenroll

```
+--------+           +-------------+     +------+
| Client |           | TEAP-Server |     | MASA |
+--------+           +-------------+     +------+
    |                         |                |
    |      ClientHello        |                |
    |------------------------>|                |
    |   ServerHello, etc.     |                |
    |<------------------------|                |
    |     etc., Finished      |                |
    |------------------------>|                |
    |     etc., Finished      |                |
    |<------------------------|                |
    | RequestAction: PKCS#10  |                |
    |<------------------------|                |
    |     CSR-Attributes      |                |
    |------------------------>|                |
    |     CSR-Attributes      |                |
    |<------------------------|                |
    |         PKCS#10         |                |
    |------------------------>|                |
    |         PKCS#7          |                |
    |<------------------------|                |
    |    EnrollmentStatus     |                |
    |------------------------>|                |
    |  Crypto-Binding TLV     |                |
    |<------------------------|                |
    |  Crypto-Binding TLV     |                |
    |------------------------>|                |
    |  Result TLV             |                |
    |<------------------------|                |
    |  Result TLV             |                |
    |------------------------>|                |
    |      EAP-Success        |                |
    |<------------------------|                |
```

Figure 3: TEAP Server Instructs Client to Reenroll

[[TODO: Hmm, as BRSKI mandates client requests CSR-Attributes, should the server send a RequestAction and include two TLVs: one CSR-Attributes and one PKCS#10 ?]]

6.4. Out of Band Reenroll

This section shows how the device does a reenroll to refresh its LDEvID directly against the Registrar outside the context of the TEAP tunnel.

7. TEAP TLV Formats

7.1. BRSKI TLVs

BRSKI defines 6 new TEAP TLVs. The following table indicates whether the TLVs can be included in Request messages from TEAP server to device, or Response messages from device to TEAP server.

TLV	Message
BRSKI-VoucherRequest	Response
BRSKI-Voucher	Request
BRSKI-VoucherStatus	Response
EnrollmentStatus	Response
CSR-Attributes	Response

These new TLVs are detailed in this section.

7.1.1. BRSKI-RequestVoucher TLV

To be completed.

7.1.2. BRSKI-Voucher TLV

To be completed.

7.1.3. BRSKI-VoucherStatus TLV

To be completed.

7.1.4. EnrollmentStatus TLV

To be completed.

7.1.5. CSR-Attributes TLV

To be completed.

7.2. Existing TEAP TLV Specifications

This section documents allowed usage of existing TEAP TLVs. The definition of the TLV is not changed, however clarifications on allowed values for the TLV fields is documented.

7.2.1. PKCS#10 TLV

[RFC7170] defines the PKCS#10 TLV as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|M|R|         TLV Type          |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           PKCS#10 Data...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-++-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

[RFC7170] does not explicitly allow a Length value of zero.

A Length value of zero is allowed for this TLV when the TEAP server sends a Request-Action TLV with a child PKCS#10 TLV to the client. In this scenario, there is no PKCS#10 Data included in the TLV. Clients MUST NOT send a zero length PKCS#10 TLV to the server.

8. Fragmentation

TLS is expected to provide fragmentation support. Thus EAP-TEAP-BRSKI does not specifically provide any, as it is only expected to be used as an inner method to TEAP.

9. IANA Considerations

TBD.

10. Security Considerations

There will be many.

11. Acknowledgments

The authors would like to thank Brian Weis for his assistance.

12. Informative References

[I-D.ietf-anima-bootstrapping-keyinfra]
Pritikin, M., Richardson, M., Behringer, M., Bjarnason, S., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructures (BRSKI)", draft-ietf-anima-bootstrapping-keyinfra-16 (work in progress), June 2018.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, Ed., "Extensible Authentication Protocol (EAP)", RFC 3748, DOI 10.17487/RFC3748, June 2004, <https://www.rfc-editor.org/info/rfc3748>.

[RFC7030] Pritikin, M., Ed., Yee, P., Ed., and D. Harkins, Ed., "Enrollment over Secure Transport", RFC 7030, DOI 10.17487/RFC7030, October 2013, <https://www.rfc-editor.org/info/rfc7030>.

[RFC7170] Zhou, H., Cam-Winget, N., Salowey, J., and S. Hanna, "Tunnel Extensible Authentication Protocol (TEAP) Version 1", RFC 7170, DOI 10.17487/RFC7170, May 2014, <https://www.rfc-editor.org/info/rfc7170>.

[RFC8366] Watsen, K., Richardson, M., Pritikin, M., and T. Eckert, "A Voucher Artifact for Bootstrapping Protocols", RFC 8366, DOI 10.17487/RFC8366, May 2018, <https://www.rfc-editor.org/info/rfc8366>.

Appendix A. Changes from Earlier Versions

Draft -00:

- o Initial revision

Authors' Addresses

Eliot Lear
Cisco Systems
Richtistrasse 7
Wallisellen CH-8304
Switzerland

Phone: +41 44 878 9200
Email: lear@cisco.com

Owen Friel
Cisco Systems
170 W. Tasman Dr.
San Jose, CA 95134
United States

Email: ofriel@cisco.com

Nancy Cam-Winget
Cisco Systems
170 W. Tasman Dr.
San Jose, CA 95134
United States

Email: ncamwing@cisco.com