

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 14, 2021

O. Friel
R. Barnes
Cisco
R. Shekh-Yusef
Auth0
M. Richardson
Sandelman Software Works
July 13, 2020

ACME Integrations
draft-ietf-acme-integrations-01

Abstract

This document outlines multiple advanced use cases and integrations that ACME facilitates without any modifications or enhancements required to the base ACME specification. The use cases include ACME integration with EST, BRSKI and TEAP.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 14, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Terminology	3
3.	ACME Integration with EST	3
4.	ACME Integration with BRSKI	6
5.	ACME Integration with BRSKI Default Cloud Registrar	8
6.	ACME Integration with TEAP	10
7.	IANA Considerations	14
8.	Security Considerations	14
9.	Informative References	14
Appendix A.	Comments	15
	Authors' Addresses	15

[1.](#) Introduction

ACME [[RFC8555](#)] defines a protocol that a certificate authority (CA) and an applicant can use to automate the process of domain name ownership validation and X.509 (PKIX) certificate issuance. The protocol is rich and flexible and enables multiple use cases that are not immediately obvious from reading the specification. This document explicitly outlines multiple advanced ACME use cases including:

- o ACME integration with EST [[RFC7030](#)]
- o ACME integration with BRSKI
[[I-D.ietf-anima-bootstrapping-keyinfra](#)]
- o ACME integration with BRSKI Default Cloud Registrar
[[I-D.friel-anima-brski-cloud](#)]
- o ACME integration with TEAP [[RFC7170](#)] and TEAP Update and Extensions for Bootstrapping [[I-D.lear-eap-teap-brski](#)]

The integrations with EST, BRSKI (which is based upon EST), and TEAP enable automated certificate enrolment for devices.

ACME for subdomains [[I-D.friel-acme-subdomains](#)] outlines how ACME can be used by a client to obtain a certificate for a subdomain identifier from a certificate authority where the client has fulfilled a challenge against a parent domain, but does not need to fulfil a challenge against the explicit subdomain. This is a useful optimization when ACME is used to issue certificates for large

numbers of devices as it reduces the domain ownership proof traffic (DNS or HTTP) and ACME traffic overhead, but is not a necessary requirement.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP 14](#) [[RFC2119](#)] [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

The following terms are used in this document:

- o BRSKI: Bootstrapping Remote Secure Key Infrastructures [[I-D.ietf-anima-bootstrapping-keyinfra](#)]
- o CA: Certificate Authority
- o CMC: Certificate Management over CMS
- o CSR: Certificate Signing Request
- o EST: Enrollment over Secure Transport [[RFC7030](#)]
- o FQDN: Fully Qualified Domain Name
- o RA: PKI Registration Authority
- o TEAP: Tunnelled Extensible Authentication Protocol [[RFC7170](#)]

3. ACME Integration with EST

EST [[RFC7030](#)] defines a mechanism for clients to enroll with a PKI Registration Authority by sending CMC messages over HTTP. EST [section 1](#) states:

"Architecturally, the EST service is located between a Certification Authority (CA) and a client. It performs several functions traditionally allocated to the Registration Authority (RA) role in a PKI."

EST [section 1.1](#) states that:

"For certificate issuing services, the EST CA is reached through the EST server; the CA could be logically "behind" the EST server or embedded within it."

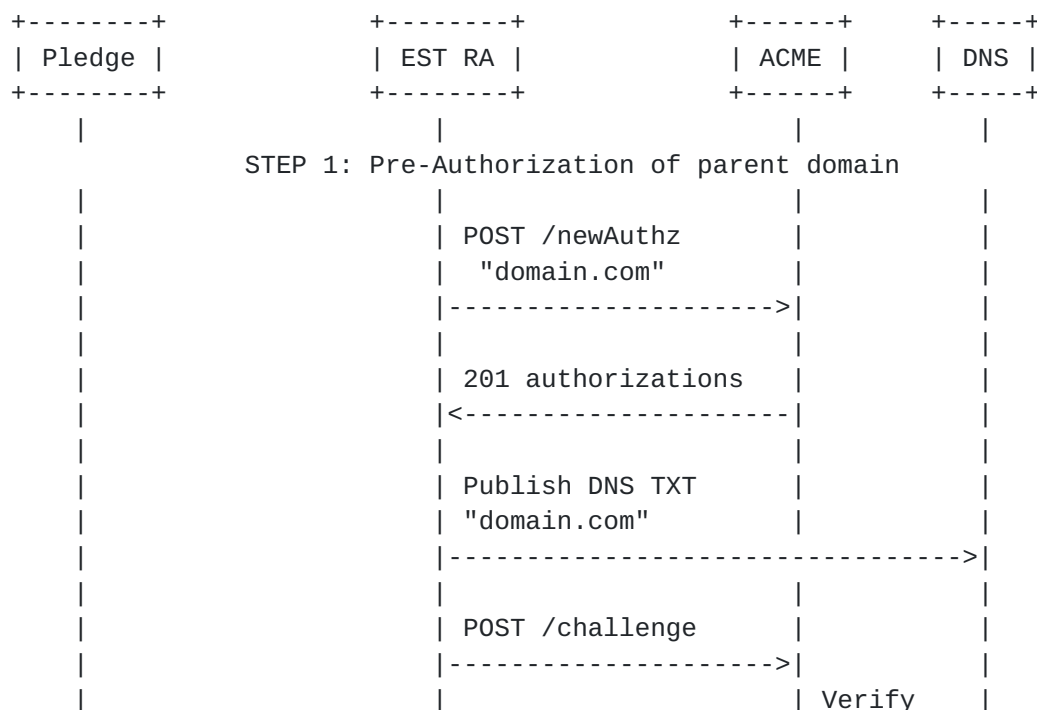
When the CA is logically "behind" the EST RA, EST does not specify how the RA communicates with the CA. EST [section 1](#) states:

"The nature of communication between an EST server and a CA is not described in this document."

This section outlines how ACME could be used for communication between the EST RA and the CA. The example call flow leverages [\[I-D.friel-acme-subdomains\]](#) and shows the RA proving ownership of a parent domain, with individual client certificates being subdomains under that parent domain. This is an optimization that reduces DNS and ACME traffic overhead. The RA could of course prove ownership of every explicit client certificate identifier.

The call flow illustrates the client calling the EST /csrattrs API before calling the EST /simpleenroll API. This enables the EST server to indicate to the client what attributes it expects the client to include in the CSR request sent in the /simpleenroll API. For example, EST servers could use this mechanism to tell the client what fields to include in the CSR Subject and Subject Alternative Name fields.

The call flow illustrates the EST RA returning a 202 Retry-After response to the client's simpleenroll request. This is an optional step and may be necessary if the interactions between the RA and the ACME server take some time to complete. The exact details of when the RA returns a 202 Retry-After are implementation specific.




```

|                                     |----->
|                                     | 200 status=valid
|                                     |<-----
|                                     |
|                                     | Delete DNS TXT
|                                     | "domain.com"
|                                     |----->
|
| STEP 2: Pledge enrolls against RA
|
| GET /csrattrs
|----->
|
| 200 OK
| SEQUENCE {AttrOrOID}
| SAN OID:
| "pledgeid.domain.com"
|<-----
|
| POST /simpleenroll
| PCSK#10 CSR
| "pledgeid.domain.com"
|----->
|
| 202 Retry-After
|<-----
|
| STEP 3: RA places ACME order
|
| POST /newOrder
| "pledgeid.domain.com"
|----->
|
| 201 status=ready
|<-----
|
| POST /finalize
| PKCS#10 CSR
| "pledgeid.domain.com"
|----->
|
| 200 OK status=valid
|<-----
|
| POST /certificate
|----->
|
| 200 OK

```


		PEM		
		"pledgeid.domain.com"		
		<-----		
STEP 4: Pledge retries enroll				
		POST /simpleenroll		
		PKCS#10 CSR		
		"pledgeid.domain.com"		
		----->		
		200 OK		
		PKCS#7		
		"pledgeid.domain.com"		
		<-----		

4. ACME Integration with BRSKI

BRSKI [[I-D.ietf-anima-bootstrapping-keyinfra](#)] is based upon EST [[RFC7030](#)] and defines how to autonomically bootstrap PKI trust anchors into devices via means of signed vouchers. EST certificate enrollment may then optionally take place after trust has been established. BRSKI voucher exchange and trust establishment are based on EST extensions and the certificate enrollment part of BRSKI is fully based on EST. Similar to EST, BRSKI does not define how the EST RA communicates with the CA. Therefore, the mechanisms outlined in the previous section for using ACME as the communications protocol between the EST RA and the CA are equally applicable to BRSKI.

The following call flow shows how ACME may be integrated into a full BRSKI voucher plus EST enrollment workflow. For brevity, it assumes that the EST RA has previously proven ownership of a parent domain and that pledge certificate identifiers are a subdomain of that parent domain. The domain ownership exchanges between the RA, ACME and DNS are not shown. Similarly, not all BRSKI interactions are shown and only the key protocol flows involving voucher exchange and EST enrollment are shown.

Similar to the EST section above, the client calls EST /csrattrs API before calling the EST /simpleenroll API. This enables the server to indicate what fields the pledge should include in the CSR that the client sends in the /simpleenroll API.

The call flow illustrates the RA returning a 202 Retry-After response to the initial EST /simpleenroll API. This may be appropriate if processing of the /simpleenroll request and ACME interactions takes some time to complete.

+-----+	+-----+	+-----+	+-----+
Pledge	EST RA	ACME	MASA
+-----+	+-----+	+-----+	+-----+
	NOTE: Pre-Authorization of "domain.com" is complete		
	STEP 1: Pledge requests Voucher		
	POST /requestvoucher		
	----->		
		POST /requestvoucher	
		----->	
		200 OK Voucher	
		<-----	
	200 OK Voucher		
	<-----		
	STEP 2: Pledge enrolls against RA		
	GET /csrattrs		
	----->		
	200 OK		
	SEQUENCE {AttrOrOID}		
	SAN OID:		
	"pledgeid.domain.com"		
	<-----		
	POST /simpleenroll		
	PCSK#10 CSR		
	"pledgeid.domain.com"		
	----->		
	202 Retry-After		
	<-----		
	STEP 3: RA places ACME order		
		POST /newOrder	
		"pledgeid.domain.com"	
		----->	
		201 status=ready	
		<-----	
		POST /finalize	
		PKCS#10 CSR	


```

|                                     | "pledgeid.domain.com"|
|                                     | ----->|
|                                     |
|                                     | 200 OK status=valid |
|                                     | <-----|
|                                     |
|                                     | POST /certificate   |
|                                     | ----->|
|                                     |
|                                     | 200 OK
|                                     | PEM
|                                     | "pledgeid.domain.com"|
|                                     | <-----|
|
| STEP 4: Pledge retries enroll
|
| POST /simpleenroll
| PCSK#10 CSR
| "pledgeid.domain.com"|
| ----->|
|
| 200 OK
| PKCS#7
| "pledgeid.domain.com"|
| <-----|

```

5. ACME Integration with BRSKI Default Cloud Registrar

BRSKI Cloud Registrar [[I-D.friel-anima-brski-cloud](#)] specifies the behaviour of a BRSKI Cloud Registrar, and how a pledge can interact with a BRSKI Cloud Registrar when bootstrapping. Similar to the local domain registrar BRSKI flow, ACME can be easily integrated with a cloud registrar bootstrap flow.

BRSKI cloud registrar is flexible and allows for multiple different local domain discovery and redirect scenarios. In the example illustrated here, the extension to [RFC8366](#) Vouchers which is defined in [\[TODO ID-TBD\]](#) and allows the specification of a bootstrap DNS domain is leveraged. This extension allows the cloud registrar to specify the local domain RA that the pledge should connect to for the purposes of EST enrollment.

```

+-----+           +-----+           +-----+           +-----+
| Pledge |           | EST RA |           | ACME |           | Cloud RA |
+-----+           +-----+           +-----+           | / MASA |
|                                                     +-----+
|                                                     |

```

NOTE: Pre-Authorization of "domain.com" is complete


```
|
|   STEP 1: Pledge requests Voucher from Cloud Registrar
|
|   POST /requestvoucher
|----->
|
|   200 OK Voucher (EST RA domain)
|<-----
|
|   STEP 2: Pledge enrolls against local domain RA
|
|   GET /csrattrs
|----->
|
|   200 OK
|   SEQUENCE {AttrOrOID}
|   SAN OID:
|   "pledgeid.domain.com"
|<-----
|
|   POST /simpleenroll
|   PCSK#10 CSR
|   "pledgeid.domain.com"
|----->
|
|   202 Retry-After
|<-----
|
|   STEP 3: RA places ACME order
|
|   POST /newOrder
|   "pledgeid.domain.com"
|----->
|
|   201 status=ready
|<-----
|
|   POST /finalize
|   PKCS#10 CSR
|   "pledgeid.domain.com"
|----->
|
|   200 OK status=valid
|<-----
|
|   POST /certificate
|----->
|
```



```

|                                     | 200 OK                                     |
|                                     | PEM                                       |
|                                     | "pledgeid.domain.com"                   |
|                                     | <----->                               |
|
| STEP 4: Pledge retries enroll
|
| POST /simpleenroll                  |
| PCSK#10 CSR                        |
| "pledgeid.domain.com"             |
| ----->                           |
|
| 200 OK                             |
| PKCS#7                             |
| "pledgeid.domain.com"             |
| <----->                           |

```

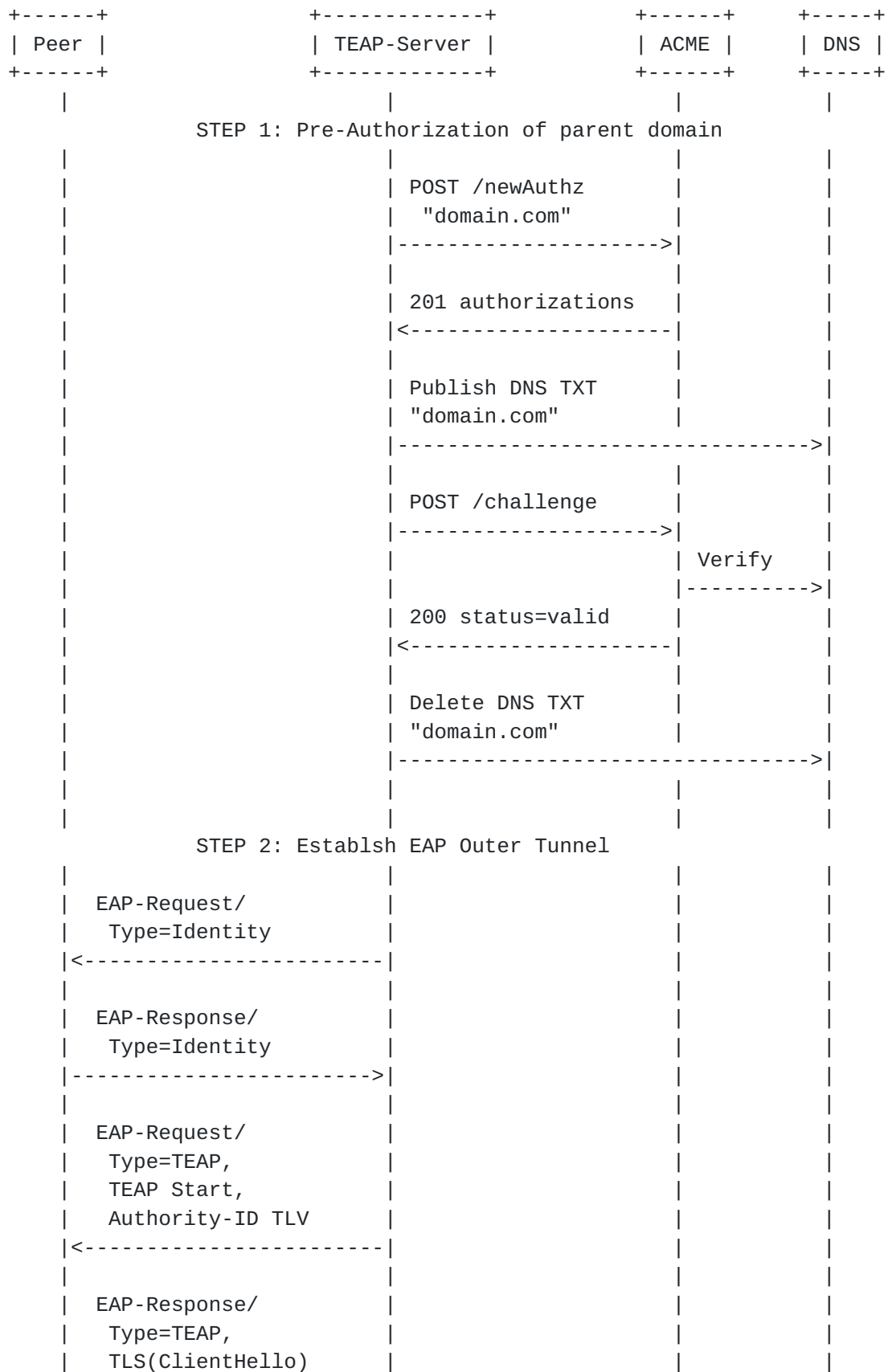
6. ACME Integration with TEAP

TEAP [[RFC7170](#)] defines a tunnel-based EAP method that enables secure communication between a peer and a server by using TLS to establish a mutually authenticated tunnel. TEAP enables certificate provisioning within the tunnel. TEAP Update and Extensions for Bootstrapping [[I-D.lear-eap-teap-brski](#)] defines extensions to TEAP that includes additional TLVs for certificate enrollment and BRSKI handling inside the TEAP tunnel. Neither TEAP [[RFC7170](#)] or TEAP Update and Extensions for Bootstrapping [[I-D.lear-eap-teap-brski](#)] define how the TEAP server communicates with the CA.

This section outlines how ACME could be used for communication between the TEAP server and the CA. The example call flow leverages [[I-D.friel-acme-subdomains](#)] and shows the TEAP server proving ownership of a parent domain, with individual client certificates being subdomains under that parent domain.

The example illustrates the TEAP server sending a Request-Action TLV including a CSR-Attributes TLV instructing the peer to send a CSR-Attributes TLV to the server. This enables the server to indicate what fields the peer should include in the CSR that the peer sends in the PKCS#10 TLV. For example, the TEAP server could instruct the peer what Subject or SAN entries to include in its CSR.

Although not explicitly illustrated in this call flow, the Peer and TEAP Server could exchange BRSKI TLVs, and a BRSKI integration and voucher exchange with a MASA server could take place over TEAP. Whether a BRSKI TLV exchange takes place or not does not impact the ACME specific message exchanges.



<pre> -----> EAP-Request/ Type=TEAP, TLS(ServerHello, Certificate, ServerKeyExchange, CertificateRequest, ServerHelloDone) <----- EAP-Response/ Type=TEAP, TLS(Certificate, ClientKeyExchange, CertificateVerify, ChangeCipherSpec, Finished) -----> EAP-Request/ Type=TEAP, TLS(ChangeCipherSpec, Finished), {Crypto-Binding TLV, Result TLV=Success} <----- EAP-Response/ Type=TEAP, {Crypto-Binding TLV, Result TLV=Success} -----> EAP-Request/ Type=TEAP, {Request-Action TLV: Status=Failure, Action=Process-TLV, TLV=CSR-Attributes, TLV=PKCS#10} <----- STEP 3: Enroll for certificate EAP-Response/ Type=TEAP, {CSR-Attributes TLV} </pre>			
---	--	--	--


```

|----->|
|
| EAP-Request/
|   Type=TEAP,
|   {CSR-Attributes TLV}
|-----<|
|
| EAP-Response/
|   Type=TEAP,
|   {PKCS#10 TLV:
|     "pledgeid.domain.com"}
|----->|
|
| POST /newOrder
| "pledgeid.domain.com"
|----->|
|
| 201 status=ready
|-----<|
|
| POST /finalize
| PKCS#10 CSR
| "pledgeid.domain.com"
|----->|
|
| 200 OK status=valid
|-----<|
|
| POST /certificate
|----->|
|
| 200 OK
| PEM
| "pledgeid.domain.com"
|-----<|
|
| EAP-Request/
|   Type=TEAP,
|   {PKCS#7 TLV,
|     Result TLV=Success}
|-----<|
|
| EAP-Response/
|   Type=TEAP,
|   {Result TLV=Success}
|----->|
|
| EAP-Success
|-----<|

```


7. IANA Considerations

[todo]

8. Security Considerations

[todo]

9. Informative References

[I-D.friel-acme-subdomains]

Friel, O., Barnes, R., Hollebeek, T., and M. Richardson, "ACME for Subdomains", [draft-friel-acme-subdomains-02](#) (work in progress), March 2020.

[I-D.friel-anima-brski-cloud]

Friel, O., Shekh-Yusef, R., and M. Richardson, "BRSKI Cloud Registrar", [draft-friel-anima-brski-cloud-02](#) (work in progress), May 2020.

[I-D.ietf-anima-bootstrapping-keyinfra]

Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructures (BRSKI)", [draft-ietf-anima-bootstrapping-keyinfra-41](#) (work in progress), April 2020.

[I-D.lear-eap-teap-brski]

Lear, E., Friel, O., Cam-Winget, N., and D. Harkins, "TEAP Update and Extensions for Bootstrapping", [draft-lear-eap-teap-brski-05](#) (work in progress), November 2019.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[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>>.

- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [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>>.
- [RFC8555] Barnes, R., Hoffman-Andrews, J., McCarney, D., and J. Kasten, "Automatic Certificate Management Environment (ACME)", [RFC 8555](#), DOI 10.17487/RFC8555, March 2019, <<https://www.rfc-editor.org/info/rfc8555>>.

[Appendix A](#). Comments

Authors' Addresses

Owen Friel
Cisco

Email: ofriel@cisco.com

Richard Barnes
Cisco

Email: rlb@ipv.sx

Rifaat Shekh-Yusef
Auth0

Email: rifaat.s.ietf@gmail.com

Michael Richardson
Sandelman Software Works

Email: mcr+ietf@sandelman.ca

