anima Working Group
Internet-Draft
Intended status: Standards Track
Expires: 7 January 2026

A. Dekok
FreeRADIUS
M. Richardson
Sandelman Software Works

6 July 2025

EAP defaults for devices that need to onboard
draft-richardson-emu-eap-onboarding-04

Abstract

This document describes a method by which an unconfigured device can use EAP-TLS to join a network on which further device onboarding, network attestation or other remediation can be done. While RFC 5216 supports EAP-TLS without a client certificate, that document defines no method by which unauthenticated EAP-TLS can be used. This draft addresses that issue.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-richardson-emu-eap-onboarding/.

Discussion of this document takes place on the anima Working Group mailing list (mailto:anima@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/anima/. Subscribe at https://www.ietf.org/mailman/listinfo/anima/.

Source for this draft and an issue tracker can be found at https://github.com/mcr/eap-onboarding.git.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

There are a multitude of situations where a network device needs to join a new (wireless) network but where the device does not yet have the right credentials for that network. As the device does not have credentials, it cannot access networks which typically require authentication. However, since the device does not have network access, it cannot download a new configuration which contains updated credentials.

The process by which a device acquires these credentials has become known as onboarding [I-D.irtf-t2trg-secure-bootstrapping]. There are many onboarding protocols, including [RFC8995], [RFC9140], [dpp], CSA MATTER, and OPC UA Part 21. Some of these protocols use WiFi Public frames, or provide for provisioning as part of EAP, such as [RFC7170]. Other systems require pre-existing IP connectivity in order to configure credentials for a device, which causes a circular dependancy.

This document defines a method where devices can use unauthenticated EAP-TLS in order to obtain network access, albeit in a captive portal [RFC8952]. Once the device is in a captive portal, it has access to the full suite of Internet Protocol (IP) technologies, and can proceed with onboarding.

This method is clearer, safer, and easier to implement and deploy than alternatives as it does not attempt to replicate the IP layers or TCP transports over an EAP layer.

This method also allows for multiple onboarding technologies to co-exist, and for the technologies to evolve without requiring invasive upgrades to the layer-2 infrastructure.

The method detailed in this document uses the unauthenticated client mode of EAP-TLS. While [RFC5216] defines EAP-TLS without a client certificate, that document defines no method by which unauthenticated EAP-TLS can be used.

This draft addresses that issue. [I-D.ietf-emu-eap-arpa] has defined the @eap.arpa domain, and this document builds upon it by showing how it can be used to provid network access for onboarding unauthenticated devices.

Note that this specification does not specify the exact method used for onboarding devices! There are many possibilities, with some methods yet to be defined. Not all of them are enumerated here. This document explains how to get the wireless equivalent of a plugged in wire, but without any promises of further connectivity.

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 term _supplicant_ is used to refer to the network device which is attempting to do EAP-TLS.

The term _pledge_ (from [RFC8995]) is used to refer to the network device which has successfully performed unauthenticated client mode EAP-TLS, and now has access to a network on which is may perform onboarding.

3. Protocol Details

The onboarding is divided into the following phases:

* Discovery - the supplicant determines that a network can do onboarding,

* Authentication - the supplicant connects to the network as an unauthenticated device,

* Authorization - the network provides limited connectivity to the device/pledge,

* Onboarding - the device/pledge uses standard IP protocols to perform onboarding,

* Full network access - the device has provisioned credentials, and can proceed with normal network access.

3.1. Discovery

The network should use 802.11u to signal that it can potentially perform onboarding, by using 802.11u and indicating that it supports the realm "eap.arpa".

When a supplicant which requires onboarding sees this realm, it knows that the network may be suitable for onboarding.

Note that not all such networks are suitable for onboarding using the technologies that a supplicant has. Some networks might have only a captive portal, intended for human use. This is the "coffee shop" case.

There may be multiple such networks available, and only one (or none) may be willing to onboard this particular device. Further, the device does not necessarily trust any such network.

There are situations where there may be many hundreds of networks which offer onboarding, and a supplicant device may need to try all of them until it finds a network to which it can successfully onboard. An example of such a situation is in a large (dozens to hundreds of floors) apartment building in a downtown core, where radio signals may leak from adjacent units, reflect off glass windows, come from other floors, and even cross the street from adjacent buildings. This document does not address this issue, but anticipates future work in 802.11u, perhaps involving some filtering mechanism using Bloom Filters.

Supplicants MUST limit their actions in the onboarding network to the action of onboarding. If this process cannot be completed, the device MUST disconnect from the onboarding network, and try again, usually by selecting a different network.

As soon as the device has been onboarded, the device MUST disconnect from the onboarding network, and use the provided configuration to authenticate and connect to a fully-capable network.

3.2. Authentication

The supplicant presents itself as an unauthenticated peer, which is allowed by EAP-TLS [RFC5216] Section 2.1.1. TLS 1.2 or TLS 1.3 [RFC9190] may be used, but TLS 1.3 or higher is RECOMMENDED.

The supplicant uses an identity of onboarding@eap.arpa, and provides no TLS client certificate. The use of the "eap.arpa" domain signals to the network that the device wishes to use unauthenticated EAP-TLS.

3.3. Authorization

Upon receipt of a supplicant without any authentication, the AAA server returns instructions to the authenticator to place the new client into the quarantined or captive portal network. The exact method is network-dependent, but it is usually done with a dedicated VLAN which has limited network access.

3.4. Characteristics of the Quarantine Network

The quarantine network SHOULD be segregated at layer-two (ethernet), and should not permit ethernet frames to any destination other than a small set of specified routers.

Specifically, the layer infrastructure should prevent one pledge from attempting to connect to another pledge on the same quarantine network.

For some onboarding protocols such as [RFC8995], only IPv6 Link-Local frames are needed. Such a network MUST provide a Join Proxy as specified in [RFC8995], Section 4.

For other onboarding protocols more capabilities may be needed, in particular there need for a DHCPv4 server may be critical for the device to believe it has connected correctly. This is particularly the case where a normal "smartphone" or laptop system will onboard via a captive portal.

Once on the quarantine network, device uses other protocols [RFC6876] to perform the onboarding action.

Note that the Pledge could also wind up in this qurantine network when using client credentials which are expired, or if the Pledge is unable to provide Evidence [RFC9334] that it is trustworthy. It is common for enterprises to force desktop/laptop Pledge systems into a quarantine network when it has been determined that the Pledge contains malware, or is might be considered vulnerable to current attacks. Such quarantine networks usually provide very limited access, but do include access to apply system patches.

4. Captive Portal

While this document imposes no requirements on the rest of the network, captive portals [RFC8952] have been used for almost two decades. The administration and operation of captive portals is typically within the authority of administrators who are responsible for network access. As such, this document defines additional behavior on, and requirements for, captive portals, so long as those changes materially benefit the network access administrator.

5. Privacy Considerations

Devices should take care to hide all identifying information from the onboarding network. Any identifying information MUST be sent encrypted via a method such as TLS.

6. Security Considerations

Devices using an onboarding network MUST assume that the network is untrusted. All network traffic SHOULD be encrypted in order to prevent attackers from both eavesdropping, and from modifying any provisioning information.

Similarly onboarding networks MUST assume that devices are untrusted, and could be malicious. Networks MUST make provisions to prevent Denial of Service (DoS) attacks, such as when many devices attempt to connect at the same time.

Networks MUST limit network access to onboarding protocols and captive portal access only.

Networks SHOULD also limit the bandwidth used by any device which is being onboarded.

Any returned configuration information from onboarding is likely to be small (megabytes at most), and it is reasonable to require a second or two for this process to take place.

Any device which cannot be onboarded within approximately 30 seconds SHOULD be disconnected from the quarantine network if there is no obvious activity. (A device with an active download of a software patch should be allowed to finish)

An idle device should not remain connected. Such a delay signals either a malicious device / network, or a misconfigured device / network. If onboarding cannot be finished within a short timer, the device should choose another network.

6.1. Use of eap.arpa

Supplicants MUST use the "eap.arpa" domain only for onboarding and related activities. [I-D.ietf-emu-eap-arpa] Supplicant MUST use unauthenticated EAP-TLS.

Networks which support onboarding via the "eap.arpa" domain MUST require that supplicants use unauthenticated EAP-TLS. The use of other EAP types MUST result in rejection, and a denial of all network access.

7. IANA Considerations

A new entry in the "EAP Provisioning Identifiers" [I-D.ietf-emu-eap-arpa] is required. It is unclear what this entry should be.

Perhaps it should be @nobody.eap.arpa. Perhaps it should be nobody@eap.arpa.

8. Acknowledgements

TBD.

9. Changelog

04: document updated to be in sync with draft-ietf-emu-eap-arpa. 03: refreshed so it does not expire 01 to 02: minor edits.

10. References

10.1. Normative References

[I-D.ietf-emu-eap-arpa]
DeKok, A., "The eap.arpa domain and EAP provisioning", Work in Progress, Internet-Draft, draft-ietf-emu-eap-arpa-08, 6 July 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-emu-eap-arpa-08>.

[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/rfc/rfc2119>.

[RFC5216] Simon, D., Aboba, B., and R. Hurst, "The EAP-TLS Authentication Protocol", RFC 5216, DOI 10.17487/RFC5216, March 2008, <https://www.rfc-editor.org/rfc/rfc5216>.

[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/rfc/rfc8174>.

[RFC9190] Preu Mattsson, J. and M. Sethi, "EAP-TLS 1.3: Using the Extensible Authentication Protocol with TLS 1.3", RFC 9190, DOI 10.17487/RFC9190, February 2022, <https://www.rfc-editor.org/rfc/rfc9190>.

10.2. Informative References

[dpp] "Device Provisioning Protocol Specification", n.d., <https://www.wi-fi.org/downloads-registered-guest/Device_Provisioning_Protocol_Draft_Technical_Specification_Package_v0_0_23_0.zip/31255>.

[I-D.irtf-t2trg-secure-bootstrapping]
Sethi, M., Sarikaya, B., and D. Garcia-Carrillo, "Terminology and processes for initial security setup of IoT devices", Work in Progress, Internet-Draft, draft-

irtf-t2trg-secure-bootstrapping-03, 26 November 2022, <https://datatracker.ietf.org/doc/html/draft-irtf-t2trg-secure-bootstrapping-03>.

[RFC6876] Sangster, P., Cam-Winget, N., and J. Salowey, "A Posture Transport Protocol over TLS (PT-TLS)", RFC 6876, DOI 10.17487/RFC6876, February 2013, <https://www.rfc-editor.org/rfc/rfc6876>.

[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/rfc/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/rfc/rfc7170>.

[RFC7542] DeKok, A., "The Network Access Identifier", RFC 7542, DOI 10.17487/RFC7542, May 2015, <https://www.rfc-editor.org/rfc/rfc7542>.

[RFC8952] Larose, K., Dolson, D., and H. Liu, "Captive Portal Architecture", RFC 8952, DOI 10.17487/RFC8952, November 2020, <https://www.rfc-editor.org/rfc/rfc8952>.

[RFC8995] Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", RFC 8995, DOI 10.17487/RFC8995, May 2021, <https://www.rfc-editor.org/rfc/rfc8995>.

[RFC9140] Aura, T., Sethi, M., and A. Peltonen, "Nimble Out-of-Band Authentication for EAP (EAP-NOOB)", RFC 9140, DOI 10.17487/RFC9140, December 2021, <https://www.rfc-editor.org/rfc/rfc9140>.

[RFC9334] Birkholz, H., Thaler, D., Richardson, M., Smith, N., and W. Pan, "Remote ATtestation procedureS (RATS) Architecture", RFC 9334, DOI 10.17487/RFC9334, January 2023, <https://www.rfc-editor.org/rfc/rfc9334>.

Authors' Addresses

Alan DeKok
FreeRADIUS
Email: aland@freeradius.org

Michael Richardson
Sandelman Software Works
Email: mcr+ietf@sandelman.ca