Machines Have Identities Too! Extending NIST's SP 800-63 for Device Identity

Tobias Hilbig[0000−0002−2904−4758], Erwin Kupris[0000−0002−2799−5197], and Thomas Schreck[0000−0002−8960−6986]

HM Munich University of Applied Sciences, Munich, Germany
{tobias.hilbig, erwin.kupris, thomas.schreck}@hm.edu

Abstract. User authentication has evolved from simple password-based procedures to phishing-resistant biometric methods. NIST, in special publication 800-63, provides definitions and requirements for digital identities. However, there is a growing need to also identify and authenticate the device in use. Such information can be included in fine-grained policy decisions to further enhance an enterprise's security posture. In addition, device authentication has been described in the literature as a significant factor in zero trust architectures. Despite the adoption of this security architecture by major stakeholders, device authentication remains lacking. Therefore, we propose extensions to SP 800-63 that cover device identity aspects. In addition, we present a best-of-breed solution using FIDO2 and an extension for OpenID Connect. Our results demonstrate that the integration of device identity aspects is feasible and aligns well with the existing guidelines. The proposed scheme can pave the way for a future where device authentication will become the norm in enterprise networks.

Keywords: Digital Identity · Device Identity · Zero Trust Architecture · Identification · Authentication · Federation · OpenID Connect · FIDO2

1 Introduction

User authentication and authorization are well-researched topics. The "Digital Identity Guidelines" by NIST, published as SP 800-63 [1], specify definitions and requirements for the digital identity landscape of users. The publication encompasses the initial identity proofing and enrollment, authentication, and federation processes. For each activity in the identity lifecycle, NIST defines three assurance levels representing their strength and security level. High-assurance levels require the usage of two-factor authentication for user authentication.

It can be desirable to identify and authenticate the device in use in addition to the user. Such a network security approach is especially useful for high-security enterprise environments. In recent years, Zero Trust Architecture (ZTA) has become an important research area in this field. This network security paradigm can be summarized as "Never trust, always verify" [2]. In ZTAs, access to data and services must always be authenticated. In addition to the acting user, this must include device authentication. Device identity and state can be used to enhance policy-based access decisions.

Although device authentication has been discussed in the literature, its usage is relatively low in practice. In this work, we aim to bridge the existing, user-focused digital identity guidelines established by NIST with the emerging concept of device identity. The goal of this study is to evaluate the guidelines' applicability to devices and to propose suitable extensions. The proposed extensions can lay the foundation for future standardization efforts. Therefore, our study can pave the way toward a future where device authentication will become the default in enterprise networks, significantly enhancing security.

Our contributions are: (1) We first provide context by giving a detailed analysis of the NIST SP 800-63. (2) Building on this, we consider how each of the three sub-standards can be extended to cover devices. (4) We also propose an extension to OpenID Connect to include device assertions in federated environments. (3) Finally, we propose ways to realize device authentication in enterprise environments using modern authentication protocols such as FIDO2.

The remainder of this paper is organized as follows: In Section 2, we explore work related, while Section 3 provides the necessary background information. Extensions to the three sub-standards of the NIST SP are given in Section 4, Section 5, and Section 6, respectively. We discuss the proposed extensions and provide ideas for future work in Section 7. This paper concludes with Section 8.

2 Related Work

We surveyed the literature on device authentication: In an extensive literature review of wireless, physical layer authentication by Xie et al. [3], existing schemes were analyzed and categorized. A detailed comparison and evaluation of device authentication schemes in the Internet of Things (IoT) domain was conducted by El-hajj et al. in 2019 [4]. In their conclusions, the authors state that a combination of software and hardware authentication should be considered. Sehn et al. proposed BASA, a blockchain-based device authentication mechanism for industrial IoT [5]. Their solution allows the establishment of privacy-preserving trust across different domains. Designs for physically uncloneable functions (PUFs) were proposed by Suh et al. [6]. These designs enable the authentication of individual integrated circuits. They concluded that low-cost authentication and secure cryptographic key generation are achievable with PUFs. The examined literature mostly focused on device-to-device authentication without user involvement. In contrast, our study primarily examines scenarios that require user participation.

In addition to academic literature on device authentication, commercial offers that implement this concept exist. However, due to their proprietary nature, technical details are not publicly available. For example, Microsoft Entra [7] and Google BeyondCorp Enterprise [8] are enterprise security solutions that support device authentication features. While "Entra" is integrated into the Windows operating system, Google uses the Chrome browser to collect information about the device in use. Both offerings allow configuration of fine-grained access policies that take the device into account.

3 Background

We present existing methods for device authentication, discuss SP 800-63, and outline our considerations and adaptations of the guidelines regarding device identity.

3.1 Existing Device Authentication Methods

A broad range of authentication techniques exist that ensure authorized devices gain access to resources. Traditional methods include MAC and IP address filtering. They offer basic levels of control but are vulnerable to spoofing. Authentication via IEEE 802.1X [9] constitutes a more sophisticated approach. This includes basic authentication mechanisms based on preshared secrets and methods that leverage digital certificates and PKIs [10]. Based on a unique certificate issued to each device, TLS [11] with mutual authentication can be used at the application layer to validate the device identity. By combining certificates with hardware-based security features such as Trusted Platform Modules (TPMs), very high levels of assurance can be achieved on modern devices.

3.2 NIST Special Publication 800-63

The Digital Identity Guidelines, published by NIST as SP 800-63 [1], comprise a set of technical requirements that focus on digital identities. In the main document, NIST specifies their "Digital Identity Model". The sub-documents define assurance levels for each activity in the digital identity lifecycle. Finally, it offers guidance on digital identity risk management and structured processes for selecting appropriate levels for a given business case.

The digital identity model includes a high-level description of all entities involved in digital identity processes: A *subject*, typically a user, can play three roles depending on its state during transactions. It is called *applicant* when initiating the identity proofing process. After successful identity proofing, they are called *subscriber* and a *subscriber account* is created. The subscriber account is a collection of the subscriber's attributes and has one or more *authenticators*, i.e., *credentials*, bound to it. Subscribers who have completed the identity proofing process and want to authenticate themselves to a service are called *claimant*. After successful authentication, the user is recognized as a subscriber.

The so-called *Credential Service Provider (CSP)* identity-proofs applicants, manages the subscriber account, and binds the established authenticators to the subscribers. The *Verifier* handles claimants' authentication requests by verifying that they are in possession and control of the authenticators bound to the claimed subscriber account. In federated setups, *Identity Providers (IdPs)* commonly function as both the CSP and Verifier. They issue authentication assertions and relay them to *Relying Partys (RPs)*. RPs offer services to authorized subscribers. To make authorization decisions in a federated model, the RP depends on the information provided in the assertions generated by the IdP. In a non federated setup, the RP relies on the information contained in the subscriber account.

In the latest revision of the guidelines, some definitions have been adjusted to accommodate device identities in future versions. Furthermore, NIST states that device identity is not explicitly addressed; however, by referring to generic subjects throughout the document, the guidelines may also be applied to devices. However, no further adjustments regarding devices were made in the draft of the next revision. Here, we clarify some terms related to devices in a real-world enterprise context: For example, the CSP can be considered part of the asset management process a company employs for devices that interact with its services. If a device is identity-proofed, it is enrolled in the asset management. Similarly, a device that is identity-proofed but has no valid session is the claimant.

4 Enrollment and Identity Proofing

NIST SP 800-63A defines the requirements for the first activity related to digital identities, i.e., the identity proofing process and subsequent enrollment. First, we describe how NIST specifies user identities. We then propose our extensions to device identities and explain how we mapped them.

4.1 User Identities

The NIST SP defines three steps for the identity proofing flow: (1) The CSP collects identifying attributes from the applicant. (2) It validates these attributes in terms of authenticity, validity, and accuracy. (3) Finally, the CSP verifies the evidence, proves the applicant, and creates a subscriber account.

The three Identity Assurance Levels (IALs) represent the strengths of the identity proofing and enrollment processes. At IAL1, there is no requirement to correlate an applicant to a real-world identity, and all attributes collected during the process should be considered self-asserted. These attributes must be validated and verified for IAL2, creating a stronger correlation between the claimed identity and the real identity. To achieve IAL3, verification must be performed either in-person or remotely under supervision over a secure and trusted channel.

4.2 Device Identities

The process flow defined by NIST can generally be adopted for device enrollment. However, we must adjust the assurance levels to better portray the reality of enrolling a device into an enterprise's asset management. Our proposed levels, called Device Identity Assurance Levels (DevIALs), are built on top of each other.

DevIAL1: For the first level of device assurance, we propose to closely align with the definition by NIST regarding user identities. Similar to IAL1 for user identities, in the proposed DevIAL1, the CSP must not validate or verify attributes. While the CSP may request self-asserted attributes from the applicant device, this is not required. Therefore, any identifier and any additional attributes provided by the device are accepted without further validation or verification. As an example for DevIAL1, consider access to enterprise resources

via a web browser. The user agent may serve as an identifying attribute which is not validated by the CSP. The resource being accessed sets a session cookie in the browser, which serves as the authenticator. Upon future visits, the resource can reliably determine that the browser in question accessed it in the past.

DevIAL2: For the second level, the identity proofing process must validate and verify the identity of the requesting device. The verification process involves checking whether the respective identifiers exist in device registries or other databases. Although such identifiers can be spoofed by other devices, the validation provides stronger security guarantees than the anonymous registration at DevIAL1. NIST defines several requirements for the second level, which we map to devices: The first requirement concerns Personally Identifiable Information (PII). This does not apply to devices because they do not provide PII. The second, third, and fourth requirements concern the strength of the evidence provided by the applicant. In this work, we refrain from categorizing all possible identifying attributes according to their strengths. For the second IAL, NIST allows for in-person and remote identity proofing, which we also allow for devices. In the device identity context, *in-person* means that the process is conducted or supported by a human, such as an administrator. *Remote* identity proofing is a fully automated process without human intervention. The remaining requirements, e.g., biometric data collection and trusted referees, are applicable only to humans; thus, they have no counterpart in device identities. An example for DevIAL2 at the application layer is the enrollment of a device into an asset management system based on the device serial number. The CSP checks whether this serial number is present in the order confirmation of the device manufacturer. If this is the case, the device is considered trustworthy and assigned an authenticator, e.g., an X.509 certificate.

DevIAL3: NIST tightens the validation and verification requirements for level three but does not add new requirements. In addition, remote identity proofing at this level is permitted only under certain restrictions: The entire process must be conducted under direct supervision of the CSP, e.g., via a high-resolution video link. As with IAL2, the requirements at this level are very specific to human users. Therefore, we restrict the allowed types of evidence to strong, cryptographically verifiable ones. Similar to people, devices can be issued verifiable identification attributes or credentials through authoritative and trusted parties, such as the device vendor. In addition, enterprises that enroll a device into their asset management system can validate these attributes to ensure the device's trustworthiness. For the third level, we propose that such credentials are protected by hardware measures to prevent malicious cloning. A vendor-assigned digital certificate stored in the device's TPM is an example for DevIAL3. The CSP can verify the certificate using the vendor's public key which provides a compelling assurance of the device's identity.

5 Authentication and Lifecycle Management

NIST SP 800-63B specifies user authentication and lifecycle management requirements. In the following, we describe NIST's approach to user authentication. We propose ideas to extend it to device authentication and define our so called Device Authentication Assurance Levels (DevAALs).

5.1 User Authentication

NIST specifies technical requirements and guidelines for secure user authentication processes. This includes the use of multi-factor authentication (MFA), password management, and cryptographic techniques. Three Authentication Assurance Levels (AALs) are defined to classify the strength and security of authentication processes: AAL1 represents the lowest level of assurance, requiring single-factor authentication, which may include a username and password or a single biometric factor. For AAL2, MFA is required to provide a higher degree of assurance. Two different factors, such as something you know, e.g., a password, and something you have, e.g., a mobile device, are required to enhance security. AAL3 offers the highest level of assurance, mandating the use of hardware-based authenticators with proof-of-possession and verifier impersonation resistance, and ensures the highest confidence in the identity assertion's validity. These levels allow organizations to select appropriate security levels based on information sensitivity and the potential risks associated with the authentication process.

5.2 Device Authentication

For human users, identity proofing and authentication procedures often differ; however, for devices, these processes are closely related. Unlike users, devices can only use technical properties, keys, and certificates for identity proofing. The same factors can also be used for authentication. Therefore, the proposed device authentication assurance levels align well with the DevIALs described in Section 4. Similar to NIST, we do not require DevIAL and DevAAL to match for individual devices. We differentiate the envisioned DevAALs along five characteristics, as shown in Table 1. In the following, we describe our proposed DevAALs and provide real-world examples.

Table 1. Overview of Device Authentication Assurance Levels

Concept	**DevAAL1**	**DevAAL2**	**DevAAL3**
Authentication Method	any	CR	CR
Credential Protection	none	Software	Hardware
Credential Secrecy	no	yes	yes
Unique Credential	no	yes	yes
Spoofing possible	yes	yes	no

DevAAL1: By definition, this level only provides *some* assurance about the authentication strength. For devices, this means that the CSP must not require a specific authentication method. The device is not required to protect the credential by specific means. This also allows credentials to be shared among multiple devices. Consider the following example for DevAAL1: At the network and internet layers, MAC-based frame and IP-based package filtering can be applied. Session cookies can be considered at the application layer.

DevAAL2: We propose that at this level, the device must authenticate to the CSP via a cryptographically strong challenge-response (CR) protocol. Using such a protocol mitigates the risk of replay attacks and credential theft. The authentication method must also employ software measures to reduce the risk of disclosing credentials. Lastly, DevAAL2 uniquely identifies a device to the CSP. For example, authentication at the network layer can occur via 802.1X for DevAAL2. The client certificate used by the device can be stored without hardware protection, e.g., in the operating system certificate store.

DevAAL3: At the third level, hardware measures are required to protect credential data and ensure that they never leave the device. Similar to the second level, a cryptographically strong CR protocol is required. This enables a phishing-resistant authentication process that is protected by multiple layers of defense. For DevAAL3, 802.1X with the certificate protected by hardware measures, e.g., by a TPM, can be considered.

6 Federation and Future Approaches

In addition to analyzing NIST's considerations for federation, we propose an OpenID Connect (OIDC) extension and a comprehensive FIDO2-based solution.

6.1 Federation

NIST SP 800-63C specifies the requirements for federated operation and assertion transmission. The three Federation Assurance Levels (FALs) provided by NIST are directly applicable to devices. Therefore, no adjusted, device-focused FALs are required: While a bearer assertion signed by the issuer is required for FAL1, the assertion must also be encrypted for FAL2. To prevent malicious parties from using stolen assertions, signed and encrypted holder of key (HoK) assertions are required for FAL3. HoK assertions require that the party presenting the assertion to cryptographically proof the possession of the key referenced within the assertion itself.

6.2 Device Assertions via OpenID Connect

OIDC is a modern identity layer built on top of the OAuth 2.0 protocol [12], designed to facilitate user authentication in a standardized manner [13]. It allows federated RPs to verify the identity of an end user after authenticating at an authorization server, i.e., the IdP. OIDC employs a signed token, called ID token,

that carries the user's identity information, thereby ensuring its verifiability. To send and receive device-specific assertions in federated setups, an extension to the OIDC protocol is required. The proposed extension covers device identities and allows both user and device authentication to occur simultaneously.

In OIDC, the RP can request user authentication using the *openid* scope. Similarly, RPs should be able to signal that device authentication should be performed by the IdP. To this end, we introduce the *device_auth* scope. If an IdP capable of device authentication receives an authorization request containing this scope, it should attempt to authenticate the device used in the transaction.

After successfully authenticating the device, the IdP generates a unique token we call the *device token*. This token is designed to carry device-specific attributes recorded in the device's subscriber account. In addition to these attributes, other relevant metadata about the device may be added to the device token, such as its DevAAL or dynamic information about the device's security posture. Like OIDC's ID token, the device token is signed by the IdP. This ensures that the token is secured against tampering, and the RP can trust that the device-specific information provided by the IdP is accurate and secure. A device token enhances security by allowing more granular access and control based on device-specific contexts, thereby providing a more robust framework for managing identities and access from multiple devices.

Finally, we define the new *deviceinfo* endpoint which functions similar to OIDC's established userinfo endpoint. This endpoint is specifically tailored to retrieve additional device-specific information, such as its hardware configuration, software versions, and security posture. The deviceinfo endpoint can be accessed by RPs possessing a valid access token, which allows them to query additional device attributes. This capability is particularly useful in environments where device integrity and context are critical for security measures and operational decision-making.

6.3 FIDO2 Solution

Following our definitions in Section 5, FIDO2 credentials [14] can be used to authenticate devices at various levels of assurance. Syncable passkeys and FIDO2 credentials stored on hardware security keys enable users to authenticate seamlessly using different devices. Therefore, these types of FIDO2 credentials cannot uniquely identify a device and thus only fulfill the requirements of DevAAL1. For higher assurance levels, only credentials bound to the exact device that is authenticating can be considered. Device-bound passkeys [15] represent such a FIDO2 credential, guaranteeing that the credential never leaves the device on which it was created. When these passkeys are stored in software and are not protected further, they satisfy the definition of DevAAL2. For DevAAL3, hardware protection of the credential is required. When a FIDO2 credential is stored within the TPM, it provides a robust layer of security by binding the user's identity to the device's hardware. In this case, the IdP can be sure that the credential is hardware-protected and can only be mapped to a single device; thus, the credential is highly resistant to phishing, theft, and replication attacks.

Table 2. Extended Digital Identity Assurance Levels Including Devices

Concept	Level	User Assurance	Device Assurance
Enrollment & Identity Proofing	IAL1	Self-asserted attributes	Anonymous
	IAL2	Remotely verified attributes	Weak identifier
	IAL3	In-person verified attributes	Strong identifier
Authentication & Lifecycle Management	AAL1	Single Factor	Any token
	AAL2	MFA	SW-based CR
	AAL3	HW-based MFA	HW-based CR
Federation & Assertions	FAL1	sig{Bearer assertion}	
	FAL2	enc{sig{Bearer assertion}}	
	FAL3	enc{sig{HoK assertion}}	

Therefore, device-bound passkeys saved in the device's TPM can authenticate devices at DevAAL3. Thus, both the user and device can be authenticated simultaneously using FIDO2, providing a seamless and secure user experience. Although this best-of-breed procedure works in a web-based context in which a user is involved, it cannot be employed in machine-to-machine scenarios.

7 Discussion and Future Work

We present exemplary technologies for each assurance level of the NIST guidelines for user authentication and our device-focused extension in Table 2. In the following sections, we discuss our findings and provide directions for future work.

7.1 Discussion

We have proposed integrating of device assurance aspects into NIST SP 800-63. During our study, we noticed the following aspects that are worth highlighting: Similar to users, devices can have unique identifying attributes, such as serial numbers, MAC addresses, and IMEI numbers for mobile devices. For these attributes, we identified the following characteristics: (1) They can either be uniquely identifying or shared among a batch of devices. (2) Some are permanent, whereas others may change over time. (3) Attributes may be public, secret, or protected from being spoofed by hardware measures. In addition, devices do not necessarily need to enroll additional credentials after the identity proofing process. In fact, devices can reuse the identifying attributes used as a means to authenticate later. For example, a MAC address can be used as an identifying attribute during identity proofing and as a credential for network access. Consequently, identity proofing and authentication of devices are more aligned compared to these same processes regarding user identities. Furthermore, users typically authenticate at the application layer, unlike devices that can authenticate at various layers of the Internet model [16]. Finally, device authentication can either be closely related to user authentication or operate independently of user involvement. In the former

case, both processes can occur simultaneously, e.g., when a user visits a website using a company-owned device. The latter case represents a scenario in which no human is involved, for example in machine-to-machine communication and IoT environments.

7.2 Future Work

We derive and develop directions for future work based on our findings: A viable extension to this work would be an analysis of all existing device enrollment and authentication protocols and procedures. These studies can assess these approaches along dimensions in terms of ease of use, technical complexity, and authentication strength. Furthermore, continuous access evaluation for user authentication is well established. How to continuously collect, transmit, and evaluate device-related information could be of interest. Moreover, we did not explicitly focus on workload identity aspects in our study. Future work should focus on how to consider workload-related information in machine-to-machine communication. Another direction for future work is the standardization of the proposed device assurance levels. Similarly, the proposed extensions to OpenID Connect need to be discussed by relevant standardization bodies. Similar to user identities, investigating the privacy implications of implementing device assurance levels, including data collection, storage, and sharing practices, is important. Future research could focus on mitigating potential privacy risks while maintaining a high level of assurance.Finally, the logical next step after device assurance is device trust, which was out of scope for this paper. Securely collecting relevant information from devices requires future attention.

8 Conclusion

Despite ongoing research and increasing industry interest, device authentication is not commonly performed. NIST, in its special publication 800-63, provides guidelines for digital identities with a focus on the user. In this paper, we propose ways to extend SP 800-63 to cover devices. Here, we discuss how the identity proofing, authentication, and federation aspects described by NIST can be mapped to devices. Our results demonstrate that extending the requirements to devices is indeed possible and aligns well with NIST's user-focused guidelines. We also presented an OpenID Connect extension for device assertions in federated environments. In addition, we have demonstrated that combining user and device authentication is possible using a highly assured, FIDO2-based solution. These proposals can lead to a future where device authentication will become the norm in enterprise contexts.

Competing interests All authors declare that they have no conflicts of interest.

References

1. Grassi, P.A., Garcia, M.E., Fenton, J.L.: NIST Special Publication 800-63-3: Digital Identity guidelines. National Institute of Standards and Technology, Los Altos, CA (2017). https://doi.org/10.6028/NIST.SP.800-63-3
2. Buck, C., Olenberger, C., Schweizer, A., Völter, F., Eymann, T.: Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero-trust. Computers and Security **110**, 102436 (2021). https://doi.org/10.1016/j.cose.2021.102436
3. Xie, N., Li, Z., Tan, H.: A Survey of Physical-Layer Authentication in Wireless Communications. IEEE Communications Surveys & Tutorials **23**(1), 282–310 (2021). https://doi.org/10.1109/COMST.2020.3042188
4. El-Hajj, M., Fadlallah, A., Chamoun, M., Serhrouchni, A.: A survey of internet of things (IoT) authentication schemes. Sensors **19**(5), 1141 (2019)
5. Shen, M., Liu, H., Zhu, L., Xu, K., Yu, H., Du, X., Guizani, M.: Blockchain-Assisted Secure Device Authentication for Cross-Domain Industrial IoT. IEEE Journal on Selected Areas in Communications **38**(5), 942–954 (2020). https://doi.org/10.1109/JSAC.2020.2980916
6. Suh, G.E., Devadas, S.: Physical unclonable functions for device authentication and secret key generation. In: Proceedings of the 44th Annual Design Automation Conference. DAC '07, pp. 9–14. Association for Computing Machinery, San Diego, California (2007). https://doi.org/10.1145/1278480.1278484
7. Microsoft Corporation, Zero Trust Model - Modern Security Architecture | Microsoft Security, (2022). https://www.microsoft.com/en-us/security/business/zero-trust (visited on 08/19/2024).
8. Osborn, B., McWilliams, J., Beyer, B., Saltonstall, M.: Design to Deployment at Google. Usenix Login **41**(1), 28–35 (2016)
9. IEEE Standard for Local and Metropolitan Area Networks–Port-Based Network Access Control. IEEE Std 802.1X-2020 (2020). https://doi.org/10.1109/IEEESTD.2020.9018454
10. Simon, D., Aboba, B., Hurst, R.: The EAP-TLS Authentication Protocol. Tech. rep. RFC 5216, Internet Engineering Task Force (2008). https://www.rfc-editor.org/rfc/rfc5216
11. Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.3. Tech. rep. 8446, 160 pp. Internet Engineering Task Force (2018). https://www.rfc-editor.org/info/rfc8446
12. Hardt, D.: The OAuth 2.0 Authorization Framework. Tech. rep. 6749, 76 pp. Internet Engineering Task Force (2012). https://www.rfc-editor.org/info/rfc6749
13. Sakimura, N., Bradley, J., Jones, M., De Medeiros, B., Mortimore, C.: OpenID Connect Core 1.0, The OpenID Foundation (2014).
14. FIDO Alliance, FIDO2 - FIDO Alliance, (2024). https://fidoalliance.org/fido2/ (visited on 08/19/2024).
15. Hodges, J., Jones, J., Jones, M.B., Kumar, A., Lundberg, E.: Web Authentication: An API for accessing Public Key Credentials Level 3. Tech. rep., W3C (2023). https://www.w3.org/TR/webauthn-3/
16. Braden, R.T.: Requirements for Internet Hosts - Communication Layers. Tech. rep. 1122, 116 pp. Internet Engineering Task Force (1989). https://www.rfc-editor.org/info/rfc1122