

# W3C Workshop on Identity in the Browser 24/25 May, Mountain View, CA

## Position paper from Netflix Inc.

### Introduction

Netflix provides a subscription service enabling subscribers to view movies and TV show over the Internet using a variety of internet-connected devices.

Our user interfaces are increasingly implemented in HTML on a variety of platforms and we interpret the term “browser” to include any application providing the HTML/CSS/JS platform on a device, be that a desktop PC, a television, games console or set-top-box.

Today, our service requires proprietary integrated software on each device (or plugins on desktop browsers). In future we would like to see services such as ours supported as web applications. Secure identity is an important part of that.

### Identity in the Netflix service

The Netflix service makes use of two kinds of identity: a user identity and a device identity. User identity is authenticated by means of common web sign-in methods, ultimately based on username and password. Device identity is ideally based on secure device credentials provided by the device manufacturer (either at the time of manufacture or dynamically in a device-specific way).

Device authentication is important to a service like Netflix for several reasons:

Firstly, it is necessary for us to reliably determine the security properties of the device, specifically with respect to content protection and the properties of the device identity itself. This enables us to make a decision as to whether the device is authorized to play premium content.

Secondly, our terms of service place limits on the number of devices single subscriber can access. Policing of these limits is only possible if devices can be securely identified (otherwise, for example, a single subscription could be shared by 100s of devices).

Thirdly, it is sometimes necessary to revoke access to the service for a device whose security has been compromised.

Many other commercial services, particularly commercial video services, will have similar requirements.

## **Our interest in Identity in the Browser**

Presently, although we make use of web technologies for user interfaces, we cannot use the emerging video playback technologies of the web platform due in part to the absence of a secure method of authenticating devices and handling protected content. We would like to see that change.

We believe device authentication is a service requirement that is not specific to content protection. A simple, open and general-purpose mechanism for device authentication, accessible by web applications through a Javascript API, would have value to a wide variety of applications.

It should be recognized that the type of identity and the security properties afforded by that identity will vary substantially between device types, and this is ok. Some devices may support mutable identities, thereby affording very weak (or non-existent) security properties. Other devices may support physically secure hardware identity modules, affording very different security properties. The security properties are implicit in the identity itself, and the secure credentials associated with it, rather than being asserted or guaranteed by the API. For example, an identity for a TV from Sony might have different properties than one in a device from Hacker Inc.

## **A note on privacy**

It is clear that any such device authentication mechanism needs to address issues of Personally Identifiable Information and provide strict user control of how and to whom such information is transmitted.

For example, different origins may see different identities for the same physical device, preventing this identity being abused for user tracking. It must be necessary for users to explicitly consent to the transmission of device identities to an origin and the technology must ensure that an origin itself is authenticated to the user's satisfaction before an identity is transmitted to it.

## Conclusion

Many commercial services require the ability to securely authenticate the device being used to access the service. Presently this can only be done through proprietary means. We would like to propose standardization of a simple, open and extensible Javascript API for secure device authentication – essentially providing access for web applications to the services of an arbitrary secure identity module on the device.

Such an API would enable a variety of services that depend on guarantees about device behavior, including commercial services based on premium content such as Netflix.