Easy 802.1X Onboarding with EAPConfig files and Supplicant Configuration Automatic Discovery

Gareth Ayres, Information Services and Systems, Swansea University, UK
Stefan Winter, RESTENA

Keywords—Wireless, eduroam, Supplicant, 802.1x, Onboarding, EAP

EXTENDED ABSTRACT

Enterprise networks are now well established and there exists many standards which are used to guarantee security and provide mechanisms for devices and users to attach to the LAN or WLAN. An example of this is eduroam[1] which is built on top of IEEE 802.1X[2] to allow academics and students from participating institutions to securely roam to networks all over the world using their home institutions credentials.

While eduroam has proven massively successful, the guarantee of secured roaming and access is based on the assumption that user's devices (that is to say, their supplicants) are configured correctly and fully. While BYOD users need to configure their devices to facilitate access, it is possible to partially configure the supplicant so that access is granted but the end device is not fully secured against all methods of attack. A fully configured device will provide network access, guarantee the security of the user's credentials and be privacy-preserving.

With the massive selection of devices and operating systems there exists many supplicants, and different interfaces and tools used to configure them. Some enterprise environments where the device is provisioned and managed, supplicant configuration can be achieved during provisioning of the device. But the typical BYOD environment usually puts the burden of configuration onto users.

The complexity and time involved in manually configuring some supplicants has resulted in some users only partially configuring their devices in order to obtain network access, but not fully securing themselves. Supplicant configuration automation tools such as eduroamCAT[3] and SU1X[4] have gone a long way to improving the laborious and confusing task of configuring a supplicant fully. These tools have simplified the onboarding of devices onto enterprise networks like eduroam where BYOD is dominant, while ensuring that devices are fully configured and secure.

These tools have largely achieved this by bundling configuration information such as EAP authentication method settings and certificates with custom built versions of the tools, catered to particular users of a particular institution. This model works well when users can download and run the packaged up configuration tool, preconfigured by their home institutions network administrators. But with the increase in devices which only run applications from trusted repositories, such as the Apple app store and Google Play, then the customised tools distribution model fails. Apple have addressed this onboarding problem by specifying a MobileConfig[5] file format that can be consumed by Apple devices in order to configure them correctly. However, this proprietary approach does not help interoperability and users of other devices.

Onboarding devices would be made easier and safer for users if all supplicants and their respective operating systems had the mechanism to consume the necessary configuration information and configure themselves fully and automatically. To achieve this a standardized approach to defining supplicant configuration information is required. It is proposed that this can be achieved through the EAPConfig file; a file format for transferring configuration information of deployments of the Extensible Authentication Protocol to supplicants.

The EAPConfig file will provide a machine readable file that can be consumed by a device, which describes all the EAP configuration information required by a supplicant in order to fully configure itself and attach securely to a network. The EAPConfig file will contain the information provided by network administrators, as would typically be bundled with current configuration tools, with the ultimate goal that every operating system would know how to consume it.

The SENSE project has developed an Android app as a working example of how EAPConfig file can be consumed, and automatically fully configure a device and connect it securely to a network. The android app consumes EAPConfig files with a MIME-Type of application/eap-config. Browsing to an eap-config file

using a typical web browser will cause the app to launch and parse the EAPConfig file. The user can then choose to inspect it, enter any additional information such as username and password, and then connect.

An eduroamCAT version of the Android app has also been developed to show how the EAPConfig file generated by institution administrators on the eduroamCAT website can be used to simplify onboarding of BYOD Android devices onto eduroam. The eduroamCAT app also demonstrates a possible solution to the app repository problem, where one app can be presented to users, instead of either requiring users to run untrusted apps or hosting an app per institution in each repository/store.

This process would ideally be made even simpler for users if devices could automatically discover the EAPConfig file, and any additional information needed without requiring users of BYOD devices to first install an app and then browse to the config file location. As installing the app typically requires internet access in order to access the app repositories/stores. This presents a catch-22 scenario that the authors fully appreciate, but this could be resolved through operating systems adoption of the processes this paper proposes.

The Supplicant Configuration Automatic Discovery (SCAD) process is an additional technique that allows devices to discover the EAPConfig file, along with any other files/information necessary to onboard a device onto a network using some context-awareness, removing the requirement of users to manually browse to an appropriate file location. We propose an initial approach with three auto-discovery techniques; DNS Lookup, Realm Lookup and Location Awareness.

DNS Lookup would perform a lookup on a addressed composed of the devices local domain name and a prepended scad subdomain, similar to the well-established WPAD process [6]. Realm Lookup would require interaction from the user, to enter an email address, so the realm could be used to perform the DNS Lookup by prepending scad to the users realm. This would work well where mobile network access is used for device configuration. The third technique uses the Location Awareness services present in most modern devices, or GeoIP, to localize the user and search online databases for configuration files. This may only be suitable for particular instances of enterprise networks like eduroam, where centralized authorities could store this information. In fact, eduroamCAT implements a version of this already.

The SCAD process could also discover additional configuration files, such as an IEEE80211Profile file, which is similar to the EAPConfig file but contains adapter specific information such as which SSID, authentication or encryption to use. While it is the aim to improve the ease which users fully configure supplicants, the SCAD process needs to be mindful of potential misuse, accidental or malicious. Signing SCAD files could help protect users, and suitable warnings/logos could be used to avoid users discovering the wrong configuration files.

ACKNOWLEDGMENTS

This work was carried out as part of the GEANT open call SENSE (Secure Enterprise Networks finally Simple and Easy) project.

REFERENCES

[1] eduroam, http://www.eduroam.org
[2] IEEE 802.1x http://standards.ieee.org/findstds/standard/802.1X-2010.html
[3] SU1X, http://su1x.swan.ac.uk
[4] eduroamCAT, https://cat.eduroam.org/
[5] Apple MobileConfig
https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/iPhoneOTAConfiguration/Introduction/Introduction.html
[6] WPAD http://tools.ietf.org/id/draft-ietf-wrec-wpad-01.txt

AUTHOR BIOGRAPHIES

Gareth Ayres – Gareth Ayres works as a Wireless Network Officer for Information Services and Systems (ISS) at Swansea University. He previously completed an undergraduate degree at Swansea before working in ISS, and then completed a master's degree in communication systems before starting a PhD with the engineering department part-time. He now works and researches in the area of wireless networking and privacy and enjoys combining the practical side of his work with ISS with the theoretical aspect of his PhD. He is a member of the JANET UK SIG on 802.1x, and developer of the SU1X supplicant configuration tool.
Contact: email: g.j.ayres@swansea.ac.uk Phone: +441792602235

Stefan Winter - Stefan Winter graduated in Computer Science at the University of Karlsruhe, Germany,

in September 2004, with a specialisation in telematics and foundations of Computer Science. He is working as R&D Engineer for the Luxembourg Research and Education Network RESTENA, where network roaming and identity federations are in the focus of his activities. He led the R&D work for eduroam during the GN2 and GN3 projects. In the GN3plus project, he is member of the eduroam Operational Team in Europe (leading the development of the eduroam CAT software), and a participant in the SENSE OpenCall project. He is one of Europe's representatives in the Global eduroam Governance Committee.
Contact: email: stefan.winter@restena.lu