Introduction

eduroam is a popular Wi-Fi roaming service with thousands of sites and millions of users in the United States alone. Using a combination of technologies, users can seamlessly join wireless networks at participating sites. This document describes shortcomings in monitoring the interdomain/end-to-end authentication that underlies the eduroam user service and proposes a distributed testing and monitoring system.

Problem Statement

Problems in the underlying technologies, primarily 802.1X and RADIUS, can be hard to detect, troubleshoot, and resolve even in a single domain. The operation of the eduroam federated RADIUS roaming system requires at least three domains: the visiting institution, the top-level (US) RADIUS service, and the user's home institution. International roaming adds additional steps.

Network operators may not be aware of a problem until users notice, *and report*, they are unable to associate to eduroam. This assumes users know eduroam is available and should be working- if our goal is to have devices join the wireless network without user action (or even knowing eduroam is available), it needs to be as seamless as cellular roaming. When a device doesn't join eduroam, a user may not even know there is a problem. Problems may occur for many reasons, such as improper RADIUS configuration and the home and/or visiting institution. Failures may even be caused by specific institution-pairwise or path-specific characteristics. Having a test-point that does not require coordination and action by users will allow us to provide better service.

Simply put - failures are often silent and unreported. Section 3 of RFC 7539 - The eduroam Architecture for Network Roaming has additional information on problems that can arise in a federation RADIUS system.

Sources of Problems: Two Examples

Two examples help illustrate the problems operators face in monitoring and troubleshooting eduroam.

EAP-TLS, a certificate-based EAP method, was failing for users visiting an overseas institution (it's also possible it was failing for other users, but went undetected or unreported). Some RADIUS messages were reaching the home institution, while others were not. The only way this was detected was by a user reporting authentication failures. The authentication path was: visiting institution--eduroam.es--[eduroam.eu?/]--eduroam.us--home institution. The network path involved a similar number of organizations. Troubleshooting required coordination with the

visiting user in a distant time zone (wasting the user's time), and a lot of guessing given limited access to RADIUS logs along the path.

The root cause was RADIUS traffic being dropped due to an MTU path problem. Having an autonomous endpoint at the visiting institution would have allowed engineers at the home institution to more quickly and easily identify authentication failures.

The automated intrusion detection system at an institution falsely flagged incoming RADIUS traffic from the US top-level servers as problematic. Rules were incompletely distributed to border firewalls resulting in some inbound traffic being dropped (but all outbound traffic being allowed). Authentication success and failures were inconsistent. This was clearly a self-inflicted problem, but having a distributed testing system, especially with regularly scheduled tests, might have detected this before a user experienced a problem.

Further Complications

Even a well performing wireless network is a messy environment, with clients associating, disassociating, roaming (and failing to roam), losing power, etc. RADIUS is also a noisy protocol requiring interaction between the client and network, and client and RADIUS servers. In the best case scenario, only three RADIUS services are involved, but an operator troubleshooting may have access to only logs from the home server. Even if logs from TLRS are consulted, that requires interpretation of logs from multiple vendors.

Additionally, high failure rates are not that uncommon. Partly for reasons stated above (uncontrolled client behavior), legitimate clients may fail to authenticate. But in higher education, approximately 25% of student users will leave each year. Unless they delete eduroam from their mobile devices, they may still attempt to associate and authenticate to any eduroam service, resulting in an authentication failure, causing 'noisy' logs where failures are easy to ignore.

Proposal

We propose a distributed authentication testing system, similar in concept to perfSONAR, RIPE Atlas, and NLNOGRing. What follows is an *initial concept* of what such a system might look like.

Small-form factor (single-board) computers that have both wired ethernet and wireless capabilities are proposed for the testing endpoints. The wired port should use power over ethernet to power the device, which allows for remote rebooting. The wired ethernet port is expected to have a stable address and be used for management. These devices would be hosted at eduroam service provider sites and provide for realistic end-to-end testing of the same authentication path that production traffic would use.

Authorized eduroam admins should be able to set up realistic tests to simulate user behavior by configuring appropriate EAP types and parameters. Such tests could be either one-off / ad-hoc which is useful for incident troubleshooting or regularly scheduled and repeated for long term monitoring and surveillance. Tests can be managed through perfSONAR components (ie, pscheduler) or an automation platform such as Ansible or Puppet.

It is *not* expected that tests other than basic Wi-Fi association/authentication be performed. Performance and availability testing, while beneficial, are not the goal of this system. Nor is it expected that testers will have direct shell access to the endpoint (NLNOGRing is useful for that type of testing).

Considerations and Open Questions

It is expected that institutions will provide their own hardware to host test points. The incentive to purchase and host is that tests deployed to these endpoints will test outbound (visiting user) authentication.

The management interface of the test point will need to allow inbound connections from the management platform.

Testing capacity might become a concern if this service becomes popular but the number of endpoints can't meet the demand. A fair system to allow access might be needed. Institutions might be incentivized to purchase and host endpoints with a credit system similar to RIPE Atlas[1], though that will add complexity.

Development and management of the environment needs to be determined.

Access to the environment should be managed through the existing eduroam administration portal.

Conclusion

eduroam is a growing service, even adding K-12 users and sites. Currently, network operators supporting eduroam lack tools to monitor and test the service. A distributed testing environment, based on low-cost single-board computers, will improve service by detecting and reporting problems before users notice.

[1] RIPE Atlas is a distributed test environment. Sites can host test endpoints and earn credits that can be used to run tests on other RIPE Atlas probes.