Securing the Mobile Network

Product Overview

Meetinghouse’s 802.1X Authentication Toolkit is a highly portable embeddable source code library for adding 802.1X authenticator functionality to Network Access Server (NAS) devices such as wireless access points, wired/wireless routers and switches.

Key Benefits

- Commercially tested and deployed by leading network equipment manufactures (e.g. Hewlett Packard, NEC, D-Link, and many others.)
- High Quality Source Code
  - Highly portable
    - All hardware and OS dependencies have been extracted from the 802.1X Toolkit library and are represented by a number of callback functions
  - Production-level code
    - Extensive QA and compatibility testing done on the 802.1X Toolkit
  - Fully Standards Compliant
    - Tested against Microsoft XP Supplicant
    - Tested against industry leading RADIUS servers including Funk Steel-Belted Radius, Microsoft IAS, Cisco ACS, Interlink and FreeRadius
  - Easy implementation of 802.1X MIBs
  - Highly configurable
- Reference Platform
802.1X Authentication Toolkit Data Sheet

- A simple-to-setup reference platform, the "asd", is included with the 802.1X Toolkit. It runs as a daemon and is intended to be used as a guide for implementation.
- A simple client tool communicating with the reference daemon is included to demonstrate how to use the tools available for 802.1X mib implementation.
- Can be compiled to operate in VxWorks or Linux
- Can be compiled for wired or wireless environments

- 802.1X Toolkit Library
  - Full compliance with IEEE 802.1X specification
  - EAPOL-KEY (dynamic re-keying) support included
  - Full RADIUS Client functionality
  - Both RADIUS authentication and accounting functionality included
  - RADIUS server failover support

How it Works/System Design
The 802.1X Toolkit is designed to carry out user authentication and accounting functionality. Meetinghouse’s design reflects the idea of multiple services that require user authentication (i.e. 802.1X, telnet, etc) and multiple protocols for doing the authentication and accounting. In order to achieve this design objective, the 802.1X Toolkit consists of six modules: (i) interface manager, (ii) 802.1X, (iii) authentication dispatcher, (iv) accounting dispatcher, (v) local authentication and (vi) RADIUS services.
(i) **Interface Manager** - The interface manager module is a simple abstraction of a port for maintaining the state of the virtual ports currently under management in the device.

(ii) **802.1X** - The 802.1X module maintains the front and back-end authenticator state machines for each port currently being controlled. It processes EAPOL packets, processes port status changes (up/down), transmits EAPOL packets and requests blocking/unblocking of ports when their state changes.

(iii) **Authentication Dispatcher** - The authentication dispatcher module is an abstraction of the authentication process. It acts as a buffer between tasks that require user authentication (such as the 802.1X task) and the authentication protocols available for carrying out the authentication process. This design was chosen so that new user authentication modules and new authentication protocols could be added with relative ease. New user authentication modules that you might add would be telnet, console logon or even a special method for using a non-802.1x compliant supplicant’s MAC address to do PAP authentication.

(iv) **Accounting Dispatcher** - The accounting dispatcher module is an abstraction of the accounting process. It acts as a buffer between tasks requiring accounting and tasks providing accounting protocols. Currently 802.1X is the only task requiring accounting and the only accounting protocol is RADIUS.

(v) **Local Authentication** - The local authentication module is one of the authentication protocols supported. This is the most simple. It will allow you to do EAP-MD5 with a single configurable user id and password. This module is present for testing functionality without requiring a RADIUS server.

(vi) **RADIUS** - The RADIUS module is the actual authentication and accounting protocol supported. This module creates RADIUS packets, reads RADIUS packets and parses RADIUS packets. It implements full RADIUS client functionality including failover to a new RADIUS server when the current one stops responding. This module supports PAP, CHAP, EAP authentication.

**Porting Requirements (hardware/OS dependencies)**

The 802.1X Toolkit was designed to be easily ported to many platforms. To achieve this objective, all OS and hardware dependencies were removed from the 802.1X Toolkit and replaced them with callback functions.

During initialization of the 802.1X Toolkit, a single initialization function is called and a single configuration structure is passed into the 802.1X Toolkit. This configuration structure defines all the callback functions that the 802.1X Toolkit requires for implementation.
The 802.1X Toolkit support required is demonstrated by the reference platform. This block diagram (figure XX) outlines the required functionality into a number of modules: (i) physical port support, (ii) timer support, (iii) message dispatcher support, (iv) semaphore support, (v) message output support and (vi) fatal error support.

(i) **Physical Port Support** - Physical port support is the largest module of support. The support consists of mapping between physical and logical port numbers, maintaining physical state of the ports, blocking/unblocking the ports and detecting changes in the link/associate status of the ports, programming WEP keys for the ports and filtering out the EAPOL packets to be sent to the 802.1X Toolkit for processing. The 802.1X Toolkit library works only with virtual or logical ports. The rest of the tasks are hardware dependent and will involve using the knowledge of specific hardware and drivers to integrate our 802.1X Toolkit to specific platform.

(ii) **Timer Support** - Timer support has two components: timer task and time tools. The timer task will send a message to a particular task at a particular time in the future. The time tools abstract the idea of a time stamp and determining a time difference between two time stamps. The time tools use the standard C time functions.

(iii) **Message Dispatcher** - The message dispatcher/task management component manages all creation/destruction of tasks and message passing between tasks. All tasks within the 802.1X Toolkit communicate through message passing.
(iv) Semaphore Support - The semaphore support provides support to the Toolkit for protecting shared resources. The current implementation uses posix mutexes.

(v) Message Output - The message output module provides support for debug output from the 802.1X Toolkit. This module may be edited to direct output to a common area for specific platforms. Each message consists of a text message and a severity level from informational to fatal.

(vi) Fatal Error Support - The fatal error handler provides control over the actions of specific platform if the 802.1X Toolkit encounters a fatal error.

Note: We believe that much of the functionality of the reference platform will be repurposed for all implementations porting to Linux or VxWorks. For example, the short list of required porting for Linux is quite small: block/unblock support, link/associate detection, bridging support, snmp/mib support (if you wish this support on your platform) and programming dynamic WEP keys into your hardware.

Standards Supported

- RFC 2284 – PPP Extensible Authentication Protocol (EAP)
- IEEE P802.1x – Standard for Port based Network Access Control
- RFC 2865 – Remote Authentication Dial In User Service (RADIUS)
- RFC 2869 – RADIUS Extensions
- Internet-Draft <draft-congdon-radius-8021x-09.txt> - IEEE 802.1X RADIUS Usage Guidelines
- RFC 2548 – Microsoft Vendor-specific RADIUS Attributes

System Requirements

- The Express-Toolkit Library.
  o No hardware requirements.
- The “asd” reference platform
  o Linux with 2.4.x kernel (We have tested on Redhat 7.1, which have a Linux kernel of 2.4.2-2).
  o Openssl version 0.9.6b installed
  o libpcap version .4 (included with toolkit)
  o To test using Linux bridging you will need Linux bridging support in your kernel as well as the bridge control program (brctl). (brctl and bridge libraries included with toolkit)
  o Wireless mode has further requirements (not available for VxWorks):
    ▪ A pcmcia card based on Intersil's PRISM2 chip. You must use a pcmcia card.
    ▪ A prism2 driver that supports HostAP mode. We supply one with the Toolkit. You will need to replace this driver with your own driver because this driver does not support programming WEP keys.
    ▪ A Linux version that supports pcmcia.

Code Footprint

- The Express-Toolkit library
  o source code is 5,650 lines of code.
- The reference platform: asd
802.1X Authentication Toolkit Data Sheet

- Source code is 4,074 lines of code.
- ROM size is 296k
- RAM size (code+stack+data) with 256 virtual ports is 1.7mb. Each virtual port requires approximately 1.3k of space.

Requirements for platform OS.
- Threads/tasks
- Recursive semaphores (same task can repeatedly lock same semaphore)
- Timers with millisecond granularity

Meetinghouse Data Communications is dedicated to providing network security and authentication products that enable the widespread deployment of high-value wireless network services. Meetinghouse enables wireless network deployment by developing high-quality, standards-based security solutions that allow network operators to manage and account for usage across disparate networks.

- Mutual authentication wireless security solutions
- Evolution of 802.1X industry standards
- Modular design of embedded solutions

Figure 5: Meetinghouse 802.1X Products

Meetinghouse Data Communications – Securing the Mobile Network

Contact Meetinghouse at 603.403.7710 or sales@mtghouse.com