802.1X, EAP and RADIUS

Martin Stanek

Department of Computer Science
Comenius University
stanek@dcs.fmph.uniba.sk

Security of IT infrastructure (2016/17)
Content

Network access control

802.1X

EAP

RADIUS

Summary
Network access control

- AAA services ~ authentication, authorization, accounting

- **authentication**: verification (proving) of subject’s identity
- authorization: determining whether the subject can perform given action
- accounting: tracking the use (consumption) of network resources
  - session duration, packets and data transferred, …
IEEE Std 802.1X

- Port-Based Network Access Control
  - http://standards.ieee.org/about/get/802/802.1.html, more than 200 pages
- the standard:
  - specifies a general method for provision of port-based network access control;
  - specifies protocols that establish secure associations for IEEE Std 802.1AE MAC Security;
    (MAC – Media Access Control, part of a link layer in OSI model), encryption and integrity for Layer 2 (default AES-128-GCM)
  - facilitates the use of industry standard authentication and authorization protocols.
- example: WPA2 Enterprise (WPA2-802.1X, Wi-Fi Protected Access II)
  - cf. WPA2 Personal (WPA2-PSK, Pre-shared key)
Windows 7

- WiFi; Wired AutoConfig service for 802.1X on wired Ethernet interfaces

802.1X, EAP and RADIUS
Ubuntu 16.04 (Wired connection)

- NetworkManager

![Editing Wired connection 2](image)

- Use 802.1X security for this connection
- Authentication: Protected EAP (PEAP)
- Anonymous identity: 
- CA certificate: (None)
- PEAP version: Automatic
- Inner authentication: MSCHAPv2
- Username: 
- Password: 

(cancel save)
Ubuntu 16.04 (WiFi connection)

802.1X, EAP and RADIUS
Subjects and roles in 802.1X

- **Supplicant (client)**
  - SW, e.g. part of an operating system
  - HW, e.g. Intel AMT (part of Intel vPro platform)

- **Authenticator** – facilitates authentication of other entities

- **Authentication server** – provides an authentication service

---

Supplicant (client)

- SW, e.g. part of an operating system
- HW, e.g. Intel AMT (part of Intel vPro platform)

Authenticator – facilitates authentication of other entities

Authentication server – provides an authentication service
What’s going on in 802.1X

- initial state: port (access point) is closed for any client’s communication except EAPoL (EAP over LAN)
- client (supplicant) performs authentication against authentication server (EAP, Extensible Authentication Protocol)
  - success: authenticator opens port, assigns VLAN etc.
  - failure: authenticator keeps port closed / opens port and assigns the client to guest VLAN etc.
Protocols in 802.1X

- **EAPoL (EAP over LAN)**
  - facilitates communication supplicant ↔ authenticator
  - runs over 802.3 (Ethernet), 802.11 (WLAN), …
  - packs EAP messages into L2 communication

- **RADIUS …details later**
  - communication authenticator ↔ authentication server
  - in this scenario: EAP messages packed into messages of RADIUS protocol
Challenges for deployment

- some EAP methods need certificates – certificate management (provisioning), both server’s and supplicant’s certificates
- network devices without 802.1X support (e.g. printers)
- Wake on LAN
- multiple devices on single network port (IP phones, hub etc.)
- unavailable authentication server

…etc. …
EAP (Extensible Authentication Protocol)

- originally an extension of PPP (Point-to-point protocol), now RFC 3748
- typically over data link layer (e.g. PPP, IEEE 802; i.e. without IP)
- general authentication framework for multiple authentication methods
- packet format:

```
  code  identifier  length (2B)  data
1 Request
2 Response
3 Success
4 Failure
```

- identifier aids in matching responses with corresponding requests
- RFC 5296: new codes introduced (5 Initiate, 6 Finish)
EAP (2)

- very simple protocol
  - (potentially) large number of request/response messages, usually finished with success/failure

- example:

```
supplicant  authenticator  auth. server
request: Identity
response: Identity
request/response: authentication
success/failure
response: Identity
success/failure
```
EAP (3)

- complexity in authentication methods

<table>
<thead>
<tr>
<th>1/2</th>
<th>identifier</th>
<th>length (2B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>data for particular auth. method</td>
<td></td>
</tr>
</tbody>
</table>

- examples of authentication methods (more than 40, optional custom extensions):

  4  MD5  21  PEAP
  13 TLS  43 FAST
  21 TTLS  49 IKEv2
EAP-MD5

- mandatory method (standard-compliant implementation must support)
- implementation CHAP (Challenge Handshake Authentication Protocol):
  - Request: \textit{challenge}
  - Response: MD5(identifier $||$ \textit{shared secret} $||$ \textit{challenge})

- avoid this method – security problems:
  - only one-sided (client/supplicant) authentication
  - vulnerable to dictionary and brute-force attacks
  - vulnerable to MITM attack …messages in clear-text without any protection of integrity/authenticity
  - identity of client revealed
  - no support for cryptographic key generation – cannot protect further communication
  - …
EAP-TLS, EAP-TTLS and EAP-PEAP

Ideas (outer EAP used mostly for solving packet fragmentation):
- EAP-TLS: using TLS authentication
- EAP-TTLS: client authentication (as AVP) tunneled in TLS
- EAP-PEAP: inner EAP instance tunneled in TLS

<table>
<thead>
<tr>
<th>Feature</th>
<th>EAP-TLS</th>
<th>EAP-TTLS</th>
<th>EAP-PEAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>client certificate</td>
<td>yes</td>
<td>optional</td>
<td>optional</td>
</tr>
<tr>
<td>server certificate</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>mutual authentication</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>key generation</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>identity protection of client</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>
Some inner authentication methods

- CHAP … with MD5 was discussed before
- MS-CHAPv2 … CHAP variant (defined in RFC 2759)
  - mutual (two-way) authentication
  - free from LAN Manager history
  - generating cryptographic keys
  - frequently used in practice
  - interesting analysis (standalone MS-CHAPv2):
    Defeating PPTP VPNs and WPA2 Enterprise with MS-CHAPv2
    (DEFCON 20, 2012)
RADIUS

- RADIUS – Remote Authentication Dial In User Service
- RFC 2865, RFC 2866 (Accounting) + other extensions
- centralized authentication of users and systems
- AAA services
- client/server protocol
  - client (NAS – Network Access Server):
    switch, router, access point, VPN server …
  - server (RADIUS server):
    FreeRADIUS, Network Policy Server (Microsoft), Secure Access Control Server / Identity Services Engine (Cisco)
Basic characteristics

- stateless protocol (UDP)
- database of users: SQL database, LDAP, text files, ...
- communication client ↔ server (initialized by client)
- proxy RADIUS server (facilitates roaming of users between realms)
Packet

- **authenticator:**
  - request auth. (in Access-Request packets) – unpredictable and unique over lifetime of a secret
  - response auth. (Access-[Accept, Reject, Challenge] packets)
    MD5(code || ID || length || request auth. || attributes || secret)
  - secret – password shared by client and server
Security (1)

- user password (P) is transmitted encrypted
  - password padded with 0x00 to multiple of 16 B
  - encryption: $P \oplus \text{MD5(secret || request auth.)}$
  - other attributes in clear-text (security?, privacy?)
- value *secret*
  - dictionary attack or brute-force attack (using response auth. or encrypted password)
  - often the same values used in multiple NAS $\Rightarrow$ fake NAS, attacking user passwords
Security (2)

- vulnerability – repeating or predictability of request auth.
  - get server’s responses in advance and repeat them later (see also Event-Timestamp attribute)

- Access-Request without integrity protection
  - see Message-Authenticator attribute (HMAC-MD5 for entire packet, key is secret)

- some risks are mitigated by employing suitable EAP method

- protection of the protocol – providing secure channel
  - IPSec, RadSec – RADIUS over TLS

- RADIUS support for EAP (RFC 3579)
Alternatives and improvements

- **TACACS+** (Terminal Access Controller Access-Control System)
  - proprietary Cisco protocol, primary for access to network components
  - over TCP, separation of authentication and authorization
  - (optional) encrypted body of the packet (without header)

- **DIAMETER**
  - intended replacement for RADIUS (slow adoption)
  - basics defined in RFC 3588
  - over reliable transport layer (TCP, SCTP)
  - over secure communication channel (IPSec, TLS)
  - both stateful and stateless models
  - easy to extend, ...
Summary – architecture (802.1X example)

- **user (server) authentication**
  - MS-CHAPv2, CHAP, …
- **secure communication channel**
- **auth. server authentication**
  - EAP-TTLS, EAP-PEAP …
- **L2/L3 layer transport**
  - EAPoL, RADIUS
Summary – messages (802.1X example)

- **Supplicant**
  - EAPoL
  - EAP Request: Identity
  - EAP Response: Identity
  - EAP Request
  - EAP Response
  - TLS handshake
  - inner EAP: MS-CHAPv2
  - EAP Success

- **Authenticator**
  - RADIUS
  - EAP Request: Identity
  - EAP Response: Identity
  - Access-Request
  - Access-Challenge
  - Access-Request
  - Access-Accept

- **Authentication Server**
  - Access-Request
  - Access-Challenge
  - Access-Request
  - Access-Accept
  - port open

802.1X, EAP and RADIUS