edurogue

Captura de credenciales de usuario de clientes IEEE 802.1X mal configurados

Reducir o eliminar el problema de seguridad que suponen los clientes IEEE 802.1X mal configurados

Alberto Martínez Setién
Middleware
Comunicación y Sistemas
Servicio Informático

Deusto

Grupos de trabajo RedIRIS 2017

DEMO

Wall of Sheep - Universidad de Deusto @ Bilbao - eduroam

---[Sheep]---

Fecha	Nombre de usuario	Protocolo	Password	Tipo de hash	Sal	Hash
Thu May 4 20:26:52	83**12	mschapv2		NETNTLM	9a6....8e1	ea8....dfc
Thu May 4 20:26:46	se*******ez@opendeusto.es	mschapv2		NETNTLM	9ef....a40	10d....ed1
Thu May 4 20:24:45	un*********ra@deusto.es	mschapv2		NETNTLM	610....e82	da9....1bc
Thu May 4 20:22:12	jo*********za@opendeusto.es	mschapv2		NETNTLM	0e0....a25	09d....97d
Thu May 4 20:19:14	am**********ti@opendeusto.es	mschapv2		NETNTLM	b37....471	be3....0c4
Thu May 4 20:13:12	in**********ez@opendeusto.es	mschapv2		NETNTLM	669....81b	109....5b3
Thu May 4 20:09:51	ab*******te@opendeusto.es	mschapv2		NETNTLM	7d4....781	aa9....9ff
Thu May 4 20:09:36	sa************ez@opendeusto.es	mschapv2		NETNTLM	095....3ba	eb8....54a
Thu May 4 20:06:18	ik*******or@deusto.es	mschapv2		NETNTLM	2c9....843	838....dba
Thu May 4 20:01:21	ga***********os@deusto.es	mschapv2		NETNTLM	abc....184	50a....5bb
Thu May 4 19:59:09	en******iz@opendeusto.es	mschapv2		NETNTLM	e10....ceb	8d3....9f9
Thu May 4 19:58:28	i.*****ta@opendeusto.es	mschapv2		NETNTLM	564....50b	e73....274
Thu May 4 19:56:58	ke***bb@opendeusto.es	mschapv2		NETNTLM	2b3....447	dac....dae
Thu May 4 19:56:05	ad**********zo@opendeusto.es	mschapv2		NETNTLM	5b4....261	a15....e94
Thu May 4 19:54:41	ai**********ia@opendeusto.es	mschapv2		NETNTLM	bef....4ab	2df....154
Thu May 4 19:53:48	ma********la@opendeusto.es	mschapv2		NETNTLM	5c8....f8d	4ed....ba4
Thu May 4 19:48:49	ob****ez@deusto.es	mschapv2		NETNTLM	248....1a7	5b4....f47
Thu May 4 19:43:00	oi*********te@opendeusto.es	mschapv2		NETNTLM	08b....e6e	bac....a83
Thu May 4 19:42:45	ai***********xe@deusto.es	mschapv2		NETNTLM	cb0....f9f	e13....fdd
Thu May 4 19:42:44	ga***********xo@opendeusto.es	mschapv2		NETNTLM	f5c....531	4a3....9d2
Thu May 4 19:34:45	lo**********al@deusto.es	mschapv2		NETNTLM	198....a28	6fd....b7c
Thu May 4 19:34:44	jg*de@deusto.es	mschapv2		NETNTLM	8ee....d32	cd6....598
Thu May 4 19:34:43	oi***********ez@deusto.es	mschapv2		NETNTLM	0ea....769	538....87d
Thu May 4 19:32:54	an************ga@opendeusto.es	mschapv2		NETNTLM	652....2ce	eba....ee2
Thu May 4 19:30:45	ix*********de@opendeusto.es	mschapv2		NETNTLM	353....94f	199....b3e
Thu May 4 19:30:39	jo*********uz@opendeusto.es	mschapv2		NETNTLM	6ae....310	c32....35d
Thu May 4 19:30:08	un******de@opendeusto.es	mschapv2		NETNTLM	f7c....5d2	03a....d42
Thu May 4 19:29:44	jo*******re@opendeusto.es	mschapv2		NETNTLM	0c5....4ee	7be....d32
Thu May 4 19:29:29	84**46	mschapv2		NETNTLM	dbd....354	3b3....227
Thu May 4 19:29:23	it**********os@opendeusto.es	mschapv2		NETNTLM	956....86f	888....d6b
Thu May 4 19:29:19	al******dr@opendeusto.es	mschapv2		NETNTLM	7c2....fce	593....cbd
Thu May 4 19:29:17	da*****ea@opendeusto.es	mschapv2		NETNTLM	a1d....561	db6....d23

Primera parte:

Captura de credenciales de usuario de clientes IEEE 802.1X mal configurados

Seguridad de 802.11 (WiFi)

IEEE 802.11i (TKIP y CCMP/AES)

implementado como

WPA2

cuya versión “enterprise” es el estándar

IEEE 802.1X

basado en

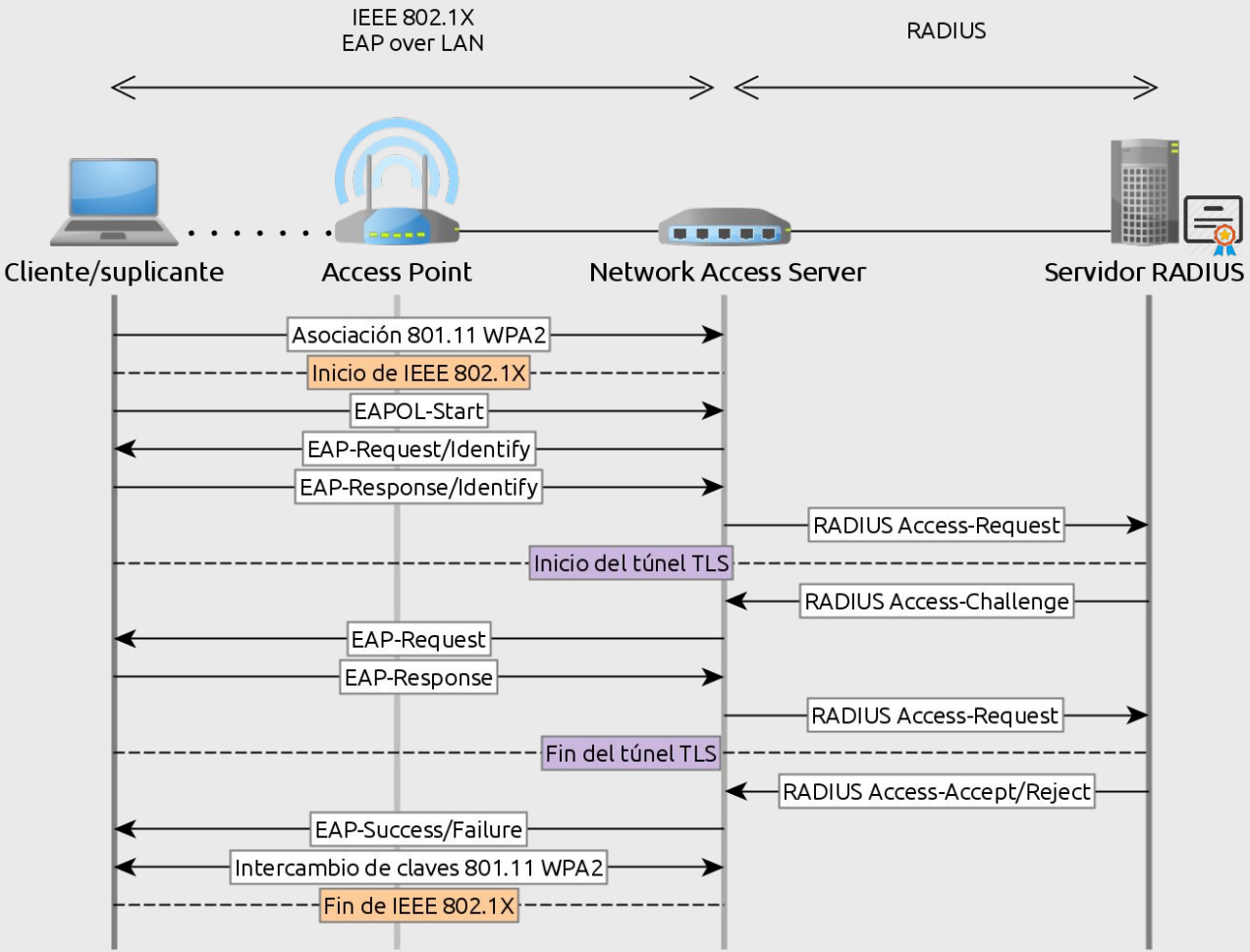
EAP (RFC 3748, RFC 4017)

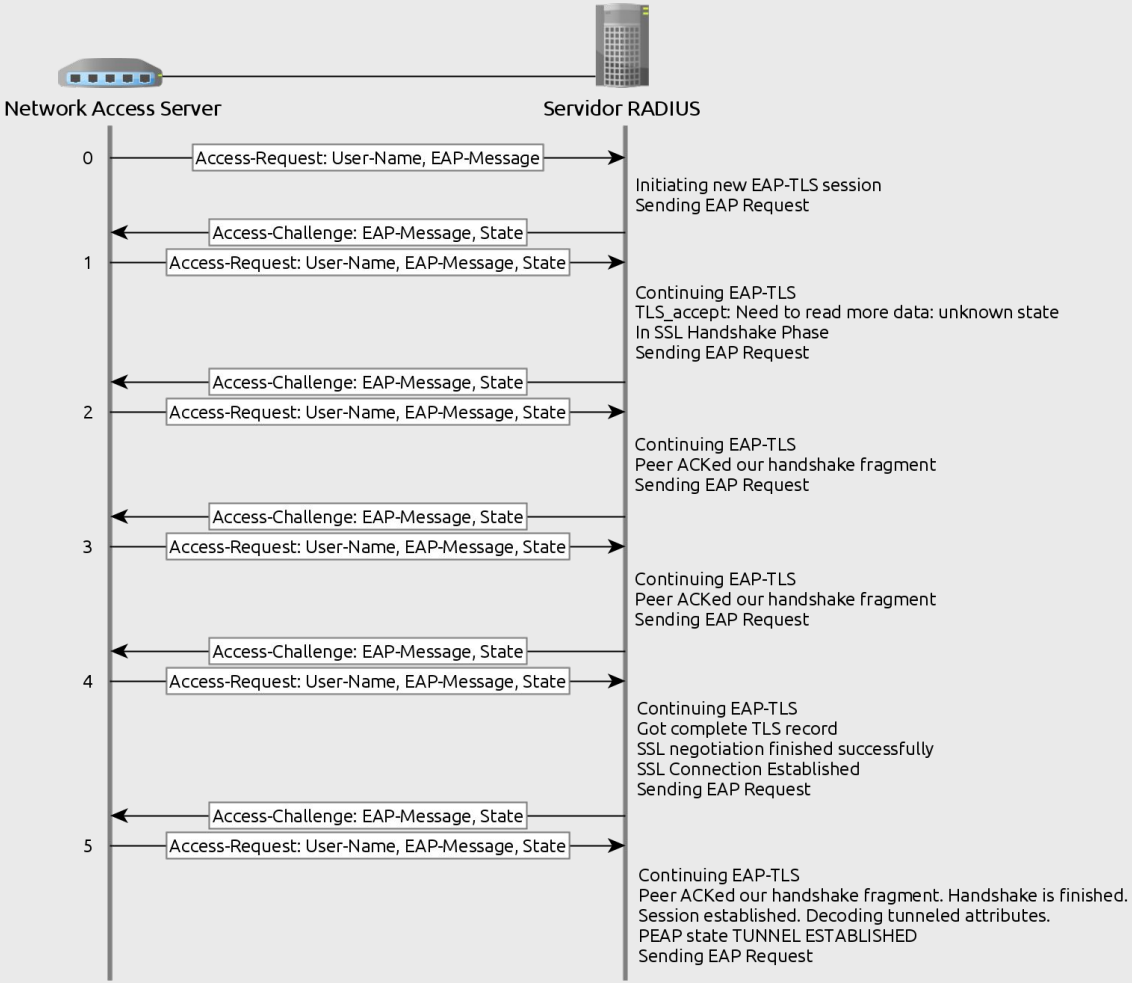
EAP-MD5 - inseguro
EAP-PEAP
EAP-SIM - RFC 4186 - Redes móviles 3G
EAP-TLS - RFC 5216 - Autenticación mutua
EAP-TTLS

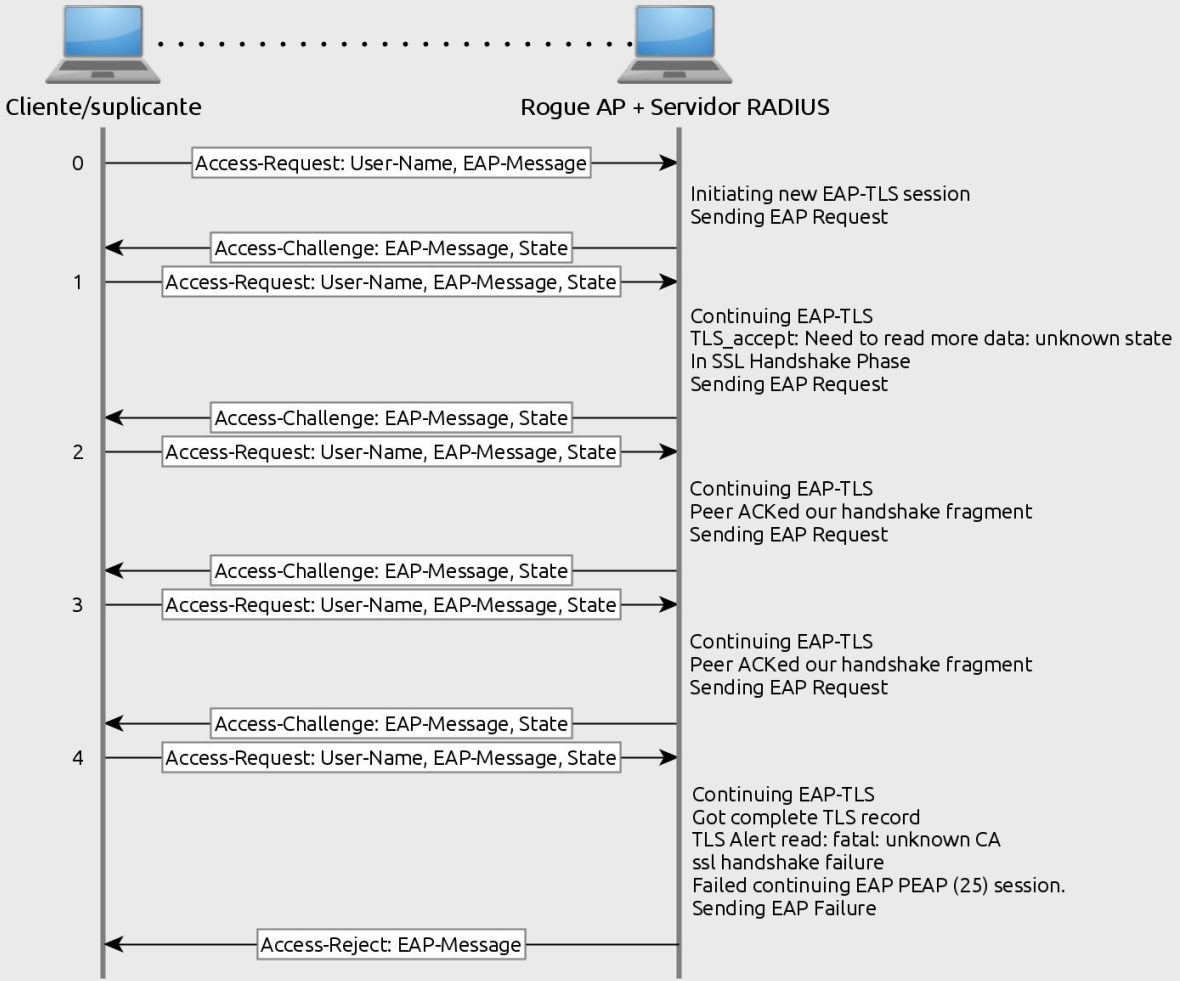
(hay unos 40 métodos EAP)

Protocolo “Remote Authentication Dial-In User Service” (RADIUS)

- **Entre otras características, encapsula mensajes EAP.**
- **No es el único protocolo de autenticación capaz de llevar mensajes EAP; también existe Diameter.**
- **El transporte de mensajes se hace mediante UDP.**
- **Stateless.**


IEEE 802.1X
EAP over LAN
RADIUS
Cliente/suplicante
Access Point
Network Access Server
Servidor RADIUS
Asociación 801.11 WPA2
Inicio de IEEE 802.1X
EAPOL-Start
EAP-Request/Identify
EAP-Response/Identify
RADIUS Access-Request
Inicio del túnel TLS
RADIUS Access-Challenge
EAP-Request
EAP-Response
RADIUS Access-Request
Fin del túnel TLS
RADIUS Access-Accept/Reject
EAP-Success/Failure
Intercambio de claves 801.11 WPA2
Fin de IEEE 802.1X


Network Access Server
Servidor RADIUS
0
Access-Request: User-Name, EAP-Message
Initiating new EAP-TLS session
Sending EAP Request
Access-Challenge: EAP-Message, State
1
Access-Request: User-Name, EAP-Message, State
Continuing EAP-TLS
TLS_accept: Need to read more data: unknown state
In SSL Handshake Phase
Sending EAP Request
Access-Challenge: EAP-Message, State
2
Access-Request: User-Name, EAP-Message, State
Continuing EAP-TLS
Peer ACKed our handshake fragment
Sending EAP Request
Access-Challenge: EAP-Message, State
3
Access-Request: User-Name, EAP-Message, State
Continuing EAP-TLS
Peer ACKed our handshake fragment
Sending EAP Request
Access-Challenge: EAP-Message, State
4
Access-Request: User-Name, EAP-Message, State
Continuing EAP-TLS
Got complete TLS record
SSL negotiation finished successfully
SSL Connection Established
Sending EAP Request
Access-Challenge: EAP-Message, State
5
Access-Request: User-Name, EAP-Message, State
Continuing EAP-TLS
Peer ACKed our handshake fragment. Handshake is finished.
Session established. Decoding tunneled attributes.
PEAP state TUNNEL ESTABLISHED
Sending EAP Request


Cliente/suplicante
Rogue AP + Servidor RADIUS
0
Access-Request: User-Name, EAP-Message
Initiating new EAP-TLS session
Sending EAP Request
Access-Challenge: EAP-Message, State
1
Access-Request: User-Name, EAP-Message, State
Continuing EAP-TLS
TLS_accept: Need to read more data: unknown state
In SSL Handshake Phase
Sending EAP Request
Access-Challenge: EAP-Message, State
2
Access-Request: User-Name, EAP-Message, State
Continuing EAP-TLS
Peer ACKed our handshake fragment
Sending EAP Request
Access-Challenge: EAP-Message, State
3
Access-Request: User-Name, EAP-Message, State
Continuing EAP-TLS
Peer ACKed our handshake fragment
Sending EAP Request
Access-Challenge: EAP-Message, State
4
Access-Request: User-Name, EAP-Message, State
Continuing EAP-TLS
Got complete TLS record
TLS Alert read: fatal: unknown CA
ssl handshake failure
Failed continuing EAP PEAP (25) session.
Sending EAP Failure
Access-Reject: EAP-Message

Validación de CA

Comprobación de la entidad emisora del certificado de servidor

¿Proviene el certificado de una entidad en la que confiamos?

El certificado de servidor es válido si su entidad emisora es concretamente una de las configuradas para el perfil de red. Típicamente sólo una.

La configuración de varias entidades emisoras en el perfil de red se debería llevar a cabo con antelación cuando se prevea el cambio de una entidad emisora a otra.

Validación de CN

Comprobación del “nombre de servidor” en el certificado de servidor

¿Se emitió el certificado para este servidor?

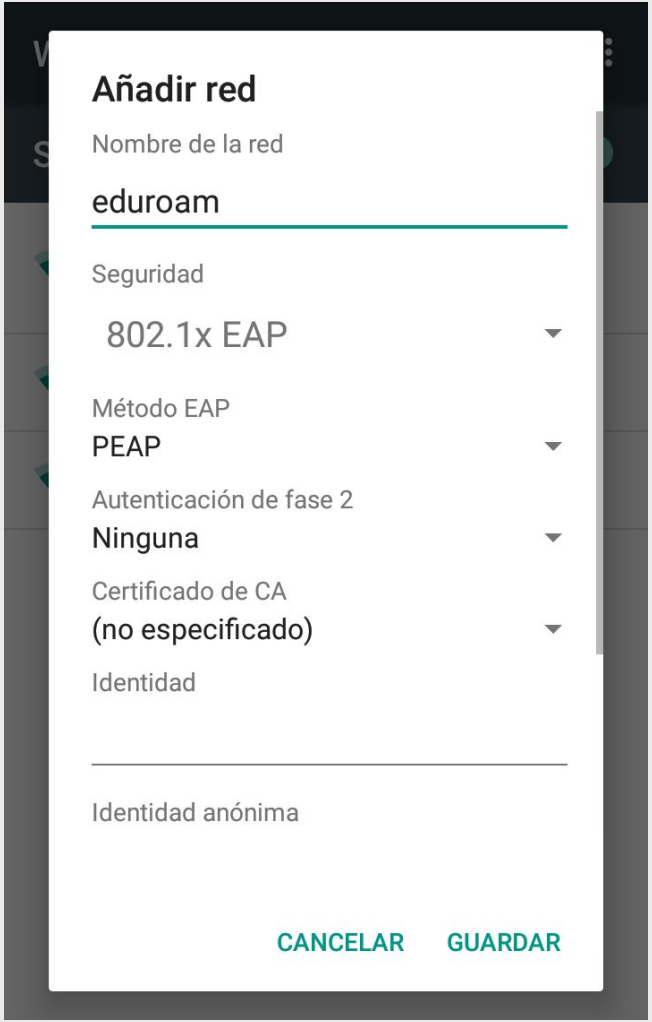
El certificado de servidor es válido si su subject, el “nombre de servidor”, coincide con el que se configuró para el perfil de red.

No tiene por qué ser el nombre DNS del servidor RADIUS.

Es más, mejor que no lo sea.

¿Qué hacen por defecto los suplicantes de sistemas operativos mayoritarios?

- **Apple**
  - **Se fija el primer certificado que se encuentre.**
- **Network Manager GUI (GNOME)**
  - **Se comprueba el certificado de servidor de forma estricta.**
  - **No hay forma de comprobar el nombre del certificado.**
- **Windows**
  - **La opción predeterminada es validar el certificado de servidor, aunque no se haya especificado uno.**
  - **Se fija la primera entidad emisora que se encuentre, siempre que esté en el sistema.**
- **Android**
  - **¡No se valida el certificado de servidor!**


Añadir red
Nombre de la red
eduroam
Seguridad
802.1x EAP
Método EAP
PEAP
Autenticación de fase 2
Ninguna
Certificado de CA
(no especificado)
Identidad
Identidad anónima
CANCELAR
GUARDAR

¿Cómo de fácil puede ser?

¿Cómo ~~reinventar la rueda~~ desarrollar un AP rogue + servidor RADIUS que vuelque credenciales?

FreeRADIUS-WPE

A patch for the popular open-source FreeRADIUS implementation to demonstrate RADIUS impersonation vulnerabilities by Joshua Wright and Brad Antoniewicz. This patch adds the following functionality:

- Simplifies the setup of FreeRADIUS by adding all RFC1918 addresses as acceptable NAS devices;
- Simplifies the setup of EAP authentication by including support for all FreeRADIUS supported EAP types;
- Adds WPE logging in $prefix/var/log/radius/freeradius-server-wpe.log, can be controlled in radius.conf by changing the “wpelogfile” directive;
- Simplified the setup of user authentication with a default users file that accepts authentication for any username;
- Adds credential logging for multiple EAP types including PEAP, TTLS, LEAP, EAP-MD5, EAP-MSCHAPv2, PAP, CHAP and others

For setup information, see the SETUP section below, or our slides from Shmoocon 4.

http://www.willhackforsushi.com/?page_id=37

Año 2008. Washington DC. 1200 asistentes.

https://github.com/OpenSecurityResearch/hostapd-wpe

```
hostapd-wpe (Wireless Pwnage Edition)
brad.antoniewicz@foundstone.com
twitter: @brad_anton
----------------------------------------

The current hostapd-wpe.patch is for: hostapd-2.6.tar.gz

About
----------

hostapd-wpe is the replacement for FreeRADIUS-WPE
(http://www.willhackforsushi.com/?page_id=37).

It implements IEEE 802.1x Authenticator and Authentication
Server impersonation attacks to obtain client credentials,
establish connectivity to the client, and launch other attacks
where applicable.

hostapd-wpe supports the following EAP types for impersonation:
    1. EAP-FAST/MSCHAPv2 (Phase 0)
    2. PEAP/MSCHAPv2
    3. EAP-TTLS/MSCHAPv2
    4. EAP-TTLS/MSCHAP
    5. EAP-TTLS/CHAP
    6. EAP-TTLS/PAP

Once impersonation is underway, hostapd-wpe will return an
EAP-Success message so that the client believes they are connected
to their legitimate authenticator.
```

Segunda parte:

Reducir o eliminar el problema de seguridad que suponen los clientes IEEE 802.1X mal configurados.

Brainstorming

- **Herramientas de detección de rogues “eduroam”**
- **Una “app password” sólo para la WiFi**
- **Algún tipo de doble factor en la autenticación**
- **“Hay una vía adicional, que sería enviar unos negros empapados en crack a las oficinas de Android, para que de una vez por todas se pueda forzar la comprobación de certificado” José Manuel Macías, 2 de Marzo de 2017**
  - **Android 7.1**

Contraseña específica

- **Al menos no es la contraseña del usuario**
- **Caducidad**
- **¿Cómo consultar la contraseña?**
- **No soluciona el problema de fondo**

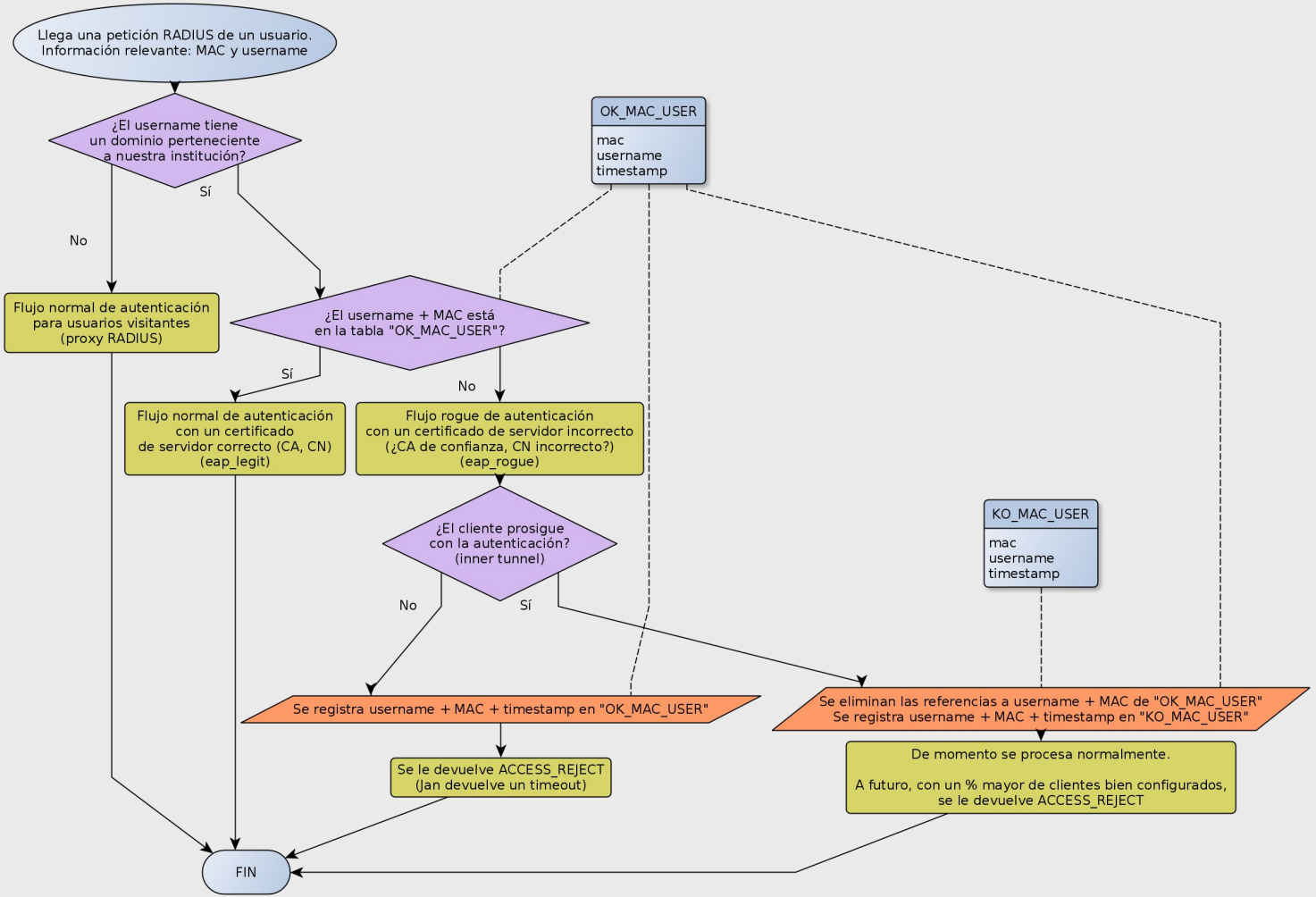
EAP-TLS

RFC 5216
Autenticación mutua

- **Difícil de provisionar**
  - **Mantener PKI**
  - **Hay que instalar el certificado de cliente en todo tipo de dispositivos**
  - **CAT ayuda poco**
    - **¿Cómo podría mejorar?**
      - **Hook de descarga de certificado cliente**
      - **Géant gestione PKI de eduroam**
      - **Problema: compartir el configurador/perfil**

¿Y si no dejásemos de montar rogue APs?

- **FreeRADIUS WPE**
- **Dos flujos de autenticación**
- **Primer intento: dos servidores RADIUS y que el NAS haga Fail-Through -> sin éxito**
- **Jan Tomasek, de CESNET (tomasek.cz, cesnet.cz) en Radiator**
  - **Flujo de un cliente desconocido: EAP rogue con timeout**
  - **Flujo de un cliente conocido: EAP normal**
  - **CESNET annual meeting**


Llega una petición RADIUS de un usuario.
Información relevante: MAC y username
¿El username tiene un dominio perteneciente a nuestra institución?
Sí
No
Flujo normal de autenticación para usuarios visitantes (proxy RADIUS)
¿El username + MAC está en la tabla "OK_MAC_USER"?
Sí
No
Flujo normal de autenticación con un certificado de servidor correcto (CA, CN) (eap_legit)
Flujo rogue de autenticación con un certificado de servidor incorrecto (¿CA de confianza, CN incorrecto?) (eap_rogue)
¿El cliente prosigue con la autenticación? (inner tunnel)
No
Sí
Se registra username + MAC + timestamp en "OK_MAC_USER"
Se le devuelve ACCESS_REJECT (Jan devuelve un timeout)
OK_MAC_USER
mac
username
timestamp
KO_MAC_USER
mac
username
timestamp
Se eliminan las referencias a username + MAC de "OK_MAC_USER"
Se registra username + MAC + timestamp en "KO_MAC_USER"
De momento se procesa normalmente.
A futuro, con un % mayor de clientes bien configurados, se le devuelve ACCESS_REJECT
FIN

mods-enabled/eap_rogue

```
eap eap_rogue {
    ...
    tls-config tls-common {
            ...
            private_key_file = ${certdir}/server.pem
            certificate_file = ${certdir}/server.pem
            ca_file = ${cadir}/ca.pem
            ...
    }
    ...
    ttls {
            ...
            virtual_server = "rogue-inner-tunnel"
    }

    peap {
            ...
            virtual_server = "rogue-inner-tunnel"
    }
    ...
}
```

sites-enabled/default

```
authorize {
    ...
    if (!ok && (&Realm == 'deusto.es' || &Realm == 'opendeusto.es')) {
            [...]
    }
    else {
            eap_legit {
                    ok = return
            }
    }
    ...
}
```

sites-enabled/default

```
        update control {
                Tmp-Integer-0 := "%{sql_edurogue:SELECT COUNT(*) FROM ko_user_mac WHERE user_mac_id = (SELECT id FROM user_mac WHERE user =
'%{request:User-Name}' AND mac = '%{request:Calling-Station-Id}')}"
        }

        if (&control:Tmp-Integer-0 > 0) {
#               reject
                eap_rogue {
                        ok = return
                }
        }
        else {
                update control {
                        Tmp-Integer-0 := "%{sql_edurogue:SELECT COUNT(*) FROM ok_user_mac WHERE user_mac_id = (SELECT id FROM user_mac WHERE
user = '%{request:User-Name}' AND mac = '%{request:Calling-Station-Id}')}"
                }

                if (&control:Tmp-Integer-0 == 0) {
                        eap_rogue {
                                ok = return
                        }
                }
                else {
                        eap_legit {
                                ok = return
                        }
                }
        }
```

sites-enabled/default

```
authenticate {
    ...
    #
    #  Allow EAP authentication.
    eap_legit

    Auth-Type eap_rogue {
            update control {
                    Tmp-String-0 := 'eap_rogue'
            }
            eap_rogue
    }
    ...
}
```

sites-enabled/default

```
post-auth {
    ...
    Post-Auth-Type REJECT {
            ...
            attr_filter.access_reject

            # Insert EAP-Failure message if the request was
            # rejected by policy instead of because of an
            # authentication failure
            eap_legit

            #  Remove reply message if the response contains an EAP-Message
            remove_reply_message_if_eap

            if (&control:Tmp-String-0 == 'eap_rogue') {
                    "%{sql_edurogue:CALL to_ok_user_mac('%{User-Name}', '%{Calling-Station-Id}', '%{Aruba-Device-Type}')}"
            }
    }
    ...
}
```

sites-enabled/rogue-inner-tunnel sin REJECT

```
authorize {
    ...
    "%{sql_edurogue:CALL to_ko_user_mac('%{User-Name}', '%{outer.Calling-Station-Id}', '%{outer.Aruba-Device-Type}')}"

    eap_rogue {
            ok = return
    }
    ...
}

authenticate {
    ...
    Auth-Type eap_rogue {
            eap_rogue
    }
    ...
}
```

Consideraciones

- **Timeout en vez de Reject**

- **Windows 10, iPhone cortan la comunicación sin “informar” al servidor RADIUS de que ellos tienen otro certificado configurado**
  - **¿Una solución de aproximación en vez de una exacta?**

E

O

T