# MSCHAPv2 against MIT Kerberos… yes, you can.

INFN
Istituto Nazionale di Fisica Nucleare

Sandro.Angius@lnf.infn.it

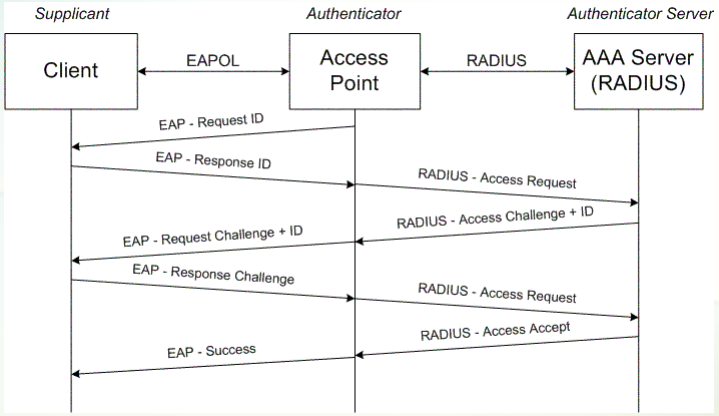
# Autenticazione su Wireless

## Requisiti

-  Sicurezza
-  Supporto multipiattaforma nativo
-  Semplicità di configurazione

# Metodi di autenticazione 802.1x

## Alcuni esempi




- TLS
  - Transport Layer Security
- TTLS-PAP
  - Tunnelled Transport Layer Security – Password Authentication Protocol
- PEAP-MSCHAPv2
  - Protected EAP – MicroSoft Challenge Authentication Protocol Version 2

# 802.1x – TLS

## Transport Layer Security

-  Nativo Windows, Apple OSX, Linux e Mobile
-  Autenticazione con certificato X.509
-  Certificato non proteggibile su Windows
  - Non utilizzabile il certificato INFN (AUP INFN)
  - Utilizzabile il certificato Terena
-  Il server di autenticazione è locale alla sede dove si effettua l'autenticazione
  - non viene demandata alla sede dell'utente tramite la struttura dei proxy

# 802.1x – TTLS-PAP

## Tunnelled TLS – Password Authentication Protocol

-  Autenticazione con username e password
-  Nativo in Apple OSX, Linux e Mobile
  - Non è nativo in Windows
    - necessario s/w aggiuntivo: Alfa&Ariss Secure-W2

-  La password viaggia in un tunnel cifrato e arriva in chiaro al server Radius
-  L'autenticazione può essere demandata via proxy

# 802.1x – PEAP-MSCHAPv2

## Protected EAP – MicroSoft Challenge Authentication Protocol v2

- 
  Autenticazione tramite username e password
- 
  Multipiattaforma
  - Windows, Mac OSX, Linux e Mobile (iPhone, Android)
- 
  Solo il challenge e il relativo reply viaggiano sulla rete (in un tunnel cifrato)
  - L'autenticazione può essere demandata via proxy
- 
  Configurazione "user-friendly"

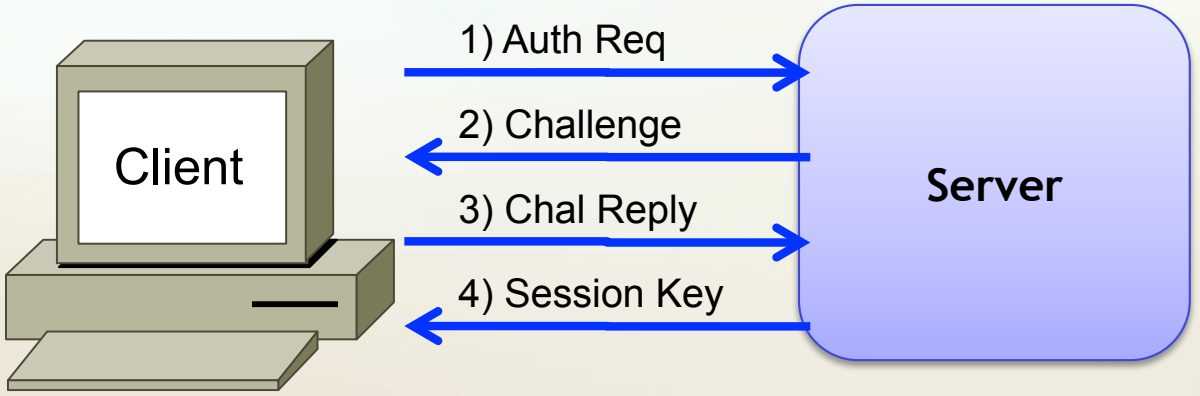
# 802.1x – PEAP-MSCHAPv2

## Impostazioni di sicurezza/privacy sul supplicant

- Verifica del certificato digitale X509 del server radius
  - Indicare esplicitamente il nome del server radius
  - Selezionare come “trusted” solo la CA di interesse
  - Non permettere l’aggiunta di nuovi server o CA
- Utilizzo dell’identità anonima (“Identity Privacy”)
  - Impedisce il passaggio in chiaro della username prima del setup del tunnel cifrato

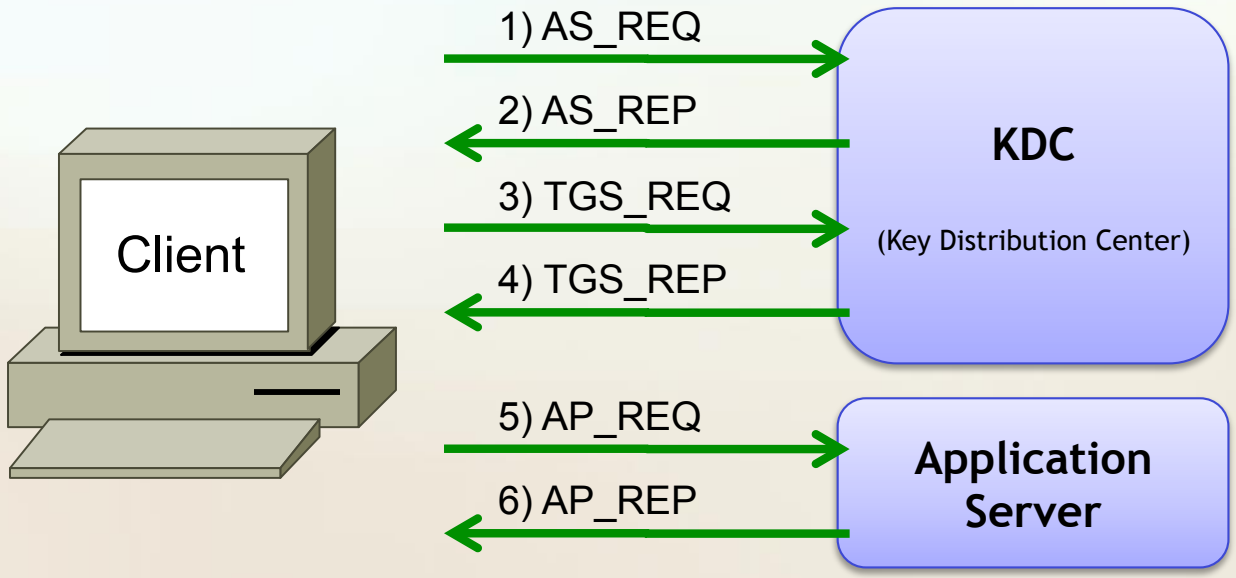
# MSCHAPv2

- Sistema di autenticazione basato su “challenge”



# Kerberos V

- Sistema di autenticazione basato su “ticket”



# Integrazione MSCHAP con Kerberos

Come?

# KCRAP

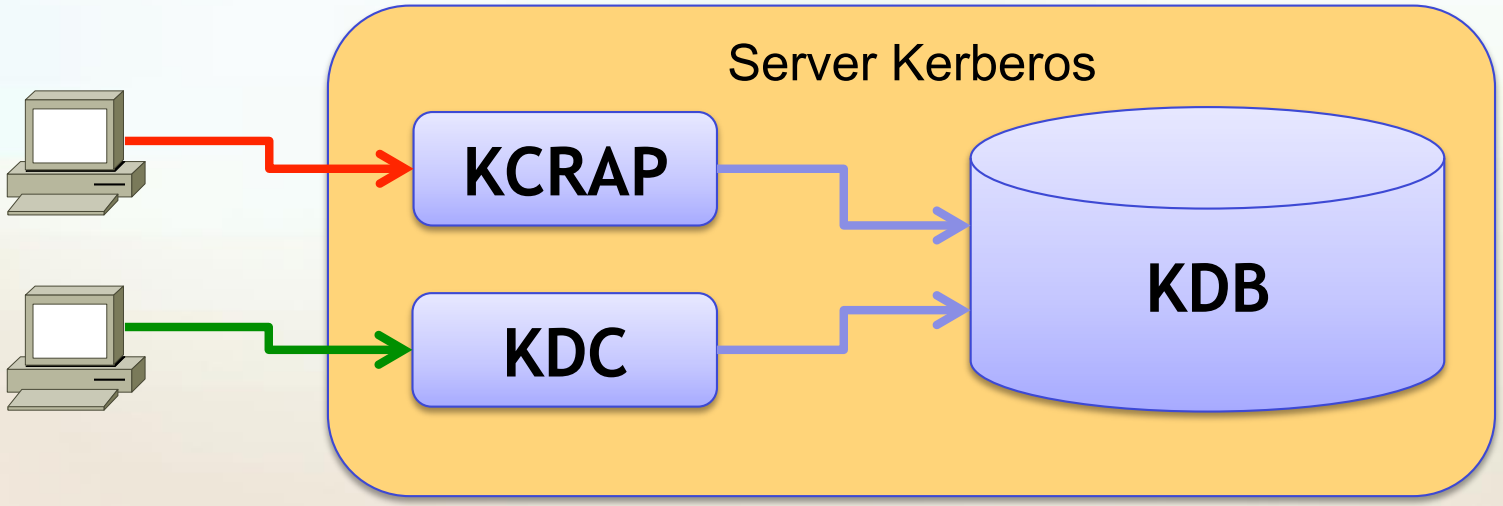
## Kerberos Challenge Response Authentication Protocol

Jonathan Chen realizza un sistema per supportare l'autenticazione NTLM (MSCHAP) con Kerberos MIT

- Nel KDC deve essere presente l'encryption arcfour-hmac:normal (= rc4-hmac:normal)
- L'accesso alle chiavi utenti avviene leggendo direttamente sul database del KDC

# KCRAP

## Kerberos Challenge Response Authentication Protocol

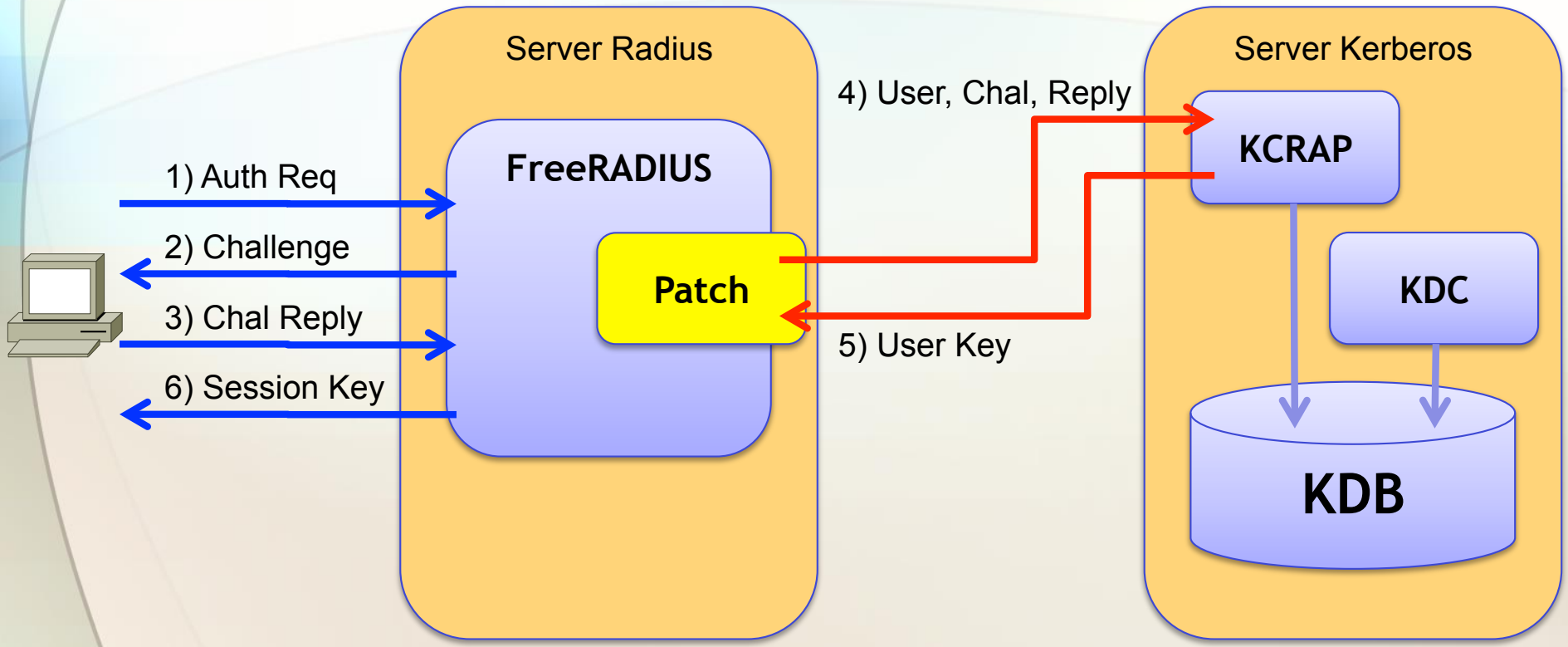


# KCRAP + Patch FreeRADIUS

Dan Fuhry realizza una patch per FreeRADIUS che permette di utilizzare il KCRAP server per l'autenticazione PEAP-MSCHAPv2

- Aggiunge il ritorno della chiave arcfour:hmac, necessaria a FreeRADIUS per completare l'autenticazione con il protocollo MSCHAPv2
- Rende KCRAP utilizzabile anche su MIT 1.9

# KCRAP + Patch FreeRADIUS



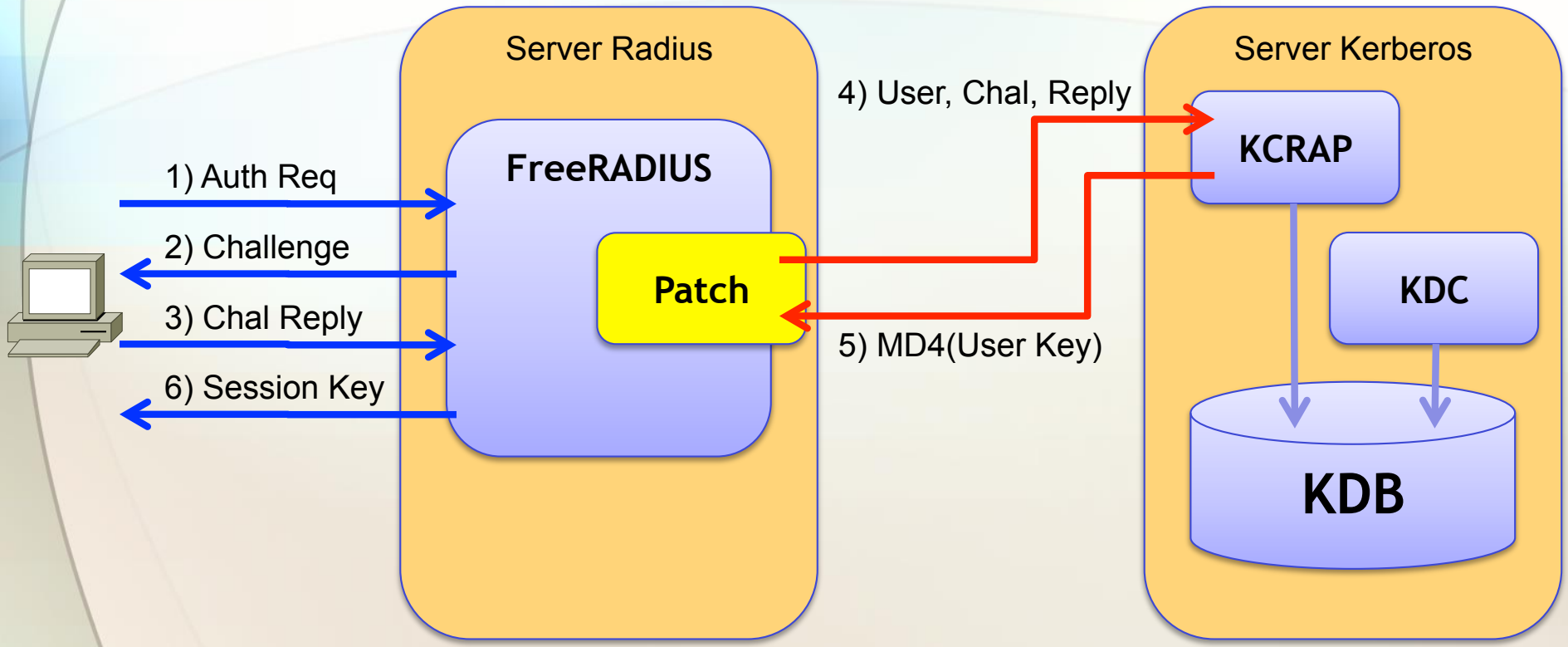
# KCRAP + Patch FreeRADIUS

## Prime modifiche

- In accordo con Dan:
  - Viene corretto un baco che impediva il corretto ritorno della chiave arcfour:hmac al FreeRADIUS
  - Viene eliminato il ritorno diretto della chiave cifrata, sostituendolo con il suo hash MD4

# KCRAP + Patch FreeRADIUS

## Prime modifiche

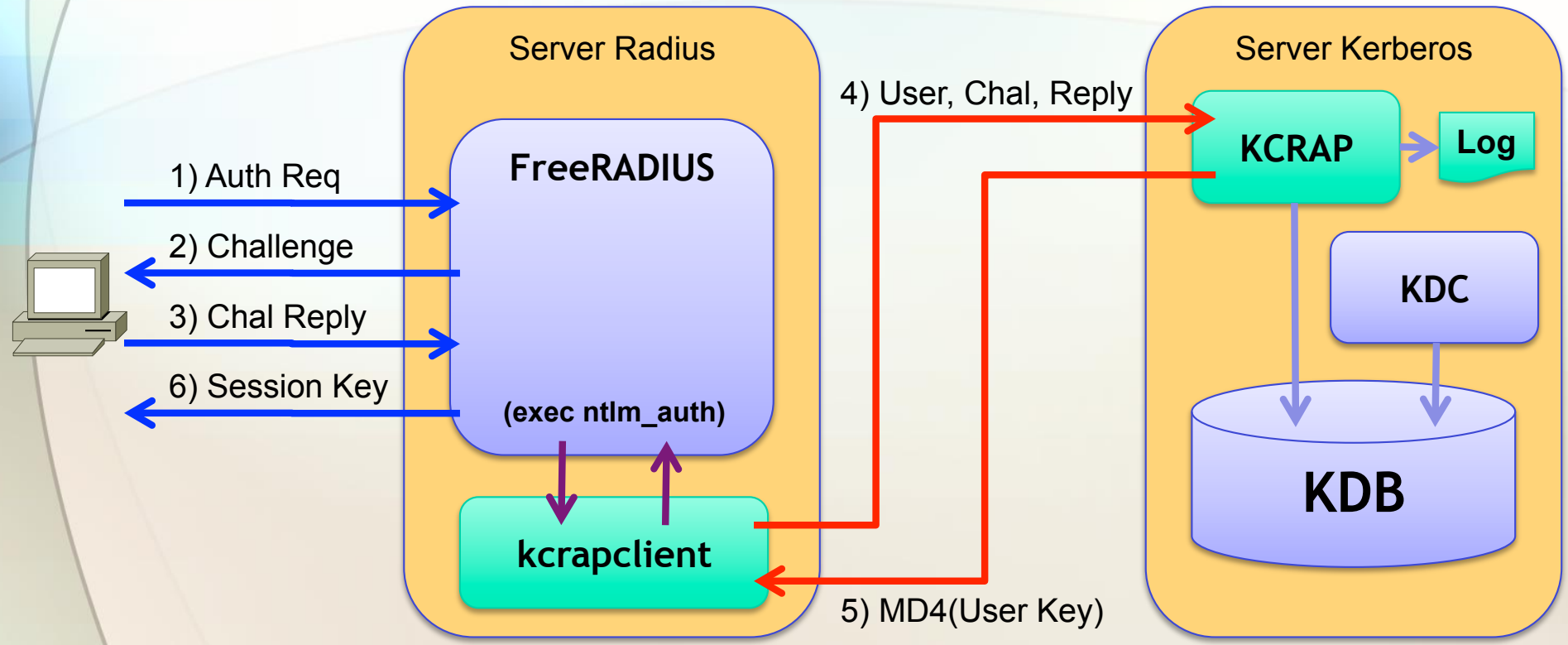


# KCRAP + FreeRADIUS
(con kcrapclient)

- Viene creato un kcrapclient che permette di eliminare la patch al FreeRADIUS
  - Di fatto simula l'utilizzo del comando "ntlm_auth" già supportato da FreeRADIUS
  - Kcrapclient controlla il numero e la correttezza lessicale dei parametri e li passa al server kcrap per la verifica del reply al challenge
- Sul kcrap server viene aggiunta la possibilità di avere il log delle connessioni

# KCRAP + FreeRADIUS

## (con kcrapclient)




**ntlm_auth = "kcrapclient %{%{Stripped-User-Name}:-%{%{User-Name}:-None}} %{%{mschap:Challenge}:-00} %{%{mschap:NT-Response}:-00}"**

# MSCHAPv2 via MIT Kerberos
## Setup

- http://wiki.infn.it/strutture/lnf/dr/calcolo/wireless/dot1x
  - Documento completo di Massimo Pistoni sul 802.1x
  - Paragrafo dettagliato sull'implementazione di kcrap:

http://goo.gl/swem5

Sw per MIT1.6: http://goo.gl/5OPB0

Sw per MIT1.9: http://goo.gl/YK7RT

# Configurazione KCRAP Server

*/etc/kcrap_server.conf*

```
[kcrap_server]
        port = 1999
        realm = LNF.INFN.IT

[realms]
 LNF.INFN.IT = {
        database_name = /var/kerberos/krb5kdc/principal
        key_stash_file = /var/kerberos/krb5kdc/.k5.LNF.INFN.IT
 }
```

*/etc/krb5.conf*

```
...
...

[logging]
        ...
        kcrap_server = SYSLOG:INFO:LOCAL3
        # kcrap_server = FILE:/var/log/kcrap.log
        ...
        ...
...
...
```

# KCRAP Server
## (logging)

*LOCAL3.debug*

```
May  2 18:01:44 kdcs3 kcrap[2766]: Datagram of 602 bytes received from address: 193.206.84.29.
May  2 18:01:44 kdcs3 kcrap[2766]: Request for username 'user1' from 193.206.84.29 with challenge 0xbd50a3fccd292811.
May  2 18:01:44 kdcs3 kcrap[2766]: Authentication for username 'user1' from 193.206.84.29 failed.
May  2 18:02:14 kdcs3 kcrap[2766]: Datagram of 610 bytes received from address: 193.206.84.29.
May  2 18:02:14 kdcs3 kcrap[2766]: Request for username 'user2' from 193.206.84.29 with challenge 0x5b994e4f1ea511d1.
May  2 18:02:14 kdcs3 kcrap[2766]: Authentication for username 'user2' from 193.206.84.29 succeeded.
```

*LOCAL3.info*

```
May  2 18:01:44 kdcs3 kcrap[2766]: Authentication for username 'user1' from 193.206.84.29 failed.
May  2 18:02:14 kdcs3 kcrap[2766]: Authentication for username 'user2' from 193.206.84.29 succeeded.
May  2 18:03:06 kdcs3 kcrap[2766]: Authentication for username 'user1' from 193.206.84.29 succeeded.
May  2 18:03:19 kdcs3 kcrap[2766]: Authentication for username 'user3' from 193.206.84.29 succeeded.
May  2 18:03:21 kdcs3 kcrap[2766]: Authentication for username 'user4' from 193.206.84.29 succeeded.
May  2 18:03:26 kdcs3 kcrap[2766]: Authentication for username 'user2' from 193.206.84.29 succeeded.
```

# Configurazione kcrapclient

- Creazione in /etc/krb5.keytab di un keytab del tipo "host/<FQDN>@REALM"
  - Es.: host/radius.lnf.infn.it@LNF.INFN.IT
- /etc/krb5.conf:

```
...
[realms]
 LNF.INFN.IT = {
        ...
        ...
        kcrap = kcrapsrv1.lnf.infn.it:1999
        kcrap = kcrapsrv2.lnf.infn.it:1999
        kcrap = kcrapsrv3.lnf.infn.it:1999
 }
 ...
```

# Configurazione FreeRADIUS

- Modificare il file “modules/mschap” per eseguire l’autenticazione MSCHAPv2 tramite il comando esterno indicato in “ntlm_auth”:

```
...
mschap {
      ...
      ntlm_auth = ”/opt/bin/kcrapclient  %{%{Stripped-User-Name}:-%{%{User-Name}:-None}}
                  %{%{mschap:Challenge}:-00}  %{%{mschap:NT-Response}:-00}"
      ...
}
...
```

# Mantenibilità della soluzione

- Dipendenze
  - Modifiche/Abbandono protocollo MSCHAPv2
  - Formato del database sul KDC
- Codice di partenza
  - “Proof-Of-Concept” quality code
  - Utile rivisitazione totale per verifica della completa gestione degli errori e dei parametri in input

# Misure di sicurezza
## KCRAP Server

- Implementazione su KDC Slave (dedicato)
- FreeRADIUS e KDC Slave su sistemi separati
- Regole iptables per limitare l'accesso ai soli radius server autorizzati

# MSCHAPv2 via Kerberos @ LNF

- In uso presso i LNF da Aprile 2012
  - Prossima comunicazione all'utenza
- Piattaforme verificate:
  -  Microsoft Windows (XP, Vista, 7)
  -  Apple OSX (Snow Leopard, Lion)
  -  Linux
  -  Mobile (iPhone, Android)

# Riferimenti e Ringraziamenti

- Jonathan Chen
  - http://www.spock.org/kcrap/
- Dan Fuhry
  - http://fuhry.us/blog/2012/01/01/mschapv2-against-mit-kerberos-yes-you-can/
- Massimo Pistoni
  - http://wiki.infn.it/strutture/lnf/dr/calcolo/wireless/dot1x
- The NTLM Authentication Protocol
  - http://davenport.sourceforge.net/ntlm.html
  - Sintesi in italiano a cura di Marina Zobov
    - http://wiki.infn.it/strutture/lnf/dr/calcolo/windows/domain/protocollo_ntlm

# Domande?

Sandro.Angius@lnf.infn.it