The Radical Realm of RADIUS, 802.1x, and You

Rodney Thayer
rodney@canola-jones.com
Beetle
beetle@shmoo.com

The Shmoo Group

Intro and Overview

- Hi there. Our fearless leader, Bruce Potter, says you shouldn't believe a word he says.
  - We completely agree. :P
- This talk still hopes to impart some security clue to you regarding RADIUS, 802.1x, EAP, etc.
  - Yeah. We'll apply it to wireless somehow, too.
- We may or may not have tools to demo.
  - You can blame Sony and the recent release of the PSP for that.

Watch out...
Rodney's gonna give a
history lesson.

Brief History of User Authentication

- Usernames and passwords, in the clear
- Hard lines, no remote
- Remote adds the same thing, at a distance (threat model changes, easier to hack)
- Modems, bbs, networking, the Internet make login more complicated
- RADIUS, Certificates, Challenge/Response, many auth mechanisms arise. They always require painful changes in the wire protocols
- Key management is re-invented, over and over again (Kerberos, IPsec, SHTTP, TLS, WEP, 802.1x...)

RADIUS & 802.1x Basics
A.K.A.
Beetle Goes Googling

What is RADIUS?

- Remote Authentication Dial-In User Service
- Allows devices that could not otherwise handle it, the ability to authenticate users for access to systems / services, by offloading the authentication work to a centralized server / database.
- Allows for profile-based access limitations.
- Very common on large networks with many devices that require authentication, or with many distinct and large groups of users that need authentication.

What is 802.1x?

- First there's PPP or Point-to-Point Protocol
  - Part of Layer 2 Tunneling Protocol that provides mechanism to authenticate remote user.
- Then came EAP or Extensible Authentication Protocol...
  - Meant to extend PPP beyond just username & password pairs.
  - Tokens, certificates, spittle, etc.
- 802.1x is a standard for passing EAP over LANs, and its protocol is so named EAPOL.
  - Wired or wireless. WithOUT using PPP.

802.1X: A Layer Violation in Progress A.K.A. Rodney Gets to Rant

How did we get here?

- IEEE (not the real world) toyed with access control several years ago
- Wireless showed up
- IEEE got the crypto wrong (twice) in Wireless
- Microsoft got involved
- Standards chasers came from PPP, IPsec, IEEE world
- Lessons of the past were forgotten (AH? Why not AH?)

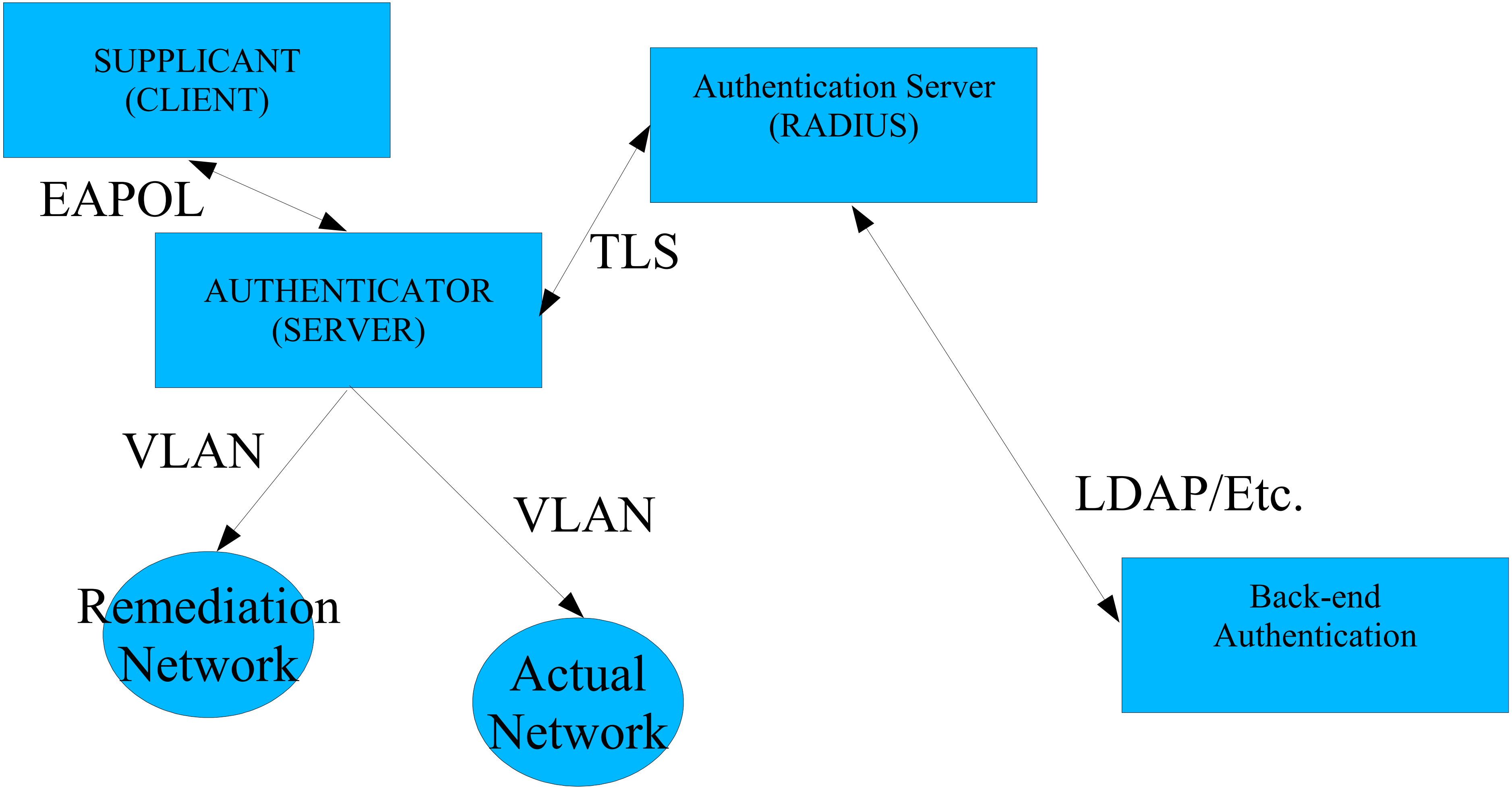
802.1X Summary

- Network access control
- Pre-IP stack
- Introduces new (layer 2.5) protocols
- Introduces authentication at link layer
- Introduces encryption at link layer
- Facilitates Policy Enforcment Points (PEP)
- Uses legacy PPP/dial-up infrastructure
- Still evolving at alarming rate

Why be concerned?

- New protocol suite
- Need to ask if the architecture is secure
- Need to ask if the implementations are secure
- Need to ask if the protocols are secure
- Need to ask if we need another protocol

802.1x Schematic



The Shmoo Group

.1X Architecture Summary

- New buzzwords for component names
- Layer 2.5 (EAPOL)
- TLS over UDP (and EAPOL), used with RADIUS
- Back-end authentication with username/password etc.
- Facilitates encryption
- Multiple EAP “methods” (protocol dialects) for authentication
- Facilitates policy delivery to end system

.1X Issues

- Uses TLS (a connection oriented protocol) with no TCP
- Uses network while no address yet (in PARALLEL with DHCP)
- Supports unattended yet allegedly authenticated access
- Uses (unauthenticated) VLAN's
- Half-implements digital certificates

RADIUS Security

- Shared secret to authenticate device (authenticator)
- TLS server certificate
- Logging
- Accounting
- Autheticable access to back-end identity infrastructure

RADIUS Insecurity?

- Datagram-only server/client protocol, no means to send back a “disconnect”
- Typically no encryption of local key material
- Poor use of certificates (no private key generation, cert naming not used, no cert status checking
- Protocol hasn't been tested for exploits recently (no fuzzer)
- Proxy nests can cause security problems due to excessive complexity

Potential 802.1x Vulnerabilities

- Poor key hygiene at the servers (shared secret for RADIUS, cert keys)
- Poor logging -> easy to hide attacks
- Poor integration with protocol stacks means old attacks work (DHCP)
- Remediation schemes make remediation servers a target
- EAPOL: Layer 2 with overly complex protocol
- RADIUS: legacy protocol, dodgy servers
- Unauthenticated VLANs

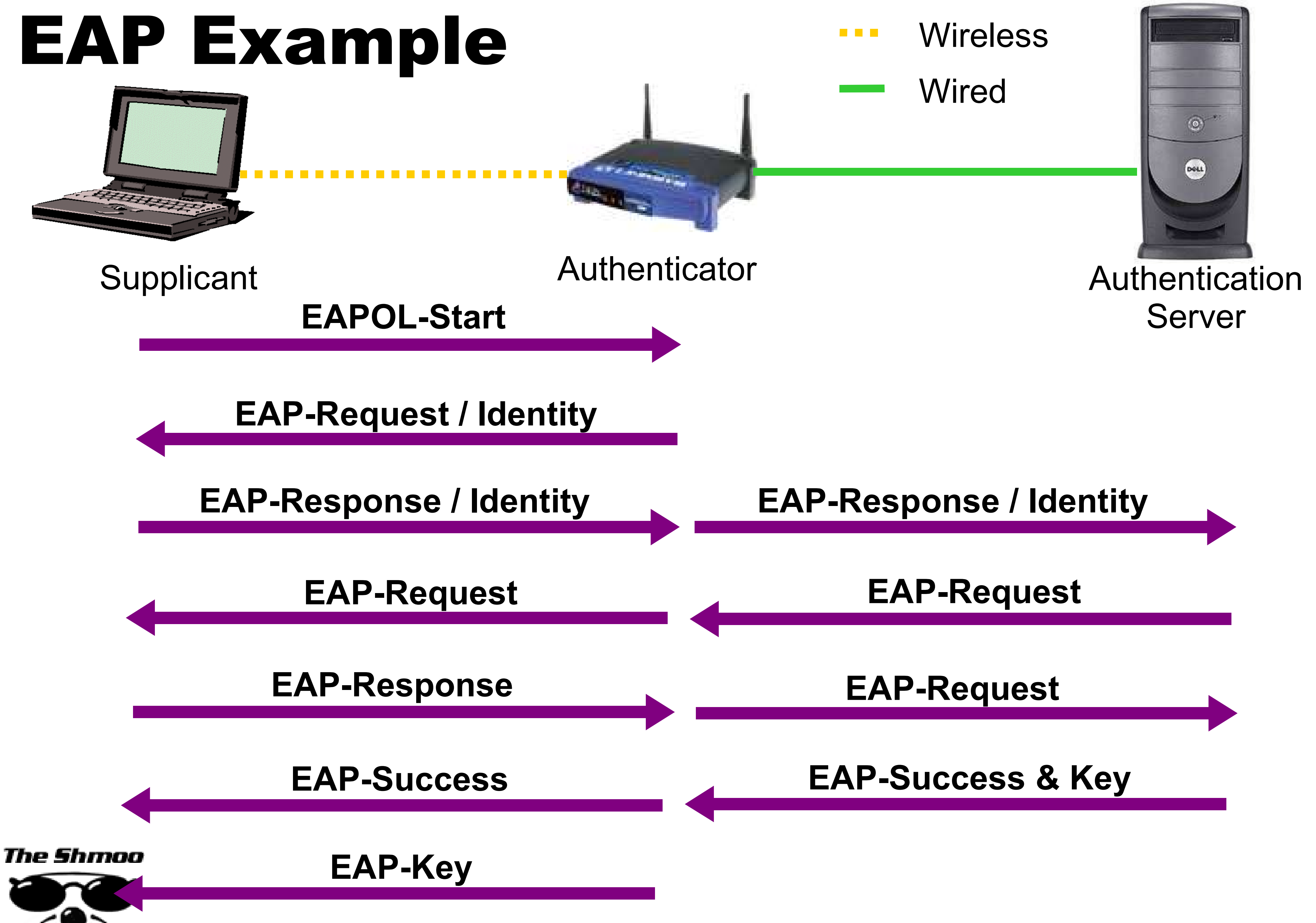
Back to EAP Basics
A.K.A.
Beetle Breaks It Down to Barney-Level

Howzat supposed to work?

- Three major components:
  - Supplicant = User / Client
  - Authentication Server = Duh. RADIUS fits here.
  - Authenticator = Device in between the two.
- Authentication goes something like this:
  - EAP-Request / Identity to Supplicant from Authenticator
  - EAP-Response / Identity to Authenticator from Supplicant which gets passed to Authentication Server
  - Challenge / Response brokered, and if successful authentication, then Authenticator allows Supplicant access to network based on what Authentication Server say is appropriate.

The Shmoo Group

EAP Example



Wireless

Wired

Supplicant

Authenticator

Authentication Server

EAPOL-Start

EAP-Request / Identity

EAP-Response / Identity

EAP-Response / Identity

EAP-Request

EAP-Request

EAP-Response

EAP-Request

EAP-Success

EAP-Success & Key

EAP-Key

The Shmoo Group

What's the Wi-Fi angle?

- Funny, but 802.1x didn't seem to hit public eye until considered for wireless.
  - You'd be surprised how many folks think it IS a wireless standard. 802.1x != 802.11 FYI
- Regardless, there's this “authentication problem” (and etc.) we have with wireless...
  - Not just authenticating users, but authenticating NETWORKS.
  - Dynamic per-session keying without pre-shared master keys would be nice, too.
- So 802.1x and EAP seem ideal for solving this...

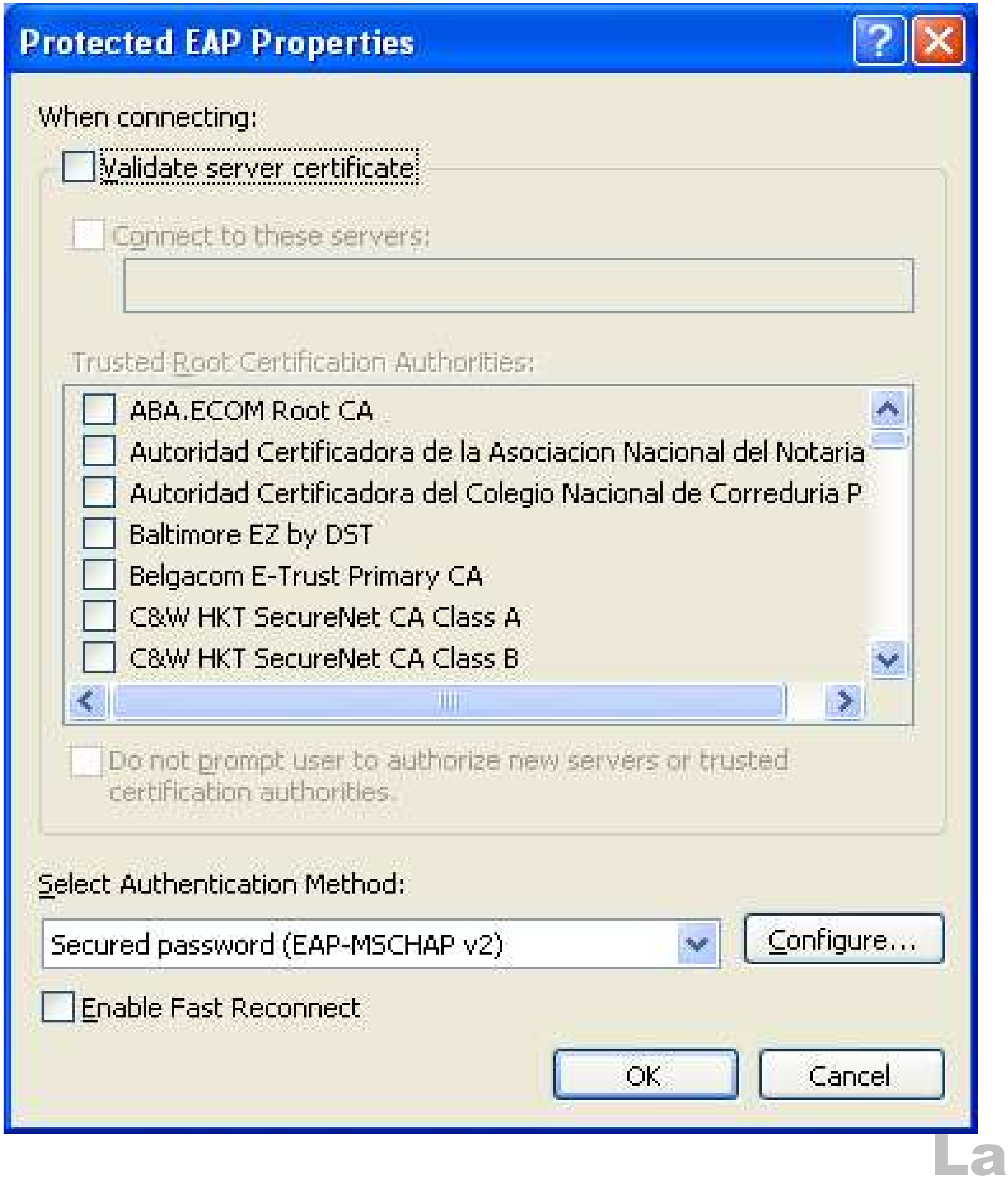
All Your EAP

- Oh crap. The EAP acronym bonanza:
  - EAP-MD5-Challenge, EAP-MSCHAPv2, EAP-GTC
  - EAP-SIM
  - EAP-TLS
  - EAP-TTLS (w/ MD5-Challenge, GTC, MSCHAPv2, PAP, CHAP, et al. variants) by Funk
  - LEAP, EAP-FAST by Cisco
  - PEAP (w/ MSCHAPv2, MD5-Challenge, GTC variants) by Microsoft et al.
- Pros vs Cons

EAP Security

- Many “methods” (protocols within protocols)
- Username, password
- Variations on whether or not the password is encrypted, hashed, or otherwise mutated
- Microsoft-embraced
- Cisco-embraced
- Standards-based
- token-based

All Your CAs...



Protected EAP Properties

When connecting:

☐ Validate server certificate

☐ Connect to these servers:

Trusted Root Certification Authorities:

- ☐ ABA.ECOM Root CA
- ☐ Autoridad Certificadora de la Asociacion Nacional del Notaria
- ☐ Autoridad Certificadora del Colegio Nacional de Correduria P
- ☐ Baltimore EZ by DST
- ☐ Belgacom E-Trust Primary CA
- ☐ C&W HKT SecureNet CA Class A
- ☐ C&W HKT SecureNet CA Class B

☐ Do not prompt user to authorize new servers or trusted certification authorities.

Select Authentication Method:

Secured password (EAP-MSCHAP v2)

☐ Enable Fast Reconnect

The Shmoo Group

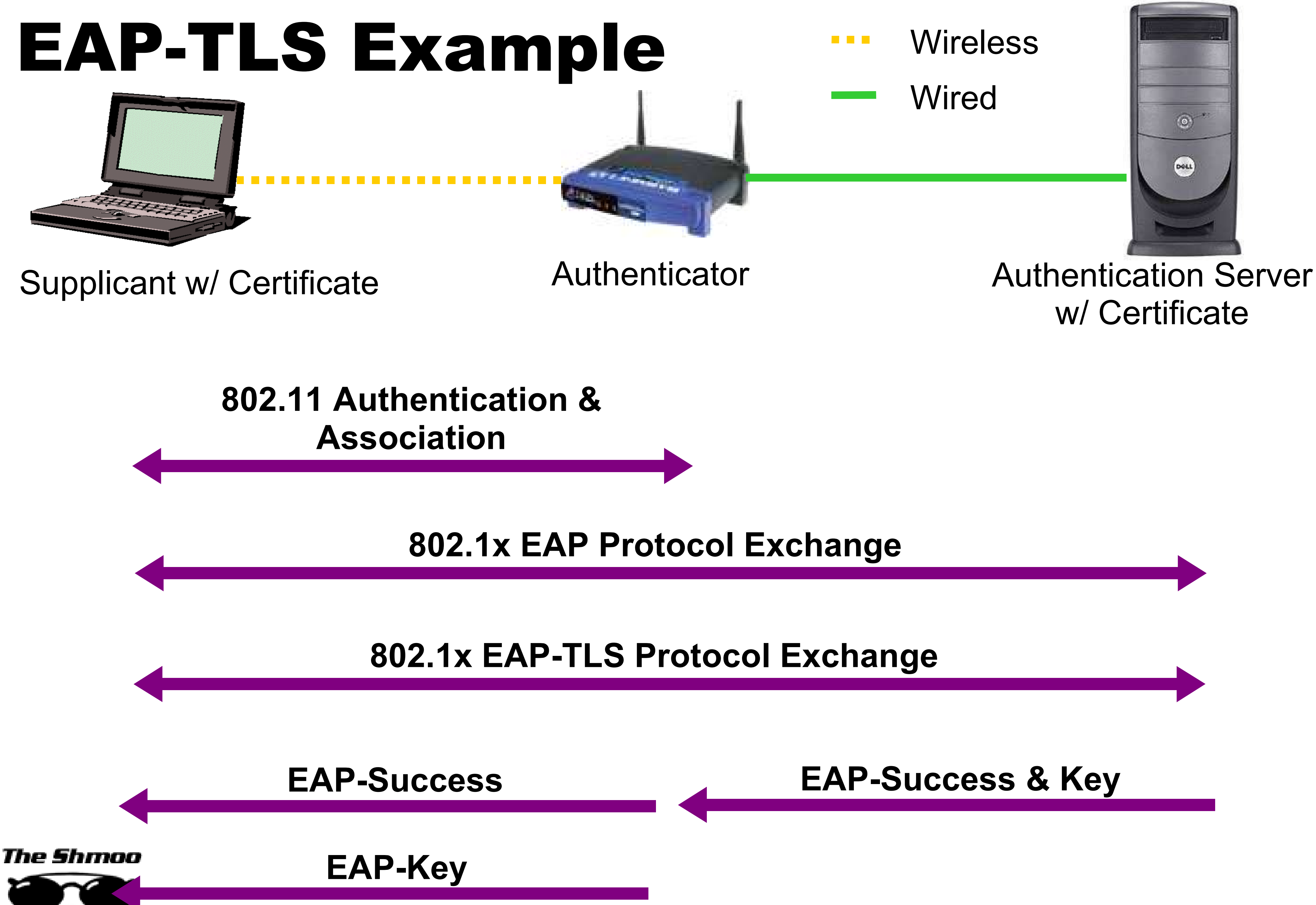
EAP Security

- EAP security really only comes in to play with its tunneled variants that use TLS.
- Two basic goals in mind with the “secure”, credential-tunneled variants of EAP:
  - Give the supplicant a way to authenticate the authentication server so they don't go spilling their guts to the wrong guy.
  - Create a secure tunnel so that the supplicant and authenticator can have a secure challenge / response exchange mechanism, which can also be used to pass dynamic keying material.

The Shmoo Group

Graphical Examples == Good
A.K.A.
Beetle's Powerpoint Fu

EAP-TLS Example



Wireless

Wired

Supplicant w/ Certificate

Authenticator

Authentication Server w/ Certificate

802.11 Authentication & Association

802.1x EAP Protocol Exchange

802.1x EAP-TLS Protocol Exchange

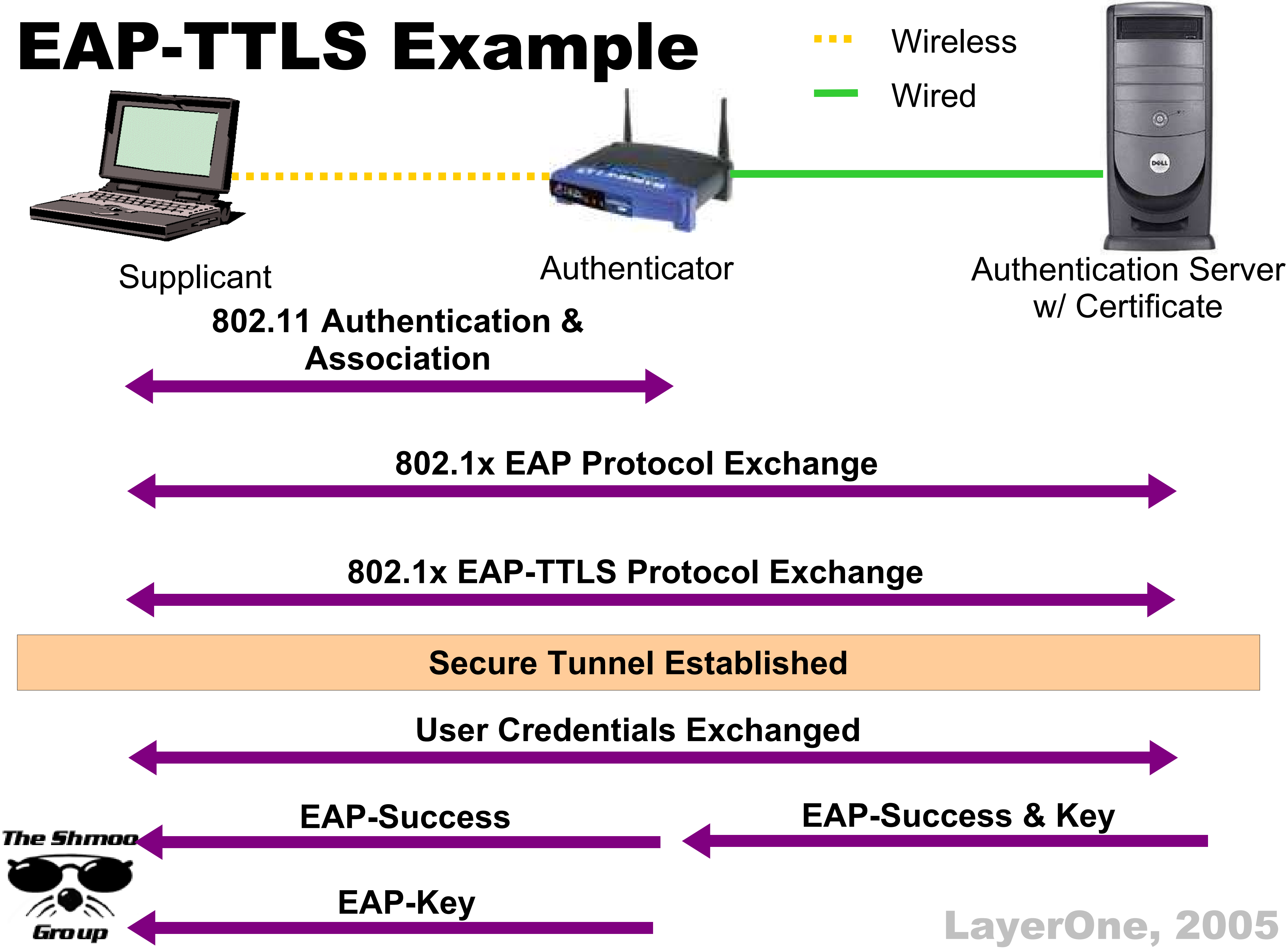
EAP-Success

EAP-Success & Key

EAP-Key

The Shmoo Group

EAP-TTLS Example



⋯ Wireless

🟩 Wired

Supplicant

Authenticator

Authentication Server
w/ Certificate

802.11 Authentication &
Association

802.1x EAP Protocol Exchange

802.1x EAP-TTLS Protocol Exchange

Secure Tunnel Established

User Credentials Exchanged

EAP-Success

EAP-Success & Key

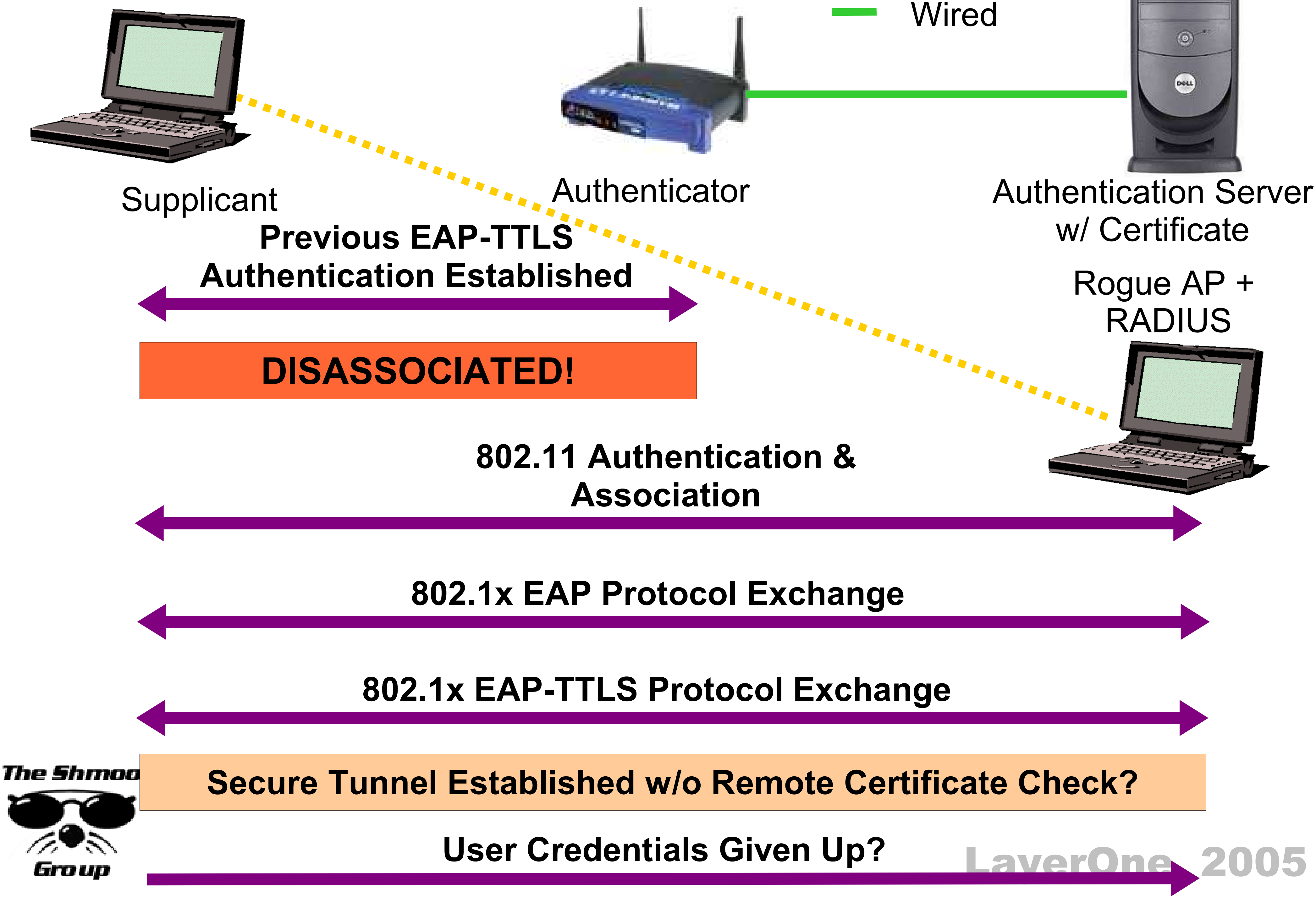
EAP-Key

LayerOne, 2005

EAP Insecurity?

- Well, naturally, there are the untunnelled EAP variants that are vulnerable to dictionary attack when the challenge / response is passively captured. Duh.
- But what about the "secure" tunnelled variants?
  - There may be valid, albeit tricky, ways to entice information from users of these "secure" wireless networks. SHOCK and HORROR, you say!
- And so... we have started working on a set of tools to attack various EAP setups.
  - "ChEAP Tricks"

EAP-TTLS Weakness



Supplicant

Authenticator

Authentication Server w/ Certificate

Rogue AP + RADIUS

Previous EAP-TTLS Authentication Established

DISASSOCIATED!

802.11 Authentication & Association

802.1x EAP Protocol Exchange

802.1x EAP-TTLS Protocol Exchange

Secure Tunnel Established w/o Remote Certificate Check?

User Credentials Given Up?

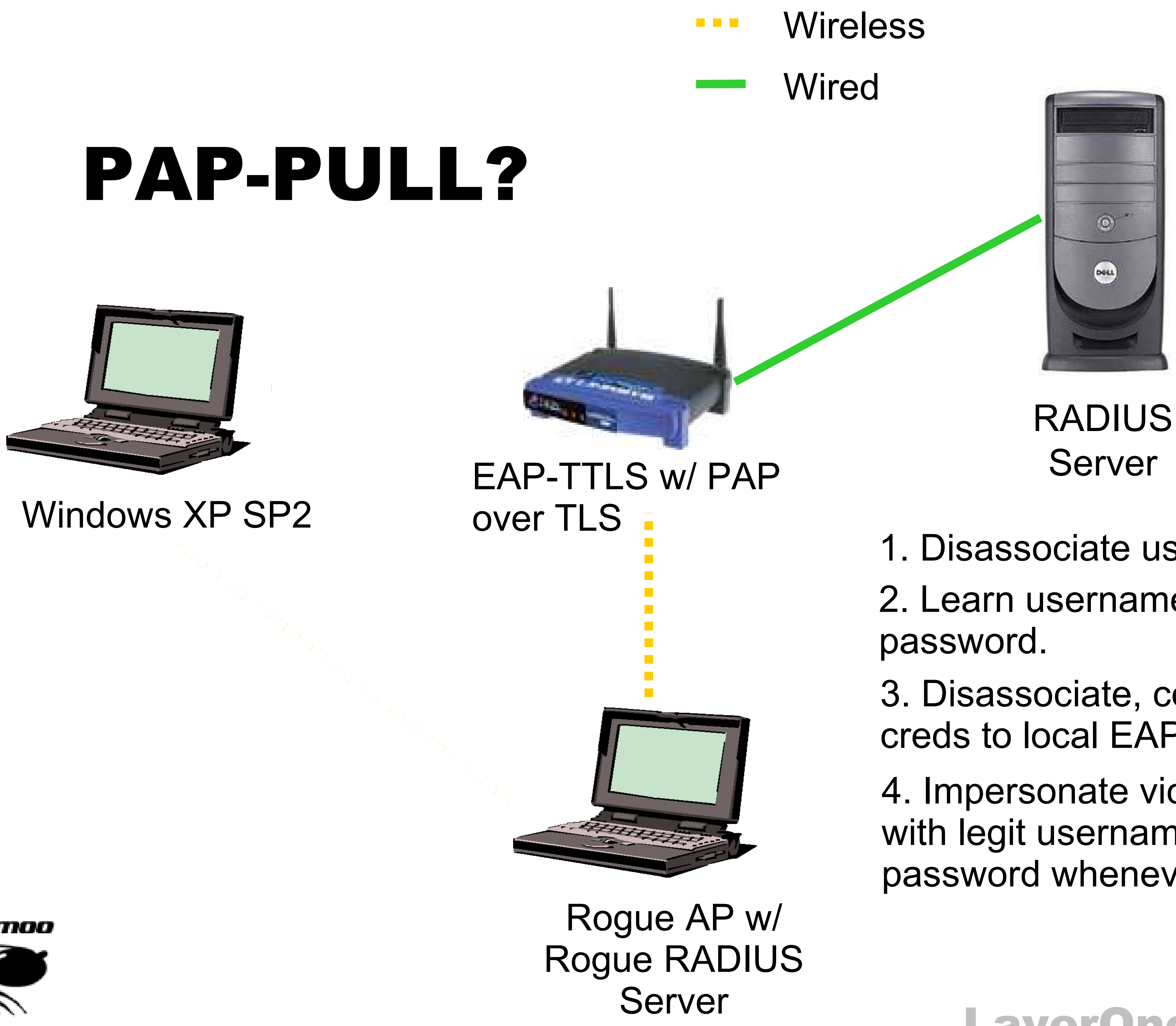
Old Attack Examples

- EAP-ACK
  - Convenient rogue AP w/ rogue RADIUS setup that accepts any EAP-MD5 client attempt for a particular SSID. Does anyone even use EAP-MD5 anymore? Hope not. This one's almost TOO easy and TOO old to bother with.
- PEAP-TRY
  - Takes a username / password combo file and just iterates through attempting to gain access to an EAP-PEAP / MSCHAPv2 authenticating network. LAME, we know.
- EAP-DUH
  - This is SOOOO ChEAP, man. Just Airsnarf asking for EAP credentials to use against an EAP protected network. Might need an enticing / similar SSID, but NOT the same SSID.

NEW Attack Examples?

- PAP-PULL
  - This is a ChEAP shot, too, but new perhaps. Mass deauth EAP-TTLS / PAP authenticated clients and gather username & password in the clear, inside TLS, as they associate to your rogue AP + rogue RADIUS. Devastating.
- PEAP-PEEK
  - Mmmmm. A new and complicated twist on a rogue AP attack that actually attempts to silently attack an EAP-PEAP / MSCHAPv2 protected network. SLOW and NOT automated right now, but potentially badass.
  - Whoa, you say. You can't DO that. Ummm...

PAP-PULL?



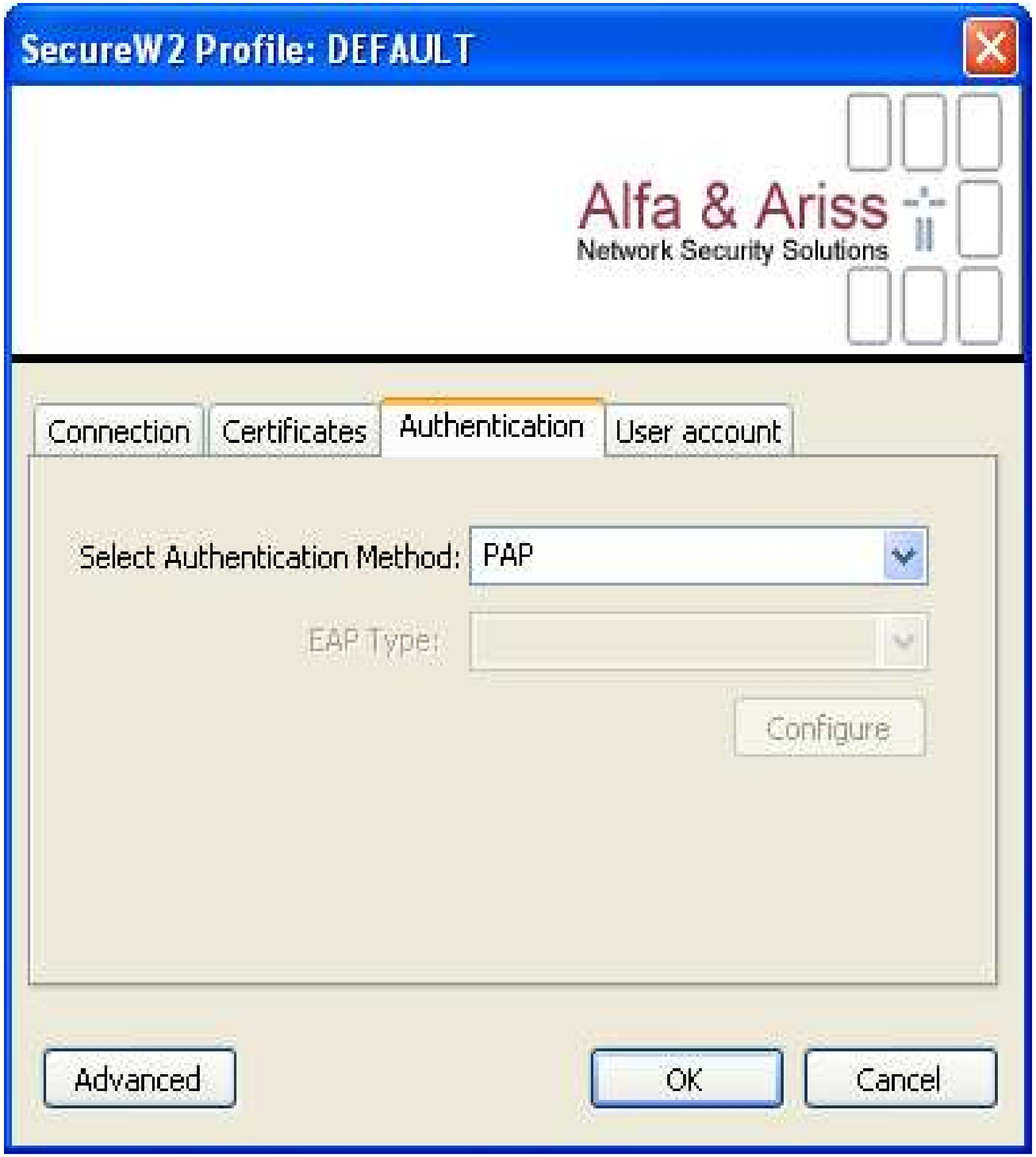
1. Disassociate users.
2. Learn username & password.
3. Disassociate, copy creds to local EAP config.
4. Impersonate victim with legit username & password whenever.

The Shmoo Group

Hey, man. NObody uses PAP w/ EAP-TTLS. Get real.

(Umm… OK.)

All Your PAP…



SecureW2 Profile: DEFAULT

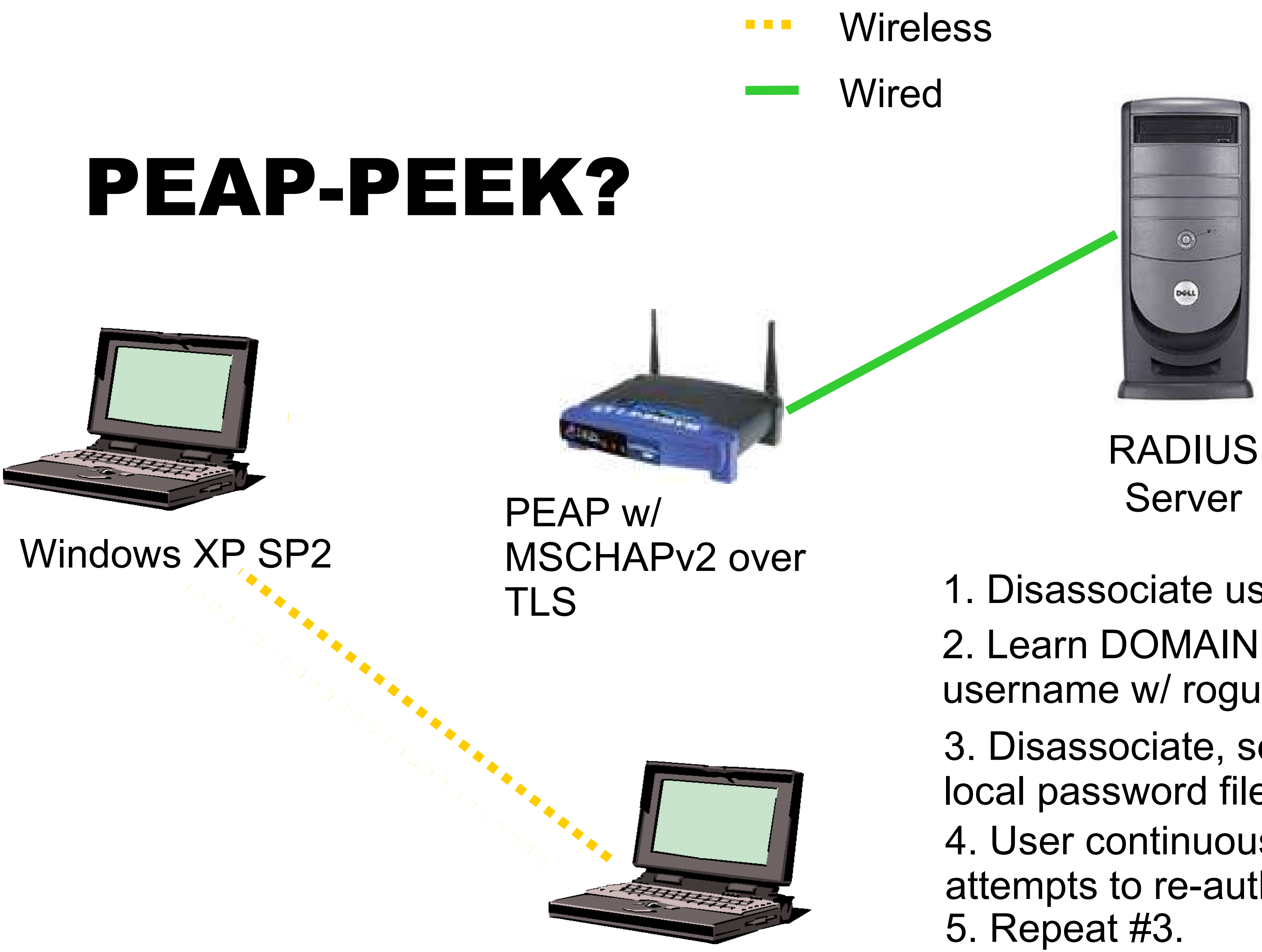
Select Authentication Method: PAP

EAP Type:

PEAP for “secure” Wi-Fi

- The P at the beginning of PEAP stands for “Protected”. Ok....
  - TLS certainly keeps folks from passively sniffing credentials. Kudos. But we're not gonna beat down TLS here.
- NOTE: Client smartcard / certificate for PEAP is OPTIONAL, since PKI is such a suck, right?
  - So PEAP allows for username / password via MSCHAPv2 over TLS and only a server side certificate.
- According to at least one Microsoft wireless security “expert” EAP-PEAP / TLS “isn't necessarily more secure than” EAP-PEAP / MSCHAPv2.
  - Oh REALLY? Assuming remote certificate checking is turned off OR common CAs are trusted, which is a common and VALID way to setup PEAP...

PEAP-PEEK?



1. Disassociate users.
2. Learn DOMAIN and username w/ rogue AP.
3. Disassociate, seed local password file.
4. User continuously attempts to re-authenticate!
5. Repeat #3.

Authentication success = correct password guessed!

Summary

- A lot of vendors throw around acronyms and architectures that will “solve” all of your authentication problems—don't believe the RADIUS, 802.1x, EAP hype.
  - Interoperability issues. Implementation flaws.
  - Use open source solutions for trial and error.
- People want a secure wireless network with minimal infrastructure that is also convenient for the users—that may be asking too much.
  - Avoiding PKI for wireless networking for the sake of simplicity has the potential to bite you in the ass. Get to work and secure your Wi-Fi.

How do we make it better?

- Analyze the protocols and implementations from a defensive view
- Get the security right, consistently
- Show how the defacto standards are insecure (automatic login, disabled cert checking)
- Knock over a few sites
- Publish a few exploits
- Get the vendor community to add security to their requirements list

Con Pimping

- ShmooCon 2006 is in the works.
  - Finalizing location and dates.
  - Will probably still be at the Wardman Park Marriott in D.C.
  - Will probably still be in Februrary.
- Pre-registration will open up as of DefCon.
  - See “ad”... <PAUSE for commercial break>
  - NOTE: We'd like YOU to submit an “ad”. It gets you in for FREE, BTW.
- And don't forget to go to ToorCon! Or DIE!

Questions?