DHCP EAP Analysis

or “where’s my pony?”
Protocol overview

• EAP Encapsulated in DHCP Protocol
• DHCP protocol starts normally
• Relay ("proxy") demands EAP authentication
• If it does, client provides credentials
• If successful, DHCP protocol resumes
Objections

- Large packets
- Non-conforming implementations
- State machine issues
- Relay vs. Proxy
- Dual stack issues
- RFC5505
Large packets

• Objection:
  – DHCP packets limited by default to 576 bytes
  – EAP may require more space
  – Some relays may not support larger packets

• Resolution
  – Fix broken relays
  – Spec references RFC3396 (encoding long options)
Non-conforming implementations

• Objection:
  – Conforming DHCP client must handle non-EAP response to DHCPDISCOVER
  – Non-conforming DHCP client may receive DHCPEAP message

• Recommendation:
  – Spec should analyze possible combinations of non-conforming implementations and recommend appropriate behaviors
State machine issues

• DHCPDISCOVER is cached
• XID not specified for DHCPEAP messages
• EAP and DHCP state machines opposed
• EAP lifetime and DHCP lease not synced
• EAP renewal not specified
• EAP message retransmission not specified
Relay vs. Proxy

• Spec talks about a DHCP proxy
• DHCP proxy functionality is never clearly delineated—it sounds like a relay agent
• Proxy seems to maintain EAP state
• BBF requires proxy in order to choose DHCP server based on results of authentication
Dual Stack issues

• A dual stack client may authenticate twice, and authentication information may be out of sync.
Decoupling state machines

• DHCPDISCOVER triggers EAP
• No EAP in client state machine
• Client receptive to EAP at all times
• Separate EAP processing from DHCP protocol
  – DHCP Client hands EAP messages to EAP client
  – Relay hands EAP responses to EAP server
• This fixes previous three slides
RFC5505

• Requires us to separate authentication and configuration

• But we really have – DHCP is not authenticating—it’s relaying.

• Also, DHCP+EAP lets us get rid of PPPoe
Observations

- Specification is very weak.
  - Many edge cases not specified or even mentioned
  - Actual intent of spec unclear
  - No chance at all that someone reading spec could do interoperating implementation

- Specification could be fixed
  - There are solutions to all the problems in the spec
What next?

• No consensus in DHCwg
• Vocal objection from outside of group
• Spec as written is harmful
• Spec could be improved
• Very contentious issue; debating whether to do this has wasted a ton of DHCwg time
• We’re going to try to pass the buck—brace yourself.