ACP status
IETF 104 Prague 2019
draft-ietf-anima-autonomic-control-plane-19

Toerless Eckert tte+ietf@cs.fau.de (Huawei USA)
Michael Behringer michael.h.behringer@gmail.com
Steinthor Bjarnason sbjarnason@arbor.net

v1.0
Status

• IETF 103:
  • draft-ietf-anima-autonomic-control-plane-18
• IETF 104:
  • draft-ietf-anima-autonomic-control-plane-19
  • Changes for IESG Security review
    • Eric Rescorla
    • Benjamin Kaduk
  • Think all outstanding IESG review comments addressed
    • Need to check aain Gen-ART/Alissa, but pretty sure all was addressed by -18, except for one formatting issue Alissa brought up again @IETF103, which is fixed in -19 too.
• No functional changes to ACP.
  • But refinements of details of mandatory IPsec/dTLS profile (to ensure interoperability)
Changes

• Many textual, sentencing improvements from feedback (thanks a lot)

• Added: “Support for constrained devices is opportunistic...”
  • Aka: We include aspect in support of constrained devices (dTLS) that helps us to justify and explain the ACP secure channel discovery/selection mechanism, BUT we are NOT complete.
  • Complete for example would mean to support profiles completely without any TCP, so we would need to do something like EST over UDP/CoAP and likewise GRASP/ACP over such non-TCP transport. We think due to size of document and desire to be standards track based on existing experience, such complete constrained device support is better for followup work.
Changes

• Explain how ACP
  • Can help secure bootstrap (automatic connectivity)
  • “security” – hides communication patterns between registrar/pledges due to hop-by-hop encryption (especially any pre-existing non-BRSKI mechanisms).

• Section 4 – “Requirements” (informational)
  • Always difficult to explain how this section goes back to charter not allowing us to write a separate requirements document...
  • Changed all “_MUST_” / “_SHOULD_”, to just must/should to avoid any confusion with actual normative requirements in normative section.

• Challenged on suggesting rfc822 field for ACP domain information, added explanation how this choice avoids to expect additional ASN.1 decoding capabilities too.
Changes

• ACP domain membership check
  • Actually doing “certificate path validation” (if there are sub-CA).
  • Refined details on what happens if there is no way to access CRL or OCSP-server and closing connection to peers for whom its learned later that certificate is revoked.
  • Explained how ACP domain membership check formally includes peer authentication and authorization (to access ACP or other domain services) via the different steps of the domain membership check.

• BRSKI details mentioned can really only be understood when understanding BRSKI
  • But BRSKI not a dependency, just an option/example for bootstrap
  • Added text pointing this out (skip BRSKI section when not using BRSKI).
Changes

• 6.1.3 added ca. 1 page explanation of Trust Points and Trust Anchors
  • Multiple disjoint cert-paths in different ACP members possible (when using different sub-CA)
  • Only (trusted) registrars of ACP domain permitted to help enroll ACP certs.
  • Operate multiple ACP domain with private trust anchors

• MacSec: Added that this is mentioned only as example of likely interesting next secure channel protocol.

• Figure 7: Added step-by-step explanation figure for (colliding) secure channel setup and how Alice and Bob determine who they are (existing text does the same, just quite terse).
Changes

• IPsec profile details added:
  • ESP with AES-256-GCM (RFC4106)
    • Seems to be well supported by HW IPsec now (even easier on some platforms ?)
  • Key establishment MUST support ECDHE with P-256.
  • Existing: SHA256 hash and not permit weaker crypto.

• dTLS profile details:
  • Rely on RFC7525 (pointer from Eric), except:
    • Only use DTLS 1.2 or later

• Added paragraph about absence of MTI secure-channel protocol:
  • Aka: IoT nodes may only do dTLS, backbone nodes only Ipsec, so only gateway nodes connecting IoT and backbone areas need to support both dTLS and IPsec.
Changes

• Addressing:
  • Text: ..choose subdomain names so that no ULA hash collisions result.
  • Paragraph explaining example how this can also be done across disjoint but administratively coordinated ACPs.

• ACP registrars text extended
  • Uncoordinated == multiple registrars can assign ACP addresses independent of each other because of addressing scheme of ACP
  • ACP registrars are PKI RA (registration authorities) with added functionality for ACP domain certificate field
Changes

• RPL
  • Fixed a lot of text/resorted paragraphs to hopefully make “Overview” section a lot easier to read – and easier to justify why we choose this profile (no routing header).
  • Also added paragraph highlighting benefit over other IGP (fewer routes, lot better scale towards the edge for low-end devices).
  • Explained why RPL security not used (running securely inside ACP)

• L2 ACP: Refined sentences about interaction of ACP and STP (Spanning Tree protocol).
Changes

• ACP benefits (informative)
  • Self-healing. Added discussion about ACP ejecting revoked/expired peers.
  • Re-emphasized how ACP domain name collision is rare (ULA) and NOT a security aspect (but instead an operational aspect), because it would only occur between ACPs with shared trust (aka: common trust anchors).

• Operations
  • Enumerated references for long list of example “operational” protocols that could run inside of ACP: SNMP ([RFC3411]), NTP ([RFC5905]), PTP ([IEEE-1588-2008]), DNS ([RFC1886]), DHCPv6 ([RFC3315]), syslog ([RFC3164]), Radius ([RFC2865]), Diameter ([RFC6733]), TACACS ([RFC1492]), IPFIX ([RFC7011]), Netflow ([RFC3954])
Changes

• 10.4 Configuration and ACP
  • reviewer ask/confusion about how much config is required for ACP
  • Only No-configuration is good configuration for ACP ;-)
  • Exceptions:
    • Bootstrap config (can be simple with BRSKI)
    • CA/Certificate renewal ”server” (EST server) config
    • ACP-connect
    • Brownfield: Explicitly enable ACP, extend ACP across non-ACP nodes
Changes

• Security considerations
  • Reworded initial paragraph highlighting initial steps to get running ACP, and from then on is not depending on configuration anymore (exception as mentioned above the non-ACP components).
  • ACP registrars are critical infrastructure, need to be hardened similar to a CA.
  • Added several paragraphs detailing peer-to-peer security group model goal/benefits of ACP.
    • Aka: we can only do peer-to-peer because we want to allow ACP to form if just two ACP members connect – without any dependency against a third-party.
  • Discuss use of ACP domain certificate for higher layer functions (e.g.: end-to-end) and discuss limits of unstructured peer-to-peer model (suggesting introduction of role differentiation as described in A.10.5)
  • Long lived ACP channels == need to check cert expiry during channel lifetimes.
Changes

- IANA considerations
  - Hopefully made explanation for why we choose SRV.<xxx> (xxx = EST...)

- Appendix A.10.8 (new)
  - Maybe contentious?
  - How to deal with compromised ACP nodes.
  - IMHO, this is not something primarily to be solved by ever more certificate management details, but by looking at the key attack vectors:
    - Application layer credential leakage (aka: passwords leaking to attackers).
    - Want to harden routers to not permit any local config of credentials (so no backdoors can happen). Also track any configuration changes on routers.
    - ACP itself hard to break because not configurable.
    - Easy then to change leaked passwords, because attacker can not prevent this to happen if automated via ACP. And kick out any established hacker sessions.
Thank You!