JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

Bundle Protocol Version 7 Administrative Record Types Registry

IETF 119 DTN WG

Brian Sipos
JHU/APL

Current Status

- Last draft of the document draft-ietf-dtn-bpv7-admin-iana-02 still in WG Last Call state
- The draft has expired in February 2024
- No comments have been received or edits are expected to be needed for this draft revision
- This document is required by draft-ietf-acme-dtnnodeid-12 to progress

Next Steps

- This would eventually be in a cluster with the ACME document registering the new code point
- The BIBE document would also eventually need code points

JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

BPSec COSE Context

IETF 119 DTN WG

Brian Sipos
JHU/APL

Background

- BPSec and its Default Security Context are usable but intentionally limited in scope:
  - A limited number of symmetric-keyed encryption and MAC algorithms
  - Defines a narrow-scoped additional authenticated data (AAD) binding to the block/bundle
  - No explicit key identifiers are available
- For internet-facing nodes, possibly as subnetwork gateways, there is a need for PKI-integrated security
  - This was indicated by IETF SECDIR review of BPSec draft and also discussed as a near-future need by NASA and IOAG DTN planning
- Don’t want to reinvent the wheel, and CBOR Object Signing and Encryption (COSE) already provides syntax and semantics for current and future PKI security
  - Even COSE (with a restricted profile as used here) still provides a lot of variability, in the same sense that TLS or S/MIME does, which must be managed out-of-band (e.g. don’t use ECC algorithms if security acceptors can’t support it)

Last Changes

- Changed AAD Scope parameter to allow use of indirect block references:
  - Uses -1 to refer to each target block identified in the ASB
  - Uses -2 to refer to the security block containing the ASB
  - These are necessary for BIB and BCB with multiple targets to have the same policy of “bind this security to the target metadata”
  - Simplifies the default AAD Scope value to {0:0b1,-1:0b1,-2:0b1}
- Added mention of existing “kid context” header parameter for identifying keys
  - Does not affect context or COSE requirements, just mentioning an existing mechanism

Open Issues

- There are currently open issues on the GitHub project
  - #23 No minimum interoperability for x5t algorithms
  - #24 No allowance for single-layer encryption with direct CEK
  - #25 No recommended algorithm for non-wrapped ECDH algorithms
  - #26 Recommend against using PartyU/PartyV
- For #24, #25, #26 these just represent recommendations in Section 3.2
- For #23 this represents a requirement for minimum support in Section 3.3
- None of these issues change the BPSec context definition or any COSE processing, just minimum interoperability

Implementation Experience

- Ongoing APL IRAD to explore secure messaging with BPSec/BPv7 in a PKIX environment
- Forked open source HDTN and COSE-C projects, added COSE Context and static policy
  - Created several new tickets in each upstream project for issues uncovered
- Forked open source dtn-demo-agent and pycose projects, modified COSE Context and policy
- So far no additional changes to the COSE Context definition have been identified

Next Steps

- This is not intended to replace or supersede existing BPSec interoperability contexts in RFC 9173
- The point of this security context is to allow BPSec in a PKIX environment in the very near term
- This document doesn’t address what kinds of policy are required in a BPA/BPSec implementation
  - There is ongoing work funded by NASA AMMOS which addresses BPSec policy design and implementation
- Document has passed WG Last Call with no additional comments

JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

Bundle Protocol Endpoint ID Patterns

IETF 119 DTN WG

Brian Sipos, Emery Annis
JHU/APL

Background

- Use cases on the following slide motivate the need for a mechanism to define a set of EIDs in a structured way
  - Goal is to ensure the writer and the user have the *same interpretation*
- Simple globs or regular expressions could be used, but these are not ideal
  - Purely text-based
  - Do not take advantage of the structure for DTN or IPN schemes
  - Do not handle numeric intervals for IPN scheme
  - Do not have an efficient binary encoding
- Pattern matching syntax has a “network effect”
  - The more tools that use a common syntax the more value it has
  - If established, new tools do not need to reinvent a robust mechanism
  - Lessens the possibility of security vulnerabilities from misconfiguration “is this parameter an EID or some glob expression?”
- This proposal is compatible with IPN Scheme update draft-ietf-dtn-ipn-update

Use Cases

- Security identities
  - Allow a certificate holder to be authorized to sign for `dtn://node/**` or for `ipn:3.*.*` or even `ipn:3.*.0`
  - The same way as wildcard certificates, it is a CA obligation to ensure endpoint ownership of all matching EIDs
- Routed blocks and authorization
  - EID Patterns are meant for a more structured situation than “huge list of EIDs”
  - The same purpose as IP CIDR notation *e.g.* `192.168.30.0/24`
- BP Agent configuration / policy
  - Allow BPA configuration to use consistent pattern syntax
  - Allow node `ipn:3.5.0` to sign bundles from `ipn:3.*.*`
  - Provide the same kind of ubiquity as CIDR does for IP configuration
  - Avoids policy engines with over-restrictive or limited expressive syntax
- Colloquial use
  - Have an understandable way to convey technical comments like:
    I’m having trouble sending to ipn:3..**
    Please allocate your services within ipn:.*.[5-10]*

Proposed Capabilities

- Draft in draft-sipos-dtn-eid-pattern-01 with pending issues in https://github.com/BrianSipos/dtn-eid-pattern/issues
- Any-scheme pattern: `***`
- IPN Scheme Patterns
  - Allow a match-all syntax `ipn:**`
  - Separate the EID into single-integer parts, each part can be one of:
    - Exact-match value (compared as integer)
    - Match-all one-part wildcard
    - Range expression (set of discrete intervals)
  - Compressed CBOR encoding using integers
  - Simple set logic (“Pattern A contains B” or “Pattern A overlaps with B”)
- DTN Scheme Patterns
  - Allow a match-all syntax `dtn:**`
  - Separate the EID into node-name and service-path segment
  - Each part can be one of:
    - Exact-match literal
    - Match-all one-part wildcard
    - Match-any-parts wildcard
    - Regular expression, percent-encoded
  - Complex or unavailable set logic (related to regular expressions)

Examples of EID Patterns

- Singleton pattern:
  `dtn://node-name/serv ipn:3.10.5`
- All services on a node
  `dtn://node-name/** ipn:3.10.*`
- One service on any node
  `dtn://**/serv/name ipn:*.*.5`
- Complex wildcard patterns
  `dtn://**/prefix/* ipn:3.*.5 ipn:3.*.*`
- Expressions and ranges
  `dtn://[prefix.*]/serv ipn:3.[5-10,100-110].5`
- Mixed patterns
  `dtn://[node%5BA-Z%5D]/** ipn:3.[10,12,14].*`
- Multiple combined patterns with pipe separator
  `ipn:3.[10,12,14].*|ipn:[4-5].*.*`
- Match-all pattern:
  `***`

Considerations

- An EID Pattern *is not* an EID, they cannot be used interchangeably
  - This is a security risk *a la* the wildcard DNS names in early PKIX certificates
  - The syntax has been designed that a range (IPN) or expression (DTN) is specifically *not* a valid EID value per the ABNF syntax
- An EID Pattern is a superset of EIDs
  - It is a design goal that an EID *is* a singleton-matching pattern for itself
- Patterns are conceptually simple but can be complex in practice
  - A common specification can allow shared-use implementations
- IPN pattern special considerations
  - IPN scheme now has three logical parts, IPN patterns always have exactly three components

Next Steps

- Feedback on current proposals
  - What is valuable immediately?
  - What should be deferred?
  - Any issues with the current syntax or special cases to be avoided?
- Trial or example implementations
  - Existing BPAs that want to try out this syntax?
  - Potential hackathon topic?