Contents

1 WG status of the various documents

- recap of various states

2 BGPSEC protocol

2.1 BGPSEC-10 (Matthew Lepinski (ML))

- origin validation is decoupled from BGPSEC validation
  - BGPSEC and RPKI results are independent
  - Wes George (WG): has anyone given a thought to the race condition when one is valid and the other isn't? this strikes me as something that is going to bite us if we don't think about how this is going to work.
    * Matt L (ML): I think you're right that it might be wise to include an example policy or something that said "this is one example policy of how to deal with these different results".

- Added reference to AS-migration
  - should be complimentary and not in conflict now

2.1.1 Open issues

- Only outstanding issues are editorial
- Is the text describing how the $\text{BGPsec}_{\text{Path}}$ attribute is used in place of AS_{Path} sufficient and clear?

2.1.2 Next steps

- discuss with IDR
- push -11 with editorial changes
- Sriram, Kotikalapudi NIST (SR):
  - Went through document
  - Have some editorial comments
    - Matt: can you give them to me quickly so I can spin the document with them?
  - With section 5, found a couple of technical errors
    - BGPSEC update is valid and invalid
      - Need to say "origin validation or path validation is valid or invalid", because some sentences still indicate both
      - Negotiating EBGP peers is establishing a relationship. When you establish a new connection during algorithm
      - ML: you don't agree on algorithms in the capabilities exchange; I just send all algorithm sigs and you ignore what you don't understand
      - SR: ok, but on page 24/25ish then if I receive alg 2 when I only understand alg 1 then i should treat this as an unsigned update. If i'm processing the sig block and I don't find alg 1 and I find a sig block with alg 2, is this an attack point and I should treat it as a protocol error or should I treat it as unsigned. i think we should carefully thing that.

- ML: I'd be happy to entertain comments on the mic or on the list about it. I think we hope the way the transition will work is that we continue to do both until all our peers plenty of time to support both algorithms.
- SR: when I don't recognize the algorithm number, i think i should treat that as an error
- Rob Austine (RA): I believe we discussed this. It's a slow algorithm transition. We agreed a long time ago that not understanding the algorithm is equivelent to an unsigned update.
- ML: lets discuss that more on the list

3 Considerations on RPKI overclaiming (John Curran (JC))

- JC gives presentation

3.1 Smaller subordinate certificates are sometimes needed

3.1.1 weird things happen when parent and children CAs disagree

- overlaps happen when a child is using a larger CA
- sometimes validation states end up in bad statesup, including "invalid"
- Ruediger Volk (RV): make before break is required if we want to use this operationally. We better design our stuff so we can fully rely on it. We need to be careful until we have perfect implementations. It's not a good idea to consider something "not completely reliable".
- RV: We do have any procedures for describing what a proper transfer procedure is?
  - JC: I think the RIRs need to contribute this. It's not a RIR only topic though. Moving from one ISP to another requires coordination between your old ISP and my new one.
  - RV: how can I, as an ISP, do the right thing when I don't understand what my parent does.

- WG: there is a definite need for consistency here. Even just RIR to RIR transfer. How prescriptive do we need to be? Some is just operational. We need to be prescriptive because if we do it wrong we'll break stuff.

- Sandy Murphy (no hat; SMNH): One thing has always confused me about actions that are planned and ones known about ahead of time vs ones that happen without any planning. The second accidental case includes power outages, crypotgraphic failures, etc. Do we need a solution that applies to all of them or just some of them?

    * JC: the transfer is the foreseen case; those occur when people move organizations but also happen when people move between regions. We can mitigate for those if we document some of them. There are some instructions from a court that says to do something you don't have a choice about what to do. These unforeseen resource changes are very hard to mitigate against.
    * SMNH: I admit there are exmaples of those (I'm not sure the court case is one). The question remains: do we need a solution that covers all the examples? Is there one more than other that needs to be addressed.
    * JC: I'm not sure we need to change, we just all need to understand the ramifications.
    * ?? to SMNH: There is a set of events that can make this happen
    * SMNH: what I was asking for was a description of cases where it is possible, because I don't believe it is. I don't see that it's possible for an RIR to have an overclaiming certificate, but I don't see RIRs able to do that.
    * JC: only if you have a global trust anchor
    * SMNH: I still don't see it.
    * JC: I'm more concerned about ISPs having overclaiming certs
    * SMNH: I don't see when RIRs overclaim and when it can happen?
    * JC: right now it can't happen

- Tim Brujnzee (TB) RIPE NCC: there is a lot of benefit of looking at foreseen cases. EG, transfers, reclaims, etc. I think there is stuff to do that we can make stuff better. There are also always a possibility that things break. You're right we need to review. Another problem can be that if you allow for this on one hand you increase the resiliance against mistakes but you also increase the risks that holes can be punched from the top with surgical precision (the validation reconsidered mechanism). Which is the bigger problem, accidental failures vs hole-punches from the top?

- RA: the design of the provisioning protocol tried to deal with this by trying not to do revokes when possible, but when you have a shrink you don't have a choice. Picture Alice->Bob->Carol tree; If Bob didn't get a notification that carol's resources got yanked, then Bob has to re-issue everything once it sees things happen. If Carol sees this later, then there are certificates that are invalid because they don't match the current resource allocations. If someone has a failure in the communication chain, then are failure modes in the big distributed database that contains errors.

- SMNH: Alice will reduce bob's cert; carol has cert from bob and has a mix of retained and removed. Bob will eventually give carol new certificates for those retained, but alice can actually issue that certificate itself. In the RIPE database it says you're responsible for the entire address database below you even if you have delegated some of it. I always thought that's the way this world thought; they had the responsbility and the authority.

- RA: I don't know where alice can get carol's [public] key from. I don't think that is possible.

- SMNH: Alice would need to get carol's key to fix this

- JC: there may need to be a mitigation step. I just don't think we've exlpored them and documeentd them.

- TB: We need a signaling mechanism. In the case of a foreseen shrink there is a lot we can do there.

4 RPKI Retrieval Delta Protocol Tim Brujnzee

4.1 Rsync and new replacement protocol discussions

- WG: why can't we just start testing this?
  - tim: we have existing deployment and we can't just change from one thing to the next; I do have confidence this will work though.
  - RA: This is a change and I'm quite sure my validation code won't work with the new OIDs, so we can't just drop it in without talking to people first
  - WG: Just to clarify, I wasn't saying flip the switch without telling anyone, I'm just saying lets roll it out soon.
- RA: this is similar to zone transfers with AXFR and IXFR. This is a new application of an old technology.
- ML: I like this approach. Where is this documented?
  - TB: it's outdated; I haven't asked for a WG document yet.
  - ML: just want it in the minutes
  - [editor: It is!]
  - TB: it is outdated, so I'll try to update it within the next 2 weeks
- Andy Newton (AN): can we adopt it now?
- AN: how do you make sure the file is complete before serving it
  - TB: have you operated a CDN? You could run your own. The cheap solution would be to write it to a disk
  - RA: there are standard unix tricks for this
- AN: how do we know when we can delete the deltas?
  - Tim; that's a good question. We need to have a discussion about that
  - RA: Handle it the same way DNS does; keep stuff around till you're tired of maintaining it.

  - WG: it's less work to pull the whole file sometimes; it's probably better to start from scratch if you need to pull more than N deltas
  - Tim; we can actually keep some stats to work on this as well. we may be able to determine how long to keep them

- Terry M (TM): has there been any security review of the file itself?
  - TB: no review, but we have object security so it is no different than rsync
  - RA: we've thought about this, are there any benefits to channel security?
  - TB: and how do you achieve it (HTTPS), but then that becomes a point of trust.
  - TM: perhaps consider just to stop the man in the middle attack. Use it to stop the man in the middle preventing an update.
  - Jeff: The notification file itself needs an integrety check on top of it. I think the caching mechanisms are needed, and the object security mechanisms as well.
  - RA: I'm not sure that https brings anything. It doesn't stop anything. Part of what we were trying to do is going light weight. I'm not convienced there is a case for https yet.
  - Tim: maybe serving over https would be heplful; we'll have to check.
  - AN: I think I agree with Terry about https. Ghost buster files have PII, and those might need to be encrypted with https.
  - Tim; but it's a public database
  - AN: you just have to call it out as a privacy issue
  - ML: if anyone is putting something in a distributed repository that they're uncomfortable with the entire world seeing, then we have a problem.
  - SM with hats (SM): The WG discussed this in the past, this is someone putting stuff in the database for publication.

- AN: there are other groups that have issues with this, and we will too because it's a vcard.
- TB: https doesn't solve this problem; you can still get the data.
- Ellie: we should do what's right for the security pieces, yes privacy is concerned. Don't worry about the IESG; we'll talk. Do the right thing for a public database.
- Chris (no hats); CNH: there is a lot of direction for focus on caching.
- Wes Hardaker (WH): caching should just be stated as possible, but point to the transport documents about how to do it (eg, http)
- RA: we're looking for a way to steal a mechanism to help with both redundancy and caching and many exist. We wanted a system to allow for current off the shelf tools to be used.
- TB: want to minimize the load on the server side and put the load on the clients.

5 Proposal for signaling consent with whacked RPKI objects

5.1 main points

- People that hold ROAs that are going to be whacked must approve the whacking.

5.2 APNIC does publish manifests

- George: Slide said that APNIC doesn't publish manifests, this is incorrect because we do publish it but we do make the statement that there are operational aspects of it. The MANIFEST may not contain files because they shouldn't be included even if they're still on disk because of the publication timeline. It's an exclusion check, not a catalog. We made the public statement that a manifest, during publication timeframe, may not perfectly match the files.

5.3 discussion

- Terry: a court order won't talk to the defendant to tell them to sign a .dead file
- WH: what happens with a signer which truly is dead (company or key loss) and can't sign the .dead?
  - the alarms would go off; whether something happens automatically in determining if a route is accepted automatically or requires human intervention is not known at this time
- ML: this is a problem to be solved and
- ??: is the .dead a mandatory or optional feature? This allows people to be able to make local choices.
- Doug Montgomery (DM): is there a garbage collection mechanism
  - yes; it's later in the slides
- TB: sadly, sometimes, consent is sometimes optional. Relying parties will have a lot of burden to make decisions about each case as they come forward.
- DM: I'm worried about the tons-of-alarms problems. Normal business operations will likely produce thousands of alarms, how would we deal with these? We've talked to a lot of people that love the idea of the RPKI because they can invalidate the people underneath them and get the addresses back.
  - The key must be held by the provider in that case so they can invalidate the child without the child's consent.
- Jeff: It is useful to be able to tell if something happened that you didn't see. Most users will only care about the most current state.
- Jeff: Any number of problems in the real world need to be accounted for in the proposal, such as system crashes, key losses, etc.
- RV: Can we figure out who did the revocation?
  - partial answer: it's coming later in the slides

- DM: This seems tailored at rare events. RPs may make different decisions. I worry about global synchronicity about looking at the RPKI, which is currently loose. Different people will have very different time-notions of the state of the world.
- ??: This is base on the premise that this problem needs to be solved for deployment of the RPKI; I think this decreases the deployability of the RPKI because it adds complexity.
  - I think it's important to signal when something is suspicious and I think that increases the trust in the system
- DM: the discussion of whacking always interests me. There are forced revocation of resources that most people would agree is necessary.
  - You would need to prove the internet community that what you're doing is right
  - DM: how?
  - Out of band.
  - DM: don't some orders come sealed?
  - Chris: yes, many cases state you can't tell the client that you're doing this
- TB: There are reasons we take back resources. And the parties doesn't agree. How does this scale, because if you have to evaluate each case and that puts a lot of burden on operators. I agree there is a problem, but this might be better for a 3rd-party auditor case that can tell you when things are going wrong [other than having each operator do it].
- ML: I don't believe it was a design goal to allow for revocations. It may be used to remove bad people from the internet, but it wasn't designed for that and i'm not sure we have working group consensus about it.
- RV: certain parties may interfere, but now maybe they'll think twice about it.

- WG: responding to take down not a design goal: while true, in practice, there is a whole set of things that we don't concern ourselves about, such as legal implications, etc. But we do have to be thinking about them none-the-less. We don't take the position about what is evil, but we do have to think about how that interacts with our systems.

- SMNH: What was a design goal was that the prefix allocation system has a particular structure and the RPKI is designed to enforce that structure. I can only allocate from what I currently have. That's the allocation system, and the RPKI models that system. Every contract says "if you mess up we get to take the allocation back". Any time there is a structure that permits enforcement, the same structure allows you to do new things that aren't good.

- WH: The thing about situations like this is that I can see the future where every parent requires the revocation keys from the children. And I also worry about grandparents being the one told to remove a grandchild.

- DM: the original goal was to ensure authorized holders of networked resources that they could announce them. It was short sighted to believe that they wouldn't need to change.