I2NSF Interim Meeting Minutes

Sept 6 2017; 16pm GMT – 17:40pm GMT

Participants: Yoav Nir, Alejandro Perez, Brian Weis, David Carrel, David Waltermire, Fermando (CUD), Gabriel Lopez (UMU), Henk Birkolz, Kathleen Moriarty, Kjeld Mortesen, Michael Richardson, Paul Wouters, Rafa Marin-Lopez, Robert Moskowitz, Russ Housley, Sowmini (Call-in-User_3), Susan Hares, Tero Kivinen, Wen Zhang, Velery Smyslov, Call-in-User_6, Call-in-User_7, Daniel Migault

Discussion Details

IPsec Deployments (Paul Wouters):

Paul Wouters give a presentation of IPsec VPN deployment status. The Discussion centered on if Transport Mode can traverse NAT.

Kathleen M: IPsec Transport Mode should have fixed the NAT issues, but the motivation hasn't been there yet. And it is not well deployed.

Michael Richardson: With effort, you can make transport mode work through NATs, but it requires significant in-kernel support to add additional keys to your socket structure. By significant, I mean, invasive. Easy if you own all the network stack, hard if you are trying to patch something in to someone else's stack.

Kathleen Moriarty: Yes and NAT is an implementation issue at this point if I understand correctly, so how do we motivate to get them fixed? I think NAT will matter in some data centers

Michael Richardson: It's not a protocol issue, it's an implementation issue.

Kathleen Moriarty: Yep, Michael, we agree - how do we get that fixed?

Michael Richardson: throw money at Redhat

Paul Wouters: use gateways site-to-site between data centers, then from client point of view, there is no NAT, and you can us Transport mode.

Michael Richardson: or deploy IPv6 in the data center.

Kathleen Moriarty: I think those that want to use it within their data center would find transport mode more attractive to be able to monitor at least the 2-tuple. I would think between many data centers, site-to-site is already in place.

Sowmini: why Transport Mode vs. Tunnel mode is a big deal?

Tero: Transport Mode send the inner IP address to the other end, therefore requiring the other end to know how to deal with the Private IP behind other site's NAT.

Scope of draft-abad-i2nsf-sdn-ipsec-flow-protection (Gabriel/Rafael)

Rafael described the preliminary assumption of the draft-abad-i2nsf-sdn-ipsec-flow-protection-03, so that the discussion can be more focused:

- A SDN controller is a trusted entity for the NSFs that it controls
- There is always a bidirectional secure communication channel between the SDN controller and the NSFs (e.g. NETCONF+SSL/SSH)
- The SDN controller can collect information from the NSFs and receive asynchronous notifications from them (e.g. if a NSF falls, the SDN controller will know). That is, the controller can monitor NSFs.
- Regardless of the SDN-based IPsec management, the SDN controller MUST be protected against attacks: it is a critic entity in the infrastructure

The purpose of the ID:

- I-D is a SDN-based automated key distribution technique. The SDN controller always generates fresh key material and parameters for IPsec management.
- The I-D specifies how to configure host-to-host (e.g. host-to-host VPN, full mesh encryption) and gateway-to-gateway (e.g. site-to-site VPN)
- Host-to-gateway (e.g. remote access VPN) deserves further study, especially in case 2
- Case 1 or case 2 can be used depending on the scenario
- Based on these general scenarios, it is possible to apply our I-D to different more specific uses cases (i.e. SD-WAN, communication between VMs in datacenters…)
- We believe the industry (e.g. SD-WAN) need to standardize an interface to operate with host-to-host and gateway-to-gateway (so far).
- It provides a solution that represents a tradeoff between powerful management and security

Yoav: Speaking as an individual, I think the scenario described by the draft is much better than before. There are a lot of IPsec VPNs being deployed, for example: The Gap has thousands of locations. Many locations are in a mall where public network is available. Typically a VPN gateway is placed in the store to connect to the public network. This local VPN gateway has no knowledge of the global network topology, nor does the controller know where it is located and what its domain is.

Kathleen: I think there are still questions about the keys.

Tero: If they use SDN then they can store all keys to and decrypt and monitor everything...

Kathleen Moriarty: that sounds scary. Group keying sounds better, but I don't know the details of how they would use the SND controller. The data center folks on this call (as far as I am aware) are just interested in automating deployment. The ones with a monitoring requirement may be a separate group

Tero: people in IPv6 community is very happy with the approach proposed by this draft. Many people have been complaining the complexity of IPsec configuration. Having a centralized place managing the keys is good. Most VPNs have some way of controller controlling the configuration of the device. There is no difference in the SDN case.

Presentation of the reasons to go against Case 2 (IPsec without IKE) (Tero)

Tero Kivinen: Use case 2 requires all keys being stored and generated by security controller. All keys in one central location can be an easy target for 3rd party to hack.

Alejandro Perez: Note that keys are not stored, they might but that would be a bad practice. They are generated, distributed and forgotten.

Tero Kivinen: They will be even if you say must not, as fast recovery requires it... the information need to send to multiple places.

Rafa: we may need to store the information for micro seconds or milliseconds.

Valery Smyslov: But you still need to store SAD of all the NSF, right?

Alejandro Perez to everyone: no, you don't. SAD is only stored on the NSFs

Valery Smyslov: OK, then you'll probably have hard time synchronizing SAs in case of any unpredictable event (NSF reboot)

Alejandro Perez: If a NSF falls, the Controller is notified and new SAs is established.

Valery Smyslov: Note, that SAs must not only be created, they must be deleted.

Alejandro Perez: I agree. Should not be worse than trying to establish the IKE SAs again

Valery Smyslov: IKE takes care of detecting and deleting erroneous SAs. Now it is SDN controller's task. So it must have a full picture of all active SAs.

Alejandro Perez: That's right. Basically a SDN controller has a full picture of the network it controls.

Valery Smyslov: So either it stores a list of all SAs (probably w/o keys) or must dynamically collect this information from NSFs, right?

Gabriel Lopez (UMU): It can collect this from the NSF.

Valery Smyslov: Note, that this list can be quite dynamic. It is really not a control plane, it is somewhere in between data plane and control plane. And you can put quite a high additional load on SDN to manage SAs. Won't it be a scalability problem here?

Alejandro Perez: The controller does not sends any data, why do you think it is in the data plane?

Valery Smyslov: I said it is in between data and control plane, because this information is not static and can change pretty quickly. For some high speed network you must create new IPsec SA every couple of minutes

Alejandro Perez: But the process of creating a SA is quite simplified. Don't even need a DH exchange. A good RNG is enough.

Paul Wouters: and have more than one between two endpoints while rolling to new key.

Fernando (CUD): normal IKE requires peer communication, case 2 is proposing to perform this communication with the controller instead with the other peer.

Alejandro Perez: The point is: the kernel does not know whether IKE or NETCONF is configuring the IPsec stack. It does not matter, nor care.

Paul Wouters: it does, whoever asks to see ACQUIRE, EXPIRE msgs etc

Michael Richardson: this has been fun,... I think the world would be better if the SDN controller would push the simplest configuration and authorization (really big PSK if you like) to the endpoints

Tero Kivinen: Usually kernel and IKE are quite tightly integrated, if you update one you need to update other too.

Michael Richardson: the data center/VM key stealing by the VM host issues are REAL.

Alejandro Perez: RFC 7296: To rekey a Child SA within an existing IKE SA, create a new, equivalent SA (see Section 2.17 below), and when the new one is established, delete the old one. It is not said that you have to wait until there is actual traffic

Paul Wouters: The core question is: at what point are you just creating a super-IKE daemon on a central node that uses PFKEY-over-TLS

Fernando (CUD): not a super-IKE daemon because the SDN controller is creating keying material that distributes to NFVs

Paul Wouters: distribute connection configs through the controller, and let the nodes run IKE.

Fernando (CUD) to everyone: there is no IKE negotiation.

Paul Wouters: you will rebuild some of it. like when there is a sha2_256 truncation bug in the kernel you have to work around

Fernando (CUD): in case 2 NFVs are not running IKE

Paul Wouters: you will only not run DH :)

Rafa: Why Case 2 is valid

Case 2 idea is from RFC4301: ".. but other automated key distribution techniques May be used". Our ID is using SDN based automated key distribution techniques.

Regarding implementation issues, we assume NETCONF

We are following the Kernel API to configure.

Tero: no one use the Kernel APIs. They might use some part of it. But mostly use proprietary interface.

Rafa: my point is that in the end, if you use any APIs to the Kernel, those APIs can be used by external controller.

Tero: everyone implements in different ways.

Michael Richardson: Linux kernel uses something like TFT that has been standardized by Forces. I don't see why can't you use standard methods.

Tero: if you assume there is standard APIs, you are set. But it is not that simple.

Rafa: In fact, there is available code from NSA to achieve this goal. If a device receive the information from controller, and pass it to the Kernel. We don't understand "Why Not".

Paul W: Kernel needs to tell it needs re-key.

Rafa: the controller can receive the notification and do the re-key.

Tero: It is more complicated. Speaking of re-key, when the kernel reach soft limit, it sends a message via API, both side wait and take actions. There is policy part. IPsec has two separate SA (Inbound & Outbound)

Rafa: we understand it. Just pull the complication to the controller.

Rafa: it is technical feasible, but have risks. Correct?

Tero: yes. Actually there are implementation of two separate parties managing the keys. But all proprietary.

Tero: it is more complicated.

Linda Dunbar: If it is technical feasible, but have risks, can we document the risks associated with the method in the draft?

Valery Smyslov: It has risks and is more complicated. So my question - what advantages case 2 has?

Paul Wouters: it does not have to be a central problem. if you have PFS via IKE

Paul Wouters: encryption keys and authentication keys do not need to be in the same egg basket

Henk Birkholz: Paul +1, there are pro&con to have specific & multi-purpose keys. Evidence relay via mitm is an attack vector, though, if keys are not entangled/associated somehow.

Regarding to NSF restart:

Rafa: if NSF restart, the NSF tell the controller, the controller tells the NSFs of the new key.

Tero: if you have any-to-any mesh connection, the controller has to inform thousands of the involved NSFs, very long process.

Rafa: understand this issue.

Rafa: our draft is only talk about if it is feasible.

Tero: it is technically possible. But not sure if you want to do.

Yoav Nir's note at the end:

Two things I thought of that were not mentioned in the call:

- Rafa said 4301 said that other automated keying methods (besides IKE) may be supported. At the time there was work to make a kerberos-based method. That ended up as RFC 4430, which has no deployment at all. I guess some of the debate will be about what "counts" as automated key management.
- IPsec tunnels are usually created on an as-needed basis. The kernel has a packet that should be encrypted according to policy, but it does not have keys, so it triggers the daemon to generate an appropriate SA. There is some latency involved, but if you add an extra network hop to the SDN controller, that latency might increase (or not, if it takes less round-trips). I did not see that trigger in the draft. So either they are planning to have tunnels that are always on, or that needs to be added.