Privacy Issue in VIPR

Marc Petit-Huguenin
03/27/2012

The problem

This problem was reported by Michael Procter, a member of the Design Team.

Let's say that VIPR is well deployed in US enterprises and that a spying group is willing to spend the money to know about the phone calls made by these enterprises. This spying group just have to add enough VIPR servers to the RELOAD overlay to register all the phone numbers in the USA. Because all registrations are considered equally untrusted, they will all be verified by establishing a TCP connection between the VIPR server of the source of the call and the VIPR server that stored the registration for a particular phone number. There is multiple pieces of information that are leaked but it is easy for example to find the enterprise that is originating the TCP connection by looking at the source address and to enter it in whois.

Information leakage

- If the spying group uses a different VServiceId for each registered phone number, the called number is always leaked.
- The called number is also leaked for methods “a” and “b”.
- For method "a", the Caller-ID is leaked (the bcrypt hash is hard to crack, but it is not impossible).
- For method "b", a random time in the middle of the call is leaked.
- For method "c", the rounded start and stop time of the call are leaked.
- The source IP address of the TCP connection for the PVP transaction is always leaked.
- The addr_port in the AppAttachReq RELOAD message that was used to establish the TCP connection is leaked.
- The certificate of the signer of the AppAttachReq RELOAD message is leaked.
- Even if the certificate does not contain information about the sender (subject, subjectAltName), it always contains the Node-ID, which can always be resolved to an IP address by using an Attach request.
- The Node-ID is leaked a second time in the via_list of the AppAttachReq message, unless an intermediary RELOAD peer replaced it with a compressed ID.

Anonymization of RELOAD

- Remove the username from the certificate (Split certificates, as proposed some time ago in p2psip). Remove also the other private information in the certificate.
- Each node needs two certificates, with two different Node-IDs. The first certificate is a normal certificate and is used for routing only. The second certificate is a Traceable Anonymous Certificate (RFC 5636), and is used to sign messages and data.
- Some nodes on the overlay needs to implement Onion Routing. The originating node select randomly 4 or 5 of these nodes (using Service Discovery) and uses the keys of each of these nodes to successively encrypt in a destination_list containing each of these nodes.
- When storing the ViprRegistration, the destination_list also contains a list of encrypted onion routers.
- Each onion node decrypts the destination_list before routing to the next destination.
- An additional protocol may be required between node and onion routing nodes to negotiate temporary keys for forward secrecy.
- AppAttach cannot be used, so additional messages needs to be added to RELOAD to tunnel the user protocol (TLS-SRP/ValExchange for VIPR) from the PVP client to the PVP server though the onion nodes.

What to do?

- Obviously it will take years of standardization to implement a perfect anonymization solution in RELOAD.

  We can:

  - Publish the drafts as they are (with a big security section) and start to work in parallel on the task of anonymizing RELOAD.
  - Put the working group on hold until we have a complete solution.
  - Other?