ZeroTier

Security Assessment (Summary)

March 23, 2020

Prepared For:
Adam Ierymenko | *ZeroTier, Inc.*
adam.ierymenko@zerotier.com

Prepared By:
James Miller | *Trail of Bits*
james.miller@trailofbits.com

Claudia Richoux | *Trail of Bits*
claudia.richoux@trailofbits.com

Paul Kehrer | *Trail of Bits*
paul.kehrer@trailofbits.com

Assessment Summary

During the week of March 23, 2020, Trail of Bits performed an assessment of the cryptographic components of the ZeroTier protocol. ZeroTier provided some documentation of the protocol and communicated further details about the protocol to Trail of Bits.

Our assessment was scoped to provide an assessment of ZeroTier's AES-GMAC-SIV construction; forward secrecy and ephemeral key generation; certificate creation, parsing, and validation; and choice of cryptographic algorithms adhering to FIPS compliance. We also aimed to provide general guidance regarding cryptographic constructions for the latest version, and to accurately describe the security guarantees and bounds associated with the protocol's design choices.

AES-GMAC-SIV

Because ZeroTier desires a secure, nonce–misuse-resistant authenticated encryption scheme that is FIPS compliant, they propose a variant of GCM-SIV called AES-GMAC-SIV. Trail of Bits assessed the security guarantees and bounds associated with this scheme. The analysis by Gueron and Lindell prove the original GCM-SIV scheme to be a secure, nonce–misuse-resistant authenticated encryption scheme. Although the AES-GMAC-SIV scheme is a variant of the original scheme, Trail of Bits concluded that the analysis by Gueron and Lindell still applies to this variant, making the AES-GMAC-SIV scheme a secure, nonce–misuse-resistant encryption scheme as well.

In order for the scheme to be secure, the message and additional data pair must be encoded uniquely. If pairs of messages and additional data are not encoded properly, encoding collisions could occur and violate the security of the scheme. Trail of Bits discussed this with ZeroTier, and they plan to ensure unique encoding in their implementation.

As part of the AES-GMAC-SIV construction, the plaintext is encrypted using AES in CTR mode. In CTR mode, it is dangerous if the same IV or counter are ever reused for the same key. The analysis by Gueron and Lindell provides security bounds on the amount of messages the scheme can securely encrypt before key rotation occurs; Trail of Bits has also concluded that the AES-GMAC-SIV construction achieves the same security bounds as the original construction.

AES-CTR has a maximum allowable plaintext size for a single encryption—typically 2^{32} -1 blocks, where each block is 128 bits—which affects this bound. Using this maximum, ZeroTier's construction allows for 2^{31} different encryptions to occur under the same key, while maintaining a probability of less than 2^{-32} of dangerous AES-CTR misuse (this probability level is a NIST requirement; see NIST SP 800-38D). According to ZeroTier, this bound of 2^{31} encryptions per key and the bound on maximum plaintext size both satisfy the constraints of the system.

Trust

The ZeroTier system operates on the assumption that the Network Controller (NC) is trusted. It uses Certificates of Membership (CoMs) signed by the NC to add new nodes by their public key to the system. Negotiation of keys for communication is done via the HELLO and OK messages in the V1 protocol, which are not vulnerable to amplification attacks or scanning. For future code assessments, we recommend ensuring that certificate parsing is done securely, and that all relevant information is properly authenticated.

CoMs expire after a set interval, and nodes compute the expiration by comparing the receiving node's most recently received CoM timestamp to the initiating node's presented CoM timestamp. The NC can also send a revocation of a CoM to all nodes, which is then passed via a gossip protocol to all other nodes. Finally, the tag system is based on the same establishment of identity, so permissions are based on this same trust system.

This system for managing node trust is mostly strong, but has some potential issues that should be considered. The protocol assumes that the NC will maintain good connectivity to all nodes, and that the NC's clock will generally be correct. When these assumptions do not hold, attacks can be mounted on the trust system that allow untrusted nodes to effectively bypass a firewall. We will present a few cases to illustrate this scenario, which can be generalized to other problematic network topologies or clocks on the NC.

First, consider a case in which the network is partitioned. Honest node A runs a service where nodes can access sensitive information. Node B has permission to access that information, but then is compromised and comes under full control of an attacker. Assume A and B are on the opposite side of the partition from the NC. If the NC learns about B's compromise, it cannot issue a revocation to A, and A will not have its certificate updated because it is not in contact with the NC. Therefore, B can indefinitely present an old CoM to A that allows it to access sensitive information. Even without compromise, CoMs will remain valid forever to all nodes that cannot update their CoMs.

Second, consider the case in which an attacker has control over the NTP servers for the NC. Incrementing this clock by the expiration period before issuing each CoM would lead to a denial of service, because no node would be able to communicate with any other node. If the clock is moved back in time, useless CoMs could be issued to some nodes and introduce a netsplit.

Suggestions for hardening this system include:

- Checking how much local time has elapsed since each node received their last certificate, and refusing communication after some period.
- Storing the most recent CoM time seen on any communication, and checking time deltas against that instead of the node's own CoM.
- Ensuring the NC verifies that time is monotonically increasing in software, and sanity-checking the timestamps of recently issued CoMs.
- Shortening the period for CoM validity, or making it configurable. This would reduce the window of time for an attack based on these situations.
- Having multiple NCs to make it harder to attack one NC's clock or successfully split all NCs off from any sizable number of nodes.
- Attempting to ensure that NCs have multiple paths to reach a node, and generally ensuring the network graph is well connected to avoid netsplits.
- Warning the user if the NC cannot access multiple trusted NTP servers.

Network Rules Engine

Our audit included a review of the network rules engine, as documented on the ZeroTier website. From a theoretical perspective this is secure, but we list some possible implementation issues below for consideration in future development and code auditing.

- Ensure that rule parsing and evaluation bugs are carefully ruled out.
- Consider potential situations in which timing attacks on rule evaluation due to non–constant-time rules and short-circuiting could leak configuration details or secret strings via a statistical attack. Think about adding functionality to recognize this situation and alert the user or drop packets from attacking hosts.
- Consider the speed of the implementation, and whether evaluation of slow rulesets could be leveraged for a denial-of-service attack.
- Consider developing functionality to help users create rulesets, and visualize or debug the effects and evaluation of their rulesets. Functionality might include generating truth tables and flowcharts, or drawing attention to slow code paths.

PKI, Forward Secrecy, and Ephemeral Keys

Key agreement is done with keys from two elliptic curves (Curve25519 and NIST P-384) for compatibility and NIST compliance purposes. Both keys are used separately to perform Diffie-Hellman, and the resulting shared keys are concatenated and hashed to create secure key material. Assuming the hash function is cryptographically secure, this construction has at least the same security of Diffie-Hellman, using whichever curve is considered more secure. The resulting protocol complies with a standard that accepts Diffie-Hellman using at least one of the two curves, e.g., FIPS requirements for the usage of NIST curves.

Forward secrecy is currently optional for compatibility purposes, but can be enforced network-wide by the NC, and the old protocol is being phased out. We recommend ensuring that the implementation enforces message–count-based and timing-based key regeneration; that messages encrypted with old keys are never accepted after key renegotiation (modulo some lag due to UDP); and that downgrade attacks disabling forward secrecy are impossible.

Node IDs

Nodes are identified by a 40-bit ZeroTier address. While precautions have been taken to ensure that the addresses are unique within a network, this may not be the case with the planned federation feature. The addresses are also difficult to compute from the public key, so collision attacks targeting individual nodes are very expensive (but perhaps not insurmountable for a nation-state attacker). Considering the birthday bound for accidental collisions and the potential size of the networks, collisions with addresses of this size may become a concern.

To mitigate this, we recommend minimizing the effects of a collision on encrypted packets routed in duplicate to the wrong host. We also recommend preventing denial-of-service attacks by adding a host with a duplicate address to the network, or advertising a duplicate address, thereby blackholing traffic to the original node. Further, we recommend preventing identification of a node by the ZeroTier address alone, without having a validated public key to back up its identity.

Finally, we recommend strictly defining where collisions are and are not permissible, how they are prevented, and how routing will occur if there is a collision. It is imperative that if node A previously held a ZeroTier address but is no longer on the network, and another node B joins with the same ZeroTier address, that no permissions, routes, keys, or sensitive data associated with A will be associated with or usable by B if that constitutes a vulnerability.

Conclusion

Overall, the assessment resulted in a series of constructive conversations about various components of ZeroTier's protocol. Trail of Bits has concluded the AES-GMAC-SIV construction satisfies its desired goals: It is a secure, nonce–misuse-resistant authenticated encryption scheme; it is FIPS compliant; and its security bounds fit within the system's constraints. The public-key infrastructure is also FIPS compliant.

As the protocol continues to evolve, we hope our recommendations and concerns are addressed. These concerns comprise theoretical attacks in which the attacker has some amount of control over the network infrastructure or sections of the code that could introduce serious vulnerabilities without careful consideration, but do not in themselves make ZeroTier an insecure protocol.

ZeroTier should also consider the effect of nodes and network infrastructure controlled by a powerful attacker, and assume nation-states have the resources to mount these attacks. Further, we recommend stating explicitly the protocol’s security guarantees and assumptions. Code implementations should be checked for compliance against the specification; writing these guarantees and assumptions clearly will help ensure compliance. Overall, we find the protocol to be well designed, and ZeroTier will be protected against wide classes of network attacks if it is implemented in line with the protocol described to Trail of Bits.