Protecting Internet Key Exchange (IKE) Implementations from Distributed Denial of Service Attacks

draft-ietf-ipsecme-ddos-protection-01

Yoav Nir (ynir.ietf@gmail.com)
Valery Smyslov (svan@elvis.ru)

IETF 92

DoS Attacks on IKE Responder

- Denial of Service (DoS) attacks on IKE responder aimed to exhaust its resources by initiating requests that either do not complete or are unnecessary
  - if attack is performed by multiple attackers then we call it Distributed Denial of Service (DDoS) attack
- The goal of the draft is to make the (D)DoS attacks costly for attackers

DoS Attacks on IKE Responder

- Depending on the point in the protocol flow, the attacks can be performed
  - in IKE_SA_INIT Exchange
  - in IKE_AUTH Exchange
  - after IKE SA is established
- IKE Resumption (RFC5723) can be used to minimize the impact of DoS attacks in IKE_SA_INIT and IKE_AUTH exchanges

DoS Attacks in IKE_SA_INIT

- The goal of attack is to exhaust responder's memory by creating large number of half-open IKE SAs
  - The attack costs nothing to attacker

DoS Attacks in IKE_SA_INIT – countermeasures

- Limit the number of half-open IKE SAs from a single IPv4 address (IPv6 prefix)
- Using stateless cookies would require attackers to only use real IP-addresses
- Using puzzles would require attackers to consume substantial computational resources to make a request

DoS Attacks in IKE_SA_INIT – puzzles

- Responder includes N(PUZZLE) along with N(COOKIE) into its response

```
<-- HDR, N(COOKIE), N(PUZZLE), [V+][N+]
```

- PUZZLE Notification contains
  - algorithm (PRF) to be used in puzzle calculation – is selected by Responder from PRF transforms in Initiator's SA Payload
  - puzzle difficulty level (requested number of trailing zero bits) – can be set to 0, meaning "get as many zero bits as you can"

DoS Attacks in IKE_SA_INIT – puzzles

- In IKE_SA_INIT the puzzle is: with given PRF and cookie find a key K such, that the result of `PRF(K,cookie)` contains the requested number of trailing zero bits
  - for example: with HMAC_SHA256 and 2.4 GHz single core i5 it takes from 0,5 to 5 seconds to get 20 bits depending on the cookie

DoS Attacks in IKE_SA_INIT – puzzles

- Legacy Initiators would ignore N(PUZZLE) and act as with stateless cookie from RFC7296
  - they would still have a chance to create IKE SA, but their requests would be marked with the lowest priority
- Initiator supporting puzzles would solve it and restart the exchange including N(COOKIE) along with the solution, which is contained in a new payload – Puzzle Solution (denoted as PS)

```
HDR, N(COOKIE), PS, SA, KE, Ni, [V+][N+] -->
```

DoS Attacks in IKE_SA_INIT – puzzles

- Upon receiving request with PS payload the responder would
  - verify the cookie and the solution
    - if verifications fail either discard the message or give the initiator a new puzzle
  - assign a priority to the request depending on
    - puzzle difficulty
    - the number of consecutive puzzles the initiator has solved
    - the amount of time it took initiator to solve them

DoS Attacks in IKE_SA_INIT – puzzles

- The responder weighs the priority of the request and its current load and either
  - accepts the request
  - rejects the request
  - gives the initiator another puzzle (in case the puzzle appeared too easy)
    - eventually the request either will be accepted or rejected

DoS Attacks in IKE_AUTH

- The attack on responder's CPU power and memory by sending garbage in IKE_AUTH request
  - The attack costs nothing to attacker once it completes IKE_SA_INIT exchange
- The attack on responder's CPU power by sending invalid credentials
  - The attack is costly for attacker since Responder must compute DH shared secret, that's why it is not considered in the draft

DoS Attacks in IKE_AUTH – puzzles

- If the puzzles were used in IKE_SA_INIT then the responder could also give the initiator a new puzzle to make the attack “garbage in IKE_AUTH request” costly; the responder includes N(PUZZLE) in IKE_SA_INIT response

```
<-- HDR, SA, KE, Nr, N(PUZZLE), [V+][N+]
```

  - PUZZLE Notification contains
    - algorithm (PRF) to be used in puzzle calculation – is selected from PRF transforms in Initiator’s SA Payload (may differ from that in IKE_SA_INIT puzzle)
    - puzzle difficulty level (cannot be 0 in this case)

DoS Attacks in IKE_AUTH – puzzles

- In IKE_AUTH the puzzle is slightly different than in IKE_SA_INIT: with given PRF, responder's nonce Nr and responder's SPI find a key K such, that the result of `PRF(K,Nr|SPIr)` contains the requested number of trailing zero bits
  - such construction allows the initiator to reuse the same puzzle for both unfragmented and fragmented IKE messages in case of switching to IKE fragmentation

DoS Attacks in IKE_AUTH – puzzles

- When initiator solves the puzzle it returns the solution in the PS payload outside the SK/SKF payload

```
HDR, PS, SK {IDi, [CERT,] [CERTREQ,] [IDr,] AUTH,
     SA, TSi, TSr} -->
```

- This would allow the responder to verify the solution before it spend a lot of CPU resources computing SKEYSEED and the SK_* keys

DoS Attacks in IKE_AUTH – puzzles

- The responder first verifies the solution
  - if verification fails the message is discarded
- If puzzle solution is OK the responder computes DH shared secret, SKEYSEED, SK_* keys and decrypts the SK/SKF payload
  - if puzzle solution is OK, but SK/SKF payload failed to pass ICV check, then the message is discarded, however the IKE SA is not immediately deleted and the computed keys are cached in it

Responder’s Strategy

- Constantly monitor peers activity and resource consumption
- Use the following countermeasures once (D)DoS attack is detected (in an order of increasing attack volume)
  - use stateless cookies
  - use IKE_SA_INIT puzzles (use also IKE_AUTH puzzles if correspondent attack is detected)
  - don’t use zero-level puzzle difficulty
  - increase puzzle difficulty

DoS Attacks after IKE SA is established

- The goal of the attacks is to force the victim to perform unnecessary work, like
  - performing continuous endless Liveness Check
  - continuous endless rekeying
  - creating numerous Child SAs with the same Traffic Selectors
- The attacks are targeted mostly on CPU power, however some of them can consume memory too

DoS Attacks after IKE SA is established

- All these requests are legal in the protocol, so the victim cannot just refuse to do them
- The amount of work is roughly the same for both the attacker and the victim, so the attacks are more likely be distributed
  - the attacker(s) could use NULL Authentication to remain anonymous

DoS Attacks after IKE SA is established – countermeasures

- Don’t increase IKE window size above the default value of 1
- Use TEMPORARY_FAILURE notification to limit the rate of rekeying
- Use NO_ADDITIONAL_SAS notification to limit the number of equal Child SAs
- Introduce artificial delays while responding to requests

Thanks

- Comments? Questions?
- More details are in the draft
- Please review it and send feedback to the authors