Investigation in HIP Proxies

draft-irtf-hiprg-proxies-02

Dacheng Zhang Huawei

X. Xu Huawei

J. Yao CNNIC

IETF 80 Prague

Current state

- **Modify the draft according to the comments on the list.**

Update according to Comments (1)

Termology Update

DI1 proxy -> DI-HIT proxy: A DI proxy which modifies the DNS answers and returns the HITs of HIP hosts instead to LHs

DI2 proxy -> DI-NAT proxy: A DI proxy which modifies the DNS answers and returns the IP addresses in its IP address pool to LHs

DI3 proxy -> DI-transparent proxy: A DI proxy does not make any modification to DNS lookup packets

Update according to Comments (2)

Comment

- *In section 3.4, you state that it is infeasible for an N-DI proxy to cache a packet and resolve it on the spot. I don't really think this is infeasible; this type of behavior occurs in on-demand routing, and probably also in HIP hosts that need to resolve HITs to IP addresses.*

- **Update**
  - Remove the “infeasible” statement in section 3.4. Instead, discuss how a N-DI proxy should be designed in this case:
    - “The information obtained from the DNS servers can be maintained within two lists. One list is for the information of HIP hosts; the other is for the information of legacy hosts.”

Update according to Comments(3)

section 3.4, you also suggest that it may be possible to return HITs instead of IP
addresses in HIP hosts' A AAAA records. I don't think this is a good idea. Since it would
effectively disable the ability to use DNS to contact the host outside of HIP, and it may confuse even other HIP hosts in the public network.

-

□ Remove the associated statement in section 3.4

□ Remove the associated statement in section 3.4

Update according to Comments (4)

- *In section 4, you categorize load balancing techniques as being those that divide up the*

  seems that there would be several scenarios in which such load balancing would work (basically, in any scenario where it can be guaranteed that the proxy selected can continue to be routed all of the packets of the flow).

  -

Update

- Add 4.1. LBMs adopting Load Balancers
  - 4.1.1. Load Balancer Supporting DI Proxy Components
  - 4.1.2. Load Balancer Supporting N-DI Proxies

Update according to Comments (5)

- **Comment**
  - *In section 5.2, you talk about a special type of HIP proxy-aware DNS server that is aware of load-balancing and modifies its behavior accordingly. I think that this could be avoided by either having the (load-balancing) proxies update the DNS on the current mapping, or if this is too much load, then use a RVS.*
- **Update**
  - Discuss the complexity imposed by this solution
  - Introduce the possibility of using RVS to solve the problem
    - “Another solution is to extend RVS servers as load balancers. After receiving an I1 packet from a HIP host, the load balancer then select an proper HIP proxy and forward the packet to it. Using this solution, a DNS server only needs to reply a record

forward the packet to it. Using this solution, a DNS server only needs to reply a record

on receiving a query from a HIP host, which reduce the traffic transported between
DNS servers and HIP hosts.”

Update according to Comments (7)

In section 9, you mention DNSSEC as a security consideration. Typically, this section

section 3 somewhere, and focus section 9 on security concerns of having these proxies in

Update

- Move the DNSSEC related discussion to section 3.5
- HIP proxies break the peer-to-peer security between HIP hosts and

LHs

  - It may be desired to let a HIP host to find out whether it is communicating with a HIP proxy or an ordinary HIP host
- DNS lookups needs to be secured

What I did else?

1. Introduction

- **Add a definition of the HIP proxy**

communicate with its desired HIP host without updating its protocol stack”

3.2. A Taxonomy of HIP Proxies

Add a clarification

- “Note that a DI proxy implementation may also be able to intercept the lookup between a LH and a resolution server other than DNS. However, currently DNS is the only resolution mechanism detailed analyzed and extended to support HIP communication. Hence, DNS is only resolution mechanism considered in this document.”

3.5. Distributed Implementation of DI Proxies

3.5. Distributed Implementation of DI

- **Add a discussion of**
  - The flexibility introduced by distributed HIP proxies.
- **Add three subsections:**
  - 3.5.1. Distributed DI-HIT Proxies
  - 3.5.2. Distributed DI-NAT Proxies
    - 3.5.3. Distributed DI-transparent Proxies
      - **In this case, a DI-transparent proxy component must be deployed on the boundary of the private network in order to guarantee that it can intercept packets.**

END

HIP

D. Kuptsov HIIT
J. Yao CNNIC

IETF 80 Prague

What’s changed (1)

- Add discussion in Security Consideration
  - Because the HI of a HIP host acts as both the identity and the public key of the HIP host at the same time. The revocation of a HI, the identity of the host is changed. Without the assistance of other mechanisms, the host will be a HI, all the TCP sessions identified with the associated HIT have to be broken.

– Because the HI of a HIP host acts as both the identity and the public key of the HIP host at the same time. The revocation of a HI, the identity of the host a HI, all the TCP sessions identified with the associated HIT have to be broken.

What’s changed (2)

secure anymore.

- Use the old keys generated by the old HI to send a suicide information

- In the cases where all the HIs of a host become invalid (e.g., the host is found to compromised), the host only can distribute the refreshment information using an out-of-band way.

What else?

non-HIP-based distinguished names (such as FQDN/NAI)."

  - Specify several procedures (how the expiration date on a HI can be set, how a HIP host finds that its HI has been compromised)
  - suggest techniques how a host may learn the new HIs from third parties

- More comments will be more than welcomed.

END