INTERNET-DRAFT
Oct 20, 2003

Erik Nordmark
Sun Microsystems
Tony Li
Procket Networks

Threats relating to IPv6 multihoming solutions

<draft-nordmark-multi6-threats-00.txt>

Status of this Memo

Abstract

This document lists security threats related to IPv6 multihoming. Multihoming can introduce new opportunities to redirect packets to different, unintended IP addresses.

The intent is to look at how IPv6 multihoming solutions might make the Internet less secure than the current Internet, without studying any proposed solution but instead looking at threats that are inherent in the problem itself. The threats in this document build upon the threats discovered and discussed as part of the Mobile IPv6 work.

Contents

1. INTRODUCTION

The goal of the IPv6 multihoming work is to allow a site to take advantage of multiple attachments to the global Internet without

having a specific entry for the site visible in the global routing table. Specifically, a solution should allow hosts to use multiple attachments in parallel, or to switch between these attachment points dynamically in the case of failures, without an impact on the upper layer protocols.

At the highest level the concerns about allowing such "rehoming" of packet flows can be called "redirection attacks"; the ability to cause packets to be sent to a place that isn't tied to the upper layer protocol's notion of the peer. These attacks pose threats against confidentiality, integrity, and availability. That is, an attacker might learn the contents of a particular flow by redirecting it to a location where the attacker has a packet recorder. If, instead of a recorder, the attacker changes the packets and then forwards them to the ultimate destination, the integrity of the data stream would be compromised. Finally, the attacker can simply use the redirection of a flow as a denial of service attack.

This document has been developed while considering multihoming solutions architected around a separation of network identity and network location. However, this separation is not a requirement for all threats, so this taxonomy may also apply to other approaches. This document is not intended to examine any single proposed solution. Rather, it is intended as an aid to discussion and evaluation of proposed solutions. By cataloging known threats, we can help to ensure that all proposals deal with all of the available threats.

2. TERMINOLOGY

upper layer protocol (ULP)
- a protocol layer immediately above IP. Examples are transport protocols such as TCP and UDP, control protocols such as ICMP, routing protocols such as OSPF, and Internet or lower-layer protocols being "tunneled" over (i.e., encapsulated in) IP such as IPX, AppleTalk, or IP itself.

interface - a node's attachment to a link.

address - an IP layer name that contains both topological

significance and acts as a unique identifier for an interface.

locator - an IP layer topological name for an interface or a set of interfaces.

identifier - an IP layer identifier for an IP layer endpoint (stack name in [NSRG]). The transport endpoint is a function of the transport protocol and would typically include the IP identifier plus a port number.

address field
- the source and destination address fields in the IPv6 header. As IPv6 is currently specified this fields carry "addresses". If identifiers and locators are separated these fields will contain locators.

FQDN - Fully Qualified Domain Name

3. TODAY'S ASSUMPTIONS

The two interesting aspects of security for multihoming solutions are the assumptions made by the applications and upper layer protocols about the identifiers that they see on one hand, and the existing abilities to perform redirection attacks today, on the other hand.

3.1. Application Assumptions

In the Internet today, the initiating part of applications either starts with a FQDN, which it looks up in the DNS, or already has an

IP address from somewhere. For the FQDN to IP address lookup the application effectively places trust in the DNS. Once it has the IP address, the application places trust in the routing system delivering packets to that address. Applications that use security mechanisms, such as IPsec or TLS, with mutual authentication have the ability to "bind" the FQDN to the cryptographic keying material thus compromising the DNS and/or the routing system can at worst cause the packets to be dropped or delivered to an entity which does not posses the keying material.

At the responding (non-initiating) end of communication today, we find applications that fall into approximately five classes with respect to their security requirements.

The first class is the set of public content servers. These systems provide data to any and all systems and are not particularly concerned with confidentiality, as they make their content available to all. However, they are interested in data integrity and denial of service attacks. Having someone manipulate the results of a search

engine, for example, or prevent certain systems from reaching a search engine would be a serious security issue.

The second class of applications use existing IP source addresses from outside of their immediate local site as a means of authentication without any form of verification. Today, with source IP address spoofing and TCP sequence number guessing as rampant attacks, such applications are effectively opening themselves for public connectivity and are reliant on other systems, such as firewalls, for overall security. We do not consider this class of systems in this document.

The third class of applications receive existing IP source addresses, but attempt some verification using the DNS, effectively using the FQDN for access control. (This is typically done by performing a reverse lookup from the IP address followed by a forward lookup and verifying that the IP address matches one of the addresses returned from the forward lookup.) These applications are already subject to a number of attacks using techniques like source address spoofing and TCP sequence number guessing since an attacker, knowing this is the case, can simply create a DoS attack using a forged source address that has authentic DNS records. In general this class of

applications is strongly discouraged, but it is probably important that a multihoming solution doesn't introduce any new and easier ways to perform such attacks.

The fourth class of applications use cryptographic security techniques to provide both a strong identity for the peer and data integrity with or without confidentiality. Such systems are still potentially vulnerable to denial of service attacks that could be introduced by a multihoming solution.

Finally, the fifth class of applications use cryptographic security techniques but without strong identity (such as opportunistic IPsec). Thus data integrity with or without confidentiality is provided when communicating with an unknown/unauthenticated principal. Just like the first category above such applications can't perform access control since they do not know the identity of the peer. [TBD: Does one-way authentication, without mutual authentication, add a different class of applications?]

The requirement for a multihoming solution is that security be no worse than it is today in all situations. Thus, mechanisms that provide confidentiality, integrity, or authentication today should continue to provide these properties in a multihomed environment.

3.2. Redirection Attacks Today

This section enumerates some of the redirection attacks that are possible in today's Internet.

If routing can be compromised, packets for any destination can be redirected to any location. This can be done by injecting a long prefix into global routing, thereby causing the longest match algorithm to deliver packets to the attacker.

Similarly, if DNS can be compromised, and a change can be made to an advertised resource record to advertise a different IP address for a hostname, effectively taking over that hostname.

Any system that is along the path from the source to the destination

host can be compromised and used to redirect traffic. Systems may be added to the best path to accomplish this. Further, even systems that are on multi-access links that do not provide security can also be used to redirect traffic off of the normal path. For example, ARP and ND spoofing can be used to attract all traffic for the legitimate next hop across an Ethernet.

Finally, the hosts themselves that terminate the connection can also be compromised and can perform functions that were not intended by the end user.

All of the above protocol attacks are the subject of ongoing work to secure them (DNSsec, security for BGP, Secure ND) and are not considered further within this document. The goal for a multihoming solution is not to solve these attacks. Rather, it is to avoid adding to this list of attacks.

3.3. Flooding Attacks Today

In the Internet today there are several ways for an attacker to use a redirection mechanism to launch DoS attacks that can not easily be traced to the attacker. An example of this is to use protocols which cause reflection with or without amplification [PAXSON01]. Reflection without amplification can be accomplished by an attacker sending a TCP SYN packet to a well-known server with a spoofed source address; the resulting TCP SYN ACK packet will be sent to the spoofed source address.

Devices on the path between two communicating entities can also launch DoS attacks. While such attacks might not be interesting today, it is necessary to understand them better in order to determine whether a multihoming solution might enables new types of

DoS attacks.

For example, today if A is communicating with B, then A can try to overload the path from B to A. If TCP is used A could do this by sending ACK packets for data that it has not yet received (but it suspects B has already sent) so that B would send at a rate that would cause persistent congestion on the path towards A. Such attack would seem self-destructive since A would only make its own

corner of the network suffer by overloading the path from the Internet towards A.

A more interesting case is if A is communicating with B and X is on the path between A and B, then X might be able to fool B to send packets towards A at a rate that is faster than A (and the path between A and X) can handle. For instance, if TCP is used then X can craft TCP ACK packets claiming to come from A to cause B to use a congestion window that is large enough to potentially cause persistent congestion towards A. Furthermore, if X can suppress the packets from A to B it can also prevent A from sending any explicit "slow down" packets to B. Similar attacks can presumably be launched using protocols that carry streaming media by forging such a protocol's notion of acknowledgment and feedback.

An attribute of this type of attack is that A will simply think that B is faulty since its flow and congestion control mechanisms don't seem to be working. Detecting that the stream of ACK packets is generated from X and not from A might be challenging, since the rate of ACK packets might be relatively low. This type of attack might not be common today because it requires that X remain on the path in order to sustain the DoS attack, but the addition of multihoming redirection mechanisms might potentially remove that constraint. And with the current, no-multihoming support, using end-to-end strong security at a protocol level at (or below) this "ACK" processing would prevent this type of attack. But if a multihoming solution is provided underneath IPsec that prevention mechanism would not exist.

Thus the challenge for multihoming solutions is to not create additional types of attacks in this area, or make existing types of attacks significantly easier.

4. POTENTIAL NEW REDIRECTION ATTACKS

This section documents the additional redirection attacks that have been discovered that result from an architecture where hosts can change their topological connection to the network in the middle of a transport session without interruption. This discussion is again framed in the context of independent host identifiers and topological locators. Some of these attacks may not be applicable if traditional addresses are used. This section assumes that each host has multiple locators and that there is some mechanism for determining the locators for a correspondent host. We do not assume anything about the properties of these mechanisms. Instead, this list will serve to help us derive the properties of these mechanisms that will be necessary to prevent these redirection attacks.

Depending on the purpose of the redirection attack we separate the attacks into several different types.

4.1. Cause Packets to be Sent to the Attacker

An attacker might want to receive the flow of packets, for instance to be able to inspect and/or modify the payload or to be able to apply cryptographic analysis to cryptographically protected payload, using redirection attacks.

4.1.1. Once Packets are Flowing

This might be viewed as the "classic" redirection attack.

While A and B are communicating X might send packets to B and claim: "Hi, I'm A, send my packets to my new location." where the location is really X's location.

"Standard" solutions to this include requiring the node requesting redirection somehow be verified to be the same node as the initial node to establish communication. However, the burdens of such verification must not be onerous, or the redirection requests themselves can be used as a DoS attack.

To prevent this type of attack, a solution would need some mechanism that B can use to verify whether a locator belongs to A before B starts using that locator, and be able to do this when multiple locators are assigned to A.

4.1.2. Premeditated Redirection

This is a variant of the above where the attacker "installs" itself before communication starts.

For example, if the attacker X can predict that A and B will communicate in the (near) future, then the attacker can tell B: "Hi, I'm A and I'm at this location". When A later tries to communicate with B, will B believe it is really A?

If the solution to the classic redirection attack is based on "prove you are the same as initially", then A will fail to prove this to B since X initiated communication.

Depending on details that would be specific to a proposed solution, this type of attack could either case redirection (so that the packets intended for A will be sent to X) or they could cause DoS (where A would fail to communicate with B since it can't prove it is the same node as X).

To prevent this attack, the verification whether a locator belongs to the peer can not simply be based on the first peer that made contact.

4.1.3. Using Replay Attacks

While the multihoming problem doesn't inherently imply any topological movement it is useful to also consider the impact of site renumbering in combination with multihoming. In that case the set of locators for a node will change each time its site renumbers and at some point in time after a renumbering event the old locator prefix might be reassigned to some other site.

This potentially opens up the ability for an attacker to replay whatever protocol mechanism was used to inform a node of a peer's locators so that the node would incorrectly be lead to believe that the old locator (set) should be used even long after a renumbering event. This is similar to the risk of replay of Binding Updates in [MIPv6] but the time constant is quite different; Mobile IPv6 might see movements every second while site renumbering followed by reassignment of the site locator prefix might be a matter of weeks or months.

To prevent such replay attacks the protocol which is used to verify which locators can be used with a particular identifier needs some replay protection mechanism.

Also, in this space one needs to be concerned about potential

interaction between such replay protection and the administrative act of reassignment of a locator. If the identifier and locator relationship is distributed across the network one would need to make sure that the old information has been completely purged from the network before any reassignment. Note that this does not require explicit mechanism. This can instead be implemented by locator reuse policy and careful timeouts of locator information.

4.2. Cause Packets to be Sent to a Black Hole

This is also a variant of the classic redirection attack. The difference is that the new location is a locator that is nonexistent or unreachable. Thus the effect is that sending packets to the new locator causes the packets to be dropped by the network somewhere.

One would expect that solutions which prevent the previous redirection attacks would prevent this attack as a side effect, but it makes sense to include this attack here for completeness. Mechanisms that prevented a redirection attack to the attacker should also prevent redirection to a black hole.

4.3. Third Party Denial-of-Service Attacks

An attacker can use the ability to perform redirection to cause overload on an unrelated third party. For instance, if A and B are communicating then the attacker X might be able to convince A to send the packets intended for B to some third node C. While this might seem harmless at first, since X could just flood C with packets directly, there are a few aspects of these attacks that cause concern.

The first is that the attacker might be able to completely hide its identity and location. It might suffice for X to send and receive a few packets to A in order to perform the redirection, and A might not retain any state on who asked for the redirection to C's location.

Even if A had retained such state, that state would probably not be easily available to C, thus C can't determine who was the attacker once C is being DoS'ed.

The second concern is that with a direct DoS attack from X to C, the attacker is limited by the bandwidth of its own path towards C. If the attacker can fool another node like A to redirect its traffic to C then the bandwidth is limited by the path from A towards C. If A is a high-capacity Internet service and X has slow (e.g., dialup)

connectivity this difference could be substantial. Thus in effect this could be similar to packet amplifying reflectors in [PAXSON01].

The third, and final concern, is that if an attacker only need a few packets to convince one node to flood a third party, then it wouldn't be hard for the attacker to convince lots of nodes to flood the same third party. Thus this could be used for Distributed Denial-of-Service attacks.

In today's Internet the ability to perform this type of attack is quite limited. In order for the attacker to initiate communication it will in most cases need to be able to receive some packets from the peer (the potential exception being combining this with TCP sequence number guessing type of techniques). Furthermore, to the extent that parts of the Internet uses ingress filtering [INGRESS], even if the communication could be initiated it wouldn't be possible to sustain it by sending ACK packets with spoofed source addresses from an off-path attacker.

If this type of attack can't be prevented there might be mitigation techniques that can be employed. For instance, in the case of TCP it would help if TCP slow-start was triggered when the destination locator changes. (Folks might argue that, separately from security, this would be the correct action for congestion control since TCP might not have any congestion-relation information about the new path implied by the new locator). Applying this technique to other ULPs which perform different forms of (TCP friendly) congestion control might be more difficult since the lower layers generally lack an API to provide such information to the ULPs. Also, for other protocols, this might be less beneficial, since other ULPs might not adapt rapidly and could view the suggestion of congestion as being more severe than a simple deficit of congestion information.

4.3.1. Basic Third Party DoS

Assume that X is on a slow link anywhere in the Internet. B is on a fast link (gigabits; e.g. a media server) and A is the victim.

X could flood A directly but is limited by its low bandwidth. If X can establish communication with B, ask B to send it a high-speed media stream, then X can presumably fake out the "acknowledgments/feedback" needed for B to blast out packets at full speed. So far this only hurts X - and the path between X and the Internet. But if X could also tell B "I'm at A's locator" then X has effectively used this redirection capability in multihoming to amplify its DoS capability, which would be a source of concern.

One could envision rather simple techniques to prevent such attacks. For instance, before sending to a new peer locator perform a clear text exchange with the claimed new locator of the form "Are you X?" resulting in "Yes, I'm X.". This would suffice for the simplest of attacks. However, as we will see below, more sophisticated attacks are possible.

4.3.2. Third Party DoS with On-Path Help

The scenario is as above but in addition the attacker X has a friend Y on the path between A and B:

```
   -----        -----        -----
   | A |--------| Y |--------| B |
   -----        -----        -----
                            /
                           /
                          /
                         /
                        /
                       /
                   -----
                   | X |
                   -----
```

With the simple solution suggested in the previous section, all Y might need to do is to fake a response to the "Are you X?" packet, and after that point in time Y might not be needed; X could potentially sustain the data flow towards A by generating the ACK packets. Thus it would be even harder to detect the existence of Y.

Furthermore, if X is not the actual end system but an attacker between some node C and B, then X can claim to be C, and no finger can be pointed at X either:


```
      -----        -----        -----
      | A |--------| Y |--------| B |
      -----        -----        -----
                                  /
                                 /
                                /
                               /
                              /
                             /
           -----       -----
           | C |-------| X |
           -----       -----
```

Thus with two attackers on different paths, there might be no trace of who did the redirection to the 3rd party once the redirection has taken place.

A specific case of this is when X=Y, and X is located on the same LAN

as B.

A potential way to make such attacks harder would be to use the last received (and verified) source locator as the destination locator. That way when X sends the ACK packets (whether it claims to be X or C) the result would be that the packet flow from B would switch back towards X/C, which would result in an attack similar to what can be performed in today's Internet.

Another way that a multihoming solution might address this is to ensure that B will only accept locators that can be authenticated to be synonymous with the original correspondent. It must be possible to securely ensure that these locators form an equivalence class. So in the first example, not only does X need to assert that it is A, but A needs to assert that it is X.

4.4. Accepting Packets from Unknown Locators

The multihoming solution space does not only affect the destination of packets; it also raises the question from which sources packets should be accepted. It is possible to build a multihoming solution that allows traffic to be recognized as coming from the same peer even if there is a previously unknown locator present in the source address field. The question is whether we want to allow packets from unverified sources to be passed on to upper layer protocols.

In the current Internet, an attacker can't inject packets with arbitrary source addresses into a session if there is ingress

filtering present, so allowing packets with unverified sources in a multihoming solution would fail our "no worse than what we have now" litmus test. However, given that ingress filtering deployment is far from universal and ingress filtering typically wouldn't prevent spoofing of addresses in the same subnet, requiring rejecting packets from unverified locators might be too stringent. A factor to take into account to determine the "requirement level" for this is that when IPsec is used on top of the multihoming solution, then IPsec will reject such spoofed packets. (Note that this is different than in the redirection attack cases where even with IPsec an attacker could potentially cause a DoS attack.)

There might also be a middle ground where arbitrary attackers are prevented from injecting packets by using the SCTP verification tag type of approach [SCTP]. (This is a clear-text tag which is sent to the peer which the peer is expected to include in each subsequent packet.) Such an approach doesn't prevent packet injection from on-path attackers (since they can observe the verification tag), but neither does ingress filtering.

5. OTHER SECURITY CONCERNS

The protocol mechanisms added as part of a multihoming solution shouldn't introduce any new DoS in the mechanisms themselves. In particular, care must be taken not to:

- create state on the first packet in an exchange, since that could result in state consumption attacks similar to the TCP SYN flooding attack.

- perform much work on the first packet in an exchange (such as expensive verification)

There is a potential chicken-and-egg problem here, because potentially one would want to avoid doing work or creating state until the peer has been verified, but verification will probably need some state and some work to be done.

A possible approach that solutions might investigate is to defer verification until there appears to be two different nodes (or two different locators for the same node) that want to use the same identifier.

Another possible approach is to first establish communications, and then perform verification in parallel with normal data transfers. Redirection would only be permitted after verification was complete, but prior to that event, data could transfer in a normal, non-

multihomed manner.

Finally, the new protocol mechanisms should be protected against spoofed packets, at least from off-path sources, and replayed packets.

6. SECURITY CONSIDERATIONS

In section 3 we discussed existing protocol-based redirection attacks. But there are also non-protocol redirection attacks. An attacker which can gain physical access to one of

- The copper/fiber somewhere in the path.

- A router or L2 device in the path.

- One of the end systems

can also redirect packets. This could be possible for instance by physical break-ins or by bribing staff that have access to the physical infrastructure. Such attacks are out of scope for this discussion, but are worth to keep in mind when looking at the cost for an attacker to exploit any protocol-based attacks against multihoming solutions; making protocol-based attacks much more expensive to launch than break-ins/bribery type of attacks might be overkill.

7. ACKNOWLEDGMENTS

This document is a product of a MULTI6 design team consisting of (in alphabetical order): Iljitsch van Beijnum, Steve Bellovin, Brian Carpenter, Mike O'Dell, Sean Doran, Dave Katz, Tony Li, Erik Nordmark, and Pekka Savola.

Much of the awareness of these threats come from the work on Mobile IPv6 [MIPv6, NIKANDER03, AURA02].

8. REFERENCES

8.1. Normative References

8.2. Informative References

[NSRG] Lear, E., and R. Droms, "What's In A Name: Thoughts from the NSRG", draft-irtf-nsrg-report-09.txt (work in progress), March 2003.

[MIPv6] Johnson, D., C. Perkins, and J. Arkko, "Mobility Support in IPv6", draft-ietf-mobileip-ipv6-24.txt (work in progress), June 2003.

[AURA02] Aura, T. and J. Arkko, "MIPv6 BU Attacks and Defenses", draft-aura-mipv6-bu-attacks-01 (work in progress), March 2002.

[NIKANDER03] Nikander, P., T. Aura, J. Arkko, G. Montenegro, and E. Nordmark, "Mobile IP version 6 Route Optimization Security Design Background", draft-nikander-mobileip-v6-ro-sec-01 (work in progress), June 2003.

[PAXSON01] V. Paxson, "An Analysis of Using Reflectors for Distributed Denial-of-Service Attacks", Computer Communication Review 31(3), July 2001.

[INGRESS] Ferguson P., and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", RFC 2827, May 2000.

[SCTP] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H.

Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "Stream Control Transmission Protocol", RFC 2960, October 2000.

[ADDR-ARCH] S. Deering, R. Hinden, Editors, "IP Version 6 Addressing Architecture", RFC 3513, April 2003.

[IPv6] S. Deering, R. Hinden, Editors, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2461.

[IPv6-SA] R. Atkinson. "Security Architecture for the Internet Protocol". RFC 2401, November 1998.

[IPv6-AUTH] R. Atkinson. "IP Authentication Header", RFC 2402, November 1998.

[IPv6-ESP] R. Atkinson. "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

AUTHORS' ADDRESSES

Erik Nordmark
Sun Microsystems, Inc.
17 Network Circle
Mountain View, CA
USA

phone: +1 650 786 2921
fax: +1 650 786 5896
email: erik.nordmark@sun.com

Tony Li
Procket Networks, Inc.
1110 Cadillac Ct.
Milpitas, CA
USA

phone: +1 408 635 7903
fax: +1 408 635 7522
email: Tony.Li@procket.com

Full Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.