INTERNET-DRAFT
Sept 22, 2004

Erik Nordmark
Sun Microsystems
Tony Li

Threats relating to IPv6 multihoming solutions
<draft-ietf-multi6-multihoming-threats-01.txt>

Status of this Memo

Copyright Notice

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 transport and application 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 transport and/or application 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, whether or not this separation implies the introduction of a new and separate identifier name space. 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.

1.1. Assumptions

This threat analysis doesn't assume that security has been applied other security relevant parts of the Internet, such as DNS and routing protocols, but it does assume that at some point in time at least parts of the Internet will be operating with security for such key infrastructure. With that assumption it then becomes important that a multihoming solution would not, at that point in time, become the weakest link. This is the case even if, for instance, insecure DNS might be the weakest link today.

This document doesn't assume that the application protocols are protected by strong security today or in the future. However, it is still useful to assume that the application protocols which care

about integrity and/or confidentiality apply the relevant end-to-end security measures, such as IPsec, TLS, and/or application layer security.

For simplicity we assume that an on-path attacker can see packets, modify packets and send them out, and block packets from being delivered. This is a simplification because there might exist ways, for instance monitoring capability in switches, which allow authenticated and authorized users to observe packets without being able to send or block the packets.

> [DISCUSSION: It isn't clear to what extent the above simplifying assumption is misleading, that is, whether we need to make a distinction between on-path attackers which can monitor packets and perhaps also inject packets, without being able to block packets from passing through.
>
> On-path attackers that only need to monitor might be lucky and find a non-switched Ethernet in the path, or use capacitive or inductive coupling to listen on a copper wire. But if the attacker is on an Ethernet that is on the path, whether switched or not, the attacker can also employ ARP/ND spoofing to get access to the packet flow which allows blocking as well. Similarly, if the attacker has access to the wire, the attacker can also place a device on the wire to block. Other on-path attacks would be those that gain control of a router or a switch (or gain control of one of the endpoints) and I think those would allow blocking as well.
>
> So the strongest currently known case where monitoring (and

injecting?) is easier than blocking, is when switches and routers have monitoring capabilities (for network management or for lawful intercept) where an attacker might be able to bypass the authentication and authorization checks for those capabilities.]

We assume that an off-path attacker can neither see packets between the peers (for which it is not on the path) nor block them from being delivered. Off-path attackers can in general send packets with arbitrary IP source addresses and content, but such packets might be blocked if ingress filtering [INGRESS] is applied. Thus it is important to look at the multihoming impact on security both in the presence and absence of ingress filtering.

1.2. Authentication, Authorization, and Identifier Ownership

The overall problem domain can be described using different terminology.

One way to describe it is that it is necessary to first authenticate

the peer and then verify that the peer is authorized to control the locators used for a particular identifier. While this is correct, it might place too much emphasis on the authorization aspect. In this case the authorization is conceptually very simple; each host (each identifier) is authorized to control which locators are used for itself.

Hence there is a different way to describe the same thing. If the peer can somehow prove that it is the owner of the identifier, then the peer can control the locators that are used with the identifier. This way to describe the problem is used in [OWNER].

Both ways to look at the problem, hence both sets of terminology, are useful when trying to understand the problem space and the threats.

2. TERMINOLOGY

link - a communication facility or medium over which nodes can communicate at the link layer, i.e., the layer immediately below IPv6. Examples are Ethernets (simple or bridged); PPP links; X.25, Frame Relay, or ATM networks; and Internet (or higher) layer

"tunnels", such as tunnels over IPv4 or IPv6 itself.

interface - a node's attachment to a link.

address - an IP layer name that has both topological significance (i.e., a locator) and identifies an interface. There may be multiple addresses per interface.

locator - an IP layer topological name for an interface or a set of interfaces. There may be multiple locators per interface.

identifier - an IP layer identifier for an IP layer endpoint (stack name in [NSRG]), that is, something that might be commonly referred to as a "host". The transport endpoint name is a function of the transport protocol and would typically include the IP identifier plus a port number. There might be use for having multiple interfaces per stack/per host.

The identifiers are not associated with an interface.

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 [FYI18]

3. TODAY'S ASSUMPTIONS AND ATTACKS

The two interesting aspects of security for multihoming solutions are the assumptions made by the transport layer and application layer protocols about the identifiers that they see on one hand, and the existing abilities to perform today various attacks related to the identity/location relationship, 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 [RFC1948], 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 because they are in any case fully open to all forms of network layer spoofing.

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. However, we note that some people think we should treat this class the same as the second class of applications.

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 based on network layer information since they do not know the identity of the peer. However, they might perform access control using higher-level notions of identity. The availability of IPsec (and similar solutions) together with channel bindings allow protocols which in themselves are vulnerable to MiTM-attacks to operate with a high level of confidentiality in the security of the identification of the peer. A typical example is the Remote Desktop Protocol (RDP) which when used with opportunistic IPsec works well if channel bindings are available. Channel bindings provide a link between the IP-layer identification and the application protocol

identification. This is an important aspect of security in

application protocols which must be preserved by a multihoming solution.

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. More detailed information about threats relating to DNS are in [DNS-THREATS].

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. And since the vast majority of applications rely on DNS lookups, if DNSsec is not deployed, then attackers that are on the path between the host and the DNS servers can also cause redirection by modifying the responses from the DNS servers.

In general the above attacks work only when the attacker is on the path at the time it is performing the attack. However, in some cases it is possible for an attacker to create a DoS attack which remains at least some time after the attacker has moved off the path. An example of this is an attacker which uses ARP or ND spoofing while on path to either insert itself or send packets to a black hole (a non-existent L2 address). After the attacker moves away the ARP/ND entries will remain in the caches in the neighboring nodes for some amount of time; a minute or so in the case of ARP. This will result in packets continuing to be black-holed until ARP entry is flushed.

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. Packet Injection Attacks Today

In today's Internet the transport layer protocols, such as TCP and SCTP, which use IP, use the IP addresses as the identifiers for the communication. In the absence of ingress filtering [INGRESS] the IP layer allows the sender to use an arbitrary source address, thus the transport protocols and/or applications need some protection against malicious senders injecting bogus packets into the packet stream between two communicating peers. If this protection can be circumvented, then it is possible for an attacker to cause harm without necessarily needing to redirect the return packets.

There are various level of protection in different transport protocols. For instance, in general TCP packets have to contain a sequence that falls in the receiver's window to be accepted. If the TCP initial sequence numbers are random then it is relatively hard for an off-path attacker to guess the sequence number close enough for it to belong to the window, and as result be able to inject a packet into an existing connection. How hard this is depends on the size of the available window. SCTP provides a stronger mechanism with the verification tag; an off-path attacker would need to guess this random 32-bit number. Of course, IPsec provide cryptographically strong mechanisms which prevent attackers on or off path to inject packets once the security associations have been established.

When ingress filtering is deployed between the potential attacker and the path between the communicating peers, it can prevent the attacker from using the peer's IP address as source. In that case the packet injection will fail in today's Internet.

We don't expect a multihoming solution improve the existing degree of prevention against packet injection. However, it is necessary to look carefully whether a multihoming solution makes it easier for attackers to inject packets since the desire to have the peer be present at multiple locators, and perhaps at a dynamic set of locators, can potentially result in solutions that, even in the

presence of ingress filtering, make packet injection easier.

3.4. 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 an 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, that is, X can disable any flow or congestion control mechanism such as Explicit Congestion Notification [ECN]. 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, in the presence of ingress filtering, it requires that X remain on the path in order to sustain the DoS

attack. And in the absence of ingress filtering an attacker would either need to be present on the path initially and then move away, or the attacker would need to be able to perform the setup of the communication "blind" i.e., without seeing the return traffic (which in the case of TCP implies guessing the initial sequence number).

The danger is that the addition of multihoming redirection mechanisms might potentially remove the constraint that the attacker remain on the path. 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 potentially 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.

3.5. Address Privacy Today

In today's Internet there is limited ability to track a host as it uses the Internet because in some cases, such as dialup connectivity, the host will acquire different IPv4 addresses each time it connects. However, with increasing use of broadband connectivity, such as DSL or cable, it is becoming more likely that the host will maintain the same IPv4 over time. Should a host move around in today's Internet, for instance, by visiting WiFi hotspots, it will be configured with a different IPv4 address at each location.

We may also observe that a common practice in IPv4 today is to use some form of address translation, whether the site is multihomed or not. This effectively hides the identity of the specific host within

a site; only the site can be identified based on the IP address.

In the cases where it is desirable to maintain connectivity as a host moves around, whether using layer 2 technology or Mobile IPv4, the IPv4 address will remain constant during the movement (otherwise the connections would break). Thus there is somewhat of a choice today between seamless connectivity during movement and increased address privacy.

Today when a site is multihomed to multiple ISPs the common setup is that a single IP address prefix is used with all the ISPs. As a result it is possible to track that it is the same host that is communication via all ISPs.

However, when a *host* is multi-homed to several ISP, e.g. through a GPRS connection and a wireless hot spot, the host is provided with

different IP addresses on each interface. While the focus of the multihoming work is on site multihoming, should the solution also be applicable to host multihoming, the privacy impact needs to be considered for this case as well.

IPv6 stateless address auto-configuration facilitates IP address management, but raises some concerns since the Ethernet address is encoded in the low-order 64 bits of the IPv6 address. This could potentially be used to track a host as it moves around the network, using different ISPs etc. IPv6 specifies temporary addresses [RFC3041] which allow applications to control whether they need long-lived IPv6 addresses or desire the improved privacy of using temporary addresses.

Given that there isn't address privacy in site multihoming setups today, the primary concerns for the "do no harm" criteria are to ensure that hosts that move around still have the same ability as in today's Internet to choose between seamless connectivity and improved address privacy, and also that the introduction of multihoming support should still provide the same ability as we have in IPv6 with temporary addresses.

When considering privacy threats it makes sense to distinguish between attacks make by on-path entities observing the packets flying by, and attacks by the communicating peer. While it is probably

feasible to prevent on-path entities from correlating the multiple IP addresses of the host, the fact that the peer needs to be told multiple IP addresses in order to be able to switch to using different addresses when communication fails limits the ability of the host to prevent correlating its multiple addresses. However, using multiple pseudonyms for a host should be able address this case.

4. POTENTIAL NEW ATTACKS

This section documents the additional 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 where the topological locators may be independent of the host identifiers used by the transport and application layer protocols. 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 host requesting redirection somehow be verified to be the same host as the initial host 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. Time-shifting Attack

The term "time-shifting attack" is used to describe an attackers ability to perform an attack after no longer being on the path. Thus the attacker would have been on the path at some point in time, snooping and/or modifying packets, and later when the attacker is no longer on the path, it launches the attack.

In the current Internet, it is not possible to perform such attacks to redirect packets. But for sometime after moving away the attacker can cause a DoS attack e.g. by leaving a bogus ARP entry in the nodes on the path or by forging TCP Reset packets based on having seen the

TCP Initial Sequence Numbers when it was on the path.

It would be reasonable to require that a multihoming solution limit the ability to redirect and/or DoS traffic to a few minutes after the attacker has moved off the path.

4.1.3. 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 cause 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 host 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.

4.1.4. 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 host 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 host of a peer's locators so that the host 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 an 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 host 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 host to flood a third party, then it wouldn't be hard for the attacker to convince lots of hosts 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 transport protocols 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 transports. Also, for other protocols, this might be less beneficial, since other transports 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

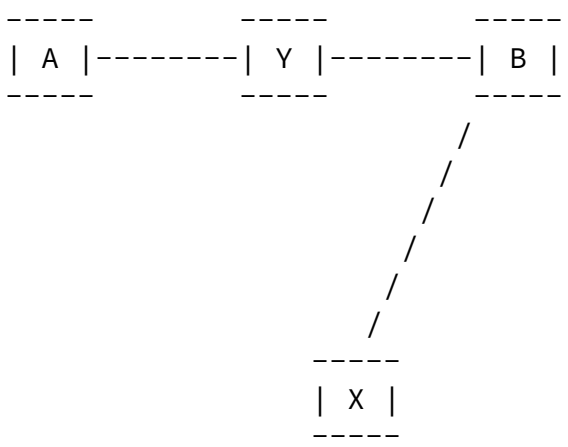
Internet. But if X could also tell B "I'm at A's locator" then X has

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.

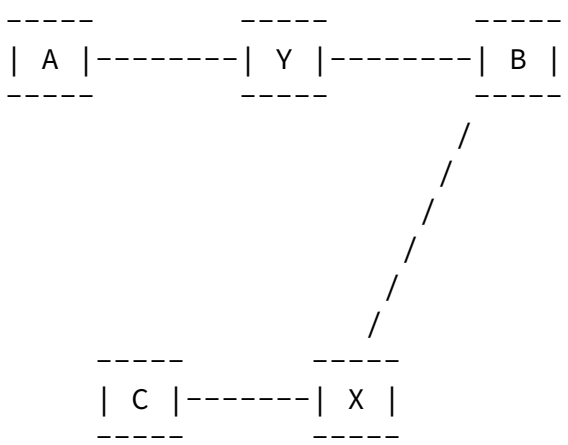
4.3.2. Third Party DoS with On-Path Help

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



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:



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 to make such attacks harder would be to perform periodic verifications that the peer is available at the locator, instead of just one when the new locator is received.

A third 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 transport and application 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.

An example of the current state are the ability to inject RST packets into existing TCP connections. When there is no ingress filtering in the network, this is something that the TCP endpoints need to sort out themselves. However, deploying ingress filtering helps in today's Internet since an attacker is limited in the set of source address it can use.

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.

4.5. New Privacy Considerations

While introducing identifiers can be helpful by providing ways to identify hosts across events when its IP address(es) might change, there is a risk that such mechanisms can be abused to track the identity of the host over long periods of time, whether using the same (set of) ISP(s) or moving between different network attachment points. Designers of solutions to multihoming need to be aware of this concern.

Introducing the multihoming capability inherently implies that the communicating peers need to know multiple locators for each other in order to be resilient to failures of some paths/locators. In addition, if the multihoming signaling protocol doesn't provide privacy it might be possible for 3rd parties on the path to discover

many or all the locators assigned to a host, which would increase the privacy exposure compared to a multihomed host today.

A solution could address this for instance by allowing each host to have multiple identifiers at the same time and perhaps even changing the set of identifiers that are used over time. Such an approach could be analogous to what is done for IPv6 addresses in [RFC3041].

5. GRANULARITY OF REDIRECTION

Different multihoming solutions might approach the problem at different layers in the protocol stack. For instance, there have been proposals for a shim layer inside IP as well as transport layer approaches. The former would have the capability to redirect an IP address while the latter might be constrained to only redirect a

single transport connection. This difference might be important when it comes to understanding the security impact.

For instance, premeditated attacks might have quite different impact in the two cases. In an IP-based multihoming solution a successful premeditated redirection could be due to the attacker connecting to a server and claiming to be 'A' which would result in the server retaining some state about 'A' which it received from the attacker. Later, when the real 'A' tries to connect to the server, the existence of this state might mean that 'A' fails to communicate, or that its packets are sent to the attacker. But if the same scenario is applied to a transport-layer approach then the state created due to the attacker would perhaps be limited to the existing transport connection. Thus while this might prevent the real 'A' from connecting to the server while the attacker is connected (if they happen to use the same transport port number) most likely it would not affect 'A's ability to connect after the attacker has disconnected.

A particular aspect of the granularity question is the direction question; will the created state be used for communication in the reverse direction from the direction when it was created? For instance, if the attacker 'X' suspects that 'A' will connect to 'B' in the near future, can X connect to A and claim to be B and have that later make A connect to the attacker instead of to the real B?

Note that transport layer approaches are limited to the set of "transport" protocols that the implementation makes aware of multihoming. In many cases there would be "transport" protocols that

are unknown to the multihoming capability of the system, such as applications built on top of UDP. To understand the impact of the granularity question on the security, one would also need to understand how such applications/protocols would be handled.

A property of transport granularity is that the amount of work performed by a legitimate host is proportional to the number of transport connections it creates that uses the multihoming support, since each such connection would require some multihoming signaling. And the same is true for the attacker. This means that an attacker could presumably do a premeditated attack for all TCP connections to port 80 from A to B, by setting up 65,536 (for all TCP source port

numbers) to the server B and causing B to think those connections should be directed to the attacker and keeping those TCP connections open. Any attempt to make legitimate communication more efficient, e.g., by being able to signal for multiple transport connections at a time, would provide as much relative benefit for an attacker as the legitimate hosts.

> Discussion: Perhaps the key issue is not about the granularity, but about the lifetime of the state that is created? In a transport-layer approach the multihoming state would presumably be destroyed when the transport state is deleted as part of closing the connection. But an IP-layer approach would have to rely on some timeout or garbage collection mechanisms perhaps combined with some new explicit signaling to remove the multihoming state. The coupling between the connection state and multihoming state in the transport-layer approach might make it more expensive for the attacker, since it needs to keep the connections open. Is this the case?

In summary, there are issues we don't yet understand well about granularity and reuse of the multihoming state.

6. MOVEMENT IMPLICATIONS?

In the case when nothing moves around we have a reasonable understanding of the security requirements. Something that is on the path can be a MiTM in today's Internet and a multihoming solution doesn't need to make that aspect any more secure.

But it is more difficult to understand the requirements when hosts are moving around. For instance, a host might be on the path for a short moment in time by driving by an 802.11 hotspot. Would we or

would we not be concerned if such a drive-by (which many call a "time-shifting" attack) would result in the temporarily on-path host being able to act as a MiTM for future communication? Depending on the solution this might be possible by the attacker causing multihoming state to be created in various peer hosts while the

attacker was on the path, and that state remaining in the peers for some time.

The answer to this question doesn't seem to be obvious even in the absence of any new multihoming support. We don't have much experience with hosts moving around that are able to attack things as they move. In Mobile IPv6 [MIPv6] a conservative approach was taken which limits the effect of such drive-by attacks to the maximum lifetime of the binding, which is set to a few minutes.

With multihoming support the issue gets a bit more complicated because we explicitly want to allow a host to be present at multiple locators at the same time, thus there might be a need to distinguish between the host moving between different locators, and the host sending packets with different source locators because it is present at multiple locators without any topological movement.

Note that the multihoming solutions that have been discussed range from such drive-by's being impossible (for instance, due to a strong binding to a separate identifier as in HIP, or due to reliance on the relative security of the DNS for forward plus reverse lookups in NOID), to systems that are first-come/first-serve (WIMP being an example with a separate ID space, a MAST approach with a PBK being an example without a separate ID space) that allow the first host which is using an ID/address to claim that without any time limit.

7. 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 hosts (or two different locators for the same host) 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.

8. 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.

9. ACKNOWLEDGMENTS

This document was originally produced 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].

As the document has evolved the MULTI6 WG participants have contributed to the document. In particular: Masataka Ohta brought up privacy concern related to stable identifiers. The suggestion to discuss transport versus IP granularity was contributed by Marcelo Bagnulo, who also contributed text to Appendix B.

10. REFERENCES

10.1. Normative References

10.2. Informative References

[NSRG] Lear, E., and R. Droms, "What's In A Name: Thoughts from the NSRG", draft-irtf-nsrg-report-10.txt (work in progress), September 2003.

[MIPv6] Johnson, D., C. Perkins, and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

[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-02 (work in progress), December 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.

[RFC3041] T. Narten and Draves, R, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", January 2001.

[DNS-THREATS] Derek Atkins, Rob Austein, "Threat Analysis Of The Domain Name System", RFC 3833, August 2004.

[FYI18] G. Malkin, Ed., "Internet Users' Glossary", August 1996, Also RFC RFC1983.

[ECN] K. Ramakrishnan, S. Floyd, D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

[OWNER] Nikander, P., "Denial-of-Service, Address Ownership, and Early Authentication in the IPv6 World", Security Protocols 9th International Workshop, Cambridge, UK, April 25-27 2001, LNCS 2467, pages 12-26, Springer, 2002.

[RFC1948] Bellovin, S., "Defending Against Sequence Number Attacks", RFC 1948, May 1996.

[PBK] Scott Bradner, Allison Mankin, Jeffrey Schiller, "A Framework for Purpose-Built Keys (PBK)", 9-Jun-03, <draft-bradner-pbk-frame-06.txt>

[NOID] Erik Nordmark, "Multihoming without IP Identifiers", July 2004, <draft-nordmark-multi6-noid-02.txt>

[MAST] D. Crocker, "MULTIPLE ADDRESS SERVICE FOR TRANSPORT (MAST): AN EXTENDED PROPOSAL", September 2003, <draft-crocker-mast-proposal-01.txt>

[HIP] Pekka Nikander, "Considerations on HIP based IPv6 multi-homing", July 2004, <draft-nikander-multi6-hip-01.txt>

[WIMP] Jukka Ylitalo, "Weak Identifier Multihoming Protocol (WIMP)", June 2004, <draft-ylitalo-multi6-wimp-01.txt>

[CBHI] Iljitsch van Beijnum, "Crypto Based Host Identifiers", 2-Feb-04, <draft-van-beijnum-multi6-cbhi-00.txt>

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
email: Tony.Li@tony.li

APPENDIX A: CHANGES SINCE PREVIOUS DRAFT

The following changes have been made since draft-ietf-multi6-threats-00:

- Added more information to section 3.1 based on comments on the mailing list.
- Made stronger statements about privacy from the WG discussion on San Diego.
- Added text about time-shifting attacks i.e. where an attacker was on-path and later moves off the path.

The following changes have been made since draft-nordmark-multi6-threats-02:

- Editorial clarifications based on WG comments.
- Removed the ULP term and it's usage since it was potentially confusing.
- Added a current state of privacy as the basis for "do no harm"
- Added some background information about authentication, authorization, and identifier ownership.
- Improvements to Appendix B

The following changes have been made since draft-nordmark-multi6-threats-01:

- Editorial clarifications based on comments from Brian.

The following changes have been made since draft-nordmark-multi6-threats-00:

- o Added reference to [DNS-THREATS] and clarified that attackers on the path between the host and the DNS servers can redirect traffic today.
- o Added a section on existing packet injection attacks to talk about TCP sequence number guessing etc.
- o Clarified ingress filtering relationship in section on today's flooding attacks.
- o Added a new section on granularity to list some issues about transport-level versus IP-level approaches and what we understand about the differences in security. This is still very much a work in progress.
- o Added a new section on movement to discuss how things change when hosts move around the network. This is still very much a work in progress.
- o Added Appendix B - but this should probably be moved to a different document to keep this document focused on the threats.

APPENDIX B: SOME SECURITY ANALYSIS

When looking at the proposals that have been made for multihoming solutions and the above threats it seems like there are two separable aspects of handling the redirection threats:

- Redirection of existing communication
- Redirection of an identity before any communication

This seems to be related to the fact that there are two different classes of use of identifiers. One use is for:

- Initially establishing communication; looking up a FQDN to find something which is passed to a connect() or sendto() API call.

- Comparing whether a peer entity is the same peer entity as in some previous communication.

- Using the identity of the peer for future communication ("callbacks") in the reverse direction, or to refer to a 3rd party ("referrals").

The other use of identifiers is as part of being able to redirect existing communication to use a different locator.

The first class of use seems to be related to something about the ownership of the identifier; proving the "ownership" of the identifier should be sufficient in order to be authorized to control which locators are used to reach the identifier.

The second class of use seems to be related to something more ephemeral. It seems to be sufficient to be able to prove that the entity is the same as the one that initiated the communication to be able to redirect the existing communication to some other locator. One can view this as proving the ownership of some context, where the context is established around the time when the communication is initiated.

Preventing unauthorized redirection of existing communication can be addressed by a large number of approaches which are based on setting up some form of security material at the beginning of communication, and later using the existence of that material for one end to prove to the other that it remains the same. An example of this is Purpose Built Keys [PBK]. One can envision different approaches for such schemes with different complexity, performance, and resulting

security such as anonymous Diffie-Hellman exchange, the reverse hash chains presented in [WIMP], or even a clear-text token exchanged at the initial communication.

However, the mechanisms for preventing unauthorized use of an identifier can be quite different. One way to prevent premeditated redirection is to simply not introduce a new identifier name space

but instead rely on existing name space(s), a trusted third party, and a sufficiently secure way to access the third party, as in [NOID]. Such an approach relies on the third party (DNS in the case of NOID) as the foundation. In terms of multihoming state creation, in this case premeditated redirection is as easy or as hard as redirecting an IP address today. Essentially this relies on the return-routability check associated with a roundtrip of communication which verifies that the routing system delivers packets to the IP address in question.

Alternatively, one can use the crypto-based identifiers such as in [HIP] or crypto-generated addresses as in [CBHI], which both rely on public-key crypto, to prevent premeditated attacks. In some cases it is also possible to avoid the problem by having (one end of the communication) use ephemeral identifiers as the initiator does in [WIMP]. This avoids premeditated redirection by detecting that some other entity is using the same identifier at the peer and switching to use another ephemeral ID. While the ephemeral identifiers might be problematic when used by applications, for instance due to callbacks or referrals, it is an interesting observation that for the end that has the ephemeral identifier, one can skirt around the premeditated attacks (assuming the solution has a robust way to pick a new identifier when one is in use/stolen).

Assuming the problem can't be skirted around by using ephemeral identifiers there seem to be 3 types of approaches which can be used to establish some form of identity ownership:

- A trusted third party, which states that a given identity is reachable at a given set of locators, so the endpoint reached at one of this locators is the owner of the identity.

- Crypto-based identifiers or crypto-generated addresses where the public/private key pair can be used to prove that the identifier was generated by the node knowing the private key (or by another node whose public key hashes to the same value)

- A static binding, as currently defined in IP, where you trust that the routing system will deliver the packets to the owner of the locator, and since the locator and the identity are one, you prove identity ownership as a sub-product.

A solution would need to combine elements which provide protection against both premeditated and on-going communication redirection. This can be done in several ways, and the current set of proposals do not appear to contain all useful combinations. For instance, the HIP CBID property could be used to prevent premeditated attacks while the WIMP hash chains could be used to prevent on-going redirection. And there are probably other interesting combinations.

A related, but perhaps separate aspect, is whether the solution provides for protection against Man-in-The-Middle attacks with on-path attackers. Some schemes, such as [HIP] and [NOID] do, but given that an on-path attacker can see and modify the data traffic whether or not it can modify the multihoming signaling, this level of protection seems like overkill. Protecting against on-path MiTM for the data traffic can be done separately using IPsec, TLS, etc.

Finally, preventing third party DoS attacks is conceptually simpler; it would suffice to somehow verify that the peer is indeed reachable at the new locator before sending a large number of packets to that locator.

Just as the redirection attacks are conceptually prevented by proving at some level the ownership of the identifier or the ownership of the communication context, third party DoS attacks are conceptually prevented by showing that the endpoint is authorized to use a given locator.

The currently known approaches for showing such authorization are:

- Return routability, that is, if an endpoint is capable of receiving packets at a given locator, it is because he is authorized to do so. This relies to routing being reasonably hard to subvert to deliver packets to the wrong place. While this might be the case when routing protocols are used with reasonable security mechanisms and practices, it isn't the case on a single link where ARP and Neighbor Discovery can be easily spoofed.

- Third trusted party. A third party establishes that a given identity is authorized to use a given set of locators (for instance the DNS)

Intellectual Property Statement

might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

Copyright Statement

Acknowledgment

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