Incentives to Help Stop Floods[1]

Clifford Kahn[2]
EMC Corporation / Atidim Industrial Park, Building 7
Tel Aviv 61580 / ISRAEL
kahn_clifford@emc.com

Abstract

In recent, well-publicized attacks, the attackers first broke into many ill protected Internet hosts, and then used these hosts as unwitting accomplices to flood a major Web site. Various expedient measures, such as blocking UDP packets, will only work until the attacker improves the attack program. In the long run, we need two things. We need suitable machinery in the Internet and suitable incentives upon the users of the Internet. The machinery and the incentives interlock and must be designed together. This calls for a change of mind set. The users we want to incent are really part of the security system we must design. .

The machinery in the Internet will trace a packet flood as close as possible to the origin, and block it there. The incentives will (a) encourage router operators to implement tracing and suppression; (b) encourage host owners to protect hosts from being hijacked; (c) encourage ISPs and others to police their regions of the Internet to enforce adequate protection of hosts.

1 Problem

In recent, well-publicized Distributed Denial-of-Service (DDoS) attacks, the hackers first broke into many ill-protected Internet hosts. They made these hosts into unwitting accomplices, which the hackers call "zombies". Press reports stated that the hackers broke into the zombies by exploiting "well known vulnerabilities".

Then the attackers caused the zombies, all at once, to pelt a victim host with requests of some sort. Either the victim host or nearby network infrastructure was flooded. Legitimate users could not get through to the victim.

The attackers flooded hosts to which they and the zombies had legitimate access. Since they were public Web sites, everyone had legitimate access to them, but a similar flood could be mounted within a protected enclave. Either way, there is no hope of using access restrictions to prevent flooding.

The attackers hid the zombies by using forged source IP addresses. This made it harder to trace the flood back to the zombies. But IP routers can apply address filtering, discarding packets when the source address does not match the wire on which the packet arrived. This will limit forgery quite a bit. Even without address filtering, it is possible to manually trace a flood upstream, router to router, with the help of each router's administrator. There are rumors that such tracing happened in the recent attacks. But it is slow and skilled-labor intensive.

Not many decades ago, telephone calls were traced in much the same way, for law enforcement purposes. Now, tracing of phone calls is automatic and instant. Suppose such automatic and instant tracing of floods were added to the Internet. Then attackers would have to work harder to hide zombies.

Imagine a Million Zombie Flood. The attacker hijacks a million zombies. They are spread all over the Internet, much as guerillas may hide themselves among a civilian population. At the appointed time, each zombie sends two or three legitimate-looking requests to the victim. If we analyze the traffic during the attack, we find that each zombie's behavior looks benign. Each part of the Internet looks benign. We cannot separate the zombies from the legitimate users by traffic analysis during the attack. Not even if we had total access to all traffic that comes out during the attack. Not even probabilistically, and not even in principle.

2 Current Techniques

We survey some of the countermeasures in use today.

New Security Paradigm Workshop 9/00 Ballycotton, Co. Cork, Ireland

[1] This paper was presented at the 2000 New Security Paradigms Workshop. Comments by the referees and the other workshop participants made the paper much better. An exchange with Alec Yasinsac helped us see more clearly how far a technical solution could go and at what point it needed non-technical help.

[2] This paper was presented on very short notice by:

Mary Ellen Zurko
Iris Associates, Inc./ 5 Technology Park Dr.
Westford, MA 01886
Mary_Ellen_Zurko@iris.com

For that we owe her one, if not more.

2.1 Blocking Certain Types of Packets

Routers and transparent firewalls can stop certain types of packets from reaching the victim host. They can do this constantly, not only when an attack happens.

Some well-known attacks involve UDP packets. Many servers offer service through TCP and not through UDP. The HTTP protocol uses TCP, for example. If there is no legitimate need for UDP packets to pass through, then a firewall or router can block them.

Some attacks involve sending an IP multicast to another subnet. Such a packet goes to many hosts. Multicasts are useful for discovering nearby services, such as printers. Multicasts from one subnet to another are not always needed. A firewall or router can block these.

To evade this defense, an attacker must use the same types of packets as legitimate users. For example, an attacker can send a flood of HTTP GET requests to a Web server.

2.2 Blocking Packets by Source Address

The host that creates an IP packet fills in the return IP address, and can put any address it wants. Forged IP addresses make it harder to trace the source of a network-based attack.

IP routers can improve traceability by discarding packets whose source address is impossible given the wire on which the packet arrived. This is *address filtering* or, in CISCO parlance, "access lists". If all routers did correct address filtering, then all source IP addresses would be accurate at least to the subnet level.

Address filtering has a performance cost and a setup cost. Some routers do it and some do not. Further, a rogue router can defeat address filtering.

3 Limits of Cryptography

The author of an influential Distributed Denial of Service program, with pseudonym Mixter, asserted in an interview that the new IP security (IPsec) would prevent DDoS attacks. No it would not, as Bruce Schneier points out. Cryptography cannot help because it takes quite a bit of computing to discover that a cryptographic checksum is incorrect. If the victim does that much computing for each illegitimate packet, the victim will be swamped.[1]

Bruce Schneier has argued that the only solution is to separate control from data signals, as was done for the phone system.[2] But this does not help against the Million Zombie Flood.

4 Other Work on Tracing

Daniels and Spafford[3] consider tracing as a way of catching the attacker. They want to trace many kinds of attacks, not only floods. Because they want to identify the attacker, they find a tension between traceability and anonymity.

In any case, when a zombie is used in the attack, it is very hard to trace past the zombie and find the puppeteer.

Our concern here is not so much to catch the attacker as to stop the attack. For our purposes, the attacker can stay anonymous as long as the attack is stopped. Still, we and Daniels and Spafford share a concern that the tracing system should be efficient. And we share a need to prevent zombies.

IDIP[4] likewise traces many kinds of attacks, not only floods. IDIP does not try to identify the attacker. It does isolate the attacks to limit damage. We will say much more about IDIP.

5 Anti-Flood Systems

One thing about a flood is that the packets keep coming. Since they keep coming, a system can trace upstream toward the source. Floods are far more traceable than many other kinds of attack.

Beyond tracing, an anti-flood system can respond. It can derive attack signatures for the harmful packets. Participating routers and firewalls can discard some or all packets that match the signature.

Deriving attack signatures automatically is not a new idea. It can be found in IDIP (though it is described differently) and in Rephart's and White's work on automatic defense against novel viruses.[5]

A flood signature can specify:

- The IP addresses being flooded
- The IP addresses generating the flood (However, addresses can be spoofed, as noted)
- The IP addresses of network nodes that the flooding traffic is traversing

A signature can use wildcards and other shorthands.

No one component needs to hold the whole signature. It can be distributed, as in IDIP, so long as components get the parts they need.

Such an anti-flood system could do some good, if people would implement it widely. But that is a big "if". We will argue that there are economic incentives for all of the key players – backbone owners, institutions, ISPs.

Suppose it were widely implemented. What about a Million Zombie Flood? If the flood really comes from everywhere, then tracing it is hopeless. The attack signature will accuse all sources and all routes. The anti-flood machinery will discard all packets bound for the targets. Service will be denied as surely as if there were no anti-flood machinery.

Actually, the attack will not come from quite everywhere. Some institutions may protect their hosts against hijacking fairly well. These institutions will not contribute to the flood. The attack signature will not accuse them. The Internet backbones, which (by hypothesis) implement the anti-flood system, will block traffic from other sources but will allow traffic from these well-protected institutions.

To get more protection, we need to get host owners to make their hosts harder to hijack. Again, why would they? We will discuss the incentives for both host owners and traffic carriers presently.

6 The Intruder Detection and Isolation Protocol

This section describes an actual anti-flood system, IDIP. It is an existence proof (or argument) for the anti-flood system described above. IDIP has properties one would want in an anti-flood system. But IDIP has the limitations we claimed all anti-flood systems must have, when facing a Million Zombie Flood.

6.1 IDIP in General

IDIP traces attacks through the network topology, and blocks them at nodes in the topology.

> (1) When an attack traverses an IDIP-protected network, each IDIP node along the path is responsible for auditing the connection or datagram stream; (2) when a component detects an intrusion attempt, the detector distributes an attack report to its neighbors who can then help trace the attack path and respond to the intrusion; and (3) these neighbors further distribute the attack description along the path of the attack.[6]

Some IDIP nodes are *boundary controllers* (BCs). They can be a router, a firewall, or other things. They do the blocking.

A node **n** and a BC **b** are *neighbors* if they can send one another IP packets that do not pass through another BC **c**; **b?c** and **c?n**. In this definition, **n** may be a BC or a non-BC. Neighbor-ness is almost entirely a matter of physical network topology.

If two non-BCs can send one another IP packets that do not pass through a BC, then the two non-BCs are considered to be in the same *IDIP domain*. So by design BCs form the boundary of a domain.

IDIP will work better if routers suppress IP source routing. Source routing lets the originator of a packet control its route through the Internet. It is rarely needed. It would let the attacker send its attack through a great many routes, making it much harder to trace the flood.

6.2 IDIP against Floods

We want to use IDIP to fight flooding. The IDIP developers have done so, and report that "it actually works."[7]

The basic message could be "I am seeing a flood for IP address xx.xx.xx.xx." The victim or an intrusion detection system (IDS) could pass this message to its neighboring BCs. Each of them could look to see if it too was seeing a flood for that IP address. If so, the BC could (a) begin discarding all or most packets bound for that address and (b) send the IDIP message on to its own neighbors in turn. Once a BC stopped seeing a flood for IP address xx.xx.xx.xx, it would stop discarding packets for that address. This would restore service.

A victim or an IDS watching all traffic to the victim can tell whether the victim is getting too much traffic. It hardly matters whether the excess is from malice or not. Either way, discarding some traffic will not hurt.

For a BC it is harder. A BC must check whether the flood is coming through it, wholly or partly. How will it check this? If the flood comes over multiple routes, then each BC has subtle evidence to consider. False positives are harmful because they deny service to some users. False negatives are harmful because they let the flood continue.

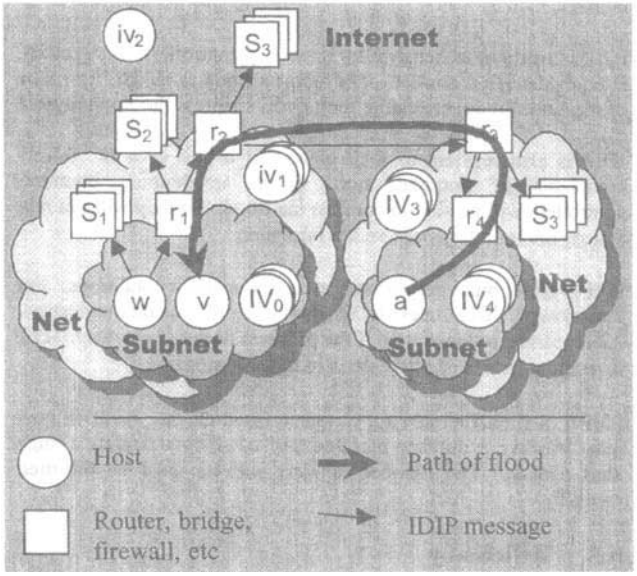
IDIP reduces the chance of a false positive because the BC will not do anything unless the victim or an IDS monitoring the victim complains. But the problem remains.

The BC may watch for an attack signature. For example, it may look for more than x packets per minute bound for a single IP address. Or it may look for more than x connection requests (SYNs) per minute for a single IP address.

And the BC may watch for anomalies, deviations from past behavior. For example, it may look for a packets-per-minute value for a single IP address that is more than two standard deviations above the mean. It would have to observe packet frequencies constantly, which would have overhead.

6.3 Example

The figure represents a possible configuration of IDIP and a possible attack. The attacker **a** is flooding the victim **v**. The flood is taking just one route through the network, passing through BCs r_4, r_3, r_2, and r_1. They are probably routers. IV_0 – IV_4 is each a set of indirect victims – those who cannot communicate with **v** because of the attack.



The intrusion detector **w** (for "watcher") notices the flood. **w** can be part of **v** or a separate program or host. **w** notifies its neighboring BCs, r_1 and S_1. S_1 is a set of BCs.

The members of S_1 begin to monitor their own traffic, but do not find a flood of packets for **v**. r_1 does find a flood of packets for **v** and begins to squelch them. This restores service for users IV_0. It also restores service for the members of IV_1 who reach **v** through S_1 rather than through r_1.

r_1 alerts its neighboring BCs r_2 and S_2. The process repeats. When r_2 squelches the flood, r_1 notices that it is no longer getting the flood and stops squelching traffic for **v**. This restores service to the remaining members of IV_1. It also restores service for the members of IV_2 who reach **v** through S_2 rather than through r_2.

In the same way, r_3 and $r_{4\,in}$ turn come to squelch traffic bound for **v**. Only IV_4 remain as indirect victims. They are in the same IDIP domain as the attacker. IDIP cannot help them.

This is only an example. A real attack may take several routes. IDIP will trace all of them. But it will be harder for a boundary controller to decide whether it is seeing flooding or not.

Also, the example assumes boundary controllers between every network and the general Internet, such as r_2 and r_3. All of these boundary controllers alert each other to floods. This approach does not scale. There are too many networks. The number of packets and the configuration labor grow as the square of the number of boundary controllers that are all neighbors of each other.

To solve the scaling problem, backbone routers in the general Internet would probably have to become boundary controllers. Whether that is likely we do not know.

6.4 IDIP's Robustness

When a security mechanism is proposed, one must ask how it could be turned against its owners, or otherwise make things worse.

IDIP is resilient to betrayal by boundary controllers. A rogue BC is equivalent to a non-BC. If I report a flood to the BC upstream of me, and if it does nothing, then I will suppress the flood myself.

IDIP is also resilient to false alerts. If a host or a BC falsely complains of flooding, none of its neighboring boundary controllers will be able to confirm the flood. So they will not take action, nor will they pass on the false alert.

IDIP's alerts must get through even under flood conditions. For that reason IDIP uses UDP (connectionless) communication. Alerts can be retransmitted if the problem does not go away. IDIP is resilient to loss of some alerts in transmission.[8]

IDIP's own traffic must not worsen a flood much. An IDIP node must observe a number of flood packets before sending a single alert packet. The number of alert packets must be throttled carefully.

6.5 Efficiency

IDIP appears to be efficient. Boundary controllers need not check packets except during a flood. Actually, they also must check packets during an accidental false alarm or an intentional false alarm. Whether this can be a problem warrants study.

Also, we understand that one can build simple filters that will keep up with 100 base T speeds. [9]

For some other work on preserving efficiency while adding security features to the routing infrastructure, see Timmerman.[10]

6.6 What It Solves and What It Doesn't

With an anti-flood protocol such as IDIP deployed, floods launched from a few places can be countered with only a little harm to innocent bystanders.

This is true where subnet boundaries correspond to useful functional boundaries. But if the attacker happens to share a subnet with unrelated and important functions, then innocent bystanders suffer more.

Now consider a Million Zombie Flood. Maybe some IDIP domains suffer hijacking and contribute to the flood. Maybe others do not. For example, a well-run company network might not.

Suppose I am in an innocent IDIP domain. Suppose your server is the flood victim, and is in an IDIP domain. Suppose the Internet backbone routers that let us communicate also implement IDIP.

Then I should be able to use your server despite the flood. The routers will not let flood-contributing IDIP domains send packets to your server. But they will let my IDIP domain send packets to your server.

Thus IDIP, if deployed this extensively, could preserve service for some clients.

But this is not a satisfying result. We need to increase that to "most clients". Next we consider the non-technical side—the incentives that may get people to do their part in implementing the anti-flood system.

7 Zombie Prevention

To stop the Million Zombie Flood we must make it much harder to hijack zombies.

Our premise is that if hosts used well known cures to well known vulnerabilities, then they would be much harder to hijack and the Million Zombie Flood would be much more expensive to mount. This premise may be true or false, but we will proceed from it to see where it leads.

A great challenge is to induce everyone to protect their hosts.

7.1 Non-Economic Approaches

We could try a "protect your host" campaign, similar to "please don't litter" campaigns. It might help. It clearly will not help enough.

Zombie owners could be held civilly liable for allowing their hosts to be used in an attack. But who can sue a million zombie owners, or even a hundred?

We could have government regulations, backed up by criminal penalties. The government could issue licenses to be on the Internet. Just as your car must have working headlights to be on the road, your computer could be required to have certain protections. But the Internet is worldwide. It would be very hard to get uniform enforcement of such standards across the globe. And do we want government regulations in this matter? Would

they be the right regulations, and would they adapt quickly enough to changes of technology? We doubt it.

7.2 Economic Incentives

Assume that an anti-flood system such as IDIP were in place on much of the Internet. Such a protocol would trace a flood upstream. Anti-flood participants upstream would block traffic bound for the flood's destination. "Destination" could be a single IP address, a net, a subnet, or other unit, depending on the anti-flood system.

Internet Service Providers (ISPs) will have an incentive to police their subscribers, or to police them better. Consider an ISP that is not participating in the anti-flood system and not policing its subscribers. Whenever some of its subscribers flood a victim, the anti-flood system will trace the flood to the ISP and block it from sending packets to the victim. All of the other subscribers will suffer. They will not like this, so that is the ISP's incentive. The ISP could run scans for well-known vulnerabilities, and could insist that subscribers fix these vulnerabilities.

Some ISPs will do this policing, and some will not. A user will have to choose which kind of ISP to subscribe to.

Companies and organizations will likewise have an incentive to make sure that their machines are not used as zombies.

In effect, the consequences of neglect (allowing hosts to be hijacked) are pushed closer to the neglector. In economic parlance, this is internalizing externalities.

Now some areas of the Internet will be well policed and suffer few flooding attacks. Other areas will be unpoliced and will suffer many of them.

8 Incentives to Join an Anti-Flood System

What would motivate a router operator to implement an anti-flood protocol?

It helps that there is incremental value as more nodes support IDIP. The farther you can trace an attack, the more selective can be your blocking. In the example, if r_4 did not support IDIP, then V_3 would continue to suffer, but others would not.

Communication providers, and anyone who runs a router, will be motivated to offer the anti-flood system as a quality-of-service feature. It is valuable to those downstream of the router – companies, ISPs, other routers, etc – who may be flooded. They will be willing to pay for this protection.

Suppose I am choosing between two Internet backbones, and one of them has anti-flood machinery while the other does not. If I want to be able to get to www.etrade.com even when someone is flooding it, I had better choose the backbone with anti-flood machinery. There is a clear incentive both to me and to the backbone operator.

If a backbone is not ready to implement the anti-flood protocol, it can at least implement address filtering. Then its subscribers will be able to tell each others' packets apart. So if some subscribers make a flood, the target subscriber can discard packets from the source subscribers. If the source subscribers participate in the anti-flood system, the target subscriber can also send "I am seeing a flood for xx.xx.xx.xx" messages to them, and they can trace the flood internally. Thus the anti-flood system can bridge over the backbone. All of this works only if the backbone itself can withstand the flood.

9 Workshop Discussion

Some points from the discussion at 2000 NSPW, with our replies.

Q: IDIP is a tool that an attacker can use to partition the network into small pieces in an automated fashion. The second step of this attack would be to start killing the individual partitions during the blackout period in which they cannot warn or help each other.

A: If the attacker can mount floods from everywhere in every direction, then indeed an anti-flood system will have the effect of partitioning the network. Our thrust is to make it harder to mount floods. But if the attacker can mount such a flood, the question is whether the anti-flood system makes things better or worse. An anti-flood system can be provoked to suppress warnings and help. But a flooded target also cannot receive warnings or help, at least not reliably. And an anti-flood system does not have to suppress all traffic. The IDIP implementers have worked with throttling rather than halting suspect traffic. This would allow some warnings or help to get through, just as they could if there were no anti-flood system. The issue needs more analysis. But on the face of things, an anti-flood system need not make the problem worse.

Q: Subnet boundaries do not always correspond to useful real-world boundaries.

A: True; the physical topology does not always correspond closely to the human organization. That limits the value of this approach somewhat.

Q: It's not likely that Internet backbones will implement an anti-flood system.

A: We (now) argue that they have incentives to. We also argue that if backbones implement address filtering it will help some.

Q: Some neutral parties are too weak to do anything effective against adversaries.

A: Yes. They may lack the skill to protect their hosts, or their whole way of doing business may require open doors. So these hosts may become zombies. The anti-flood system will block the attack, and anyone in their part of the network may become an indirect victim.

Q: Some ISPs already do a pretty good job of policing.

A: Good point. And with the new incentives they may do an even better job.

Q: Policing is not very realistic yet. The Internet is not evolved enough. The Internet is like the Wild West – everyone needs to carry a gun.

A: Flooding attacks a community resource, a commons. Only a community response will do any good. People can carry guns and still work together to protect their communities.

Q: Will the covenant model proposed here really work? If I buy a house I can tell whether the grass is mowed. If I buy a computer I cannot tell whether the system is secure, or whether the system next door is secure. How can I decide whether to sign up with an ISP that polices well, when I cannot tell? How can economic incentives work in an environment completely devoid of information?

A: Consumers need to factor the danger of floods into their choices. If one ISP is always suffering from floods by its own subscribers, and another is not, then this is a hard fact. If the consumer can get at this hard fact, then she can factor it into her choice of ISP. Perhaps disclosure laws are needed. (No more than .003 floods per serving ...)

Q: We don't leave doors locked all the time or unlocked all the time. We unlock them when we need to go in and out. But firewalls do not work like that.

A: Maybe when a program expects a lot of legitimate traffic from point a to point b, the program should give the anti-flood system notice – a heads up. That way, the traffic will not be mistaken for a flood. The notice up is like unlocking a door. The notice would need to be authenticated and authorized.

Q: For some applications multicast is important.

A: True. So it must not be disabled when those applications are in use.

Q: A router once melted down because of an outbound Distributed Denial of Service attack, effectively denying service to the *attacker's* net.

A: Here, tracing and blocking the flood may not help, but zombie prevention will help.

10 Conclusion

Things are not all that hopeless.

A potential victim of flooding cannot do much to protect itself. Protection will require infrastructure changes and action by many.

Since we must rely on economic incentives, zombie protection and flooding protection will be uneven. But each participant in the Internet has self-interest in doing its part to prevent floods.

Some areas of the Internet will be unpoliced. An attacker will still be able to hijack many zombies in those areas. When they do, and mount a flood, those unpoliced areas will be cut off from sending packets to the flood's victim. Users in those areas will suffer. But the well-policed areas will be fine.

[1] Bruce Schneier. Distributed Denial-of-Service Attacks. In *CRYPTO-GRAM*, February 15, 2000. http://www.counterpane.com/crypto-gram-0002.html

[2] Bruce Schneier, same place.

[3] Thomas E. Daniels & Eugene H. Spafford. Network Traffic Tracking Systems: Folly in the Large? In *Proceedings of New Security Paradigms Workshop 2000, Ballycotton*, Ireland, September 2000.

[4] Dan Schnackenberg, Kelly Djahandari and Dan Sterne. Infrastructure for Intrusion Detection and Response. In Proceedings of the DARPA Information Survivability Conference & Exposition Volume II of II. 1998. http://www.nai.com/nai_labs/media/pdf/DISCEX-IDR-Infrastructure.pdf

[5] Jeffrey O. Kephart, Gregory B. Sorkin, Morton Swimmer, and Steve R. White. Blueprint for a Computer Immune System. Presented at the Virus Bulletin International Conference in San Francisco, California, 1997. http://www.av.ibm.com/InsideTheLab/Bookshelf/ScientificPapers/Kephart/VB97/

[6] Dan Schnackenberg et al. Same place.

[7] Dan Schnackenberg, personal communication, 8/18/00

[8] Dan Schnackenberg, personal communication, 8/18/00

[9] John McHugh. Remark at 2000 New Security Paradigms Workshop, during discussion of this paper.

[10] Brenda Timmerman. Secure Dynamic Adaptive Traffic Masking. In *Proceedings of New Security Paradigms Workshop 1999,* Caledon Hills, Ontario, Canada, September 1999.