The COOK Report on Internet

Poorly Understood DDoS Attacks Reveal Internet's Vulnerability to Disruption

No Consensus as to Solution -- We Present Hypothesis That Analysis of What Happened May Be Faulty -- Paradigmatic Shift in Understanding of Internet Mechanics Outlined by Ed Gerck

During the second week of February the largest, and most diverse denial of service attacks in the history of the Internet caught several of the most important commercial web sites off guard and exposed an previously unsuspected operational vulnerability that affects the entire commercial Internet. We contend that the official view of what happened as demonstrated by Gene Spafford's statement in the next paragraph provides a misleading and superficial explanation of what is really a far more subtle and difficult structural weakness inherent in the basic structure of the Internet. The long article that follows will demonstrate that Internet backbone engineers are by no means agreed on precisely what happened or on how to deal with it.

The follow up interview with and essays by Ed Gerck, co-founder of the Meta Certificate Group, form the basis for a fresh and compelling analysis of what we may really be dealing with. We conclude that there is a possibility that the fundamental nature of the attacks may have been completely misunderstood. We also contend that Gerck's theories, published here for the first time, may provide an entirely different mathematical basis for understanding the Internet as a quantum information structure possessing significantly different capabilities and potentials than could be extrapolated from our current understanding. As a part of the preparation of this report we have obtained enthusiastic evaluations of his ideas from four senior internet figures whose technical knowledge surpasses our own. They are all agreed that Gerck is suggesting a potentially very important new calculus for the Internet. In view of the impressions he is making, it is likely that research will be rapidly undertaken to ascertain if his own experimental results from1998 are verifiable and reproducible. Without a correct diagnosis of our current problems, we will be unlikely to find solutions. As a result, the Internet's behavior of early February will become more rather than less commonplace.

The Official Point of View

Security expert Gene Spafford described the distributed denial of service incident on February 19: "Last week (ca. 2/8/00), a massive distributed denial of service attack was committed against a number of Internet businesses, including e-Bay, Yahoo, Amazon.com, and others. This was accomplished by breaking into hundreds (thousands?) of poorly-secured machines around the net and installing packet generation "slave" programs. These programs respond by remote control to send packets of various types to target hosts on the network. The resulting flood effectively shut those target systems out of normal operation for periods ranging up to several hours. The press jumped all over this as if it was something terribly new (it isn't — experienced security researchers have known about this kind of problem for many years) and awful (it can be, but wasn't as bad as they make it out to be)."

In our opinion Professor Spafford's introduction to his summary of the White House meeting of February 15 may be somewhat more reassuring than the situation warrants. While analyses of tribe and trin00 posted in December 1999 show that security experts knew that such attack were possible, to the best of our knowledge, if they also knew that it could be carried out with such sweeping scale and scope - with the tracks of the perpetrators very well covered, they were very careful not to say so in public.

As one would expect, discussion on network mail lists (NANOG, Inet-Access, and IETF) was intense. The implications of these discussions for electronic commerce are ominous. There is some discussion that adequate defense against this kind of attacks will be impossible because of the basic architectural structure of the internet.

Volume IX, No. 1, April 2000
ISSN 1071 - 6327

With the hype that has propelled the growth of electronic commerce, the stability of 24 by 7 access to the largest portals and web sites has never been seriously questioned. We have evidence that such assumptions were unwarranted. Therefore, if instead of resting on the solid foundation of the famed "five nines" of reliability of the PSTN, electronic commerce on the Internet rests on feet of clay, we are surely in for rough times ahead. Another problem, according to the views of Ed Gerck presented in Part Two of this issue, major Internet e-commerce web sites are likely to be subject to unpredictable performance breakdowns for the foreseeable future. Such breakdowns will be perturbations in server behavior that may be triggered by DDos Attacks, and even by viruses or software bugs.

Part One of this month's COOK Report presents a running summary of the analysis of the Internet's engineering community as ex-

On the Inside:

pressed in intense discussions during the days of and following the attacks. For the advocates of electronic commerce on the grand scale of Amazon and Yahoo the ominous message is that, while the net engineers understand what happened, for the time being at least, there is no consensus as to the precise nature of the attacks. Furthermore, given the distributed and non uniformly controlled structure of the network, this kind of attack may be impossible to prevent in the future.

Consider for example the following frustrated complaint: "Eight Linux machines at Syracuse University were hacked into and used to generate packets. We might not even have noticed except that after attacking other Internet sites they decided to flood our campus network with packets, shutting down our access to the Internet. FBI was on campus Thursday afternoon. Anybody on this list an expert on Linux security? One of my research machines got shut down and, until we can figure out how to reinstall Linux in a more secure way, the Net administrators won't let me turn it back on."

While even greater security precautions will be taken, and doors will be closed, three weeks after the attacks began, no one seems to know precisely how they were initiated, structured and controlled, let alone who was behind them, nor even how many machines were infected and may remain so today.

A close reading of the technical discussion that follows will show that short of policing every fixed IP address server on the internet, not just inside the United States but also globally, there is no security against this kind of attack. The best engineering and security minds in the Internet can not ascertain the precise nature of the attack used. They know the general kinds, (trinoo, tribe, smurf, flood etc) but they know neither the extent, nor the identities of the launching machines. Most important they have no easy way of identifying the number of compromised machines that were not used in these attacks but could be used in future attacks.

Also while many experts have ideas as to how security may be tightened. One of the problems is that security needs to be tightened everywhere. Machines that were likely involved run the gamut from commercial machines with high security to machines at universities and colleges, and personal machines with static IP addresses connected by xDSL or cable modems. The sheer variety of kinds of hosts co-opted means that a security plan based on policing millions of hosts is impossible. While this is being thoroughly hushed up, its reality should be obvious to any technically literate reader of the 20,000 words that follow. The technical community has some ideas that are little better than band aides, in contrast to the reassurances of Professor Spafford from the White House meeting on Feb 15 that we are in control of the situation.

We need to do some fundamental reassessment of what security on the Internet means. Thinking along these lines, we wrote to Ed Gerck on February 20: "stitching together all the technical discussion on the DDoS attacks helps me wonder whether, if all systems are compromised, or at least we must assume that all systems are compromised, then we must change from building higher and higher firewalls to deter outside intruders from making raids on our hosts and data to systems where our intent is to monitor the traffic sent to our servers and sort that traffic into hostile, or friendly or uncertain intent. And should we do this by using software that monitors whether or not that traffic knocking on our "doors" can pass the barriers of trust worthiness that we may set for it?"

Gerck replied: "First, this was not a DDoS — this was a CDoS. A Coherent Denial of Service attack. The difference is that a distributed but incoherent attack would not have done any major harm. In order to explain how such an attack was possible and why it was effective, one needs to understand first that nothing is coherent in the Internet — all packets travel in what may seem to be a random fashion, each host has unsynchronized time and even the path traveled by each packet is non-deterministic. Thus, achieving a stream of packets arriving in coherent fashion at one location, from a large number of coordinated locations is a feat. How this was done is what should be asked here — how such coherency could be achieved, in a network based on incoherency? The answer is that I have demonstrated that the Internet can sustain coherent traffic *amplification* after a certain critical number of hosts is reached — in which case a process called "stimulated emission" wins over "spontaneous emission". In other words, the Internet becomes closer to a quantum system. I posit that this number was reached in those attacks, which shows that the Internet will start to show this behavior more and more as it expands to more hosts."

Gerck makes some bold assertions to which we shall return in Part Two of this issue of ***The COOK Report.***. In the meantime as people act out of desperation to control what is essentially an uncontrollable situation, they are likely to undertake many strict "security" measures. We are concerned that the Clinton administration may use these newly apparent weaknesses of the Internet as an excuse to implement programs of control. So far it looks as though the administration has resisted these entreaties. We do not however trust it toward the long term future because there are too many powerful interests wanting to move in positions where they can increase there power by alleging that they can control something that is uncontrollable. Given the distributed and non regimented nature of internet technology, it seems very likely that the Internet in the next year or so will not ever attain the reliability and stability of the PSTN.

Looking at a horizon more than one year out it is possible that software may be developed that can ameliorate some of the present problems. We are also beginning to understand ideas for the development of real time bandwidth markets, where tools might enable large internet sites to defend themselves against denial of service attacks by changing their bandwidth provisioning on the fly. We will begin to explore these ideas in the May issue of the COOK Report. In the meantime we begin this issue with the following narrative of the reaction of many of the internet's technical and operational experts to the attacks.

The Beginning — The "Yahoo Lessons Learned" Thread from NANOG

Reporting of problems began with **David Glynn's** note to the inet-access list at 13:57 pm on Monday February 7. "Looks like traces to Yahoo die at pos7-0622M.cr2.SNV.gblx.net (206.132.151.22) from here, and I see on CNBC that Yahoo has confirmed as of 2PM CT that they've been down for an hour. What's everyone else seeing? Glynn asked. Is it just Gerbilcenter [Editor: Frontier Global center] or are others seeing more interesting angles from different perspectives? Any other sites taking a hit from the same problems? Maybe at least one of those reporters on the list will generate a story I can point our users to, so they won't blame us for some screw-up out on the coast. ;)

Then an hour later, according to CNET: "We haven't identified the problem, but the network has been down intermittently for the last hour and a half," Shannon **Stubo**, a Yahoo spokeswoman, said early this afternoon. "Our engineers are working vigorously to identify and remedy the situation." Stubo added.

Stubo said the outage began at 10:35 a.m. PST. At about 1:30 p.m., some users were again able to access the site, but it was unclear if the problem had been corrected. According to one Internet performance measurement firm, Keynote Systems, the problem primarily affected users in the United States. Keynote analyst Dan Todd said Yahoo was inaccessible within the United States, but was only 59 percent accessible internationally," CNET concluded.

Then early on February 8 Sean Donelan asked on NANOG: Was there something Yahoo!, Global Center or other providers could have done, either individually or in cooperation, to prevent the problem? Likewise, could they, individually or in cooperation with other providers, have shortened the duration or severity by doing something different? And finally, would they be more successful in tracking the source of the problem by doing something different?

K. Graham: One of the emails sent in, mentioned that a network they work with or for was being utilized as an amplifier It was mentioned that this was a co-ordinated attack. That meant a bit of planning and access to various machines. As to the number of attackers, only Yahoo's internal people may know. Even then it may have only been one individual with a script that accessed many locations at one time and initiated the commands. There is the ability to do such an attack.

Senie: Please refer to RFC2644/BCP34 on the subject of directed broadcasts. This RFC recommends router vendors disable directed broadcasts by default. It also recommends ISPs disable directed broadcast on ALL routers. In light of the recent events, it would be good to see a concerted effort made by everyone to ensure this has been done.

Kai Schlichting I recall that SprintLink had some plans to put ingress (and egress?) filters on all interfaces facing dedicated customers that were not multi-homed. This came after realization that education of the end-user was a fruitless and herculian task: Network smarts are virtually non-existent in IT departments, and even loads of smaller ISPs everywhere. Whatever became of this project ? At what traffic level (across the entire box) do Cisco 7{0;2;5}00 routers with RSP{2;4} cards fall over and die because of CPU load?

K. Graham: the reality of "stay connected 24/7" at the household level with high speed internet, makes the possibility of this attack more of a multi level victim attack. Home users do not know that they are leaving the door open to exploitation with simple Window's shares. Savvy people gain access to the cable and DSL modem user's PCs and then launch their attacks. Small utilities are put in place to make it easier to find the exploited machines. Thus creating a network of available attack, harder to track connections.

Education is a tool that can be used to inform customers. If each node on the Internet takes care of it's own doors then there will be less available launching pads. Thus making it a bit simpler to track an attack. Who or what will do the education is a question. Who are the responsible parties if no education is taken or given? To me, the responsibility question is a nightmare at best.

Senie: Of course as Paul has mentioned, we wrote RFC 2267 several years ago to address this very issue. I strongly encourage folks to take a hard look at ingress filtering. Hardware vendors have implemented features in dialup servers and routers which can help.

Kai Schlichting Without wanting to bash my favorite NAS vendor: I have asked for 'IP verify unicast reverse path' in their boxes as much as 2+ years ago. They recently admitted to having no record of this request, and it has just now become a request for engineering. Vendors do not have their focus on security, just like most everyone else in the Internet "industry". Skating on thin ice has a price...

Senie: While implementing these measures may not directly benefit your network, doing so may thwart an attack against someone else's net. Tomorrow, the roles could be reversed. As with many areas of managing the Internet, cooperation is key.

Kai Schlichting; Like the kind of cooperation that is making people close their open SMTP relays voluntarily because closed relays are A Good Thing <tm> or are a BCP? That always and only worked with threats of loss of connectivity or humiliation through public exposure. Some networks have taken it upon themselves to shield their customers from such well-deserved scrutiny from the outside (Hi Dave!).

Nothing will change until Yahoo decides that the legitimate operators of the Trinoo/Tribe/whatever slaves have acted with reckless neglect by not keeping their system secured with vendor-issued patches. But when they do, duck and cover for the wave of lawyers hitting like an Ion-storm.

Vadim Antonov: Yep. Actually, tier-1 ISPs can write the requirement for reverse-path source IP address verification on customer access circuits into their peering agreements. An enforcement can take a form of penalties per verified incident of forged source address attack originating in peer's network. (The adversarial IP prefix filtering was needed to institute such prefix-reduction policies as aggregation and address allocation out of ISP blocks. I remember that purely voluntary efforts were pretty much derailed by negligence of some ISPs (why AS 174 comes to the mind? :) I do not expect reverse path filtering to be any different in terms of deployment problems.) The DoS prevention functions (not letting directed broadcast in, and not letting forged addresses out) should be done at provider's side.

Dan Hollis: Unfortunately I suspect its going to take some high profile lawsuits before this gets widely enough deployed by providers to be effective. There just isn't the financial incentive for providers to be bothered with it, so its going to have to end up being a legal liability if they don't, before they will take action.

Really, I think things like RPF and other *basic* filters should be a contractual requirement before allowing customers to connect to the network. Hell, I'm thinking Cisco and others should make it a *default*. ;)

Andrew Brown (responding to Antonov's remark that The DoS prevention functions (not letting directed bcast in, and not letting forged addresses out) should be done at provider's side.) "nope, won't work. well...it might, but you also might find very irate customers jumping up and down screaming about the filtering. The provider simply cannot know what is and what is not a broadcast address, simply because the customer gets to set up his own networks. I, for one, am using what is "technically" a broadcast address as a unicast address (think point to point). Others may be doing the same. just because an address is an one end or another of a cidr block (or c or b block), doesn't mean that it's broadcast.

Daniel Senie: You're correct. Directed broadcast can only be properly identified in the equipment on the specific subnet. In other words, EVERYONE has to fix this, from end users to ISPs.

To Vadim's main point, though, where to place protections: the answer I normally give to clients (whether ISPs or end users) is do it everywhere. There's no reason NOT to filter the egress from a corporate network, and then at the provider side filter the ingress from that same corporate network. There is plenty of router gear which can handle the needed filtering.

Brown: right, and there *are* things that providers *can* do in the way of egress filtering. for customers that are either (a) not multi-homed or (b) not providing transit to their peers, they can do source filtering.

Senie; Dialup pools should also be protected. No sense in permitting problems to originate on a dialup modem or ISDN line. I know the Lucent/Ascend MAX product accepts an attribute Ascend-Source-IP-Check, which can be applied as a part of the RADIUS authentication. Have the large dialup wholesalers implemented this?

Brown: probably not. I'd be willing to be that a majority of the equipment that's in use these days for providing dialup service doesn't have that sort of capability. one simple "cure", then, would be to place some-

thing (a packet pilfering router) in between the dialup servers and *their* means of internet connectivity. a small separate lan for your dialup pool.

Senie: There'll be no magic cure for this issue. It will take a lot of measures from everyone. [Editor's Note: This refrain pops up again and again in these discussions. The Internet is simply not well equipped to deal with these attacks.]

Brown:Yep

Senie: The wholesalers are allowing (requiring?) filters be added to block port 25 to all but the retail ISP's mail servers.

Dan Hollis: when you're not their customer they don't have much incentive to lift a finger to stop denial of service attacks. Its also the excuse they gave me why they couldn't be bothered to disable directed broadcasts, by the way. "We don't have enough CPU to filter them.

Chris Cappuccio: Funny. On an as5300 with compression turned on, and 96 56K users dialed up and active, I've never seen the CPU load go above 15%

Brandon Ross: And we have anti-spoofing filters on for all of the 3com TC's that we own, and we don't see any performance hit.

Dan Hollis: Maybe its time for a spoofing-source registry similar to the smurf-amp registry. Eg networks which allow spoofing to originate from them

Jared Mauch: [claiming absence of CPU capability] is a total shield. These days if anyone claims that, they either don't know how to manage their equipment, or have other serious issues. The only exceptions would be people who are entirely at the OC-n speed, and then it gets more difficult to filter. Everyone comes down to at least 100M/sec (I guess, unless they're talking gig e) and more likely down to 45M or 10M at some point. It's not that difficult to filter traffic. The problem becomes deploying it in an existing infrastructure. You don't want to break your existing customers. That's why it can sometimes take a few days to shut down an open relay. You have to determine who is allowed to use it and who is not. There is no excuse for directed broadcasts these days though.

Hollis: I think all the tier1 networks need to seriously clean out the complacent dead wood and dust off the clue by four.

Mauch: I agree, but I also understand that the job is not quite as simple as you state. I'm sure it would take a group of people a day or two to just do a single POP at a large provider. Many people would be easy to take care of because they have a single t1 or something, but once you're multihomed things become extremely painful. My rule of thumb is that if you're not speaking BGP though, you can source filter easily, using the existing Cisco knobs. (With your customer that is). I recommend that the contracts which the tier 1 providers write require that the people who they provide access to run a secure network, and list a 'security contact' before they will turn on services. It's fairly simple.

Hollis (regarding Senie's statement that Dialup pools should also be protected). No sense in permitting problems to originate on a dialup modem or ISDN line. I know the Lucent/Ascend MAX product accepts an attribute Ascend-Source-IP-Check, which can be applied as a part of the RADIUS authentication. Have the large dialup wholesalers implemented this?: When I asked a couple dialup wholesalers this question point blank last year, the answer was no - because their routers/term servers didn't have enough CPU to do filtering.

Randy Bush: I am rather amused at folk who fear dialup systems being used as distributed denial of service (ddos) slaves.

Brown: It rather depends on whether you consider DSL to be dialup. And they don't have to be slaves. I could be a ddos *controller* over a 2400 baud modem.

List user: An ascend TNT makes a rather nice smurf amp

Dan Hollis: funny... there is a "Forward Directed Broadcast" option in the config. Turning this off does seem to work. Of course that doesn't mean folks HAVE turned it off..

Senie: Yup, makes you wonder why they haven't done it. Doubly confusing if ascend hasn't made it default. Which term server vendors have RPF?

Mauch: I believe you can do it with cisco and USR TC, along with the mentioned Ascend.

Robert: [With dial up systems] I'm more worried about a master being connected there. Remember, with at least one of the tools you can trigger the "slaves" via forged ICMP reply messages. It doesn't take a fat pipe to do that and it makes finding the perpetrator that much harder, especially since dial connections are generally more anonymous.

Yes, we have tested "source validation" in our live dial network. Yes, there is a performance impact. "Can do" or "Can't do" depends on how many dial customers you are trying to pile into one box, and what equipment you are using. Also, ingress filtering one-hop-up isn't necessarily so easy. Some of us will dynamically route prefixes other than /32 to certain dial customers. This complicates things.

Bush: and worse, sometimes one does not have control over the cpe, and the next hop, the pop aggragation box, is getting highly aggregated telco with hundreds of dedicated customers per physical interface. hence one can run into the not-enough-horses-to-packet-filter condition on the first level aggregation.

Joe Shaw: As far as the directed broadcast option being left on, ignorance is usually the problem here.

Senie: To be fair, Router Requirements (RFC 1812) required directed broadcast be on by default. RFC 2644/BCP 34 was only published in August 1999, changing the requirement to be OFF by default.

Some router vendors, notably Cisco and Proteon (a.k.a. OpenROUTE, now NxNetworks) made the change prior to that new BCP, recognizing that melting down the Internet was the logical result of leaving directed broadcast turned on.

Bryan Bradsby: I put the emphasis back on the server admins. Security patches were readily available on the Sun site. Ignoring applicable security patches for months is likely to get you hacked and abused on todays net.

Wayne Bouchard (on February 10): Yes.. and new patches appear each and every week. Do YOU want to schedule reboots for 80 some servers on a weekly basis? *IF* you get approval for such frequent reboots, you still have the problem of the administrative nightmare. Especially if you've made custom modifications to the systems and have to be careful exactly which patches you apply instead of doing a blanket install.

Now, from the other end of this, this is no excuse not to keep your servers up to date. You may just end up checking it, say, monthly instead of weekly. [But] how do you go about doing a packet trace on routers passing gigabits of traffic every second without killing the router/network and actually get usefull information out of it?

John Fraizer: You bridge another device in line and have THAT device collect your data. Not as trivial for OCx connected routers but still possible.

Paul Ferguson: see also: http://www.cisco.com/warp/public/707/22.html

Brett Watson: [You packet trace with] passive monitoring. we don't have anything yet

to run at oc-x speed (Packet over SONET) but CAIDA is working on several versions of passive monitors and at least one commercial vendor is working on one that is ip capable.

Richard Steenbergen, (Abovenet): With a Foundry BigIron you should be able to monitor high speed ports (they presently offer up to OC12 Packet over SONET as well as the usual ethernet) with no additional load on the device, just dump it directly to a gige port, get a PC with a $300 netgear gige nic, and you're off and running. Its not OC192 monitoring but if thats your customer aggregation device you're much better off, and Foundry is always up to something with bigger and better on the way...

Sean Donelan: I've wondered what type of statistical sampling could be used to find these attacks, but not require huge amounts of storage. The theory is these are very large traffic flows which congest the pipe and push other traffic out of the way. If you sample 1% of the traffic, and 99% of the sample is the same src/dest pair, something may be fishy.

VJ Gill: How about looking at 1 in n packets, for some large value of n, perhaps as a percentage of line rate? This leads into the entire issue of building ASICS in the fast path that punt 1 in n out towards some collator mechanism, with perhaps the first order data reduction done in the router itself before it is handed off.

Once again, these things will cost money to build, take time to debug, and the entire data collection system will be non trivial to scale. Problems that can be solved given enough talent/time/money, but is anyone willing to put forth the effort?

The Yahoo Offline Thread from Nanog

On February 8 **Joe Shaw** wrote: I'd be one to argue that implementing egress filtering, as opposed to ingress filtering, would do more to stop DDoS attacks since one of the most crippling attacks uses forged valid source addresses to start the attack (smurf/fraggle). If you stop forged packets from leaving the offending networks (which you mention in your RFC, but only to say it's impractical to do both ingress and egress filtering and advocate ingress) and the need to track attacks goes no farther than the people in Company X's dialup pool who's causing the CPU on the router to go up. However, neither ingress or egress filtering helps stop any of the latest "seen in the wild" DDos attacks like trinoo, tribe, etc. because the floods are all unforged packets.

Roeland Meyer: You've nailed the heart of the problem right here and never noticed. It is significant that the packets were NOT forged. IOW, they were legitimate packets of sufficient number to cap those very large pipes. I recently performed the Platform Architect role in a large .COM deployment. As part of site evaluation I had a chance to visit the facility where eBay is hosted. In fact, that is the same facility that I wound up using. Lots of dark-fiber capacity and over 20 Gbps capacity at the facility and they support 10000baseSX back planes. I swear that I saw a few Cat 6509's in eBay's racks. This means 1 Gbps pipes, scalable in 1 Gbps increments, using gig-Ether link aggregation.

Joe Shaw: Though they've been sketchy on details, it sounds like these or their descendants are the likely candidates for both Yahoo and Buy.com. Also, ingress filtering certainly doesn't help Tier3.net when their 4 inverse-muxed T1's are clogged with 20Mbps of traffic, forged or otherwise. Sure, the router is dropping the traffic like mad, but it's not going to help them unless their upstream will block the traffic as well once the attack starts. Egress filtering would stop the attack before it started if the traffic were forged. If it's just unforged traffic, you'd expect the attacking sites to notice the spike in bandwidth utilization and increased traffic flows from one or several machines to one destination, but that may be asking too much.

Meyer: Gentlemen, this is a very large site, with plenty of spare capacity. It is significant that those pipes were capped, via excessive, non-forged, traffic. Although it speaks well for the infrastructure that delivered that traffic, it also scares the shit out of me. There are a very large number of very large systems, sitting behind some very large pipes, that are compromised. Think about that for a moment. These are not small machines deployed by college kids and internet newbies. No one trusts the operation of a $1.5M Sun e6500 by a group of rookies. They can probably afford to hire the best SA's that they can find and no one running equipment behind anything larger than a T1 can afford to hire the ignorant. Not at the prices charged for that size of a pipe. Just the same, those systems were compromised.

Jim Mercer: I beg to differ. I've seen a large number of .com's which have mega gear, and minor-league SA's. The problem is that a lot of .com's seem to think that the vendor will be able to solve their serious problems, or that they can just hire some "certified" SA's to do the job.

Meyer: You can differ all you want, but I've never seen a large .COM deployment without at least one senior SA swinging the cluebat, backed up by a network planner and a senior architect. How do you think they write all that code and have it work? Design is much more difficult than maintenance and takes more skilled personnel

Neil McRae: Only if the design was a good one. If its a bad one, maintenance can be a nightmare. [and few designers want to do post-deployment maintenance, as its traditionally an operators job].

Joe Shaw: Unfortunately, the rush to .COM riches has brought with it a lot of people who have only half a clue as to what they're doing if we, as the Internet community, are lucky, making the Internet landscape even more dangerous with the amount of ignorance that's out there when it comes to security issues. It should also be said that some established educational institutions seem to be having issues stopping attacks like smurf and fraggle as well. The media certainly isn't helping, classifying all DoS attacks as packet flooding attacks, which is not the case either, though all DDos attacks are (if you're a journalist, please feel free to ask what the difference is; I'll be more than happy to explain it).

Meyer: I smell denial here. The compromised systems (only 52?) had to have access to pipes at least 1 Gbps in size, in order to carry out this attack (do the math yourself). Either there were many more systems participating (in itself a scarey thought) or many of these large and professionally run systems are owned and their operators don't know it. The only other alternative is the conspiracy theory from hell.

George Herbert: No, they don't. Assume there's 40k of data in the homepage. How many bytes of SYN-SYNACK-ACK-GET / HTTP/1.0\n does it take to do a TCP connect and request? I just tested, I show 160 bytes. That's a 250:1 leverage for the attacker. To fill 1 GBPS worth of outbound trunking you only need to generate 4 MBPS (32 Mbps) worth of input. 50ish systems with T-1 connectivity gets there with margins. [Note that this is an a priori analysis; I haven't bothered to find the attack codes in question and see if that's what they're doing, nor am I involved in any of the current operational response].

Meyer: Okay, but you've still missed the point. Even if I stipulate everything you said here, that's still 50 largish systems that are compromised. I would almost wager that the perpetrators didn't use all of their assets either. That's a shit-load of large compromised systems on the Internet. Doesn't that thought worry you in the slightest?

Phil Sykes (Cable and Wireless): It worries everyone! Dave Dittrich in his analyses of DDOS tools (available from http://www.washington.edu/People/dad/) suggests: "Trinoo networks are probably being

set up on hundreds, perhaps thousands, of systems on the Internet that are being compromised by remote buffer overrun exploitation. Access to these systems is probably being perpetuated by the installation of multiple "back doors" along with the trinoo daemons." CERT suggests (http://www.cert.org/incident_notes/IN-99-07.html).

Havard Eidnes: That sounds like good things to do. Others have pointed to RFC 2267 which is somewhat the same. However, it doesn't seem that we're doing all that well on actually following up those suggestions? As if that isn't enough, may I also draw your collective attention to draft-ietf-grip-isp-expectations-03.txt

How are we collectively doing on following up on those points?

During this discussion I've seen some claim that the recent attacks were not being carried out using spoofed source IP addresses. That may be so, but still is not a valid argument for not protecting the network from source address spoofing and the effects thereof.

Sykes: Prevent installation of distributed attack tools on your systems Prevent origination of IP packets with spoofed source addresses Monitor your network for signatures of distributed attack tools. Should we as network operators be taking a pro-active role to police our users for DDOS running boxen?

Eidnes: Sounds like a good idea. However, is it a sufficiently good idea so that a sufficient number of people actually find the time to do something about it?

Sykes: It seems to me that educating end-users is the problem here, just as educating people to use 'no IP directed-broadcast' was back in 1997.

Eidnes: Well, according to the list on http://www.powertech.no/smurf/ we're not done on that front by a long shot:

114951 networks have been probed with the SAR 19589 of them are currently broken 14682 have been fixed after being listed here

May I suggest that we all get off our collective butts and do something about these items? Even by going so far as to proactively probe our customer networks and/or extracting info from the list available from the above site?

Or are we once again going to hear the kneejerk and in my humble opinion irresponsible reaction from some ISPs (no, I don't have any particular in mind — you know who you are) that essentially says "more packets on our networks means more business for us"? Another common claim seems to be "this is none of our business". IMHO not a very responsible reaction that either...

Herbert: Back in Nov 1996 when Sun was pushing WebNFS initially with the Solaris 2.6 release, I wrote up a vulnerability analysis white paper using the UDP NFS functionality and this leverage approach and sent it in to Sun. I suspect the ultimate inability to secure against it was one reason WebNFS died on the vine. With full HTTP, you need more request bytes and a valid originating IP address since it's TCP... you need the SYN, SYNACK, ACK to work before you send the request. But there's enough leverage anyway with modern page sizes (8k was big then, it's nothing now... 40k worth of html is typical) for it to work. The only downside to doing it in HTTP is that all the attacking systems are clearly identified since they have to use real routed IP addresses.

Scott Crowby: Yes, you can send 160 bytes and the HTTPD will attempt to send 40kb, but the TCP stack won't actually send it all unless it gets ACK's from the reciever, which means that the reciever has to be able to accept at least some of that traffic. If there is sufficient congestion to keep the traffic from arriving and ACK's being sent, the sender will slowdown. So this type of attack would be throttled on the initiator's side through TCP slowdown and missed ACK's.

Herbert: If attacker has raw socket or TCP stack manipulation on the attacking box then they can "cheat" and pre-send ACKs for data not actually received yet once the connection opens up. This is explained in detail in several articles in the ACM SIGCOMM journal over the last year and other sources. It requires a bit more work by the attacker but forces the victim to send all the data (most of which is then discarded silently by routers somewhere upstream of the attacker due to congestion, and not noticed by the victim because of the faked ACKs).

In reality the technique hits statistical limits due to that congestion losing the SYN/SYNACK/ACK/HTTP GET packets needed to set up the connections in the first place, although all of those are re-sent if not properly acknowledged the throughput of TCP drops through the floor as loss rates increase as high as they will when doing this type of attack. But if bigger packets are more likely to get dropped (typical attack total packet 60 bytes, response 1k) then you can get a fair leverage out of it even so.

No Agreement on How it Was Done

Meyer: I suspect that this is not a kiddie-cracker activity. It is too well planned and carried out with too much discipline, over too long a time. I suspect that whomever is doing this has been silently "owning" systems for the past 18 months. I suggest that everyone start looking for signs of mwsh and its cousins. Because, I further suspect that the perpetrators have NOT used all of their assets. There are still a good many systems that are compromised, and not taking part in the current fracas, we just haven't found them yet.

Herbert (referring to Meyer's earlier statement that "that's still 50 largish systems that are compromised"): 50 systems across the internet with enough CPU capacity to near-fill a T-1 on a sustained basis with identical HTTP requests. Which is to say any modern multi-hundred-mhz RISC or x86 box with a reasonable OS, not really "largish". The processing needed in the OS TCP and IP stacks on the attacking system is most of the effort, and we're only talking in rough numbers 1,000 connects/sec for the attacker.

Do I believe that there exist 50 or more T-1 connected hosts with that capability level or higher which still have vendor default setups and thus are vulnerable to this sort of attack, penetration, and then use as a distributed DOS attack participant? Yes, without a doubt. 50 simultaneous sites compromised by one attacker would be on the ambitious side these days, but some of the remote exploit scripts (and corresponding known holes in vendor supplied system configs) are pretty damn easy to use and it wouldn't be out of the realm of the practical for someone to do it if they worked hard, or got a small cooperating team to work on it.

Of course the significance of this is highly worrisome. But the numbers have been in this rough performance range for attacker capabilities for several years now. That the tools used by attackers took that long to catch up is actually somewhat surprising to me, I was expecting this sort of thing some time ago.

John Payne (referring to Havard Eidnes - [we should] do something about these items? Even by going so far as to proactively probe our customer networks and/or extracting info from the list available from the above site?

Payne: and actively use the logs to contact peers when they're used as amplifiers against you, rather than just filtering?

Richard Steenbergen (Abovenet): Actually I had some ideas for a fairly interesting method to fight smurf attacks directly while being attacked without causing further disruption on anyone's network (just need time to finish writing it), and of course more granular information about the attackers (complete list of broadcasts unsed in an attack, the ability to track multiple attacks simultaniously for use in a span port envi-

ronment, information for amount of bandwidth seen from each broadcast sorted by worst-attacker, etc) would be a part of that.

Payne: Does anyone have a CIDR to broadcast address script handy? (where the network address is part of the CIDR format :-)

Steenbergen: Yes, I used to do a fairly large smurf amplifier scanning and emailing operation back in the day (infact a lot of the netscan.org stuff was directly based off of it). Code was never (and will never) be distributed because of potential for abuse, but my scanner was quite fast (the only way to improve speeds would be to throw much more cpu/boxes at it or do some kernel land work). Almost all of this proactive work has stopped since smurf ceased to be such a "huge" problem in itself.

But, depending on the size and design of your network, at least one or more of the following should be considered: #1 filter your damn customers so they can't spoof out (I really can't stress this enough, particularly if you are an educational institution with a large dorm resnet! DO IT!), "ip verify unicast reverse-path" #2 rate-limit ICMP echo/echo-replys at ingress and egress points #3 filter/rate-limit ICMP 8/0 to the most obvious natural mask broadcast addresses (.0 and .255) both ingress and egress

And one of the more interesting ones... With the high packet/sec syn/ack floods, the kids have realized that attacking the upstream routers is often far more effective then a well protected downstream. All it takes is pegging the CPU (against Cisco's this isnt very hard, even GRPs fall over at extremely low (realtively speaking) rates against closed ports or under high volumes of UDP attacks) until BGP falls over and suddenly victim A is off the air (*). Consider numbering your core loopbacks and link /30s out of a single block which can then be filtered/rate-limited at network borders.

* As an interesting side note, the "best" and "worst" devices in this area... The Foundry gear, in particular the BigIron series (mgmt3 card is nice), has the highest survivability rate for a layer 3 device dealing with not only attacks through it but against the box itself, that I have yet seen. The worst seems to be the Cabletron SSR series, which does switching based on src ip/port dst ip/port combinations, and can only program new flows into the ASIC at a rate of about 3000 per second.

And (referring to Meyer's earlier statement that "that's still 50 largish systems that are compromised") **Steenbergen** continued: You've all missed the point. I've done a fair bit of research into this, and I would put my money on the numbers looking something like this:

75-200 compromised systems 90% on 10Mbps Ethernet Around 75% on compromised university servers and dorm Ethernets Around 24% on compromised commercial connections, 1% other Somewhere around 35-40% of these will be non-US, a large number of .fi and .se universities where gov't funding has produced large university backbones, and these are often the ones with the most direct bandwidth being applied to the victim.

The compromises will be done through standard script kiddie methods. (I highly suspect the recent influx of compromised attack hosts is directly linked to the discovery of more and more remote bind exploits which can be easily AXFR'd and scanned for script-kiddie style), bind imap qpopper anything that someone can write a scanner script for and they can fire off against fast places they think might net them more attack-shells.

I suspect the numbers of the attack are closer to 600-800Mbps and people like to round up. I also suspect that are very few "real" numbers of the attacks since 5 minute averages and MRTG are very bad at getting these things accurately (especially when routers are bogged down or unreachable). You'll see some hosts putting out more bandwidth then others, but probably around 40 will be the primary smurf "bandwidth generators", doing about 6-8Mbps, and getting amplified.

Charles Sprickman: Now I haven't seen these DDoS "tools", but if you want to imagine something really scary, imagine one exists that works like this: -attacker scans for the known OS vulns that will cough up a "#" prompt -attacker installs root kit with DDoS tool -that tool runs as a daemon that has the following features: -remote 'admin' via ICMP (payload of echo-request includes password, host to attack, duration of attack -daemon launches the http "GET" flood as described earlier based on the info contained in that icmp echo-request -daemon continues this attack as prescribed with no further intervention

So the attacker need only send a few packets to each compromised host to cause extreme amounts of damage. How would you track down the attacker?

Charlie Kline: You've just described stacheldraht (http://staff.washington.edu/dittrich/misc/stacheldraht.analysis). It wasn't built with forged IP sources on the ICMP "trigger messages", but we did just catch such an attack here recently, and the presumed ICMP trigger message had a forged source IP address. The bitch about it is that the DoS floods used forged source addresses, but only among the last octet of the source IP address; the first three octets are "valid" for the machine that's doing the flooding. This means that flood packets will get out even with ingress filtering down to the subnet level, one has to catch the attack in progress and stick in an access-list with log-input to snag the hardware address of the attack packets in order to be able track down the actual machine involved. It's very, very nasty.

Often we can go back through our netflow logs and find the original break in to the machine once we know its IP address, which of course leads us back to a valid IP address and possibly the perpetrator.

Sprickman: Sure, you could slowly find the compromised hosts and block them. You could even then look for where the ICMP "control" message that starts the thing comes from, but if it's a one-way control channel, the source the attacker sends the control packet from could easily be forged and you could easily miss the one magic 'ping' that starts the thing off...

The idea of such a tool is scary, and from what I've read about TFN and friends, it seems that they could be modified to work as outlined above. The worst thing about any effective DoS is, in my mind, the lack of an identifiable "attacker".

Steenbergen: They do work as above, with encrypted control messages. If you look at some of the code (and then manage to stop laughing) you will find some interesting ways to counteract, trace to the control nodes, and in some cases even immediately kill the daemon on every attacking node. Keep in mind that the people writing these things are doing it with often very little clue, experience, or thought. Most are blindly stabbing at things they do not understand trying to tweak things and test them out to see if it makes their victim "die any faster", ripping mismatched code from various places (like blowfish code from eggdrop), and creating what will quite possibly be one of the quickest ways to spend a long long long LONG time in jail when they get caught and lawyers and accountants start adding up the "cost" of their distributed fun and games...

Shawn McMahon (referring to George Herbert's claim that any modern multi-hundred-mhz RISC or x86 box with a reasonable OS, not really "largish"): Multi-hundred-mhz, nothing; a 486/33 can do that.

50 cast-off 486 motherboards with $50 AMD 5x86 processors could saturate those T1s and still get good GUI response. 50 Pentium IIs could do that, running even Windows 95, and probably have enough CPU left to get good RC5 cracking rates. :-) I think we're leaping to majorly unwarranted conclusions here.

Meyer: A simple case of denial here, T1's are not cheap. It isn't the CPU horsepower that is significant here. It is the access to the required bandwidth that makes this so worrisome.

In order to operate stealth-mode in a system, one must be on a box that has sufficient power such that the operation of your code consumes less than 3% of the box's available capacity. In addition, your network should consume less than 5% of the site's pipe, even during an attack. Remember, it appears that these hosts have been compromised for some time. Further, Sean indicates that the entire attack system was tested at least once and no one noticed. These guys have to be frugal with the assets if they want to contnue using them undetected. This indicates planning and discipline. These are NOT ignorant cracker-kiddies.

Shawn McMahon: Nonsense. Fire it up on all the Windows 95 workstations at a few public libraries around the country, throw in a hacked screensaver running on lots of RoadRunner PCs, and you've got more bandwidth than you can shake a stick at. The whole thing could be fired off by one trip to a public library, or from a high school.

You cannot draw any conclusions about this attack from the amount of bandwidth used. You will have to track down source sites, track down who hacked them, and follow the chain. Either that, or get the big providers to tcpdump their user's IRC traffic and grep for keywords; somebody will shoot his mouth off about this. This could be anybody of any level of ability, but I'm telling you that this is not in any way beyond the ability of script kiddies.

For purposes of this kind of attack, bandwidth is *FREE*.

Remember what we're positing here: 1) The attacks come from compromised sites. 2) The trigger is a single ICMP packet sent to each of those sites. You could run this over a 14.4k modem, no problem. You could run this over a Palm Pilot, plugged into a pay phone. You could run this from a PC sitting in your local public library, for free. It just takes setup time, and that can be done by writing a program that does something else, and has this lying in wait.

Or, an ActiveX control sitting on a site somewhere that fires up when it's hit and attacks. Put some information on the site (DeCSS info, maybe?), post a link on Slashdot so lots of folks hit it, and whammo, hundreds or even thousands of dupes running Internet Explorer suddenly use all their bandwidth launching bits of your attack. 200 dupes with 33.6k modems can flood a T1. 200 dupes with 512k ADSL can flood multiple T3s. 200 dupes with Road Runner can flood OC-[insert small integer here]. Multiply by your worst nightmares. Again, the fact that X amount of bandwidth was consumed tells us *NOTHING* about the nature of the attack. (Which is the only point I'm arguing, here, and is the fallacy the initial poster fell victim to.)

Meyer: This indicates one or two compromised hosts per site with 50-ish sites penetrated, at minimum (probably, 100's). I would wager that even the 50-ish sites actually used in the attacks had no idea that they were participating. This indicates low resource usage on part of the attacking code, since the first indicator SA's usually look for is abnormally high usage of resources.

Let's quit assuming that all other operators are incompetent and start assuming the worst, [namely] that crackers got this one by "competent" SAs, shall we? If this is the case, then any of us are vulnerable. I find it difficult to believe that there are 50 sites, with T3 connectivity or better, that are all staffed exclusively by incompetent operators, let alone 100's or 1000's.

Steenbergen: You are quite confused. T1's are cheap, OC12s are not cheap. CPU is the limiting factor in many of these attacks, but not because of the ability to saturate a T1 with HTTP GETs or any other such nonsense.

These attacks often taken down the attacking-victim as much as the attacked-victim, in fact often times they run their attacks so strongly that they are unable to access the systems to stop them, which is why all the distributed attack programs have a built in length of time for the attack to run, any signal to "stop" would often never be received.

The belief that previously seen problem were some kind of "test" is totally unsubstantiated guesswork, of little quality. Your numbers are totally random with no basis in reality.

Meyer: About a year and a half ago (ancient times) I had a client where three of their names servers were penetrated by the MWSH program (Millennium Worm Shell). The first one exhibited just the behavior you describe here, the second one operated at the 3% level that I indicated, and the third one stayed dormant until I provoked it. The client was all set to believe that only the one name server was compromised. Whereas all three systems were completely "owned" by MWSH. We wound up scrubbing all the DASD down to bare magnetic particles (format with 0xe5 in all sectors) and rebuilding all three systems from known good sources. We also upgraded them to BIND8 and placed specific blocks in "/tmp/..." and "/ ...." (fs level 0000).

Steenbergen: You are correct that most sites do not realize they are participating even after a huge attack that cripples BOTH networks. It has not so much to do with "competency" as attention to detail and careful network monitoring, though you could easily make the argument that operators who do not do such are incompetent. If you find this difficult to imagine you need a better imagination.

Meyer: Please remember that cable-modems are asymmetric and the aggregate upstream pipe is shared.

Christopher B. Zydel: Some MSOs choose to rate limit their user's upstreams as low as 128kbit/sec, others do not. For example, we limit our users to 1mbit/sec currently. As for the upstream communications channel, this is not much of a limitation. Typical DOCSIS configurations include multiple upstream ports tied to a single downstream. It is typical to combine a small number of optical receivers to a given upstream port (1 or 2). Each optical receiver typically carries 500 homes passed. Operating a 16 QAM carrier with a channel width of 3.2MHz yields ~10.24mbit/sec of bandwidth. Subtract a little for overhead, and figure you're doing pretty well and subscribe 10% of your passed homes, or roughly 100 users per upstream port. Your average user isn't pounding on the upstream too hard, so figure less than a quarter of these users really hit it hard, and they're not likely to all be doing it at the same time. I'd consider a few cable or DSL networks with handfuls of compromised hosts sitting on them a large threat given that it doesn't take a huge amount of bandwidth to create a rather damaging TCP flood. I realize that these users are not as threatening as a dorm network attached to a T3/OC-3c, but the CM/DSL population is growing a lot faster than the dorm population.

No Agreement as to Motivation

Deepak Jain: If we assume that the attacks are being lead by competent attackers, we must also assume that their motive could be more complex than just "hah hah, let's see if we can make Yahoo disappear." In fact, it could be far more interesting than just a technical display of capabilities. In light of Yahoo, Exodus and UUNET's issues over the last three days, anyone who doesn't consider this a mandate to improve the accountability of net-connected sites is seriously missing the boat.

Meyer: You mean, like the guy that threatened to publish 50,000 credit card numbers, with x-dates, if he wasn't paid off?

Jain: Extortion is a very sloppy motivation. How about something like "Our website

stays up, our competitor's doesn't." And the investors make out (either by shorting one, or going long on the other)... No threats, just marketing. My cup of tea may have been sour this morning. If am offending anyone's sensibilities, please disregard me.

Patrick Greenwell: It would make one hell of an excuse for those wishing more government(s) involvement/control....

Declan McCullagh: Even though I live in DC, I'm not that conspiratorial. Though I do note that the attack started the same day the president introduced his budget, which has a big increase in "cyberterrorism" defense and a HUGE increase in wiretappability funding :) More: http://www.wired.com/news/politics/0,1283,34164,00.html

Shawn McMahon: As co-moderator emeritus of Fidonet's CONSPRCY echo, I love a good conspiracy theory as much as the next guy, but let's get real, folks. No deeper motive need be applied to this than "ActiveX sucks, and I'm gonna show 'em" or "wow, look what I can do if I combine these two script-kiddie exploits I found on rootshell.com" or "oh, yeah? Prove it." The latter being the driving force behind many destructive hacks over the years. Until we have enough facts, Occam's Razor isn't very useful, but it would seem to exclude government-backed weirdness.

Joel Baker: One hard, solid data point: I was talking to a friend who is a part-time SA on a box colocated at his place of business (behind a 2xT1) which he found out was participating in the attack.

He found this out when the links suddenly spiked through the roof and his ethernet switch lit up with a nice, solid traffic light. The only reason he spotted it? He was at work at the time. Had it occured at night, it's quite probably that nobody would have noticed, given how rarely they check the traffic stats (since it doesn't really matter to them until the traffic is pushing their ability to carry it).

Travis Pugh: Lots of NSPs and ISPs are tracking customer utilization of links, either by MRTG or RRD ... .and many of them bill by utilization using these or other SNMP-based tools. It should be trivial, during a DDoS attack of the scale that took down Yahoo, to find participating sites. A jump from normal utilization to 100% link utilization should be easily noticible if it lasts more than 15 minutes (3 polling intervals, if you are doing it at 5 minutes). It seems to me that a customer would be more than willing to have a rate-limit or filter installed on their routers during this kind of event, especially if it helps them track down the compromised machine.

Host-by-host prevention, during an attack, should be very easy ... assuming a minimal amount of cooperation between upstream provider and compromised network, if link utilization is tracked and the spike is noticeable. Perhaps we should be notifying operations staff to be on the lookout for suddenly saturated circuits, and to be prepared to help out owners of compromised hosts with filter configuration?

Christopher B. Zydel: This sort of alarming is fairly trivial. Just about any network management system can be configured to poll interface counters on a regular basis and alarm when some threshold is reached. The difficult question to answer is "How long should the link be saturated before sending an alarm". With high speed links this is a lot easier. It's relatively easy to saturate a T1 with a file transfer, however the same would not be true for an OC-3c. This type of alarming should be based upon deviation from the established mean as well. (For example, if a circuit sees around 50mbit/sec worth of usage on a regular basis, and then spikes to 130mbit/sec and stays there, something is clearly wrong.)

Bouchard: I've seen instances where workstations of experienced people had been compromised for considerable periods of time without their knowledge. This, to me, is not surprising. My view of security is that its all about trust. Major public servers are watched quite closely simply as a result of the attention that has to be given to the applications they support. However, those same administrators generally don't watch smaller, auxiliary systems (ie, a 3rd name server several thousand miles away that serves no other function.)

Consider the responsibility of a corporate security dude and IT guys who is trying to watch over the network used by 3 or 4 thousand employees, most of whom have desktop computers and few of which know how to do more than email 3 meg excel files to 30 or 40 people all over the corporate network several times a day. If the network is not kept absolutely tight, everything is a risk.

I always work from the maxim (and those I work with have heard this at least a hundred times before) that "the easiest way to break into one computer is to break into another computer that it trusts." (eg. personal workstations... how many times have you looked at your process table this week?)

[Moreover] consider this thought: Random user breaches 10 sites each behind a T1. This user leaves these servers up and writes a script to take the IPs out of a file and start the attack. The user publishes the script to the user's friends. One of them goes and adds another 25 hosts to the list and re-advertises it. However, this user has found sites on 10 meg Ethernet being fed by a T3 and figures that 5 megs can be had from these hosts on average. This user publishes this AGAIN to someone who adds another 15.. repeat ad-nauseum.

People, that's 45 hosts that are just kind of let up, open for all to use. There is no reason that there can't be HUNDREDS of hosts on that list. There is no reason that there cannot be HUNDREDS of lists with a couple of dozen hosts each. The possibility of being able to use large numbers of hosts to launch such an attack is VERY REAL. And at that level, if you have an average of, say, 768K from 150 hosts, you are sending 115 megabits at the target. If you manage to pull 2 megs each from these (say, cable modem or something), then that goes up to 400 megs. The possibility is there, people.. And it gets worse.

Joel Baker: [Continuing his description of the attack.] The box? RedHat 6.0 without the security patches; from logs, it appears to have been taken by an automated attack, via the old NFS bug. Nothing at all surprising there, of course. This sort of thing is not exactly rare. Compromised boxes at .edu sites have been a thorn in many operator's sides for a long time now, and other sites happen as well; the difference is that the attackers are now biding their time (which may not be all that long) before launching an attack, so that they have enough points to fire it off from.

While this hardly rules out a more "professional" attack, it's quite possible for this sort of thing to be accomplished by a bored or angry kid with nothing better to do. Or more likely, a group of half a dozen of them doing it for kicks, scanning for hosts for an hour while doing homework, all week, until they have a sizeable list. If you think that's bad, wait until they find a way to compromise Windows hosts on DSL lines. That... will be deep pain.

Joe Shaw: It's very similar to your friends RedHat box with NFS holes. It's called File/Printer Sharing and any of the available trojan packages out there. You'd be amazed at how many unprotected Windows file shares are open with full access enabled.

Dan Hollis: Its already happening. It's called backorifice and netbus (and a legion of other trojans).

Henry R. Linneweh: here is a little trojan list http://nethog.com/feeds/niteryder/trojans.htm

Jared Mauch: I firmly believe that the security groups will be working closer together after this week than ever before. I spent some time talking to the C&W security group today about a problem related to the stream.c

exploit being used.

Is anyone dropping traffic from src/dst ips that are currently reserved and do not have any allocations out of them (such as 60/8 for example) anywhere inside their network? I know that MAPS has gone and done something similar to this as it relates to their RBL, but I am not aware of any providers doing anything but route filtering on these prefixes, not packet filtering.

But What Exactly Was the Attack?

On February 9th **Rodney L. Caston**, Southwestern Bell , Internet Services wrote to NANOG: "I spoke with a person that claimed to understand the attacks that are going on, while I have no proof, I offer this as an example of what to look for on your own systems. So I am presenting this only as a possible example of what has taken place, and until proven correct I concede this is only a "rumor." "

"Basically it began by combining many scripts already in use for scanning system security holes, the script initially scans a range of IPs scanning each target system for various known exploits, once a system is compromised, the second half of the attack goes into effect. I believe it uses some form of remote execution via rcp once its been compromised to copy and execute what seems to be a specially made "DoS Daemon" to the host, once there it this daemon runs waiting to receive its orders from the people who put it there. Therefore, once enough systems were compromised in this fashion and enough systems on the net were unknowingly running this daemon, the attackers simply gave the order to hit the targets this week and their daemon's went to work. With this in mind we would need someone to find a box with this daemon on it so we can find a way to detect its existence on other systems. Logically, since the compromise of the systems was done with a script, this "DoS Daemon" would be setup the same way on every compromised system. Therefore, if someone can find it on one box, we will know exactly what to look for on other hosts. This of course will only help us if our own systems have been compromised and wouldn't be of any use at all for those boxes not within our control.

One final note, a friend from Verio suggested that in the above scenario that this daemon would probably be using TCP to be communicated with as UDP is more difficult for a lot of people to code.

Sean Donelan: Has anyone else noticed the dearth of technical information about these attacks? Although some of the largest web sites, and networks have been hit, I still haven't read a confirmed description of exactly what is happening. It has been three days. After the Morris Worm, by this point in time I had seen several technical descriptions and even portions of decompiled code. And I was just an interested Internet user in those days.

In this case I still haven't seen confirmation if it was trinoo, tfn, something new, or what. Or even confirmation if it was a series of HTTP GETs or random packets, or some interesting corruption of a packet. Or if confirmation the attacks are coming from the same set of hosts or different ones for each attack. If it is the same set of IP addresses, could we RBL (or create a new RBL) them?

Carson: It's because people are being very closed mouthed with this, the corp[oration]s either have no idea what is going on or do not wish to share what they know, and those involved with the attacks have done a good job of keeping silent. Besides comparing Morris's worm to what is going on now is hardly fair, the net was a very different place then, and his cpu cycle hog of a program was a lot easier to deal with and detect.

Donelan: In past cases, the "crackers" have eventually been caught not by closed-mouth corps and law enforcement, but by "civilians." The closed-mouthedness just seems to be extending the list of victims because the later ones didn't know how to protect themselves until after they got hit. Then after a few hours of downtime, they figure out which filters to install. Wouldn't it be nice to know what filters to install to protect your web site before it gets hit?

Carson: I think we just had this discussion about that 'c' word, cooperation... and why it will never happen.

Larry Synder (Lexis Nexis): From what I've read so far, it's still not clear whether it was an attack on a host(s) or a pipe(s). It probably wouldn't be a bad idea to release at least that much info....

Donelan: I'm not too concerned about clueless media and PR flacks [who don't know]. But at NANOG I spoke with several people I thought would know, who didn't. I didn't talk to any GlobalCenter folks because I couldn't find any. They disappeared on Monday. But I did speak with several security people with other providers, and they hadn't heard any confirmed technical details. Just speculation about what had happened. In particular, everyone was wondering what made the attack so hard to detect as a DoS. Ok, I know, I don't work at an ISP anymore, so I'm not a member of the club. I think several departments at WorldCom are under orders not to speak to me. But instead I found the security folks at other providers were happy to talk about it, but didn't know any more than me. This worries me.

Bouchard: I ran into half a dozen Global Center folks while I was there both Monday and Tuesday. On Monday, of course, most of them get paged back to the office to deal with the problem and later spent time writing up incident reports. However, as expected so soon after such an event, they did not desire very much information to be let loose since they had not yet finished correlating everything and had not yet finished their discussions with yahoo on what occured. As for now, well, I too would be interested in learning exactly what happened (which attack, roughly how many relays, secondary effects, etc) but the chances of learning that are slim just because of the fear of PR problems it might create.

Daniel Hagerty: Sharing information with the edge about what's happening, and what to look for needs to happen. Why aren't the attack destination NOCs getting this info out? Three days into this, and I have *no* idea what I would be looking for if I was the network manager for a large under managed edge site (mmm, .edu). "Look for a big traffic peak" just doesn't cut it; I've got locally "more urgent" problems than watching for a blip on an mrtg.

Given the highly distributed nature of this attack and thought being put into it, our lusers probably realize this and are *avoiding* pegging invidual ingress wires excessively. Until large numbers of sites are educated about exactly what's going on right now, and what they need to do their part of fixing it (with text for linux HOWTO level of clue; what many high-bandwidth capable sites are running w/ now), NOC staff isn't going to be sleeping much. Glad it's not my problem right now. Now, do you see the problem in this attitude? How many edge network managers are thinking exactly this way?

Joe Shaw: Do a search of the Bugtraq archives for trinoo, tribe, etc, or take at look at Dave Dittrich's page at http://www.washington.edu/People/dad/. He posted detailed breakdowns of the discovered DDoS daemons in December for the CERT workshop on DDoS's from last year. Verbose information on these attacks has been available since November/December of 1999. Referring to Caston's remark " that this daemon would probaly be using TCP to be communicated with as UDP is more difficult for alot of people to code," Shaw concluded that, some are using ICMP, and UDP is not that hard to code, especially if the programs are just combinations of scripts that have already been written.

[Editor's note: Shaw then lists long URLs that lead to detailed descriptions of TRIN00 and Tribe Flood Network Attacks. They are:

http://www.securityfocus.com/templates/archive.pike?list=1&date 99-12-01&msg=Pine.GUL.4.20.9912071041410.9470-100000@red7.cac.washington.edu and http://www.securityfocus.com/templates/archive.pike?list=1&date 99-12-01&msg=Pine.GUL.4.20.9912071044490.9470-100000@red7.cac.washington.edu

We have scanned the material. It is very ominous. We hesitate to have an opinion but it looks to us like given the state of the internet while individual sites may cleanse their systems, keeping the internet free of infestation by programs like this appears to be impossible.**]**

Shaw: I'd be worried if they didn't have theories or know about the known DDoS attacks, but not if they didn't have specifics. Tier1 NSP's seem to be very tight lipped about these sorts of things when they are the victim. I'm sure there are GC employees on this list, but none have come forward to give any details. Could be a gag order, which wouldn't shock me at all. Hopefully we'll know something eventually, but for now we're all mushrooms when it comes to official information.

Donelan: I guess the techies reduced to reading the New York Times for technical details. Today's New York Times has a description from GlobalCenter's PR person. The Yahoo attack was a large number of ICMP EchoReplies(PINGs) coming via GlobalCenter's 50 peering connections (about half of GlobalCenter's total peering connections). Which may explain the "50" number I've been hearing. The original ICMP EchoRequests listed Yahoo as the source address and were directed to other networks which replied to Yahoo. GlobalCenter installed rate-limits, but didn't know if they are effective in preventing attacks. Yahoo's spokesperson confirmed GlobalCenter's account. (Free NYT registration required) http://www.nytimes.com/library/tech/00/02/biztech/articles/10attack.html

Bino Gopal: As Charles says, from what I've read of the CERT advisories, there is nothing proactive one can really do for these DDos attacks, besides securing machines from being hacked, correct?

Randy Bush: yes, that is essential. but also, what has to be done is to go to all dedicated cpe and prevent source spoofing, see RFC 2267.

Robert: A site can use anti-spoofing filters on their router/firewall (even if the ISP can't or won't do it on their end) to make sure that their machines don't forge source addresses. This might stop any of their machines which have been compromised from really doing participating in the attack. (I say "might" because the slave daemons don't have to forge addresses.) Other misc. ideas are in: http://www.cert.org/reports/dsit_workshop.pdf (BTW, a "advisory" version of unicast RPF-type stuff would be immensely helpful in deploying URPF, "source validation," ingress filtering, or whatever you want to call it.)

No Real Defense

Travis Pugh: On the subject of cooperation, has anyone set out to catalog where these attacks are coming from, at least in terms of compromised networks, and share said information?

Shaw: As far as I know (thank you C-SPAN), the FBI has logs of the hosts used to originate the traffic, and are now going through them to find the "innocent third parties." At this time, since it's part of a current criminal investigation, this information will not be made available to "the public," though they are saying this is going to be a joint venture between the FBI and the Internet Community.

Pugh: I know similar catalogs sprang up in response to smurfs ... is it time to start listing offending networks? Even better, does anyone know if the attacks are using something like TFN2K and using dummy addresses to obfuscate real attacking hosts?

Shaw: Not sure, since it seems the discovered DDoS programs don't seem to have the capability to forge the traffic, though it's not too terribly difficult to modify existing exploits to do so.

Pugh: I see a lot of talk of attacked sites putting up router filters to stop attacks. Can anyone who knows let the rest of us in on what was filtered ... was Yahoo taken down with a flood of HTTP GETs, ICMP, UDP, SYN floods, or what? If this is a DDoS, the attack could probably be fingerprinted ... this would be very useful information if we are going to see more tomorrow. Do we know if the source addys are spoofed, and if an attacker could turn off spoofing, revealing the source of the traffic but getting around some filtering?

Shaw: I have a feeling you're going to see many more in the next couple of days, and certainly some peripheral meltdown as an after effect. While no official details regarding the attacks have been announced that I've read, there are a few advisories on some of the known DDoS attacks. Dave Dittrich has posted some excellent material on the DDoS's that have been found and you can view them at his homepage located at http://www.washington.edu/People/dad/.

He also has links to scanners (written by NFR President Marcus Ranum, Dittrich, and others) that can help look for the known DDoS daemons on servers.

Pugh: I am making the assumption that the last three days' attacks were caused by the same person or persons. But the intent is the same regardless ... we can all go back and forth on NANOG about what might be happening, and wait for the feds to chase down the attacker(s), or people who have been attacked or might be attacked can compare notes and try to get an idea of where the attacks are coming from and exactly what they are.

Shaw: Well, to quote a Wired article, "A Yahoo source close to the problem told Wired News that they hadn't contacted the Feds during their trouble yesterday because it would do no good."

Pugh: Any relevant info would be appreciated. Nobody knows who is next.

Shaw: Indeed...

Shawn McMahon: cooperate with each other, find the son of a bitch who started it, and prosecute him in each and every state the packets passed through that has computer intrusion laws, so that even if he's found not guilty or given token sentences, he gets that way one trial at a time in 50 states. Lather, rinse, repeat. Should be good for 15 to 25 years of trials. :-)

Bouchard: You forget.. the network is not US only. Any views of this nature are short sighted. International connectivity is still poor but its getting much better. Now consider that given the distributed attacks, you can have sources and relays located in the country you want to conduct your attack even though you are half way across the globe in a country that considers such actions worthy of no more than a shrug.

Laws in one country DO NOT apply to other countries. Now consider someone setting up such sites, starting the attack from a hacked account (which can't be traced back to him since he dialed in from a pay phone somewhere) and then just leaving it. This attack could go on to some degree for *WEEKS* without being completely squashed.

Mike Bird: For whatever it is worth, we've seen several failed exploits from APNIC addresses in the last two months. I generally ping flood them for a couple of minutes to encourage them to go away. The ping flood responses usually show modem level connectivity although one appeared to have T1 bandwidth.

The FBI has never shown the slightest interest when we've told them of compromised

systems in the US or told them where rootkits are being stored on public FTP servers in the US. What are they going to do when the crackers are working via systems in Korea or Japan?

Longer Range Implications

On February 11 discussion shifted to the IETF mail list where **Bernie Volz** asked: Regarding the recent TCP SYN Flooding attacks, why aren't ALL ISPs required to put filtering on their networks that PREVENTS packets with invalid source addresses ever entering their infrastructure?

Michael Warfield: Clue alert...The recent attacks were not TCP SYN Floods. Please check recent Bugtraq and Cert information regarding Distributed DoS attacks.

Further references:

http://xforce.iss.net/alerts/advise40.php3 http://www.cert.org/advisories/CA-2000-01.html http://www.fbi.gov/nipc/trinoo.htm

Detailed analysis of TFN (Tribe Flood Network), Trin00, and Stacheldraht (Barbed Wire) are here:

http://staff.washington.edu/dittrich/misc/tfn.analysis http://staff.washington.edu/dittrich/misc/trinoo.analysis http://staff.washington.edu/dittrich/misc/stacheldraht.analysis

Contrary to popular belief and the common press, TFN2K (Tribe Flood Network 2000) also has Windows versions of the slave daemons as well as Solaris and Linux versions. A lot of these attacks appeared to be SMURF style attacks and TFN (Tribe Flood Network) and TFN2K have distributed smurf capabilities. This wasn't even close to being a TCP SYN flood.

As far as spoofing goes, in their SMURF mode, the only spoofing is the src_addr part of the ICMP echo that the slave systems send to their LOCAL broadcast address. That src_addr is the address of the system being attacked by ICMP_ECHOREPLY packets that simply consume all its bandwidth. Check out the analysis. Anti spoofing entry filters would have been of zero effect.

Valdis Kletnieks: See RFC2267. The problem is that the IETF doesn't have the legal authority to beat ISP's into submission on this one. There's also the problem that many ISP's are somewhat marginal in cluefulness, so things like RFC2267 tend to be of the "preaching to the choir" variety. Given that RFC2267 is over 2 years old now, what *do* you suggest the network community at large do to motivate the sites that still haven't implemented it? Would somebody be interested in running a BGP blackhole feed of prefixes known not to be filtering, similar to the maps.vix.com feed for closing off E-mail spam? Perhaps if that became prevalent, ISPs would cleanup their act when their legitimate users couldn't get anyplace because their ISP wasn't filtering.

Paul Ferguson (Cisco): Do you think that if RFC2267 was advanced as a BCP that it would carry more weight, and therefore more ISP's would implement RFC2267-style filtering? Coupled with the latest denial of service attacks?

Valdis Kletnieks: On the one hand, I think it would make a good candidate for BCP. It seems to be similar in tone with RFCs 2502 and 2644. I'd have to re-read it to see if it would need any textual changes, or if it's OK as it is. I was talking to a co-worker on this topic, and his exact quote was "We have our s—t more together than most sites, despite our best efforts". The problem is that he was right - our site may have clue, but there's a lot of uneducated sites out there.

Does anybody have statistics on how effective RFC2350 (Expectations for Computer Security Incident Response) was? Or RFC2502 (Anti-Spam Recommendations for SMTP MTAs)? Or RFC2644 (Changing the Default for Directed Broadcasts in Routers)? It would seem reasonable that moving 2267 to BCP should have a similar effectiveness...

Stephen Kent (BBN) (responding to Paul Ferguson's call for RFC2267 filtering): A more technically focused answer is that most routers are not designed to perform the filtering without adversely impacting throughput, and because dual homing and the mesh nature of Internet connectivity makes it hard to apply appropriate filters for all subscribers (remembering that some subscribers are really down stream service providers ...)

Ferguson: That's simply propagating FUD, and I think that by making such sweeping assumptions, you are doing the Internet community a disservice. Is RFC2267-style filtering a perfect solution for every situation? No. Sure there are some scenarios where it foo bars transit traffic, but in the larger scheme of things, most dual-homed networks are not providing transit. Does it impact router performance? Perhaps. It really depends on various arbitrary issues.

From an architectural perspective, it is very important _where_ you filter to be effective, not cause transit problems, and not make smoke roll out of the back of the equipment. Will it stop bogon source packets from being injected into the Internet, so that anyone foolish enough top launch a denial of service attack can be traced back, identified, and prosecuted? Absolutely.

List member: best compromise is the built-in access filter. The solution must be general enough to work for multihomed, defaulting out customers with blocks from n providers,

Paul Ferguson: No, that is a common misconception, or rather, an overstatement of a pretty easily described situation. It only breaks things in transit situations, and only in transit situations where you might not have the same forwarding path back to the source as you would via the same interface a packet came in on.

Vijay Gill: This is more common than you might believe. For Dialup and single homed, yes, this is not a problem in most cases. For a very large customer base, this problem does not scale all that well, especially for the large backbone carriers who are transiting a lot of traffic. As the internet grows more important to business, more and more people multihome.

Ferguson: This is a small percentage, I would thing, since the percentage of ISP's offering transit pales in comparison to all other "access" ISP's that do not. And in cases where ISP's _do_ offer transit, or have transit agreements, will they really do this on their transit interfaces? I think not.

Gill: I think you're solving something else. I submit that almost _all_ isp's offer transit for their customers. Thats where the I part of the SP comes in. For _peering_ links (peering being defined elsewhere), yes, this is a hard problem, but on the edges of the _peers_, this is not. If everyone filtered their T1/DSx/OCx/E1/E3/STMx customers at their edges, using Unicast RPF where appropriate and filters where appropriate, life would become better.

On February 14 **Donald Eastlake**: | I think that making egress filtering a BCP, applying community pressure, bringing law suites, etc., will be about as effective at eliminating forged source address packets on the Internet as similar measures have been in eliminating open SMTP relays...They help, but not much.

Robert Elz: I'm not sure there is a good analogy there. There's no good purpose in sending packets with incorrect source addresses I can think of, and stopping the practice is the basic intent of the filters. The only justification for not doing it is cost - and then just just becomes a part of the cost benefit analysis - will it cost us more or less to implement this?

On the other hand, SMTP relays are not a problem anyone cares about of themselves - just the contrary in fact, smtp relaying can

be a very useful function to have available. Eg: you're travelling with your laptop and have a bunch of mail waiting to go - you get a connection for a few minutes between flights, but your normal home relay is unreachable. Right now - being able to pick some other friendly relay and simply park your mail on it can be a real advantage. The thing to be fixed there is the unwanted spam that is also using the services of the relay. In other words, it isn't the relay that is really the issue, it is the spam. If all the spam went away, no-one would care about relays any more (other than now to regret that whereas previously most sites would be happy to relay mail now far fewer are).

Further, it isn't at all clear that preventing relaying will do much, if anything, to stop spam - certainly blocking receiving mail from relays will currently cut the amount of spam you receive a lot - but that's because comparatively few people do that, and so the spammers are content to ignore them and just continue making use of the relay services that they can latch onto. But should everyone stop relaying, does anyone really believe that all the spammers are simply going to decide that there is no point continuing, and all just go away? Really? Even if it means that the spammers have to send all their mail directly, they'll do it as long as the benefit from sending spam (at least appears to) outweigh the costs.

So, the two issues are really not much alike. In one there's no good purpose to be served by not blocking outgoing packets with bogus source addresses. In the other there are lots of philosophical reasons for not stopping relaying - hence there are some of us quite willing to do one but not the other.

Spam is a social problem, and needs to be solved by social/legal means, not technical ones. (There is no technical difference between spam and any other mailing list mail - it all looks the same - the only difference is whether the recipient wanted to receive it or not.) But we're technocrats. All our tools are technical ones, so it is easy to see how we grasp at any technical solution we think might help - the only technical solution we've been able to find that seems like it might help (a passing illusion really). Unfortunately, the appearance of a technical solution reduces the pressure on the social/legal types to come up with a solution that really works. If we all would just admit that technically there's nothing that can really be done about spam, and simply stopped trying at all (allow it all through) the user pressure to get this problem solved some other way would be much much greater... On the other hand, sending packets with an incorrect source address is a technical problem - those packets don't meet the IP specs - what is supposed to be in that field is the IP address of the sending node. This is a problem entirely open to a technical solution.

But Filters Are Not the Answer Either

Anders Feder: "In his early days at Intel, Andy Grove was approached by an employee who suggested the company start work on a personal computer based on its chips. Skeptical, he asked what a personal computer might do. The employee, searching for a good example, said it could be used to store recipes. Grove thought about the millions he'd have to spend on research, development, and marketing, then considered the imperfect but steady quality of an alphabetized loose-leaf binder. He finally passed on the idea and decided to concentrate on the lucrative business of supplying chips for traffic lights."

It is rarely very easy to see what requirements the future will bring and particularly in this business you can't be sure what the technology of tomorrow demands. And, agreed, bogus source IPs _does_ at present time look like nothing but the devils work. But in, say, 10 years a new flashy techology could be requiring that you have the ability to stamp packets with other IPs than your own. Unfortunately, back in year 2000, somebody put in IP filters at all ISPs and now, 10 years after, these filters is so integrated a part of the ISP software that reprogramming would cost a fortune. Also consider the size of the group of Internet users that send out packets with incorrect source IPs. Using IP filters would be like illegalizing coffee because a fraction of the people on the earth is allergic to caffeine.

Charles Perkins (Nokia): Mobile IP would like to send out packets with the mobile node's home address, while it is attached to a network in a foreign domain. The home address is likely to look "incorrect" from the standpoint of such a filter.

Plus I don't think the gain is worth the pain. I'd rather see a technology that actually solves the problem instead of swatting at gnats with a sledge hammer.

What if routers could preferentially keep track of things like SYN packets and so on for a few seconds, and we had some traceback management software and security associations with our neighbors enough to do some automatic detection? It might cost 2% more for the memory buffers, geez I don't know.

On February 14**, Phil Karn** (Qualcomm and mobile IP expert): There already are some perfectly legitimate reasons to send packets with "alien" IP source addresses. Mobile IP is the best example, but various virtual private networking schemes also do this. For example, I have a VPN set up over my cable modem so I can have a block of static IP addresses for my home network. I had to do some work to evade the $#@!! source IP address ingress filtering in my cable network. I do it by tunneling the upstream packets through a machine at work.

Daniel Senie: Yes, and you chose the CORRECT solution. Think about it... VPN in most cases also means encryption, and at that probably back to a central site. Gabriel wrote RFC 2344 for reverse tunnels, and it does essentially what you did.

Karn [2/16/00]: Yes, I often use encryption, but not to a central site. Generally I apply it at the application layer (SSH/SSL) so the peer is whoever I happen to be talking to. However, this is irrelevant to the issue of upstream IP address filtering.

Karn [2/14/00]: Being forced to tunnel not only increases the size of each packet, but also entails suboptimum routing and reliance on yet more network elements. I use the new Linux policy routing mechanisms to tunnel only those packets that have to be tunneled, which helps. But it would sure be nice if I didn't have to tunnel my outbound packets at all.

Senie: Sorry. You're at the point where technology meets policy. Fact is, your host identifier in the IP stack is the source IP address. Enforcing the validity of that identifier has become necessary.

Karn [2/16/00]: This makes no sense at all. I've shown both that there are legitimate reasons to send packets that the ISP might consider "alien", and also how easy it is to circumvent ingress filtering if you are so motivated. By the way, ingress filtering breaks things other than Mobile IP. Consider the DirecPC service, which gives you a one-way (forward) satellite channel at 400 kb/s. Your return link is via local dialup service provider. If the local ISP (or its upstream provider) does source filtering, you can't send your perfectly legitimate packets into the network over that ISP without tunneling them all through DirecPC's own network connection, which may be on the other side of the continent.

Karn [2/14/00]: Source address ingress filtering is one of those ideas that seems like a good one when you first think about it, but it just doesn't pan out. It interferes with some perfectly legitimate activities, it doesn't really stop the bad guys, and it deflects attention away from the real solutions.

Senie: The case for "legitimate use" of source spoofing has not been sufficiently made. Operational reality is such that it's not sustainable.

Karn [2/16/00]: The operational reality is that you'll never be able to implement ingress filtering widely enough to provide much of a benefit. Even if you do, it'll be circumventable (consider the many Linux and Unix boxes out there that support IP-in-IP tunneling). And the energy you spend doing so will be energy that could have been better spent implementing other mechanisms to defend the Internet against MS-DOS (multiple source denial of service) attacks.

Karn [2/14/00]: In the case of MS-DOS (Multiple Source-Denial of Service) attacks like the ones we saw last week, we need to deploy better inter-router mechanisms to deal with congestion by moving the packet drop points as far upstream as possible toward the senders. And if these mechanisms can work for deliberate flooding attacks, they'll also deal with non-deliberate congestion.

John Hawkinson: The correct "solution" here is for Phil [Karn] to communicate with his provider who presently does ingress filtering and notify them that he is sourcing packets from a particular set of source addresses (because of tunneling), and request they add these to the ingress filter list. I think we recognize this may be politically infeasable for many people to do, because tunneling is often used to circumvent administrative restrictions, but that really is a different degree of the problem.

More generally: It seems to me that there has been a lot of wasted discussion on this list on this topic. I think it is a well-accepted conclusion within the provider community that ingress filtering is a necessary thing (perhaps a "necessary evil" to some, perhaps a "good" to others). In the vast majority of cases, the router CPU resource consumption issues are not relevant, either. If people within the IETF feel differently, they should realize that they are currently against recently-accepted-operational-practice. I don't think everyone who has to date spoken is aware of that.

Valdis.Kletnieks on February 15: [Referring to Karns ' statement that we need to deploy better inter-router mechanisms to deal with congestion.] The problem here is that there is often a limit to how far away you can move the detection. In the case of multiple sources, it's *probable* that the inbound packets will arrive on as many as 5 to 20 different links, and not get aggregated onto one path until the last-hop link to the victim's site.

And if you have 20 inbound links into a routing swamp, each one will only see a 5% fluctuation in load in order to cause a 100% congestion on the victim link. If you move the detection 2 hops out, you may be trying to spot a 1% ripple in the traffic, if there's 100 different paths that far out. The more hops you try to move the detection away, the smaller the "ripple" you need to be able to detect *without a high rate of false positives*.

There's another issue, in that if you're trying to do this 2-3 hops out, you will need *secure* *low-bandwidth* communications regarding who is talking to whom, at what rates. And you get transitivity problems - some of our border routers are 3 hops from a vBNS gateway, and therefor would need to talk to them, plus are 3 hops from other routers that are probably NOT going to want the vBNS information. So you end up with a ugly mess of many overlapping "circles" containing different subsets of routers. This gets you into key management issues, and the like.... I'm sure there's other issues that need to be solved as well, these are just the first few problems that come to mind...

[Referring to Karns ' statement that "source address ingress filtering… doesn't pan out."]: Well.. as soon as somebody presents us with "the real solution", we'll start implementing. In the meantime, filtering is the best we know how to do.

Vernon Schryver: I really wish "we" actually knew how to filter. Just as I feared when the news broke, I'm seeing more paths where neither traceroute nor ping work, apparently because some of "us" are so expert that we turn off all of ICMP, and never mind intermittent blackholes from Path MTU Discovery or diagnosing routing or other mere technical problems.

Michael Warfield: But some of us turn off ICMP except for ICMP_FRAG_NEEDED and keep MTU discovery alive while cutting off the ICMP food fights and script kiddie probes that seem to be endemic in our current mess. You betcha traceroute and ping are broken (figuratively and literally). Just as broken as I can make them. You don't need to be tracerouting or pinging into my network and that's my choice to make. I can break them without breaking MTU discovery. You want to diagnose routing or other technical problems, why aren't you contacting me?

You cut off ICMP_ECHOREPLY and all but a tightly restricted set of UDP and you have just starved these DDoS zombies of the vast majority of their communications facility. TCP is much easier to trace, if they fall back to that (I don't see how TFN2K could - it utilizes a blind forward-only non-responding channel). Blocking spoofed packets won't do nearly as good a job since a lot of the packets aren't spoofed. Doesn't help at all in some examples I have some first hand experience with.

Edge filtering and spoof protection would have been absolutely no help for one site I was help with forensics after TFN2K. They were a source of ICMP_ECHOREPLY packets in one of the storms. We STILL haven't located the zombie amongst the hundreds of potential Windows boxes (we already cleared the single Solaris and single RedHat box on the subnet and TFN2K is known to run on Windows).

No spoofed packets crossed router boundries. The TFN2K command sequence was executed well before anyone noticed any bandwidth anomolies, so there's no track back to the master. (Who would have noticed 20 ICMP_ECHOREPLY packets that merely didn't correspond to any internal request?) The slaves shut down after two hours and before the admins realized that the smurf was being generated internally. They did find a bug in a router blocking of directed broadcasts. That red-herring slowed them down, thinking it was externally generated. No trace back of the smurf triggers back to the zombie (having that MAC address would do wonders) was caught since it quiet before they started sniffing the network.

They broke it's back by blocking ICMP_ECHO and ICMP_ECHOREPLY (fortunately, this attack was not using one of the UDP options). I have not heard if any of the TFN scanners have turned up anything yet. TFN2K zombies are a real bummer since it can lay dormant on a network until triggered and they don't reply back to the master. These guys are on a machine by machine, file by file snark hunt looking for files that could be any one of a number of names on systems that are all different anyways. Don't you know they're having fun?

The attackers are gaining in sophistication. We are beginning to see effort at "covert channels" for communications. The tricks of communicating via ICMP_ECHOREPLY packets with commands in the payload and sending blobs of forward only commands with no reverse channel responses are just examples. How long before they start sneaking past those defenses by sending ICMP_UNREACHABLE/ICMP_FRAG_NEEDED with a command data payload?

They don't have to resort to spoofing to pull off a lot of this. Long command time delays, covert channels, and unidirectional command direction can make it just as difficult to track down as spoofing. Diagnostics are fine. Performance is fine. Diagnostics and/or performance at the expense of security and/or robustness is not.

Charles Perkins: But maybe filtering is putting the cart before the horse.

Schryver: I agree. If I were building a DDoS engine today, I'd write a conventional (Microsoft) DOS virus that does nothing except once every 3 minutes do the equivalent of: echo "GET /index.html HTTP/1.0"; echo) | telnet -r $1 80 (maybe instead with a random request instead of /index.html)

After a few 1,000,000 desktops have been infected by familiar virus vectors, the victim might notice the traffic. How would you filter for them? Even if you could give routers enough processing power, what would you learn from the filtering that you'd care to apply?

John Hawkinson: I believe in general we have been having a discussion about operational practices, not really about theoretical ideas of what we might be able to do twelve to eighteen months out. But discussing those theoretical things is good too, but I think it's important to make clear what the scope of any particular message in this thread here is, otherwise confusion runs rampant.

Incidently, while we're here, you might look at Stefan Savage's paper "http://www.cs.washington.edu/homes/savage/traceback.html", which kc pointed NANOG at yesterday. I describe it as "having each router probabilitically mark transit IP packets with a router-specific locational referent in the ip_id field."

Perkins: From that analogy, I claim that the appropriate action is to develop tracing systems that will help to identify a wrongdoer. Here is a possible design. - Create a router feature, able to be remotely activated, to keep track of "suspicious" packets for a few seconds up to maybe a couple dozen seconds. Suspicious packets might be SYNs, or even packets with "incorrect" source addresses.

Hawkinson: That is a lot of suspicious packets. Is a packet to a random UDP port "suspicious"? If not, then this scheme is mostly useless against our most recent series of attacks. If it is suspicious, then the attackers will switch to TCP packets — are those suspicious too? Are all packets suspicious?

Perkins: - If a violation is noticed, determine the interface on which a bad packet was received, and send a request to that neighbor along with appropriate credentials that can be checked.

Hawkinson: This generates more traffic in the network for each of these; when the problem is traffic flooding, as it has been most recently, this may be unwise. What would a credential check involve? Where would these credentials be issued from and how-validated?

Perkins: If, on the other hand, a neighbor sends a request for tracing a problematic packet, check the credentials that accompany the request and look at the stored data for that packet. If no stored data exists, send back the bad news, and possibly enable the pattern-matchine function to capture similarly featured packets in case of future requests.

Hawkinson: This suggests some sort of AI, or at least relatively clever heuristics. My brain says "research problem—IRTF—bye!" Perhaps that's unfair?

Perkins: If a neighbor's request refers to a stored packet, figure out what interface the stored packet arrived on. If the packet came from another neighbor, forward the request. Otherwise, look for the malicious source internal to the domain.

Hawkinson: Where look for is "send a message to an operator"? How does this work across multiple routing domains (e.g. provider boundaries)?

Perkins: The basic idea then would be to trace back bad packets that conform to some typically innocent, but occasionally troublesome, profiles. The profiles will become self-evident with experience, and once people know they will be caught by this traceback system they will think twice before spreading their crap around.

According to one knowledgeable source, this strategy is undesirable because it would cause routers to maintain more state.

Hawkinson: It appears to imply state on the order of 1 unit for each packet the router transits. That's an immense amount of state.

Perkins: But, I claim that we have to maintain state to do any tracing, and I also claim that we need to enable tracing in order to detect and identify wrongdoers.

Hawkinson: Yes, tracing requires state. How to distribute or limit that state is the essessence of most of the discussion of tracing schemes. Marking packets puts the state inside them. Only collecting state on some kinds of flows or in response to some trigger also limits the state.

Perkins: So the costs boil down to more memory, more software, some pattern-matching hardware, and maintaining security relationships with your routing partners.

Hawkinson: Sure. I think that doing anything other than what already has been done that might loosely fit your description is likely to be terribly unpractical and feels like a research problem.

Perkins: I think it's a very worthwhile tradeoff. Much better than mostly ineffective measures that have big negative effects and small positive effects, characteristics of ingress filtering as I understand it.

Hawkinson: Ingress filtering is something we can do today, and has a very small negative effect — it limits the ability of some people who use the network in non-traditional ways (half-tunneled setups, etc.), and those people can simply request a broader ingress filter; I haven't seen an example of how that fails, except when the people doing the ingress filtering refuse, and that's not a problem with ingress filtering, but instead a problem of politics.

Steve Kent [BBN]: [Filtering] it is a bad thing if one bases defenses on the assumption that ALL the access points into the Internet will perform such filtering, and will do it consistently. Even if all ISPs, and down stream providers performed the filtering, there is no guarantee that attackers could not circumvent the filter controls, either through direct attack on the routers, or through indirect attack on the management stations used to configure them. I'm just saying that while edge filtering is potentially useful, it would not be a good idea to assume that it will be effective.

Elliot Lear [Cisco]: Let's be clear: a DOS attack is something the end point itself can do very little to prevent, since it usually fails or succeeds upstream of that end point. Therefore, the end point relies on its upstream ISPs to "do the right thing" and indeed, each of those ISPs relies on other ISPs to similarly filter. Each point can mitigate the damage to the point where in sum these attacks become ineffective. Each RPF check can remove bad packets. Each violated ACL can remove and LOG the bad packets. These are the best controls available today. Shall we not use them? Also, we raise the bar from some kid injecting packets to someone breaking into an ISP, a more difficult challenge (at least a level 3 attack on my Dungeons and Dragons guide of Hackers ;-).

On February 16, **Kent**: Some of the DoS attacks we saw last week were good, old-fashioned SYN floods. Hosts do have a responsibility here, more than ISPs, since it is quite feasible to tie up a host's pool of TCBs with a small number of packets, even if the attack tool does not use spoofed sourced addresses (or if the spoofed addresses are from a legitimate pool allocated to a subscriber site).

Senie: And at least one of the attacks last week indeed was a smurf. Filters would have helped in some cases and not others.

Kent: The point I have tried to make, unsuccessfully, is not that performing ingress filtering is bad, and thus should not be performed. Rather, I am pointing out that it is a bad idea to rely on such filtering as a primary means of defense.

Senie: Reliance on any single method will be insufficient. It will take several methods.

Kent: There are several reasons for saying [that reliance on filtering is bad]: - not all ISPs will find it feasible to provide such filtering - not all ISPs are trusted to do the filtering (in the global Internet)

Senie: This issue can be addressed by their upstream providers with terms of service agreements. Will it be perfect? Unlikely.

Kent: A number of DDoS attacks can be launched without using spoofed addresses outside of those "appropriate" to the subscriber site Senie: Which is NO reason to permit attacks which DO spoof. Filtering is going to limit the types of attacks which can be launched, and permits idenitification of the systems and networks where the packets come from. It may still be necessary to chase down those sites and get action, but at least it's possible to know WHICH SITES are the sources. Some applications may legitimately make use of non-local addresses, as others have suggested.

Senie: A point which is in dispute. Beyond that, operational reality is that filtering DOES occur, and may occur somewhere neither you nor your upstream ISP can control. To at least some extent it's useful to understand present operational reality in choosing methodologies.

Kent: I have seen a long history of suggested solutions to security problems which are only partially effective against current forms of attacks, vs. providing protection against a larger class of attacks. I'm trying to suggest that we not follow this pattern.

Senie: Please suggest another course of action. To date, the IETF has spent a considerable amount of its security mindshare on IPSec. While developing that functionality, operational reality resulted in the deployment of huge numbers of NAT boxes, and IPSec and NAT aren't interoperable. Another case of operational reality and design colliding.

Ingress filtering does not provide a complete solution. Neither did the patches to operating systems to guard against SYN flooding. By using both of those, we limit the effectiveness of those types of attacks. I expect we will have to add a whole suite of techniques to bring the problems under some reasonable level of control. What is unclear is why some think we could or should do nothing in the short term.

Kent: Finally, there is a diminishing difference between what a script kiddie can do, vs. a clever attacker, because the clever attackers are freely distributing higher quality attack tools. "Empowerment" is a hallmark of modern Internet attacks :-).

[And to Dan Senie]: I'll suggest one course of action, but I keep emphasizing the issue is not one of alternates, but of recognizing the limitations of proposals now on the table and considering approaches that may work irrespective of whether everyone performs filtering.

With regard to a wide range of DoS or DDoS attacks, it seems quite feasible to monitor traffic to the web site to detect such attacks irrespective of whether source addresses are spoofed or not. (this differs from IDS for broader attacks, where the recognition problem is much harder and the false negative rate is on the order of 20%.) Such monitoring can be done by a web-hosting facility through purely passive monitoring, so as not to adversely affect the performance of the network used by a web-hosting site. Once an attack is detected, one can trigger a semi-automated response. If one believes that the source addresses are not spoofed, then one can use this to direct filtering to selected ingress points, but the filtering can now be very focused, based o the characteristics of the detected DoS traffic. If one believes that source addressed might be spoofed, then one needs to activate the selective filtering on a much wider range of ingress points. Since the true sources may be outside of the ISP's sphere of control, filtering at connections to other ISPs may be required in either case.

If the response is rapid enough, the attack may not have significant impact, which reduces the attraction of mounting such an attack in the first place. One can begin disabling the filters once the offending traffic flows have diminished, which provides another means of determining the sources of traffic, as others have noted in previous published work on this topic.

An advantage of this style of approach is that while it can be even more effective if source address filtering is widespread, it also would work if such filtering is not completely effective, which is the sort of self-defense approach I prefer It supports what the security community refers to as the Principle of Least Privilege.

Steve Bellovin [referring to Kent's statement that "some applications may legitimately make use of non-local addresses, as others have suggested"]: The problem here is that flooding attacks target the network, not the host. The host is thus not capable of mounting a defense — it's not the victim, in some sense.

Conventional security methodology would say that the aggrieved party should do the authentication. Of course, that's hard on the Internet — in fact, we don't *want* people to have to authenticate themselves to the network elements in order to transmit. (There are, of course, networks that do have such requirements. The most common form is known as the telephone system. I don't think we want to reinvent that.) Filtering is a very coarse form of address-based authentication to the first outside hop; I don't see a better choice.

Perhaps the network can use beefed-up congestion control mechanisms to stop such floods. I hope so, but it seems to be a research issue; I'd be surprised if such new mechanisms could be deployed sooner than 2002. What do we do in the meantime? (Trying to secure all of the myriad endpoints is even more hopeless than trying to get all ISPs to do proper filtering.) Do you have any specific suggestions? Seriously — what do you recommend as a defense against flooding attacks?

Jon Crowcroft: so in the case that the server resource is overloaded, but not the link, what you can do is extend the servers scheduling discipline (e.g. in dealing with existing connections at a higher priority than new ones, and applying filters to new ones based on content/application level metrics (e.g. popularity indices, which is something many web servers have got anyhow), to priortise and starve the bad guys...

If the links/routers near you are also being clobbered, than some form of "loose unicast RPF source quench" mechanism would be a semi-automated way that server sites could extend this scheduling back "towards" the bad guys - the idea is to take a mix of modified versions of the resource control messages such as icmp source quench ECN RSVP Reject etc, and use them to install soft state upstream towards the bad guy.

This state can either block, or just add a weighted RED higher drop probability, or even move traffic into a lower class in a diffserv priority, CBQ, WFQ or other...

- but note unlike intserv/rsvp, this state doesn't have to be in all hops - just the appropriate border/choke point....so it has lots of nice scaling properties (its only their for a minority of addresses/flows/ports, its only in a few places, and only when you need it) - now if someone spoofs a real source address, this represents a problem since you can use this mechanism or something like it to launch an indirect attack using this ....but there are ways to block that (including making it mandatory to use ipsec to use this mechanism...)

Anyhow, its basically a sort of BGP policy controllable, internet friendly scalable automated version of phoning up your ISP and having them back track through all the further ISPs...

Continued on page 30

Part Two: A New Calculus for the Internet
The COOK Report Explores Ed Gerck's Ideas
The Relationship of a Quantum State Internet to Security and Privacy and of Data that Obeys Physical Laws to Mechanisms for Conveyance of Trust

Editor's Introduction: This interview is the result of the questions that occured from our edits of Ed Gerck's essays that begin on page 23 below. Further reflection has caused us to use this interview as an introduction to the two essays. We begin the interview on February 25, with an examination of **Gerck**'s basic premise that the February DDoS was not merely a DDoS — but rather a CDoS. A Coherent Denial of Service attack.

"The difference is that a distributed but incoherent attack would not have done any major harm. In order to explain how such an attack was possible and why it was effective, one needs to understand first that nothing is coherent in the Internet . All packets travel in what may seem to be a random fashion, each host has unsynchronized time (oftentimes, even wrong time zones), and even the path traveled by each packet is nondeterministic. Thus, achieving the coherent arrival of a stream of packets at one location that originate from a large number of coordinated locations is a feat."

What Is Meant by Coherency

COOK Report: Precisely what is meant by coherent stream of packets? If I am amazon.com and, all of sudden, over a five minute span of time my incoming traffic triples or goes up by a factor of ten, or of 100 and these are just http requests that force me to send out ten or 40 times more bytes than I receive by way of answer, I do not understand why it makes any effective difference whether my Amazon servers get hammered with the increase in bogus packets within one fifth of a second, with five seconds, or within five minutes?

Gerck: Coherency, in broad terms, is any natural or logical connection. In mathematical terms, it is the amount of "overlap" between two events or functions in regard to some parameter. Coherency can be 100% (total), partial or none (0%). Say you have a certain amount M of packets randomly delivered within 10 seconds to your server and say they come through 100 different network paths. If all these packets are coherent within 1 millisecond in regard to *your server*, then they "overlap" in time to a large extent and become 10,000 times more effective in inducing a congestion or a bottleneck somewhere within your server. Why? Because there are 10,000 milliseconds in that ten second arrival window. And instead of being evenly spread out throughout that time window of ten seconds (that we are choosing arbitrarily), all packets are delivered together in a single one thousandth of a second with a mighty "wharumph!" This creates the congestion that becomes the DoS, and it does so while still being 100 times less in volume in each of the network paths leading to your server (in the given example). Thus, by being coherent (i.e., by "overlapping" in time) within 1 millisecond at your server, these M packets could create the same effect as 10,000M packets being delivered to your server and still be within the traffic limits of the networks leading to your server. This technique is especially effective because packet bursts have a higher bandwidth than that of average traffic, in general.

COOK Report: So the assumption is that dozens, hundreds, or even thousands of slave machines may be configured with instructions to send a bogus pattern of http requests to amazon when activated by a master machine.

Gerck: Or, to each other, creating an amplified stream of packets before reaching for the machines that send to amazon.com.

COOK Report: If the delivery is coherent, my understanding is that the packets arrive in sync - that is at the same moment.

Gerck: Yes, mostly, but also somewhat "depolarized" or stretched out by the traffic conditions inherent in the Internet through which they must travel.

COOK Report: So if they hit all bunched up together, rather than in more random drawn out sequences, under such conditions that the impact that 10 machines have could perhaps seem like an incoherent attack from 100 machines?

Gerck: Or 10,000. Funny thing, the more congested the IP network is, the larger the "benefit" of the coherent attack — because a congested network would naturally spread the packets from a distributed but incoherent DoS over a few seconds, while a free network would naturally introduce less delays and lost packets. Ross Anderson, a security researcher in the UK, reminds us also that the Internet was designed to provide a communications channel that is as resistant to DoS attacks "as human ingenuity can make it" [in http://www.cl.cam.ac.uk/users/rja14/eternity/eternity.html], due to redundancy — but redundancy is of no help to defend against a massive coherent DoS attack, and actually allows such attack to occur by providing redundant channels to mount the attack. So, human ingenuity can undo what human ingenuity has previously done.

COOK Report: But we really don't know how many machines were involved. A large number of machines directed to send packets to Amazon over a period of thirty minutes or three hours would hit Amazon without coherency. Nevertheless, it would mess it up pretty badly I think.

Gerck: No . I disagree because such a "brute force" attack would probably shut itself off as the Internet's back off congestion mechanisms would decouple more and more the attacker machines from the target machine. Doing this would more and more "depolarize" that is to say spread out the possibly already small time coherency between the packets.

But what happens if it is a flood of HTTP GETs versus ICMP, UDP, SYN floods or something else? Then, the congestion mechanisms would not apply for ICMP for example but, still, the target machine would receive packets that must travel through nondeterministic routes which are also traffic dependent -- thus, suffering "depolarization". If we are looking for "brute force" attacks and we try to see something more fundamental than just simple protocol features, we thus recognize that the principle of "depolarization" will apply to any protocol and will make any "brute force" attack less efficient than it might have been.

COOK Report: If we knew a smallish number of machines did great damage, it seems to me that coherency could be seen as an explanation. But we don't know how many machines did it.

Gerck; This already shows you that the number of *critical* machines is relatively small, doesn't it? That is, it is harder to hide a huge number of machines.

COOK Report: Now perhaps you are trying to say that in a larger network like the flap-

ping wings of the butterfly in a chaotic atmosphere can send out perturbations that can become a hurricane 10,000 miles away, random actions at a few places may coalesce into a coherent impact somewhere else on the network. OK. Agreed. But this just happens. It is surely not a deliberately triggered action of a trin00 network.

Gerck: No, but this is also contemplated. It could be triggered by software bugs, as well as intentional acts.

COOK Report: Are you claiming that the attacks were not deliberate but the result of accidental perturbations?

Gerck: No. I am saying that natural perturbations, bugs, virus, trojans, etc. and simple bad luck can also provide an environment with what I call "prepared state" such that a small perturbation (intentional or not) can trigger a large traffic oscillation (eg, a DoS). Transit of packets through the Internet is like the passage of light through a turbulent atmosphere — packets too are subject to an effect similar to the atmospheric turbulence that shifts the light paths, which for example appears to make stars twinkle when we look at them. 'Turbulence' mitigates against the arrival of packets at a target server all at the very same instant, even though they were sent at the same instant from another server. If, however, a large number of packets do arrive all at once rather than in spread out fashion, the bandwidth at the server will spike and the spike will trip the server into a state of instability that can lead to a temporary or permanent (i.e., until reset) Denial of Service.

COOK Report: But earlier you said, "This technique is especially effective because packet bursts have a higher bandwidth than that of average traffic, in general." Please explain what you mean.

Gerck: In the Internet TCP/IP congestion mechanisms take a moment to react. They don't kick in completely with a single sudden burst. So a sudden burst will transmit rapidly but as you send more traffic, that additional traffic will begin to be dampened. The congestion mechanisms work on average traffic over some defined period of time. However when a server is reached by a burst, it saturates immediately. For example if you send a server requests for things like CGI pages, the server has to take memory and cpu cycles to create the page and then send it back. Hit the server in a very short period of time with an overwhelming number of such requests and it can potentially go into congestion — running out of memory and cpu. At this point it can't serve any more requests. Service by it is effectively denied.

COOK Report: How do you respond to someone who says simply — denial of service attacks have been around for a couple of years now and it is a well known fact that when you throw more packets at a server handle it can handle you effectively crash it. What's different about the current situation?

Gerck: If a server sees that a flood of packets are coming at it from the same machine (e.g., the same IP address), one thing that you can do to deflect it is to simply drop all such packets before processing the request, before allocating cpu time, memory, etc. So, the level of abuse that a server protected by this mechanism can handle can be very large. Also, you can, for example, use intrusion detection as given by the program TripWire — which defends its systems by means of integrity assessment. With this technology, TripWire's first wall is intrusion detection. That is reinforced by constant monitoring for unauthorized system change. For example, TripWire can trip up DoS Trojan infections by finding those programs even when hidden in the operating system. Once discovered, the system administrator can delete them. And one can also use TripWire, or other programs, in order to track attacks, so you'll know exactly what happened.

But these solutions work only for a given pattern, for example — a given pattern of Trojan horse programs, a given pattern of IP source machines, a given pattern of occurrences in your system. They cannot protect you against a coherent attack that can have *any* pattern, that can use standard components of your operating system, that can use any IP source address in rotation, that can cause *any* pattern of occurrences in your systems. To protect against coherent attacks one would need to look for coherence, not look for patterns.

Now imagine what happens when I put the IP number of Amazon.com as the return address of the packets being sent from me. By sending a ping flood stream of packets to various machines on the internet where the packets I send are marked as originating from Amazon.com, I can cause them to send their replies to Amazon.com inducing them in effect to attack Amazon.com. I can even cause them to rotate their responses through other machines and then on to Amazon so that they don't feel they have been used.

Packets sent in this fashion would have difficulty arriving at Amazon.com in a coherent fashion. They would be emitted at different times and sent through network interfaces by different routes where some would be dropped and some not and where all would be acted up by whatever congestion mechanisms were in place on the routes they traveled. At the beginning of an attack the network is relatively free and uncongested. When a stream of packets is injected, that stream moves rather quickly towards its destination. When some packets reach the servers for which they are targeted, those servers send out their responses which must compete, in their journey back to their sources, against more and more incoming requests. The network's congestion mechanisms begin to come into play. A built in feedback loop — under normal conditions — begins to disperse the potential coherency of the inbound attack.

COOK Report: So this feedback loop prevents the arrival of a large number of packets at the same set of servers at the same time? And this is why you say a distributed denial of service attack could not have created the havoc that we saw?

Gerck: Yes. The Internet is not designed to prevent coherently caused congestion. So, what we saw must not be just a distributed DoS.

COOK Report: So if this were a distributed denial of service attack, the longer it went on, the more it would be damped by back off mechanisms throughout the network paths that the attack traveled? And to be distributed, presumably, the packets are getting to their destination via hundreds or even thousands of routes?

Gerck: Yes. And with a very high degree of depolarization along the way. This would also happen to ICMP packets because packet routes are non-deterministic and may change with traffic.

COOK Report: The more routes being taken the more opportunity for depolarization?

Gerck: Indeed, and the more traffic the more routes you will have to take because the routes will become congested. This produces an isolation effect that generates a kind of cloud blocking access as the attack progresses. Now if you have distributed denial of service, precisely because you do not have coherence, you will really need to strike hard with an enormous number of machines. A really enormous number of machines ranging into the tens of thousands, would be much harder to imagine as captured and controllable by a single attacker.

COOK Report: So the problem is that there were some reports that with high volume sites like Amazon and Yahoo where traffic is normally in the 100 megabit per second range traffic spiked to a gigabit per second! And, if you follow the discussion that we have just had about the network mechanics to be expected from a distributed attack, it would take every machine in the entire Internet to be able to produce such a spike! Even then it would likely not produce the peak in volume seen at the servers itself, but produce much congestion enroute to the servers.

Gerck: Exactly.

COOK Report: But why wouldn't the leading TCP/IP engineers and network designers be asking themselves these questions? Unless what we are dealing with here is a major paradigmatic shift that in Thomas Kuhn's (*Structure of Scientific revolutions)* terms simply doesn't impact the consciousness of those brought up in the old paradigm?

Gerck: It's very hard to say. Perhaps one reason is that, for many people, trust in their own world-view inhibits their receptivity to new ideas.

How Might a State of Instability Be Reached?

COOK Report: So in your view what does happen with a trin00 attack? How is the state of instability reached?

Gerck: What we have here is a problem in collaboration between machines. One machine infects another and they then work together. Collaboration here means doing different things at different times, but working for the same objectives. The question becomes one of how you start a process that becomes coherent at the end. Consider a chain of amplifiers. What happens when you put a microphone too close to the loudspeaker which is putting out the amplified input to the microphone? You hear a strong noise that sounds like a whistle.

Where does the whistle come from? Perhaps from a movement of air that is captured by the microphone, was amplified and them came back out of the loudspeaker pretty much like a movement of air, and then was again picked up by the microphone and further amplified in a kind of chain reaction. As this process continues, the amplification becomes strongest at the frequency that has the highest gain. Now this system, for each passage of amplification, amplifies thus even further those frequencies to which the microphone-amplifier-loudspeaker system is most sensitive. Other frequencies are amplified but not as strongly while a small set of sensitive frequencies becomes greatly pumped up. Its gain in amplitude in comparison to the other frequencies becomes greater and greater. And what was essentially a white noise becomes colored peaking into a whistle.

COOK Report: But help me to understand the process involved in one of these attacks where someone starts a cascading process down a tree branching structure of machines.

Gerck: It is a coherent amplification process. One packet sent by one machine is picked up by two machines which send packets that are picked up by four machines, which send packets that are picked up by 16 machines and then by 256 machines and so on exponentially. The very first machine is then "hidden" in the ensuing avalanche.

But note also that the traffic output can be skillfully divided between an ever larger number of machines while remaining at essentially the same level on a machine-to-machine basis of 'background traffic' and therefore very hard to detect at the originating end. Dispersed at the beginning, they come together with quite a bang at the end. Why? Because their arrival is coherent in time, i.e., they overlap in time to a large extent.

COOK Report: What has to happen for the attack to become coherent?

Gerck: There are three processes involved, in my model of it. The one that produces coherency is called stimulated emission. In this case, when you receive a packet you send a packet with a close correlation to the one you received. The second is absorption. This occurs when you merely receive a packet. The third is spontaneous emission, when all of a sudden you transmit a packet with no apparent external cause. These are the only three basic processes that need to happen, everything else in my model is reasoned out in terms of these three processes. What happens depends on which process wins out over the others, in a dynamic fashion.

In experiments done in a university environment in 1998 I showed that the Internet can indeed sustain coherent patterns of amplification after a critical number of involved hosts is reached — in which case the process which I call stimulated emission wins out over spontaneous emission. Spontaneous emission is essentially incoherent. You just send out a packet. In stimulated emission you respond to a packet by sending out a packet that has a high correlation to the one that you just received — you have micro coherency, but not macro coherency. The condition to go from micro to macro coherency (i.e., coherency that you can observe as a system property) is that one needs to have a critical number of hosts sharing a same prepared state from which stimulated emission can occur. The prepared state could result from a trin00 tree, from a bug in widely shared software or from a virus. A single ping to the root of a primed trin00 tree with the address of Amazon.com could cause the stimulated emission process to avalanche.

COOK Report: So your equations would show that, given one series of inputs, you would reach criticality with "x" number of machines in the prepared state and given another series of inputs you would reach criticality with only "y" number of machines?

Gerck: Yes. Spontaneous emission is the result of normal use of the network by normal users, as well as absorption. I am a spontaneous emitter and absorber (i.e., receiver) of packets and so are you. This is pretty much uniform in terms of average load for each class of user and, unsurprisingly, follows the 80/20 rule (that the majority of traffic is generated by a minority of users, usually in the 80/20 proportion), but stimulated emission can be prepared so that a "y" number of machines can be much more effective than a "x" number of machines in this regard.

COOK Report: What you describe sounds like the process of pumping up a laser and getting it ready to discharge?

Gerck: Yes. The word laser comes from Light Amplification by Stimulated Emission of Radiation. Within the laser a small amount of light is amplified by being passed back and forth between mirrors until it reaches saturation, exiting as a very coherent stream of light through a (usually) partially transmitting mirror (as one of the mirrors used).

COOK Report: What are the common identities that become coherent. Are they in the software, the hardware, or in the protocol?

Gerck: Coherency can defined by all three. For example how much lag time do you have between sending a packet and receiving a packet? That time may be very small or fairly large according to conditions affected by the other variables, but it may be coherent even though it can be large. The main idea is that when you send a packet and the machine responds you can calculate the coherence between what you send and what you receive. If you send something and you get nothing back, the coherence can be zero or if you send something and you get something back in 10 miliseconds it may be zero also. But if you send a series of packets and almost precisely two *seconds* later after each packet being sent a packet is received, then you may say that you have a high degree of time coherency in that response — in spite of the large time delay. In a coherent system, there is a very tight relationship between cause and effect. Now I may send out one packet and get ten back. If those ten are sent back to me randomly, I would also say that the coherence is very low. If the ten always come back to me in the same predictable and precise sequence of arrival, we would say that the coherence is very high.

You are, in such a case, using a server to achieve predictable and uniform amplification of your packets. The function of a server, remember, is to reply to a query. When a server replies, you can measure the delay time and you know the different routes that are likely involved and you can, making these observances, switch on the attack in

such a way as to maximize coherency *at the target*.

COOK Report: So can you get a kind of a finger print of the routing system as it functions over time. And then you assume it keeps this configuration for a small number of seconds, so that you can construct a model to predict the majority of the paths?

Gerck: Well that would be a kind of manual way of doing it. Actually, in the equations, these 'calculations' happen automatically, because as the packets go back and forth they seek the least amount of time. They optimize themselves normally on the shortest path available. But even this is irrelevant ˜ they could reach a local minimum but with a non-minimum global time. The relevant thing is that they arrive at the same time. Whether they take a millisecond to arrive or two whole seconds is not relevant. What is relevant is that they get there at the same time — mostly.

Can You Prove These Assertions?

COOK Report: Do you have the mathematical proof for what to a non mathematician sounds persuasive?

Gerck: Yes and the mathematical proof is founded on an understanding of the real nature of data. When we talk about data on the Internet, we generally do not attribute any substance to it beyond its existence as a string of numbers. We tend to view data as an absolute abstract value without physical characteristics. However, in my model, data is relative — data is a measure of the difference between two states. But people intuitively chose one state as the reference point — generally considered to be zero or the empty string — and thus this aspect is unseen. If I ask what is the number ten, I have to define whether I am talking about a base ten numbering system or about a binary system and so on.

Also the question has to be seen in regard to a reference level and units — meters, feet, miles? So, if I say I have the data "ten" I am implying that "ten" exists in some trusted context. The problem on the Internet is that the trusted context of my computer likely has nothing to do with he trusted context of your computer. When we try to compare data between two such computers, we must realize that we are essentially dealing with apples and oranges -- even though the data may appear to be apples and apples. In other words, I have no way to tell whether "2=2" is true or false unless I know the trusted context for it -- for example, as a C expression it is false to a compiler.

When we see data as the difference between two states and no longer as an absolute measurement, then we are able to write specific laws for data. Today on the Internet people think "we can do anything we want, because the laws of nature do not apply to cyberspace." But in real space we are confronted every day with the laws of gravity, for example. We cannot deny such laws. But in Cyberspace, everyone thinks that there are no laws — and I am not only talking about criminal laws (laugh). Everyone thinks that with the Internet protocol they can do whatever they want sending stacks of bytes here and there. They think that Cyberspace is unaffected by natural laws, by things defined by Nature (i.e., irrespective of the observer, us). But I would ask them: why would Cyberspace be any different from real space? Why would Cyberspace have no natural laws?

COOK Report: Could you explain what you are saying in the following way? If my priorities are to achieve a certain defined set of objectives, I would find that I would need to tune my computer system in a certain way to best use the Internet in maximizing my ability to attain those objectives? And that through some fluke you might have a set of priorities that really matched my objectives. If so, fine. In such a case we have a very trusted mechanism for communicating with each other. But I think you are also saying that out of any given pair of people the chances, if you get down into this fine tuning level, that the way that they would tune their machines to attain the objectives which they desired to achieve would ordinarily be quite different. Just as people are quite different. People use their computers in Cyberspace quite differently and for quite different purposes.

Gerck: That is true and if you assume that all people use the Internet in the same way you are creating a single system where you have only apples and apples. So my system is separate from yours, in some way. Mine is an apple and yours is an orange. But once we decide in order to create trusted contacts between our systems, that will in effect unify and integrate both, then we are taking those two isolated systems and for protocol purposes merging them into one.

COOK Report: And this is good, bad or indifferent?

Gerck: This is necessary — as a basic approach. You cannot compare references from two different systems. We need to have an integration which creates a common system and then you compare the references from that common system. This is the common wisdom.

Now, if we understand data as a relative measurement between two states, and we apply the principles of thermodynamics, we see that they are purely mathematical — having nothing to do with the physical make up of objects but rather can be used to describe the relationship between the two systems. Then, it turns out that there is a second way of measurement. That second way is, when you have two separate systems, to build a third system. Instead of requiring the first two systems to become integrated, you can quite independently unify their references in the third system while keeping the first two systems separated. Of course, I am still talking in conceptual terms (in this section) and one should not yet think here of implementation or attacks such as DoS, DDoS, CDoS, hackers, virus, etc. However, in implementation or attack terms, the third system needs to be trusted to the two systems, to some qualified extent, which is exactly that which defines the functionality limits of the third system at each point in time.

This is a systemic way of looking at data. We can say that data is not a figment of my imagination but that there is something physical in it that obeys physical (i.e., natural) laws. And that such laws may provide for theorems, building blocks, and concepts that we can use for our inferences, for what we can logically deduce. I would maintain that a data system can thus be described in both a classical way (as absolute data) but also as a quantum system where we see the interaction of data in terms of differences between two states, coherence, stimulated emission and so on. So the idea behind all this is that data is not just a figment of our imagination but is a physical entity obeying physical laws yet to be discovered — I am just scratching the surface here. It is thus perhaps time to put Science into Computer Science — to link various disciplines with one another at such interfaces and leverage for example, from the work of Einstein, Pauli, Schroedinger and Caratheodory. Then, computer protocols would not be just ad hoc recipes but actually result from the application of real world models.

COOK Report: Are you advocating the development of some kind of protocol system to enable people to negotiate with each other their ability to handle these kinds of data?

Gerck: Protocols today are based on "feeling." There are no models today for the Internet protocols. People simply "feel" or imagine that they should be done a certain way. Suppose we were to go back a thousand years. Gravitational laws were unknown. We would watch an apple fall but we would be at a loss to be able to explain exactly what it was that dragged it downwards. We could not derive or extrapolate, under such circumstances, any other useful information from the observed fact that objects fall.

COOK Report: Are you saying then that at one level you do have the mathematics to demonstrate your ideas about the internet transporting data in a quantum mechanical like way under certain conditions but that you have not worked out all the mathematics for working with data under all conditions?

Gerck: What I am saying is that if you exclude what I have been able to figure out in my limited capacity, the world views data solely as an abstract concept that can be whatever the abstractor wants it to be. That data is nothing more and nothing less than a string of abstract symbols, nowadays. But what I say to all this is "no." According to the theory that I have developed, data is a natural quantity that obeys natural physical laws in the same way that other things we see in our four dimensional world (three dimensions of space and one of time) must obey physical laws.

It may seem, however, a bit bold to say that data has inherent natural properties apart from that which shrouds the data. But, I am not talking about properties of data itself and by itself (which would be absolute, and non-natural) but about the *relationships* that data must obey (which are relative, and natural). Clearly, there are no natural data laws for which data is absolute, in my studies. To posit that data is not absolute is in fact the 0-th natural law of data, so to say. Another natural law of data is my observation that *any* process of data exchange must fall within one of three process, exclusively: absorption, spontaneous emission or stimulated emission -- with their respective rate equations as outlined in my essay. Now, when I say *any* process of data exchange I mean not only the Internet, but also radiowaves, paper, phone conversations, voice and even (if it exists) telepathic communication. Thus, irrespective of how data is embodied, its relationships will obey those rate equations -- and this is an example of a natural law obeyed by data, in relationships.

COOK Report: And you have some of the mathematics worked out to demonstrate this?

Gerck: Only some. This is a whole new frontier of exploration. Compared to what I think may be out there I am presently just scratching the surface right now. The essential thing that I have seen so far is that data is not just an abstract concept. Data obeys natural laws that I do not control.

COOK Report: Doesn't some of this go back to the work of Claude Shannon?

Gerck: Yes. Shannon was able to explore some of these ideas. However, he was starting from an ad hoc or arbitrary set of assumptions. But he soon found a very important law - the law for the capacity of a data channel. He was the one who said: "look — a data channel has a finite capacity. We cannot stuff as much information as we want to into any given data channel. We are limited by bandwidth, by the power applied and by the noise of the system." If you have a channel of so many megahertz and a transmitting power of so many watts, there is the maximum amount of data you can stuff through it, for a given noise level, in a given encoding. So channel capacity is a physical law of data for information.

COOK Report: One may say that you seem to undercut your own "data" theory, by requiring that data exists as a relationship or quantity as defined strictly with reference to two states---one state being one computer, and the other state being the other computer. As such, the qualities of "data," must be defined by the specific computers. However, the logical conclusion would be rather that "data" is an artifact of the specific computers and the physical transmission system constraints.

Gerck: In my theory, the concept of data is defined as "the difference D2 ^ D1 that can be measured between two states of data systems". So, the notion of measurent is very relevant here -- it is a difference that you can measure, not any difference, not a figment of your imagination. Note also that the data systems do not even have to be computers -- can be anything (paper, phone, voice, etc.). However, if you regard data in the classical way as "information in some form that can be digitally transmitted or processed," where would this take you? In this view, data is indeed an artifact of computers and transmission systems, an absolute quantity in each computer.

The Internet has revealed that data systems can completely falsify and change the information transported from one Internet host to another, without any possibility whatsoever of experimental detection over the communication channel used ^ which introduces the concept of trust in secure communication systems. Thus, *measuring* that difference D2 - D1 between two states may well reveal a *different* value of data, when compared with the absolute value assigned to it at a computer. In short, my definition of data takes into account both that data is always relative to two (not one, not three, not four, ...) states and that data is defined by measurement. These two predicates are essential to reproduce what we see in the real-world, and that is why this data theory is IMO very powerful.

So What Should Be Done?

COOK Report: Could we conclude by asking what all this implies that current policy makers who are reading this material right now should be thinking about doing over the period of the next year?

Gerck: Two hundred years ago people thought that heat was a fluid contained inside a solid. They thought that when you rubbed the solid the heat fluid would go out. When you rubbed your hands together they thought the fluid called heat would go out of the pores of your hands and create on the skin the sensation of warmth. This was known as the caloric theory of heat. Heat was thought to be a caloric fluid contained inside of bodies.

They started to do experiments that would prove this theory. In one such experiment cannons were being bored in a water tank. As horses went around the water tank to propel the device boring the hole, the water would eventually boil! They finally figured out that the heat was caused by the transfer of energy from the horses to the water and had nothing to do with the cannon itself. When people realized that heat was a form of energy that could be added to a body rather than a quality that was somehow intrinsic to the body, the whole situation changed.

The analogy is that today people think that data is an absolute value. This is why they think about hierarchical systems and about rules of reference. This can be useful in the three dimensional world where we can see a reference with its measurement. This gives us the apparent simplicity of switching data as an absolute variable. However in cyberspace we can have no absolute law or references. There is no path down which everyone can peer at an almost limitless basis. There are no walls and no physical references. As a consequence, in Cyberspace these analogies absolutely break down. Data can no longer be predefined as an absolute variable. Data has to be seen in physical terms, namely as the relative difference between two states. Consequently all the policies being written nowadays and all the work from certification authorities all rely on the completion of a reference - namely the starting point of the root hierarchy.

Now I was recently at UCLA at a meeting where they were talking about governance models of corporations and they were defending what they called the adult supervision model. After they finished their presentation I asked the first question saying that while I had heard an interesting theory I had never heard three special words: "separation of powers." We cannot all be jury judge and executioner. We all need to have separation of powers — namely the first well known three, the fourth being the people and the fifth a moderator. While the separation of power applies, it is never complete because there is always an overlap. So, if we do have

separate attributes of power and checks and balances, then we can have a relative control system and we do not have to answer the absolute control question: who supervises the adults. As any teenager can tell you, this absolute control question has no valid answer. The same applies to the absolute control system used today for data — from the DNS root system to the CA root in a PKI to a network administrator in the Internet.

COOK Report: Looking at the current dilemmas presented by what is perceived to be this denial of service, we are looking at something that likely has been totally misunderstood. If one begins to understand it in the terms of this discussion, then what does one have to do? It certainly makes no sense to station an FBI agent or other legal enforcer at every terminal sever!

Gerck: That is why I wrote in my earlier essay that making real or imagined transgressions more criminal won't solve anything. What we need to recognize is that no longer can security be based on confinement.

Security May No Longer Be Based on Confinement

COOK Report: Because the system is too big, too diverse, and too complex for such policies and subject to too many inputs, outputs and other variables to solve any real problems by confinement?

Gerck: And if you confine to make yourself secure, you are actually making yourself more of a potential victim. Here is an actual analogy from Japan where children were being kept in an apartment isolated from the ground and very much isolated from germs in a sterile environment. Whenever they did have a contact with someone who had been onto the dirt or the ground, they would get sick. Their bodies had been so isolated for so long they did not understand the micro-organisms with which the rest of us must cope. When the security of their confinement was breached, the outcome was much worse than it otherwise would have been. This shows that when security is based on confinement, a security breach becomes almost inevitable. The most sound basis on which to establish one's security is to understand the complexities of the environment in which one is working. This is true for two reasons. Maintaining isolation inhibits the ability to increase functionality. Indeed the very reason for having security is to be able to preserve one's functionality. So to produce security from the confinement that breaks functionality is completely illogical when the very reason for the security in the first place was to preserve functionality.

COOK Report: Isolation also mitigates against the diversity that is needed for the continued maintenance of system health because you said that when some of these errors come into effect, if we have end-to-end transparency on the internet and every machine were running the same operating system, there would be no barriers to the spread of the chaos that such an outbreak would unleash.

Gerck: Indeed the answer is not to isolate machines behind a firewall because the firewall itself then becomes the bottleneck for the denial of service. So the answer is not to isolate systems but rather to use the concepts of security based on an understanding of one's environment. In such a situation confinement can be a tool for achieving security but there it is by no means the total or only approach.

Some critics, tired of this strugle, say that the global internet should go back to its natural function --- which is "not business, commerce, and war, but rather education, debate, and exchange of non-monetized non-commercially critical ideas, and the furtherance of human emotion and connection and culture formation." But even if such were possible, should we give up progress there where we need most? In understanding the needs of security without using segregation or isolation?

Nowadays the object of security is to confine — which is paradoxical because, when you confine you reduce functionality. Because the goal of what you did was to maintain the functionality and not decrease it in any way. This leads me to my favorite way of expressing the paradox in that, to have my security, I have to break your privacy. In certification in order to have my security I have to know all your data. So the way today to cope with denial of service is to know the data about the service and about the owners. Thus the way to increase security is to break into privacy, but this is, again, a paradox because the purpose of security is to protect privacy. To paraphrase what Franklin might say today: those who prefer security over privacy deserve neither.

Conclusion

COOK Report: To sum up. Couldn't it be said that your analysis fails the test of Occam's razor—in that you have introduced in your new theory that some kind of *new* meta state of information has been reached, simply because the sheer number of connected host computers has passed a threshhold, which you say you predicted 3 years ago.

Gerck: First, such a conclusion is off the mark because I am not talking about a meta state of information. I am talking about the failure of the current empirical model of information to account for what we just observed in the distributed DoS as one example, and pointing out that a quantum model would do it with great elegance and *less* assumptions (i.e., only three basic processes). So, it is actually the current model that does not pass Ockam's test of simplicity — because it does not even work.

COOK Report: Here is something else that is not clear. I too am thinking that your analysis says the net behaves this way because of some threshold in sheer number of machines attached to it. Yet in conversations we have had, you have said it is only when some threshold number of machines is involved in a DoS attack or in a software bug incident that the thing goes quantum, coherent and so on. Which is it?

Gerck: My model says that information is *always* quantum. For a given system, the equations say that a least number of hosts in a prepared state is needed in order for coherent amplification (eg, distributed DoS, etc.) to be observed.

COOK Report: But a critic complained that you are positing that "the holy grail of artificial intelligence theory which is a step function based on some exponential number of information connections, has been demonstrated by the attacks on amazon/ebay several weeks ago."

Gerck: Let's not compare apples with speedboats. The holy grail of artificial intelligence is based on a non-quantum model for information.

COOK Report: The critic went on to say that you are also saying that the net has reached critical mass, in a sense consistent with the experience of atomic energy physics, or perhaps a plasma state in the sense of sustained nuclear fusion. At any rate, you are proposing that the net has just changed state from what it was, to what it will now be—and this "new" internet is bound by completely different laws of physics, because it has passed into a completely different state of information. The critic concludes that he is going to need to see a lot more experimental evidence, and a whole lot of math, to accept these conclusions.

Gerck: This is a good measure of skepticism.

Editor's Conclusion: We have found Ed Gerck's ideas especially challenging and fascinating. We see him as an "Internet Guy" who got here "the hard way" -- he's trained as a physicist, he thinks about the world from a perspective of how do you model the stuff you perceive around you in mathematical terms -- and this leads him to different ob-

Continued on page 30

In Two Essays Ed Gerck Looks at DNS as the Sole Handle of Internet Control and Explains Why the February DoS Attacks Were Coherent Rather than Distributed

Thinking, by **Ed Gerck**

[Editor's Introduction: Ed Gerck has a physics PhD from the Max Planck Institute. Since the mid 1980s he has focused extensively on questions of trusted modes of communication between digital systems. He is the co-founder of the Meta Certificate Group which has members in 28 countries (www.mcg.org.br) More recently he has become the CEO and VP of Technology of Safevote, Inc. (www.safevote.com), based in San Rafael, Calif. and the Chairman of the Board of the Internet Voting Technology Alliance (www.ivta.org), based in Washington, D.C. "Thinking", the first essay that follows, we believe to be the most intelligent and fertile analysis of the issues involving DNS and ICANN ever written. It is the most plausible explanation yet of what has happened and why. The second essay analyzing the DDoS attacks as result of an Internet turned quantum in its behavior is harder to follow, but as we have shown in our February 25 2000 interview with him about this second essay, we believe that its experimental verification may render it just as important.**]**

I have been quite successful in convincing others that there is nothing to be gained by opposing ICANN, because ICANN is just the overseer of problems to which we need a solution.

My point is that there is something basically wrong with the DNS and which precludes a fair solution — as I intend to show in the following text, the DNS design has a single handle of control which becomes its single point of failure. This needs to be overcome with another design, under a more comprehensive principle, but one which must also be backward-compatible with the DNS.

In reverse, others have also been quite successful in convincing me that indeed there is no alternative but to seek that solution which provides a fair code, a fair protocol to all. Including those that continue to oppose ICANN, irrespective of the results to be obtained.

It is my belief that the interests of small business, of big business, the public and that of governments would be better served by helping to seek a solution to the domain name issue, rather than by fighting the symptoms of the Domain Name Syndrome. The rhetoric around ICANN has been so intense though, and with such bad animus, that it requires a certain distance to see the forest for what it is.

I speak here as an individual, as a scientist, and as a netizen. I am also the CEO of Safevote, Inc., a co-founder and the coordinator of the MCG, a co-founder and Chairman of the Board of the IVTA, and a co-founder and coordinator the SRoot Workgroup.

So, the subject is domain names. The subject could also be Internet voting. But I will leave voting aside for a while. In my opinion, the subject, in a broader sense, is information control. If domain names could not be used for information control (as they can now by default under the DNS — see below), I posit that we would not have any problems with domain names.

But, domain names provide even more than mere information control — they provide for a single handle of control. DNS name registration is indeed the single but effective handle for information control in the Internet. No other handle is possible because:

o there is no distinction in the Internet between information providers and users (e.g., as the radio spectrum is controlled),

o there is no easily defined provider liability to control the dissemination of information (e.g., as advertisement and trademarks are controlled),

o there is no user confinement to control information access (e.g., as state or country borders in the Canadian Homolka case), etc.

But, how did we end up in this situation? After all, the Internet was founded under the idea of denying a single point of control — which can be seen also as a single point of failure.

The problem is that certain design choices in the evolution of the DNS, made long ago, have made users fully dependent on the DNS for certain critical Internet services. These design choices further strengthened the position of DNS name registration as the single handle of information control in the Internet. And, in the reverse argument, as its single point of failure.

Indeed, the DNS was never intended to be essential to the Internet, since all Internet hosts are accessible by their IP numbers alone. However, engineering choices in the design of the resource records and various e-mail protocols make it nowadays quite impossible for an average user to send or receive e-mail in the Internet without a DNS service. In short, DNS names have become the addresses of mailboxes and the addresses of e-mail forwarders in MX resource records. This is relevant in terms of failure and control analysis because the e-mail is by far, the most important application on the Internet for many users, businesses, e-commerce, governments and the academic community. In fact, the large majority of technical and policy discussions on Internet developments is done using e-mail and e-mail listservers. Thus, contrary to usual folklore in the Internet, the DNS is nowadays essential to Internet usage — as anyone can verify simply by trying to send an email to an IP number.

Still, some may still think that it is an overstatement, to highlight the importance of DNS as a technical "handle" for Internet control.

However, without the DNS there is no email service, search engines do not work, and webpage links fail. Since email accounts for perhaps 30% of Internet traffic — an old figure, it may be more nowadays — while search engines and links from other sites allow people to find out about web sites in about 85% of the cases (for each type, see http://www.mmgco.com/welcome/) I think it is actually an *understatement* to call the DNS a "handle." The DNS is the very face, hands and feet of the Internet. It is the primary interface for most users — that which people "see". Its importance is compounded by the "inertia" of such a large system to change. Any proposal to change the DNS, or BIND nameservers, or the DNS resolvers in browsers in any substantial way would be impractical.

Now it is indeed true that, I can send an email to an IP number by using a shell account

and telneting to port 25, while following the SMTP protocol, but it will fail if the other side does a "reverse DNS" verification and checks for my DNS name and I happen to have none in the arpa kludge (even though I may actually have it).

One of other fallacies in email is to ask the same system you do not trust (DNS, with the in-addr.arpa kludge) to check the name you do not trust (the DNS name), when doing an IP-check on a DNS name. There are more problems and they have just become more acute with the need to stop spam. Now administrators have begun to do a reverse DNS check by default. Under such circumstances you MUST have both DNS and IP.

Further, having witnessed the placing of decisions of network address assignment (IP numbers) together with DNS matters under the ruling of one private policy-setting company (ICANN), we see another example of uniting and making everything depend on what is, by design, separate. The needs of network traffic (IP) are independent of the needs of user services (DNS). They also serve different goals, and different customers. One is a pre-defined address space which can be bulk-assigned and even bulk-owned (you may own the right to use one IP, but not the right to a particular IP), the other is a much larger and open-ended name space which cannot be either bulk-assigned or bulk-owned. They do not belong together — they should not be treated together.

But, there are other examples. In fact, my full study conducted with participation of Einar Stefferud and others has so far catalogued more than forty-one essential problems caused by the current design of the DNS. Thus, a solution to current user wants is not to be reached simply by answering "on what" and "by whom" control is to be exerted, as presently done in all such discussions, without exception — for example, those led by ICANN. In this view, ICANN is not even the problem (as usually depicted by many) but simply the overseer of problems. At least, of 41+ main problems — all of which involve information control.

Thus by realizing both what these 41 and other problems are and the underlying issue of information control in the Internet (which issue is not ignored by governments), the study intended to lay the groundwork to provide for a collaborative solution to information flow in the Internet without the hindrance of these 41+ problems. The study also intends that the possibility of information control will be minimized as a design goal.

Indeed, if information control is reduced by design, a further benefit occurs. Since the danger that Internet information control might fall in the hands of criminals, monopolists or unfriendly powers is thus likewise reduced, governments may allow themselves to exert oversight to curb local abuse and do so mostly by applying existing local laws and rights — and not by trying to regulate worldwide Internet expansion. Thus, governments may face a task that is essentially doable and is based on collaborative behavior at many levels, very similar to traditional commerce, for example. By contrast, increased use of information control would increase the bounty for all and provide an ever stronger and powerful single handle of control, possession of which everyone wants more and more. But, in a case such as this, the handle becomes too "hot" to be handled in terms of all the liability gathered from exerting power over all. Those in charge find that the maintenance of such a handle cannot be viable unless it is made essentially unaccountable. As a result, the existence of the single control handle becomes an impossible problem in a fair system where liability is the counterpart of power.

And, unfortunately, this is what confronts us today — a phenomenon which I call "putting all our eggs in one basket".

What do I mean by "putting all our eggs in one basket"? The metaphor is precise, I mean putting all Internet management under one hood, one rule, one policy, one law. This, which at first might be thought as beneficial would actually establish a single handle of control to the local level of users and would go counter the very history of trade and law itself. There we do find value in diversity, we do find value in using different rules for different purposes, we do find value in separating governing powers so as to introduce a system of checks and balances, and we also do find value in different laws for different countries, for different states in one country an even for different communities in one state. Why would the Internet, being a much larger system, be different? Or, have different needs?

Of course, one may still argue that the "one basket" DNS approach has been made very solid. Indeed, that "one basket" is:

1. not market-accountable (it is regulated by a private non-profit company with no measurable market value in stock shares, for example),

2. not community-accountable (no elections),

3. not anti-trust accountable (it is a government appointed private company, and the government can only appoint one company),

4. not legally-accountable (has no assets to be put at stake; has no owner, has no profits to be seized),

5. allows registrars and registries to be also non-accountable with the Shared Registry Protocol — a TLD registry can say that it does not choose the registrar, the registrar was chosen and imposed; a TLD registrar can say that the rules make it just a pass-through service, and

6. makes end-users the only entity accountable to the entire system, the only ones that are actually controlled, the only ones that can be easily sued and the only ones that may lose anything (time, money and domain names).

In the above reasoning, such a finesse of seeking lack of accountability in all levels of name control can only be justified because it must have been presumed by its creators to be necessary for a central control system — the one basket. The fact, as discussed technically above, is that no one needs central control over name assignment in DNS (but only exclusively at the single root) is however a reason for the eggs not to be put into what has become an unfair trap. And, the more solid and unified the trap looks, the less reasons for any egg to be put there.

So, the issue is Internet and information control. Yes, certainly, people would bemoan a sloppy Internet. But they would scream long and loud if it were highly controlled all the way down to the local level. Unfortunately, they don't understand what they are looking for until they lose it. Thus, the issue is also user awareness. A perfect technical solution to a non-perceived problem is a perfect market flop.

Regarding "time" — readers may ask what is the schedule to propose new standards based on what I and my group are working on for domain names? As I see it and as I also comment in regard to the work on advancing standards for Internet voting at the IVTA (where IMO the same principles apply), time is not a trigger for the events needed to get us out of our predicament, but understanding is. Cooperation has its own dynamics and we must allow for things to gel, naturally. We can motivate, we can be proactive but we must not be dominating. We seek collaboration, not domination. Both technically as well as market-wise.

Regarding technology, the objective is to provide for unification without integration. In other words, to allow for local diversity instead of ironing it out in the good name of "interoperation" or "global rules". There are no global rules and this absence of rules is the only global rule we may accept ;-)

This essay, in spite of its length, will leave for the reader the task to infer its much greater significance in regard to the use of DNS as a political and legal "handle" for control — that unsavory interests or unfriendly powers may be able to grab a hold

of and exploit. I leave this thinking part to the reader — hence the name of this essay — because certain things are perhaps better thought and reflected than said. In this regard, my central observation is that the political and legal handle would simply not even exist if the technical handle did not exist. This is actually the sum of what I am trying to convey. "To grab a hold of and exploit" is not possible for anyone who may wish to try if it cannot be "grabbed" — thus being intrinsically fair to all.

Comments?

Coherent Effects in Internet Security and Traffic by **Ed Gerck**

There are many comments nowadays on Internet security, especially in reference to the current Denial-of-Service (DoS) attacks, which have been called Distributed DoS or DDoS. I believe that the commentators have so far missed some critical elements of the situation, in spite of a thorough analysis.

First, this was not only a DDoS — this was a CDoS. A Coherent Denial of Service attack. The difference is that a distributed but incoherent attack would not have done any major harm. In order to explain how such an attack was possible and why it was effective, one needs to understand first that, normally nothing is coherent in the Internet. All packets travel from source to destination in what may seem to be a random fashion; each host has unsynchronized time — oftentimes, even wrong time zones; and even the path traveled by each packet is also nondeterministic. Thus, achieving the coherent arrival of a stream of packets at one location by sending them from a large number of coordinated locations is a feat.

How this was done is what should be asked here — how such coherency could be achieved, in a network based on incoherency? The full answer is too long but I will summarize it here, to the best of my abilities in an email format, mostly as I have shown it some three years ago in a draft for Internet discussion groups. In experiments carried out in a university environment, I also have experimental proof that the Internet can sustain coherent traffic *amplification* after a certain critical number of hosts involved in the amplification process is reached. In such a situation a process called "stimulated emission" wins over "spontaneous emission". In other words, the Internet becomes closer to a quantum system as I intend to explain in the following paragraphs. I posit that this number was reached in the February 2000 distributed denial of service attacks (DDoS). Except that these attacks were actually a coherent DoS (CDoS).

In the model that I shall portray, we also understand that the Internet will start to show this behavior more and more as it expands to more hosts.

In summary, the press intuition is right on target here. What is new is that so many successful attacks were launched almost simultaneously. The reason for this is not coincidence or "chance." The reason is the onset of coherent effects in the Internet. The attacks show that a critical number of hosts has been reached. Consequently, we should expect new 'coincidences' and other modes of coherent attacks because even a low level of 'noise' (attacks) becomes sufficient to provide the seed for amplification.

I hope that these comments may provide a broader picture of understanding and a better action framework than current protocol-oriented after-the fact explanations being offered and may thus help devise real solutions to a mounting problem as I see it — the onset of coherent effects in Internet security. **[Editor's Note**: we ask our readers to perservere through the rest of this essay recalling the interview that precedes it. There we probed Ed Gerck with the questions that we had at the completion of our own reading of this essay. For us the impact of both this essay and the interview - done on February 25, - has been quite stimulating. We believe that it will be the same for our readers.**]**

In Search of Solutions to the Problem of Internet Security

In my opinion, the past days have once again confused identification with authentication. However, social engineering of loss of privacy as a "Good Thing" is not a solution to security problems. Privacy is a long term asset while security is a short term goal — one should not trade one for the other. The current paradigm that in order to have security I must break your privacy seems to approach and perhaps transgress the limit of what you are willing to forfeit to an stranger.

The motivation for this draft on the now surfacing coherent aspects of Internet security is thus also to call attention to the fact that everything is not "hopeless". Cicero said, as I may recall, the principle that "in war, law is silent." While some advocate it as the "only solution," I contend that it does not really apply to the Internet "right now." From the point of view of plain common sense, law enforcement is not the only part of a solution here. We need also to understand and use the "enemy's" drive.

So, we need in the first case to protect privacy, since privacy may be the first casualty in this "hacker war" if we are not careful. But privacy is an essential liberty, the liberty to be ourselves — and must thus be preserved as the actual purpose of security. For what is security without privacy? A prison. As was once said: "They that can give up essential liberty to obtain a little temporary safety deserve neither liberty nor safety." - Ben Franklin, ca. 1784.

If we go on to consider Internet voting, we realize that here is a case where privacy *must* be protected — and arguments to justify losing voter privacy in the good name of security will simply be impossible for coming applications that may use some form of Internet voting. Such applications will indeed posit security as a *protection* of privacy — not as an enemy of privacy.

True, the Net has security problems and the recent DoS attacks may be seen as just one more example of a known fact. And, indeed, the current "technical" solution for e-commerce security is insurance — 20% of Internet credit-card transactions are bogus and half of this 20% result in the loss of real money by fraud. In order to tolerate such losses we use insurance. But, of course, this cannot go on in such laissez-faire attitude where the victims end up paying for their own misfortune, even though in small monthly amounts and even if they are just innocent by-standers.

Further, the "insurance solution" to Internet security cannot be acceptable for one-shot applications such as Internet voting — and we need to recall that the Internet was not even designed for one-shot applications. Whenever there is an Internet security outbreak, what follows is a sowing of fear, uncertainty and doubt (FUD) — a tactic first used against AMDAHL many years ago. Sowing FUD will not move us to a better position, though to outsiders it may seem "knowledgeable" and may look like "something is being done".

Another "solution" to Internet security might be to raise the stakes in committing a crime. This will not go very far, though, as 30 years of Internet history tells us. Fear of punishment only works to the extent that the probability of punishment is high, which is not by far the Internet case. And, even if it were, making crime more illegal has never stopped crime. Law is no substitute for Engineering, as Bruce Schneier (a cryptographer) has pointed out.

It is time, perhaps, for the Internet to abandon also the hand-waving arguments found in some security workgroups and commer-

cial software and start to model what the Internet is and how it works.

An Alternative Model

In this regard, I would like to present that model which I have developed some three years ago and which is able to predict, explain and provide a path for pro-active solutions to problems that will be more and more prevalent as the model itself predicts. As I have pointed out these problems are the increasing coherent effects in Internet security due to continuous Internet growth. At first sight, the model may seem more complicated than what it explains, but there is a limit for simplicity even for models. A model cannot be simpler than that which is required in order to convey reality, otherwise it will not be able to reflect reality.

My work in this model has confirmed some and also negated some of a number of parallel observations made in a more intuitive framework by Einar Stefferud. For example in his "Internet Paradigm" talks, Stefferud compares the Internet to an amplifier and comments the idea that in the Internet "1 + 1 = 1" in the sense that adding or decreasing users does not make much difference. However, while my model also predicts that the Internet is an 'amplifier', it finds that the linear vision breaks down after a certain "size" of the Internet of the Internet is reached. At such point the "amplifier" becomes quantum — coherent effects become possible and allow a very small fraction of users to control almost the entirety of Internet dynamics for a limited time. The consequences of this observation to Internet security can no longer be overlooked, as facts proved on February 8, 2000: see http://dailynews.yahoo.com/h/nm/20000208/ts/tech_hackers_2.html

The Internet is akin to an amplifier in the sense that it allows us to do "more" — whatever that is — more work, more confusion, more productivity, more loss of time, etc. In other words, the Internet "amplifies" information both in time as well as in space — whatever that information is.

The Internet also follows the mathematics of "1 + 1 = 1", and is similar to a plenum in this regard — it is like a "heat bath" as used in Physics. A heat bath can always exchange heat (+ or -) without changing its own temperature. In other words, 1 + 1 = 1 whether it is for a plenum, or a heat bath. I find that the Internet has grown to be so large that it can be regarded as a collective, a plenum, a heat bath. The increase or decrease in entropy (information, in Shannon's terms) is null when another host is added, or taken out.

However, there are other collective effects, ones which can only be described with Quantum Mechanics, that show that *one* small and localized change in the Internet heat bath can trigger a type of "laser wave" of coherent information — a coherent amplification. Under such conditions the Internet becomes a macro quantum-mechanical system.

Such phenomenon would require the right boundary conditions as well as the right initial conditions, plus a minimum number of entities. My research takes these requirements into with the introduction of three processes in order to describe, by means of first-order rate-equations, the state of Internet information content at any time:

- **emission**: when an entity sends information to the Internet. The rate of emission is a probability, which is a function of a cross-section A. That is to say that it is characteristic for the entity and also depends on the information content itself. However it does not depend on the amount of information present in the Internet and interacting with the entity (to our chagrin, sometimes ...)

- **absorption**: when an entity receives information from the Internet. The rate of absorption is a probability, which is a function of a cross-section B (that is characteristic for the entity and also depends on the information content itself) times the amount of information present in the Internet and interacting with the entity.

and the essentially quantum—mechanical process of

- **stimulated emission**: when an entity emits information that has a large degree of coherence with incoming information. The rate of stimulated emission is a probability, which is a function of a cross-section C (that is characteristic for the entity and also depends on the information content itself) times the amount of information present in the Internet and interacting with the entity.

The last process is exemplified by this essay and constitutes the *unique* aspect of the Internet vis a vis television and newspapers, since in the Internet case, the stimulated emission information is sent to the *same* medium and in the *same* interaction. The rate equation for information in the Internet can then be fully considered as a function of exchanges between the three processes above. Note that no other processes are needed in order to explain information dynamics in the Internet.

To set-up the full model, I just add the assumptions that:

- each entity can contribute within a characteristic but infinite number of degrees of freedom (i.e., subjects),

- some degrees of freedom have more/less losses due to the boundary conditions (i.e., akin to resonant frequencies)

The model then leads to a series of equations which say that if the number of entities N is large enough then:

1. adding or decreasing one agent does not significantly change the probability distribution as a whole at any location/degree-of-freedom (i.e., entity address and subject).

2. A number of M entities in cooperation can achieve a "macro-stimulated" process such that very large fluctuations in information can occur at any location/degree-of-freedom, even if M << N.

The equations explain the recurring nature of themes such as e-mail viruses of the "Good Times" type in mailing lists as well as the sudden traffic bursts that can be observed, both for location and/or degree-of-freedom. I have recordings of actual examples of sudden and large traffic oscillations, which seem to support the quantum model and will be further reported elsewhere.

Further, they provide guidance for effective marketing models, by showing that the Internet marketing model is very different from TV or radio marketing models (which are not quantum, but classical). In the Internet, for example, it is necessary to avoid media saturation.

Another application is for Internet governance principles, because self-governance models can be disrupted by coherent actions by M entities, even when M << N. This contradicts the "plenum principle" and even the "peer-to-peer" expectations we might have. Some entities can be "more excellent" than others ... if they cooperate. This can be both very good and very bad.

Further, the model shows that slow phase transitions of Internet behavior are to be expected until N < Nc (where Nc is a critical number of entities) but that after N > Nc then quantum behavior can set in as a function of the right conditions — which can lead to sudden "oscillations" and transients. To exemplify the consequences, the model says that after a certain level of number of entities Nc is reached, Internet traffic will no longer follow linear extrapolations and that much larger dynamic windows of bandwidth may be necessary to accommodate traffic within a quality of service target. Unfortunately this flies in the face of current backbone/ISP traffic rules. The solution is to devise quantum models that may prevent unwanted stray oscillations.

In more general terms, what my model describes is a "coherent avalanche effect" —

the Internet is able to amplify, coherently, a given bit of information so that a very large (1,000,000 times or more) effect can be generated in a few "avalanche" steps of stimulated emission. Such emission will have coherent properties both in time as well as in space.

This, what I called "unwanted stray oscillations" in 1998, is what happened on February 8, 2000. As I have commented, "This can be very good and very bad ;-)". According to this model, the solution will not be to recognize attack patterns, nor to block hosts as in a "black list". The solution will be to develop tools that can detect and compensate for coherent effects in information transfer space-wise, time-wise and context-wise. Recall that these coherent effects can be caused by a concerted attack, by simple *spontaneous* amplification of a disturbance caused by error, by a software glitch, by natural oscillations in a large system, by simple bad luck, etc.

The idea of "safety in numbers" is gone, as the Internet passed that critical size Nc. We now can have large effects caused by a very small disturbance and this reality will begin to confront us more and more.

The problem is that such a small disturbance, though very effective, will, very likely, be as untraceable as that single photon which starts a laser emission many orders of magnitude more intense. Or, in a classical metaphor, as that fleck of snow that started the avalanche. We will find after the fact, that virtually all evidence is swamped by the sheer amount of data even if perfect logs are kept. A different approach is needed, as noted above. The Internet has reached another level in its growth and not all problems are nails any more, not all solutions have hammers. Since the nails have become waves. The solutions must likewise change.

Note: As a laser physicist who has become more and more involved with information security in the past 13 years, the above model seems natural to me in light of some of my published papers in both theoretical and applied quantum physics. The idea that the Internet has quantum properties may seem an exaggeration but it is based on a very basic observation — the Internet provides for "stimulated emission", which is a coherent process that has marked the very birth of quantum physics when Albert Einstein used the Planck quantum hypothesis in 1905 to explain the spectrum of black-body radiation, solving the paradox of the "UV catastrophe" and "saving" Physics from mounting internal contradictions. For some of my early references that relate to the large onset or large change of oscillations in quantum mechanical systems, caused by small disturbances in single-photon or multiple-photon interactions, please see:

E Gerck and A.B. d'Oliveira, "Continued fraction calculation of the eigenvalues of tridiagonal matrices arising from the Schroedinger equation", *J. of Comp. and Appl. Math.*, vol. 6, p. 81, 1980.

E. Gerck and A.B. d'Oliveira, "The three-body non-relativistic problem with potentials of the form K1.ln(r) + K2/r + C", *Rev. Bras. de Fisica*, vol. 10. p.405. 1980.

E. Gerck and E. Fill, "Blue-green atomic mercury photodissociation laser", *Proc. Int. Conf. on LASERS'80 - New Orleans*, STS Press, McLean, VA, p. 828, 1980.

E. Gerck, E. Fill and K.J. Witte, "Neue gepulste Laserlinien von Atomaren Metalien in sichtbarem Spektralbereich", in *Verhandlungen DPG* (VI) 16, p. 498, Germany, 1981.

E. Gerck, E. Fill and M. Irion, "Atomic mercury photodissociation laser", *Opt. Commun.*, vol. 37, p. 289, 1981.

E. Gerck and E. Fill, "Blue-green atomic photodissociation lasers in Group llb: Zn, Ds and Hg", *IEEE J. Quantum Electron*, vol. QE-17, p.2140, 1981.

E. Gerck and E. Fill, "1.3 u m excimer emission with I(2p1/2)", Proc. XII Int. Quantum Electron. Conf.-Munich, *Appl. Phys. B*, vol. B 28, p. 285, 1982.

E. Gerck and E. Fill, "Infrarot Fluoreszenz von Jod-Exzimeren" in *Verhandlungen DPG* (VI) 17, p. 448, Germany, 1982.

E. Gerck and E. Fill, "XeI excimer emission at 1.3 um", *Opt. Lett*, vol. 7, p. 25, 1982.

E. Gerck, "Collisional enhancing of the I(2p ∏) infrared emission by parent molecules CF3I, C2F5I, i-C3F7I and n-C3F7I", *Opt. Commun.*, vol 4l, p. 102, 1982.

E. Gerck, J.A.C. Gallas and A.B. d'Oliveira, "Solutions of the Schroedinger equation for bound states in closed form", *Phys. Rev. A*, Vol. 26, p. 662, 1982.

E. Gerck, "Quantenausbeute von 1(2p ∏) von Jodlasermedien bei 308 und 248 nm", in *Verhandlungen DPG* (VI) 18, p. 438, Germany, 1983.

E. Gerck. "Spektral aufgeloeste Emission von I(2p ∏) - Puffergas-Exziplexen bei 1,3um", in *Verhandlungen DPG* (VI) 18. p. 447, Germany, 1983.

E. Gerck, J.A.C. Gallas and R.F. O'Connell, "Scaling laws for Rydberg atoms in magnetic fields", *Phys. Rev. Lett*, vol. 50, p. 324, 1983.

E. Gerck, "Quantum yields of I(2p ∏) for CF3I, C2F5I, i-C3F7I, n-C3F7I, n-C6F13I and 1,2-C2F4I2 at 308 nm and 248 nm", *J. Chem. Phys.*, vol. 79, p. 311, 1983.

E. Gerck, J.A.C. Gallas and A.B. d'Oliveira, "A new approach to calculate bound state eigenvalues", *Rev. Bras. Fisica*, vol. 13, No.8, p. 183, 1983.

E. Gerck, A.B. d'Oliveira and H.F. de Carvalho, "Heavy barions as bound states of three quarks", *Il Nuovo Cimento*, p. 34-39, 1983.

E. Gerck and L.C.M. Miranda, "Quantum-well lasers tunable by long wavelength radiation", *Appl. Phys. Lett*, 1984.

Executive Summary

Understanding DDoS pp. 1 - 16

During the second week of February the largest, and most diverse denial of service attacks in the history of the Internet caught several of the most important commercial web sites off guard and exposed an previously unsuspected operational vulnerability that affects the entire commercial Internet. We contend that Gene Spafford's February 19th summation of the White House meeting provides a soothing but superficial explanation of what is really a far more subtle and difficult structural weakness, that is apparently inherent in the basic structure of the Internet. We present in Narrative form the NANOG and IETF technical discussions that resulted from the attacks. The discussion demonstrates that Internet backbone engineers are by no means agreed on precisely what happened or on how to deal with it..

On February 9, Lauren Weinstein, partner to Peter G. Neumann of the Risks mail list and co-sponsor with Neumann of People for Internet Responsibility had the following observation. " It seems apparent that the rush to move all manner of important or even critical commercial, medical, government, and other applications onto the Internet and Web has far outstripped the underlying reality of the existing Internet infrastructure. Compared with the overall robustness of the U.S. telephone system, the Internet is a second-class citizen when it comes to these kinds of vulnerabilities. Nor will simply throwing money at the Internet necessarily do much good in this regard. More bandwidth, additional servers, and faster routers--they'd still be open to sophisticated (and even not so sophisticated) attacks which could be triggered from one PC anywhere in the world. In the long run, major alterations will be needed in the fundamental structure of the Internet to even begin to get a handle on these sorts of problems, and a practical path to that goal still remains fuzzy at this time."

Ed Gerck's Ideas pp. 17-22, 30

Part Two of this issue contains an interview with Ed Gerck and two essays by him. He is co-founder of the Meta Certificate Group, http://mcg.org.br , CEO of Safevote, Inc. and Chairman of the IVTA.. The COOK Report suggests that his ideas form the basis for a fresh and compelling analysis of what we may really be dealing with. We conclude that there is a possibility that the fundamental nature of the attacks may have been completely misunderstood. We also contend that Gerck's theories, published here for the first time, may provide an entirely different mathematical basis for understanding the Internet as a quantum information structure possessing significantly different capabilities and potentials than could be extrapolated from our current understanding. Although this is quite a statement to make, his ideas have reached enough people so that it is likely that research will be rapidly undertaken to ascertain if his own experimental results dating from1998 are verifiable and reproducible.

Gerck asserts that the major reason the attacks were so successful is that the packets arrived at the target servers with a high degree of coherency - that is to say at almost the same instant. He points out that the technical functionality of the Internet mitigates against the coherent arrival of large numbers of packets at a specific target and thus a ten fold spike in incoming bandwidth would be very unlikely unless other unusual mechanisms are also at play."

How then could the observed effects of the arrival of very large numbers of packets have happened? He explains how his work in the quantum mechanics of lasers in the early 1980s gave him a hypothesis that he successfully tested in a university environment in 1998. Namely he suggests that the number of entities in the Internet has reached a critical mass where a single event such as a packet sent to a trin00 network, can result in an avalanche of coherent data amplification. The result is similar to the coherent amplification process that sets off the sudden flash of a laser. Under such conditions he posits that when this occurs, it creates conditions where packets can provide for a much different behavior as they reach a target. Gerck suggests that such events trigger a kind of quantum behavior, which however always exists but which then becomes visible at the user observed level and strongly contrasts with the classical behavior that it has replaced."

The scope of his ideas reaches to the root of what we call data. He suggests that data be thought of in terms of a natural quantity and as something that can be modeled with absorption, spontaneous emission and stimulated emission processes -- the last being a behavior associated with quantum systems. He finds that stimulated data emission can win out over spontaneous data emission. This will happen when a minimum threshold of affected systems is disturbed by what may be a hacker attack, or the interaction of a virus with multiple systems or even by the unexpected appearance of a bug in operating software that everyone assumes to be stable. His findings lead to the conclusion that such perturbations, resulting in web site and or network congestion, will happen with increasing frequency. Of course if he is right, when they do happen the next time, they may have absolutely nothing to do with hackers."

After compiling the technical discussion from NANOG and IETF, it seems to us that the emphasis on tradition security measures is rather futile. The Internet is too large with too many machines under too many levels of control for traditional security measures of confinement of people and machines to be effective.

Gerck has some very interesting ideas about constructing mechanisms where two parties which are not known to each other may use a third neutral environment in which to securely negotiate conditions of trusted operation. He seems to have an uncanny sense of political power and psychology and how to reflect this in technical situations to build trust between parties that have no common grounds for negotiation.

As recently as a week ago we intended to publish only his two essays. However when we called him on the 25th of February to ask for answers to questions about the second essay on coherency , we found ourselves in the midst of a far ranging discussion that opened up some of his ideas of the physics of data and mechanics of trust that we had not heard before. This discussion lead to the interview on pages 17 to 23. This interview which we expanded by asking several of our own experts to read and ask their own questions of Ed, begins to throw some light on the breadth and scope of his ideas.

Gerck's ideas lead to a paradigm change on such fundamental questions as data flow in the internet and the nature of security and trust in computer networking. Having a world view different from the prevailing gestalt often presents problems for everyone involved. We have known of Ed for perhaps almost two years and known him directly for six months. An unusual quality about him is that he is laid back. He is intuitive and skillful in dealing with people. His ideas may succeed precisely because he doesn't push too hard.

We have been a bit gun shy about walking out on the end of a limb on behalf of the ideas of someone who is not yet well known and whose views are iconoclastic. For the last few weeks we have made some serious efforts to get some sanity checks from people in better positions than we are to judge what he presents. Three very senior people have returned thumbs up. We introduced a forth such person with the strongest technical background of all to Gerck two weeks ago.

When we asked this person how we might describe Gerck in this newsletter he replied:

continued on next page

Continued from page 28

You might describe him as one of those bright people who are so frequently overlooked because he's happier working on hard problems than talking about it all. You might describe him as an Internet Guy who got here "the hard way" -- He's trained as a physicist. He thinks about the world from a perspective of how do you model the stuff you perceive around you in mathematical terms -- and this leads him to different observations than those made by those of us who "grew up" in the Internet and distributed computing in general."

One of the problems facing the Internet, is that we have, sometimes with chewing gum and bailing wire built it into something on which a very large proportion of our economy is riding. The prevailing opinion in the wake of the DDoS attacks is to call in law enforcement, build the security walls ever higher and hunker down with publicly reassuring words to the effect of don't worry we are in charge here. A careful reading of the technical discussion on pages 2 through 16 of this issue will show the that this position is founded on quicksand. A reading of the Gerck essays and interview will reinforce this conclusion

We contend that the official views issued in the aftermath of the White House meeting of February may be well intentioned. Nevertheless they are misguided. Without a correct diagnosis of our current problems, we will be unlikely to find solutions. As a result, the Internet's behavior of early February may become more rather than less commonplace.

Essays, pp. 23- 27

Thinking

We present roughly half of Ed Gerk's **Thinking** Essay in the belief that readers will begin to understand why we consider it the single best short essay on the topic of information control, DNS Governance and ICANN ever written.

"...there is nothing to be gained by opposing ICANN, because ICANN is just the overseer of problems to which we need a solution.

My point is that there is something basically wrong with the DNS and which precludes a fair solution — as I intend to show in the following text, the DNS design has a single handle of control which becomes its single point of failure. This needs to be overcome with another design, under a more comprehensive principle, but one which must also be backward-compatible with the DNS. [. . . .]

So, the subject is domain names. The subject could also be Internet voting. But I will leave voting aside for a while. In my opinion, the subject, in a broader sense, is information control. If domain names could not be used for information control (as they can now by default under the DNS — see below), I posit that we would not have any problems with domain names.

But, domain names provide even more than mere information control — they provide for a single handle of control. DNS name registration is indeed the single but effective handle for information control in the Internet. No other handle is possible because: (1) there is no distinction in the Internet between information providers and users (e.g., as the radio spectrum is controlled); (2) there is no easily defined provider liability to control the dissemination of information (e.g., as advertisement and trademarks are controlled); (3) there is no user confinement to control information access (e.g., as state or country borders in the Canadian Homolka case), etc.

But, how did we end up in this situation? After all, the Internet was founded under the idea of denying a single point of control — which can be seen also as a single point of failure. The problem is that certain design choices in the evolution of the DNS, made long ago, have made users fully dependent on the DNS for certain critical Internet services. These design choices further strengthened the position of DNS name registration as the single handle of information control in the Internet. And, in the reverse argument, as its single point of failure. [. . . .]

However, without the DNS there is no email service, search engines do not work, and webpage links fail. Since email accounts for perhaps 30% of Internet traffic — an old figure, it may be more nowadays — while search engines and links from other sites allow people to find out about web sites in about 85% of the cases (for each type, see http://www.mmgco.com/welcome/) I think it is actually an *understatement* to call the DNS a "handle." The DNS is the very face, hands and feet of the Internet. It is the primary interface for most users — that which people "see". Its importance is compounded by the "inertia" of such a large system to change. Any proposal to change the DNS, or BIND nameservers, or the DNS resolvers in browsers in any substantial way would be impractical.

[. . . .] One of other fallacies in email is to ask the same system you do not trust (DNS, with the in-addr.arpa kludge) to check the name you do not trust (the DNS name), when doing an IP-check on a DNS name. There are more problems and they have just become more acute with the need to stop spam. Now administrators have begun to do a reverse DNS check by default. Under such circumstances you MUST have both DNS and IP.

Further, having witnessed the placing of decisions of network address assignment (IP numbers) together with DNS matters under the ruling of one private policy-setting company (ICANN), we see another example of uniting and making everything depend on what is, by design, separate. The needs of network traffic (IP) are independent of the needs of user services (DNS). They also serve different goals, and different customers. One is a pre-defined address space which can be bulk-assigned and even bulk-owned (you may own the right to use one IP, but not the right to a particular IP), the other is a much larger and open-ended name space which cannot be either bulk-assigned or bulk-owned. They do not belong together — they should not be treated together.

But, there are other examples. In fact, my full study conducted with participation of Einar Stefferud and others has so far catalogued more than forty-one essential problems caused by the current design of the DNS. Thus, a solution to current user wants is not to be reached simply by answering "on what" and "by whom" control is to be exerted, as presently done in all such discussions, without exception — for example, those led by ICANN. In this view, ICANN is not even the problem (as usually depicted by many) but simply the overseer of problems. At least, of 41+ main problems — all of which involve information control.

Thus by realizing both what these 41 and other problems are and the underlying issue of information control in the Internet (which issue is not ignored by governments), the study intended to lay the groundwork to provide for a collaborative solution to information flow in the Internet without the hindrance of these 41+ problems. The study also intends that the possibility of information control will be minimized as a design goal. [. . . .]

Regarding "time" — readers may ask what is the schedule to propose new standards based on what I and my group are working on for domain names? As I see it and as I also comment in regard to the work on advancing standards for Internet voting at the IVTA (where IMO the same principles apply), time is not a trigger for the events needed to get us out of our predicament, but understanding is. Cooperation has its own dynamics and we must allow for things to gel, naturally. We can motivate, we can be proactive but we must not be dominating. We seek collaboration, not domination. Both technically as well as market-wise."

Continued on page 30

Continued from page 16

Karns: Fortunately, secure tunneling protocols will always make it possible for knowledgeable users to overcome these administrative restrictions and to keep the carriers down at the physical level where they belong, albeit with a loss in efficiency.

Elz [on February 17]: Well not always - it is possible to imagine an ISP that only permits connections via their proxies, and blocks everything else. They could even market it as a "premium extra secure internet service - no nasty packets from outside will ever reach your system" ... But aside from that nonsense, tunneling is not a problem for ingress filtering, it isn't defeating it (or not its purpose) at all. To tunnel, you need a remote tunnel endpoint - and then you can only send source addresses through the tunnel (and from there to the universe) with addresses from the remote-endpoint's address range (assuming ingress filtering is being done there as well).

That's all fine then, you're just acting as if you were a node at their location (which is the point, more or less), and packets you send that way are attributed to them, just the same as any other packet sent from there. If some site is willing to decapsulate packets from any random source and forward them, then they take the heat for any packets that are sent that way.

Continued from page 22

servations than those made by those who "grew up" in the Internet and distributed computing in general. Given greater exposure it seems likely that Gerck's ideas will be tested by experimentation. We'd like to see such tests rapidly proceed.

Continued from page 29

Coherent Effects in Internet Security and Traffic

Here is a paragraph from Gerck's second essay.

"This was not only a DDoS — this was a CDoS. A Coherent Denial of Service attack. The difference is that a distributed but incoherent attack would not have done any major harm. In order to explain how such an attack was possible and why it was effective, one needs to understand first that, normally nothing is coherent in the Internet. All packets travel from source to destination in what may seem to be a random fashion; each host has unsynchronized time — oftentimes, even wrong time zones; and even the path traveled by each packet is also non-deterministic. Thus, achieving the coherent arrival of a stream of packets at one location by sending them from a large number of coordinated locations is a feat.

The COOK Report on Internet
COOK Network Consultants
431 Greenway Ave.
Ewing, NJ 08618