Spam Zombies And Inbound Flows To Compromised Customer Systems

Joe St Sauver, Ph.D. (joe@uoregon.edu)
MAAWG Senior Technical Advisor

MAAWG General Meeting, San Diego
March 1st, 2005

http://darkwing.uoregon.edu/~joe/zombies.pdf

I. The Magnitude of the Spam Zombie Problem

Spam zombies are end-user systems (virtually always PCs running some version of Microsoft Windows) which have been compromised by parasitic malware.[1] That malware, once installed on the system, enables spammers to use the compromised system as a spam distribution channel without the knowledge of the system's owner.

Routing spam through spam zombies is now standard spammer practice because:

— it obfuscates the true identity of the spam sender, thereby hindering prosecution of the spammer under CAN-SPAM (or other anti-spam laws), while simultaneously virtually eliminating complaints to the spammer's ISP or upstream network carrier

— it allows a single spam source to be "smeared" or "spread out" across multiple IP addresses, thereby facilitating attempts to avoid per-dotted-quad rate limits, while also helping the spammer to maintain "fly under the radar,"[2] and

— it allows spammers to try to do an "end run" around DNSBLs or other filters which may be preventing direct mail delivery by the spammer.

When functioning as a spam zombie, the zombified system will often be slower than normal (or unstable and more likely to crash than normal), but there will generally be no other overt symptoms indicating that the system is being remotely accessed and controlled, at least when the system is observed by a typical non-technically-trained system owner.

For many novice system owners, the first indication they may have that their system has been converted into a spam zombie may be loss of network access, occurring when the customer's ISP takes the customer offline in response to spam complaints the ISP has received.

Technical news sources,[3] as well as popular mainstream media such as CNN,[4] have reported on the spam zombie phenomenon for some time now. It is a well established problem, and a serious one, with Sandvine estimating that **80% of all spam is now being sent via spam zombies.**[5]

At root, the fundamental problem is that end user systems, increasingly powerful and well networked,[6] are routinely and easily compromised by miscreants, and the user awareness, required software tools, technical expertise, motivation, time, and money required to keep the family PC properly maintained and under positive local control often isn't there.[7]

Thus, it should hardly be surprising that large fractions of the world's PCs are reportedly infected with some form of malware. Estimates vary from source to source, but a couple of examples are illustrative of the scale of the problem:

— AOL/National Cyber Security Alliance (October 2004):[8]

— 19% of all users had a virus on their computer

— 80% of all users had some form of spyware

— As of late February 2005, Panda Software was reporting[9] that roughly 35% of all PCs are virally infected (down from over 50% in November of 2004).

Remediation strategies for infected customer systems which might have been practical when a few percent of customers were infested may be deemed operationally (or financially) impracticable when the infection rate is much higher.

---

1. In this document, the term "malware" is used to encompass the full spectrum of viruses/worms/trojan horses/bots, etc.
2. It's routine at many ISPs to prioritize attention on the "hottest" [highest volume, or highest volume/unit time] spam sources when prioritizing/targeting anti-spam efforts.
3. "Rise of the Spam Zombies," SecurityFocus, Apr 25, 2003, http://www.securityfocus.com/news/4217
4. "Your Computer Could Be a 'Spam Zombie' -- New Loophole: Poorly Guarded Home Computers," Feb 17, 2004, http://www.cnn.com/2004/TECH/ptech/02/17/spam.zombies.ap/
5. "China: Please Delete," http://www.sandvine.com/news/article_detail.asp?ART_ID=314
6. Because malware slows systems down and consumes network bandwidth, infested users are good candidates for system upgrades and connection upgrades, which in turn means that abusers get yet more capacity at their disposal via poorly maintained hosts.
7. "Joe Average User Is In Trouble," Security Focus, Oct 22, 2003, http://www.securityfocus.com/columnists/193 and "Home PC Users Weigh Price of Protection," MSNBC, Nov 24, 2004, http://www.msnbc.msn.com/id/6560512/
8. "AOL/NCSA Online Safety Study," October 2004, http://www.staysafeonline.info/news/safety_study_v04.pdf
9. Panda Software Virus Infection Map, February 26th, http://www.pandasoftware.com/virus_info/map/map.htm

II. Hear *New Virus*? Think *New Spam Zombies*

You should also understand that while virally infected hosts may potentially be used for many different tasks (such as performing denial of service attacks, scanning other hosts for vulnerabilities, sniffing network traffic to try to capture passwords, etc.), the prime focus of many recent viruses is the conversion of end user hosts into spam zombies.

For example, if you consider the current top viruses as reported by Virus Bulletin,[10] the must important ones were:

Virus	*Prevalence per VB*	*Virus Spam Related?*
W32/Netsky	50.71%	Yes (anti-Bagle)[11]
W32/Sober	28.53%	Configurable[12]
W32/Zafi	8.84%	Configurable[13]
W32/Bagle	8.79%	Yes[14]
[others had prevelance <1%]		

The "use-viruses-to-make-spam-zombies" strategy is one that has received too little emphasis to date in part because most antivirus vendors have traditionally focussed on how to identify and remove a virus, or how it propagates, rather than emphasizing strategic analysis aimed at the bigger picture question of "*why* was this new virus created and released?"

In fairness, at least in some cases, the reason why antivirus companies haven't attempted to determine the purpose behind a particular virus's release may be related to the architecture the virus' authors used. Consider three potential approaches:

(i) *Purpose-built virus designed to create spam zombies:* In this approach, a viral payload is specifically programmed to turn a compromised host into a conduit for spam, usually by creating a Socks proxy or an HTTP proxy.[15] It is trivial to tag a virus of this sort as being intended to create spam zombies, however this sort of purpose-built virus is relatively uncommon today.

(ii) *General purpose "bots" which include other functionality as well as spam-zombie-related capabilities:* In this second approach, the viral payload may actually be a general purpose "bot," offering a variety of miscreant functions selectable by the miscreant as requirements arise "swiss army knife-style."[16] Because there is a corkscrew, fish hook disgorger, magnifying glass, tooth pick, tweezers and some tiny little scissors, it is easy to forget there's also a knife blade.

When you read or hear "bot" you should think "spam zombie" unless analysis of that bot specifically establishes that the particular bot does *not* include spam zombie functionality.

(iii) *Multistage viral systems:* In a third approach, the viral payload installed on a system may be just a "bootstrap" "loader" program, part of a multistage viral system whereby an undifferentiated/configurable initial virus gets distributed, only to subsequently download additional "plug-in modules" to tailor itself to the miscreant's then-current requirements.

Use of a multistage approach minimizes the size of the viral payload that needs to be directly distributed, while also ensuring that the miscreants' "inventory" of compromised systems will be running fresh code optimally appropriate to the miscreant's needs when ultimately enabled.

A multi-stage strategy, however, *also* allows the miscreant to obscure the ultimate purpose for which compromised hosts are being infected. In many cases, hosts compromised by a loader program will ultimately end up being turned into spam zombies.[17]

Thus, if you hear about a new multistage virus or downloader virus in circulation, think "spam zombie" unless events demonstrate that that's *not* the case.

III. Why Do Spammers Need *So Many* New Zombies?

Clearly, if indeed many viruses are designed primarily to create new spam zombies, and there are already a lot of existing virally infested hosts, why would spammers need still more?

Spammers constantly need fresh spam zombies for a variety of reasons, including:

(i) Spam zombies are a "wasting asset" whose value decays (and risk increases) over time, like cut up-but-uncooked chicken parts left to sit on the kitchen counter:

---

10. http://www.virusbtn.com/resources/malwareDirectory/prevalence/index.xml?current (as of February 26th, 2005)
11. http://www.pandasoftware.com/virus_info/encyclopedia/overview.aspx?lst=det&idvirus=46889 [to understand this, please note that competing crews of virus authors appear to be engaged in a "virus war" with each other, competing to create zombied hosts which they can then market to spammers]
12. http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=WORM%5FSOBER%2EI&VSect=T
13. http://www3.ca.com/securityadvisor/virusinfo/virus.aspx?id=41012 states "The worm accepts connections on port 8181 in order to download and execute files on infected systems."
14. "Wave of Bagle Worms Targeted Home Users," Information Week, November 2, 2004, http://informationweek.com/story/showArticle.jhtml?articleID=51202194
15. For example: http://www.f-secure.com/v-descs/bagle_m.shtml
16. See Lurhq's Phatbot writeup: http://www.lurhq.com/phatbot.html
17. See Lurhq's analysis of Sobig:
http://www.lurhq.com/sobig.html
http://www.lurhq.com/sobig-e.html and
http://www.lurhq.com/sobig-f.html

— the legitimate owner of the system may clean up the infection (e.g., she may download and install updated virus definitions)

— as particular spam zombies get used to send spam, those hosts end up getting listed on DNS black lists, with the result that eventually spam from those hosts ends up getting blocked at many sites which use DNSBLs

— providers themselves may disconnect or filter traffic from customer spam zombies after receiving complaints, again with the result that eventually spam doesn't go out

— due to inadequate access controls built into the zombieware, one spam gang may hijack another gang's spam zombies, thereby causing poor performance for the original spam gang, unless the original spam gang can successfully defend the zombies it has created

— it may be impossible for a remote spammer, having initially compromised a vulnerable host, to then perfectly secure that host against further compromises by other attackers; eventually, the cumulative weight of multiple successful compromises will likely cause stability issues and a loss of usability even for the spammer

— the older a spam zombie, the greater the risk that it has been instrumented to record the network activity of that host, with that intelligence then becoming available for provider (or law enforcement) action

(ii) Assuming that 80% of all spam is sent via spam zombies, there are several hundred active spammers (a conservative number, but one which jives well with Spamhaus' estimate of the number of hardcore spam gangs out there), and each of them wants to spread their traffic out over somewhere between ten and a thousand fresh spam zombies per day,[18] 365 days a year, that implies a potential demand for something on the order of:

0.8*200*10*365= 584,000 spam zombies/year to
0.8*200*1000*365=58,400,000 spam zombies/year.

IDC reported that 152 million new PCs were shipped worldwide in 2003,[19] so the zombification of roughly one third of those new systems would be sufficient to meet the 1000/day/spammer demand requirement estimate, assuming there are in fact only a few hundred spammers world wide and all new PCs are attached to the public Internet.

(iii) There are multiple competing sources creating spam zombies for resale to spammers. Each of those sources requires its own "inventory" of compromised hosts, and that redundancy/duplication inherently increases the aggregate compromised host count requirement.

(iv) The creation of spam zombies is not (yet) an exact science; the virus writers' attempt to employ a given exploit may create more spam zombies than are needed, or fewer than the miscreant had hoped. Because of these problems, and the comparatively crude "knobs" available to virus authors to try to control the rate at which zombies are created, more zombies might be created than are actually needed, or extra zombies might need to be intentionally created as "buffer stock."

By implication, then, you should view the problem of spam zombie creation as being a sustained one, and one which may well involve a large fraction of your customers' PCs.

FIXING INFESTED CUSTOMER HOSTS

IV. Typical Users Can No Long Self-Clean Infested Hosts

Traditionally, the assumption has been that when a typical user's system becomes infected with malware, they will be able to install and run an antivirus or antispyware product to clean up their system's infestation on their own. Unfortunately the emerging tendency is for malware to resist that sort of casual *in situ* attempt at removal.[20] Some strategies in use by virus authors include:

— *Anti-antivirus routines:* if any of a number of popular antivirus products is running, it is killed; if an attempt is made to relaunch an antivirus product, that's also prevented.[21] The antivirus product itself may be deleted or replaced with a copy of the malware.[22] Access to antivirus companies' web sites may also be blocked to prevent downloading of specific antiviral tools, updates to antivirus programs or access to manual malware removal instructions.[23] Obviously, interfering

18. Consider the quote from the pseudonymous author of *Inside The Spam Cartel* (Syngress, 2004, ISBN 1-932266-86-0) at page 33: "... an average spam run would never use just one proxy server. At the very least, I use ten and they have be very solid, newly found proxy servers that are not already in an RBL. (Ten is still a fairly low number, as I have used close to 300 before.) Generally, the more you use the better the results, as distributed spam will have fewer hosts blacklisted and more e-mails sent simultaneously."

19. http://www.technewsworld.com/story/32381.html
20. "RSA: Microsoft on 'rootkits': Be afraid. Be very afraid." http://www.nwfusion.com/news/2005/0217rsa-mic.html?nl
21. http://securityresponse.symantec.com/avcenter/venc/data/w32.klez.h@mm.html
22. http://securityresponse.symantec.com/avcenter/venc/data/w32.erkez.b@mm.html
23. http://securityresponse.symantec.com/avcenter/venc/data/w32.mydoom.b@mm.html

with normal antivirus products significantly complicates the clean up process for a novice user.

— *Loss of access to key system management tools:* the task manager (which would normally allow arbitrary malware-related processes to be manually killed) may no longer be able to be launched with ctrl-alt-del; the registry editor (which would often be used to remove some malware-created registry entries) may be closed as soon as you try to open it, etc.[24] Some malware may also remove these and other tools from the infested system entirely.[25] Loss of access to basic system administration tools significantly complicates the process of manually detecting and removing malware.

— *Persistence features:* if an attempt is made to kill some types of malware, it will immediately restart itself; if you succeed in killing the malware and keep it from immediately restarting, unless you get it fully removed from the system, it will reinstall and restart itself the next time you reboot.[26] In an effort to prevent erradication, the malware may stash multiple hidden copies of itself in various locations around the system so that if one copy is removed, the malware can reinstall itself from one of its redundant hidden backup copies; these copies may simply be saved as new normal files with innocuous names in obscure locations, or they may insert themselves into existing files,[27] be saved as ADS (alternate data stream) files,[28] be saved as zero-byte files,[29] etc. The presence of aggressive persistence features will greatly complicate the process of removing an infestation for novice users.

— *Destructive behavior:* finally, we should recognize that malware writers can simply elect to write code which may damage system components if/when it is rebooted[30] or otherwise modified or interfered with. This sort of malware "hostage taking" behavior would generate entire new categories of clean up challenges for non-technical users. While most modern viruses have not been destructive (successful parasites don't tend to kill their hosts), there have been examples in the past where viruses have overwritten hard drives or have damaged flash BIOS code stored in ROM.[31]

Vern Paxson and others have specifically flagged damage to hardware as a component of a worse case worm[32] which could potentially inflict fifty billion dollars in damages.

V. User "Nuke-and-Pave" May Not Be An Option

So if we assume that users may no longer be able to self-remove malware infestations, another strategy (and from the perspective of many professionals, the only approach which can restore some level of assured security to a system once it has been compromised) would be to have the user execute a so-called "nuke-and-pave" strategy:

— after carefully backing up any critical uninfested files which have not previously been backed up, the user would begin by formatting all hard drives on the system using the hard drive vendor's formatting utility (or a third party tool),

— the user would then continue by reinstalling the operating system from original media, including patching the system, configuring it to access the network and local peripherals, and installing suitable security software, and reinstalling all applications from original media (also patching those applications as may be required), and

— the final phase would involve restoring user files from a verified clean system backup.

Regrettably, this process, while more straightforward than battling removal-resistant malware, typically takes hours and would still likely implies a level of expertise that exceeds that possessed by a typical user. It also implies availability of original media, network access (to download patches, etc.) and a solid/current backup of user files, all of which may be absent in the case of a typical infested system.

VI. The Economics of Professional Virus Removal

Assuming users truly are unable to self-clean, what of bringing a system in for cleaning by a professional? Most professionals will recommend nuke-and-pave because of the difficulty and time involved in trying to remove one or more infestations from a given system, and because of the possibility that despite best efforts, when all that can be done has been done, full removal of the malware and stabilization of the system may not have been possible.

The issues of poor user file backups and poor access to original software media noted in the user-managed nuke-and-

24. http://securityresponse.symantec.com/avcenter/venc/data/w32.erkez.b@mm.html
25. http://securityresponse.symantec.com/avcenter/venc/data/w32.petch.b.html
26. One of the most persistent infestations is that associated with CoolWebSearch; see the discussion at http://www.spywareinfo.com/~merijn/cwschronicles.html
27. http://securityreponse.symantec.com/avcenter/venc/data/w32.magistr.24876@mm.html
28. http://www.windowsecurity.com/articles/Alternate_Data_Streams.html
29. http://securityresponse.symantec.com/avcenter/venc/data/w32.spybot.worm.html
30. http://securityresponse.symantec.com/avcenter/venc/data/js.gigger.a@mm.html (formats C drive if system is rebooted)
31. http://securityresponse.symantec.com/avcenter/venc/data/cih.html
http://securityreponse.symantec.com/avcenter/venc/data/w32.magistr.24876@mm.html
32. http://www.icir.org/vern/papers/worst-case-worm.WEIS04.pdf

pave case also pertain here, however an additional factor also enters the equation when a professional is attempting to clean an infested host, and that additional factor is the comparatively high cost of professional virus removal in comparison to the comparatively low cost of purchasing a new more powerful system.

For example, as I write this, a user could purchase a new Dell Dimension 2400 2.4GHz Celeron (or any of a number of similar name brand systems) for <$300 after rebates; if you consider the process of removing an infestation like the one described in the Washington Post this past August[33] which ended up costing that journalist $800 to remove, it would have been cheaper for her to simply replace her aging infested host outright -- but the economic rationality of that hard choice will no doubt be a difficult thing for some users to accept.[34]

SUPPRESSING THE SYMPTOMS ASSOCIATED WITH INFESTED HOSTS

VII. If We Can't Fix Infested Customer Hosts, Can We At Least Block Their Problematic Outputs?

While most of us would obviously prefer to get infested customer hosts cleaned up and secured, if that's not possible, can we at least block problematic output from those hosts?

For example, what about deploying port 25 filters at the network level, blocking all customer SMTP traffic except for mail sent via a small number of "blessed" (and closely monitored) mail servers, as many may already be doing? While that sort of strategy will certainly eliminate the problem of direct-to-MX spam from your customers, it has its own issues:

— you still need to throttle or block spam sent by your infested customers via your approved mail servers; failure to effectively do so may result in those key mail servers being blocked, with resulting disruption to many legitimate customer mail deliveries (and yes, spammers *are* working to insure that they will be ready to route their spam via your blessed SMTP servers when that's necessary)[35]

— While you may *hate* the thought of compromised customer systems being used to send spam, if you succeed in making it impossible for those compromised customer systems to be used to send spam, will you like it any better if those compromised customer systems are used to conduct denial of service attacks instead? (E.g., I think it would be a big mistake to think that the only thing that a compromised customer system could emit might would be spam on port 25...)

— The efficacy of port 25 blocks will depend in part on where they're implemented. If you implement port 25 blocks on your border routers, a spammer can still use an infested host to send spam *within* your network (inside your filter's boundary); if you implement port 25 blocks at the subnet level, the spammer can still use his compromised hosts to spam *intra-subnet*[36]

— Blocking port 25 traffic, while acceptable for dynamic residential customers, will typically not be well accepted for business class customers with static IP's.

— Output filters, like all perimeter-based blocks, assume that there will be no leakage via VPNs, no leakage due to politically mandated exceptions, etc.

At root, for all those reasons, I believe that strategies that focus on trying to control the ***outbound*** behavior of compromised customer systems will never be fully satisfactory.

IF WE CAN'T FIX INFESTED HOSTS, AND TRYING TO CONTROL OUTBOUND TRAFFIC ISN'T A SOUND STRATEGY, WHAT THEN SHOULD WE DO...?

VIII. Spam Zombies: Pipelines, Not Factories

When thinking about spam zombies, it is important to recognize that spam zombies are spam "pipelines," not spam "factories." Modulo leaks, there's conservation of volume — what comes out is what went in. A proxy is a nice example of this: output equals input.[37]

The alternative model, whereby spammers would use compromised hosts as spam "factories," largely isn't seen. Remember that in order to build and deliver spam in volume

33. "What A Tangled Web I Wove: Computer Naivete Cost Me a Bundle and a Bit of Sanity," http://www.washingtonpost.com/wp-dyn/articles/A64483-2004Aug14.html
"A Digital Doctor Treats Contamination," http://www.washingtonpost.com/wp-dyn/articles/A64481-2004Aug14.html

34. Note, too, the problem of what happens to those now-obsolete systems — do they get nuked-and-paved before being sold at a garage sale (or before they're sent to one's corporation's surplus property department)? Or do they get passed along to some new user with a largely intact hard drive that's "pre-infected" (and also a potential source of improperly divulged private information)?

35. http://news.zdnet.co.uk/internet/security/0,39020375,39186364,00.htm

36. Do not discount the potential issues associated with intra-subnet activity. For example, consider the "GetTunes" paradigm, which effectively circumvent peer to peer traffic shaping appliances deployed at subnet boundaries. (For more information on GetTunes, see http://sourceforge.net/projects/gettunes/)

37. Nice illustration of this at http://www.rsc-london.ac.uk/technical/network/monitoring/ at "Spotting open proxy servers."

these days, spammers need to use a host that has been provisioned with a variety of "raw materials" or "inputs:"

— a database of email addresses,

— the "creative text" that's being spamvertised,

— often a list of redirecting URLs that are to be used as part of the spamvertised text (it is so *passe* to spamvertise the name of your actual web site),

— target mail servers to route particular traffic through,

— DNS servers to use,

— obfuscatory text (such as content from online books) that can be included in an effort to defeat Bayesian contentfilters and insure that substantially identical spam isn't being generated[38]

— some reporting mechanism to insure that intelligence such as connect time rejections can be recovered

All those inputs are needed, plus a copy of the spamware itself (the "machinery" needed to combine all those inputs and actually generate spam for delivery from the spam "factory").

With all that "wholesome goodness" loaded up on a compromised customer host, the spammer could then begin making spam -- maybe. Unlike a nice tight little open proxy, maybe the compromised machine doesn't have the room to store all the required bits and pieces. Maybe the customer's machine is detected spamming and suddenly is subjected to forensic analysis -- now the good guys have a *whole bunch* of data to analyze about the spammer and his "production plans." Maybe some other spammer hijacks the factory and begins using it to make and send his spam, not the original spammer's spam.

For many reasons, the spam-zombie-as-spam-pipeline model is a lot more attractive to the bad guys than the spam-zombie-as-spam-factory model (even though the spam-zombie-as-spam-factory model would let the bad guys set up spam factory nodes, configure them to start pumping out spam after some interval, and then just walk away, never to go near them again -- but we know that the spammers DON'T do that).

So now that we know that spam zombies have a persistent (or periodically reoccurring) upstream logistics/command and control "tail," we now understand that the users of those zombies are vulnerable because of that connection.

38. Invariant spam messages are easily addressed by checksum based methods such as those used by DCC; see: http://www.rhyolite.com/anti-spam/dcc/

IX. What Could One Potentially Do With Information About Who Is Using Zombied Customer Systems?

Once you know that zombie exploiting spammers might be routinely identifiable by looking at upstream flows going into compromised customer boxes, assuming you could collect that sort of information, what might you do with it? You might:

— watch to see what other as-yet-undetected compromised customer hosts are also seeing TCP flows from the evil upstream zombie-stoking hosts, using the traffic analysis as a "finger pointing" tool...

— block access from the evil upstream zombie-stoking hosts, cutting the spammers off from "their" zombies...

— endeavor to get the ISP(s) hosting the evil upstream zombie-stoking hosts to terminate service to their abusers...

— seek civil remedies against the abusers, obtaining either monetary damages or injunctive relief...

— cooperate with law enforcement to pursue criminal sanctions against the abuser...

— or you could just publicize what you've found out.

This last approach, exposing upstream proxy abuse sources, is something that has been done previously by a number of parties based on proxypot data, including:

— Ron Guilmette's "WHO'S SPAMMING YOU?" "Top 40 Proxy-Hijacker-Friendly Nets" postings to Usenet circa August-September 2003 (Ron stopped collecting and sharing this data after being DDoS)

— Reports from the http://www.proxypot.org/reports website (Pacman's breakdown by host is fascinating)

While proxypot data can be surprisingly useful, data from compromised live customer hosts is even better, and potentially reaches a variety of abusers who may be careful enough to avoid proxypot deployments.

X. Prior Legal Review of Any Data Collection Project

CAUTION: Before undertaking <u>any</u> collection of data, and again before <u>any</u> usage or disclosure of data you may collect, I strongly, <u>STRONGLY</u> urge you to have legal counsel review and approve your proposed activity, both for consistency with all applicable laws and regulations (e.g., 18 U.S.C. 2511(2)(a)(i) and 18 U.S.C. 2511(2)(g)(iv) among others here in the U.S.), as well as for congruence your own corporate/institutional privacy policies.

XI. The Mechanics of Getting Flows For Analysis: Netflow

So assuming you wanted to do flow based traffic analysis on input traffic to compromised customers, how can you get the flow data you'll need? Several approaches are technically possible; choice of one or another is largely a matter of your network architecture, deployed gear, and personal preference.

One classic approach is to export network flow data from edge or core routers using vendor supplied tools such as Cisco's IOS Netflow,[39] perhaps in conjunction with popular open-source products such as flow-tools.[40] Netflow data is routinely used for billing, traffic engineering/peering analysis, performance monitoring[41] and security-related purposes.[42]

It is beyond the scope of this paper to describe Netflow in detail, however in a nutshell the process involves:

— one or more routers are configured to export flow data to a collector PC (hardware modifications to the network are not required)

— a collector PC receives flow data from the routers, saves the flow data to disk, and is then used to archive, summarize, or otherwise processes the flow datasets

Data that is typically available for each flow in an IOS Netflow dataset includes, for each flow:

— flow start time
— IP protocol
— IP type of service (TOS)
— source interface
— source IP
— source port
— source AS
[along with corresponding destination values]
— flow ending time
— flow duration
— total packets
— total bytes
[etc]

Obviously this may be far more data than is required (a key objective in doing inbound traffic analysis is insuring that you don't drown in unneeded/irrelevant data). There's also the simple reality that the more detailed data you save about a particular flow, the smaller the number of flows you can archive in a fixed amount of disk space.

Collecting netflow data also potentially has a processing impact on the router on which it is collected and exported. On routers with medium to high speed interfaces (OC3, OC12, GigE, OC48, OC192), sampling netflow may be needed, collecting only 1 packet in a 100, for example. For a variety of statistical reasons, sampling netflow as typically implemented is often not wholly satisfactory.[43] You may decide that you'd prefer to use an alternative approach that doesn't rely on routers exporting flow data at all.

XII. Passive Collection Methods

For example, you might elect to use a passive collection data collection approach instead, typically via an optical splitter. Passive methods have a number of advantages, including:

— potentially allowing collection of full packet-level data (rather than just flow summaries ala netflow)
— there's no router or line card CPU "hit"
— the splitters themselves are unpowered, have no electronics or mechanical parts, and are inexpensive enough to deploy on a ubiquitous "just in case" basis

The primary disadvantages of passive methods are:

— even moreso than was the case with netflow, the data volume needs to be carefully dealt with
— because full packet captures are possible, collection hosts need to be very carefully secured
— unlike Netflow, where collection can be commenced via remote software configuration changes, passive collection methods potentially require physical installation of splitters, a local collection host, etc.

Splitters are available from a variety of vendors including Netoptics,[44] Fiberdyne,[45] and most other fiber optic suppliers.

Those splitters in turn are often used in conjunction with DAG Cards from Endace,[46] which directly interface with tcpdump and other common network analysis and intrusion detection software (Snort, Bro, CoralReef, ntop, etc.)[47].

Metanetworks also has interesting gigabit and 10 gigabit hardware capture cards.[48]

---

39. http://www.cisco.com/go/netflow
40. http://www.splintered.net/sw/flow-tools/
41. http://netflow.internet2.edu/
42. http://www.cisco.com/warp/public/732/Tech/nmp/docs/netflow_security.pdf
43. "Building a Better Netflow," http://www.caida.org/outreach/papers/2004/betternetflow/betternetflow.pdf
44. http://www.netoptics.com/
45. http://www.fiberdyne.com/
46. http://www.endace.com/ (Cards are available to support the full range of interface speeds from T1 to OC192)
47. http://www.endace.com/products.htm
48. http://www.metanetworks.com/products.html

XIII. SYN's May Be Enough

For the purpose of identifying the upstream hosts that are hitting your compromised customer hosts, you may need neither full netflow records nor full packet captures; it may be enough to just capture SYN packets associated with TCP connection instantiation,[49]and as mentioned in the tcpdump man page, it is trivial to configure tcpdump to pass SYN packets only.[50] Because you're only saving a single line of data per connection, you should be able to routinely collect and store SYN data for all connections, even on loaded fast pipes.

XIV. Processing/Using The SYN Data

That sort of routine SYN collection process would then make it possible for you to execute the following strategy:

— SYNs get routinely collected and archived for all flows

— AOL scomp (spam complaint feedback loop reports)[51] complaints from Spamcop,[52] and other consistently formatted complaints are received and routinely summarized/tracked[53] to identify potential compromised hosts/spam zombies

— per-user outbound port 25 direct-to-MX connection levels are routinely tracked (or if you block random customer access to port 25, per-user mail volumes through your official SMTP server are logged), again with an eye toward identifying suspicious traffic levels

— high levels of AOL scomp/Spamcop reports, or high levels of outbound port 25 activity, triggers retrospective review of archived SYN data with respect to inbound flows to the local host that may be compromised

As you begin to look at those inbound flows, you'll probably see some interesting things. For example:

— with rare exception, the spammers who are feeding spam through the spam zombies are doing so from a comparatively small number of American and Canadian colos, not from overseas.[54] You may not have "long enough arms" to reach spammers with criminal or civil actions overseas, but domestic ISPs should not be a problem either on a legal or technical[55] level.

— Moreover, while the bad guys could potentially do proxy chaining,[56] they're are NOT -- they're hitting zombied customer systems directly. The upstream path to those abusing your customers is short and direct.[57]

— At any given time, the total number of IPs required to account for all upstream zombie feeding spammers is only the equivalent of a /23 or so, albeit scattered a few dotted quads here and a few dotted quads there. If you just wanted to block those IPs using technical means, you could easily do so.

When doing traffic analysis of this sort, the inbound flows from the zombie masters tend to stand out like a sore thumb; the only potential confounding traffic that you may need to differentially diagnose will tend to be associated with P2P-related flows, but they tend to have an entirely different pattern of dotted quads (largely consumer broadband, with rDNS, and often appearing as a mix of domestic and international sites; remember that the spammers stoking spam zombies will be colo'd hosts, usually without rDNS, and will be located at US and Canadian colos with only rare exceptions).

Sometimes as you look at slices of data for a compromised host you'll see:

— the host get zombied in the first place, and the "original" bad guys begin to use the host

— then, after a period of time, rival spammers apparently learn of the compromised host, perhaps by watching public reports of hosts sending spam. Knowing a host of interest, the rivals then scan that host to see what's open, and begin to also use that zombie.

---

49. Setting up a TCP connection involves sending a SYN, which gets replied to with an ACK, and which in turn is acknowledged by a SYN/ACK.
50. `tcpdump 'tcp[tcpflags] & (tcp-syn) != 0)'`
51. http://postmaster.info.aol.com/fbl/index.html
52. http://www.spamcop.net/
53. http://word-to-the-wise.com/scompfilter/ may serve as a basis for rewriting scomps to a more readily summarizable format
54. Using overseas connectivity for feeding spam zombies would generally be undesirable because of the extra latency it would introduces (bandwidth delay products can be problematic for chatty protocols, see http://www.psc.edu/networking/projects/tcptune), and because of the costs of trans-oceanic transit relative to what's available from discount domestic colo providers.
55. Technical action (filtering or rate limiting traffic from problematic prefixes, depeering security ambivalent or uncooperative ASN's, etc.) may be a "bigger stick" than threat of legal action, particularly for domestic American and Canadian providers.
56. "The Open Proxy Problem: Should I Worry About Half a Million Trivially Exploitable Hosts?" http://darkwing.uoregon.edu/~joe/jt-proxies/open-proxy-joint-techs.pdf (or .ppt) at slide 52
57. We can speculate about why spammers don't bother to proxy chain, but when you get right down to it, I suppose that they haven't done so because they haven't needed to do so, and proxy chaining when you don't need to introduces extra overhead, burns through spam zombies or open proxies at an accelerated rate, and multiplies the spammer's risk of being monitored or backtracked.

— the original miscreant "owner" then may make contact with the zombie via a separate command and control port, apparently reconfiguring the zombie to now talk on a different port. As the active port changes, the rival/random users magically go away (at least for a while), and the original upstream spammers return to enjoying their exclusive access.

You may find those apparent command and control hosts to be of special interest, although often they will come from hard-to-backtrack overseas blocks or be proxy chained.

From time to time as you watch inbound traffic, you may also see the good guys reflexively scanning the zombied host,[58] or anti-spammers scanning the compromised host trying to see what ports are open and being used. You should easily be able to identify these by verifying their rDNS; from that point forward it is easy enough to automatically ignore them.

XV. Other Traffic Analysis Notes

As you begin looking at traffic associated with spam zombies, in addition to the key idea of paying attention to the inbound hosts hitting your compromised customers, you may also find it interesting to "fingerprint" the outbound traffic of your zombied customers.

You'll quickly notice that some spammers only spam only a single ISP (such as AOL, or Hotmail); others may only spam overseas ISPs from your customer zombies.

If, for example, you are not set up to get AOL scomps, you may never know you have a compromised customer who is hitting AOL customers with spam.

Likewise, unless your network is instrumented, you may never know that you have zombied customer systems which are spamming certain overseas targets. (For some reason, some overseas spam targets seem to generate zero complaints, either because zero spam actually gets through to those spam targets, or perhaps because of language issues or a lack of familiarity with American and Canadian abuse handling/no user-level knowledge about how to complain).

Some spammers generate a tremendous volume of DNS queries via the zombie(s) they're using; others may generate virtually none. Based on what you see for your zombied customers, you may also want to take appropriate action against anomalous DNS traffic, just as you may limit port 25 direct-to-MX traffic, the rebuttable assumption being that "normal" customers will use your official DNS servers. Obviously customers who intentionally run their own DNS servers will have different traffic patterns, and will need to be accommodated just as business class customers may be granted the ability to send port 25 traffic directly.

"I UNDERSTAND THAT DOING INBOUND FLOW ANALYSIS OF MY ZOMBIED CUSTOMERS MAY REDUCE MY <u>OUTBOUND</u> SPAM LEVELS, BUT WHAT WILL IT DO TO HELP <u>INBOUND</u> SPAM LEVELS?"

XVI. Local Actions, General Impacts; It's *Not* All About Inbound Spam

While it is obviously good to avoid having compromised customers actively spewing, your efforts to "look upstream" and deal with the command and control hosts stoking your own customer zombied hosts will likely *also* impact spam zombies running at other sites, including spam zombies spewing at your customers.

That is, the zombie command and control host driving your zombied customer likely was not dedicated just to your network; it was probably also driving other zombies, including some that were likely spamming you and your customers. If you get enough of those zombie C&C hosts torn down, or their operators arrested, the overall level of spam for everyone will float down, including for you and your customers.

Similarly, if you get a reputation for being well instrumented and aggressive in dealing with upstream hosts hitting customer zombies, spammers may retarget their efforts at other less well instrumented ISPs. Thus, even if examination of inbound flows to compromised customer hosts does not cause a direct reduction in inbound spam levels, you may at least see fewer compromised customers, fewer support calls, lower customer churn, happier abuse desk employees, etc. Those are, again, all worthwhile objectives, even if they are not associated with a direct reduction in inbound spam levels.

XVII. Conclusion

We hope that this discussion has at least served to stimulate your thinking when it comes to dealing with the 80% of all spam that's currently being sent via zombied customer hosts.

58. http://scurity.rr.com/probing.htm