

[Full-Disclosure] Swen Really Sucks

Source: <http://www.derkeiler.com/Mailing-Lists/Full-Disclosure/2003-09/1458.html>

From: Jason Coombs (jasonc_at_science.org)

Date: 09/24/03

To: <full-disclosure@lists.netsys.com>

Date: Tue, 23 Sep 2003 16:49:35 -1000

So, Swen/Gibe.F uses NNTP to scrape newsgoogleroups (a.k.a. usegooglenet, formerly known as usenet newsgroups) for new e-mail addresses to which to spam copies of itself.

For those of us lucky enough to have e-mail addresses inside domains we control pop up in a newsgoogleroup posting recently, Swen/Gibe.F is still throwing at least one virus e-mail per minute at our SMTP servers...

Personally, I take all e-mail seriously. I try to respond to every message that was typed by a human being directed at me or at any organization I represent. Since Friday I've personally received about 10,000 copies of Swen/Gibe.F and I'm not alone, others are experiencing the same thing.

The question is this: what do we do about it?

We pay for bandwidth by the byte. We're presently paying to receive these viruses. My inbox now has nearly a gigabyte of viruses in it, since Friday, and I'm ready to have the attack end already.

Swen/Gibe.F randomizes the From: and To: addresses, as all good viruses do, and the end-user with the infected Windows box randomizes the source IP address thanks to the twin technological miracles of dial-up Internet service and DHCP. Therefore, no IP, e-mail, or domain filter will solve the problem completely without filtering every single possible permutation of From: address that the virus spits out... and using the "From" address rather than the "From:" address for the filter doesn't work, either, because the "From" address appears to be a different non-randomized e-mail address, possibly the real e-mail address of the infected victim (? haven't read any forensic analysis on this point yet...)

Since filtering isn't an option, and contacting the infected parties to stop the onslaught isn't viable, we're left with a couple alternatives:

- 1) Install a virus scanner on the SMTP server and tell it to reject all "infected" messages. This doesn't solve the bandwidth cost problem, and it adds another cost on top of the bandwidth cost. It also rejects all false positives, and it prevents one from engaging in infosec research where it is

Full-Disclosure: [Full-Disclosure] Swen Really Sucks

essential to be able to send and receive malware and certain scripts and source code of value to penetration testing—mail without having to smuggle it in passed the border guard using non-executable compressed attachments.

2) Permanently turn off the e-mail addresses that the virus finds and to which it starts sending e-mail copies of itself. Redirecting this e-mail to /dev/null would be great. Bouncing the e-mail messages would be terrible, since the sender isn't a real sender this would result in our SMTP server spreading new potential infections to any address that the virus comes up with or finds in the wild. This is a non-starter for important e-mail addresses that have been in use for almost a decade, like my own e-mail address.

3) Pay the double bandwidth cost of redirecting infected messages to secure@microsoft.com so that Microsoft's superior knowledge, experience, industry contacts, and law enforcement leverage can track down the infected hosts and shut them off.

We choose option #3. The ISP to whom we outsource our SMTP service doesn't give us the option of redirecting to /dev/null ... we have to bounce the message or eat the message. Bouncing would result in greater harm; while eating the message means that I personally am out of business — unable to use e-mail without changing my e-mail address — because there isn't enough time in the day for me to manually purge my inbox of virus mail in order to find the three messages that human beings sent to me today. Redirecting the virus e-mails to another e-mail address other than my own, to the extent that this can even be done thanks to the fact that many of them are not being addressed to my personal e-mail address but to some role account or some troll address that a random spammer or maker of straw men used arbitrarily in a frivolous posting, would clean up some of the clutter of my personal inbox, but do nothing to solve the DoS problem we're having due to the hundreds of megabytes of unplanned SMTP traffic sitting on our limited-quota hard disk space provided to us by our ISP... We've already paid to expand our allotted hard disk quota. How much more paying are we supposed to do according to "best practices" in the industry before we have a cause of legal action against the manufacturer of the defective product(s) that allow this to happen in the first place?

This whole situation sucks, and I want my Friday, Saturday, Sunday, Monday, and Tuesday back, please. Who do I have to sue in order to get lost days of my life back? People who knowingly write and release insecure software products should be put in small, wet, insect-infested holes and made to eat Hormel brand processed meat byproducts from a can. They are far worse than virus authors because at least virus authors have the common decency to HIDE — many software vendors just stick their middle finger out at you, call the FBI to start a malicious and abusive prosecution, and then drive home in their luxury cars to their immense castles and their privileged stock option-fueled lifestyles...

Something needs to be done to stop the vicious cycle of abuse that flawed software products cause. And it's not "put the virus authors in jail"... It is also definitely not "force everyone to buy antivirus software" <- antivirus

Full-Disclosure: [Full-Disclosure] Swen Really Sucks

software is the second-biggest scam in the computer industry and it gets used as a defense against software vendor product liability. Antivirus software needs to be stopped completely. What we need are software products that don't allow uncontrollable abuse.

Sincerely,

Jason Coombs
jasonc@science.org

Full-Disclosure – We believe in it.

Charter: <http://lists.netsys.com/full-disclosure-charter.html>