The Lurking Threat of Malware

Monday, July 26th, 2004
by Victor Oppleman

If you're part of the information security industry in any way, you probably remember threats like *Slammer*, *Nachi*, *SoBig*, and *Blaster*. They impaired and disrupted networks globally infecting tens of thousands of systems and spreading like wildfire. Slammer, in particular, doubled in size every 8.5 seconds and saturated points on the Internet like nothing before it, reaching global Internet saturation in less than fifteen minutes (as corroborated by CAIDA's independent analysis). This unprecedented speed of infection guaranteed that if you were vulnerable to this type of attack from the Internet, you were infected within fifteen minutes—long before any anti-virus update or patch could possibly be made available (and probably before you even heard of the problem). Many organizations that weren't immediately vulnerable from the Internet found out they were vulnerable within their own organization as consultants plugged into their local area networks or when employees brought disks in from home or opened e-mails containing these and other threats. Of course, this kind of activity continues today with old threats and new ones. Vendors continue working together to provide fixes and patches and many of us, infected or not, deploy these patches quickly as a reactive or preventive measure.

What many experienced security professionals don't know is that these threats or their more dangerous derivatives were, in many cases, merely a confusing first step in a larger and well thought out plan. Some of these "threats" were purpose engineered, some just happened to fit a mold and were easily copied and repurposed for a grander scheme. The attackers' goal was two-fold: 1) anonymously find vulnerable systems and cause those systems to come back to them and 2) create a degree of separation and anonymity between the attacker and the infected systems while introducing the real (second) threat payload. Since the worms propagated so well and found other hosts with which to spread, they were everywhere in no time at all. What security engineers didn't consider until some time later was that their scanning and spreading algorithms essentially notified a great number of listening posts—purposely-compromised hosts all over the Internet that were waiting to see infected systems' discovery packets (propagation attempts), identifying (by specific IP address) the vulnerable/infected systems. Put simply, the infection was in many cases, simply a beacon that alerted the attackers to vulnerable systems. This is best illustrated through a simple example: Your system was infected with some worm, so it started sending data all over the Internet attempting to infect other hosts ("scanning"). While your computer was scanning, it likely passed by one or more of the listening posts that had been previously set up by the attackers and that was programmed to receive the beacon and then inject the second, more sophisticated software agent onto your systems. There was little chance that you'd notice this second wave of attack since all the news stories centered on the virus or worm threat and not the follow-on threat. Everyone was so focused on the worm or viral delivery mechanism; the entire industry almost completely missed the real, lurking threat: the stealthy delivery of intelligent malware through a sophisticated, multi-phase attack.

So, what does this second phase malware do? It usually includes software that spawns elusive, automatically-upgrading agents (basically Trojans) that are now ravaging the Internet and are being remotely controlled by remarkably sophisticated miscreants. I say elusive because these attackers are very good at hiding their code. The only easy way to detect it is by using software that allows you to view currently-in-use layer-4 IP ports or through a previously-put-in-place file integrity solution that will notify you if any critical files changed on your production systems without your approval. The second phase malware of these attacks often removes the initial worm (the delivery mechanism), making it even harder to detect.

Anti-virus software is, for the most part, useless in this situation because the attacker's code is polymorphic—it changes faster than anti-virus companies can track its signature. The highly efficient worms (just the delivery mechanisms) I noted earlier allow the miscreant hacker to make vulnerable systems come to them and then later, through a command and control network under the miscreants' control, send new malicious code to the infected systems after they initially send their beacon or when they "check in" afterwards at predetermined intervals. The real malware payload (the 2nd phase) code seems to include one or more of an IRC client, an IRC server, SMTP relays with highly efficient SMTP/MTA software, and more effective methods of hiding on the host systems and spreading to other adjacent networked systems. Of course, many also include efficient DoS (denial of service) engines that aid the attackers in proliferating DDoS attacks.

The miscreants behind these threats are often employed by various criminal organizations around the world that seek to attack and extort money from legitimate on-line sites, distribute unsolicited e-mail, and disrupt some forms of global commerce and Internet infrastructure. It is estimated that forty percent or more of all spam is being distributed this way, using organizations' network resources to send the unsolicited mail while providing the attacker a cash-cow business with no accountability since he's using someone else's resources and implicating them as the source. The DDoS capabilities are mainly aimed at extortion (threatening large Internet-dependent companies to pay them or they'll shut down the site ("packeting" it, to use their terminology)).

Back to the specific problem, when the infected system connects to the command and control network, besides downloading the latest new code, it often downloads the "next contact" time and location for the command and control operation—this allows the miscreants controlling everything to remain mobile. So, it downloads new operating code, new ways to hide, new ways to spread, new software to meddle with your networks, new time and means to communicate with the command and control network, and maybe worst of all, a list of targets to participate in DDoS attacks against without your knowledge.

Essentially, your infected system becomes one node in several thousand that form a "bot net" used to launch coordinated attacks against targets as a team. Other instructions have been verified to include sending millions of spam messages through their MTA software and some forms of espionage. The latest malware contains keystroke loggers that record usernames and passwords being used locally and with large Internet sites such as popular on-line banks. The malware uploads the usernames and passwords it collects to command and control periodically giving the miscreants access to your accounts. I can only reiterate that this is a significant threat to your networks and potentially your businesses.

Your infected system is viewed as merely another resource in a sea of resources. No infected system is being attacked specifically and no infected company is being singled out—this is criminal activity on a tremendous and impersonal scale.

You may be asking, "how do I protect myself or my company?" The simple answer is the answer any good security engineer might give: through layered security. For those organizations that are still under the impression that layered security means using a firewall, anti-virus software, and a virtual private network, ***wake up!*** Layered security means having more than one defense for any specific threat. By way of example, if you have one set of firewall filters keeping things out of your network, you should have another layer of firewall filters keeping data between your DMZ and your internal network secure and yet another layer filtering the egress of data leaving your network. Layered security means having a Plan B for every Plan A. In this specific case, if Plan A is to use your anti-virus and firewalls to keep the threats out, Plan B should (at least) be a procedure for managing and removing them once they're inside. In this case, tools that monitor egress network flows are the most useful in order to find software phoning home (to command and control). When you find a system making questionable network connections and that system passes anti-virus tests, migrate services off it, wipe it, re-provision it, and redeploy it. I can't stress enough that the entire adjacent network segment should be investigated as well.

In today's threat-rich network environment, the posture of "I won't let them in" is now simply flawed thinking and must be replaced as quickly as possible with the posture of "I'm doing everything I can to keep them out, but I know how to identify and handle them when they get inside." Oh, and you should add "and I am not relying on any one vendor to protect me, especially not the anti-virus vendors."

About the Author

Victor Oppleman is an accomplished author, speaker, and teacher in the field of network security and a specialized consultant to some of the world's most admired companies. Victor's open source software has been distributed to hundreds of thousands of computers worldwide and some is used in graduate-level college curricula to demonstrate advanced networking techniques. Early in his career as an engineer, Victor developed portions of the backbone systems infrastructure for Genuity, the first Internet data center company. Later, as a senior architect for BBN and GTE Internetworking, Victor developed security-related products and services centered on public key infrastructure (PKI). A great deal of Victor's professional career has been dedicated to tactical engineering and consulting for global telecom operators and critical infrastructure organizations in industries such as power and water, financial services, and defense. Some of the largest global companies frequently call upon Victor to perform advanced vulnerability assessments, provide expert counsel, and navigate complex regulatory issues concerning information security. An accomplished executive and engineer in network security, data hosting services, and software development, Victor also holds US intellectual property patents in distributed adaptive routing and wireless consumer applications.

Contacting Victor:
Website: http://oppleman.com | E-mail: public-articles@oppleman.com

Printed in the U.S.A.

The information in this publication is subject to change without notice and is provided "AS IS" WITHOUT WARRANTY OF ANY KIND. THE ENTIRE RISK ARISING OUT OF THE USE OF THIS INFORMATION REMAINS WITH RECIPIENT. IN NO EVENT SHALL ITS AUTHORS BE LIABLE FOR ANY DIRECT, CONSEQUENTIAL, INCIDENTAL, SPECIAL, PUNITIVE OR OTHER DAMAGES WHATSOEVER (INCLUDING WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION OR LOSS OF BUSINESS INFORMATION), EVEN IF ITS AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.