

Lessons Learned from Virus Infections

Jason Gordon 2004-10-04

0. Introduction

There are so many vulnerability scanners and penetration testing services or utilities available that many organizations use at least one of them gauge their security posture. Each tool has its own strengths and weaknesses and generally does a fair job at assessing an organization's network defense.

Viruses, including network worms, Trojans, and more can provide equally good, and often times better, views of the network in a true production environment -- and there is quite a bit an administrator can learn from a security compromise. This article does not intend for security administrators to intentionally infect machines; instead it is a guide to what an unintended infection can uncover about a network. With security companies such as Symantec reporting that 40% of Fortune 100 companies have been infected with viruses over a period of six months, it is well worth the exercise to see what can be learned from these infections. Specifically, after an infection is a time to evaluate the technical pieces of the defense perimeter (including firewalls, ACLs, etc.) and the non-technical pieces (continuity plans, emergency response, etc.).

Many "lessons learned" documents discussing worm outbreaks focus on a single issue: computers that need to be patched. Over the last few years, the focus has expanded to: computers that need to be patched very quickly. Certainly this is an extremely important part of network defense, but it doesn't address other important information that can be gleaned from an infection.

Many of the vulnerabilities a worm finds are, in fact, not things that can be patched (such as bad file share protections, poor user policies, and so on). This leads many administrators to repeatedly take only a single lesson away from infections, and often it is the conclusion that their users are entirely to blame. While everyone has a few stories about odd user behavior, they are not always at fault. Dismissing virus infections as a side effect of simply having network users also prevents many people from making the most of an outbreak.

Internet worms are not the only things that can be addressed with information taken from a virus infection. Malicious attackers will often exploit these same vulnerabilities used by the

worms on a manual basis, and these people specifically target an organization in the hopes of stealing critical data or causing a lot of havoc. Moreover, Internet worms that were once simply an irritant are now more likely to carry a backdoor, a Trojan, or open a session to the author's IRC server. Worm infections are clearly not desirable, even if one thinks there is a great deal to learn from the outbreak, but they are an unfortunate reality in large network environments. The ideas below will help bolster your defenses for the next round.

1. Viruses Push the Limits

Even the most exuberant vulnerability auditor or penetration tester will use safe, reserved methods when testing production network hosts for holes. This makes sense; dropping a room full of servers just to prove a DoS attack is possible may not make the best impression on one's manager (ie., what's the security ROI in that?). Even penetration testing that intends to simulate a full-scale attack may not be launched because of concerns over the impact to production and its associated costs.

An unwanted virus infection can provide real insight to the security of a network in ways that human-driven tests cannot. It will attempt things that a careful penetration tester would not. It is free from worrying about such things as whether all of your file servers drop offline, whether you really needed those documents on your hard disk, or if the traffic it generates makes everyone's web surfing slow. Second, a network worm is coded for one thing: exploiting as many hosts as it can reach -- a worm's life depends on propagating quickly. It will test for vulnerabilities in your network like no tool can.

Unlike a vulnerability-testing package, however, a worm will have a very specific focus and normally a set of vulnerabilities that it exploits, giving you a narrow (but deep) look at only one or two facets of your network. Once the response effort is complete and the clean up is under control, it is time to take a hard look at what that infection has uncovered.

2. Lesson Plan

Each type of threat, as listed below, provides a unique look at what was (and still is) vulnerable on a network.

2.1 Internet Worms

Worms like Sasser are designed to exploit a single hole. If a vulnerability scanner showed that

98% of the machines on the LAN were patched with hotfix 835732 (MS04-011), for example, yet 15% of the machines were infected with Sasser, something is clearly wrong. Either the scanner has reported a lot of false negatives (perhaps it had checked for another one of the 14 patches included in this fix, or else it just made a mistake), or someone in your organization has redeployed a number of unpatched machines since the scan. It deserves to be noted, as obvious as it may seem, that network worms use real exploits to test for holes, not "safe" scans -- and they don't provide too many false positives.

Worms also give a quick look at how well the segments of your LAN are guarded against various types of traffic. Using Sasser in a second example, there would be no question that internal hosts can be reached on TCP 445 if they are compromised by this worm. If that is a shock to a security administrator, then an examination of the firewall, ACL, or host filtering rules is in order. The open port may be the result of a specific application that has been deemed necessary, but was never evaluated by a security engineer.

Furthermore, worms often provide a network stress test as a secondary function of their attempts to propagate. Sasser's launching of 128 to 1024 threads in its attempts to spread are more than capable of providing this test, as you may have seen.

2.2 Mass Mailers

A mass mailer infection will give an administrator a good deal of information about the email infrastructure. First, the infection likely indicates a path for executable files that have made their way into the network. This may be the impetus needed to adjust corporate policy and block EXE, VBS, ZIP, or any other file extension at the external mail relay. Possibly, the mail came in via webmail clients, providing a justification for blocking popular public domains such as Hotmail, Yahoo, and Gmail at the firewalls.

Second, user training is an issue that should be examined after a mass mailer infection. The health of the user base and their virus knowledge is always important when dealing with these and other socially engineered types of attacks.

The extent of the infection that resulted is also a good indicator as to how well users are shielded from one another. Most modern mass mailers utilize their own SMTP engines, meaning that if mail is not accepted at the mail servers from internal network hosts (but rather must come from a trusted mail relay) there is little chance that the worm will leave the network it

started in.

2.3 Parasitic Viruses

A virus in the traditional sense infects files by adding its own code to that of trusted executables at run time. The existence of such a compromise would indicate poor software download/sharing practices on the part of users. Whether it came in via floppy disk, web download, USB device, or any other means, the infection points to the fact that host protection is not sufficient to block these kinds of attacks. For critical systems, a host based IDS (something akin to Tripwire) may be required. Again, it may be discovered that user education is unacceptably low. Finally, client antivirus software should be evaluated.

2.4 File Share Worms

Worms that spread via network file shares point to weak internal security. Whether it is a worm such as Lovgate that (in addition to its mass mailing attributes) spreads to every visible network share, or it something like Nebiwo (Deborm) that spreads via administrative shares, an infection of this type indicates there is at least one unprotected share/host on the LAN. Almost always this means there are some shares without passwords on the network, or at best they are easily guessed passwords that need to be addressed. If there are any boxes that don't have local administrator passwords, these worms will surely find them for you.

2.5 Trojans

Much like viruses that infect legitimate executables, Trojan infections indicate a breakdown in client security. There is often a problem either with a user's browsing habits, the types of code users are allowed to download (maybe ActiveX controls should be blocked at the gateway), or the media an end user is bringing into the network.

Unique to backdoors is a means to communicate with an external controller. When Agobot infects a network asset and is then used to scan/infect other hosts, there is an indication that there is a path out of the LAN allowing for commands to be passed back and forth. At that point, port and packet filtering at each level should be analyzed. There are free firewalls/filtering tools with many versions of Windows/Linux, there should be tight controls over what ports are open to external addresses on the organization's firewall, and even the most modest of IDS products should detect external connections passing commands to client machines.

3. The Lessons Learned

In each case mentioned above, there is at least one technical and one non-technical problem that needs to be examined. Each type of problem requires a corresponding solution. Trying to address non-technical issues with technical tools is often a frustrating game of the proverbial "square peg in a round hole" for administrators of all kinds. Security professionals know all too well that there are few technical protections that a determined user can't undo if he hasn't been educated. Similarly, a determined user or attacker will have little problem evading poorly configured or under engineered solutions.

Revisiting our first example, a Sasser outbreak, shows how an infection can point to non-technical problems. In this case, if administrators are rebuilding clients and are not aware of required patches, the results of a vulnerability scan can be invalidated quickly. This is a problem of information flow and configuration management -- and in larger organizations, can often be resolved with policy changes.

The lessons from such infections often do a lot to organize the organization's tactics for layered defense as well. Whenever a virus causes a disruption of service, the likely reaction by management is to ask what happened, and why. An engineer can summarize the vulnerabilities, point to each location, and make recommendations as to which part of the network should be changed. In most cases, the engineer presenting such recommendations will look at the costs involved in each change, the effectiveness of each change, and the future administration necessary to make the adjustments successful over the long term. This is surprisingly close to the process of providing ROI data to managers.

In more cases than not, traditional thinking dictates that changes to a central choke point are often more effective and cheaper than touching every workstation, recalling mobile devices, and so on. In other words, making a change to the firewall rules is a better choice than installing a new filter on every single desktop, provided each solution has comparable levels of success. The actual step here is not important, it's more the fact that a virus infection may challenge the notions that engineers and managers alike had about where the network was strongest and weakest. For instance, a Lovgate outbreak within the protected LAN may expose the user/laptop policy as being weak, as the firewall and mail relays would have properly prevented infection via email or network shares. Depending on the costs involved in cleaning up the infection(s), the compromises required may serve as the needed catalyst to spend the money on education and better client-side security tools.

If MyDoom had spread across your network, it is likely that the mail relays were not dropping attachments of EXE, COM, BAT, CMD, PIF, SCR, or ZIP files. If there is a business case for distributing these files, then another layer of the defenses will need to be reinforced, such as user training. If there is no training possible (because of money concerns, time constraints, or the size of the organization comes into play), then the gateway/client side AV software will need to be tight -- as it is all that's left to combat this threat.

4. Detection and Alert Mechanisms

If network worms completely blindside the network several times a year, there is likely a need for better detection tools. In very large organizations with thousands of clients it may be difficult to keep all client AV software updated and running properly, particularly with a large mobile workforce that have personal firewalls on each machine. It is always wise to have another line of viral defense in front of the clients, and larger organizations tend to employ a second AV vendor's tool at the gateway and/or an IDS with worm recognition features.

Many security professionals have debated the use of an IDS to detect viral activity. One's personal beliefs in this matter notwithstanding, an existing (or inexpensively built) IDS can always improve worm detection and mitigation efforts. Although it is not the core competency of such a device, many IDS platforms allow for quick and customizable virus signature additions. Furthermore, by its very nature, the IDS is in a good position to identify worms as it needs to inspect every packet traversing the network. Also, an IDS can see a worm propagating from clients that don't have their AV client running properly (or running at all), something that even the best AV management console can't provide.

Virus signatures are written and published constantly on sites like [Bleeding Snort](#), and although they are presented with a number of warnings about false positives, they should be more than enough of a foundation to build a generic worm detection machine. One thing that multiple worm infections have likely taught every administrator is that a worm has to do a lot of reconnaissance to spread quickly. Blaster and Sasser were certainly not trying to emphasize stealth with routines that open up to 1024 threads to scan for new hosts (example: Sasser.C). Basic anomaly detection would have triggered alerts for such activity. Mass mailers, of course, have to send a lot of messages out to compromise additional hosts. That means detecting TCP 25 activity for non-SMTP servers/relays can tip an administrator off to an attack before it gets too far out of hand. File share worms (such as the vector included with the Lovgate variants) are likely to require more specialized signatures, something that actually uses a content field

composed of the actual worm binary. However, IDS detection is certainly capable of pointing out a lot of failed logins to SMB resources, which is an anomaly that often indicates a worm is trying a weak set of logins/passwords against a host in an effort to access the machine and propagate.

Alerting users and administrators to brand new viruses and infection mechanisms is a different story. Fortunate security officers may work in organizations that provide a few hours of safe computer training every year, however gathering everyone for a conference each week to talk about viruses is not realistic. After an infection, take a look at how users have learned about the mitigation and cleanup activities. Before an infection, evaluate how they receive updates to their security training, whether it's via email, a lunchroom poster, a personal visit, or some other method.

5. Establishing a Defense Plan

Regardless of the technical course of action, a virus event can help open lines of communication with company officials regarding their security policy and budgets. Like no other event, virus outbreaks, and the subsequent virus hysteria within an organization are capable of granting the security administrator an immediate audience with upper management. This is likely the most important part of learning from an outbreak: presenting your findings to the executive staff, gauging their reaction, and making a case for additional security funds. Share with them what has been learned. If the organization's management is generally unreceptive to hearing about requests for additional money and information assurance, take advantage of this heightened opportunity before the window closes. The discussion does not need to be a technical one; many business continuity officers and risk managers will be exceptionally receptive to prevention measures. Again, the direction of this meeting is dictated by what was found in the discovery. A social or technical problem often needs the same type of financial solution.

If the latest worm has ravaged the organization it is certainly time to take a hard look at correcting the deficiencies in the security plan, whether they are social or technical. It should not be hard to estimate some costs of the infection, particularly downtime; that data will help a lot when it's time to talk about funding. Furthermore, one can diagram vital systems and point out where the additional defenses are needed. This not only helps demystify the role of firewalls, IDS devices, virus scanners, and more, but also will help the security team present a clear technical request to the management team.

If the network defenses are already in good shape (or the organization has perhaps just been lucky), it is still a good opportunity to map out what went well. This is not just a time to boast how great a job the security department is doing, but also to mention what company initiatives and funding have allowed the network to remain safe from the latest threats. Recommend that these programs be extended to cover more of the enterprise and further reduce risks. Take some of the more reasonable "infection/cleanup cost" numbers to help provide some idea of what is being saved by avoiding virus infections.

6. Conclusion

As should be evident from the examples in this article, a virus outbreak will produce a few unique opportunities to examine the health of the network defense. It can also be a great opportunity to justify to senior management what additional financial resources may be needed to contain future outbreaks. Your daily, non-emergency auditing and mitigation efforts can be greatly improved by taking a few additional moments after an infection to detail exactly how the emergency plan really did work, and not just how well it should work.

Comments or reprint requests can be sent to the [editor](#).

[Privacy Statement](#)

Copyright 2006, SecurityFocus