ADVISING THE GOVERNMENT ON BOTS

In response to the USA Homeland Security, NIST and NTIA RFI

(latest advice at: http://www.ciphersbyritter.com/COMPSEC/ADVISING.HTM)

A *Ciphers By Ritter* Page

Terry Ritter

ritter@ciphersbyritter.com

2011 October 4

Introduction

This is a response to the Notice by the US Homeland Security Department, the National Institute of Standards and Technology, and the National Telecommunications and Information Administration on 09/21/2011, entitled: "Models To Advance Voluntary Corporate Notification to Consumers Regarding the Illicit Use of Computer Equipment by Botnets and Related Malware."

Clearly, the issue here is bot malware, although some questions seem oriented toward using ISP's (Internet Service Providers) for detection and remedial action. Unfortunately, that particular view of the bot problem fails to address vulnerability, instead limiting itself to destroying bots after they already may have exposed everything they can reach. Accepting infection is a problem, because, generally speaking, the effects of bot malware cannot be reversed. Fortunately, the "Request for Information" also states: "The Departments invite comment on the full range of issues that may be presented by this Request for Information."

The underlying issue of bot malware is that the equipment we use has infection vulnerability designed in. The main problem is the ability of malware to change boot files and boot data on writable boot devices, like hard drives and flash drives. Relatively simple hardware and software changes could design infection out by making boot devices only partly writable.

Existing computers would require some new hardware (and software), but then would actually address the real problem. In contrast, ISP's cannot possibly detect every bot instance, and every infection they do detect would still require technical response from nontechnical users, which is not much of an improvement over requiring new hardware.

Background

Malicious software (or malware), which acts as an in-computer agent (or bot) for a remote bad actor, is not new. What is new is a dawning recognition of both the widespread societal danger of the problem, and our general inability to control it. At the risk of seeming “over the top,” malware is nothing less than a threat to our economic national security: Our ever-increasing commercial dependence upon the Internet is at risk.

Malware supposedly has been addressed for decades, but the conventional wisdom of scanning and patching demonstrably has failed, since malware is now more common and also more dangerous than it was before. It is easy to blame the user for this, but unless we get rid of users, we have to deal with what they are, not what we think they should be. Anti-malware scanners are ineffective until someone has found the malware and added it to the blacklist, and even that lasts only until a minor change is made in the malware design, often within a day. Patching is ineffective against unknown attacks, failure to apply security patches within days or hours leaves the user exposed to advertised known faults, and there seems to be no end to the patching process. While scanning and patching can improve the odds against malware infection, they cannot be depended upon as malware protection.

Microsoft Has Some Explaining to Do

At this point, we need to talk about what those who are harmed by bots generally have in common, and that is the Microsoft Windows operating system (OS). That could be a natural consequence of the market domination of Windows systems. But malware surprisingly targets Windows *far beyond* even the large proportion of Windows on the Net.

All operating systems are large, complex, logic systems, and they all have errors and exploitable flaws, so they all can have malware. (Generally speaking, there can be no large, complex, secure systems, and we cannot even know when we have the exception to the rule.) The dominance of Windows malware is not so much about Windows weaknesses as it is about attacker motivation: When they have a choice, attackers prefer to develop for a single target, and Windows is by far the largest target. The malware results are what we should expect from market domination, assuming that malware is essentialy uncontrolled.

What we should *not* expect or excuse is that Microsoft has not controlled a problem which predominantly affects their own products but threatens an infrastructure of society. Instead, Microsoft has continually “doubled down” on methods which cannot be expected to work, and now have been explicitly *demonstrated* to not work. Anti-virus scanning and patching have not solved and will not solve the malware problem. For now, the most effective way to avoid almost all malware is to “step away from the target,” and simply use something other than Microsoft Windows on the Internet.

In the future, things will change. Even now, some attackers do not want to deal with Microsoft Windows code and so choose another target. Desktop applications already tend to be more vulnerable than the OS. And the rise of smartphones is changing the malware target equation. But it would seem that Microsoft is going to dominate the desktop for some time to come.

As opposed to trying to corral every ISP to detect bot messaging, which probably cannot be particularly effective anyway, a better alternative would be to require OS manufacturers to produce a free program or DVD which could certify that any particular installation of their system had not been modified after installation, and, thus, did not have a hidden bot. It seems ludicrous to blame nontechnical users for hosting a bot infection which even the system manufacturer cannot quickly find. A bot detection program probably would be unnecessary on DVD-boot systems which “install” from DVD on every boot.

Another alternative would be for Microsoft to issue a DVD-boot form of Windows, perhaps mainly for online use. Even DVD-boot systems, however, need to support security updates. As an example, Puppy Linux includes an apparently unique ability to update programs by writing a new “session” on a multi-session DVD. At boot time, only the latest files are loaded, and then the DVD can be removed. While Puppy Linux security can and should be improved, the ability to update browser, add-ons, and other files is a substantial security advantage.

What is Bot Infection?

A “bot” (short for “robot”) is a program in the user’s computer which can execute commands from a remote bad actor. Normally, a bot can do everything and more than a typical user can do, because the bot is being commanded by a computer expert. A bot can read every file, change any file, including dates and terms in legal and commercial records and contracts, write new files, record keyboard actions, copy the screen display, and send all this to the controller over broadband. Much of this can be done in ways which probably would not trigger ISP attention. Sometimes bots are used to send large amounts of email spam, which probably can be detected, but that is by no means a universal indication of “bot-ness.”

Our main bot problem, surprisingly, is not the bot itself. When some bot code gets through the “front end” protections of firewall, anti-virus and browser, if the bot code is designed to run under Microsoft Windows, but arrives on a Linux system, it probably will not run. But if that same bot arrives on a Windows system, it may well run. It may call home, attempt to infect the boot process, and generally be a threat until that session ends. But the larger bot problem is that bots *keep coming back* on each new session, which is bot “infection.” When a single bot infection can create hundreds of later sessions of bot-running, the infection is the fundamental problem, and infection largely can be eliminated by appropriate system design.

The problem for design is to prevent a running bot from infecting the next session. To infect a system, a running bot must change some code which will be executed in the next run-up. We can prevent infection by preventing the bot from changing run-up code or data. That is not theory: We have that now when we boot from CD or DVD. With a DVD boot, if we want to get rid of any malware which might have come in, we just boot the DVD again. With a DVD boot, malware only lasts to the end of a session. More importantly, the user can assure a clean system for online banking simply by rebooting.

A boot DVD option is available in a variety of Linux-based “LiveDVD” systems, but there are wide variations. The US Air Force Lightweight Portable Security is a reasonable example, but has issues, such as not allowing user security patching. The inability to quickly patch is a security problem because attackers exploit the issues the patches reveal, often within hours. A boot DVD scheme which *does* allow patching is used in Puppy Linux, but no volunteer hobbyist implementation is going to compare either to a funded security project or the commercial result of many millions of dollars of work.

We do have usable DVD-boot systems now; what we do not have is the ability to boot Microsoft Windows from DVD. If we did have a usable bootable Windows DVD which would come up without using data saved in a previous session, and if people would use it, much of the concern about bot infection would be over, and ISP involvement would just be icing on the cake.

THE QUESTIONS

A. General Questions on Practices To Help Prevent and Mitigate Botnet Infections

(1) What existing practices are most effective in helping to identify and mitigate botnet infections? Where have these practices been effective? Please provide specific details as to why or why not.

Currently, no tool or set of tools is known to exist which guarantees to detect any arbitrary bot infection. The general inability to guarantee to detect a bot is a fundamental problem with all bot-control activities and measures of effectiveness.

Anti-malware scanners are particularly limited, due to the delay between when a malware is released and when the corresponding signature is installed in the user computer. Many bot malwares subvert the operating system with “rootkit” technology, which can prevent the OS from even seeing malware files.

Other bots ("polymorphic" malwares) "encrypt" the bot code uniquely for each computer, making the general signatures useless.

It actually may be *impossible* to build a general tool which can decide whether any given program contains malware code. However, it *should* be possible for operating system manufacturer to produce a tool which detects unauthorized additions to their system, and so expose malware infection. Substantial OS changes may be required, but the cost to society of not being able to identify an infected computer is what we are now starting to see.

(2) What preventative measures are most effective in stopping botnet infections before they happen? Where have these practices been effective? Please provide specific details as to why or why not.

The most effective way to stop bot infections is to boot and use a Linux LiveDVD when on the Internet, particularly on systems which have no writable drive (known as "thin client" systems). Unfortunately, many people find the DVD boot idea offensive, in that it would intrude on their ownership: It would require them to do things differently than they are accustomed to do. And a boot DVD is currently not an option for those who need to use particular applications which work only under Microsoft Windows.

(3) Are there benefits to developing and standardizing these practices for companies and consumers through some kind of code of conduct or otherwise? If so, why and how? If not, why not?

It might seem that all we have to do is to educate those sad users so they can keep their machines clean. Unfortunately, there are many ways to get malware, and the approaches become increasingly devious. It is not always possible to distinguish the correct choice, even for an expert. All it takes is one human error to get the hard drive infected, and ordinary users will not even imagine they have a problem. "Experts" will claim they could not possibly have been infected, despite having no way to prove any such thing. The problem is not dumb users, but instead the equipment which supports bot infection.

Rules or a "code of conduct" can at best make a minor difference in the probability of encountering malware. A minor reduction in probability is not sufficient. User-enforced rules cannot stop malware.

(4) Please identify existing practices that could be implemented more broadly to help prevent and mitigate botnet infections.

Hardware and software manufacturers could produce (relatively) simple upgrades resulting in systems which are inherently difficult to infect. To protect hard drive and USB flash drive boot storage, for example, a new definition of the drive concept could introduce hardware-protected storage for OS and boot files, updateable only with owner approval. Long experience demonstrates that OS software alone cannot provide that protection. The currently-planned massively-complex systems which are claimed will be secure (someday) seem unlikely to fulfill their claims.

(5) What existing mechanisms could be effective in sharing information about botnets that would help prevent, detect, and mitigate botnet infections?

1. Bots infections can be *prevented* by not allowing malware to change boot run-up files and data. The operating system has long attempted to protect OS files with permissions and privilege-restricted user modes, and has failed, because running malware subverts, controls and "owns" the OS. However, simply using a boot DVD can make it "difficult or impossible" for malware to change boot and other data on that DVD, even on a writable DVD, and so prevent bot infection.
2. Bots could be *detected* if the operating system manufacturer released a tool to detect any changes to an installation of their particular product. A general tool to reliably detect bot code in arbitrary

programs seems unlikely.

3. Bot infection consequences might be *mitigated* by not allowing broadband communication, which may be more than we are willing to do. Using a DVD operating system boot disc to prevent the infection seems more acceptable.

(6) What new and existing data can ISPs and other network defense players share to improve botnet mitigation and situational awareness? What are the roadblocks to sharing this data?

The problem is not a lack of data describing botnets, the problem is that our equipment supports bot infection. After infection, data about the infection are nearly meaningless, since whatever secrecy or privacy we had has gone, and the best-practices response is to re-install the operating system and apps so we can support new secrets. Users who copy programs from their infected drive may be re-installing their bots.

(7) Upon discovering that a consumer's computer or device is likely infected by a botnet, should an ISP or other private entity be encouraged to contact the consumer to offer online support services for the prevention and mitigation of botnets? If so, how could support services be made available? If not, why not?

If an ISP detects a possible bot, measures should be taken. However, it is quite possible for a bot to hide from external ISP detection, and it is even more possible for an ISP to develop a “false positive,” and so “detect” something which is not there. Trying to coerce the user to remove something which does not exist would be not only futile but extremely inequitable, and probably actionable under law. All of this can be avoided by requiring the OS manufacturer to deliver a tool which guarantees to detect a bot infection in hard drive or flash drive installations.

If a bot infection is found, there really is no alternative to a complete re-install of the operating system and applications. Unlike old-style virus attacks, a bot program is not a few self-contained actions which can be identified and reversed. Instead, a bot communicates with a controller (or “bot handler”) and follows his orders, so there is no way to know what has been done, and no way to reverse it. The OS must be completely re-installed, along with applications (apps) and user data.

(8) What should customer support in this context look like (e.g., web information, web chat, telephone support, remote access assistance, sending a technician, etc.) and why?

Customer support for malware starts with the OS manufacturer. The manufacturer should be required to provide an easy, complete, bot-cleaning OS re-install which a nontechnical computer owner can apply. Users must be given the right to have, and be provided with, easy re-install media for all their applications. One advantage of a boot DVD is that it essentially re-installs the OS and apps on every boot.

(9) Describe scalable measures parties have taken against botnets. Which scalable measures have the most impact in combating botnets? What evidence is available or necessary to measure the impact against botnets? What are the challenges of undertaking such measures?

Both patching and scanning have been used for many years. Because malware continues to be an increasing problem, we have direct evidence that neither approach will solve the problem.

No tool, nor set of tools, exists which guarantees to detect any arbitrary bot infection. Without such a tool, there is no way to collect accurate statistics about the proportion of bots being detected. It is easy to claim to have no bot when no tool can detect the bot which hides really well. There is no way around needing a malware detection tool to gain an accurate understanding of the extent of the problem.

The OS manufacturer should be able to provide a malware detection tool, although rising to that challenge may require fundamentally different OS implementation approaches. The problem is that current OS designs love to change files as they run, but disallowing changes is how we prevent infection. OS's intended to be "secure" should be "static," or basically unchanging. We can force that now with OS boot DVD's.

B. Effective Practices for Identifying Botnets

(10) When identifying botnets, how can those engaged in voluntary efforts use methods, processes and tools that maintain the privacy of consumers' personally identifiable information?

The question would seem to indicate some lack of understanding of what having a bot means: When a bot is in place, it can read any file in the system including all past email, and send it off to the bad actor via broadband. A bot can read SSL-encrypted secure banking messages after decryption in real time, capture any typed password or digital authentication token and so gain full access to online accounts and corresponding data. In general, a bot can defeat all cryptographic authentication, including certificates and external 1-time 2-factor dongles, because the problem is not the cryptography, the problem is the insecure machine.

As data move into "the cloud," the security of the individual machine hardware becomes increasingly important. Even a single infected machine may constitute a major exposure risk for cloud storage.

Yes, information workers can be a vector for exposure, but when a bot exists, everything on that machine, and everything it has accessed, should already be considered insecure. Nothing can make that worse. Effort spent to avoid friendly exposure is wasted.

The larger problem is that there is no tool which guarantees to detect a bot, so the exposure may continue essentially forever. Bots which send home only small amounts of the most tasty information are going to be very difficult to detect externally.

(11) How can organizations best avoid "false positives" in the detection of botnets (i.e., detection of behavior that seems to be a botnet or malware-related, but is not)?

Absent an appropriate detection tool from the OS manufacturer, false positives *cannot be avoided*, and will cause much friction and "blowback."

(12) To date, many efforts have focused on the role of ISPs in detecting and notifying consumers about botnets. It has been suggested that other entities beyond ISPs (such as operating system vendors, search engines, security software vendors, etc.) can participate in anti-botnet related efforts. Should voluntary efforts focus only on ISPs? If not, why not? If so, why and who else should participate in this role?

It is long past time for the hardware and software manufacturers to take responsibility for fundamental security flaws in their designs. Without those flaws, bot malware would be more of a curiosity than a problem. These same issues recur with many different networked devices. But this is not an ISP problem, so ISP's cannot fix it.

C. Reviewing Effectiveness of Consumer Notification

(13) What baselines are available to understand the spread and negative impact of botnets and related malware? How can it be determined if practices to curb botnet infections are making a difference?

Government should be able to require banks to confidentially disclose malware losses, which is a malware impact.

To judge the effectiveness of anti-bot measures, it is first necessary to be able to detect bots so they can be counted. Currently, no tool, nor set of tools, exists which guarantees to do that. It may be somewhat possible to relate anti-bot measures to reduced losses from online banking.

(14) What means of notification would be most effective from an end-user perspective?

Online communications should remain as undisturbed as possible, because many bot "detections" will be false positives, and it is impossible for the user to correct a problem which does not exist. A reasonable notification could be by email, although some way of forcing the user's browser to open a new tab to an ISP notice and collected data evidence page for that user might be appropriate. A single simple notice for everyone is not sufficient; the notice must include the data used to judge that the user has a problem, since only that can indicate the likelihood of a false positive.

(15) Should notices, and/or the process by which they are delivered, be standardized? If so, by whom? Will this assist in ensuring end-user trust of the notification? Will it prevent fraudulent notifications?

Standardizing notices is not important, and having a single notice for everyone is wrong. The real problem is that the most dangerous bots will be difficult or impossible to detect externally, and they will also be best at avoiding internal detection. So all this anxiety about the mechanics of ISP response is about something unlikely to control the worst part of the problem.

(16) For those companies that currently offer mitigation services, how do different pricing strategies affect consumer response? Are free services generally effective in both cleaning computers and preventing re-infection? Are fee-based services more attractive to certain customer segments?

Unfortunately, it is generally IMPOSSIBLE to "clean" bot malware from a computer, in the sense of a guaranteed full recovery of the pre-infection state. Producing a "clean" or malware-free environment requires a complete OS re-install or the recovery of an untainted OS "image." However, OS installation can and should be made much easier and more effective.

(17) What impact would a consumer resource center, such as one of those described above, have on value-added security services? Could offers for value-added services be included in a notification? If not, why not? If so, why and how? Also, how can fraudulent offers be prevented in this context?

This question would seem to rely on the concept that bots can be "removed," a dubious assumption. Yes, individual bot codes can be erased, but finding what the bot did when it was active and then reversing all of those changes is just not possible. Moreover, the usual multiplicity of bots implies a sequence of "removals" which ends when no more bots can be detected, but there is no tool to tell us that unremoved bots still exist. "Best" practices require a complete OS and apps re-install. Even "basic" practices require a complete OS re-install. While re-install may be painful, it is not optional.

(18) Once a botnet infection has been identified and the end-user does not respond to notification or follow up on mitigating measures, what other steps should the private sector consider? What type of consent should the provider obtain from the end-user? Who should be responsible for considering and determining further steps?

Until the user has a tool which guarantees to detect a bot if it exists, coercion is inappropriate and quite disturbing. Users simply cannot be held responsible for not eliminating something they cannot detect. However, if a bot detection tool were to exist, and given to the user, then the user reasonably could be responsible for correcting the situation.

(19) Are private entities declining to act to prevent or mitigate botnets because of concerns that, for example, they may be liable to customers who are not notified? If so, how can those concerns be addressed?

Bot detection will be problematic in any case. Surely this situation will have to be announced to all users.

Best Practices for Consumer Notification

(20) Countries such as Japan, Germany, and Australia have developed various best practices, codes of conduct, and mitigation techniques to help consumers. Have these efforts been effective? What lessons can be learned from these and related efforts?

The whole point of such “codes of conduct” is to reduce malware infection rates. Having been in place for some time, have they done that? Not to my knowledge. Indeed, developing such knowledge probably requires bot-detection technology which does not exist. There currently can be no way to know whether the approach is “effective.” We cannot depend upon user action to solve the malware problem, other than to re-install their OS and apps when necessary.

(21) Are there best practices in place, or proposed practices, to measure the effectiveness of notice and educational messages to consumers on botnet infection and remediation?

Is it “ineffective” for a user to not “remediate” a bot which does not exist? Yes, some bots can be detected and removed, but those are unlikely to be the dangerous ones we care about. It is not possible to measure the effectiveness of messages to instigate remediation when we cannot know whether or not the bot actually exists. There is no tool which guarantees to tell us when an arbitrary bot exists.

D. Incentives To Promote Voluntary Action To Notify Consumers

(22) Should companies have liability protections for notifying consumers that their devices have been infected by botnets? If so, why and what protections would be most effective in incentivizing notification? If not, why not? Are there other liability issues that should be examined?

Any communications utility or service should be required to produce a best-effort toward full communications at all times. Companies which take it upon themselves to cut off customers should be held liable for consequential business damage, medical expenses and death compensation in the case of loss of 911 emergency service.

(23) What is the state-of-practice with respect to helping end-users clean up their devices after a botnet infection? Are the approaches effective, or do end-users quickly get re-infected?

We are long past the era when malware could simply be “removed.” First of all, malware can only be “removed” if it can be found, and the "best" modern malware hides really, really well. Next, it is common for modern malware to download all its friends, so each must be “removed,” perhaps leaving some still hiding. Absent a tool to detect the hiding malware, the system has not been “cleaned.”

While a general tool for checking all files for “malware-ness” probably is impossible, a tool which simply checks each necessary OS file for existence and correctness is possible and also reasonable. However, the

OS itself would have to stop depending upon dynamic files holding "state" from previous sessions. The tool probably would have to run from a clean OS, and so probably would be a "LiveDVD" itself.

Moreover, when a bot is present, it may have added or modified many different files, for various reasons. Someday we will find that malware has changed or re-dated various official records, contracts, deeds or statements; recovering from that will require backups of the old versions. No reasonable form of “removal” can possibly correct such damage, although recovery from a saved drive image might, if users in fact make such an image.

(24) What agreements with end-users may need modification to support a voluntary code of conduct?

Since no code of conduct can possibly solve the malware problem, except perhaps an agreement to not use Microsoft Windows online or for online banking, support or non-support would seem irrelevant.

(25) Of the consumer resource scenarios described above, which would be most effective at providing incentives for entities to participate? Are there other reasons to consider one of these approaches over the others?

Most effective would be a Microsoft Windows LiveDVD which also supports code updates by way of DVD multisessions. The most effective way to get Microsoft to participate would be FCC type acceptance to require such a product to make Windows less of an “attractive nuisance.‟

Less effective would be a financed project to produce a more acceptable version of Linux that more people would use. It is possible that many Windows apps could be supported by further developing the Linux Wine system, but that would of course also increase the chances that Windows malware would run even under Linux.

(26) If a private sector approach were taken, would a new entity be necessary to run this project? Who should take leadership roles? Are the positive incentives involved (cost savings, revenue opportunity, etc.) great enough to persuade organizations to opt into this model?

Since Microsoft owns Windows, only they could build and distribute a Windows boot DVD.

Instead of trying to micro-manage development, a better government approach would be to issue a request for DVD-boot secure-OS project proposals, and then fund two or three of those, aiming to get multiple products to the marketplace so they can compete.

(27) If a public/private partnership approach were taken, what would be an appropriate governance model? What stakeholders should be active participants in such a voluntary program? What government agencies should participate? How could government agencies best contribute resources in such a partnership?

A public/private development partnership seems likely to take much longer than we have, and so seems quite unwise. It seems more appropriate to request DVD boot project proposals and fund a few.

(28) If a government-run approach were taken, what government agencies should play leading roles?

Government should not be actually doing projects. Instead, government should be requiring that equipment meet minimum security levels as needed by society. One approach might be for the FCC to type-accept computer equipment to be “difficult or impossible to infect.” This would of course include computers, routers, cell phones, printers, etc.

(29) Are there other approaches aside from the three scenarios suggested above that could be used to create a consumer resource and to incentivize detection, notification, and mitigation of botnets?

The problem is not incentives, the problem is basic technology: Our equipment is vulnerable to bot infections, and we cannot trust our ability to detect them. The solution is forced to be technical, not legal.

(30) Are there other positive incentives that do not involve creation of an organized consumer resource that could encourage voluntary market-based action in detection, notification, and mitigation of botnets?

Absent a special bot-detection tool built by the OS manufacturer, it is nearly impossible to guarantee to detect a bot. Some bots can be detected, of course. But bot programmers would rapidly gravitate to bot designs and actions which are difficult to detect. Simply imagining that things will stay as they are so there would be no need to detect all bots is just “whistling in the dark.” The most serious threats will be the hardest to detect.

About Terry Ritter

I am a retired former Professional Engineer in electronics and computers. I built my first working computer from parts in 1974, and at first programmed it by flipping switches. My early interests were in computer architecture, where I helped specify, design and verify the Motorola MC6809 in 1979. Then I worked to find a deep understanding about why software development is difficult and error-prone, and used that insight to design and implement an early multi-processor industrial LAN. Eventually I became interested in Cryptography, where I innovated and patented several low-level cryptographic mechanisms. More recently, I asked myself: “How can it be that the efforts of a great many very smart people have not managed to make PC's secure?” Analysis produced a range of answers, but the particular problem of malware infection is exactly the deep understanding of lowest-level computation that I have long addressed.

1. For more about me, see my Author page.
2. For Computer Security analysis, see my articles: Basic PC Security and Simplified PC Security.
3. For online banking security analysis, see my article: The Banking Malware Mess, and a range of comments to various articles on the Brian Krebs site, KrebsonSecurity.