GSEC v1.3

About Heuristics

Stephen M. Sladaritz Sr.

23 March 2002

Abstract

So what exactly is heuristics? Is it the perfect paranoiacs tool, proving the world really is out to get them? Or is it a digital hypochondriac seeing viruses everywhere? This paper will discuss what heuristics is, why we should use it, warts and all, and some ideas for how to use it best. Finally we'll talk about how to be a good neighbor while using it, and wrap it up with a discussion on including heuristics in our antivirus policies.

Introduction

There are a couple of ways we can look at heuristics, it will either be our savior in a world riddled with computer viruses, or it will be a novelty item that occasionally makes our lives a bit more interesting for no apparent reason. But no matter how you look at it, you have to admit… the internet is not getting any safer, and even your e-mail is a gateway to destruction of your home or office PC.

We could just count on the old standards to protect us in the big bad world-wide-web, but the old ways aren't always the best either. So different ways of protecting our systems and data really need to be explored. Scanning a for list of known viruses would be great if no-one was writing new virus code, but that's a little bit unrealistic and way too optimistic. The flaws in this method are obvious, what is not known, you are not protected from, you can only be protected from a virus that has already chewed someone else's data up. Or maybe even chewed up your data a bit.

Then there are the behavior blockers, which are all well and good, as long as the nasty package doesn't do too much damage before the blocker realizes it's doing bad things. I kind of liken those to letting the dog in the house unless he starts chewing on sofa cushions… once he's started chewing, your sofa is still ruined, even if you kick him out after the first cushion.

OK, so each one has it's flaws, but each also has strengths. Which one is best? Well, for my money, a product should be a blend of the best of the available techniques, and it had better have some way of sniffing out the new bug without letting it eat my hard drive first. So where does that leave us… in the mysterious world of the heuristic scanner, and while it's not perfect either, it's better than a sharp stick in the eye.

What is Heuristics

> Heuristic *adj.* 3. <u>*Computer Science.*</u> Relating to or using a problem-solving technique in which the most appropriate solution of several found by alternative methods is selected at successive stages of a program for use in the next step of the program. [From Greek `heuriskein`, *to find*.] **1**

So, heuristics is the discoverer. It uses different methodologies, technologies, tricks, rules, or techniques to make educated guesses as to whether or not a file is infected by a virus. The problem is that any time you venture a guess; you probably stand a really good chance that you will be wrong, even a well-educated guess is just a guess. Both Network Associates' McAfee VirusScan and Symantec's' Norton Antivirus *missed* the Melissa virus completely with their heuristics scanners. But the question at hand is how does it work, not just how well.

Imagine that your antivirus program is a police station. It is manned with a few good cops who are good at their jobs. We talked about the typical scanner, this one uses a book of mug shots, every known criminal is listed here, and he checks everyone who passes his desk against the book. Known criminals are immediately arrested, but those who haven't committed a crime yet can stroll right on by without a second glance. So the advantage that scanner has is that it is very good at what it does, identifying known viruses. The main disadvantages are that the virus list must be continually updated, and the scanner will pass an unknown virus without a challenge.

Then there's the behavior blocker, the cop on the beat. This cop keeps a close eye on everyone who is active to make sure they aren't breaking any laws. If they start to do something they are not allowed to, exhibiting some apparently criminal (read virus-like) behavior (erasing executables, writing itself to files, etc.) then this cop will arrest them immediately. The advantage here is that it is not tied to a list, like the scanner, so it can potentially catch the unknown virus on the system. The main disadvantages are that perfectly legal programs can exhibit some of the characteristics that make it appear to be a virus to the behavior blocker, so false alarms are possible. Another disadvantage is that the behavior must be seen; the problem here is that sometimes irreversible damage may already be done before the blocker can put a stop to it. While they are both good at what they do, they leave holes in the security that needs to be filled.

That leads us to heuristics, the detective. A real sly one this detective is, and if he were a real cop he would be in trouble constantly for his methods. The Detective doesn't look for known criminals, or watch for virus-like behavior (although he will in the right circumstances, as you'll see later). The Detective makes judgment calls, all day long, and he does it with all the best tools the department can provide him, in the hopes that he will be right more often than he is wrong.

Your heuristics scanner is armed with extensive knowledge about viruses. Where is a virus likely to hide in a file (near the beginning or end of the file is most common), what virus code looks like, how viruses use encryption to hide their payload, and other nifty tidbits. Now given all the knowledge that the scanner is given, it may work in one of two ways, static or dynamic.

Static heuristics is pretty dull really; the detective is watching everyone that goes by using his knowledge of viruses to profile everything that it sees. If a file meets a predetermined number of the qualifications for a virus, the scanner will put the drop on it and lock it down. So the detective notices encrypted data at the beginning and end of a file, this one may also seem a little to interested in the files it sees around it, or attempt to remove from or write to the file, generally being too nosy for it's own good. After observing enough suspicious traits in the file, the Static Heuristics detective will take it in to the station.

Dynamic heuristics scanners get to have all the fun, because in our digital world no file or virus has any rights to speak of, so entrapment is not only legal… it has become an important tool in detection. The dynamic scanner gets to take a file and place it in a controlled environment, hopefully indistinguishable from the real world, where it is shown tempting treats. If the file jumps up and starts picking pockets, and actively seeks out new victims, then writing parts of itself to other files… well then, we have a highly suspicious file, possibly even a virus. The advantage is that the dynamic method gets to catch the virus-like activity without risking any *actual* files. So that potentially infected file can be quarantined before any real damage can be done to your system, and this is obviously preferable to the behavior blocker, which only works after the virus code has executed in your systems real time and on real files.

So to be fair to the scanner and behavior blocker, lets take a quick shot at heuristics. The main disadvantage is that heuristics are only as good as the code used to write them, and they are all different. And with all their differences come a slew of different ways to generate false positives, I present the following from Anthony Harrington's "Blind Mans Buff," since it illustrates this point quite nicely:

> In May this year McAfee was back in the headlines when it thought it had found the Homepage virus in a newsletter sent by Sophos.
>
> The mix-up was caused by McAfee's heuristic engine detecting the string ?VBSWG' (the toolkit used to create the homepage virus) and a quoted filename with a double extension.
>
> This was enough for McAfee to block the email. While McAfee was quick to point out that it detected the virus without the need for an update, it illustrates that heuristic engines need to be very carefully designed to avoid this kind of problem.

> In October, Norton caused trouble when its heuristic engine falsely detected a trojan in the MSN web site. The suspect file (www.msn.co.uk/webinclude/mc.vbs) was in fact completely harmless and Norton had to release an update for its software so that the web site could be visited.
>
> Similar problems can occur when antivirus suppliers try and write detection code. Norton got itself in trouble (again) with F-Prot and InstallShield in November, when a badly written definition file started detecting the Nimda virus.
>
> It claimed that executables used by both programs were infected and either quarantined or deleted the files, rendering them useless. Norton had to release an update that correctly detected the virus and release instructions on how to recover files affected by the mistake. [HA]

The main advantage is also quite nicely stated in the same article, "If your scanner doesn't have a heuristics capability, you're wide open to new viruses and variants." [HA]

So What's So Great About Heuristics Then

By now, I hope you're not asking this question. The great thing about it is that it provides a means for protecting your systems and the information you store on them from unknown viruses. There is certainly value added in that one simple fact. Every network administrator has probably taken a moment to shudder when a particularly nasty bug gets loose on the internet, just as surely as we each breathe a sigh of relief once we verify we are safe from it. It seems that despite our best efforts, there are will always be folks out there trying to make our lives interesting.

Putting it to Good Use

Certainly, the first good thing you can do is get to know your personal virus scanner. While it seems that most virus scanning companies seem to be trying their level headed best to use a standardized naming convention for the known viruses, each one has it's own proprietary scheme for how it identifies unknowns that their heuristics engine finds. Symantec's Bloodhound uses a very logical naming convention for labeling what it detects (fig. 1). Each one is designed to clearly identify it as a heuristics detected alert, and to identify the type of infection it believes it has detected. You know immediately when you see an alert for a Bloodhound.WordMacro [SYM2], that it is possible that you have a previously undetected Word macro virus on your system. Now you have a pretty good starting point for your response to the alert, you have an idea what you might be up against, and you can tailor your response to the type of alert.

Alert	Type of possible infection detected
Bloodhound.ResCOM, Bloodhound.ResEXE	memory-resident virus
Bloodhound.DirActCOM, Bloodhound.DirActEXE	direct infection virus
Bloodhound.HybridCOM, Bloodhound.HybridEXE	file virus
Bloodhound.WordMacro	Word Macro Virus (in Word 6, 95, 97)
Bloodhound.ExcelMacro	Excel Macro Virus (in Excel 5, 95, 97)
Bloodhound.Poly	string detection for some polymorphic virus
Bloodhound.Boot	boot sector virus
Bloodhound.MBR	Master Boot Record (partition) virus
Bloodhound.File.String	file virus detected by IBM string scanning engine
Bloodhound.Boot.String	boot/MBR virus detected by IBM string scanning engine

Fig 1. [SYM2]

Fine-tune your settings to fit your needs, this is probably the simplest thing of all to do, although it can take time to complete. The first and most obvious point is to make sure heuristics is enabled. I know that this may seem a bit of a silly thing to mention, but if you have users on your network with any degree of control at all over their scanner configuration, then it might be wise to check all the workstations and ensure that everything is still on. You might be surprised at how many users will disable their virus scanners and "forget" to re-enable them. While you're at it, remove any user level privileges to change the configuration of the virus scanner or disable it. The only thing they should be able to control is when to initiate a manual scan.

Once you have the business of enabling everything and ensuring that only authorized persons have access to the controls taken care of… pick a couple of workstations and start testing some configurations (I would recommend using regular systems for this vice test or lab systems, since you will get a better feel for how it will perform in your actual network environment). My favorite way to do this is to start with the pucker factor set to maximum. If a file sneezes, I want to see if I get a virus alert. This will help with several things, first you will learn a great deal about your scanner and how your particular heuristics engine behaves, and second you will soon find your threshold for false alarms. Don't be surprised if you find yourself chasing a few ghosts when you first do this, that's the point.

Now consider how you configure your scanner to treat infected files. Symantec's Norton Antivirus gives several options, at home I prefer to set mine to attempt to repair the file and quarantine if unsuccessful when. At work I take a much different approach, I lean heavily toward the Quarantine option there and I'll tell you why; if through some strange turn of events a virus is actually detected on my network, I want to know who, what, when, where, how, and why it got on my network. Of course since I work in a closed environment, on a closed network (which is not even connected to the Internet) one might reasonably expect that we don't have to worry much about viruses, right? Wrongo bucko! We have as much reason to worry as anyone else, because they do happen on isolated LANs, because with floppies, ZIP disks, and now USB thumb-drives there are plenty of remaining avenues to introduce a virus. We won't even go into the possibility that someone might try to use a modem to circumvent the security you have in place and dial themselves up to the Internet from their desktop. One more thing, if you do get a successful repair on a heuristics alert, then you have effectively burned the body… now you will not be able to find out what the virus was. You need to be able to send a sample to your vendor for analysis, then they can tell you what you have, that's also being a good neighbor.

Won't you be My Neighbor

So how does any of this equate to being a good neighbor? Well, a well-documented incident, with a sample of the virus presented to the right agencies will save a lot of people headaches. For example, you'll recall that earlier I mentioned that neither Network Associate's McAfee VirusScan nor Symantec's Norton Antivirus detected the Melissa virus. Well if you happened to be one of the lucky users of Command Software Systems antivirus program, you were the only folks protected (by heuristics) when Melissa hit the streets. (See [LM], they are rightfully proud of this fact.) But if you happened to be one of those lucky folks, and none of you bothered to report the incident, then there would be no sample of the virus to use for developing ways of protecting systems from it.

In this case it's pretty easy to see, there is no personal gain from taking the time to send a sample out and report the infection… except of course the satisfaction of knowing that you may have helped make the internet a little bit safer again. That's what being a good neighbor is all about.

Now Make it a Part of Your Policy

Now we're starting to get to where the rubber meets the road, because we can talk all day long about what heuristics is and how great it is… or isn't. But if we only give it lip service and don't take the time to put our beliefs into some kind of action, then we're all just wasting our time. Here comes the pitch, include heuristics in your antivirus policies, both personally and professionally.

Why personally, because at home I have a radically different computing environment than I do at the office. At home I'm directly connected to the Internet through cable, my network is continually at risk from hack attacks and viruses downloaded from the mail or web pages. At home I have a policy of "how much can *I* handle." That is, how much grief and tracking am I willing to do before I make the Bloodhound heel. Well, frankly I've given the dog all the leash I can on my system and so far I haven't heard a peep from him. Now I am without a doubt the most adventurous surfer in the house, so I have had a few shouts from the standard scanner. And it seems that some folks still haven't gotten all the "Real Story of Snow White" bugs worked out of their e-mail programs "hahaha," and a heavy sigh when we see that one again. But I haven't been chasing after wild geese like I thought I might be, so I leave it cranked up.

Professionally I have a slightly different approach, now it's no longer my own threshold for alerts that I have to concern myself with. I have to take into consideration my customers, my users. In a perfect administrators world we would be able to tell all of our users what is best for our network… er, that is what's best for *them*, and they would just toe the line. In the real world, they are really our customers, and we have to keep them happy if we want to keep working. So we have to strike a delicate balance between the security of our network and the usability of it, fortunately there is always an owner that makes that final decision so you don't have to bear the full brunt of the more unpopular policies. At home I can shut down the modem and find out what caused an alert without suffering anything greater than perhaps annoying one of the kids who wanted to go to www.Disney.com. At the office, if I take us off the network for a virus whether real or just potential, time and money are lost. As you can imagine that is not a very good thing from the perspective of management.

So at home we can have an informal policy, which is as flexible as it needs to be, but at the office… it needs to be carved in stone and the name of the big cheese better be chiseled in next to the last line. That advise in mind, I offer the following as a sample of an antivirus policy including heuristics.

A.1 Antivirus

A.1.1 All MyCompany systems which are used as servers or workstations will have antivirus software installed and configured using the following guidelines:

- Antivirus programs will be configured to 1) quarantine infected files upon detection, 2) copy infected files to quarantine before attempting repair, or 3) attempt repair and quarantine if unsuccessful. This is a prioritized list of settings, choose the highest priority setting available for the antivirus program installed. Quarantining of files is desired to facilitate the investigation process during a virus incident.

- Antivirus programs will be configured to automatically start at system boot, and scan all files on access. Heuristics scanning will be enabled at the default level of protection for auto-protect and manual scan modes at all workstations and servers. Laboratory workstations and servers will have heuristics enabled at the highest level of protection for auto-protect and manual scans to ensure highest possible level of protection while testing software. Laboratory system settings may be adjusted to meet the requirements of approved test plans.

- Automatic update will be configured on all workstations and servers to occur without requiring user intervention. Current definition files are required to ensure the highest level of protection at the workstation or server.

- Antivirus programs will be configured to perform a complete system scan on a weekly basis at a time selected to ensure that daily operations are not impacted by the task.

- Antivirus programs will be configured with full logging options enabled. Complete logs are required to facilitate the investigation process during a virus incident.

- Antivirus program log files will be of a sufficient size to prevent overwriting of logged events during a virus incident. The Information Systems Security Officer (ISSO) will review the logs monthly. Logs may be cleared at the server/workstation by the ISSO providing the log has been copied to a secure location for archival purposes. A secure location may be defined as a directory or folder where access is limited to the ISSO and his/her alternate. Logs will be backed up to Compact Disk (CD) or DVD media at the end of the month for long term secured storage in order to preserve network storage space. Once backed up and verified, the logs may be removed from the ISSO directory and the disks must be stored at a secure location.

- Log Backup Disks must be clearly labeled with the network name, and the month and year of the backup, version numbering of backup disks is required only if the backup must span two or more CDs. Logs will be retained for 5 years

- The ISSO may use data reduction software to collect logs and generate reports. Commercial Software and/or Freeware must be fully tested and have been approved by the Configuration Control Board (CCB) prior to use on the network. “Homegrown” scripts/programs may be created and used to collect logs and generate reports after receiving the approval of the CCB.

- The Information Systems Security Officer (ISSO) will perform spot checks of at least 10% of the active systems bimonthly to verify settings.

- Infected systems will be disconnected from the network immediately.

- Alerts on a system from a known virus will be documented, logs and files will be secured for further investigation (as required) and the system will be confirmed clean before being reconnected to the network.

- Alerts on a system from an unknown virus (detected by heuristics) will be documented and logs and files will be secured for further investigation. The ISSO will follow the established procedures of the Antivirus program vendor to ensure a sample of the virus is submitted to the vendor for analysis. Vendor analysis will be authoritative as to the type of virus and methods required to clean the infected system. The system will be confirmed clean before being reconnected to the network.

This is of course just a piece of the pie when it comes to policy. A comprehensive policy will probably be significantly longer, just trying to keep it simple and still cover everything will drive you to distraction. You may consider writing a different policy for each platform; Windows, Macintosh, Unix/Linux, Solaris, etc. Much of what you see here is considered site specific, in that what applies to one site, won't necessarily work at another. You may not have a lab, or you may only use one vendors antivirus program. I was taught to use at least two antivirus programs, one on the servers and a different one on the workstations, hence the multiple choices possible at the first bullet. The most important point to come away with here is that if you have a policy, update it to include heuristics and enforce it. If you don't have a policy, write one and get it signed by upper management, then enforce it. A signed policy is beautiful thing to a security officer, it's an order that everyone has to follow and that saves you from having to justify your actions to anyone other than your boss.

Wrapping it up

We covered a lot of ground here, starting with what heuristics is (in a nutshell) and giving an idea of how it works to defend your systems. We looked a bit at how it is really our best line of defense against the unknown viruses. Knowing that it is not perfect, just like every other antivirus tool, we discussed some of the ways we can best put it to use on our systems and networks. And I couldn't resist plugging "being a good neighbor," after all one of the most important points to remember is that it is heuristics is designed to alarm on the things that the definitions don't catch, the unknowns. When you get the first heuristics alarm you've ever seen, get that thing to your vendor if there is any way that it can be done. And finally, for the sake of covering your tail, we covered creating policy to give you the support you need to do what you have to do to protect your systems and networks through antivirus scanners and of course, heuristics.

And just a quick note to the reader who may be wondering, my interest in this particular topic was born about four months ago. I was in the unique position of being able to watch an entire node of a wide area network get shut down for 24 hours due to a heuristics alert. Unfortunately, the sysadmin at this site had the scanners configured to delete infected files. You can guess what happened next, the site was off the WAN for 24 hours and they didn't even know if it was a real virus or a false alarm. I decided two things then… 1) I was glad I wasn't that sysadmin and 2) I had to do everything I could to make sure I never was in his shoes. I hope you're never in his shoes either.

Sources/Bibliography

(n.d. indicates no date)

[IDI] In-Defense Inc. “A New Technology for Protecting Computer Systems Against All Kinds of Viruses: In-Defense ™ (Intelligent Dynamic Defense)“ ©1999 URL: http://www.primex-synergy.com/datadefense/indefense/whitepaper.doc (15 March 2002)

[GJ] Gritzbach, Jan with Odenhal, Petr and Zahrednicek, Petr “How to Detect Unknown Computer Viruses Using Heuristic Analysis” August 1998 URL: http://www.grisoft.com/unique/wpheur.doc (15 March 2002)

[GD] Gryaznov, Dmitry O. “Scanners of The Year 2000: Heuristics” n.d. URL: http://vx.netlux.org/texts/html/scan2000.html (15 March 2002)

[HA] Harrington, Anthony “Blind man’s buff’ 05-12-2001 URL: http://vnunet.com/News/1127335 (15 March 2002)

[HJ] Hruska, Jan “Computer Virus Prevention: A Primer” August 2000 URL: http://www.oucs.ox.ac.uk/viruses/documents/artdef.pdf (15 March 2002)

[LM] Landesman, Mary “The Computer Virus Continuum” n.d. URL: http://www.commandsoftware.com/products/continuum.html (15 March 2002)

[SM] Schmall, Markus “Heuristic Techniques in AV Solutions: An Overview” February 4, 2002 URL: http://online.securityfocus.com/infocus/1542 (15 March 2002)

[SYM] “Understanding Heuristics: Symantec’s Bloodhound Technology” 9/97 URL: http://securityresponse.symantec.com/avcenter/reference/heuristc.pdf (15 March 2002)

[SYM2] Symantec Knowledge Base “Explanation of Bloodhound Alerts” 07/21/99 URL: http://service2.symantec.com/SUPPORT/ent-security.nsf/pfdocs/1998100109260548 (15 Mar 02)