Global Information Assurance Certification Paper

Tips for Making Security Intelligence More Useful

GSLC Gold Certification

Author: Mason Pokladnik, mason@schwanda.cc

Adviser: John Bambenek

Accepted: June 26, 2008

© SANS Institute 2008, Author retains full rights.

Security Intelligence Cycle

I work in a small IT department – there are only 4 operations people – so the idea that anyone could work on IT security full-time is almost out of the question. At the same time, we still face all of the threats any Internet connected organization faces, whether they are a small business or a Fortune 500 company. I try to keep up with new threats a little bit each day, but the amount of information out there is just too much to review in a reasonable amount of time. In this paper, I will share some tips I have learned along the way to cut through some of the clutter out there and focus on what is relevant. Then, share some tricks for analyzing what is left in order to produce actionable information that can then be used to implement temporary and long term controls.

While a detailed description of the intelligence cycle is outside the scope of this paper, a quick summary will help us understand what a formal intelligence program looks like. Not every organization has a dedicated information security intelligence team, but everybody involved in information security, at least informally, performs parts of the process whenever they are exposed to new information about a threat or control. This paper is really about taking diverse pieces of information and analysis and forwarding them on: 1. to the right people and 2. in a usable format. Along the way, the information will have to be analyzed and packaged for decision makers and other audiences.

I am going to borrow the cycle model[1] from the U.S. intelligence community, since that is what they do for a living. This is to help organize the information - not a recommendation that you need to follow a strict process.

The first step of the intelligence cycle is the planning phase. In general, IT security is a responsive organization. They wait for a threat to emerge and then try to setup a control to prevent that threat from being exploited – or failing that, at least detect that it was exploited. However, since people are actively trying to locate weaknesses in our networks, we need to respond by becoming aware of the same information about our own networks. That way we can decide what should be secured and what will wait until circumstances warrant action.

The second phase is collection. Through various sources, we attempt to improve our available information about the threats we face. Whether it is as simple as reading a website or as complicated as posing as a dealer in stolen identities in a chat room, the information gathered is the raw material that through the next two phases becomes useful input to the risk analysis processes.

In phase three, processing, information is organized, graded for reliability and filtered for relevance. Some information can be discarded at this phase, such as a vulnerability in a database that is not present anywhere on your network - or any connected partner networks.

The fourth phase is analysis. Using human judgment, someone must interpret how the information you have collected impacts the organization. By examining exploit vectors and your existing controls, the likelihood of a threat occurring and potential damages it could cause can be assessed. The quality of the analysis is directly linked to the experience of the people performing it and the information they

[1] http://www.intelligence.gov/2-business.shtml, Visited 5/2/08

have available. For example, understanding the scope of the information that could be lost in a SQL injection attack requires knowledge about what is in the database.

Finally, we reach the last step dissemination. The now analyzed intelligence is packaged in a appropriate form for decision makers, IT staff, end users, etc. and distributed as needed. Depending on the threat posed, a reaction may be needed immediately, as time and money allows or never.

The intelligence cycle is intended to be a loop. When new systems and controls are deployed, they become new targets; therefore, new threats should be tracked, and new information learned. Then, decisions made as a result of current intelligence information should guide future planning and information gathering phases.

Using the intelligence cycle phases as an outline, I am going to share some tips that have allowed me to reduce the time through one or more of the phases and produce a useful analysis that could then be used by decision makers.

Phase 1 – Tips for planning

The quality of any intelligence product is highly dependent on the people involved in producing it. If you are responsible for delegating responsibilities to people, consider giving intelligence functions to your more experienced staff. The ideal person would be someone with significant experience in operations and security. Operational experience gives an analyst insight into the day to day processes - such as backups, patch\change management and network architecture – that have an effect on implementing an organization wide response to an incident. Experience designing and administering security controls gives the analyst knowledge of the current defenses – including known weaknesses. The combination of those skills allows the creative application of existing tools to add another layer of defense-in-depth while more permanent solutions are explored. In the unlikely event that you have many candidates to choose from, then experience in the fields of IP packet analysis, writing rules for intrusion detection systems, incident handling, forensic techniques and the ability to perform behavioral and binary analysis of unknown files are all excellent things to see on a resume.

If you are responsible for intelligence collection and analysis, make sure you know your own limitations. No one is an expert on everything. Try to identify people you can go to for help when you run into something outside of your experience. Make friends with these people - as you will learn over time to trade information.

Phase 2 – Tips for information gathering

There are many more sources of potential information out there than anyone has time to keep track of. Besides all of the information available through blogs and more traditional websites, mailing lists, vendor notifications and a myriad of other external sources, our organizations are full of potentially useful sources. Since our goal is to get the best information, with the least amount of resources, the trick is to identify high signal to noise ratio sources. In most electronic devices, there is a certain amount of interference that can enter into a circuit and degrade the signal level of the information passing through. Think of static in an analog radio station. According the inverse square law every time you get

twice as far away from the transmitter of the station, the power of the signal you receive becomes four times weaker. As you get further and further away, the level of background noise in the signal relative to the music you were hoping to listen to gets higher and higher, so you hear more static and less music.

If we apply this concept to our information, there are many sources that generate a lot of information, but very little of it may be relevant to your organization. Back when I started trying to keep up with new threats, I used to try and scan through most of the postings to two mailing lists: bugtraq[2] and full-disclosure.[3] While this kept me quite well informed about hundreds of application vulnerabilities and websites vulnerable to cross site scripting issues, the time involved reading compared to the actionable information I got out of it was completely unbalanced. Over time, I have stopped following full-disclosure, and I will scan the subject lines of bugtraq for interesting content, but the time I am willing to devote to those mailing lists has decreased substantially.

Here are some of the sources I have found to be useful and high signal to noise; organized by how often I use them.

Daily

Internet Storm Center – There are many security sites out on the Internet, but the Internet Storm Center (ISC) has the dual benefits of having both what is probably the largest distributed IDS system in the world, and a large readership that actively sends in a wide variety of threat information. The most popular feature is the diary page[4] where the handler of the day summarizes any newsworthy events or just some tips and tools to help with incident handling. I can easily say this site is my number one external source of reliable threat information as the handlers are usually very careful to detail what is known about a threat – and just as important what they do not know – as well as any useful workarounds.

IDS – If you have a well tuned IDS system, it can certainly be a useful source of information. The easiest data to get out is real alerts, but most of the intelligence value lies in trends. From a tactical point of view, identifying one computer infected with malware gives you an incident that needs to be addressed. What you should be looking for is sudden spikes where multiple computers seem to be infected. That kind of trend should lead to a deeper examination of the underlying cause. Is there a misconfiguration or vulnerability that is allowing the automatic installation of software? Once the root cause of the trend is identified you have information that can be analyzed and sent to decisions makers.

Even if your IDS is generating thousands of alerts everyday and you are pretty much ignoring it, if you can come up with a way of categorizing and summarizing those alerts then the volume of alerts in each category can still give you some insight into what it going over your network. A good example of using

[2] http://www.securityfocus.com/archive/1, Visited 5/2/08
[3] http://lists.grok.org.uk/pipermail/full-disclosure/, Visited 5/2/08
[4] http://isc.sans.org, visited 5/2/08

this trend analysis in the face of an overwhelming number of alerts was given in a spring 2008 presentation by a fellow SANS Technology Institute student.[5]

Darknet/Honeypots – With the enormous amount of information that organizations are logging now for compliance purposes, it is becoming more and more difficult to pick out the important security details from the avalanche of logged events. A reasonable response to this problem is to deploy alert systems that should never be tripped in the first place.

I prefer to deploy these in conservative locations on internal networks. You can setup a honeypot in your DMZ, but eventually any Internet routable address is going to be pinged, port scanned and have random exploits sent at it. Even deployed on an internal network, network configuration issues and curious users will cause enough false positives. Still the signal to noise ratio of events out of these types of systems should be considerably higher than a typical IDS or system event log.

News wires – While the news wire services are not exclusively security focused, there are many threats to availability that do not originate on the Internet. I try to keep track of several news sources on a regular basis to make sure that hurricanes, snow storms, and other semi-predictable events do not catch us by surprise. There are numerous other tidbits including pending legislation, new technologies, and many other topics that can affect security, architectural and purchasing issues for your environment. In addition to the Internet Storm Center, I visit the following sites nearly every day.

Slashdot.org provides a nice daily summary of what fellow geeks think is interesting on a wide variety of topics. The moderators do an excellent job of summarizing the topics discussed so that the headlines can be reviewed in just a couple of minutes, and you can choose to follow any story for more details.

Drudgereport.com is another headline aggregation site. I will forewarn you that the site's editorial staff does not shy away from sharing their political views. What the site does excel at, is monitoring the news wires (United Press International, Associated Press) and hundreds of other news sites, then posting links to that content with amazing rapidity. If there is a major news event unfolding, the Drudge Report will almost certainly be on top of it 30-45 minutes before other news sites gets stories written, edited and posted.

CNN.com may not be as fast as the Drudge Report on a lot of breaking stories, but with more news bureaus and reciprocal coverage agreements than anyone else, CNN will get video coverage or some onsite report of many events. They have further increased their abilities by soliciting "Ireports" from citizen journalists and their ubiquitous cell phone cameras.

Most days I can cover these sites in less than 15 minutes. I have a folder of bookmarks in Firefox that contains all of the sites that I use on a daily basis. By using the "open all in tabs" choice at the bottom of the folder, all of the websites are loaded and ready to be reviewed. When something interesting does come up I can email a link or a brief summary to my department or just discuss it at lunch. Occasionally, a more formal process is invoked such as an alert to those responsible for monitoring the disaster

[5] Meyer, Russell. **Creating Actionable Information from Intrusion Detection System (IDS) Alerts.** http://www.sans.edu/resources/student_presentations/, visited 6/9/08

recovery process whenever a hurricane may be headed for landfall. A fact I usually see first – along with a quick link to a tracking map – on the Drudge Report.

Weekly

Incident reports and helpdesk tickets – Our end users are not all security experts, but humans are quite good a noticing differences. When a computer starts "acting funny," (e.g. running slower, programs start crashing, pop-up ads appear, etc.) end users are quite capable of noticing the difference and notifying the helpdesk. That makes helpdesk tickets another good place to look for trends. A large uptick in complaints about performance or pop-ups could indicate a break down in your defenses. It is also a good idea to periodically review security incidents for trends as well. As the saying goes "sometimes it is hard to see the forest for the trees." When you are busy responding to an incident, thinking about how it fits into the overall picture is not the primary concern. Since I work in a relatively small department, tracking both of these is relatively easier for me than it would be in a larger organization. First, I am involved in the majority of all security incidents, and second, we have consciously designed our space so that it is easy for my boss and I to overhear many of the helpdesk calls. This keeps us situationally aware of what is going on much of the time, and makes reviewing tickets and incidents a faster process.

Log sources – Over and above monitoring some of my daily early warning systems mentioned earlier, I try to use information gathered from other sources like the Internet Storm Center to mine my gateway logs for suspicious traffic. Typically, whenever a new piece of malware hits, such as the Storm worm, there are multiple pieces of information that are useful as search strings into your logs. Some examples include: DNS domains, IP addresses, user-agent strings for software that phones home using HTTP/HTTPS, less used ports 6667 (IRC) and odd protocols (overnet peer-to-peer, etc.)

I then search for that type of information in my proxy logs, DNS resolver cache, anti-virus alerts, and any suspicious files I may have collected lately. If you have other resources available to you such as full packet captures of your network or a massive centralized logging system that you can query, you might be able to get additional useful information from those sources as well. For a few additional ideas along these lines, I recommend you check out a presentation by Mike Poor on Network Early Warning Systems.[6] Slides 39 and after describe using LaBrea as a honeynet to capture traffic sent to unused IP addresses both in a DMZ and in unused network space.

Newsbytes – Bi-weekly I read the SANS Newsbytes email newsletter. It is yet another aggregation of security related news stories and threats. By now you have probably noticed a trend yourself, I like to let other people do the work for me when possible, but Newsbytes comes with a bonus. Many of the headlines are followed by commentary from an editorial board comprised of real security experts.

[6] http://www.intelguardians.com/mikepoorkeynote.pdf, visited 5/10/08

Monthly

Crypto-gram[7] - Available as an email newsletter, running blog entries and now a podcast, Crypto-gram is Bruce Schneier's ongoing commentary about security in its many forms. While he often departs from cryptographic topics, nearly every tangent still reminds you that what may be the most important consideration is how security fails. I can easily get mired down by politics, usability issues, implementation details, etc., and this once a month reminder to backup a step and reexamine how a control might fail may help you decide whether a security change you are considering really makes sense. Reading about other people's bad decisions can help us to not follow in their footsteps.

Trade time for money

Do you have more money than time? Then you might consider buying intelligence from a variety of sources. There are quite a few companies out there who sell what they call security intelligence information. Two of the more novel services I have seen offered are information on "zero-day" and unpublished exploits and external brand monitoring that notifies you in the event that your organization's brand is being used for a phishing attack or some other scheme. Outside of that, most services seem to be collecting and offering preliminary analysis of information you could have gathered on your own if you had the time. If you use an outsourced security service, then hopefully the service provider is monitoring for new threats across multiple organizations and making recommendations for new permanent and temporary controls.

Selecting a vendor can be a little tricky. You can review the public version of the alerts that some vendors, such as Secunia, iDefense (Verisign) and Tippingpoint (3Com), make available after vendor patches have been released, and then decide if knowing that type of information earlier than the general public will help protect your organization. Vendors need to provide actionable intelligence including ways to identify an attack, workarounds if available and an accurate assessment of the threat level. Your response to a threat will be different if a vulnerability is being actively exploited than if an exploit was purchased, but never used. If a vendor can tailor reports to your environment and filter out unneeded alerts, then that further increases the value.

Supplementary Research

Remember that we are going to grade information on timeliness and relevance later in the intelligence cycle. When picking a source try to remember those factors. The alerts that the U.S. Computer Emergency Response Team (CERT[8]) put out are often well researched and written. The problem is, they are usually not the first place you could have learned about that information. If CERT email alerts are your primary source of information, you could be artificially reducing the time you have to analyze and prevent a new threat. The reason they are not in the list above is the majority of the time when a U.S. CERT alert arrives in my inbox I use it to improve my analysis of a threat, not to start one.

Another set of email alerts you should plan on reading are the security alerts from your important vendors. Microsoft, Cisco, HP and nearly every other large commercial vendor has a security email list

[7] http://www.schneier.com/crypto-gram.html, visited 5/10/08
[8] http://www.us-cert.gov/cas/techalerts/index.html, visited 5/2/08

that you can sign up for that will alert you to new vulnerabilities and the availability of patches. Attackers are growing increasing skilled at reverse engineering patches to design an exploit against the unpatched computer population. You do not want to find out that a patch is available because of an incident.

Phase 3 – Tips for processing

In the processing phase we need to filter, manipulate and grade any remaining information to prepare it for analysis. Unfortunately, much of the rest of the world has a slight advantage over us in this step. Many of them can understand English. How many of us read Chinese or Russian? This somewhat limits the total amount of information we may be able to gather on a threat, but that can be offset by what standards we do share. IP and the various protocols that ride on top of it still deliver nearly all Internet traffic. All Intel X86 compatible processors still run the same assembly level instructions, and most of those computers run Windows.

The less formal your approach to the intelligence process the more likely the lines between information processing, information gathering and information analysis will tend to blur. I do not follow a step-by-step approach myself, and I often find myself researching and gathering information to further an analysis. Just make sure that you are not completely ignoring any step, especially the dissemination step!

Filter information

Know your network – If the exploit *du jour* is against PHP based websites and you only have ASP.NET application servers, then it should be fairly safe to ignore that piece of information for now. What would be a problem is if you did have some PHP application servers and did not know about it. This is the main reason why we want experienced operations people who know the network involved in the process. You will also want to have seen the results of any recent network scans that could have identified an unauthorized server on the network, as well as, know about any exceptions to standard configurations. This type of information may be available in configuration management databases or network aware tools like the RNA[9] add-on to the Sourcefire IDS sensors.

Manipulate information

Other than some of the tools for identifying trends that we have already covered, one of the situations I have to deal with is gathering data on an unknown executable file. All such files in our company are automatically assumed to be malicious until we can determine otherwise. In the past few years, one of the malware analyst's best tools was using VMWare, Virtual PC or another virtualization product to perform behavioral analysis of a file to see what it did. VMWare allowed us to take a snapshot of a machine prior to infecting it, and then when an analysis was complete or needed to be repeated we could revert back to that snapshot and have a clean system waiting with all of our tools ready in a few seconds. Malware authors have taken notice of this and responded by adding various detection and countermeasure to analysis in their software. This has lead to an ongoing cat and mouse game where

[9] http://www.sourcefire.com/products/3D/rna, visited 5/10/08

security analysts try to improve the speed and quality of their analysis and malware authors try to slow them down. On the security analyst side, we can use a wide range of tools from slower static analysis of programs to new automated dynamic analysis systems which emulate computers in a variety of ways.

Static Analysis

Static analysis techniques include debugging, disassembly and manual behavioral analysis of programs. Static analysis is often time and labor intensive, as well as, requiring a highly trained analyst. While a malware author can use a wide variety of anti-debugging tools, virtual machine detection, binary packing tools and encryption, in the end all they can do is slow down an analyst skilled in reverse engineering as they peel away each layer of protection to reveal the program itself and then figure out what it does. Static analysis is highly effective, but not scalable. With today's malware that can modify itself for every download we have to be able to analyze programs and their intent faster.

Dynamic Analysis

Computers are extremely well suited to performing repetitive tasks, so it is natural that the security community would attempt to create automated solutions to analyze malware. Several different approaches to the problem exist and each of them has its own strengths as weaknesses. Before I continue with my tips, a quick tour of the technology is warranted.

Sandboxes have been around since IBM started using them in mainframes in the 1960's. The concept is to run a program within an isolated environment. Recent incarnations of the idea include JAVA, Flash Player and virtual machine software such as VMWare. The security community has started using a wide variety of different sandbox environments in the quest for an automated analysis platform. The following are the most common types of sandboxes used for analyzing malware today:

Hardware assisted virtualization – This category includes the various virtual machine programs, such as VMWare, Virtual PC, Xen, etc. These tools allow entire "guest" operating systems to run as a process on a "host" operating system. To increase the speed of execution they allow unprivileged operations to run directly on the processor. However, privileged processor operations (or opcodes) are intercepted and handled through an emulated processor. These tools have been a staple of many security analysts' toolkits since they reduced the number of physical machines needed, as well as, helped speed the analysis process. For that very reason malware authors are specifically writing detection mechanisms into their code that alter or disable the program from running in a virtual environment. This greatly reduces their usefulness as an automated analysis platform.

Emulated systems – Emulators run in software only and are not currently as widespread as the virtual machines we see in wide use. They may only emulate hardware or they may include an emulated operating system as well. Emulators can still be detected through the lack of support for a CPU or operating system function, but it is more difficult and the techniques for doing so are not yet in wide use. One way to detect software emulators will be with exploits for real versions of windows. If the software emulator does not contain the same vulnerabilities as a real windows system, you might be

able to detect the difference.[10] Examples include Norman Sandbox and antivirus behavioral/heuristic detection engines. Although they use the same technology, a sandbox and behavioral engine have very different goals. A behavioral detection engine needs to rapidly assess what threat an application possess, and then make a decision on whether to stop it from running. A sandbox used for dynamic analysis will allow a suspicious program to run because we want to monitor its actions. Due to these different aims, what code is out there to detect an emulated system is primarily aimed at defeating or avoiding detection by behavior engines.[11]

System call hooking – Instead of trying to emulate parts of a computer, some sandboxes implement their monitoring by hooking into the operating system in the same way that a rootkit does. When this is implemented in the operating system running natively on the hardware, nearly all of the techniques used to identify emulated and virtualized environments are neutralized. Just like any security software attempting to identify a rootkit, malware authors would have to catch the system in a lie to identify the sandbox. Malware authors usually want the same thing that regular developers want. They want the program that they write to run on the most versions of an operating system possible. Microsoft has been fulfilling that request by keeping the highest level of Windows application programming interfaces (API) – the Win32 API – as backwards compatible as it can. That is how the same program that runs on Windows NT – and possibly back to Windows 95 – can run on the latest versions of Windows. Since system call hooking is done at a much lower level of the operating system, it can remain invisible to many programs. A sandbox called CWsandbox uses this technique. System call hooking is not perfect either. If a program installs its own rootkit protections at a lower level than the sandbox's hooks into the system, such as using a kernel mode driver, it can then blind the sandbox from monitoring its actions after that point. [12]

Pros and Cons

The traditional VM detection methods,[13] such as looking for the memory address of the interrupt descriptor table or checking for how the processor handles certain opcodes, will not detect many sandbox environments. When malware authors do come up with detection techniques – such as calling rarely used Windows APIs - sandbox vendors can quickly respond, by implementing functions to safely pass through such calls to the native operating system. For now, using a dynamic analysis system based on emulation or system call hooking will provide superior results due to the lack of easy and reliable detection methods like the ones that exist for virtual machines. Attackers will eventually start testing their ability to evade detection on these systems just the way they now test whether they can be detected by antivirus, make debugging more difficult and alter behavior when running in a virtual environment.

Currently, I only analyze one or two files a week using sandboxes, so I have come to rely on the sandbox provided as a community service by www.cwsandbox.org. CWsandbox provides a wealth of information

[10] http://www.symantec.com/avcenter/reference/Virtual_Machine_Threats.pdf, visited 6/19/08
[11] http://wiki.castlecops.com/Technical_Stuff_antivirus, visited 5/13/08
[12] http://www.disog.org/2007/09/blog-post.html, visited 6/19/08
[13] http://handlers.sans.org/tliston/ThwartingVMDetection_Liston_Skoudis.pdf, visited 5/10/08

in several formats. A human readable logfile shows a summary of all of the activities that happened on the system including network connections and system changes. All network traffic is provided in pcap format for further analysis, and a detailed log in XML format allows you to mine for information such as network hosts that can then be checked against proxy logs as discussed earlier. The following excerpt from a sample XML output file shows:

An API call to attempt to see if a user-mode debugger is attached to the malware process.

```
<system_info_section>
  <check_for_debugger apifunction="IsDebuggerPresent" />
  <get_system_time apifunction="GetLocalTime" />
  <get_system_time apifunction="GetSystemTime" />
</system_info_section>
```

A http get request to pull down a secondary stage binary, with just a little white lie regarding the browser user-agent string.

```
<connection transportprotocol="TCP" remoteaddr="151.1.24.160" remoteport="80" protocol="HTTP"
      connectionestablished="1" socket="1368">
      <http_data>
         <http_cmd method="GET" url="151.1.24.160/project/config/b.php/../install.exe"
         http_version="HTTP/1.1">
                  <header_data>
                           <header>Host: ilsitodeiservizi.it</header>
                           <header>Accept: text/html, */*</header>
                           <header>Accept-Encoding: identity</header>
                           <header>User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en;)
                            Gecko/30060309  Firefox/1.5.0.7</header>
                  </header_data>
         </http_cmd>
      </http_data>
 </connection>
```

This level of information allows an analyst to make much better interpretations of an executable file's intentions, and combined with log file searches, allows you to make a preliminary estimate of the level of infection in your organization.

When the workload justifies it, I would like to acquire a copy of the Norman Sandbox Analyzer. A discussion\test I ran with a Norman sales representative in April of 2008 shows the product to be capable of handling batches of malicious files for processing at a relatively high speed. Sometimes the free online resources, such as CWsandbox, can take awhile to analyze a file due to a queue of files in front of yours. As discussed earlier, Norman Sandbox Analyzer uses a different technology from CWsandbox, so by using both I can decrease the likelihood of having to use static analysis against a program. Also, their product literature states that another option you can purchase is access to a list of IP addresses and URLs accessed by all files submitted to Norman in the last 24 hours.[14]

[14] http://www.norman.com/microsites/malwareanalyzer/Products/reporter, visited 5/15/2008

Virustotal – One of the other free services I regularly turn to for information on an executable file is www.virustotal.com. Virustotal provides two quick pieces of information. First, it scans the file against 32 different antivirus products to see which ones can detect your file. This can save you a little time and effort instead of having to manually scan the file against the 2 or 3 different antivirus products many companies use for defense-in-depth. Second, it runs a quick analysis to identify any packing utilities that may have been used to protect the file. The presence of a packing utility does not necessarily mean that a file is malicious, but I usually only see it in two places. One place is from commercial software manufacturers who are trying to prevent people from pirating copies of the product, and the other is malware. In both cases, the software's creator is attempting to prevent the reverse engineering of their "product," but it does not take much effort to tell the two apart. The most recent file I ran through the system in June 2008 also included a brief report on the binary from Norman Sandbox Analyzer. I am unaware if that is going to be a permanent addition to the report or not.

For some additional tools you can use for malware analysis - and the malicious webpages that are often used to deliver them - I highly recommend you take a look at the following webcast from SANS. http://www.sans.org/webcasts/show.php?webcastid=90771

One additional tip for working with malicious code: do not do this using production resources. Malicious code authors have been known in the past to use time based triggers, VMware detection and a whole host of other tricks to prevent security researchers from discovering the full capabilities of their software. This especially goes for using your organization's main Internet connection. The Storm worm was the first I can recall that would automatically respond to someone trying to scan a machine to see if it was infected by having the botnet launch a distributed denial-of-service attack on the scanner.[15] Taking down your organizations' Internet connection is usually a career limiting move, so get a DSL line or some other sort of connection that cannot be traced back to your organization.

Grading information

The final part of the processing phase is grading the information and its source. When an analyst generates information themselves, using a tool like a sandbox, that information is considered both highly reliable and from a trusted source. If you gather a piece of intelligence off of a random security blog, then its source and reliability may be much lower. Grading a source of intelligence is usually a factor of experience with that source. I consider most of the information from the Internet Storm Center to be highly reliable. When they are providing preliminary analysis or do not know what is going on, they usually come right out and tell you that.

Grading information itself is often much easier. Information Security is a very fast moving topic with a lot of different people worldwide sharing information. Getting a second source verification of many things is as simple as searching on Google. Good searches include IP addresses, domain names, strings found inside an executable and file hashes. The more generic your information, the less likely you will be able to find confirmation. I have noticed it is not often that I come across a threat that no one else

[15] http://www.informationweek.com/news/internet/showArticle.jhtml?articleID=201800635, visited, 5/18/08

has experienced. When it does happen, this is the time to use any contacts you have developed with other security researchers to see if they are seeing the same kinds of things that you are.

Phase 4 – Tips for analysis

The analysis phase is concerned with taking the information you have, and distilling it down to usable intelligence information that people can use to make decisions. Usually, you are at one of the two extremes. You either have a small mountain of information that has to be analyzed for trends and summarized, or you do not have enough information to make a reasonable assessment. While in the analysis phase, remember that people will need to be able to make decisions based on the information you are presenting them. There are some questions that tend to come up over and over again, so I try to answer them ahead of time in the analysis if possible.

Intent – While understanding the intent behind a threat may not make it any less of a problem, it can give you a better picture of the situation. The intent behind the latest version of a botnet trying to work its way into your network and a competitor or government trying to exfiltrate your intellectual property is an important distinction when trying to decide the scale and timing of your response.

As a smaller organization I keep asking myself the question “Have we become "interesting" enough to encounter economic or other espionage attempts?” In reality it is about how interesting your information is to attackers. If you are in a smaller organization, but have employees with Department of Defense top secret clearances, you could easily find yourself under attack.

Threat – Understanding the threat itself is really the whole point, but in some cases it can be frustratingly difficult. This is one reason that sources of information are graded. If you can corroborate what you are seeing on your own network with a security vendor and/or multiple sources on the Internet, then you may have more confidence in your assessment. Classic mistakes at this point include being unwilling to admit what you do not know, incorrectly rating the trustworthiness of your sources to support a preconceived hypothesis, and telling people what you think they want to hear. Also, threat vectors and techniques can change quite rapidly, especially if an attacker thinks they might have been detected. Make sure as updated information becomes available, it gets incorporated into your analysis and distributed as needed.

At risk assets – If you can figure out what systems are already affected, and which are potentially vulnerable, then management can combine that with how critical those systems are to determine an appropriate and prioritized response.

Time to respond – How immediate is the threat and are there any workarounds or temporary controls that can be put in place to mitigate the vulnerability until it can be fixed? Many gateway products include rules engines that will allow you to detect or stop a new threat before it enters your network. If you can quarantine or detect a threat using your Firewall, IPS or Email/Web gateway, then you may be able to give decision makers the ability to respond in a more considered and cost effective manner. The case where this comes up nearly every month is patch Tuesday for Microsoft products. Decision makers

are faced with a repeating set of issues on how fast to put patches into production. If a workaround can effectively reduce a threat, some patches may not need to be tested and deployed at a breakneck pace.

Malicious files – In addition to the overall threat information, whenever analyzing an unknown file I try to answer the following questions:

What does the program do? Display pop-up ads, keylogger, spam, ID theft, worm, espionage

What are its interactions with the host system? Files installed, services created, network traffic

What does the distribution target seem to be? Is it targeted specifically at your organization or a more general distribution?

Is the file detected by your security controls, such as Antivirus, HIPS or network gateways?

Phase 5 – Tips for dissemination

By now you may – or may not – have a decent amount of information on the threat, but now you must package that information in an appropriate fashion for different audiences with different needs. Very rarely do I send out alerts to the entire organization. When I do, they are specific, relatively brief and when possible, include an example of what to look out for. On the other hand, I send out more technical alert information to the rest of the IT department on a regular basis.

The following tips are more personal observations based on what has actually elicited some sort of response from my user community.

Try not to be "The Boy Who Cried Wolf" – repetition may be the key to learning, but sending every single security alert to the whole company just means they will stop reading your emails. I try hard to limit the distribution of alerts to the appropriate parties. Many recent spear phishing attacks like the Better Business Bureau, federal subpoena, and IRS refund attacks rely on publicly available information. Therefore my distribution list for these alerts can usually be built with a few minutes spent searching Google.

A picture may be worth a thousand words. If you can show people what to look out for – even if it changes – they will absorb the information better. Humans are symbol processors. We can easily take the context of an example Paypal phishing message and apply that to any of a thousand financial institutions. Unfortunately, there are many other ways to be fooled.

If you identify a new trend or type of threat, consider passing that information on to as many IT staff as possible. The people responsible for designing and implementing your security controls are a natural audience, but helpdesk technicians will see many of your security incidents first. Operations staff can think over their areas of responsibility and recommend improvements in system configurations. Programmer's can consider those attack vectors in their own code and future designs.

If you are willing to deal with a potential flood of email, then try adding a tag line to your alerts informing people that they should feel free to let you know when things seem out of the ordinary.

Some users will just live with pop-up ads, browser toolbars and slow computers. I have had others blame themselves or think they will get in trouble for surfing to the wrong sites. With each alert I do send out, I usually get a few responses about unrelated issues, phishing messages, etc. On the bright side, at least I know some people read the alert.

Finally, make one last check before it goes out the door for jargon and reread your summary. It is the only thing some people will read.

Summary

Hopefully, you found a few ideas that can save time and help protect your organization better. Security intelligence, along with knowledge of organizational assets and priorities, will allow you to continually improve your defenses in response to the rapidly changing threats we face. This paper is highly focused on the tips and tricks I have noticed over the years which allow me to perform the security intelligence function more efficiently. If you are interested in more information about the security intelligence process itself, take a look at a presentation from Maarten Van Horenbeeck on the subject.[16]

Security Intelligence is really an input to the active, ongoing process of securing our networks. It is about knowing what is "normal", whether that is "normal" network traffic or the "normal" processes and files on an end user computer. Clint Kreitner, the President of the Center for Internet Security, summed it up rather well in a recent editorial note regarding whether compliance laws were improving security:

> "I'm hoping to live long enough to see greater realization that pursuing compliance as an end in itself is hypocrisy. Instead, I'd like to see us track trends in security outcomes in terms of frequency and impact of security incidents and then work back upstream in the process chain to correlate those incidents with use or non-use of various security practices. Only then will we have a rational basis for informing our security."[17]

Trends and incidents show us when security is working. When security is failing to meet the organization's intended goal of reducing risk, we have to reevaluate our controls with the benefit of new security intelligence information. Just imagine what improvements could be made if we spent anywhere near the effort investigating the security implications of IT projects as we do the compliance issues.

That brings me to my final tips. The effort required to run a security intelligence program is worth it. As shown here, it does not need to be a formal process or full-time position to be useful. When you are successful, make sure you let people know about it once in awhile. Organizations make an investment in security and compliance, and they like to know what they are getting in return from that investment.

Better information leads to better decision making. Ideally, decision makers would receive the threat and trend information they needed in advance to budget for and implement appropriate controls before

[16] http://www.daemon.be/maarten/VanHorenbeeck-Information_Security_Intelligence_20070618.pdf, Visited 5/2/08
[17] http://www.sans.org/newsletters/newsbites/newsbites.php?vol=10&issue=45, Visited 6/9/08

a crisis hits. In the past few years, there has been more than enough information available to have foreseen the attack vectors moving from scan and exploit publicly available network level services to web-based, email-based and application level attacks. By looking at today's trends and feeding that information in the appropriate format to decision makers, new systems, controls, purchases and development projects can all be affected in a positive way.